

РОССИЙСКАЯ АКАДЕМИЯ НАУК
ИНСТИТУТ ПРОГРАММНЫХ СИСТЕМ

УДК 004.7

№ госрегистрации

10002-251/ОИТВС-04/103-111/260503-202

УТВЕРЖДАЮ

И.о. директора ИПС РАН

_____ С. М. Абрамов
«___» _____ 2004 г.

ОТЧЁТ
О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

СРАВНИТЕЛЬНОЕ ИССЛЕДОВАНИЕ ПАРАМЕТРОВ И
ХАРАКТЕРИСТИК ПЕРСПЕКТИВНЫХ
АППАРАТНО-ПРОГРАММНЫХ АРХИТЕКТУР КЛАСТЕРНЫХ
МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ

по теме:

“Оптимизация вычислительных архитектур под конкретные классы задач, информационная безопасность сетевых технологий” Программы фундаментальных научных исследований ОИТВС РАН “Оптимизация архитектур микропроцессоров и коммуникационных систем в кластерных мультипроцессорных системах”

в части:

“Разработка технических решений для экономически эффективных телекоммуникационных компьютерных сетей учреждений науки и образования, расположенных в сельской местности России”

Руководитель темы

_____ С. М. Абрамов

Ответственный исполнитель

_____ Ю. В. Шевчук

Переславль-Залесский 2004

Список исполнителей

1. Абрамов Сергей Михайлович, д.ф.-м.н., директор Исследовательского центра мультипроцессорных систем
2. Шевчук Юрий Владимирович, к.т.н., зав. лабораторией телекоммуникаций “Ботик”
3. Пономарев Александр Юрьевич, ведущий инженер
4. Котельников Виктор Петрович, ведущий инженер
5. Скосырев Олег Вильямович, инженер
6. Ивахненко Игорь Александрович, инженер
7. Гайдар Люся Алексеевна, ведущий инженер программист
8. Чередниченко Артем Владимирович, инженер 2 категории
9. Файзуллин Расим Тимурланович, инженер 2 категории
10. Балабанов Сергей Несторович, техник
11. Безрук Андрей Алексеевич, инженер
12. Бурчу Сергей Владимирович, инженер 2 категории
13. Карлаш Анастасия Владимировна, инженер
14. Парменова Вера Валерьевна, инженер
15. Ермилова Екатерина Вячеславовна, инженер 2 категории
16. Жбанов Павел Геннадьевич, инженер 2 категории
17. Несторов Александр Сергеевич, инженер
18. Гумин Михаил Владимирович, инженер 2 категории

19. Кузнецов Антон Александрович, инженер
20. Якубов Борис Валерьевич, инженер
21. Васильева Маргарита Евгеньевна, техник
22. Скосырева Виктория Вильямовна, инженер
23. Сидорина Нина Васильевна, техник
24. Белышев Дмитрий Владимирович, инженер
25. Чудновский Максим Александрович, инженер

Реферат

В данном отчете описаны результаты полученные в 2004 году по НИР “Разработка технических решений для экономически эффективных телекоммуникационных компьютерных сетей учреждений науки и образования, расположенных в сельской местности России”, выполняемой в рамках исследования “Сравнительное исследование параметров и характеристик перспективных аппаратно-программных архитектур кластерных мультипроцессорных систем” из раздела № 2 “Оптимизация архитектур микропроцессоров и коммуникационных систем в кластерных мультипроцессорных системах” Программы фундаментальных научных исследований ОИТВС РАН “Оптимизация вычислительных архитектур под конкретные классы задач, информационная безопасность сетевых технологий”.

Работа выполнялась коллективом, имеющим отлаженный комплекс решений для построения экономически эффективных сетей в условиях городской застройки. Для переноса этого опыта в условия сельской местности были выполнены перечисленные ниже работы.

1. Изучена специфика сельской местности с точки зрения построения компьютерных сетей. Она состоит в следующем:

- (а) большие расстояния между подключаемыми к сети объектами;
- (б) отсутствие легкодоступных мест в зданиях для установки магистрального оборудования (в городских условиях используются подъезды, технические этажи зданий);
- (с) ненадежность и низкое качество электропитания.

2. Выбран набор решений для построения сети:

- (а) Использование для передачи данных технологии RadioEthernet (IEEE 802.11, WiFi), а при отсутствии прямой видимости между объектами — одномодовых волоконнооптических линий связи (укладка в траншею или подвеска на столбы).

- (b) Магистральное оборудование монтируется в термоизолированных, вандалоустойчивых контейнерах (*термобоксах*), размещаемых на стандартных бетонных осветительных столбах. На тех же столбах монтируются антенны, что обеспечивает прямую видимость между соседними магистральными узлами, необходимую для работы RadioEthernet на больших расстояниях.
 - (c) В качестве абонентского комплекта оборудования используются мосты RadioEthernet с удешевленной антенно-фидерной системой.
3. Проведено изучение текущего состояния рынка, выбраны компоненты для построения магистральных узлов и абонентских комплектов:
- (a) Оборудование магистрального узла строится на основе ПК-маршрутизатора с процессором Pentium III. Разработана конструкция ПК-маршрутизатора с низким энергопотреблением, что позволило исключить из конструкции ненадежные механические элементы (вентиляторы охлаждения), а также снизить требования (по мощности) к системе электропитания узла.
 - (b) В качестве радиоинтерфейсов магистрального узла используются сетевые адаптеры с интерфейсом PCI на основе чипсета Atheros AR5212, выпускаемые несколькими фирмами (D-Link, Planet);
 - (c) В качестве радиооборудования в абонентском комплекте нами используются устройства D-Link 900AP+, 2000AP+, Linksys WRT54G.
4. Разработана система электропитания магистрального узла, обеспечивающая работоспособность ПК-маршрутизатора при напряжении питающей сети от 100 до 300В, и автономное (от аккумулятора) питание магистрального узла в течение не менее 8 часов.
5. Разработана конструкция термоизолированного, вандалоустойчивого контейнера с внешними размерами $600 \times 400 \times 450$ мм. Контейнер разделен на две части горизонтальной внутренней перегородкой. Верхняя часть термоизолирована, в ней размещается ПК-маршрутизатор и другое оборудование, не способное работать при отрицательных температурах. В нижней части размещается аккумулятор и один из двух блоков системы электропитания. Для обогрева верхней камеры предусмотрен подогреватель, для охлаждения — вентилятор, обеспечивающий воздухообмен между верхней и нижней камерами. Температурный режим в верхней камере поддерживается термоконтроллером, специально разработанного для данного проекта, на базе микроконтроллера Atmel ATtiny26.

6. Разработан практический метод выбора точек для установки (или выбора) высотных сооружений для обеспечения прямой видимости между антеннами. Используется гелиевый аэростат яркой расцветки объемом 3 куб. м., и автовышка, обеспечивающая подъем наблюдателя на высоту до 18 м.
7. Освоено использование столбов бетонных осветительных для размещения антенн и оборудования магистральных узлов компьютерной сети.
8. Реализован пилотный сегмент сети, содержащий шесть магистральных узлов, два магистральных волоконнооптических сегмента, четыре сегмента магистральных беспроводных линий передачи данных, и обслуживающий семь абонентов, в том числе Международный детский компьютерный центр им. А. К. Айламазяна ИПС РАН (урочище Кухмарь), школу в селе Купанское, музей железнодорожной техники в селе Талицы Переславского района Ярославской области.
9. Работы по программному обеспечению:
 - (a) разработана новая версия системы администрирования сети Nadmin;
 - (b) разработан модифицированный вариант (MON/Botik) свободной системы мониторинга MON;
 - (c) разработана специализированная геоинформационная система BotikMap, работающая во взаимодействии с системами Nadmin и MON/Botik;
 - (d) доработан драйвер madwifi для обеспечения мониторинга пакетов MAC-уровня 802.11;
 - (e) разработана программа микроконтроллера блока 220–12, входящего в систему электропитания магистрального узла.

Материалы проекта представлены на конференции “Научный сервис в сети Интернет”, Абрау-Дюрсо, 2004 (5 работ, [11, 12, 13, 14, 15]) и на Международной конференции “Программные системы: теория и приложения”, Переславль-Залесский, май 2004 (4 работы, [7, 8, 9, 10]).

Отчет содержит 243 страницы, 24 таблицы, 70 рисунков, 18 источников.

Содержание

Введение	21
1 Региональные сети для сельской местности	22
1.1 Общее описание, технические решения	22
1.1.1 Особенности создания сетей в сельских районах	22
1.1.2 Технические решения для построения протяженных линий связи .	23
1.1.3 Технические решения для размещения оборудования связи вне помещений	27
1.1.4 Технические решения для учета низкого качества электропитания в сельских районах России	28
1.2 Общая структура компьютерной сети в сельской местности. Основные компоненты	29
1.2.1 Магистральные линии, созданные по технологии беспроводной связи	30
1.2.2 Радиосоты, узлы беспроводного доступа к сети абонентов	31
1.2.3 Магистральные волоконнооптические линии связи	32
1.2.4 Абонентский комплект беспроводного доступа	33
1.2.5 Абонентский комплект доступа по технологии Ethernet	35
1.3 Определение на местности условий прямой видимости	35
1.4 Пилотный проект компьютерной сети в сельской местности	38
2 Типовой магистральный узел	53
2.1 Общее описание	53
2.2 Маршрутизатор	53
2.2.1 Надежность маршрутизатора	55
2.2.2 Дисковая подсистема маршрутизатора	56
2.2.3 Снижение потребляемой мощности	56
2.2.4 Программное обеспечение маршрутизатора	61
2.3 Средства радиодоступа	61

2.3.1	Выбор типа радиокарт	61
2.3.2	Опыт работы с радиокартами Atheros	63
2.3.3	Антенны	66
2.3.4	Кабели и разъемы	67
2.4	Термобокс	67
2.4.1	Технические характеристики	69
2.5	Подсистема электропитания узла	77
2.6	Блок 220–12	78
2.7	Аккумуляторная батарея	83
2.8	Блок 12–ATX	85
2.9	Термоконтроллер	89
3	Абонентский комплект беспроводного доступа	91
3.1	Общее описание	91
3.2	Выбор решения	92
3.2.1	Установка сетевого адаптера беспроводного доступа в компьютер абонента	92
3.2.2	Установка ПК-маршрутизатора у абонента	92
3.2.3	Использование мостов 802.11–802.3	94
3.3	Выбор оборудования для мостов 802.11–802.3	94
3.4	Обеспечение безопасности	95
3.4.1	Выбор оборудования с поддержкой WPA	96
3.4.2	Уязвимости WEP	97
3.4.3	Уязвимости механизмов аутентификации	98
3.4.4	Выводы	99
3.5	Программное обеспечение. Установка и конфигурирование	99
4	ПО сервера аренды www-пространства	103
4.1	Общее описание	103
4.2	Руководство системного программиста	106
4.2.1	Описание основных компонент	106
4.2.2	Установка	107
4.2.3	Эксплуатация сервера аренды	110
4.3	Руководство пользователя	110
4.3.1	Доступ	111
4.3.2	Использование Mysql	111
4.3.3	Использование Apache	111

5 Система администрирования Nadmin для гражданских сетей	113
5.1 Описание программы	113
5.1.1 Общее описание	113
5.1.2 Архитектура системы	115
5.1.3 Сенсоры	117
5.1.4 База данных	126
5.1.5 Архивация данных	132
5.1.6 Расчетная подсистема	133
5.1.7 Процесс обновления системной конфигурации	145
5.1.8 Программы синхронизации баз данных систем Nadmin	146
5.1.9 Журнал событий и класс Nadmin::Eventlog	150
5.1.10 Конфигурационный файл config	152
5.1.11 Вспомогательный класс Nadmin::Lock	154
5.1.12 Вспомогательный класс Nadmin::YMDHMS	155
5.2 Руководство администратора гражданской сети	157
5.2.1 Web-интерфейс администратора	157
5.3 Руководство абонента	161
5.3.1 Web-интерфейс абонента	161
6 ГИС BotikMap	166
6.1 Общее описание	166
6.2 Руководство системного программиста	167
6.2.1 Сервер	167
6.2.2 Nadmin::GisObject.pm	168
6.2.3 Временная БД	169
6.2.4 Web-приложение giserver.cgi	170
6.2.5 Клиент	172
6.2.6 Взаимодействие Nadmin и ГИС	172
6.3 Руководство администратора гражданской сети	173
6.4 Руководство пользователя	173
6.4.1 Создание объектов	173
6.4.2 Основы работы в BotikMap	174
7 Система мониторинга MON/Botik	178
7.1 Мониторная система MON/Botik. Описание программы	178
7.1.1 Ключевые слова и термины	180
7.1.2 Архитектура системы MON. Компоненты системы	180

7.1.3	Процесс работы системы MON	181
7.2	Руководство системного программиста	183
7.2.1	Модификации основного процесса системы MON	183
7.2.2	Дополнительные модули для построения сайта состояния сети	184
7.2.3	Система оповещения с обратной связью	186
7.3	Руководство пользователя	188
7.3.1	Общее описание	188
7.3.2	Конфигурирование системы	188
7.3.3	Параметры, используемые при запуске основного процесса системы	190
7.3.4	Использование мониторов	190
7.3.5	Использование системы оповещения с обратной связью	191
7.3.6	Настройка crontab в случае использования Web-сайта и оповещения с помощью sms-сообщений	192
7.3.7	Использование Web-сайта системы MON	193
Список использованных источников		196
A	Система администрирования Nadmin	199
A.1	Сенсор IP-статистики	199
A.1.1	Запуск сенсора IP-статистики	199
A.1.2	Структура иерархии каталогов для выходных файлов утилиты tcpdump	200
A.1.3	Работа сенсора IP-статистики	200
A.1.4	Функции программы sensor_ip.pl	201
A.2	Класс Nadmin::Account	205
A.2.1	Функциональность класса Nadmin::Account	205
A.2.2	Методы класса Nadmin::Account	208
B	Блок 220–12	212
B.1	Расчеты характеристик	212
B.1.1	Технические требования на Блок 220–12	212
B.1.2	Расчет импульсного блока питания для UPS 50Вт	212
B.2	Перечень элементов	218
B.3	Принципиальная схема	221
C	Программы микроконтроллера Блока 220–12	222
C.1	Контроль потребляемого тока и защита от перегрузки	222
C.2	Подстройка напряжения зарядки аккумулятора	223

C.3 Отключение нагрузки при разряде аккумулятора	223
C.4 Индикация режима работы Блока 220–12	224
C.5 Управляющий интерфейс	224
C.6 Калибровка	227
D Блок 12–ATX	229
D.1 Расчеты характеристик	229
D.1.1 Индуктивности	229
D.1.2 Конденсаторы	229
D.1.3 Расчет цепи коррекции, R_c , C_c	230
D.1.4 КПД для преобразователя 12В–3.3В при токе 10А (мощность на нагрузке 33Вт)	232
D.1.5 Расчет делителя обратной связи для IRU3037	233
D.1.6 Потери в верхнем транзисторе	234
D.2 Перечень элементов	234
D.3 Принципиальная схема	237
E Термоконтроллер	238
E.1 Перечень элементов	241
E.2 Принципиальная схема	243

Список иллюстраций

1.1	Структура компьютерной сети в сельской местности	29
1.2	Адаптер беспроводной связи D-Link DWL-G520	30
1.3	Антenna направленная параболическая ГРАД-2497 (слева) и антenna штыревая ГРАД-2401 (справа)	31
1.4	Медиа-конвертеры FastEthernet 100BaseTX — 100BaseFX (fiber optic, single mode, разъем SC) Planet FT-702S15 и Planet FT-802S15	33
1.5	D-Link 900AP+	35
1.6	Схема определения наличия прямой видимости при помощи метеорологического шара	36
1.7	Шар метеорологический	37
1.8	Схема пилотного проекта беспроводной сети	39
1.9	Магистральный узел в термобоксе на крыше ИПС РАН	40
1.10	Топографический план пилотного проекта	41
1.11	Использование мачты собственной разработки для организации магистрального узла в селе Криушкино	42
1.12	Урочище Кухмарь: магистральный узел расположен в котельной, для размещения антенны (и обеспечения прямой видимости) используется существующее высотное сооружение (труба котельной)	43
1.13	Село Купань: магистральный узел размещен вне помещения, на специально установленном для этого осветительном бетонном столбе с дополнительной мачтой	44
1.14	Поселок Купанскоe: магистральный узел (он же узел доступа) размещен вне помещения, на специально установленном для этого осветительном бетонном столбе	45
1.15	Школа в поселке Купанскоe. Параболическая антenna беспроводного подключения ЛВС школы к компьютерной сети	46
1.16	Музей паровозов: размещение магистрального волоконнооптического узла	47

2.1	ПК-маршрутизатор	55
2.2	Файловые системы ПК-маршрутизатора	56
2.3	Процессор РIII 666 МГц работает на пониженной частоте	60
2.4	Пример выдачи статистики драйвера <code>madwifi</code>	65
2.5	Пример выдачи изменений (с интервалом 1 с), собранной статистики драйвера <code>madwifi</code>	66
2.6	Информация о клиентах, ассоциированных с базовой станцией	67
2.7	Дампирование трафика на радиоинтерфейсе	68
2.8	Получение дампа пакетов уровня 802.11 путем перевода интерфейса в режим Monitor	68
2.9	Мониторинг пакетов и регулировка антенны для максимизации уровня принимаемого сигнала	69
2.10	Схема термобокса и расчетная зависимость мощности, расходуемой на поддержание рабочей температуры внутри контейнера от температуры среды	70
2.11	Термобокс: вид спереди / сзади	70
2.12	Термобокс: вид внутри	71
2.13	Термобокс: 3D-схема	74
2.14	Термобокс: вид сбоку	75
2.15	Термобокс: вид спереди	76
2.16	Структурная схема системы электропитания	78
2.17	Блок 220–12	80
2.18	Аккумуляторная батарея	83
2.19	Блок 12–ATX	85
3.1	Конфигурация D-Link DWL-2000AP+: установки в меню Home/Wireless	100
3.2	Конфигурация D-Link DWL-2000AP+: установки в меню Home/Lan	100
3.3	Конфигурация D-Link DWL-2000AP+: установки в меню Home/DHCP	101
3.4	Конфигурация D-Link DWL-2000AP+: установки в меню Advanced/Mode	101
3.5	Конфигурация D-Link DWL-2000AP+: установки в меню Advanced/Filters	101
3.6	Конфигурация D-Link DWL-2000AP+: установки в меню Tools/Admin	101
3.7	Конфигурация D-Link DWL-2000AP+: установки в меню Advanced/Performance	102
4.1	Пример конфигурации для GrSecurity через sysctl	112
5.1	Архитектура системы Nadmin	115

5.2 Схема работы сенсоров с log-файлами и ОПО	118
5.3 Сенсоры: ротация log-файлов	119
5.4 Сенсоры: сохранение закладок	121
5.5 Порция статистики	122
5.6 База данных: структура каталогов и файлов	127
5.7 Иерархия классов Perl-интерфейса базы данных	128
5.8 Структура архива порций статистики	132
5.9 Основной процесс-обработчик	134
5.10 Структура порции статистики	136
5.11 Сообщения ОПО при запуске в режиме <code>verbose-mode</code>	139
5.12 Web-интерфейс администратора: главная страница	157
5.13 Web-интерфейс администратора: страница поиска	158
5.14 Web-интерфейс администратора: страница “Абоненты”	159
5.15 Web-интерфейс абонента: проверка прав доступа	162
5.16 Web-интерфейса абонента: главная страница	163
5.17 Web-интерфейса абонента: лицевой счет	164
7.1 Архитектура системы MON	181
7.2 Схема оповещения с обратной связью	187
7.3 Пример конструкции <code>period</code>	188
7.4 Использование конструкции <code>unacked</code>	189
7.5 Запись в <code>crontab</code> для архивирования log-файлов и использования поповещания по SMS	192
7.6 Пример конфигурационного файла системы MON	194
7.7 Пример файла, сгенерированного для построения Web-сайта	195
B.1 Принципиальная схема Блок 220–12	221
D.1 Принципиальная схема Блока 12–ATX	237
E.1 Принципиальная схема термоконтроллера	243

Список таблиц

1.1	Сравнение волоконнооптических и беспроводных решений для передачи данных	24
1.2	Стоимость организации ВОЛС (по технологии Ethernet)	25
1.3	Стоимости высотных сооружений для монтажа беспроводных сетей . . .	26
1.4	Сравнительные характеристики различных радиочастотных кабелей, ис- пользуемых в данном проекте в системах беспроводного доступа	34
1.5	Тактико-технические характеристики шара метеорологического	36
1.6	Медиа-конвертеры Planet FT-702S15 и Planet FT-802S15. Технические ха- рактеристики	49
1.8	Таблица основных тактико-технических характеристик адаптера беспро- водной связи D-Link DWL-G520	51
1.9	Тактико-технические характеристики антенн ГРАД-2401, ГРАД-2492 и ГРАД-2497	52
2.1	Состав аппаратных средств типового магистрального узла	54
2.2	Потребляемая мощность ПК-маршрутизаторов с различными типами про- цессоров	58
2.3	Технические характеристики материала Пенофол® типа “А”	73
2.4	Технические характеристики Блока 220–12	79
2.5	Предельные эксплуатационные данные Блока 220–12	79
2.6	Технические характеристики Блока 12–ATX	86
2.7	Предельные эксплуатационные данные Блока 12–ATX	86
3.1	Оценка стоимости малогабаритного маршрутизатора	93
3.2	Сравнение различного оборудования для мостов 802.11—802.3	96
6.1	Временная база данных ГИС BotikMap. Таблица GO	170
6.2	Временная база данных ГИС BotikMap. Таблица U_INFO	170

C.1 Коррекция выходного напряжения в зависимости от температуры аккумулятора	223
C.2 Коррекция выходного напряжения в зависимости от тока нагрузки	224
E.1 Технические характеристики термоконтроллера	238
E.2 Предельные эксплуатационные данные термоконтроллера	239

Определения

В тексте работы используются следующие термины:

Базовая станция беспроводного доступа — устройство, находящееся в центре радиосети и выполняющее функции управления и сетевого моста между беспроводными клиентами и проводной сетью.

Блок 220–12 — часть системы электропитания магистрального узла, обеспечивающая зарядку аккумулятора и питание блока 12-ATX.

Блок 12–ATX — часть системы электропитания магистрального узла, выполняющая преобразование напряжения 12В в напряжения питания системной платы персонального компьютера стандарта ATX.

Магистральный узел — элемент компьютерной сети, соединенный каналами связи с другими магистральными узлами и обеспечивающий подключение к сети нескольких локальных сетей абонентов.

ПК-маршрутизатор — TCP/IP маршрутизатор, построенный на базе персонального компьютера.

Радиокарты — сетевые адаптеры беспроводного доступа к компьютерным сетям стандартов 802.11a/b/g.

Радиосота — окрестность базовой станции беспроводного доступа, в которой возможна работа беспроводных клиентов.

Термобокс — вандалоустойчивый металлический контейнер собственной разработки, предназначенный для размещения оборудования связи вне помещения (outdoor). Термобокс состоит из двух камер (рис. 2.10):

- **Верхняя камера термобокса** — термоизолированная камера, в которой размещается оборудование, для которого необходимо выдерживать рабочий диапазон температур.

- **Нижняя камера термобокса** — нетермоизолированная камера, где находится оборудование, для работы которого температурный режим не критичен.

Термоконтроллер — электронное устройство, управляющее поддержанием заданной температуры в термобоксе.

Фидерный тракт — кабельная линия, соединяющая антенну с приемопередатчиком.

BotikKey — система подтверждения права доступа Абонента к Сети в СТ “Ботик”.

BotikMap — специализированная гео-информационная система, предназначенная для графического представления информации из базы данных системы администрирования сети Nadmin.

MON — система для наблюдения за доступностью сервисов в компьютерной сети и отправки сообщений при обнаружении неисправности.

Nadmin — административно-информационная система СТ “Ботик”.

Обозначения и сокращения

АЦП — аналого-цифровой преобразователь;

БД — база данных;

ГИС — гео-информационная система;

ЛВС — локальная вычислительная сеть;

ОПО — основной процесс-обработчик;

ПК — персональный компьютер;

ПЭВМ — персональная электронная вычислительная машина;

СТ — система телекоммуникаций;

ШИМ — широтно-импульсная модуляция;

ЭСППЗУ — электрически стираемое перепрограммируемое постоянное запоминающее устройство;

ATA — Advanced Technology Attachment — серия интерфейсов и протоколов, используемых для организации доступа к жестким дискам;

CGI — Common Gateway Interface — одна из технологий динамической генерации WWW-страниц;

CSMA — Carrier-Sense Multiple Access — многостанционный (сетевой) доступ с контролем несущей;

DNS — Domain Name System — служба доменных имен в сети Internet;

ICMP — Internet Control Message Protocol — протокол управляющих сообщений в сети Internet;

LAN — Local Area Network — локальная вычислительная сеть;

- NFS — Network File System — протокол доступа к файловым системам удаленных компьютеров в сети Интернет;
- OFDM — Orthogonal Frequency Division Multiplexing — ортогональное частотное мультиплексирование (метод модуляции);
- PCI — Peripheral Component Interconnect — локальная шина соединения периферийных устройств персональных компьютеров;
- PG — сигнал PowerGood — “напряжение в норме”;
- POP3 — Post Office Protocol version 3 — протокол передачи почты с почтового сервера конечному пользователю в сети Internet;
- RCS — Revision Control System — система хранения версий и исходных текстов программ;
- RTS — Request-To-Send — запрос разрешения на передачу;
- RTT — Round Trip Time — задержка в канале при эхо-тестировании;
- SMTP — Simple Mail Transfer Protocol — протокол передачи электронной почты в сети Internet;
- SSH — Secure Shell — протокол и приложение, использующее криптографию с открытым ключом для организации защищенного удаленного доступа в сети Internet;
- UDP — User Datagram Protocol — протокол транспортного уровня Интернет с пакетным режимом передачи;
- WEP — Wired Equivalent Privacy — средства шифрации передаваемых данных, предусмотренные в стандарте беспроводных сетей 802.11;
- WPA — Wireless Protected Access — протокол защищенного беспроводного доступа, принятый комитетом WiFi.

Введение

В данном отчете описаны результаты полученные в 2004 году по НИР “Разработка технических решений для экономически эффективных телекоммуникационных компьютерных сетей учреждений науки и образования, расположенных в сельской местности России”, выполняемой в рамках исследования “Сравнительное исследование параметров и характеристик перспективных аппаратно-программных архитектур кластерных мультипроцессорных систем” из раздела № 2 “Оптимизация архитектур микропроцессоров и коммуникационных систем в кластерных мультипроцессорных системах” Программы фундаментальных научных исследований ОИТВС РАН “Оптимизация вычислительных архитектур под конкретные классы задач, информационная безопасность сетевых технологий”. В работе:

1. рассмотрены особенности, структура и основные компоненты компьютерных сетей в сельских районах России, созданный pilotный сегмент такой сети (Глава 1);
2. описаны технические решения для реализации *магистрального узла* сельской компьютерной сети (Глава 2, Приложения В, С, Д, Е);
3. рассмотрены различные варианты реализации *абонентского комплекса беспроводного доступа* для сельской компьютерной сети (Глава 3);
4. дано описание программных средств, используемых при эксплуатации сельских компьютерных сетей:
 - комплект программного обеспечения для реализации сервера аренды www-пространства (Глава 4);
 - система администрирования Nadmin для региональных компьютерных сетей (Глава 5 и Приложение А);
 - специализированная гео-информационная система BotikMap (Глава 6);
 - система мониторинга MON/Botik для региональных компьютерных систем (Глава 7).

Глава 1

Региональные компьютерные сети для сельской местности России

1.1 Общее описание, технические решения

1.1.1 Особенности создания сетей в сельских районах

Лаборатория Ботик с 1994 года занимается созданием региональных сетей для городской застройки, и большинство вопросов, возникающих при построении городских сетей, решены. Сельская местность России обладает рядом особенностей, которые не позволяют применить для создания сельских компьютерных сетей те технологии, которые отработаны в городских условиях.

1. Первой из особенностей сельской местности России является *низкая плотность застройки*, большие расстояния между строениями, а также достаточно большие расстояния между поселениями. Требуется искать *технологии для построения каналов связи большой протяженности*, которые позволяли бы за разумные средства покрывать большие расстояния. Поэтому здесь в первую очередь речь пойдет о беспроводных технологиях передачи данных.
2. Вторая особенность: *большинство строений находится в частной собственности*, в отличие от города, где большинство строений находятся в муниципальной собственности или, по крайней мере, в зданиях есть места общего пользования, доступ к которым не затруднен: подъезды, чердаки, лестничные площадки и т. д. В местах общего пользования легко размещать оборудование связи для компьютерных сетей в городских условиях. В сельских условиях всё это недоступно, поэтому *потребуется решать задачу установки оборудования связи вне помещений (outdoor)*.

3. Третья особенность — это низкое качество электропитания в сельских районах России. Здесь возможны большие перепады и скачки напряжения в течение суток при включении или отключении крупных потребителей. Отметим и низкую защищенность сельских электросетей от наводок (в том числе и во время грозовых разрядов), что так же приводит к большим скачкам напряжения. Тем самым, в сельской местности России в электросети вместо положенного напряжения 220В вполне может быть от 150В до 300В и скачки напряжения до 500В. Так же довольно часто в сельских электросетях наблюдаются длительные перерывы в электроснабжении в связи с большой изношенностью сельских электросетей и частыми авариями.

Из анализа данных трех особенностей ясно, что нам придется подбирать (среди доступных) или разрабатывать следующие технические решения (подробнее обсуждено в последующих разделах):

- Технические решения для построения протяженных линий связи (раздел 1.1.2).
- Технические решения для размещения оборудования связи вне помещений (раздел 1.1.3).
- Технические решения для учета низкого качества электропитания в сельских районах России (раздел 1.1.4).

1.1.2 Технические решения для построения протяженных линий связи

На сегодняшний день существует две широко применяемые технологии, позволяющие создавать линии связи большой протяженности — волоконнооптические и беспроводные линии передачи данных.

Конечно, известны разработки для линий передачи данных большой протяженности, построенные и на других принципах¹. Однако эти технологии применяются не так широко и соответствующие изделия выпускаются не массово. Поэтому они весьма дороги и мало доступны на российском рынке. По данной причине мы считаем целесообразным их исключить из рассмотрения, при подборе технических решений для сельских компьютерных сетей России. Дополнительной причиной отказа от рассмотрения “редких” решений служит то, что, как правило, они обеспечивают передачу данных на не очень большие расстояния (около 2–4 км), а также качество связи в таких каналах весьма чувствительно к различным атмосферным явлениям.

¹ В основном используются “проводные” решения: xDSL, Grunch, HomePna и другие.

Таблица 1.1: Сравнение волоконнооптических и беспроводных решений для передачи данных

Технология	Сильные стороны	Слабые стороны
Волоконно-оптические линии связи	<ul style="list-style-type: none"> – Высокая скорость передачи данных: 10–100–1000 Мбит/с – Большое расстояние: 15–30–45 км (и более) – Внешние условия (включая атмосферные явления) не влияют на качество передачи данных 	<ul style="list-style-type: none"> – Высокая цена кабеля и оконечных устройств – Большие капитальные затраты на организацию каналов
Беспроводные линии связи	<ul style="list-style-type: none"> – Низкая цена оборудования – Быстрая развертывания и возможность переноса оборудования на другую линию – Достаточно большая ёмкость каналов: 2–5–11 Мбит/с (и более) 	<ul style="list-style-type: none"> – Необходимое условие — прямая видимость между антеннами приемника и передатчика – Если для обеспечения прямой видимости требуется построение высотных сооружений (для размещения антенн), то это приводит к серьезному увеличению и срока реализации и цены проекта создания линии связи – Необходимость лицензирования в ГКРЧ (большие затраты и большие сроки разрешительных мероприятий)

Таким образом, в дальнейшем мы будем рассматривать только решения, основанные на волоконнооптических линиях связи (ВОЛС), и беспроводные решения. В табл. 1.1 показаны сильные и слабые стороны данных технологий.

Подвеска волоконноптического кабеля на столбы. При условии наличия столбового хозяйства вдоль трассы ВОЛС организация одного километра ВОЛС методом подвески волоконноптического кабеля на столбы обходится от \$2 000 (две жилы) до \$6 000 (8 жил), в зависимости от количества жил в кабеле (табл. 1.2). Эти расходы включают в себя стоимость самого кабеля, столбового крепежа, аренду автомобиля с подъемником и оплату работ. Однако в сельской местности столбовое хозяйство развито слабо — как правило, или отсутствует или не приспособлено к подвесу подобного типа кабелей. Конечно, при отсутствии столбового хозяйства, его можно построить, однако это удорожит проект создания ВОЛС примерно на \$5000 на каждый километр (табл. 1.3).

Укладка волоконноптического кабеля в грунт. Вторым способом организации ВОЛС является укладка кабеля в грунт. При использовании специальной техники (ка-

Таблица 1.2: Стоимость организации ВОЛС (по технологии Ethernet)

Тип расхода	Стоимость
Организация 1 км волоконнооптической линии способом подвеса на столбы при наличии столбового хозяйства (расстояние между столбами — 30 м) вдоль трассы	\$2 000 – \$6 000
Организация 1 км волоконнооптической линии способом подвеса на столбы при отсутствии столбового хозяйства	\$7 000 – \$11 000
Организация 1 км волоконнооптической линии способом укладки оптической линии в грунт	\$5 000 – \$6 000
Оконцовка и организация передающего оборудования на двух концах линии при скорости передачи 100 Мбит/с	\$230
Оконцовка и организация передающего оборудования на двух концах линии при скорости передачи 1000 Мбит/с	\$300

белеукладчик, позволяющий укладывать волоконнооптический кабель в грунт без предварительной подготовки траншеи) затраты на организацию 1 км ВОЛС составляют от \$5 000 (8 жил) до \$6 000 (16 жил), в зависимости от количества жил в кабеле (см. табл. 1.2). Эти расходы включают в себя стоимость самого кабеля и оплату работ по укладке кабеля в грунт.

Оконечное оборудование ВОЛС. При реализации ВОЛС помимо расходов на закупку и прокладку волоконнооптического кабеля необходимо учитывать расходы связанные с оконцовкой кабеля (разъемы, организаторы кабеля, splices, pigtails, patchcords, расходные материалы) и активным оборудованием — как правило это трансиверы (медиа-конверторы) из витопарного Ethernet в волоконнооптический. Данные расходы зависят от требуемой скорости передачи данных (см. табл. 1.2). Данные в таблице указаны для одномодового кабеля и для длин ВОЛС до 15 км.

Расходы при организации магистральных узлов беспроводных линий связи. Стоимость создания узла магистрали на радио линиях не зависит от расстояний и составляет примерно \$1 000 за узел. В указанную стоимость не включаются затраты на лицензирование, которые могут быть достаточно существенны².

Условие прямой видимости и высотные сооружения для беспроводных линий связи. Самой большой проблемой для беспроводных технологий является тре-

²Из практики известно, что процесс лицензирования может длится около 9 месяцев и затраты на данные работы могут составлять около \$6 000.

Таблица 1.3: Стоимости высотных сооружений для монтажа беспроводных сетей

Сооружение	Высота, м	Стоимость, руб
Вышка	30	$\approx 1\ 000\ 000$ (с доставкой и установкой)
Бетонный столб	9	5 000
Бетонный столб удлиненный	14	7 000
Мачта металлическая (собственной разработки)	12	9 000

ование наличия прямой (оптической) видимости между двумя антеннами на концах линии связи³. Если не удается использовать какие-либо высотные строения или особенности местности и за счет этого выполнить условие обеспечения прямой видимости, то приходится прибегать к установке вышек для размещения антенн радиооборудования. Однако организация таких вышек связана с дорогостоящими конструкциями и строительными работами. Тем самым, если не удается избежать строительства таких вышек, то дешевизна организации беспроводной линии сводится на нет (см. табл. 1.3).

В рамках проведения данного проекта мы попробовали использовать установку обыкновенных бетонных осветительных столбов вместо применения дорогостоящих вышек, стоимость установки приведена в табл. 1.3. В некоторых случаях для обеспечения необходимой высоты столб наращивался специальной конструкцией, для закрепления антенн (рис. 1.13). Кроме всего прочего столбы использовались для размещения контейнеров с оборудованием связи. Размещение данных контейнеров на вершине столба позволило:

- Уменьшить длину антенно-фидерной системы (и тем самым уменьшить потери в ней).
- Усилить сохранность оборудования связи (взламывание вандалоустойчивого контейнера на вершине столба — осложнено).

Данный опыт использования обыкновенных бетонных осветительных столбов в подобных проектах можно признать успешным.

³В упрощенном виде данное требование означает, что с точки размещения одной антенны должна быть видна другая, никакие препятствия не должны находиться на луче, соединяющем антенны. В более точной формулировке: препятствия нежелательны не только вдоль луча, но и в прилегающей к нему так называемой *зоне Френеля*.

Другие недостатки беспроводных линий связи

Использование технологии RadioEthernet (IEEE 802.11, WiFi) для организации магистральных линий связи очень привлекательно с точки зрения стоимости оборудования и отношения стоимость/производительность. С этой точки зрения конкурентов оборудованию 802.11 на сегодняшний день нет. К недостаткам его использования на магистральных линиях можно отнести следующее:

- Оборудование стандарта 802.11 полудуплексное, и алгоритм арбитража доступа к среде (MAC-уровень 802.11) рассчитан на небольшие расстояния между узлами. Используемый в протоколе временной слот (slot time) составляет 20 мкс для 802.11b и 9 мкс для 802.11g. Поскольку временной слот должен быть не меньше, чем задержка (RTT) распространения сигнала между узлами, предельным расстоянием, при котором протокол арбитража работает как предусмотрено разработчиками, составит 2 км и 900 м соответственно. Как следствие, на длинных магистральных линиях протокол арбитража работает неэффективно; часто конфликт доступа к среде разрешается за счет потери пакета с последующим таймаутом (acktimeout) и перепосылкой, и пропускная способность канала оказывается существенно ниже максимально возможной.
- Дешевизна оборудования WiFi и сегодняшняя популярность этой технологии заставляют предположить, что через несколько лет во всех населенных пунктах существенно повысится загрязненность эфира в результате работы стихийных беспроводных сетей. Их работа может создать помехи работе магистральных линий большой протяженности, которые приведут к снижению производительности магистральных линий. Когда это произойдет, можно предпринять отход от стандартного оборудования WiFi: например, использовать его с радиочастотными конвертерами⁴ (2Гц → 6Гц), использовать оборудование стандарта 802.16, либо использовать нестандартное радиорелейное оборудование с интерфейсом Ethernet. При этом потребуется заменить только магистральные радиоинтерфейсы, доля которых в общей стоимости сегодняшнего проекта составляет незначительную часть

1.1.3 Технические решения для размещения оборудования связи вне помещений

При организации компьютерной сети в городских условиях в домах и подъездах можно было устанавливать вандалоустойчивые ящики, где в тепле, не заботясь о защите от осадков, можно было устанавливать оборудование связи для компьютерных

⁴<http://www.raitek.ru>

городских сетей. В селе мы лишены таких условий. Поэтому требуется разработать универсальный вариант внешнего (outdoor) размещения сетевого оборудования. Речь идет о разработке специального контейнера — *термобокса*, — который удовлетворял бы следующим условиям:

- в специальном термоизолированном отделении для установленного в нем оборудования связи обеспечивались бы
 - температурный режим в диапазоне 0–50°C;
 - защита от осадков для размещенного в контейнере оборудования;
- размеры термобокса позволяли бы размещать в нем оборудование (*ПК-маршрутизаторы*⁵, коммутаторы, медиа-конверторы) необходимых габаритов;
- термобокс обеспечивал бы достаточную устойчивость к попыткам его взлома и другим проявлениям вандализма.

В рамках данного проекта такое изделие было разработано, оно подробно обсуждается в разделе 2.4. Отметим, что себестоимость разработанного термобокса (\$140, включая нагреватель, блок терморегуляции и вентилятор) меньше, чем у доступных на рынке изделий данного типа (\$700/435, www.diamond.ru).

1.1.4 Технические решения для учета низкого качества электропитания в сельских районах России

В сельской местности России вместо предусмотренного по стандартам напряжения 220В в электросети может быть от 150В до 300В (и скачки напряжения даже до 500В). Также достаточно часты продолжительные перерывы в электропитании.

Чтобы учесть эту особенность, мы должны предусмотреть в проекте разработку блока питания, который бы удовлетворял следующим техническим требованиям:

- рабочий диапазон входного напряжения: от 100 до 300В;
- допустимое напряжение на входе, не приводящее к выходу из строя блока питания: до 500В;
- температурный диапазон работы: от -40 до 50°C — для размещения блока питания вне термоизолированной части термобокса (в “холодной” части);

⁵ПК-маршрутизатор — маршрутизатор, построенный на базе стандартного оборудования класса IBM PC с использованием ОС Linux, подробно обсуждается в Разделе 2.2.

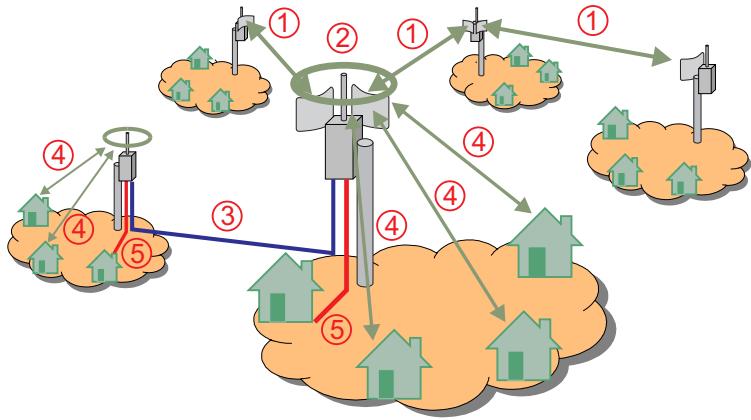


Рис. 1.1: Структура компьютерной сети в сельской местности

- обеспечение времени автономной работы полезной нагрузки: до 16-ми часов⁶

Тогда в совокупности с термобоксом такой блок питания позволял бы размещать вне помещений (outdoor) в условиях плохого электропитания все необходимое связное оборудование, которое может потребоваться при организации сельских сетей: ПК-маршрутизаторы с различными сетевыми адаптерами (как беспроводных сетей, так и адаптеры Ethernet для витой пары или ВОЛС), коммутаторы Ethernet, медиа-конвертеры (трансиверы).

1.2 Общая структура компьютерной сети в сельской местности. Основные компоненты

Исходя из приведенного выше анализа, мы получаем следующую структуру для компьютерных сетей сельских районов России, изображенную на рис. 1.1, где отражены основные компоненты этой сети (подробно обсуждаемые в последующих разделах):

1. магистральные линии, созданные по технологии беспроводной связи (позиции 1 на рис. 1.1), магистральные узлы системы (см. раздел 1.2.1);

⁶Измеренная мощность, потребляемая ПК-маршрутизатором с двумя интерфейсами 802.11 и один интерфейсом 802.3 при средней нагрузке (100% загрузка трафиком обоих беспроводных интерфейсов) составила 12.65 Вт. При потреблении такой мощности от аккумулятора напряжением 12В ток разряда составит 1.05А. При емкости аккумулятора 17Ач время автономной работы составит 16.13ч. При необходимости увеличения времени автономной работы возможно использование аккумулятора большей емкости.



Рис. 1.2: Адаптер беспроводной связи D-Link DWL-G520

2. радиосоты (позиции 2 на рис. 1.1), магистральные узлы системы с поддержкой функции базовой станции (*узла беспроводного доступа к сети абонентов*) (см. раздел 1.2.2);
3. магистральные волоконнооптические линии связи (позиции 3 на рис. 1.1), см. раздел 1.2.3;
4. абонентский комплект беспроводного доступа (позиции 4 на рис. 1.1), см. раздел 1.2.4;
5. абонентский комплект доступа по технологии Ethernet (позиции 5 на рис. 1.1), см. раздел 1.2.5.

1.2.1 Магистральные линии, созданные по технологии беспроводной связи

Магистральные линии, созданные по технологии беспроводной связи (позиции 1 на рис. 1.1) организуются при помощи направленных антенн (рис. 1.3), подключенных высокочастотным кабелем (табл. 1.4) к сетевым адаптерам беспроводных сетей, (рис. 1.2), установленным в ПК-маршрутизаторы. Такой ПК-маршрутизатор будет выполнять функцию *магистрального узла* системы.

Магистральная линия, созданная по технологии беспроводной связи, как правило организуется при помощи двух ПК-маршрутизаторов, установленных на естественном возвышении (высотные хозяйствственные постройки: водонапорные башни, силосные башни и т.д.), либо на специально установленном высотном сооружении (см. табл. 1.3).

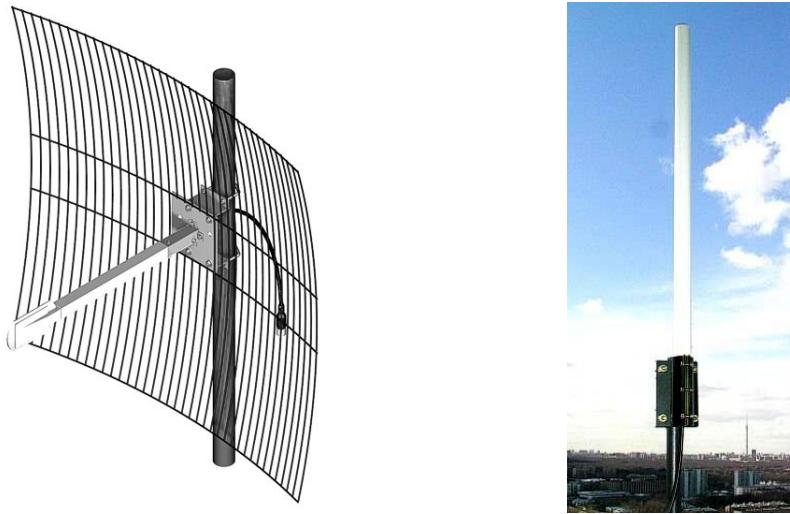


Рис. 1.3: Антенна направленная параболическая ГРАД-2497 (слева) и антenna штыревая ГРАД-2401 (справа)

Для создания магистральных линий на базе технологий беспроводных сетей были исследованы различные технические решения и принято для дальнейшего использования в проекте следующее оборудование:

1. адаптер беспроводной сети D-Link DWL-G520 (см. рис. 1.2 и табл. 1.8);
2. антенны параболические ГРАД-2492 и ГРАД-2497 (см. рис. 1.3 и табл. 1.9).

Антенно-фидерное оборудование фирмы СиБи-ГРАД было выбрано нами ранее в процессе создания городской компьютерной сети в городе Переславль-Залесский как наиболее доступное, производимое в России, достаточно надежное и с приемлемой ценой. Аналогичные зарубежные изделия при более высокой стоимости не предоставляют более высокие технические характеристики.

1.2.2 Радиосоты, узлы беспроводного доступа к сети абонентов

Радиосоты (позиции 2 на рис. 1.1) организуются при помощи всенаправленных штыревых антенн, имеющих круговую диаграмму излучения. Из данного класса изделий были выбраны антенны всенаправленные штыревые ГРАД-2401 (см. рис. 1.3 и табл. 1.9).

Штыревая антenna радиосоты подключается высокочастотным кабелем к сетевому адаптеру беспроводной связи. Адаптер устанавливается в ПК-маршрутизатор. Такой ПК-маршрутизатор будет выполнять функцию *узла доступа, базовой станции* для абонентов с беспроводным доступом к сети.

Для организации радиосеты (базовой станции) используется такой же тип адаптеров беспроводной связи, какой используется и для организации магистральных линий. В зависимости от назначения применяются различные режимы работы адаптера:

- *Режим Managed* — для организации магистральных линий.
- *Режим Master* — для организации радиосетей беспроводного доступа.

Всего в один ПК-маршрутизатор может быть установлено до 6 сетевых адаптеров. Поэтому, как правило магистральный узел и базовая станция объединяются в одном устройстве, за счет установки в данный ПК-маршрутизатор:

- одного адаптера беспроводного доступа с режимом работы Master и с подключением к нему штыревой всенаправленной антенной;
- одного или нескольких сетевых адаптеров для организации магистральных линий, это могут быть:
 - адаптеры беспроводного доступа с режимом работы Managed и с подключенными к ним параболической узконаправленной антенной — для организации магистральных беспроводных линий связи;
 - адаптеры FastEthernet 10/100 Base TX и, если используется волоконнооптический кабель, медиа-конверторы Planet FT-702S15 и Planet FT-802S15.

Учитывая данное обстоятельство, нигде далее отдельно не рассматриваются вопросы организации базовых станций, рассматривается только организация магистральных узлов (см. раздел 2.2). Магистральные узлы создаются на базе ПК-маршрутизаторов, модифицированных для применения в условиях сельских компьютерных сетей, то есть установленных в термобоксах (см. раздел 2.4) и оборудованных блоком питания собственной разработки (см. раздел 2.6).

1.2.3 Магистральные волоконнооптические линии связи

Магистральные волоконнооптические линии связи (позиции 3 на рис. 1.1) организуются за счет прокладки волоконнооптического кабеля. На концах данного кабеля устанавливается и подключается к ВОЛС соответствующее сетевое оборудование — в данном проекте это:

- ПК-маршрутизатор с адаптером Ethernet 10/100BaseTX, к которому подключается медиа-конвертер FastEthernet 100BaseTX — 100BaseFX (fiber optic, single mode);



Рис. 1.4: Медиа-конвертеры FastEthernet 100BaseTX — 100BaseFX (fiber optic, single mode, разъем SC) Planet FT-702S15 и Planet FT-802S15

- коммутатор FastEthernet Planet SW-801, подключаемый к волоконнооптическому магистральному кабелю с помощью медиа-конвертера FastEthernet 100BaseTX — 100BaseFX.

В качестве медиа-конвертера FastEthernet 100BaseTX — 100BaseFX (fiber optic, single mode) в данном проекте использовались изделия Planet FT-702S15 и Planet FT-802S15 (рис. 1.4, табл. 1.6), как весьма надежные в работе и имеющие приемлемую цену (\$75 – \$85).

1.2.4 Абонентский комплект беспроводного доступа

При разработке абонентского комплекта беспроводного доступа (рис. 1.1, позиция 4) самой важной характеристикой была минимизация его стоимости.

“Естественное” на первый взгляд решение — установка радиокарты в компьютер абонента — оказывается сравнительно дорогим, поскольку требует прокладки фидерного тракта от антенны (обычно устанавливаемой на крыше или чердаке) до компьютера абонента. Длина тракта оказывается существенной (20 м и более в типичных случаях), что делает необходимым использование “толстого” кабеля с малыми потерями, например Belden 9313 (см. табл. 1.4), что удорожает и усложняет решение.

Лучшие характеристики фидерного тракта можно получить, если разместить радиочастотное оборудование вблизи антенны. В предельном случае, можно разместить его непосредственно на мачте, на которой установлена антenna. Но это требует оборудования в исполнении outdoor, или монтажа оборудования в защитном контейнере. Таким

Таблица 1.4: Сравнительные характеристики различных радиочастотных кабелей, используемых в данном проекте в системах беспроводного доступа

Марка кабеля	Затухание на 1м, dB	Диаметр, мм	Цена за 1м
RG-58	1.05	5	\$0.5
Belden 9313	0.26	10.3	\$4

образом, рассмотренный выше вариант размещения радиочастотного оборудования в непосредственной близости от антенны минимизирует потери в фидерном тракте, но приводит к неприемлемой стоимости технического решения.

Оба варианта приводят к неприемлемо дорогому решению для абонентского окончания. Поэтому, наиболее удачным представляется промежуточный вариант: радиочастотное оборудование размещается в ближайшей к антенне отапливаемой части помещения, например, крепится к стене под потолком жилой комнаты. Связь его с компьютером абонента осуществляется по стандарту Ethernet 10/100Base-TX. Кабель для фидерного тракта выбирается исходя из бюджета потерь радиоканала для данного абонента: при наличии большого запаса по уровню сигнала может быть использован дешевый и более удобный в монтаже кабель RG-58. Если запас по уровню сигнала невелик, то приходится использовать более дорогой кабель.

В качестве радиочастного оборудования выбрано широко представленное на рынке устройство D-Link 900AP+ (см. рис. 1.5 и табл. 1.7). Стоимость этого устройства порядка \$70. Основное назначение D-Link 900AP+ — работа в режиме точки доступа (Master). Однако оно может быть сконфигурировано для работы в режиме беспроводного клиента (Managed), что и требуется в случае организации абонентского комплекта.

Устройства DWL-900AP+ (802.11b) и DWL-2000AP+ (802.11g) выбраны в качестве первого варианта реализации абонентского окончания. Возможно, по результатам опытной эксплуатации они будут заменены на другие изделия подобной функциональности из того же ценового диапазона. Критерии, по которым выбирается оборудование для абонентского комплекта, рассмотрены в главе 3.

Варьируя различные антенные изделия, мы можем влиять на запас по уровню сигнала и в каких-то случаях обеспечить условия более дешевого и удобного в монтаже кабеля RG-58.

При небольших расстояниях до базовой станции для организации абонентского комплекта беспроводного доступа возможно использование устройства D-Link 900AP+ со штатной антенной 2dBi. В этом случае абонентский комплект состоит из одного устрой-



Рис. 1.5: D-Link 900AP+

ства D-Link 900AP+ стоимостью \$70; кабеля “витой пары” 20–100м стоимостью \$0.08 за метр; и сетевой карты Ethernet для компьютера пользователя стоимостью \$5.

Если от абонента до базовой станции расстояние велико, то придется использовать антенну, имеющую лучшие характеристики по уровню сигнала (см. табл. 1.9).

1.2.5 Абонентский комплект доступа по технологии Ethernet

Абонентский комплект доступа по технологии Ethernet (позиции 5 на рис. 1.1) организуется при помощи подключения абонентской ПЭВМ (или ЛВС) к близко расположенному узлу (к коммутатору или ПК-маршрутизатору) магистрали системы за счет использования кабельного решения.

В зависимости от условий прокладки и расстояния может применяться кабель витой пары SFTP или волоконнооптический кабель и соответствующее сетевое оборудование у абонента и в магистральном узле. Как правило это сетевые адаптеры FastEthernet 10/100 Base TX и, если используется волоконнооптический кабель, медиа-конверторы Planet FT-702S15 и Planet FT-802S15.

1.3 Определение на местности условий прямой видимости

При организации беспроводных компьютерных сетей очень важным является определение *условий прямой видимости*⁷ между магистральными узлами системы. Как пра-

⁷Под этим подразумевается умение отвечать на следующий вопрос: если в пункте А антенна размещена на высоте H_1 , то на какой высоте H_2 надо разместить антенну в пункте Б, чтобы между

Таблица 1.5: Тактико-технические характеристики шара метеорологического

характеристика	значение
диаметр	2 м
объем	8 м ³
наполнитель	гелий
грузоподъемность	3 кг
длина фала	100 м
цена	12 000 рублей

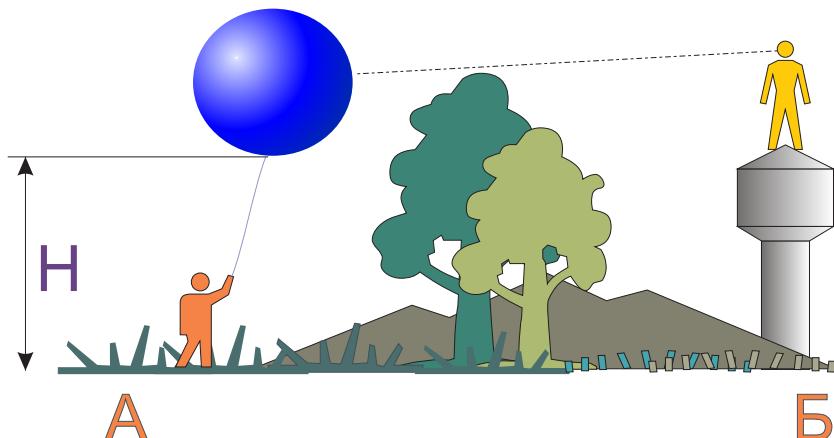


Рис. 1.6: Схема определения наличия прямой видимости при помощи метеорологического шара

вило, такие работы выполняются в виде заказа в определенные организации. Такие заказы приводят к временным задержкам в проектировании и дополнительным расходам. Заказ профиля в ГСПИ выполняется в течении месяца (для районов, удаленных от крупных городов) и стоит 3 – 5 тыс. рублей в зависимости от длины и сложности участка. При массовом строительстве компьютерных сельских сетей, размещение внешних заказов на проектирование может стать серьезным тормозом на пути реализации проекта. Поэтому для оценки наличия прямой видимости нами был применен простой метод, позволяющий самостоятельно выполнять все необходимые работы (рис. 1.6).

Для оценки наличия прямой видимости и определения высот размещения антенн для магистральных беспроводных линий связи и антенн радиосотов беспроводного до-

антенными была прямая видимость?



Рис. 1.7: Шар метеорологический

ступа был закуплен шар метеорологический с чехлом яркой расцветки (см. рис. 1.7 и табл. 1.5).

Из-за особенностей эксплуатации метеорологического шара, а так же в силу сложностей с приобретением газовых баллонов с гелием, было решено минимизировать необходимость заправок газом шара. Тем самым, по возможности шар требовалось хранить и перевозить в надутом состоянии. Был выбран шар диаметром 2 метра и грузоподъемностью 3 кг. что достаточно для наших целей и удобно для транспортировке в несдутом виде в стандартном автомобильном прицепе. Покупку произвели в специализированной фирме РусБал⁸.

Для определения необходимой высоты установки антенн над поверхностью земли производился подъем шара в точке А предполагаемой установки первой антенны и велось наблюдение в точке Б установки второй антенны.

В точке Б наблюдатель располагался в месте предполагаемой установки второй антенны. Если это было не возможно, то наблюдатель располагался на любой рядом расположенной постройке, достаточной высоты (высокие жилые здания, колокольни церквей, высокие постройки хозяйственного назначения). В случае отсутствия и такой возможности (отсутствие высоких построек) для размещения наблюдателя в точке Б использовалась автовышка с корзиной (максимальная высота подъема корзины: 18 метров).

В точке А начинался подъем (на фале) метеорологического шара. При установке

⁸<http://www.rusbal.ru/>

наблюдателем (в точке Б) визуального контакта с шаром отдавался сигнал на прекращения подъема. Для наблюдения использовались простейшие оптические приборы (бинокли), для связи между пунктами А и Б использовались любые доступные средства связи (сотовые телефоны, стационарные телефоны и т.п.).

После получения сигнала о визуальном контакте, подъем шара останавливался и производился замер длины выпущенного фала. Таким образом, выяснялась требуемая высота установки антенн для обеспечения условия прямой видимости между ними.

1.4 Пилотный проект компьютерной сети в сельской местности

В соответствии с техническим заданием данной работы, необходимо было не только разработать эскизную конструкторскую документацию для основных решений, необходимых для развертывания компьютерных сетей в сельских районах России, а также, в соответствии с этой эскизной конструкторской документацией, изготовить несколько экземпляров изделий каждого вида и на базе этих изделий развернуть пилотный проект беспроводной сети в сельской местности. Цель создания такого пилотного проекта понятна: в режиме опытной эксплуатации выявить все недочеты разработанных конструкций, чтобы внести корректизы в конструкторскую документацию и в дальнейшем уже предлагать проверенные в опытной эксплуатации изделия. Тем самым для разработки пилотного проекта нужно было выбрать таким образом точки размещения оборудования, чтобы, во-первых, провести полномасштабную опытную эксплуатацию различных типов разработанных изделий, во-вторых, рассмотреть различные варианты (по составу оборудования и функций) организации магистральных узлов, в-третьих, отработать типичные случаи размещения оборудования, а также случаи использования различного вида местных особенностей для точек размещения оборудования, в-четвертых, отработать различные решения организации как магистральных каналов (беспроводные решения и ВОЛС), так и различные решения подключения абонентов — беспроводной доступ и кабельный доступ.

С целью проверки всех данных обстоятельств, мы выбрали схему пилотного проекта сельской компьютерной сети, состоящей из шести магистральных узлов (см. соответствующие позиции на рис. 1.8 и 1.10):

1. Институт программных систем РАН.
2. Село Криушкино.

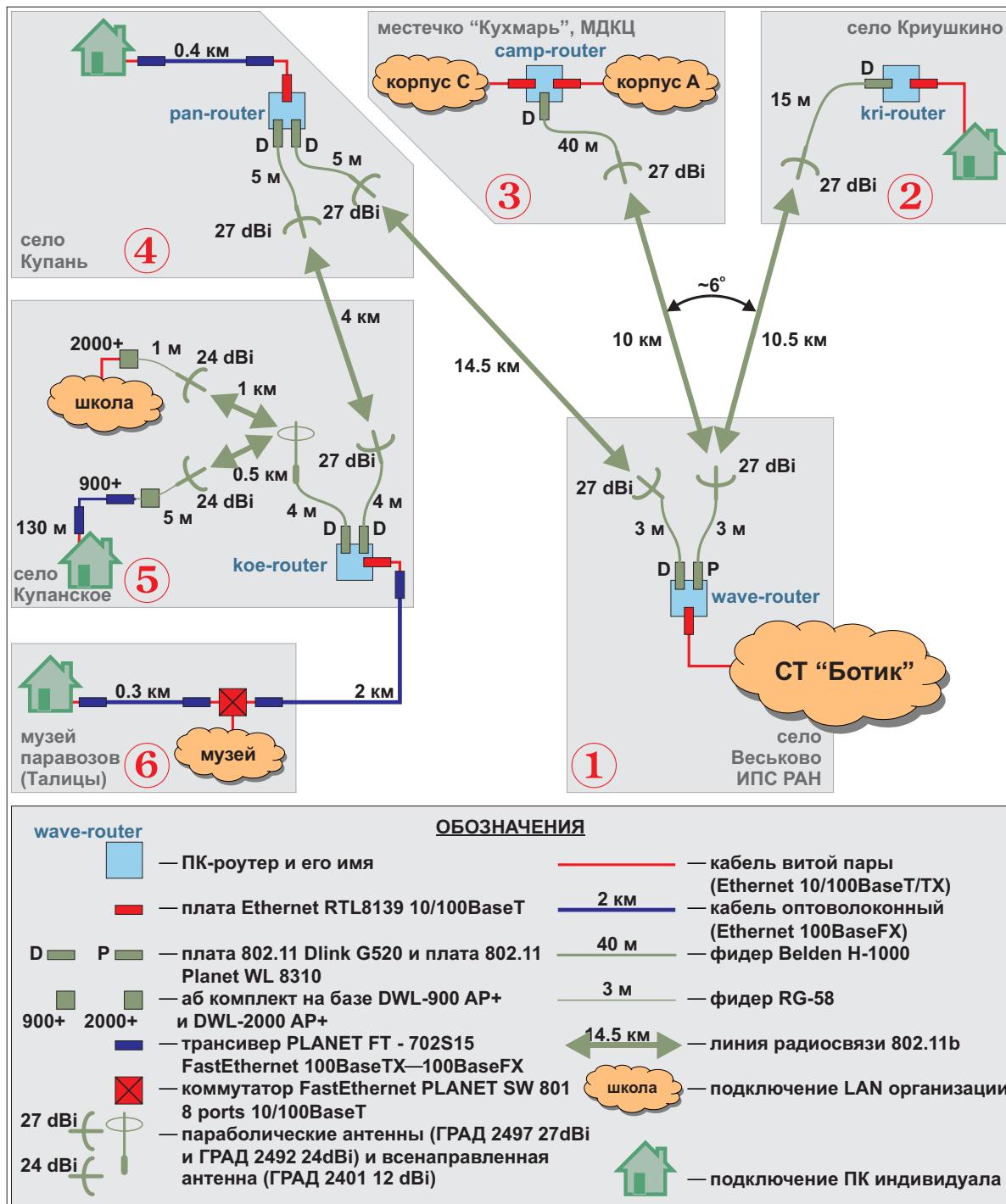


Рис. 1.8: Схема пилотного проекта беспроводной сети



Рис. 1.9: Магистральный узел в термобоксе на крыше ИПС РАН

3. Урочище Кухмарь.
4. Село Купань.
5. Поселок Купанское.
6. Село Талицы, музей паровозов.

Сегодня к данной сети подключено семь первых абонентов, эксплуатирующих данный сегмент. В дальнейшем, конечно, планируется подключение к магистральным узлам большего количества абонентов, проживающих в населенных пяти населенных пунктах данной сельской местности.

В последующих разделах рассмотрены особенности организации всех шести узлов pilotной сети.

Институт программных систем РАН.

В данном месте отрабатываются следующие решения (рис. 1.9):

- термобокс размещен на крыше высокого строения, а не на столбе;
- магистральный узел в ИПС РАН не является базовой станцией: здесь нет штыревой антенны, нет радиосоты для беспроводного доступа абонентов;
- в данной точке сходятся три канала точка-точка беспроводной связи:
 1. ИПС РАН – село Криушкино;



Рис. 1.10: Топографический план пилотного проекта



Рис. 1.11: Использование мачты собственной разработки для организации магистрального узла в селе Криушкино

2. ИПС РАН – урочище Кухмарь;
 3. ИПС РАН – село Купань.
- в ПК-маршрутизаторе данного магистрального узла стоят только два адаптера беспроводной связи:
 - магистральные узлы в урочище Кухмарь и в деревне Купань обслуживаются одной направленной антенной, подключенной к одному из этих адаптеров;
 - магистральный узел в селе Купань обслуживается другой направленной антенной, подключенной ко второму адаптеру.

Тем самым отрабатывается решение, когда две магистральные станции подключаются к третьей за счет разделения радио-канала и использования одной антенно-фидерного системы.

Село Криушкино.

Данная точка интересна тем, что в ней реализована схема установки антенны на металлической мачте собственной разработки, позволяющей достаточно быстро монтировать и демонтировать магистральный узел (рис. 1.11).



Рис. 1.12: Урочище Кухмарь: магистральный узел расположен в котельной, для размещения антенны (и обеспечения прямой видимости) используется существующее высотное сооружение (труба котельной)

Оборудование магистрального узла (ПК-маршрутизатор с сетевым адаптером беспроводной сети и адаптером FastEthernet) расположено в жилом доме единственного (пока) абонента. Абонент подключен к магистральному узлу по кабелю витой пары (FastEthernet 100Base-TX).

Пока абонент в данном населенном пункте один, допустимо решение о размещении оборудования магистрального узла в его (частном) доме. Такое решение позволило сократить время создания данного узла.

В случае подключения других абонентов в данном населенном пункте будет решен вопрос переразмещения оборудования магистрального узла вне частного дома.

Магистральный узел в селе Криушкино соединен магистральной беспроводной линией с узлом в Институте программных систем РАН.

Урочище Кухмарь.

Здесь расположен Международный детский компьютерный лагерь им. А. К. Айламазяна (МДКЦ им. А. К. Айламазяна ИПС РАН) и ряд других детских учреждений.

В магистральном узле в урочище Кухмарь расположен ПК-маршрутизатор, в котором установлена сетевой адаптер беспроводного доступа и два Ethernet-адаптера. Были использованы местные особенности для размещения оборудования (рис. 1.12):

- сетевое оборудование магистрального узла размещено в здании котельной МДКЦ

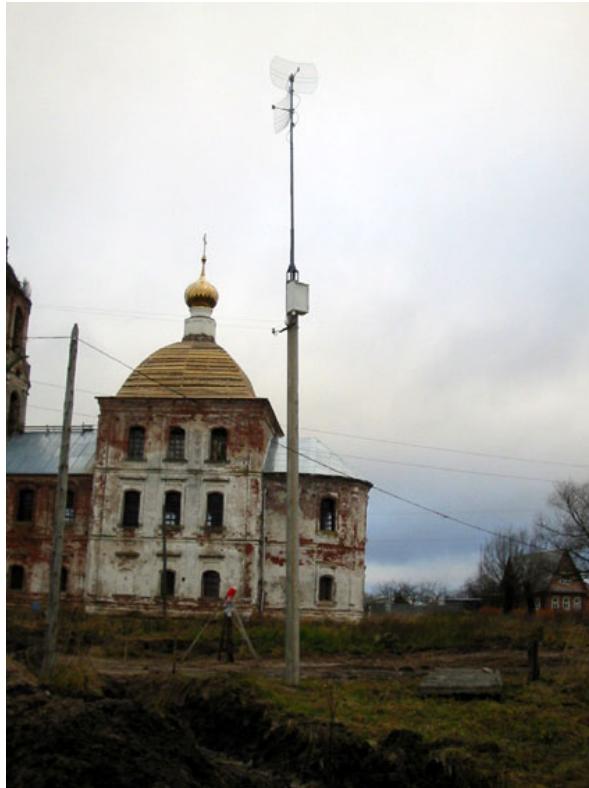


Рис. 1.13: Село Купань: магистральный узел размещен вне помещения, на специально установленном для этого осветительном бетонном столбе с дополнительной мачтой

им. А. К. Айламазяна ИПС РАН;

- направленная антенна для беспроводного канала связи “ИПС РАН – урочище Кухмарь” установлена на высоте 25 метров на трубе котельной (см. рис. 1.12).

Заметим, что труба котельной является единственным сооружением, доминирующим над вершинами деревьев соснового леса, окружающего здания МДКЦ им. А. К. Айламазяна ИПС РАН.

Сетевой адаптер беспроводного доступа в ПК-маршрутизаторе используется для организации беспроводного канала связи “ИПС РАН – урочище Кухмарь”.

К двум Ethernet-адаптерам ПК-маршрутизатора при помощи кабеля витой пары подключены две ЛВС в двух учебных корпусах МДКЦ им. А. К. Айламазяна ИПС РАН. В результате создана достаточно крупная компьютерная система МДКЦ им. А. К. Айламазяна ИПС РАН, которая покрывает компактно расположенные строения, расположенные в радиусе 100 метров и к которой подключено более тридцати компьютеров.

Село Купань.

Магистральный узел интересен тем, что здесь установлены две направленные антен-



Рис. 1.14: Поселок Купанское: магистральный узел (он же узел доступа) размещен вне помещения, на специально установленном для этого осветительном бетонном столбе

ны, одна из которых организует канал “Купань – ИПС РАН”, вторая — канал “Купань – Купанское”.

К магистральному узлу близлежащий (пока единственный) абонент подключен при помощи кабельного решения.

В настоящее время данный узел является только магистральным, он не выполняет функции базовой станции беспроводной сети.

Поселок Купанское.

В данном случае узел реализует и функции магистрального узла, и точки беспроводного доступа. В ПК-маршрутизатор установлены два адаптера беспроводной сети:

- адаптер, работающий в режиме Master, подключен к штыревой всенаправленной антенне, организует радиосоту беспроводного доступа к сети абонентов (рис. 1.14);
- адаптер, работающий в режиме Managed, подключен к параболической узконаправленной антенне, используется для организации канала “село Купань — поселок Купанское” (рис. 1.15).

Кроме того, в данном ПК-маршрутизаторе установлен адаптер FastEthernet 100Base-TX, с подключенным к нему медиа-конвертером 100Base-TX — 100Base-FX(SC). Дан-



Рис. 1.15: Школа в поселке Купанское. Параболическая антенна беспроводного подключения ЛВС школы к компьютерной сети

ное сетевое оборудование используется для организации ВОЛС “поселок Купанское — село Талицы”.

К радиосети беспроводного доступа в поселке Купанское при помощи абонентских комплектов беспроводного доступа сегодня подключено два абонента: школа (ЛВС компьютерного класса, рис. 1.15) поселка Купанское и частный дом (одиночная ПЭВМ).

Абонентской комплект беспроводного доступа для школы реализован по схеме, описанной выше: средства радиодоступа расположены в отапливаемом помещении, в месте, максимально приближенном к антенне (фидер: RG58, около 1 метра). ЛВС школы со средствами радиодоступа соединены линией Ethernet 100 Base-TX (UTP).

В подключении частного дома отрабатывалось решение для случая отсутствия прямой видимости из дома (и с кровли) абонента штыревой антенны радиосети. В данном случае было применено следующее решение:

- в ближайшем к абоненту месте, откуда была обеспечена видимость центра радиосети, был установлен осветительный бетонный столб;
- на вершине столба были смонтированы направленная (на центр радиосети) параболическая антенна и термобокс с абонентскими средствами радиодоступа, к Ethernet-интерфейсу которого подключен медиа-конвертер 100Base-TX — 100Base-FX(SC);
- между термобоксом и домом абонента были проложен (в грунт) волоконнооптический кабель, подключенный в термобоксе к медиа-конвертеру 100Base-TX — 100Base-FX(SC);



Рис. 1.16: Музей паровозов: размещение магистрального волоконнооптического узла

- ПЭВМ абонента подключена к волоконнооптическому кабелю при помощи сетевого адаптера Ethernet 100 Base-TX и медиа-конвертера 100Base-TX — 100Base-FX(SC).

Село Талицы, музей паровозов.

Магистраль “поселок Купанское — село Талицы” выполнена с использованием волоконнооптического кабеля потому, что данный пролет небольшой (по протяженности) и здесь растет густой и высокий лес. Поэтому, при использовании беспроводных технологий в районе деревни Талицы пришлось быставить вышку высотой не менее 35 метров, что удорожило бы проект примерно на полтора миллиона рублей.

Волоконнооптический кабель для данного пролета обошелся в 210 000 рублей (длина пролета 2 100 метров, стоимость кабеля 100 руб./м). Использован волоконнооптический одномодовый кабель ДПС-П-06-8М 10/125, предназначенный для укладки в грунт.

Стоимость укладки кабеля в грунт удалось снизить до 10 500 рублей, за счет того, что расходы были разделены с местной телефонной компанией: у телефонной компании были планы прокладки телефонного кабеля в село Талицы и нам удалось договориться об укладке двух кабелей (нашего волоконнооптического и их телефонного) за один проход кабелеукладчика. Тем самым, с минимальными затратами для обеих сторон удалось осуществить прокладку обоих кабелей.

Магистральный узел, расположенный в этом месте (рис. 1.16) представляет собой коммутатор Planet SW-801, подключаемый к волоконнооптическому магистральному кабелю с помощью медиа-конвертера Planet FT-702S15.

Магистральный узел располагается в музее паровозов и к нему подключены два абонента

- ЛВС музея паровозов, подключена по Eternet 100 Base-TX (UTP);
- ПЭВМ в частном доме, подключена по Eternet 100 Base-FX (ВОЛС, использованы два медиа-конвертера Planet FT-702S15 на обоих концах кабеля и сетевой адаптер FastEthernet в ПЭВМ абонента).

Общая оценка пилотного проекта.

Тем самым, выбор пилотного участка удалось осуществить таким образом, что все разработанные решения и различные варианты привязки к местным условиям смогут быть проверены в период опытной эксплуатации данного сегмента сети.

Таблица 1.6: Медиа-конвертеры Planet FT-702S15 и Planet FT-802S15. Технические характеристики

Продукт	FT-702S15 (FT-802S15)
Модель	FT-7 серия (FT-8 серия)
Протокол	IEEE 802.3u, 100Base-FX, 100Base-TX
Размеры	70 × 94 × 26 мм
Вес	0.2кг без блока питания
Питание	5V DC, 1A, max.
Витая пара	RJ45, Категория 5 (EIA/TIA 568), до 100 м
Одномодовое волокно (SM)	9/125 μ m, до 15 км
Рабочая температура	0–50°C
Рабочая влажность	5–90%

Таблица 1.7: Тактико-технические характеристики D-Link 900AP +

Характеристики	IEEE 802.11b wireless LAN IEEE 802.3 10BASE-T Ethernet IEEE 802.3u 100BASE-TX Fast Ethernet
Чувствительность приемника при нормальной температуре	22 Mbps: 10–5 BER @ -80 dBm minimum 11 Mbps: 10–5 BER @ -79 dBm minimum 5.5 Mbps: 10–5 BER @ -83 dBm minimum 1 Mbps: 10–5 BER @ -89 dBm minimum
Диапазон частот	2.4 – 2.4835 GHz
Модуляция	Технология широкополосной передачи данных (DSSS). Бинарное кодирование пакетов (PBCC)
Скорость передачи	22 Mbps / 11 Mbps / 5.5 Mbps / 2 Mbps / 1 Mbps / Auto fall-back
Скорость и модуляция	22 Mbps/8.5 db: PBCC; 11 Mbps/4.5 db: PBCC; 11 Mbps/8.5 db: CCK; 5.5Mbps/1.5 db: PBCC; 5.5 Mbps/5.5 db: CCK; 2 Mbps/3 db: Barker
Аппаратная часть	Преобразование беспроводная сеть/LAN Через порт RJ-45 10/100 Mbps Fast Ethernet
Антenna	Съемная 2dBi всенаправленная (reverse SMA)
Радиус действия	До 100 м (328 feet) в помещении; До 400 м (1,312 feet) вне помещения

Продолжение таблицы 1.7

Характеристики	IEEE 802.11b wireless LAN IEEE 802.3 10BASE-T Ethernet IEEE 802.3u 100BASE-TX Fast Ethernet
Светодиоды диагностики	Power; Link; Tx/Rx
Системное программное обеспечение: режимы работы	Точка Доступа; Мост между Точками Доступа; Мост в режиме Точка – много Точек; Клиент
Роуминг	Междуд Точками Доступа в одной подсети
Сетевые протоколы	TCP/IP; IPX/SPX; NetBEUI; ARP; SNMP; DHCP; NDIS3; NDIS4
Назначение IP адресов	Через встроенный DHCP сервер
Безопасность	Шифрование: 64-bit, 128-bit, 256-bit WEP (Wired Equivalent Privacy); Списки доступа (ACL)
Управление	Web-управление; TFTP клиент для обновления firmware; DHCP клиент; Установка IP сессии (ARP/ping);
Питание	DC 5V, 2.5A; внешний блок питания
Размеры	142 × 145 × 32 mm
Вес	288 grams (0.636 lb.)
Диапазон рабочих температур	0 – 55°C(32 – 131°F)
Диапазон температур при хранении	-20 – 65°C(-4 – 149°F)
Влажность	5% – 95%, без образования конденсата
Совместимость	TM Wi-Fi
Стоимость	\$75

Таблица 1.8: Таблица основных тактико-технических характеристик адаптера беспроводной связи D-Link DWL-G520

Описание	Беспроводной адаптер PCI 802.11g
Серия	AirPlus Xtreme G
Мощность передатчика	15 dBm ±2dB
Частота	2,4 – 2,462 ГГц
Скорость передачи данных	54 /48/ 36/ 24/ 18/ 12/ 11/ 9/ 6/ 5,5/ 2/ 1 Мбит/сек с автоматическим снижением
Метод доступа	CSMA/CA с ACK
Соответствие стандартам	IEEE 802.11, IEEE 802.11b, IEEE 802.11g-Draft
Чувствительность	54 Мбит/сек OFDM, 10% PER, -68 dBm 48 Мбит/сек OFDM, 10% PER, -68 dBm 36 Мбит/сек OFDM, 10% PER, -75 dBm 24 Мбит/сек OFDM, 10% PER, -79 dBm 18 Мбит/сек OFDM, 10% PER, -82 dBm 12 Мбит/сек OFDM, 10% PER, -84 dBm
Модуляция	OFDM, CCK
Безопасность	WEP-кодирование с 64- или 128-битным ключом, поддержка протоколов аутентификации 802.1x и защищенного беспроводного доступа WPA
Антenna	С коннектором reverse SMA
Интерфейс	PCI
Потребление энергии	Режим PowerSave: 28 mA, режим Standby: 4,66 mA, режим передачи: 248 mA
Размеры (ширина x высота x глубина)	114,3 × 54 × 8,7 мм
Вес	55 грамм
Рабочая температура	0 – 55°C
Стоимость	\$75

Таблица 1.9: Тактико-технические характеристики антенн ГРАД-2401, ГРАД-2492 и ГРАД-2497

Модель антенны	Град-2401	Град-2492	Град-2497
Диапазон частот, МГц	2400–2483	2400–2500	
Коэффициент усиления, dBi	12,0	24,0	27,0
Макс. подводимая мощность, Вт		100	
Ширина диаграммы направленности в вертикальной плоскости	7°	7°	6°
Ширина диаграммы направленности в горизонтальной плоскости	360°	7°	6°
Входное сопротивление, Ом		50	
KCB в раб. диапазоне частот, не более	1,5	1,5	1,6
Диаметр мачты, мм	40–62	35–50	25–48
Масса, кг	2,0	3,5	4,2
Габаритные размеры, длина x ширина x высота, мм	170 × 90 × 1040	800 × 800 × 525	1150 × 800 × 575
Макс. скорость ветра, км/час	160	140	140
Поляризация	вертикальная	линейная, сменная	
Диапазон рабочих температур	−40°C – +50°C	−50°C – +50°C	
Размер зеркала антенны, мм	—	800 × 800	1150 × 800
Разъем	Розетка N типа	Вилка N типа	
Дополнительно	Вход антенны замкнут по постоянному току		
Стоимость	\$70	\$80	\$120

Глава 2

Технические решения для типового магистрального узла

2.1 Общее описание

Магистральный узел осуществляет связь с соседними с ним магистральными узлами, а также подключение к магистрали ПЭВМ и ЛВС абонентов. В состав оборудования магистрального узла входит ПК-маршрутизатор, в который может быть установлено до 6 сетевых интерфейсов. Для данного проекта типичными являются радиointерфейсы 802.11b/g, возможно также использование интерфейсов 10/100Base-TX (Ethernet по кабелю типа “витая пара”) и 100Base-FX (Ethernet по одномодовому волоконнооптическому кабелю), табл. 2.1.

В данном проекте магистральные узлы (как правило) монтируются на бетонных осветительных столбах в металлических контейнерах, обеспечивающих защиту от погодных условий (термобокс) и вандализма.

Кроме маршрутизатора, в контейнере размещен аккумулятор и блок питания маршрутизатора, обеспечивающий работоспособность маршрутизатора при низком качестве внешнего электропитания, а также поддержание работы маршрутизатора в течение минимум 8 часов при отключении внешнего электропитания.

2.2 Маршрутизатор

В качестве активного оборудования магистрального узла используется ПК-маршрутизатор под управлением ОС Linux (рис. 2.1). В региональной сети Переславля такие маршрутизаторы используются с 1994 года, и показали себя как сравнительно недорогие (\$300), надежные и открытые для доработки устройства с модульной конструкцией.

Таблица 2.1: Состав аппаратных средств типового магистрального узла

Наименование компонента	Количество	Стоимость, \$
ПК-маршрутизатор	1	\$300
Радиоинтерфейсы 802.11b/g	*)	\$70
Интерфейсы 10/100Base-TX	*)	\$4
Интерфейсы 100Base-FX	*)	\$109
Термобокс	1	\$110
Аккумулятор	1	\$30
Блок 220–12	1	\$25
Блок 12–ATX	1	\$20

*) – количество зависит от назначения узла, всего до 6 интерфейсов

Одним из принципиальных решений при построении региональной сети явился отказ от дорогих устройств (Cisco, Sun) и заменой их на недорогое массовое оборудование класса IBM PC, дополненное собственными программно-аппаратными разработками, что обусловлено прежде всего

1. низкой стоимостью оборудования;
2. широкой доступностью и большим выбором изделий, что позволяет получить необходимую функциональность;
3. гибкостью и многофункциональностью, широким спектром возможных сетевых интерфейсов и возможностью пошаговой модификации.

ПК-маршрутизаторы, используемые в данной работе, собраны из следующих компонентов (см. рис. 2.1):

1. Материнская плата EPOX EP-3PTA;
2. Модуль RAM PC133 128MB M.tec;
3. Процессор Intel Pentium III 666 MHz (или 750 MHz);
4. Модуль Flash-памяти 128MB Transcend TS128MDOM40V;
5. Блок 12–ATX (собственные разработка и производство);
6. Сторожевой таймер Botik Wdog 3 (собственные разработка и производство).

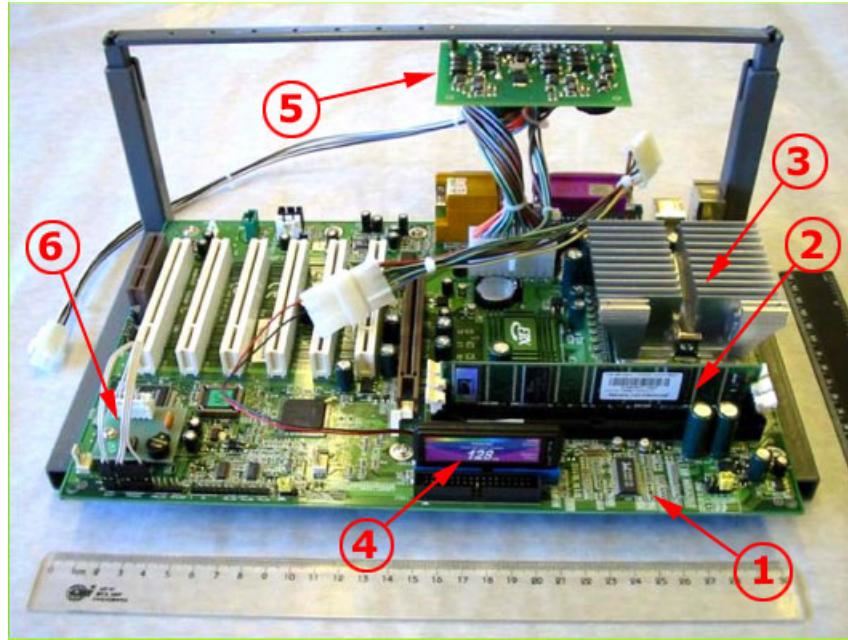


Рис. 2.1: ПК-маршрутизатор

Все компоненты (кроме позиций 5 и 6 на рис. 2.1) являются массово производимыми, дешевыми и доступными на компьютерном рынке России, хотя нужно заметить, что конкретные модели процессоров и особенно материнских плат держатся на рынке недолго (полгода–год). Поэтому при выполнении проектов на базе ПК-маршрутизаторов проводится анализ текущего состояния компьютерного рынка, выбор подходящей материнской платы, приобретение одиночного образца, тестирование, и затем закупка партии комплектующих, полностью покрывающая потребности проекта. Таким образом, достигается однородность парка ПК-маршрутизаторов в пределах одного проекта. Подробное описание ПК-маршрутизаторов доступно на сайте <http://www.botik.ru/tech>.

2.2.1 Надежность маршрутизатора

Десятилетний опыт эксплуатации ПК-маршрутизаторов показал, что наиболее слабыми с точки зрения надежности элементами являются механически-подвижные компоненты: вентиляторы охлаждения блока питания и процессора, и жесткий диск (HDD). В связи с этим в новом поколении ПК-маршрутизаторов эти компоненты исключены.

Устранение из конструкции ПК-маршрутизатора вентиляторов стало возможным благодаря принятым мерам по снижению мощности, потребляемой центральным процессором (раздел 2.2.3). Как следствие, снижается мощность, рассеиваемая процессором и блоком питания, и для охлаждения становится достаточно естественного воздушного

потока. Мощность, рассеиваемая блоком питания, дополнительно снижена благодаря высокому КПД блока питания собственной разработки (Блок 12–ATX).

Еще одним источником ненадежности маршрутизаторов является воздействие атмосферного электричества на интерфейсы маршрутизатора, к которым подключены внешние проводные (SFTP) линии. Использование устройств грозозащиты улучшает ситуацию, однако не обеспечивает полной защиты. Проблему решает использование волоконнооптических линий, но они все еще существенно дороже медных (см. таб. 1.2). Поэтому использование проводных (SFTP) линий *оправдано* в случаях, не требующих большой надежности (например, абонентское окончание отдельного абонента). В случае повреждения интерфейской карты атмосферным электричеством карта заменяется на новую (\$4).

2.2.2 Дисковая подсистема маршрутизатора

Вместо HDD в маршрутизаторе используется модуль Flash-памяти объемом 128МВ и интерфейсом ATA (IDE). Емкости модуля 128МВ достаточно для установки необходимых для работы маршрутизатора пакетов из общепринятого дистрибутива Debian GNU/Linux. Реальная емкость диска Transcend TS128MDOM40V составляет 125168 блоков, что равно 122МВ, установленное программное обеспечение занимает 93% этой ёмкости (рис. 2.2).

```
wave-router:~# df
Filesystem      1k-blocks   Used   Available  Use% Mounted on
/dev/root        121183    106600      8325  93% /
/dev/ram0         59903      934     55693   2% /volatile
193.232.174.1:/var/log/wave-router
                    29293500  24152784   5140716  82% /remote
```

Рис. 2.2: Файловые системы ПК-маршрутизатора

2.2.3 Снижение потребляемой мощности

Работы по снижению потребляемой маршрутизатором мощности преследуют две цели: избавиться от необходимости интенсивного охлаждения маршрутизатора, и увеличить время работы узла от аккумуляторов.

Для снижения потребляемой маршрутизатором мощности приняты следующие меры:

1. выбран тип процессора, имеющий наилучшие характеристики по энергопотреблению;
2. понижена тактовая частота процессора;
3. понижено напряжение питания процессора;
4. из конструкции маршрутизатора исключена видеокарта.

Выбор процессора. Из числа массово производимых и доступных на рынке типов процессоров был выбран тип процессора, имеющий наилучшие характеристики по энергопотреблению. Для этого было проведено сравнение потребляемой мощности из следующего ряда процессоров: ADM Athlon, Intel Pentium IV, Intel Pentium III, Intel Celeron, Via C3. Результаты сравнения приведены в табл. 2.2.

Во время данных экспериментов измерялись потребляемые мощности в режиме полной нагрузки на процессор (колонка “P@100%”) и при отсутствии нагрузки — в режиме простоя idle (колонка “P@0%”). Материнская плата не во всех случаях была одна и та же, но тем не менее полученные величины могут быть использованы для принятия решения, поскольку наибольшая доля мощности потребления падает на процессор.

В этой же серии замеров проводилось тестирование пропускной способности при установке нескольких одновременно передающих сетевых интерфейсов Ethernet 10/100, замкнутых попарно (multicast, promiscuous mode). Типы примененных сетевых интерфейсов:

rtl — на микросхеме RealTek RTL8139;

int58 — на микросхеме Intel EtherExpress i82558;

int59 — на микросхеме Intel EtherExpress i82559;

3com — на микросхеме 3Com 3C905-TX.

В колонке “Bandwidth” табл. 2.2 указана скорость передачи данных на каждой из установленных в материнскую плату сетевых плат.

По результатам сравнения был выбран вариант реализации ПК-маршрутизатора на процессоре Intel Pentium III.

Таблица 2.2 — Потребляемая мощность ПК-маршрутизаторов с различными типами процессоров

Тип процессора, частота	P@0%	P@100%	Bandwidth	
Pentium133MMX +4rtl +2int59 +4int59	14.5 W 15.7 W 15.8 W 20.3 W	17.7 W 18.4 W 18.9 W 20.3 W	2.71 MB/s 7.60 MB/s 3.33 MB/s	
VIA C3 500/66/100 +2int59 +6int59 +2rtl +6rtl		12.7 W 15.8 W 12.1 W 13.5 W	16.0 W 20.4 W 15.7 W 17.8 W	11.71 MB/s 4.87 MB/s 8.45 MB/s 2.70 MB/s
VIA C3 1000/133/133 +2int59 +6int59		13.3 W 16.5 W	17.6 W 24.3 W	11.72 MB/s 6.08 MB/s
Pentium III 300/66/100 +2rtl +6rtl +4 3com +2int59 +4int59 +6int59 +2int58	9.5 W 11.9 W 11.5 W 15.0 W 11.4 W 12.8 W 14.6 W 13.8 W	13.4 W 15.5 W 15.6 W 19.1 W 15.1 W 17.6 W 19.4 W 17.4 W	8.45 MB/s 2.69 MB/s 4.62 MB/s 11.72 MB/s 7.21 MB/s 4.74 MB/s 11.71 MB/s	
Pentium III 450/100/100 +6rtl +4 3com +4int59 +6int59	9.8 W 11.7 W 15.2 W 12.9 W 14.8 W	17.3 W 17.8 W 21.3 W 19.6 W 22.1 W	3.06 MB/s 5.34 MB/s 8.52 MB/s 5.52 MB/s	
Pentium III 600/133/133 +6rtl +4 3com +2int59 +4int59 +6int59 +2int58	9.9 W 11.9 W 15.3 W 11.6 W 13.1 W 14.9 W 14.6 W	19.9 W 19.9 W 23.4 W 16.8 W 21.7 W 23.2 W 19.4 W	3.65 MB/s 5.99 MB/s 11.72 MB/s 9.25 MB/s 6.05 MB/s 11.72 MB/s	

Продолжение таблицы 2.2

Тип процессора, частота	P@0%	P@100%	Bandwidth
Celeron 730/66/100 +6rtl +6int59			
	17.8 W	19.5 W	3.41 MB/s
	19.1 W	21.8 W	5.67 MB/s
Celeron 2000/400/266 +6rtl +6int59	23.0 W		
	23.2 W	33.0 W	4.46 MB/s
		33.6 W	6.46 MB/s
SMP 2×Xeon 2800	26.0 W	132.0 W	
SMP 2×AMD Opteron	64.2 W	112.1 W	

Снижение тактовой частоты процессора. Управление встроенным в процессор умножителем частоты на большинстве рассмотренных процессоров невозможно (если не рассматривать “неофициальные” способы — <http://www.overclockers.ru>), но материнская плата Ерох ЕР-3РТА позволяет понизить частоту внешнего тактирования процессора (блок перемычек JP2) с 133 МГц до 66 МГц. Таким образом, рабочая частота процессора снижается с 666 МГц до 333 Мгц (раздел 2.2.3). Соответственно снижается производительность процессора, однако для выполнения функций маршрутизатора производительности РIII на частоте 300 МГц более чем достаточно. В результате снижения рабочей частоты снижается потребляемая процессором мощность (ср. данные Pentium III 300/66/100 и Pentium III 600/133/133 приведены в табл. 2.2).

Понижение напряжение питания процессора (CPU core voltage) при работе процессора на пониженной частоте. По умолчанию материнская плата ЕРОХ ЕР-3РТА использует напряжение питания процессора 1.65 В. BIOS Setup позволяет понизить напряжение до 1.55 В, но процессор должен сохранять работоспособность и при более низких напряжениях.

Снижение напряжения до 1.30В было достигнуто за счет прямого программирования регистров микросхемы W82627HF на этапе загрузки операционной системы. Программирование регистров микросхемы реализует простая программа `corev.c`, прилагаемая к отчету на носителе.

За счет снижения напряжения питания процессора потребляемая процессором мощность снижена еще на 0.36Вт в режиме простоя и на 3.84Вт в режиме полной нагрузки.

Исключение видеокарты из конструкции маршрутизатора. Традиционно в ПК-маршрутизаторы нами устанавливалась видеокарта, что позволяло обслуживать его,

```
wave-router:~# cat /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
model          : 8
model name     : Pentium III (Coppermine)
stepping        : 6
cpu MHz         : 332.268
cache size     : 256 KB
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level    : 2
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 sep
                  mtrr pge mca cmov pat pse36 mmx fxsr sse
bogomips        : 663.55
```

Рис. 2.3: Процессор PIII 666 МГц работает на пониженной частоте

подключив к нему компактный монитор (диагональ экрана 9 дюймов) и клавиатуру. Однако на сегодняшний день на рынке нет низкопотребляющих видеокарт: современные видеокарты включают в себя графические акселераторы и имеют потребляемую мощность порядка 6 Вт.

Поэтому было принято решение исключить из конструкции видеокарту и осуществлять обслуживание через интерфейс RS-232 (в режиме Linux serial console). Недостаток этого решения — на материнских платах EP-3PTA, как и на большинстве недорогих (несерверных) материнских плат, доступ к BIOS Setup через интерфейс RS-232 невозможен. Поэтому при необходимости изменения установок Bios Setup в маршрутизатор приходится временно устанавливать видеокарту, что является некоторым недостатком принятого решения. Однако практика эксплуатации ПК-маршрутизаторов показала, что необходимость изменения установок BIOS Setup возникает исключительно редко,

поэтому с этим недостатком можно мириться.

Выводы. В результате всех принятых мер энергопотребление маршрутизатора без интерфейсов в режиме простоя составляет 8.52 Вт, в режиме полной загрузки 13.2 Вт.

При выборе типа сетевых интерфейсов для ПК-маршрутизаторов также уделяется внимание потребляемой мощности.

2.2.4 Программное обеспечение маршрутизатора

Программное обеспечение маршрутизатора построено на базе Debian GNU/Linux, версия ядра 2.4.x. Использование общепринятого дистрибутива снимает заботы о поддержке большей части пакетов.

Приняты меры, обеспечивающие устойчивость файловой системы маршрутизатора к внезапному отключению электропитания. Корневая часть файловой системы работает в режиме “только чтение”. Временные файлы (в частности, `/var/log`) размещаются на RAM-диске, и при выключении маршрутизатора теряются. Модифицируемые файлы, которые требуют сохранения при выключении маршрутизатора (в частности, долговременные лог-файлы) размещаются на разделе, монтируемом по NFS с центрального сервера. Этот подход используется в ПК-маршрутизаторах с 1994 г и неплохо себя зарекомендовал [1].

Поддержка протоколов динамической маршрутизации BGP4, OSPF2 обеспечивается использованием программного обеспечения *gated* версии 3.5.5.

2.3 Средства радиодоступа

В качестве средств беспроводного доступа в проекте используется оборудование стандарта IEEE 802.11. Выбор обусловлен тем, что популярность технологии 802.11 (RadioEthernet) на сегодняшний день привела массовому конкурентному производству оборудования стандарта 802.11, что повлекло резкое снижение цен на него. В то же время, стандарт обеспечивает приемлемые скорости передачи: 11 Мбит/с для 802.11b и до 54 Мбит/с для 802.11a/g, при этом накладные расходы на MAC-уровень составляют около 50%.

2.3.1 Выбор типа радиокарт

Выбор конкретного типа радиокарт производился из следующих соображений:

- Интерфейс PCI, чтобы карту можно было устанавливать непосредственно в слоты расширения ПК-маршрутизатора без дорогостоящих и снижающих эффективность переходников PCMCIA–PCI.
- Возможность подключения внешней антенны, и желательно через дешевый разъем. Для сравнения, на картах Orinoco подключение внешней антенны производится через миниатюрный разъем, приобрести который отдельно невозможно, а кабель—переходник на стандартный разъем N-type стоит \$45. В то же время, разъемы типов SMA и TNC доступны на рынке по ценам порядка \$1-\$3.
- Желательна поддержка как устоявшегося стандарта 802.11b, так и нового 802.11g, потенциально позволяющего получить большие скорости передачи при хороших условиях прохождения сигнала.
- Наличие драйвера для ОС Linux с исходными текстами.
- Возможность работы как в режиме клиента (Managed), так и в режиме точки доступа (Master).
- Возможность мониторинга эфира в целях отладки и обнаружения помех.

Оборудование стандарта 802.11a, работающее в диапазоне 5ГГц, не рассматривалось, поскольку в России пока не определена процедура оформления разрешений на использование этого диапазона. Выбор был ограничен только оборудованием, работающим в стандартах 802.11b, 802.11g.

Были рассмотрены следующие чипсеты:

- Broadcom BCM4306/BCM2050

Закрытая архитектура, нет открытого драйвера для ОС Linux.

- Intel Centrino (PRO/Wireless 2200, 2915)

Не поддерживает режим Master. Применяется главным образом в портативных компьютерах, карт с интерфейсом PCI не производится.

- Intersil Prism54

Есть открытый драйвер для Linux (<http://www.prism54.org>). Режим Master поддержан в микрокоде (firmware) карты.

- Atheros AR521x

Есть открытый драйвер для Linux (<http://madwifi.sourceforge.net>). Режим Master поддержан на уровне драйвера.

Таким образом, в дальнейшем мы выбирали из числа сетевых адаптеров беспроводного доступа, представленных на компьютерном рынке, адAPTERы на чипсЕтах Atheros или Prism54 с интерфейсом PCI и дешевым разъемом внешней антенны.

На рынке было найдено два типа сетевых адаптеров, построенных на чипсЕте Atheros AR5212: D-Link DWL-G520 (\$70), и Planet WL-8310 (\$58), оба с одинаковым дешевым разъемом антенны RP-SMA. Эти сетевые адAPTERы и были выбраны для использования в проекте.

Кроме того, на рынке присутствуют сетевые адAPTERы на чипсЕте из семейства Prism54: SMC 2802W (\$73) с коннектором RP-SMA. Эта сетевые адAPTERы рассматриваются как запасной вариант, в случае если с основным вариантом (Atheros) обнаружатся какие-либо проблемы.

Кроме того, на рынке есть предложения сетевых адAPTERов Netgear WG311 (\$73). Относительно этих адAPTERов нет точной информации: по некоторым данным, они построены на чипсЕте Prism54, по другим — на Atheros AR5212. Этото сетевой адAPTER также рассматривается нами как запасной вариант.

2.3.2 Опыт работы с радиокартами Atheros

Для сетевых адAPTERов на основе микросхемы Atheros AR5212 существует открытый (лицензия GPL), хорошо откомментированный, и продолжающий развиваться драйвер `madwifi`¹ для ОС Linux. Компонента драйвера, непосредственно оперирующая регистрами микросхемы AR5212 (HAL, Hardware abstraction Layer) закрыта (поставляется без исходных текстов), что мотивируется необходимостью предотвратить некорректную конфигурацию микросхемы, способную вызвать превышение допустимой излучаемой мощности и появление гармоник в излучаемом сигнале.

Кроме того, существует альтернативный драйвер² для родственной микросхемы AR5210, и его версия³ для ОС OpenBSD. Это позволяет получить дополнительную информацию об архитектуре микросхем семейства AR521x.

Драйвер `madwifi` поддерживает возможность подстройки параметров (ACK timeout, CTS timeout) протокола MAC-уровня 802.11. Эта подстройка необходима для работы на значительных расстояниях (6–8 км и более).

Подстройка производится путем записи величины тайм-аута в микросекундах в файлы `/proc/sys/dev/ath0/acktimeout`, `/proc/sys/dev/ath0/ctstimeout`. Для расстояния 10 км минимально необходимое значение тайм-аутов составляет 85–90 (определенено

¹<http://madwifi.sourceforge.net>

²<http://team.vantronix.net/ar5k>

³<http://www.openbsd.org/cgi-bin/cvsweb/src/sys/dev/ic/ar5xxx.c>

экспериментально).

Текущая (август 2004 г) версия драйвера не позволяет регулировать чувствительность приемника (`iwconfig sens`). Возможность снижения чувствительности полезна при работе с антенной с большим коэффициентом усиления для устранения паразитного приема слабого сигнала от чужих беспроводных сетей. Следует учитывать, что при снижении чувствительности должна быть по меньшей мере на ту же величину снижена мощность передатчика, чтобы излучаемый сигнал не создавал помех работе чужой беспроводной сети.

В используемой версии драйвера (август 2004 г) алгоритм автоматического выбора скорости передачи опирается на статистику числа перепосылок. При увеличении числа перепосылок происходит автоматическое снижение скорости передачи. Поскольку при связи на больших расстояниях перепосылки могут быть вызваны не проблемами демодуляции сигнала, а проблемами арбитража доступа к среде, алгоритм без надобности снижает скорость вплоть до минимальной (1Мбит/с). На данный момент проблема решена путем установки фиксированной скорости (`iwconfig rate 11M fixed`), но это неполноценное решение, так как при действительном снижении качества сигнала (например, во время мокрого снега) не происходит снижения скорости, которое необходимо для сохранения работоспособности канала. В последних версиях драйвера процедура выбора скорости вынесена в отдельный модуль, что позволит в будущем провести эксперименты по реализации специального алгоритма выбора скорости, подходящего для больших расстояний.

2.3.2.1 Средства отладки и мониторинга

Поставляемая с драйвером `madwifi` программа `athstats` позволяет вывести накопленную драйвером статистику (рис. 2.4). При запуске с численным аргументом N программа `athstats` показывает изменения собранной статистики с периодом N секунд (рис. 2.5).

При работе радиоинтерфейса в режиме базовой станции (Master) возможно получение информации о клиентах, ассоциированных в данный момент с базовой станцией, из файла `/proc/sys/net/wlan1/associated_sta` (рис. 2.6).

Возможно включение отладочной выдачи из драйвера в `/var/log/syslog`:

- Включение отладки MAC-протокола 802.11 осуществляется записью битовой маски `/proc/sys/net/wlan0/debug`; значения битов `IEEE80211_MSG_*` определены в файле `net80211/ieee80211_var.h` в исходных текстах драйвера.
- Включение отладки остальной части драйвера осуществляется записью битовой

```
wave-router:~# athstats
93 tx management frames
3 tx encapsulation failed
56 tx failed 'cuz too many retries
904 short on-chip tx retries
1218701 long on-chip tx retries
22735 tx frames with no ack marked
4608 tx frames with rts enabled
6728255 tx frames with short preamble
7811 tx frames with an alternate rate
2703533 rx failed 'cuz of bad CRC
13052528 PHY errors
    13050452 CCK timing
    2076 CCK restart
36733 periodic calibrations
rssи of last ack: 24
rssи of last rcv: 32
```

Рис. 2.4: Пример выдачи статистики драйвера `madwifi`

маски `/proc/sys/dev/ath/debug`; значения битов `ATH_DEBUG_*` определены в файле `ath/if_ath.c` в исходных текстах драйвера.

2.3.2.2 Дампирование трафика

При дампировании трафика на радиоинтерфейсе в дамп включаются только пакеты уровня 802.3, пакеты уровня 802.11 отсутствуют (рис. 2.7).

Возможно получение дампа пакетов уровня 802.11 путем перевода интерфейса в режим Monitor (рис. 2.8). Однако в режиме Monitor интерфейс выполняет только функцию пассивного мониторинга эфира, и перестает выполнять функцию передачи данных.

Возможна организация мониторинга путем установки дополнительного радиоинтерфейса и подключения его к той же антенне через антенный разветвитель Град 2410.

Кроме того, разработан пакет изменений (patch) к драйверу `madwifi`, позволяющий дампировать пакеты уровня 802.11, принимаемые (но не передаваемые) радиоинтерфейсом на вспомогательном устройстве `ath0mon`. Пакет изменений приложен к отчету на носителе (файл `live-monitoring-patch.txt`).

camp-router:~# athstats 1												
	input	output	altrate	short	long	xretry	crcerr	crypt	phyerr	rssi	rate	
8056125	5407648	11185	54261	1007446	58	49811	0	1272274	16	11M		
1	1	0	0	0	0	0	0	0	1	17	11M	
176	112	3	0	83	0	0	0	0	77	22	11M	
296	185	6	0	138	0	0	0	0	144	24	11M	
316	183	1	0	129	0	0	0	0	117	24	11M	
294	172	4	0	110	0	0	0	0	147	24	11M	
59	30	0	0	16	0	0	0	0	24	16	11M	
1	1	0	0	0	0	0	0	0	0	14	11M	
1	1	0	0	1	0	0	0	0	3	14	11M	
1	1	0	0	2	0	0	0	0	0	15	11M	

Рис. 2.5: Пример выдачи изменений (с интервалом 1 с), собранной статистики драйвера `madwifi`

Мониторинг пакетов уровня 802.11 с успехом применяется при наведении антенн: в заголовках пакетов уровня 802.11 (Prism2 monitoring header) присутствует значение уровня сигнала, с которым был принят данный пакет. При наведении настраивается мониторинг пакетов, поступающих от конкретного источника (фильтрация по MAC-адресу) и выполняется регулировка антенны для максимизации уровня принимаемого сигнала (рис. 2.9).

Мониторинг пакетов полезен также для диагностики всевозможных проблем MAC-уровня 802.11 и обнаружения помех от чужих беспроводных сетей.

2.3.3 Антенны

На магистральном уровне используются параболические антенны с усилением 27dBi отечественного производства (ЗАО “НИИДАР-ГРАД”⁴), см. рис. 1.3 и табл. 1.9. Высокий коэффициент усиления антенны позволяет обеспечить необходимый уровень сигнала без использования антенных усилителей на больших расстояниях (10–15 км) и при наличии препятствий на пути распространения сигнала (например, отдельно стоящие деревья).

⁴<http://www.antenna.ru>

```
wave-router:~# m /proc/sys/net/wlan0/associated_sta

macaddr: <00:0d:88:8b:d4:eb>
    rssi: 26 dBm
    capinfo: ess privacy shortpreamble
    freq: 2457 MHz (channel 10)
    opmode: b pureg
    txrate: 1Mbps 2Mbps 11Mbps*
    txseq: 24461 rxseq: 10912
    fails: 0 inact: 59

macaddr: <00:0d:88:82:ed:2d>
    rssi: 33 dBm
    capinfo: ess privacy shortpreamble
    freq: 2457 MHz (channel 10)
    opmode: b pureg
    txrate: 1Mbps 2Mbps 11Mbps*
    txseq: 59557 rxseq: 25248
    fails: 0 inact: 57
```

Рис. 2.6: Информация о клиентах, ассоциированных с базовой станцией

2.3.4 Кабели и разъемы

Для минимизации потерь в кабеле на магистральных каналах используется кабель с низкими потерями (Belden 9913, 0.26dB/m). За счет установки контейнера на том же столбе, на котором размещается антенна, длина кабеля небольшая (2–5 м).

Антенны и устройства грозозащиты оборудованы разъемами N-type. Для подключения фидера к разъему радиокарты (тип разъема RP-SMA) используется короткий (30 см) кабель-переходник, выполненный из кабеля RG-58.

2.4 Термобокс

Изучив опыт и изделия других других разработчиков, в частности, опыт польской компании ZPAS (Zaklad Produkcji Automatyki Sieciowej) нами был спроектирован и изготовлен *термобокс* — вандалоустойчивый металлический контейнер собственной разра-

```
wave-router:~# tethereal -lni ath0
wave-router:~# tethereal -lni ath0
Capturing on ath0
0.000000 00:30:4f:31:09:4a -> ff:ff:ff:ff:ff:ff ARP
Who has 192.168.66.129? Tell 192.168.73.2
0.001180 00:0d:88:8b:d4:eb -> 00:30:4f:31:09:4a ARP
192.168.66.129 is at 00:0d:88:8b:d4:eb
0.001230 193.232.174.1 -> 192.168.66.129 ICMP Echo (ping) request
0.002895 192.168.66.129 -> 193.232.174.1 ICMP Echo (ping) reply
```

Рис. 2.7: Дампирование трафика на радиоинтерфейсе

```
wave-router:~# iwconfig ath1 mode Monitor
wave-router:~# tethereal -lni ath1
wave-router:~# tethereal -lni ath1 | head
Capturing on ath1
0.000000 00:30:4f:31:09:4a -> ff:ff:ff:ff:ff:ff IEEE 802.11 Beacon frame
0.102395 00:30:4f:31:09:4a -> ff:ff:ff:ff:ff:ff IEEE 802.11 Beacon frame
0.129699 00:30:4f:31:09:4a -> 00:0d:88:8b:d4:eb IEEE 802.11 Data
0.130100 00:30:4f:31:09:4a -> 00:0d:88:8b:d4:eb IEEE 802.11 Data
0.130288 -> 00:30:4f:31:09:4a (RA) IEEE 802.11 Acknowledgement
0.131175 00:0d:88:8b:d4:eb -> 00:30:4f:31:09:4a IEEE 802.11 Data
0.131272 -> 00:0d:88:8b:d4:eb (RA) IEEE 802.11 Acknowledgement
0.204789 00:30:4f:31:09:4a -> ff:ff:ff:ff:ff:ff IEEE 802.11 Beacon frame
0.236006 00:0d:88:8b:e9:1c -> ff:ff:ff:ff:ff:ff IEEE 802.11 Probe Request
0.307190 00:30:4f:31:09:4a -> ff:ff:ff:ff:ff:ff IEEE 802.11 Beacon frame
```

Рис. 2.8: Получение дампа пакетов уровня 802.11 путем перевода интерфейса в режим Monitor

ботки, предназначенный для размещения оборудования связи вне помещения (outdoor). Термобокс состоит из двух камер (рис. 2.10):

- *Верхняя камера термобокса* — термоизолированная камера, в которой размещается оборудование, для которого необходимо выдерживать рабочий диапазон тем-

```
wave-router:~# tethereal -lni ath0mon -V  
      '-R wlan.addr eq 00:0D:88:8B:D4:EB' | grep Signal  
Capturing on ath0mon  
  Signal: 0x1c (DID 0x60044, Status 0x0, Length 0x4)  
  Signal: 0x1c (DID 0x60044, Status 0x0, Length 0x4)  
  Signal: 0x1b (DID 0x60044, Status 0x0, Length 0x4)  
  Signal: 0x1d (DID 0x60044, Status 0x0, Length 0x4)  
  Signal: 0x1a (DID 0x60044, Status 0x0, Length 0x4)  
  Signal: 0x1d (DID 0x60044, Status 0x0, Length 0x4)  
  Signal: 0x1d (DID 0x60044, Status 0x0, Length 0x4)  
  Signal: 0x1a (DID 0x60044, Status 0x0, Length 0x4)  
  Signal: 0x1e (DID 0x60044, Status 0x0, Length 0x4)  
  Signal: 0x1c (DID 0x60044, Status 0x0, Length 0x4)
```

Рис. 2.9: Мониторинг пакетов и регулировка антенны для максимизации уровня принятого сигнала

ператур.

- *Нижняя камера термобокса* — нетермоизолированная камера, где находится оборудование, для работы которого температурный режим не критичен.

Термобокс оснащен терморегулятором, вентилятором и нагревателем. Терморегулятор в зависимости от температуры включает либо охлаждающий вентилятор, либо нагреватель. Термобокс приспособлен для монтажа вне помещений на высотных сооружениях (мачты, столбы, здания).

2.4.1 Технические характеристики

2.4.1.1 Массо-габаритные характеристики

Внешние размеры контейнера — $600 \times 400 \times 250$ мм (без учета элементов крепежа). Имеется также вариант контейнера размером $650 \times 400 \times 250$ мм. Вес нетто контейнера составляет 14 кг.

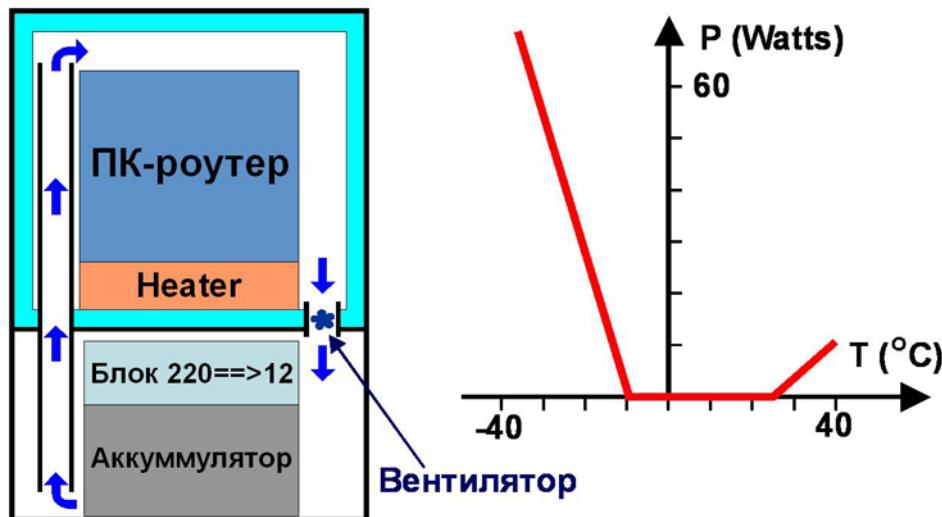


Рис. 2.10: Схема термобокса и расчетная зависимость мощности, расходуемой на поддержание рабочей температуры внутри контейнера от температуры среды



Рис. 2.11: Термобокс: вид спереди / сзади

2.4.1.2 Обеспечение вандалоустойчивости

Контейнер выполнен из металлического листа толщиной 2 мм. Передняя стенка контейнера представляет собой сварную раму из металлического профиля с сечением 40×20 мм, что обеспечивает необходимую жесткость всей конструкции и надежное прилегание дверцы контейнера. Эскизные чертежи термобокса приведены в данном разделе:

- 3D-схема — рис. 2.13, стр. 74;



Рис. 2.12: Термобокс: вид внутри

- вид сбоку — рис. 2.14, стр. 75;
- вид спереди — рис. 2.15, стр. 76.

Дверь контейнеры имеет рамную основу из металлического профиля с сечением 20×20 мм и выполнена из металлического листа толщиной 2 мм. В конструкции используется внутренний, четырехригельный замок, сувальдного типа. Так же применяются дополнительные штыри, предотвращающие снятие двери в случае, когда петли срезаны. Все это обеспечивает достаточную степень устойчивости контейнера к взлому.

2.4.1.3 Поддержание температурных режимов в термобоксе

При проектировании термобокса решались две проблемы: возможный перегрев аппаратуры в теплое время года и переохлаждение в холодное время года. Таким образом решалась основная задача — поддержание рабочего диапазона температур в верхнем отсеке термобокса.

В процессе проектирования были рассмотрены различные варианты обогрева и охлаждения верхней камеры. В частности, для охлаждения была предпринята попытка использовать элементы Пельтье (элементы, одна сторона которых нагревается, а другая одновременно охлаждается). Однако, в связи с низкой эффективностью данных элементов, высоким энергопотреблением, необходимостью применять либо пассивные теплообменники (радиаторы) большой площади либо системы принудительного охлаждения

пришлось отказаться от этой идеи.

В связи с тем, что в магистральных узлах необходимо обеспечивать бесперебойное электропитание оборудования, вопрос энергопотребления вспомогательного оборудования (нагреватели, вентиляторы и т. п.) становится чрезвычайно критичным, определяющим длительность автономной работы магистрального узла.

Поэтому для охлаждения верхней камеры было применен маломощный вентилятор (2Вт), который через воздуховоды нагнетает из нижней камеры воздух в верхнюю камеру до тех пор, пока в верхней камере не будет достигнута оптимальная температура.

В качестве теплоизоляции верхней камеры используется материал Пенофол® типа "А" (с односторонним фольгированием) 4×10 мм, производства ЗАО "Завод ЛИТ"⁵. Теплоизоляционные характеристики данного материала (табл. 2.3) позволяют поддерживать оптимальный температурный режим внутри изолированной секции при температурах внешней среды от -40°C до $+40^{\circ}\text{C}$ при минимальных энергозатратах (рис. 2.10). Выбранная толщина изоляционного материала позволяют поддерживать оптимальный температурный режим внутри рабочей камеры при разнице внутренней и внешней температур более 40°C . Расчет данного показателя велся с учетом тепловыделения ПК-маршрутизатора.

В качестве нагревательного элемента используется электрогрелка 220В/40Вт (цена: от 100 до 200 рублей).

Для используемого (в верхней камере термобокса) оборудования был установлен рабочий диапазон температур в верхней камере от 0°C до 305°C . Для автоматического управления подогревом и охлаждением верхней камеры, в которой расположено сетевое оборудование, используется термоконтроллер. Он представляет собой малогабаритное устройство, к которому подключены датчик температуры, датчик влажности, подогреватель и вентилятор охлаждения. В автоматическом режиме термоконтроллер:

- включает подогреватель, если температура в верхней камере термобокса снижается до уровня 5°C , и выключает его при повышении температуры до 10°C ;
- включает вентилятор, если температура в верхней камере термобокса превышает 35°C , и выключает его при снижении температуры до 30°C .

Зависимость энергопотребления нагревательного элемента и вентилятора от внешней температуры показано на рис. 2.10. При среднесуточной температуре -30°C на нагрев идет 50Вт/ч., при среднесуточной температуре -20°C энергопотребление составляет 25Вт/ч, в диапазоне от -10° до $+25^{\circ}$ энергопотребление на обогрев или охлаждение отсутствует.

⁵<http://www.botik.ru/lit/>

Таблица 2.3: Технические характеристики материала Пенофол® типа “А”

Параметры	Значения
Коэффициент теплового отражения (%)	90
Температура применения	от -60° до $+100^{\circ}$
Сопротивление теплопередачи R_0 , $\text{м}^2\cdot\text{С}/\text{Вт}$	1,14–1,36
Коэффициент теплопроводности l , при $+20.0^{\circ}$, не более $\text{Вт}/(\text{м}\cdot^{\circ}\text{С})$	0,037–0,038
Удельная теплоёмкость, С_0	1,95
Коэффициент теплоусвоения (при периоде 24 часа) s , $(\text{Вт}/\text{м}^2\cdot^{\circ}\text{С})$	0,45–0,51

Таким образом, минимальные затраты на поддержку оптимального температурного режима достигаются при температурах наружного воздуха от -10°C до $+25^{\circ}\text{C}$, что является типичными температурами для большей части года для большей территории средней полосы России.

2.4.1.4 Размещение оборудования

Размещение оборудование в контейнере производится по схеме, изображенной на рис. 2.10. В верхней термоизолированной камере находятся:

- нагревательный элемент;
- терморегулятор;
- сетевое оборудование: ПК-маршрутизатор (с блоком 12-ATX) и интерфейсное оборудование.

В нижней камере находится:

- Блок 220–12;
- Аккумуляторная батарея 12В, 17 А/ч;

Причина, по которой аккумуляторная батарея была вынесена за пределы термоизолированной камеры заключается в том, что чем ниже температура окружающей среды, тем выше срок эксплуатации аккумуляторной батареи.

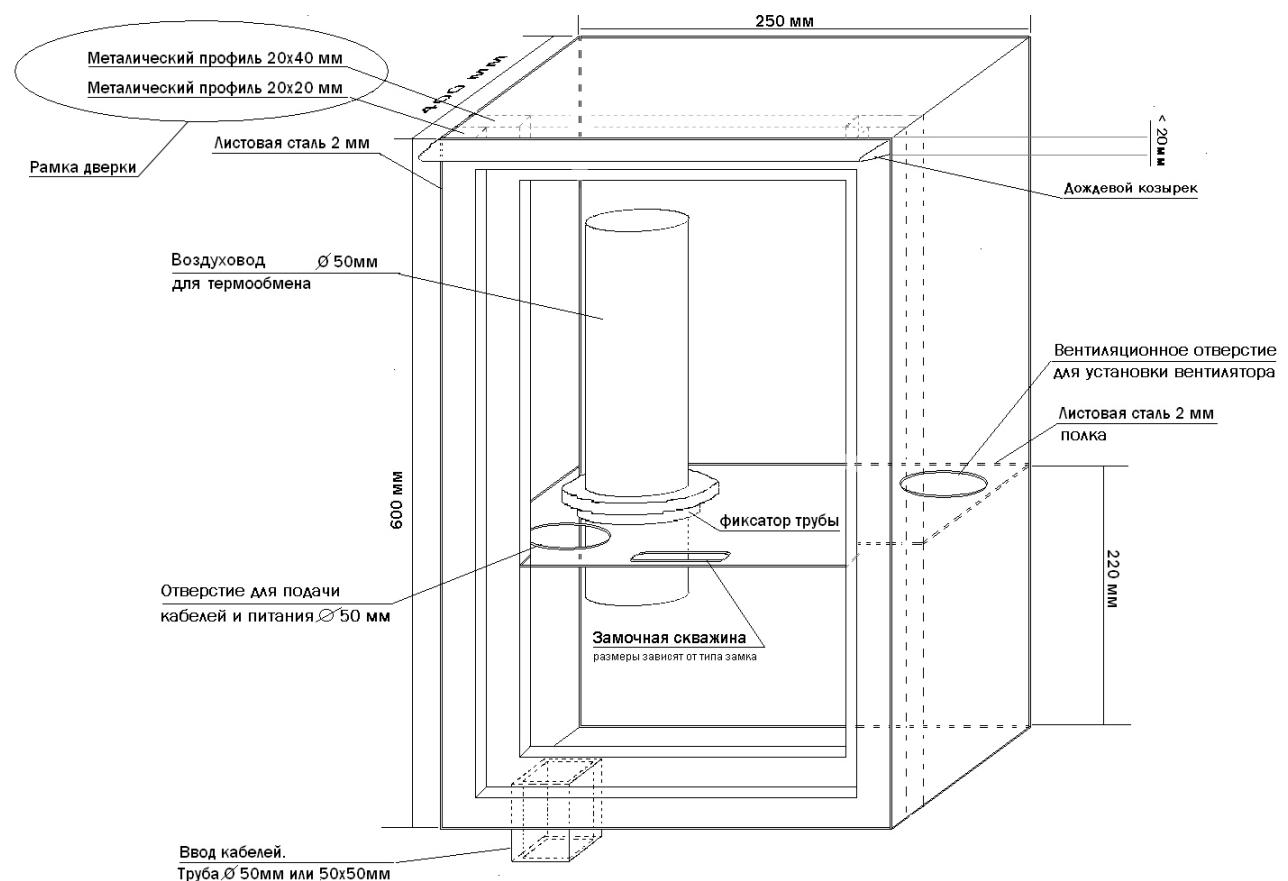


Рис. 2.13: Термобокс: 3D-схема

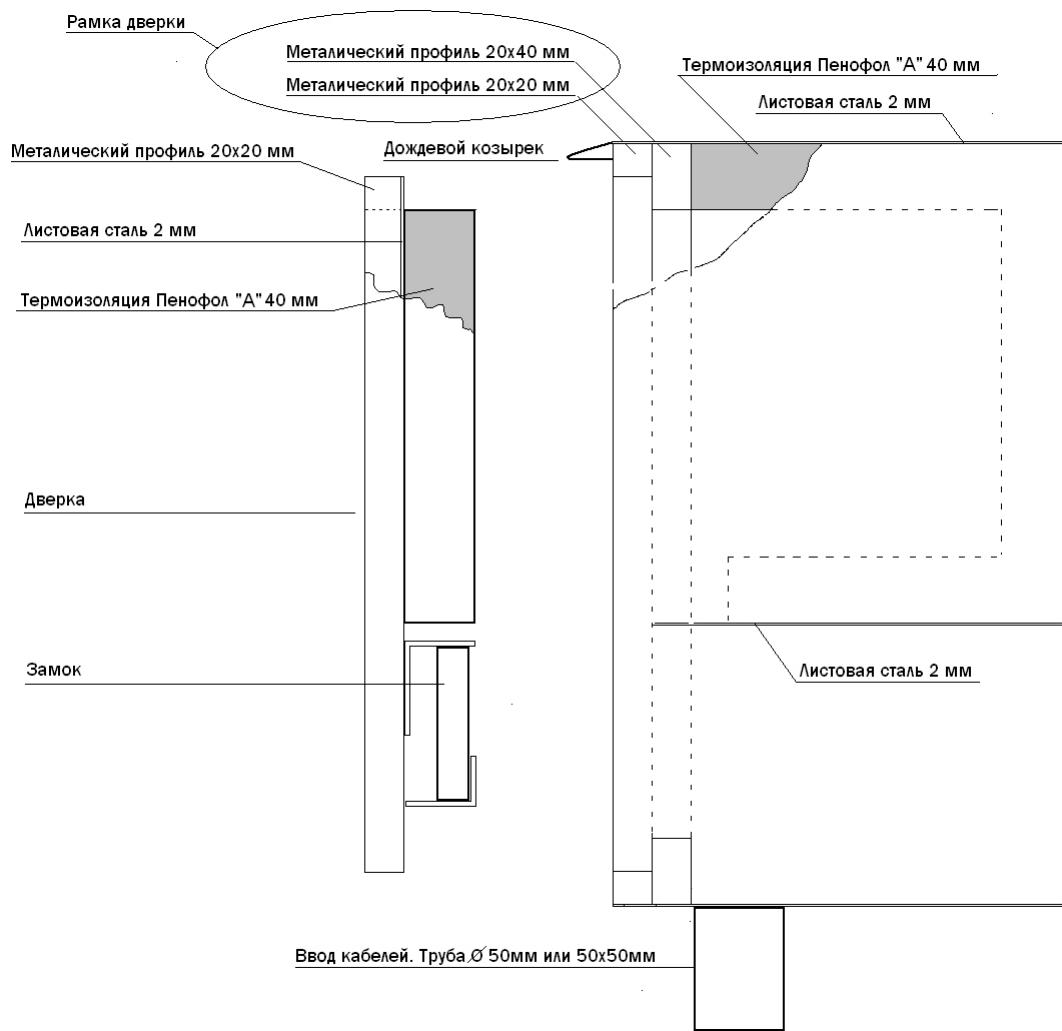


Рис. 2.14: Термобокс: вид сбоку

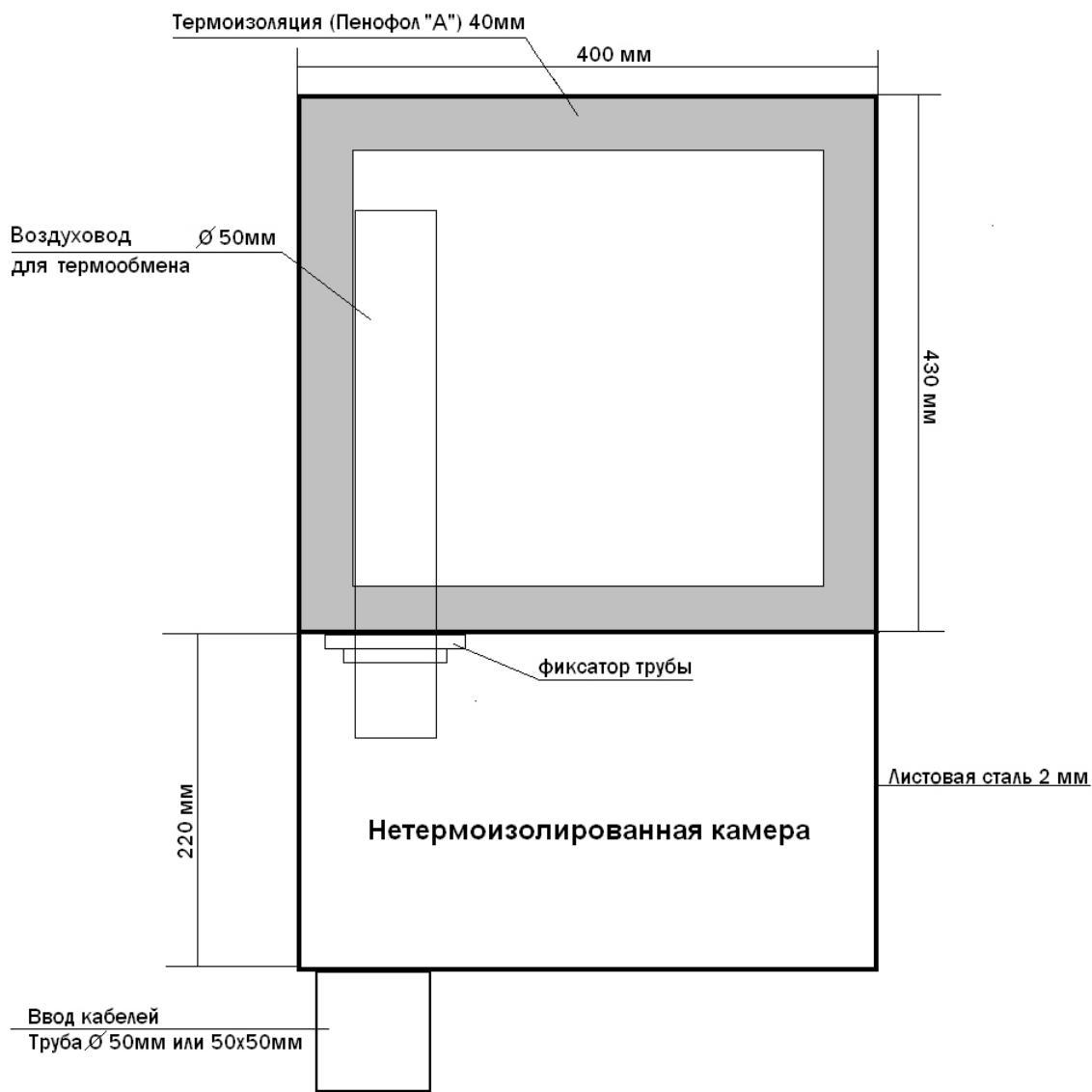


Рис. 2.15: Термобокс: вид спереди

2.5 Подсистема электропитания узла

К подсистеме электропитания узла предъявляются следующие требования:

- мощность 50 Вт:
 - 20Вт для зарядки аккумуляторной батареи;
 - 30Вт для питания сетевого оборудования: ПК-маршрутизатора и иного оборудования, например, коммутаторов и медиа-конвертеров, с напряжением питания 12В и суммарной потребляемой мощностью не более 30Вт;
- поддержание работоспособности узла при отсутствии внешнего питания в течение времени, типичного для длительности аварий электросети в сельской местности (около 8 часов)⁶;
- поддержание работоспособности узла при низком качестве внешнего питания (от 100В до 300В);
- передача ПК-маршрутизатору информации о состоянии системы электропитания (наличие внешнего электропитания, степень зарженности аккумуляторов);
- малые габариты и вес (поскольку данное оборудование размещается в ограниченном по объему термобоксе);
- отсутствие вентиляторов как потенциально ненадежного элемента;
- работоспособность при отрицательных температурах (чтобы подсистему электропитания можно было монтировать нижней камере термобокса, и тем самым уменьшить необходимый объем верхней камеры, а следовательно сократить теплопотери и энергозатраты на обогрев в зимнее время).
- стоимость, соответствующая ценовому масштабу остального оборудования узла (около \$100).

Перечисленные требования выполнены путем разработки собственной системы электропитания, структурная схема которой показана на рис. 2.16.

Основными компонентами системы электропитания магистрального узла являются:

- *Блок 220–12* — преобразователь входного напряжения внешней электросети 220В в напряжение 12В;

⁶С полезной нагрузкой 30Вт и со штатной аккумуляторной батареей, емкость которой можно и увеличить.

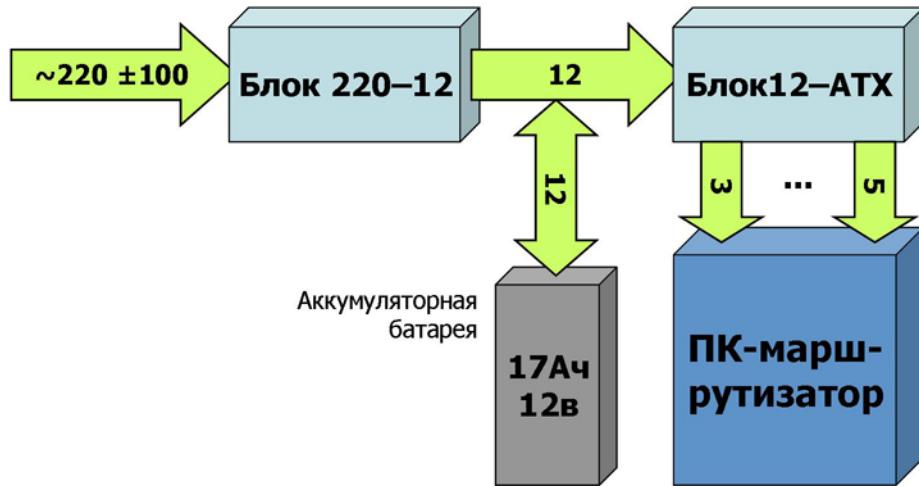


Рис. 2.16: Структурная схема системы электропитания

- Аккумуляторная батарея 12В, 17 А/ч;
- Блок 12-ATX, преобразующий напряжение аккумулятора (10–15В) в напряжения, необходимые для питания компьютера конструктива ATX (+3.3В, +5В, +12В, -5В, -12В), ()..

2.6 Блок 220-12

Блок 220-12 (рис. 2.17) является составной частью системы электропитания магистрального узла и предназначен:

- для зарядки аккумуляторной батареи;
- для питания сетевого оборудования (ПК-маршрутизатора и иного оборудования например, коммутаторов и медиа-конвертеров) с напряжением питания 12В и суммарной потребляемой мощностью не более 30Вт;

в условиях нестабильного электропитания. Основные технические характеристики блока указаны в табл. 2.4. Предельные характеристики Блока 220-12 указаны в табл. 2.5.

Блок 220-12 преобразует напряжение сети 220В в напряжение, необходимое для зарядки аккумулятора, контролирует процесс зарядки-разрядки и коммутирует нагрузку. Блок 220-12 включает в себя следующие узлы:

- сетевой фильтр;
- выпрямитель 220В;

Таблица 2.4: Технические характеристики Блока 220–12

Входное напряжение переменного тока	100 – 300В
Выходное напряжение	9.5В – 15.5В
Ток зарядки аккумулятора при входном напряжении > 150В	1.5А
Частота преобразования	100кГц
Коэффициент полезного действия (при выходном напряжении 13.6В и мощности нагрузки 30Вт)	80%

Таблица 2.5: Предельные эксплуатационные данные Блока 220–12

Входное напряжение переменного тока	480В
Входное напряжение (амплитудное значение)	680В
Выходное напряжение	25В
Ток аккумулятора	6А
Максимальный ток нагрузки	3А
Температура окружающей среды	от -40° до $+60^{\circ}$ С

- импульсный преобразователь напряжения;
- коммутатор нагрузки;
- схема управления.

Сетевой фильтр ограничивает проникновение помех от импульсного преобразователя напряжения в сеть 220В и защищает Блок 220–12 от влияния помех в сети 220В. Сетевой фильтр состоит из дросселя с синфазными обмотками L1 и специального пленочного конденсатора C1.

Выпрямитель преобразует переменное напряжение сети 220В в постоянное. Выпрямитель выполнен по мостовой схеме на диодах VD1–VD4. Выпрямленное напряжение сглаживается конденсаторами C2, C3. Для надежной работы при повышенном входном напряжении диоды выпрямителя выбраны с повышенным максимальным обратным напряжением (1000В), а конденсаторы с допустимым напряжением 350В включены последовательно. Ток зарядки конденсаторов при включении блока ограничивается на безопасном для выпрямителя уровне терморезистор RK1. Резисторы R4 и R5 образуют делитель напряжения для выравнивания напряжений на конденсаторах C2, C3.

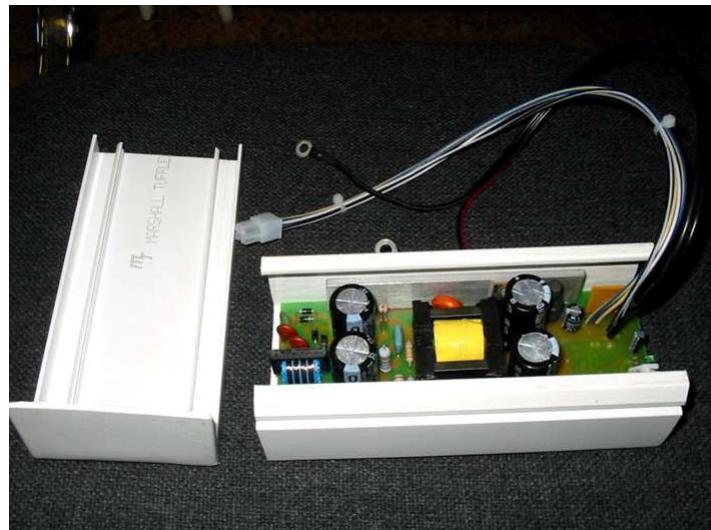


Рис. 2.17: Блок 220–12

Ток, протекающий через делитель, используется для питания ШИМ контроллера DA1 в режиме запуска.

Импульсный преобразователь выполнен по схеме обратноходового преобразователя с фиксированной частотой преобразования. Эта схема исключает прямое прохождение импульсной внешней помехи с входа на выход преобразователя, что выгодно отличает ее от прямоходового и двухтактного преобразователей. ШИМ контроллер DA1 включен по нестандартной схеме: обратная связь по выходному напряжению подключена к выходу усилителя сигнала ошибки (вывод СМР). Это упростило схему и позволило использовать этот усилитель для контроля входного напряжения преобразователя: вход усилителя подключен к делителю R1–R2–R3. При превышении входным переменным напряжением 300В (425В амплитудное значение), напряжение на выходе делителя превысит пороговые 2.5В и преобразователь выключится. В выключенном состоянии преобразователь способен выдержать повышенное входное напряжение, до 480В переменного тока. Частота работы преобразователя и максимальная скважность импульсов задаются цепочкой R8–C5. Максимальная скважность ограничена на уровне 0.57, поэтому при любом входном напряжении преобразователь работает в режиме прерывистых токов, что благоприятно сказывается на уровне пульсаций выходного напряжения.

ШИМ контроллер управляет силовым транзистором VT1. Использован транзистор типа 2SC2717 с максимально допустимым затворным напряжением 30В. Это позволяет варьировать напряжение питания ШИМ-контроллера в пределах 10 – 30В, соответственно напряжение на выходе блока может изменяться от 6 до 20В, что с запасом перекрывает необходимый для зарядки аккумулятора диапазон (9.5В – 15.5) В режиме

запуска ШИМ-контроллер питается током делителя R4–R5, затем напряжение питания вырабатывается диодом VD7 отдельной обмотки выходного трансформатора T1. Цепь VD6,C4,R6 и R7 ограничивает выбросы, вызванные индуктивностью рассеивания T1, на безопасном для силового транзистора уровне.

Трансформатор T1 преобразует импульсы с выхода силового транзистора по амплитуде и току и развязывает вход и выход преобразователя по постоянному току. Высоковольтный конденсатор C13 компенсирует помехи. Импульсное напряжение с выхода T1 выпрямляется диодами VD8 и сглаживается выходным фильтром C16–C18 и L2. Использованы диоды Шоттки с большим запасом по току, которые имеют малое падение напряжения в прямом направлении. Это уменьшает тепловыделение и позволяет использовать небольшой радиатор для теплоотвода. С этой же целью использовано параллельное включение диодов и вторичных обмоток T1. Демпфирующие цепочки R23–C15 и R24–C14 уменьшают помехи переключения диодов.

Для повышения надежности преобразователя используется три петли обратной связи: по выходному напряжению, по току силового транзистора и по напряжению питания DA1. Основная обратная связь, по выходному напряжению, состоит из параллельного стабилизатора DA2 и оптрана VU1. При повышении выходного напряжения увеличивается ток через светодиод оптрана и открывается его фототранзистор, который уменьшает напряжение на входе СМР контроллера DA1, в результате уменьшается скважность импульсов преобразователя, и выходное напряжение стабилизируется.

Обратную связь по току силового транзистора образует датчик тока на резисторах R12–R14 и фильтр низких частот R10–C10. Ток транзистора ограничивается на уровне 1.6 А, при этом максимальный выходной ток преобразователя уменьшается при уменьшении входного напряжения. Это допустимо, так как в первую очередь происходит уменьшение зарядного тока аккумулятора.

Обратная связь по напряжению питания DA1 — резервная, на случай отказа основной обратной связи. При этом напряжение питания DA1 повышается, пока не превысит напряжение стабилитрона VD5. После этого проходящий через стабилитрон ток увеличивает напряжение на входе токового датчика ISC DA1 и напряжение питания и пропорциональное ему выходное напряжение стабилизируется на уровне 15–16 В.

Коммутатор нагрузки реализован на транзисторе с низким сопротивлением в открытом состоянии IRF7424 (VT3). Ключ на VT2 и R25, R26 преобразует логический уровень схемы управления в необходимый для VT3 уровень. Делитель R27–R28 защищает VT3 от пробоя по затвору при повышенном выше 20 В выходном напряжении, кроме того совместно с входной емкостью VT3 делитель образует RC-цепочку с постоянной времени порядка 10 мксек, процесс открывания или закрывания VT3 происходит

плавно, благодаря этому уменьшаются броски тока при коммутации нагрузки.

Схема управления реализована на микроконтроллере Attiny26 (DD1) и выполняет следующие функции:

- измеряет ток нагрузки;
- измеряет ток заряда/разряда аккумулятора;
- измеряет напряжение на аккумуляторе;
- измеряет температуру аккумулятора;
- подстраивает напряжение зарядки блока аккумулятора в зависимости от его температуры и степени разреженности;
- коммутирует нагрузку в зависимости от степени разрядки аккумулятора, тока потребления или по внешнему сигналу;
- индицирует состояние блока.

Для измерения напряжения, температуры и токов используется встроенный АЦП, коммутатор и дифференциальный усилитель микроконтроллера DD1. В качестве датчиков тока используются проводники печатной платы (R29 и R30), которые начинаются у выходного конденсатора блока и заканчиваются в месте подпайки проводов аккумулятора и нагрузки. Для измерения падения напряжения на R29 и R30 используются дифференциальные входы ADC0, ADC1, ADC2, R31–R33, C23 и C24 отфильтровывают высокочастотные помехи.

Для измерения температуры аккумулятора используется внешний терморезистор на 22кОм, подсоединяемый к разъему X10. Вместе с R36 терморезистор образует делитель напряжения, подключенный к встроенному опорному напряжению микроконтроллера DD1.

Для управления блоком может использоваться кнопка В1 или внешняя кнопка, подсоединеная к разъему X9. R41 и R38 защищают входы микроконтроллера при ошибочном подключении и при воздействии статического напряжения.

Для программирования, тестирования внешнего управления и получения статистики от микроконтроллера предусмотрен разъем сервисного интерфейса. В разъем выведены сигналы программирования Reset, MOSI, MISO, SCK и сигналы последовательного интерфейса SCL и SDA, которые могут использоваться как прием и передача RS-232. Резисторы R39 и R40 защищают входы от статического электричества и нормируют уровни RS-232 к стандартным логическим уровням.



Рис. 2.18: Аккумуляторная батарея

Напряжение питания микроконтроллера формирует стабилизатор DA3, фильтр L3 C22 препятствует прохождению помех от цифровой части микроконтроллера в его аналоговую часть. Стабильную работу DA3 обеспечивают конденсаторы C19 и C20. Диоды VD9 и VD10 позволяют использовать для питания микроконтроллера внешний блок питания, подключаемый к разъему сервисного интерфейса, это удобно при программировании и наладке изделия. Благодаря малому току потреблению DA3 и использованию sleep-режима микроконтроллера DD1 средний ток потребления схемы не превышает 2mA, что сравнимо с током саморазряда аккумулятора.

Дополнительные материалы по Блоку 220–12 вынесены в приложения:

- Расчеты характеристик, перечень элементов, принципиальная схема и монтажный чертеж Блока 220–12 представлены в Приложении В.1.
- Описание программного обеспечения приведено в Приложении С. Текст программы микроконтроллера приложен к отчету на носителе ([upss-20041106.zip](#)).
- Принципиальная схема Блока 220–12 приведена в Приложении В.3.

2.7 Аккумуляторная батарея

При выборе аккумуляторной батареи блока электропитания магистрального узла учитывалось следующее:

- требование по напряжению — батарея должна обеспечить непосредственное питание ПК-маршрутизатора как в процессе подзарядки, так и в процессе автономной работы;
- требование по емкости — должно было быть обеспечено не менее 8 часов автономной работы ПК-маршрутизатора;
- экологические и гигиенические требования — должно допускаться расположение батареи в рабочих помещениях с присутствием персонала, батарея не должна выделять вредные для человека вещества;
- иные требования, к которым можно отнести: массогабаритные характеристики, срок службы, рабочий температурный диапазон и т.п.

В результате исследования большого числа аккумуляторных батарей была выбрана батарея NPL17-12I фирмы YUASA Battery (UK) Ltd.⁷, со следующими характеристиками:

- напряжение заряда в буферном режиме 13.5–13.8 В;
- напряжение разряда: 12.5–10.5 В;
- емкость при разряде током 1А: 17Ah;
- тип: свинцово-гелевая, необслуживаемая (sealed lead-acid, Pb), допускается использование в помещениях с людьми, батарея не выделяет вредные для человека вещества;
- срок службы: 7 – 10 лет.

Время автономной работы ПК-маршрутизатора от батареи NPL17-12I до 16-ми часов⁸ отвечает требованиям по длительности автономной работы (не менее 8 часов). Кроме того, данная батарея имеет хорошие массогабаритные характеристики и приемлемую стоимость.

Допустимо в системе электропитания магистрального узла использовать аналогичные аккумуляторные батареи других производителей (рис. 2.18).

⁷<http://www.yuasa-battery.net>

⁸Измеренная мощность, потребляемая ПК-маршрутизатором с двумя интерфейсами 802.11 и один интерфейсом 802.3 при средней нагрузке (100% загрузка трафиком обоих беспроводных интерфейсов) составила 12.65 Вт. При потреблении такой мощности от аккумулятора напряжением 12В ток разряда составит 1.05А. При емкости аккумулятора 17Ач время автономной работы составит 16.13ч. При необходимости увеличения времени автономной работы возможно использование аккумулятора большей емкости.

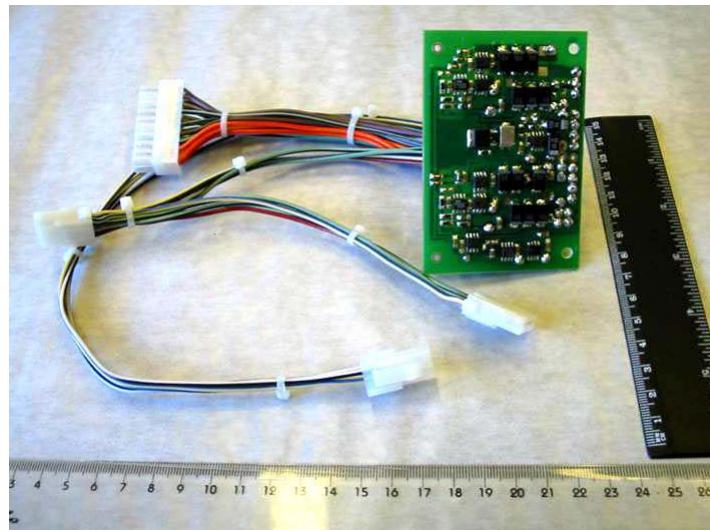


Рис. 2.19: Блок 12-ATX

2.8 Блок 12-ATX

Блок 12-ATX является составной частью системы электропитания магистрального узла и предназначен для питания ПК-марштуризатора на базе процессора Intel Pentium III. Допустимо использовать Блок 12-ATX и для питания ПК-марштуризатора на других процессорах (Intel i486, Intel Pentium I/II).

Блок 12-ATX преобразует напряжение аккумуляторной батареи (10–15В) в напряжения, необходимые для питания компьютера конструктива ATX (+3.3В, +5В, +12В, -5В, -12В) и вырабатывает сигнал PG (PowerGood — напряжение в норме).

Блок 12-ATX не содержит механически-подвижных частей (вентиляторов). Полнотью заряженная штатная аккумуляторная батарея в (может наращиваться) должна поддержать автономную работу штатного ПК-марштуризатора в течение 8 часов и более. Основные технические характеристики блока питания указаны в табл. 2.6

Блок 12-ATX включает в себя следующие узлы:

- преобразователь 12В–5В;
- преобразователь 12В–3.3В;
- линейный стабилизатор +12В;
- стабилизаторы отрицательных напряжений -12В и -5В;
- формирователь сигнала PG;

Таблица 2.6: Технические характеристики Блока 12–ATX

Входное напряжение	9.6–16В
Выходные напряжения	+5В ±2% +3.3В ±2% +12В +5..–20% -5В ±5% -12В ±5%
Коэффициент полезного действия *) — при входном напряжении 12В и мощности нагрузки 15Вт	95%*)

Таблица 2.7: Предельные эксплуатационные данные Блока 12–ATX

Входное напряжение	20В
Выходной ток 5В	10А
Выходной ток 3.3В	10А
Выходной ток 12В	1А (внутреннее ограничение)
Выходной ток -5В	0.1А (внутреннее ограничение)
Выходной ток -12В	0.1А (внутреннее ограничение)
Температура окружающей среды	0 – 60° С

- вспомогательный стабилизатор +6В для питания преобразователей 12В – 5В и 12В – 3.3В.

Преобразователь 12В–5В выполнен по схеме понижающего прямогоходового конвертора с синхронным выпрямлением. В качестве силовых МОП-транзисторов VT1, VT2 использованы IRF7828 с низким сопротивлением в открытом состоянии (0.01Ом) и небольшими внутренними емкостями (энергия переключения порядка 10nC). Благодаря этому достигнут высокий КПД преобразователя.

Для управления транзисторами в Блоке 12–ATX используется специализированная микросхема IRU3037 (DA2), которая содержит:

- задающий генератор на частоту 200кГц;
- ШИМ-контроллер;

- усилитель обратной связи;
- выходные драйверы верхнего и нижнего плеча;
- защиту от КЗ в нагрузке;
- схему отключения при понижении входного напряжения;
- схему “мягкого старта”.

Микросхема IRU3037 включена по рекомендованной производителем схеме. Она сравнивает выходное напряжение, поступающее через делитель R7–R9 на вход FB, с встроенным опорным напряжением 1.25В и формирует на выходах HD LD управляющие импульсы для силовых транзисторов VT1 и VT2 соответственно.

Повышенное напряжение для питания драйвера верхнего силового транзистора VT1 вырабатывается диодами VD1–VD3 из импульсного напряжения на выходе преобразователя. Диоды включены нестандартно: они запитаны от отдельного стабилизатора +6В (DA1). Благодаря этому даже при входном напряжении 25В напряжение питания драйвера верхнего плеча IRU3037 не превышает максимально допустимых 30В. (в стандартной схеме это происходит при входном напряжении 16В).

Цепочка R1–C9 компенсирует фазо-частотную характеристику узла. C10 определяет длительность нарастания выходного напряжения при запуске (“мягкий старт”) — около 75 мсек.

Резисторы R7–R9 образуют делитель обратной связи. Нижнее плечо делителя образуют два параллельно включенных резистора, это позволяет скорректировать выходное напряжение на этапе наладки Блока 12–ATX.

L1 C4–C6 образуют входной, а L3 C24–C27 — выходной фильтры преобразователя.

Преобразователь 12В–3.3В аналогичен преобразователю 12В–5В, отличие только в делителе обратной связи. Преобразователь 12В–3.3В выполнен по схеме понижающего прямогоходового конвертора с синхронным выпрямлением. В качестве силовых МОП-транзисторов VT1, VT2 использованы IRF7828 с низким сопротивлением в открытом состоянии (0.01Ом) и небольшими внутренними емкостями (энергия переключения порядка 10nC). Благодаря этому достигнут высокий КПД преобразователя.

Для управления транзисторами в Блоке 12–ATX используется специализированная микросхема IRU3037 (DA3), которая содержит:

- задающий генератор на частоту 200кГц;
- ШИМ-конроллер;

- усилитель обратной связи;
- выходные драйверы верхнего и нижнего плеча;
- защиту от КЗ в нагрузке;
- схему отключения при понижении входного напряжения;
- схему “мягкого старта”.

Микросхема IRU3037 включена по рекомендованной производителем схеме. Она сравнивает выходное напряжение, поступающее через делитель R10–R12 на вход FB, с встроенным опорным напряжением 1.25В и формирует на выходах HD LD управляющие импульсы для силовых транзисторов VT3 и VT4 соответственно.

Повышенное напряжение для питания драйвера верхнего силового транзистора VT3 вырабатывается диодами VD6–VD8 из импульсного напряжения на выходе преобразователя. Диоды включены нестандартно: они запитаны отдельного стабилизатора +6В (DA1). Благодаря этому даже при входном напряжении 25В напряжение питания драйвера верхнего плеча IRU3037 не превышает максимально допустимых 30В (в стандартной схеме это происходит при входном напряжении 16В).

Цепочка R2–C16 компенсирует фазо-частотную характеристику узла. C17 определяет длительность нарастания выходного напряжения при запуске (“мягкий старт”) — около 75 мсек.

Резисторы R10–R12 образуют делитель обратной связи. Нижнее плечо делителя образуют два параллельно включенных резистора, это позволяет скорректировать выходное напряжение на этапе наладки Блок 12–ATX. L1 C11–C13 образуют входной, а L4 C32–C35 — выходной фильтры преобразователя.

Стабилизатор напряжения +12В — компенсационный, с низким падением на регулирующем транзисторе. В качестве регулирующего VT5 использован IRLR2705: сопротивление в открытом состоянии 0.06Ом, пороговое напряжение 2В. Для обеспечения низкого падения напряжения на VT5 его затвор подключен через R3 к источнику с повышенным относительно питающего напряжением, в качестве которого используется питание драйвера верхнего транзистора преобразователя 12В — 5В. При нормальных условиях это напряжение на 4.5 — 5В выше входного питающего напряжения.

Параллельный стабилизатор TL431 (DA4), сравнивает выходное напряжение делителя обратной связи R4–R6 с встроенным опорным напряжением 2.5В и управляет выходным транзистором VT5.

Делитель обратной связи R4–R6 выполнен аналогично делителям преобразователей 12В – 5В и 12В – 3.3В, нижнее плечо делителя образуют два параллельно включенных

резистора, это позволяет скорректировать выходное напряжение на этапе наладки Блок 12–ATX.

К выходу транзистора VT5 подключен самовосстанавливающийся предохранитель F1(MF-SM100), который защищает Блок 12–ATX от превышения по току (1A) и по рассеиваемой мощности: при нагревании транзистора выше 120°C расположенный в непосредственной близости от него самовосстанавливающийся предохранитель отключит нагрузку. Конденсатор C22 корректирует частотную характеристику каскада, конденсатор C23 уменьшает импеданс выхода +12В Блока 12–ATX на высоких частотах.

Для получения выходных напряжений –12В и –5В на дросселе преобразователя 12В – 5В L1 намотана дополнительная обмотка L2. При синхронном выпрямлении в ней наводится электрический ток, который выпрямляется диодом VD5 и сглаживается С28, С30. Напряжение на С28 и С30 пропорционально выходному напряжению преобразователя 12В – 5В и при нормальных условиях составляет –14В. Маломощные линейные стабилизаторы DA5, DA6 доводят напряжение до –12В и –5В соответственно, кроме того они обеспечивают защиту от замыкания в нагрузке по этим выходам. Конденсаторы С29, С31 обеспечивают стабильность линейных стабилизаторов.

Выходной сигнал PG (PowerGood — напряжение в норме) — формируется по упрощенной схеме, контролируется только напряжение 3.3В. Это оправдано, так как современные материнские платы имеют встроенный контроль остальных напряжений, и 3.3В является для них основным питающим напряжением. При включении Блока 12–ATX микросхема монитора питания DA7 (MAX809T) контролирует выход +3.3В и через 140–560 мсек после достижения на нем 3.08В выдает “1” на выход PG.

Дополнительные материалы по Блоку 12–ATX вынесены в приложения:

- Расчеты характеристик, элементная база, принципиальная схема и монтажный чертеж Блока 220–12 представлены в Приложении D.
- Принципиальная схема Блока 12–ATX приведена в Приложении D.3.

2.9 Термоконтроллер

Термоконтроллер предназначен для поддержания заданного температурного режима (от 0°C до 305°C) в верхнем отсеке термобокса. Он представляет собой малогабаритное устройство, к которому подключены датчик температуры, датчик влажности, подогреватель и вентилятор.

В автоматическом режиме термоконтроллер:

- включает подогреватель, если температура в верхней камере термобокса снижается до уровня 5°C, и выключает его при повышении температуры до 10°C;
- включает вентилятор, если температура в верхней камере термобокса превышает 35°C, и выключает его при снижении температуры до 30°C.

Указанные пороговые значения температур хранятся в ЭСППЗУ микроконтроллера, и при необходимости могут быть изменены через управляющий интерфейс термоконтроллера. Кроме того, управляющий интерфейс термоконтроллера позволяет удаленно контролировать температуру и влажность в термобоксе, и осуществлять “ручное” управление подогревателем и вентилятором в непредвиденных случаях.

Дополнительные материалы по термоконтроллеру вынесены в приложения:

- Технические характеристики и перечень элементов термоконтроллера приведены в Приложении Е.
- Принципиальная схема термоконтроллера приведена в приложении Е.2.

Глава 3

Технические решения для абонентского комплекса беспроводного доступа

3.1 Общее описание

Задачей абонентского комплекса беспроводного доступа является обеспечение сетевой связности для компьютера (локальной сети) абонента, расположенного в пределах прямой видимости на расстоянии до 6 км от базовой станции беспроводного доступа. Роль базовой станции здесь выполняет магистральный узел, один из беспроводных интерфейсов которого сконфигурирован в режиме Master и подключен к всенаправленной антенне.

Специфика абонентского комплекса состоит в следующем:

- оборудование устанавливается на территории абонента (“под присмотром”) и нет необходимости применять стальные контейнеры для защиты от несанкционированного доступа;
- желательно, чтобы оборудование было компактным и в аккуратном корпусе;
- оборудование должно работать бесшумно;
- нет необходимости в источнике бесперебойного питания: абонентский комплект обслуживает единственного пользователя, и если в доме отключено электричество, компьютер абонента, скорее всего, тоже не работает. Впрочем, для системы мониторинга было бы полезно, если бы при отключении питания устройство проработало еще несколько секунд, чтобы успеть передать информацию об отключе-

нии электропитания, таким образом, обслуживающий сеть персонал знал бы, что проблема не в его зоне ответственности.

3.2 Выбор решения

Нами проведен анализ трех вариантов создания абонентских комплектов беспроводного доступа, оценивались потребительских качества получавшихся решений и их конечная стоимость.

- *Первый вариант:* установка сетевого адаптера беспроводного доступа в компьютер абонента (раздел 3.2.1).
- *Второй вариант:* установка ПК-маршрутизатора у абонента (раздел 3.2.2).
- *Третий вариант:* Использование мостов 802.11—802.3 (раздел 3.2.3).

3.2.1 Установка сетевого адаптера беспроводного доступа в компьютер абонента

Самым “очевидным” вариантом организации беспроводного доступа абонента к сети является установка сетевого адаптера беспроводного доступа в компьютер абонента. При этом стоимость абонентского комплекта для беспроводного доступа к сети составляет порядка \$290. В эту стоимость включается сетевой адаптер беспроводной связи (\$70) и антенно-фидерная система (при длине кабеля около 20 м), стоимостью \$220. У этого варианта два серьезных недостатка:

- Дорогая антенно-фидерная система, потери ВЧ сигнала в кабеле, необходимость прокладки толстого жесткого кабеля RG-8 до места расположения компьютера абонента, перенос компьютера внутри помещения затруднен;
- Параметры соединения устанавливает абонент. Нет гарантии их правильной установки, невозможно их оперативное (и тем более регулярное автоматическое) изменение.

Как следствие, данный вариант далее не рассматривается.

3.2.2 Установка ПК-маршрутизатора у абонента

Опробованным надежным способом реализации абонентского комплекта беспроводного доступа к сети является установка на площадке абонента ПК-маршрутизатора с

сетевым адаптером беспроводной связи. Маршрутизатор устанавливается в отапливаемом помещении, по возможности в верхней части здания. Локальная сеть абонента подключается к порту 100Base-TX маршрутизатора.

Преимущества данного варианта: маршрутизатор находится под управлением оператора сети и дает возможность оперативной модификации параметров соединения (частота, ключи шифрации, порог RTS и т.д), управления трафиком (QoS), сбора статистики по уровню и качеству приема и диагностики проблем радиосвязи с обоих концов соединения. Степень открытости архитектуры абонентского маршрутизатора такая же, как у магистрального узла, и могут применяться однотипные радиосредства и драйверы, что позволяет использовать эффективные схемы обеспечения безопасности (WPA). Управление маршрутизатором происходит по защищенному протоколу SSH.

Бескорпусные ПК-маршрутизаторы, применяемые в магистральных узлах, для абонентского комплекта не годятся: для установки в помещении необходим аккуратный корпус. Для решения данной проблемы возможна сборка ПК-маршрутизатора в стандартном корпусе ПК со стандартным блоком питания (поскольку бесперебойность питания не требуется).

Маршрутизатор для абонентского комплекта не требует большого количества слотов расширения. Необходимо всего два интерфейса: радио (802.11) и проводной Ethernet (802.3). Поэтому возможна сборка ПК-маршрутизатора на базе малоформатной материнской платы (mini-ITX) в малогабаритном корпусе, стоимость которого приведена в табл. 3.1.

Таблица 3.1: Оценка стоимости малогабаритного маршрутизатора

Материнская плата VIA EPIA VE5000A с процессором VIA C3	\$85
Корпус Morex CUBID 3677	\$90
SDRAM 128M	\$20
Flash Transcend TS128MDOMV (128MB)	\$80
Радиокарта D-Link DWL-G520	\$70
ИТОГО	\$345

Стоимость антенно-фидерной системы в данном случае варьируется в зависимости от расстояния и условий прохождения сигнала до базовой станции. В благоприятных случаях возможен прием на всенаправленную антенну 2 dBi, поставляемую с радиокартой, и стоимость антенно-фидерной системы оказывается “нулевой”. При больших расстояниях и наличии препятствий может использоваться антenna 24 dBi (\$70) с де-

шевым фидером (RG-58, затухание 1 dB/m, стоимость \$0.5 за метр).

3.2.3 Использование мостов 802.11—802.3

Стоимость абонентского комплекта можно значительно (примерно в три раза) снизить, если вместо предложенного во втором варианте ПК-маршрутизатора использовать одно из имеющихся на рынке устройств, выполняющих функции беспроводного моста (bridge). Имеется в виду мост между интерфейсами 802.11 и 802.3, причем интерфейс 802.11 должен работать в режиме беспроводного клиента (client, Managed mode).

Этот вариант имеет следующие недостатки по сравнению с установкой ПК-маршрутизатора у абонента (раздел 3.2.2):

1. закрытая архитектура устройства, невозможность программирования устройства (только предусмотренные производителем средства конфигурации), значительно меньше средств диагностики;
2. выбор средств безопасности ограничивается общим подмножеством возможностей базовой станции и моста.

Тем не менее, вариант использования мостов 802.11—802.3 имеет следующие достоинства по сравнению с установкой ПК-маршрутизатора у абонента:

1. компактность устройства;
2. сравнительно низкая стоимость (\$70–\$120).

3.3 Выбор оборудования для мостов 802.11—802.3

В качестве оборудования для мостов 802.11—802.3 нами рассматривались устройства, представленные в табл. 3.2. Для изучения нами были выбраны устройства фирмы D-Link DWL-900AP+ и DWL-2000AP+, привлекательные по цене¹, габаритам и типу разъемов. Накоплена следующая информация:

- устройство DWL-900AP+ построена на базе GL2422AP фирмы GlobalSun (Тайвань), устройство DWL-2000AP+ — на базе GL2454AP той же фирмы;
- Микрокод построен базе ОС реального времени TreadX фирмы Express Logic Inc²;

¹Цены на оборудование и количество предложений получены на момент написания отчета на основе данных из <http://www.price.ru>

²<http://www.expresslogic.com/txttech.html>

- DWL-2000AP+ (микрокод v115b13) поддерживает современную схему безопасности WPA-PSK, однако добиться ее работоспособность в паре с базовой станцией (драйвер `madwifi`, режим Master) пока не удалось;
- DWL-900AP+ и DWL-2000AP+ поддерживают стандартную схему безопасности WEP с длиной ключа 64, 128, 256 бит. Ключи длиной 256 бит не поддержаны в `madwifi`;
- DWL-2000AP+ (микрокод v115b13) поддерживает снижение излучаемой мощности на 12dB относительно номинальной шагами по 3dB; работоспособность регулировки подтверждена измерением уровня приема;
- DWL-2000AP+ и DWL-900AP+ при работе в режиме “клиент” не позволяют производить выбор базовой станции по SSID. Требуется явная установка MAC-адреса базовой станции, то есть при замене базовой станции или радиокарты на ней потребуется переконфигурация всех клиентов, или подмена MAC-адреса базовой станции.

3.4 Обеспечение безопасности

При выборе оборудования для беспроводных сетей существенным моментом являются соображения обеспечения безопасности. Безопасность беспроводных сетей вообще представляет собой серьезную, многостороннюю проблему, но в первую очередь требуется решить следующую задачу: обеспечить невозможность использования ресурсов сети лицом, обладающим совместимым радиооборудованием, но не являющимся зарегистрированным абонентом.

В стандарте 802.11 предусмотрены средства шифрации передаваемых данных (wired equivalent privacy, WEP) и механизм аутентификации (Shared key authentication), который опирается на использование WEP. Эти средства поддержаны во всех моделях оборудования и были бы достаточны для поставленной цели, но применение их требует учета ряда обнаруженных в них уязвимостей.

Поддержка более современной схемы обеспечения безопасности — WPA (Wi-Fi Protected Access) и WPA2 (IEEE 802.11i) начинает появляться в оборудовании стандарта 802.11g. В оборудовании стандарта 802.11b ее, как правило, нет. Таким образом, задача обеспечения безопасности делится на две:

1. выбрать абонентское оборудование, в котором поддержка WPA в режиме “клиент” уже есть, или есть уверенность в том, что она появится в последующих версиях

Таблица 3.2: Сравнение различного оборудования для мостов 802.11—802.3

Модель	Цена/кол-во предложе-ний	Разъем внешней антенны	Чипсет	Режим “кли-ент”	Примечания
D-Link DWL-G810	\$104/2			только клиент	Слабо представлена на рынке
D-Link DWL-900AP+/C2	\$65	RP-SMA	TI TNETW1100BGHK	да	Сертификат CCC. Только 802.11b.
D-Link DWL-2000 AP	\$90/1	RP-SMA	Atheros AR5212A	да	Сертификат CCC. По-видимому, снята с производства
D-Link DWL-2000AP+/A1	\$75/195	RP-SMA	TI TNETW1130GVF	да	Построена на базе GlobalSun GL2454AP
SMC2804WBR	\$115/22	RP-SMA	Prism54		Маршрутизатор, 4 порта. WPA.
Netgear WG602	\$136/5	RP-SMA	Prism54 (PrismGT)		Микрокод на базе Linux. Слабо представлена на рынке
Linksys WRT54G	\$90/24	RP-TNC	Broadcom	да	Маршрутизатор, 4 порта. WPA. Микрокод на базе Linux. Открытые исходные тексты кроме драйвера радио. Хорошо освоена сетевым сообществом. Чувствительность -65dBm@54Mbps на 3dB хуже, чем у изделий с чипсетом Prism.
Buffalo WBR2-G54	\$96/3	MC (orinoco)			Слабо представлена на рынке
Trendnet 410APB+	\$117/43	RP-SMA	TI?		
USR5450	\$137/8	RP-SMA		да	Построена на базе GlobalSun GL2454AP, как и D-link DWL-2000AP+
Linksys WET11	\$125			да	Только клиент (нет режима AP). Только 802.11b.
ASUS WL-500g	\$117	RP-SMA			Linux, но исходных текстов нет

микрокода, поставляемых производителем оборудования;

2. при отсутствии возможности использовать WPA, найти наилучший способ применения существующих средств безопасности (WEP).

3.4.1 Выбор оборудования с поддержкой WPA

Поддержка WPA в DWL-2000AP+ реализована, но работоспособность ее в режиме “клиент” пока не выяснена.

В качестве альтернативного варианта рассматривается устройство Linksys WRT54G, для которого микрокод поставляется с исходными текстами, и есть возможность его доработки. Для исследования нами заказаны два устройства WRT54G.

3.4.2 Уязвимости WEP

В рамках данной работы нами были рассмотрены следующие уязвимости WEP:

1. *Получение ключа методом “грубой силы”.* Для атаки достаточно получить один WEP-шифрованный пакет. Производится перебор ключей по словарю или исчерпывающий перебор в пространстве 2^{40} (для WEP-64) или 2^{104} (для WEP-128). Для каждого ключа делается попытка дешифрации пакета и проверка результата декодирования: быстрая (по известным сигнатурам в пакете) и медленная (ICV — integrity check value, CRC32).

Противодействие: атаку по словарю можно предотвратить путем использования в качестве ключа случайного числа, вводимого в шестнадцатеричном режиме. Полный перебор возможен при использовании WEP-64, но для WEP-128 он требует экстраординарных вычислительных ресурсов.

2. *Получение ключа методом FMS* (существуют программные средства: AirSnort, WEP Crack). Атака требует получения 10^5 – 10^6 шифрованных пакетов.

Противодействие: в современных реализациях WEP при генерации переменной части ключа (IV, initialization vector) учитывается возможность атаки FMS, уязвимые ключи не используются. Реализация WEP в драйвере `madwifi` содержит код, исключающий уязвимые ключи. Уязвимость реализации WEP в устройствах абонентского комплекта требует проверки для каждого типа устройства и версии микрокода.³

3. *Формирование шифрованных пакетов без знания ключа* (message injection). Для атаки требуется получение шифрованной и нешифрованной версии одного и того же пакета. По ним можно восстановить наложенную на шифрованный пакет крипто последовательность и использовать ее для создания поддельного шифрованного пакета.

Противодействие: невозможно. Для этого потребовалось бы запретить прием сообщений с ранее встречавшимся IV (initialization vector, переменная часть ключа), а это невозможно в силу ограниченности пространства IV (2^{24}). Осложнить задачу атакующему может частая смена ключа.

³По результатам тестирования реализации WEP в DWL-2000AP+ методом “черного ящика”, она не является уязвимой к атакам методом FMS.

3.4.3 Уязвимости механизмов аутентификации

Протокол Shared key authentication работает по следующему принципу: базовая станция передает клиенту случайное число (challenge), клиент возвращает пакет, содержащий это число, зашифрованный ключом WEP. Если базовая станция может расшифровать полученный пакет, значит клиент имеет верный ключ WEP, и аутентификация считается успешной.

Уязвимость механизма Shared key authentication проистекает из того, что challenge посыпается в нешифрованном виде. Это позволяет атакующему, перехватив запрос и ответ во время аутентификации другого клиента, использовать уязвимость WEP № 3, чтобы сформировать верный ответ в сессии Shared key authentication, не зная ключа. Кроме того, он может использовать полученную криптопоследовательность в дальнейшем, чтобы все посыпаемые им на базовую станцию пакеты были зашифрованы. В результате становится бесполезной опция безопасности базовой станции “принимать только шифрованные пакеты”. Таким образом, использования механизма Shared key authentication следует избегать.

Альтернативный протокол аутентификации — “Open system authentication” не имеет уязвимостей, но и не реализует никакой защиты. Если на базовой станции включен этот режим аутентификации, любой клиент, пославший запрос на аутентификацию, получает положительный ответ. Таким образом, чужой клиент получит возможность передавать пакеты на базовую станцию. Тем не менее, не зная ключа WEP, он не сможет расшифровать пакеты, приходящие от базовой станции. Таким образом, учитывая применение на базовой станции системы авторизации BotikKey⁴, отправленные чужим клиентом пакеты будут задержаны базовой станцией.

Однако знание ключа WEP базовой станции не требуется для передачи пакетов другому клиенту той же радиосети. При такой передаче базовая станция выполняет функцию моста: полученный пакет тут же передается через тот же радиоинтерфейс клиенту-получателю. Таким образом, использование Open system authentication дает возможность двум или более лицам, имеющим совместимое с базовой станцией оборудование, не являющимся зарегистрированными клиентами, использовать базовую станцию для передачи данных между собой (тем самым используя ресурс сети — пропускную способность радиосети).

Для предотвращения этого злоупотребления на базовой станции можно запретить выполнение функции моста в случае, если базовая станция не может расшифровать пакет (требуется доработка драйвера `madwifi`). Но даже при этом ограничении чужой клиент-отправитель может использовать уязвимость WEP № 3, и пакет будет воспри-

⁴<http://www.botik.ru/~botik/tools>

нят и передан базовой станцией. В случае, если получатель — тоже чужой клиент, он не сможет расшифровать пакет, поскольку не имеет ключа. Но если получатель — свой клиент, пакет будет успешно расшифрован и доставлен. Таким образом, незарегистрированный клиент сможет посылать пакеты зарегистрированным клиентам радиосети. Можно представить некоторые практические применения этой уязвимости (например, popur spam, UDP порт 1026). Тем не менее, поскольку уязвимость не дает возможности двусторонней передачи, ее полезность для атакующего ограничена, и можно с ней мириться до внедрения WPA.

3.4.4 Выводы

По результатам выполненного анализа в рамках данных работ нами запланированы следующие мероприятия для обеспечения информационной безопасности беспроводных сетей:

1. До освоения WPA/WPA2, для обеспечения безопасности используется WEP-128 с шестнадцатеричными ключами.
2. Аутентификация выполняется в режиме Open System authentication.
3. Все устанавливаемое оборудование должно быть протестировано на уязвимость к атакам методом FMS.
4. На маршрутизаторе, выполняющем функции базовой станции беспроводного доступа, необходимо ведение журнала активности клиентов для раннего обнаружения попыток несанкционированного доступа.

3.5 Программное обеспечение. Установка и конфигурирование

Конфигурация D-Link DWL-2000AP+ для работы в абонентском комплекте производится через HTTP-интерфейс. Через меню Tools/Firmware производится загрузка микрокода v115b13_nml.bin⁵.

На рис. 3.1–3.7 дано описание конфигурационных параметров для настройки абонентского комплекта для работы D-Link DWL-2000AP+.

⁵Доступен на [ftp://ftp.dlink.ru/pub/Wireless/DWL-200AP+](http://ftp.dlink.ru/pub/Wireless/DWL-200AP+)

```
AP Name: DWL-2000AP+ (не используется в режиме “клиент”)
SSID: koeonmi (должен совпадать с установкой на базовой станции)
Channel: 10 (в соответствии с разрешением ГКРЧ на
           использование радиочастоты)
Authentication: OpenSystem
Passphrase: не используется в режиме Open System
WEP: enabled
WEP Encryption: 128 bit
WEP Mode: HEX
WEP Key1: c9b748284a995973ddc1820084c745f7
WEP Key2: 764cbfe322d9890ee8f943b6b6b793e6 x
WEP Key3: b722fcf95382a3705cbbdf4df3c3dac2
WEP Key4: eadb2e1f2f6e66e486d9cbcd852c23f4
```

Рис. 3.1: Конфигурация D-Link DWL-2000AP+: установки в меню Home/Wireless

```
IP Address: 192.168.132.3
Subnet Mask: 255.255.255.0
Gateway: 192.168.132.1     IP-адрес базовой станции
```

Рис. 3.2: Конфигурация D-Link DWL-2000AP+: установки в меню Home/Lan

На рис. 3.1 ключи WEP — случайные строки, одинаковые на всех абонентских комплектах и на базовой станции. Ключ 1 используется для передачи от базовой станции, ключ 2 для передачи абонентскими комплектами, ключи 3, 4 зарезервированы.

DHCP disabled

Рис. 3.3: Конфигурация D-Link DWL-2000AP+: установки в меню Home/DHCP

AP Mode: Wireless Client

Remote AP Mac: 000d888be821 (00:0d:88:8b:e8:21 – MAC-адрес базовой станции)

Рис. 3.4: Конфигурация D-Link DWL-2000AP+: установки в меню Advanced/Mode

Disabled MAC filters (не используется в режиме “клиент”)

Рис. 3.5: Конфигурация D-Link DWL-2000AP+: установки в меню Advanced/Filters

Administrator Settings: пароль для доступа к WEB-интерфейсу устройства. Обеспечивает защиту парольной информации от удаленного доступа к устройству по порт tcp 80, и в случае несанкционированного физического доступа к устройству (в частности, с случае кражи).

SNMP: disabled

Рис. 3.6: Конфигурация D-Link DWL-2000AP+: установки в меню Tools/Admin

Beacon Interval: 100 (не используется в режиме ‘‘клиент’’)

RTS threshold: 300 (использование протокола RTS/CTS необходимо в абонентских комплектах, чтобы снизить вероятность коллизии при одновременной передаче несколькими клиентами, не слышащими друг друга -- hidden node problem)

Fragmentation Threshold: 4096 (фрагментация не используется, максимальный размер пакета 1500 байт)

DTIM interval: 3 (не используется в режиме ‘‘клиент’’)

TX Rates: Auto (автоматический выбор скорости из множества скоростей, согласованных с базовой станцией в момент ассоциации)

Wireless Mode: mixed mode (в сети могут присутствовать устройства как 802.11g, так и 802.11b)

Preamble Type: short preamble (все используемое оборудование поддерживает режим short preamble)

SSID broadcast: enabled (не используется в режиме ‘‘клиент’’)

Antenna transmit power: lowest (устанавливается исходя из уровня приема сигнала от данного клиента, измеренного на базовой станции. Как правило, lowest, если в абонентском комплекте используется направленная антенна 24 dBi)

Client Security: enabled (недокументированная установка, возможно как-то связанная с не до конца исследованной поддержкой WPA)

Speed Enhanced Mode: disabled (нестандартный ускоренный режим, предположительно использующий полосу частот двойной ширины)

Рис. 3.7: Конфигурация D-Link DWL-2000AP+: установки в меню Advanced/Performance

Глава 4

Программное обеспечение сервера аренды www-пространства

4.1 Общее описание

Естественное желание (часто — потребность) абонентов гражданской сети создавать свои собственные информационные ресурсы сдерживается сложностями, связанными с созданием собственного web-сервера — а точнее сказать, серверного комплекса аппаратных и программных средств. Вот неполный список таких сложностей и трудностей:

- необходимость содержать качественно настроенную аппаратную часть сервера, постоянно включенную в Интернет (24 часа, 7 дней в неделю), что, как правило, подразумевает:
 - серьезный комплекс серверного и вспомогательного оборудования (постоянное подключение к Интернет с реальным IP-адресом, средства гарантированного электропитания серверного комплекса, средства резервного копирования содержимого сервера и т.п.)
 - наличие высокооплачиваемых электронщиков, способных создавать, обслуживать и модернизировать аппаратную часть описанного выше серверного комплекса (вообще говоря, в круглосуточном режиме его эксплуатации)
- необходимость содержать качественно настроенную программную часть сервера, что включает в себя:
 - необходимость организации на сервере хорошо настроенного и постоянно обновляемого набора программных средств, реализующего функции (как минимум): собственно web-сервера, поддержки баз данных, средства создания

- динамического контента (CGI, PHP, Perl), средства поддержки информационной безопасности, средства мониторинга работоспособности сервера и его служб и оповещения персонала о проблемах, программные средства регламентных работ (например, резервного копирования содержимого сервера)
- наличие высокооплачиваемых системных программистов и системных администраторов, способного обслуживать и модернизировать аппаратную часть серверного комплекса (в общем-то, в круглосуточном режиме...)

Не каждое предприятие или учреждение российской глубинке способно решить указанные выше проблемы. Данные обстоятельства нами учитывались еще в самом начале создания СТ “Ботик”.

Уже в 1994 году (в те времена даже и термина Web-hosting не существовало) нами для Абонентов СТ “Ботик” была введена услуга “Аренда WWW/FTP-пространства”.

Данная услуга подразумевает следующие возможности:

1. размещение html, изображений, текстов и других файлов;
2. возможность использования CGI ограничена счетчиком(wwwcount), использование своих CGI скриптов запрещено;
3. возможность использовать ограниченный SSI(Server Side Includes).

Даже столь примитивная услуга позволила Абонентам легко и комфортно создавать на сервере www.botik.ru свои собственные статические web-ресурсы.

Все, что было сделано, так это созданы примитивные, но сравнительно комфортные условия для создателей web-наполнения. Как результат мы получили интенсивный рост наполнения на сервере www.botik.ru. Этот инициативный (и не финансируемый извне) труд сотен web-мастеров города Переславля-Залесского привел к сегодняшним результатам: web-контент, созданный в Переславле широко известен и востребован в Рунете.

Последнее утверждение можно проиллюстрировать результатами проекта “Яндекс цитирования”¹, цель которого, оценка “полезности” и “популярности” для Рунета того или иного ресурса. Данная оценка проводится здесь не за счет анализа “числа посещений” ресурса (посещения могут быть и случайными), а используя проверенный в науке метод подсчета индекса цитирования.

Совокупный индекс цитирования web-ресурсов сервера www.botik.ru² сегодня (в ноябре 2004 года) составляет 2 600, что обеспечивает серверу www.botik.ru 20 место

¹<http://www.yandex.ru/info/ci.html>

²<http://search.yaca.yandex.ru/yca/cy/ch/www.botik.ru/>

в списке так называемых “Универсальных ресурсов России”. Всего в данном списке сегодня перечислено около 1,500 сайтов, а места с первого по двадцатое занимают:

1. www.km.ru — информационный мультипортал “Кирилл и Мефодий” — индекс цитирования 18 000.
2. www.rol.ru — Россия-он-лайн, информационно-развлекательный портал — индекс 12 000.
3. www.kulichki.com — “Чёртовы Кулички” - информационный портал — индекс 11 000.
4. www.perm.ru — Пермский региональный сервер — индекс 6 200.
5. www.livejournal.com — он-лайн дневники — индекс 5 500.
6. www.irk.ru — “Твой Иркутск” - городской портал — индекс 5 300.
7. www.kaluga.ru — Калужский региональный сервер — индекс 5 000.
8. www.samara.ru — “Samara.Ru” - самарский информационный портал — индекс 4 900.
9. www.uaportal.com — Украинский портал — индекс 4 100.
10. www.novosibirsk.ru — Новосибирск, справочник города — индекс 3 900.
11. www.tyt.by — Белорусский портал и хостинг — индекс 3 600.
12. www.citycat.ru — Информационно-познавательный сервер “Городской кот” — индекс 3 500.
13. www.tomsk.net — информационный сервер “Томск on-line” — индекс 3 300.
14. www.webboard.ru — система бесплатных досок объявлений “Webforum.ru” — индекс 3 200.
15. www.nn.ru — информационный сервер города “Нижний Новгород он-лайн” — индекс 2 800.
16. www.udm.ru — региональный портал “Удмуртия” — индекс 2 700.
17. www.vl.ru — справочная информация “Виртуальный Владивосток” — индекс 2 700.
18. www.primorye.ru — региональный портал “Приморье-он-лайн” — индекс 2 600.

19. www.kazan.ru — портал “Город Казань и его обитатели” — индекс 2 600.

20. www.botik.ru — телекоммуникационный проект “www.botik.ru” — индекс 2 600.

Сегодня уровень услуги “Аренда WWW/FTP-пространства” не соответствуют современных технологий построения Web-наполнения: статический контент уступает место динамическому, который не возможно развивать в рамках услуги “Аренда WWW/FTP-пространства”.

Тем самым, в рамках проекта создания типовых аппаратных и технических средств для гражданских (сельских и городских) сетей России очень важным является разработка и реализация программного обеспечения современного сервера аренды Web-пространства (Web-hosting сервера).

Этот вопрос и обсуждается в данной документации.

Современный сервер аренды должен предоставлять возможности размещения динамических web-ресурсов с возможностью использования баз данных, ведением статистики по трафику, приходящемуся на каждый из отдельный web-ресурс, обеспечивать достаточный уровень безопасности.

Основой для сервера аренды www-пространства является компьютер класса IBM PC. Программное обеспечение сервера аренды построено на базе операционной системы Debian GNU/Linux, версия ядра 2.4.x и использует следующие пакеты:

- Web-сервер Apache.
- Сервер баз данных Mysql.
- SSH.
- GrSecurity.
- Система сбора данных по трафику пользователей — sacct-collect.

Пакеты 1-4 входят в поставку ОС Debian/GNU Linux, пакет sacct-collect является собственной разработкой. Ядро операционной системы модифицировано.

4.2 Руководство системного программиста

4.2.1 Описание основных компонент

Основные модули системы (сервера аренды):

- Apache

Основной задачей является обеспечение функции Web-сервера. Используется версия из поставки ОС Debian GNU/Linux(кодовое название sarge).

- Mysql

Основной задачей является обеспечение функции сервера баз данных. Используется версия 4.0.18-5 из дистрибутива Debian GNU/Linux(кодовое название sarge).

- sacct-collect

Данный пакет рассчитан на использование совместно с модифицированными ядрами ОС Linux 2.4.2x (смотри процесс установки). Основной задачей данного пакета является обеспечение функции получения данных о трафике пользователей системы.

- GrSecurity

GrSecurity — пакет изменений (patch) для ядра ОС Linux, повышающий безопасность использования ОС Linux. Сюда входят устойчивая система ACL, некоторые возможности аудита системы и меры, чтобы остановить некоторые общие методы, использующиеся в exploit'ах:

1. дополнительные проверки для chroot();
2. дополнительные проверки для fork();
3. различные ограничения для сети;
4. различные randomизации (pid);
5. защита страниц памяти для некоторых архитектур.

Основной задачей является повышение уровня безопасности сервера. При должной конфигурации позволяет обойти ряд проблем с использованием популярных эксплойтов. Установка рекомендуется, но не обязательна.

4.2.2 Установка

Процедура установки программного обеспечения на сервер аренды:

1. Установка ОС Debian GNU/Linux.

Подробно об установке и настройке ОС Debian GNU/Linux можно прочитать по следующему адресу <http://www.debian.org.ru/>.

2. Установка SSH.

В режиме командной строки выполнить следующие команды:

```
su root  
apt-get install ssh
```

Подробно о настройке SSH следует читать в документации по этому пакету на сайте <http://www.openssh.org/ru/index.html>.

3. Установка Apache и PHP.

В режиме командной строки выполнить следующие команды:

```
su root  
apt-get install apache php4-cgi
```

Подробно о настройке Apache следует читать в документации по этому пакету на сайте <http://httpd.apache.org/docs-project/>.

Для безопасности рекомендуется использовать suexec. PHP4 в настройках Apache рекомендуется настроить на использование как CGI скрипт.

4. Установка MySQL В режиме командной строки выполнить следующие команды:

```
su root  
apt-get install mysql-server
```

Подробно о настройке MySQL следует читать в документации по этому пакету на сайте <http://dev.mysql.com/doc/>.

5. Установка sacct-collect и GrSecurity.

Установка данных пакетов требует накладывания пакетов изменений (patches) для ядра ОС Linux, поэтому оптимальным вариантом установки этих пакетов будет следующая процедура:

- (a) Накладывание пакетов изменений (patches) GrSecurity на ядро ОС Linux.

Пакет изменений (patch) для ядра и основные утилиты берутся с сайта <http://www.grsecurity.net/download.php>.

Подробнее об установке и настройке системы следует читать в документации GrSecurity и документации по установке ядра ОС Linux:

- <http://www.grsecurity.net/papers.php>;
- <http://www.digitalhermit.com/linux/Kernel-Build-HOWTO.html>.

Конфигурация возможна как во время конфигурации ядра операционной системы, так и во время ее работы через proc или sysctl.

Пример конфигурации через sysctl на рисунке 4.1.

- (b) Накладывание пакетов изменений (patches) sacct-collect на ядро ОС Linux.

Процедура установки:

- i. Взять исходники с cvs. В режиме командной строки:

```
cd /usr/src
cvs -z3 -d :pserver:anoncvs@revenge.botik.ru:/cvsroot login
cvs -z3 -d :pserver:anoncvs@revenge.botik.ru:/cvsroot co nadmin
```

- ii. Наложить пакет изменений (patch) на ядро ОС Linux. Пакеты изменений для ядра лежат в директории

```
nadmin/project/hosted-utils/sacct-collect/kernel-patches.
```

В режиме командной строки:

```
cd linux
patch -p0 < path_to_patch
```

- iii. Сконфигурировать ядро.

- Запускаем процедуру конфигурации ядра:

```
make menuconfig
```

- В меню “General Setup” включаем опцию “Kernel support for logging any data”.
- В меню “Networking options” включаем опцию “Botik socket accounting”.
- Выходим в главное меню и сохраняем результаты.

- iv. Скомпилировать и установить deb-пакет с ядром.

```
fakeroot make-kpkg kernel-image
su root
dpkg -i ./kernel-image-yourversion.deb
```

- v. Сконфигурировать boot-manager для загрузки установленного ядра. Данная операция показана на примере конфигурирования пакета lilo:

```
edit /etc/lilo.conf
lilo
```

Подробнее о конфигурации lilo можно узнать прочитав man lilo.conf:

```
man lilo.conf
```

- (c) Собрать и установить deb-пакет с sacct-collect, загрузить установленное ядро.

Исходники находятся в nadmin/project/hosted-utils/sacct-collect/:

```
cd /usr/src/nadmin/project/hosted-utils/sacct-collect/
fakeroot debian/rules binary
su root
dpkg -i ../sacct-collect_0.2-4_i386.deb
shutdown -r now
```

4.2.3 Эксплуатация сервера аренды

Эксплуатация сервера аренды ничем не отличается от эксплуатации обычного linux-сервера. Во время эксплуатации системному администратору следует опираться на соответствующую документацию используемых нами компонент.

4.3 Руководство пользователя

Пользователь сервера аренды является непrivилегированным пользователем и работает в обычной стандартной среде ОС Linux с добавочным пакетом повышенной безопасности GrSecurity посредством терминального доступа через протокол SSH. В связи с этим существуют некоторые несущественные отличия:

- Вывод команды `ps` отличается от стандарта тем что показываются процесс только текущего пользователя.
- Команда `dmesg` сообщает о невозможности своего выполнения ввиду ограничения прав.
- Запрещен доступ к некоторой информации в `/proc`.

В качестве документации следует использовать обычные руководства пользователя по ОС Linux: <http://www.tldp.org/LDP/intro-linux/html/index.html>.

В оставшейся части раздела мы только отмечаем некоторые дополнительные аспекты связанные с использованием сервера аренды.

4.3.1 Доступ

Абоненту предоставляется доступ к серверу посредством протокола SSH. Никаких других форм доступа не предусмотрено. Для закачки файлов на сервер используется scp (secure copy, работает через SSH). Клиентские программы для реализации SSH и SCP есть как для linux/un*x, так и для MS Windows (WinSCP):

<http://winscp.sourceforge.net/eng/> — реализация SCP для Windows.

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html> — Putty, достаточно функциональный терминал под MS Windows, поддерживает SSH.

4.3.2 Использование Mysql

По запросу абоненту предоставляется возможность использования сервера баз данных Mysql. Доступ осуществляется через стандартную программу mysql.

4.3.3 Использование Apache

Web-сервер Apache при заведении нового пользователя настраивается так, что все файлы, вкладываемые для доступа в WWW, должны располагаться в каталоге

/home/user/www.

Пользователь может использовать интерфейс CGI для создания динамического контента своих сайтов. При создании CGI скриптов пользователь может использовать различные языки программирования (C/C++, Perl, PHP).

```
# grsec

kernel.grsecurity.dmesg = 1
kernel.grsecurity.rand_ip_ids = 1
kernel.grsecurity.rand_isns = 1
kernel.grsecurity.rand_pids = 1
kernel.grsecurity.rand_rpc = 1
kernel.grsecurity.rand_tcp_src_ports = 1
kernel.grsecurity.signal_logging = 1
kernel.grsecurity.timechange_logging = 1
kernel.grsecurity.linking_restrictions = 1
kernel.grsecurity.forkfail_logging = 1
kernel.grsecurity.fifo_restrictions = 1
kernel.grsecurity.execve_limiting = 1
kernel.grsecurity.exec_logging = 1
kernel.grsecurity.chroot_restrict_nice = 1
kernel.grsecurity.chroot_findtask = 1
kernel.grsecurity.chroot_execlog = 1
kernel.grsecurity.chroot_enforce_chdir = 1
kernel.grsecurity.chroot_deny_unix = 1
kernel.grsecurity.chroot_deny_sysctl = 1
kernel.grsecurity.chroot_deny_shmat = 1
kernel.grsecurity.chroot_deny_pivot = 1
kernel.grsecurity.chroot_deny_mount = 1
kernel.grsecurity.chroot_deny_mknod = 1
kernel.grsecurity.chroot_deny_fchdir = 1
kernel.grsecurity.chroot_deny_chroot = 1
kernel.grsecurity.chroot_deny_chmod = 1
kernel.grsecurity.chroot_caps = 1
kernel.grsecurity.audit_mount = 1
kernel.grsecurity.audit_ipc = 1
kernel.grsecurity.audit_chdir = 1
kernel.grsecurity.audit_group = 1
kernel.grsecurity.audit_gid = 108

# lock grsec, now disabled, for testing only
#kernel.grsecurity.grsec_lock = 1
```

Рис. 4.1: Пример конфигурации для GrSecurity через sysctl

Глава 5

Система администрирования Nadmin для городских и сельских гражданских сетей

5.1 Описание программы

5.1.1 Общее описание

5.1.1.1 Предпосылки создания новой версии системы Nadmin

В течение предыдущих шести лет администрирование городской сети города Переславля-Залесского и расчеты с абонентами обеспечивались с помощью системы Nadmin. Система была реализована в 1997 году. За время своей работы система зарекомендовала себя надежной и удобной в эксплуатации. Однако за эти годы произошел существенный прогресс в программных и аппаратных средствах, и это надо было учесть в системе Nadmin. Так же был накоплен ряд пожеланий к системе Nadmin, как со стороны абонентов, так и со стороны обслуживающего персонала. Внесение изменений в старую версию системы Nadmin для учета новых потребностей было тяжелым и громоздким решением, поэтому возникла необходимость разработки следующей версии системы.

В России региональные сети не могут рассчитывать на постоянное внешнее финансирование — следовательно, они должны самостоятельно покрывать свои эксплуатационные расходы. Источником финансирования, на который можно рассчитывать, являются средства абонентов сети (оплата услуг связи). Чтобы их можно было эффективно использовать, необходимо решить две задачи:

- выработать политику цен на услуги;

- создать систему поддержки расчетов с абонентами (billing system): расчета стоимости услуг, генерации счетов, отчетов, печати договоров и бухгалтерских документов.
- создать эффективное взаимодействие системы, обслуживающего персонала и абонентов.

В той или иной мере эти задачи были решены в старой системе Nadmin. Однако в ней отсутствовал ряд полезных возможностей. Например, сегодня очень простые запросы, которые абонент мог бы исполнить сам, выглядят как электронные письма с просьбой на адрес группы технической поддержки. Это отвлекает обслуживающий персонал от более важных дел и вынуждает его заниматься рутинными задачами, хотя с этой работой мог бы справиться сам абонент.

Тем самым стало актуальным создание новой административной системы, которая обладала бы более полной и удобной функциональностью для решения задач администрирования.

Таким образом, новая версия административной системы реализуется, как для выполнения следующих административных функций:

- автоматизация деятельности администратора сети;
- учет трафика в реальном времени (real-time) каждого абонента;
- предоставление абонентам возможности управления собственными услугами;
- поддержка новых услуг, таких как: IP-телефония, web-хостинг;
- возможность создания новых тарифных планов;

так и для удовлетворения следующих экономических потребностей:

- необходимость поддержания конкурентной способности СТ “Ботик” на рынке высоких технологий;
- повышение привлекательности СТ “Ботик” среди абонентов — этот аспект в бизнесе является одним из основных источников успеха и динамического развития;
- перенос рутинных операций с персонала СТ “Ботик” на абонентов, что позволит не увеличивать персонал СТ “Ботик” и тем самым снизит эксплуатационные расходы;
- снижение расходов абонентов СТ “Ботик” (как за счет снижения эксплуатационных расходов, так и за счет предоставления возможностей управления своими расходами).

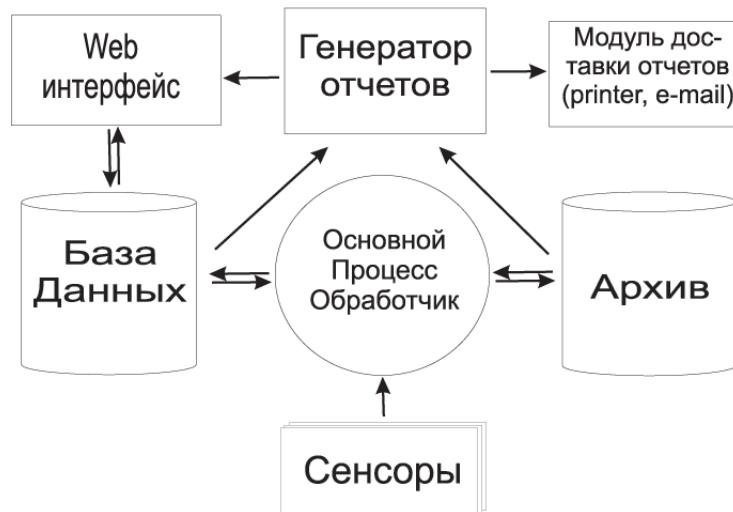


Рис. 5.1: Архитектура системы Nadmin

Также в новой версии административной системы Nadmin особое внимание было уделено дизайну и гибкости системы, и возможности дальнейшего расширения без кардинальной перестройки всей системы.

Следовательно, основной целью данной работы было создание новой административной системы Nadmin. Для этого нужно было решить следующие задачи:

- исследовать и реализовать наиболее эффективную и удобную базу данных;
- исследовать и реализовать наиболее эффективные статистическую, расчетную и административную подсистемы;
- исследовать и реализовать web-интерфейсы (пользовательский и администраторский).

В последующих разделах будут рассмотрены решения поставленных задач.

5.1.2 Архитектура системы

Система Nadmin реализуется на языке программирования Perl, так как он представляет собой мощный и удобный механизм для работы со многими программами и системами в ОС Linux (Debian), такими как: grep, tcpdump, rcs, chmod, ssh и т. д., а также для работы с сетевыми ресурсами.

Система Nadmin состоит из подсистем, которые перечислены ниже.

Сенсоры. Понятие сенсоров было введено для того, чтобы можно было легко добавлять в систему поддержку новых услуг. Сенсоры срабатывают от событий вида “оказана услуга *Service* некоторому потребителю с идентификатором *Id*” и формирует *порции статистики* для обработки *основным процессом-обработчиком* (подробнее см. раздел 5.1.3).

База данных (БД). БД используется для хранения разнородных объектов типа: абоненты, подключения, лицевые счета и т. д. В системе Nadmin реализована собственная БД с использованием пакета RCS, позволяющим хранить любые изменения, которым подвергался тот или иной объект базы данных за всё время своего существования (подробнее см. раздел 5.1.4).

Архив. Архив используется для долгосрочного хранения информации. В нем поддержана возможность сжатия данных, что позволяет хранить большие объёмы информации. Кроме того, в нем реализована возможность удобной выборки заархивированной информации (подробнее см. раздел 5.1.5).

Основной процесс-обработчик (ОПО). ОПО является центральным процессом системы, который выполняет сбор статистики от сенсоров, расчет стоимости оказанной услуги, модификацию лицевых счетов абонентов и архивацию полученной статистики в архив (подробнее см. раздел 5.1.6).

Генератор отчетов. Генератор отчетов по запросу, используя информацию из базы данных и из архива, строит отчеты следующих видов:

- справка о состоянии лицевого счета абонента;
- перечень услуг, предоставляемых абоненту;
- технические сведения о той или иной услуге;
- общие сведения о состоянии системы;
- и др.

Сгенерированные отчеты бывают трёх видов: web-страница, печатная форма, отчет для отправки по e-mail (подробнее см. раздел 5.2).

Web-интерфейс. Web-интерфейс является средством взаимодействия системы с администратором и пользователем, обеспечивающим следующее:

- ввод и изменение информации в базе данных;
- выборку информации из базы данных;

- управление составом и режимами предоставляемых пользователям услуг;
- передачу системе команд на выполнение определенных действий, а именно: печать договоров, рассылка писем клиентам, выполнение тестирования указанных частей сети;
- доступ к архиву статистики и результатам обработки статистики (отчетам).

Процесс обновления системной конфигурации. Процесс обновления системной конфигурации (подробнее см. раздел 5.1.7) занимается синхронизацией базы данных с информацией, хранящейся в системных сервисах, таких как:

- named (DNS сервер);
- FTP-пространство;
- UUCP (сервер почтовой службы);
- BotikKey;
- и др.

Eventlog. Механизм Eventlog является неотъемлемым связующим звеном всей системы. С помощью него разнородные подсистемы могут обмениваться сообщениями друг с другом (подробнее см. раздел 5.1.9).

5.1.3 Сенсоры

Для реализации статистической подсистемы в структуру системы Nadmin введено понятие сенсоров. Сенсор является обработчиком сырой¹ статистики. Сенсор берет сырую статистику от конкретного источника, обрабатывает ее и отдает ОПО. Обычно естественным источником статистической информации по использованию ресурсов пользователями через различные серверы системы являются log-файлы этих серверов. Например, proxy-сервер Squid формирует свой стандартный log-файл, обычно расположенный по адресу /var/log/squid/access.log, а соответствующий сенсор берет из него данные, которые являются для него сырой статистикой (см. рис. 5.2).

Таким образом, каждый сенсор умеет разбираться в конкретном формате сырой статистики, то есть он анализирует и сортирует поток сырой статистики из log-файлов.

¹Сырая статистика появляется из различных источников из “внешнего” (по отношению к Nadmin) мира, в момент, когда абоненту оказывается та или иная услуга.

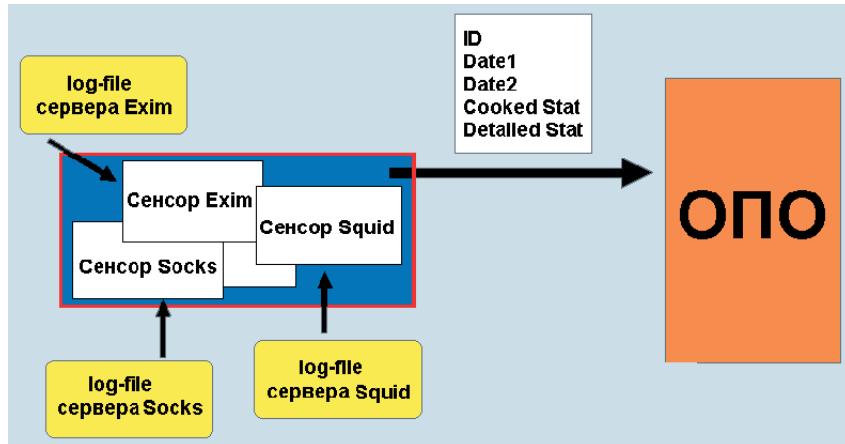


Рис. 5.2: Схема работы сенсоров с log-файлами и ОПО

Обычно в ходе одного такта работы сенсора анализируется одна (очередная) строка из log-файла. В ходе этого анализа сенсор определяет потребителя ресурса и объем потребленной информации. В конце сенсор формирует фрагмент детальной статистики для каждого потребителя (если было несколько потребителей ресурса), считает суммарный объем востребованного ресурса по каждому фрагменту детальной статистики и выносит эту информацию в заголовок фрагмента детальной статистики. Вместе заголовок и фрагмент детальной статистики образуют порцию статистики (см. рис. 5.5).

Далее порции статистики (*slice*) через поток (UNIX-pipe)² передаются для ОПО. За счет заголовка порции статистики отпадает необходимость в дальнейшем разборе детальной статистики, ОПО может относиться к детальной статистике как к данным, предназначенным только для показа пользователю.

Сенсор не обращается к базе данных Nadmin и к другим конфигурационным файлам системы, он сделан максимально автономным. Это сделано для того, чтобы UNIX-процесс, исполняющий код сенсора, требовал как можно меньше ресурсов (память, процессорное время). Кроме того, эта независимость от базы данных и других компонент системы позволяет сенсору работать на удаленном компьютере.

Сенсоры реализованы как обычные процедурные программы на языке Perl. Каждый сенсор запускается ОПО как программа-демон, постоянно находящаяся в рабочем состоянии. После получения очередной порции статистики из log-файла и ее обработки сенсор “засыпает” на определенное время, по прошествии этого времени сенсор “просыпается” и процесс повторяется. За счет постоянно открытого указателя (*handler*) на log-файл сенсор всегда “помнит”, до какого места в log-файле он уже обработал стати-

²Pipe — системный механизм, позволяющий устанавливать каналы связи между родственными процессами.

стику, таким образом, одна строка никогда не обрабатывается дважды.

Серверы, записывая информацию в log-файлы, используют небуферизированную запись³. При этом возникает опасность прочтения сенсором неполной строки (обрыв строки). Следовательно, в этом случае произойдет потеря части информации. Во избежание этого сенсор должен проверять каждую строку, полученную из файла, на ее полноту⁴. В случае, если строка из log-файла оказалась неполной, сенсор производит перевод указателя (`handler`) в log-файле на позицию, где кончается последняя обработанная строка. То есть происходит возврат к позиции в log-файле, до которой были успешно обработаны все предыдущие строки. После этого сенсор “засыпает” на некоторое время, для того, чтобы сервер успел полностью дописать строку в log-файл.

Многие источники сырой статистики — серверы с log-файлами — используют понятие ротации log-файлов. Например, в конце каждого суток закрывается текущий log-файл, переименовывается, после этого создается новый log-файл, в который заносится информация, относящаяся к следующим суткам. Таким образом, сенсор должен уметь отслеживать ротацию log-файлов: при смене log-файлов открыть указатель на новый файл. Большинство сенсоров отслеживают данное событие как смену индексного описателя файла (`inode`). В случае изменения `inode` происходит закрытие старого log-файла и открытие нового (см. рис. 5.3).

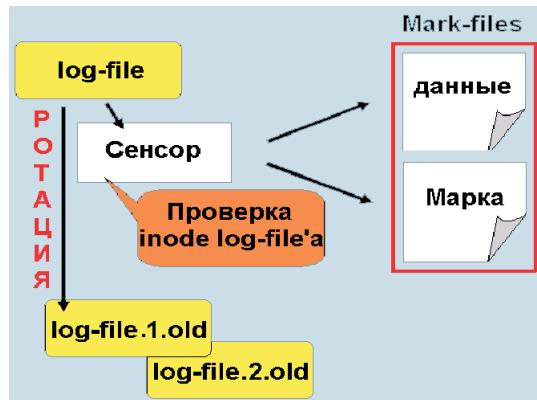


Рис. 5.3: Сенсоры: ротация log-файлов

5.1.3.1 Класс `Nadmin::Mark`

Класс предоставляет используемое всеми сенсорами средство: *закладки* (marks). Класс включает в себя методы для работы с закладкой (создание, сохранение, чтение):

³Небуферизированная запись — посимвольная запись, при которой не происходит разовой записи строки в log-файл.

⁴Признаком полноты строки является наличие символа конца строки '\n'.

- `Nadmin::Mark::new($id)` — создает закладку с идентификатором `$id`. Как правило, `$id` — это имя log-файла, обрабатываемого данным сенсором.
- `Nadmin::Mark::put_mark($str)` — сохраняет строку `$str` в качестве закладки для сенсора с идентификатором `$id`.
- `$str = Nadmin::Mark::get_mark()` — возвращает последнюю закладку, сохраненную с помощью `Nadmin::Mark::put_mark()`.

5.1.3.2 Марки

Марки (закладки) позволяют сенсору, обрабатывающему log-файл, возобновить работу после перезапуска в том месте log-файла, где он остановился (см. рис. 5.4). Содержимое закладки может быть разное для каждого конкретного сенсора. Как правило — это последняя обработанная строка из log-файла. Для сенсора абонентской платы — это дата, за которую уже произведено начисление абонентской платы.

В случае остановки и последующего запуска сенсора происходит поиск марки (закладки). Так как сенсор мог простоять в течение длительного времени, то за это время могли произойти ротации log-файлов (одна или несколько). Поэтому при запуске сенсор первоначально определяет все доступные log-файлы, в том числе и те, в которых данные уже заархивированы. После этого происходит сортировка log-файлов в порядке возрастания (от тех, которые были созданы позднее, до тех, которые были созданы раньше) и создается упорядоченный список log-файлов. После этого список последовательно в прямом порядке просматривается с целью нахождения последней обработанной строки. В случае успешного поиска сенсор начинает обрабатывать просмотренные из списка файлы в обратном порядке, начиная с наиболее старого и кончая самым новым log-файлом. В случае ненахождения марки сенсор выдает сообщение о невозможности продолжения работы и заканчивает процесс, ожидая вмешательства администратора.

Марки сенсоров сохраняются в файле, обычно совпадающем с именем сенсора (подробнее см. раздел 5.1.3.4), в заранее определенном каталоге. В случае отсутствия марки для какого-либо сенсора (пустой файл) сенсор в качестве марки берет первую строку из базового log-файла для данного сенсора.

Если необходимо задать (изменить) марку вручную, то нужно в качестве марки записать строку из базового log-файла или других log-файлов, относящихся к данному сенсору. Марка должна обязательно содержать символ конца строки (переход на новую строку). Исключением является марка для сенсора `socks` (подробнее см. раздел 5.1.3.6). Для сенсора `abplata` в качестве марки указывается дата, с которой необходимо произ-

вести расчет, в формате YYYYMMDD⁵ без символа конца строки.

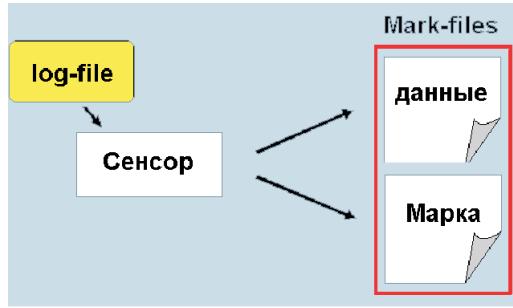


Рис. 5.4: Сенсоры: сохранение закладок

5.1.3.3 Формат порции статистики

Порция статистики, передаваемая сенсором для ОПО, имеет вид, специфичный для каждого конкретного сенсора, но общий формат включает в себя заголовок с результатами обработки детальной статистики, и далее, после пустой строки, саму детальную статистику. Заголовок содержит обязательные поля Date1 и Date2 (даты начала и конца периода, к которому относится статистика) и поле Id, по которому происходит идентификация потребителя и поиск услуги, к которой относится данная порция статистики. Пример порции статистики см. на рис. 5.5.

5.1.3.4 Параметры работы сенсоров

В качестве основных параметров сенсоров можно выделить следующие:

- период работы сенсора;
- имя файла с маркой;
- имя базового log-файла;
- путь к директории, содержащей log-файлы;

Данные параметры указываются в начале исходного текста каждой программы-сенсора в виде констант и при необходимости могут быть изменены.

⁵YYYYMMDD — формат даты, где YYYY обозначает год, MM — месяц, DD — день указанной даты (см. раздел 5.1.12).

```
Date1: 20030606160105
Date2: 20030606160110
Id: 193.232.174.115
Count-in: 19356
Class-in: HIGH
Count-out: 1346
Class-out: NORMAL

144:120 81.117.239.98.3333 x.8081 tcp
144:120 81.117.239.98.3233 x.3128 tcp
144:120 81.117.239.98.3162 x.80 tcp
144:120 81.117.239.98.3268 x.8080 tcp
144:0 81.117.239.98.2982 x.1080 tcp
```

Данная порция IP-статистики описывает следующее событие: компьютер с IP-адресом 193.232.174.115 за указанный период (2003/06/06 16:10:05-2003/06/06 16:10:10) принял 19356 байт с высоким приоритетом и отправил 1346 байт со средним приоритетом.

Рис. 5.5: Порция статистики

5.1.3.5 Сообщения об ошибках

Имя_сенсора: `bad mark has been found`. Данное сообщение может быть выдано при перезапуске сенсора. Предупреждение выдается в случае неправильного формата сохраненной марки.

Имя_сенсора: `something wrong with logfile имя_log_файла`. Это сообщение может быть выдано при запуске сенсора в случае отсутствия марки. Сенсор пытается сохранить в качестве марки первую строку из log-файла. В случае ошибки (log-файл не существует или он пустой) выдается предупреждение.

Имя_сенсора: `mark is not found anywhere`. Данное сообщение может быть выдано при запуске сенсора. Предупреждение выдается в случае, если марка относится к времени, не принадлежащему временному интервалу⁶ log-файлов. Обычно эта ошибка выдается в случае очень старой марки.

⁶Временной интервал задается датой первой строки самого раннего log-файла и датой последней строки текущего (базового) log-файла.

Имя_сенсора: mark is not found in logfiles. Данное сообщение может быть выдано при запуске сенсора. Предупреждение выдается в случае, если марка относится к времени, принадлежащему временному интервалу log-файлов, но не совпадает ни с одной строкой из log-файлов.

5.1.3.6 Особенности реализации сенсоров

Так как сенсоры могут обрабатывать сырью статистику от самых разных источников, имеющих разную природу, то каждый сенсор помимо общих черт, описанных выше, может иметь и свои особенности. В последующих разделах будут рассмотрены особенности реализации для некоторых сенсоров.

Сенсор squid. Сенсор `squid` обрабатывает сырью статистику от squid-сервера. При обработке log-файлов сенсор определяет, является ли очередной запрос внутренним (локальным, то есть запрашивается информация из домена `botik.ru`), внешним (вне домена `botik.ru`), или информация поступила пользователю из кеша сервера. Сенсор суммирует объем полученной информации для каждого типа запросов и отдает порцию статистики ОПО.

Сенсор socks. Сенсор `socks` анализирует сырью статистику от socks-сервера. Сенсор суммирует входящий и исходящий трафик и в заголовке фрагмента передает эту информацию ОПО. Особенности сенсора:

- в log-файлах socks-сервера при записи даты не указывается год. Сенсор проводит дополнительный анализ для определения полной даты. Для этого определяется системное время. После этого сенсор проводит проверку на принадлежность анализируемой строки текущему году, в случае непринадлежности текущему году строке сопоставляется предыдущий год;
- базовый log-файл socks-сервера `log` заполняется раз в сутки, и сразу после этого происходит ротация log-файлов. Как следствие сенсор `socks` не держит постоянно открытого указателя на log-файл, и в качестве базового log-файла для сенсора объявлен следующий log-файл после базового log-файла socks-сервера `log.1.bz2`.

В качестве марки записываются первая строка уже обработанного базового log-файла сенсора `log.1.bz2` и дата обработки, разделенные символом `#`. В случае, если произошел сбой во время работы сенсора, в качестве марки записываются первая строка обрабатываемого log-файла и последняя успешно обработанная строка из этого файла, разделенные символом `#`. Марка не содержит символа конца строки.

Сенсор tproxy. Сенсор `tproxy` берет сырую статистику от `transproxy`-сервера. Сенсор суммирует входящий и исходящий трафик и отдает эту информацию ОПО.

Сенсор exim. Сенсор `exim` занимается анализом почтовой статистики. Особенность работы сенсора `exim` связана со следующим:

- информация о доставке письма в log-файлах `exim`-сервера занимает не одну, а несколько строк;
- почтовый трафик часто является многоадресным.

В log-файлах информация о письме соотносится как минимум с двумя строками. Первая (в порядке поступления в log-файл) строка идентифицирует письмо как поступившее на учет. Данная строка позволяет определить отправителя. Вторая сообщает о том, что письмо было доставлено по адресу, и по ней идентифицируется получатель.

Таким образом, в некий момент времени сенсор хранит информацию обо всех поступивших на учет, но еще не доставленных письмах. Эта информация должна сохраняться на случай сбоя работы сенсора, иначе при последующем запуске сенсора некоторые письма будут не учтены. Для этого после обработки очередной порции статистики сенсор сохраняет как дополнительную закладку (`mark`) информацию о письмах, уже поступивших на учет, но еще недоставленных получателю. При запуске сенсора эта информация извлекается и восстанавливается картина, которая была на момент сбоя работы сенсора.

Также почтовый трафик является многоадресным, то есть получателями почты могут быть пользователи и списки рассылки (к которым можно отнести также перенаправления почты, организованные на сервере, в частности файлы `.forward`). Все это должно учитываться при анализе.

- **Письма, отправленные абонентами.** Каждая доставка письма засчитывается отправителю либо как местное письмо, если получатель также является абонентом региональной сети, либо как исходящее внешнее письмо, если получатель расположен за пределами региональной сети.

Доставка локального письма в список рассылки засчитывается отправителю и владельцу списка как местное письмо. Дальнейшие доставки, выполняемые в результате обработки списка, засчитываются уже не отправителю письма, а владельцу списка (как местные или исходящие внешние, в зависимости от получателя).

- **Письма из внешнего мира.** Доставка письма абоненту засчитывается ему как входящее внешнее письмо. Если письмо адресовано нескольким абонентам, то каждому из них засчитывается целое входящее внешнее письмо. При доставке в список

рассылки письмо засчитывается владельцу списка как входящее внешнее письмо. Дальнейшие доставки, выполняемые в результате обработки списка, засчитываются владельцу списка (как местные или исходящие внешние, в зависимости от получателя).

- **Определение отправителя письма.** Машина-отправитель определяется по IP-адресу, с которого письмо было передано по SMTP на почтовый сервер. Альтернативный метод определения, по адресу отправителя (*envelope address*), не используется, так как адрес отправителя может быть фальсифицирован.

Сенсор abplata. Сенсор *abplata* в отличие от большинства сенсоров обращается к базе данных Nadmin. После анализа базы данных по абонентам сенсор *abplata* выдает ОПО информацию для расчета абонентской платы. Второй особенностью сенсора является то, что порции статистики, передаваемые этим сенсором, не содержат детальной статистики.

Раз в сутки примерно в одно и тоже время сенсор анализирует объекты *Connect* и *Net* в базе данных (подробнее см. раздел 5.1.4). В ходе анализа определяется основной тип каждого подключения. Затем в соответствии с заранее заданным порядком типов (например, в системе Nadmin задается следующий порядок: *NET*, *DIALUP*, *LAN*, *MAIL*, *lan*) происходит сортировка подключений. Внутри типа сортировка подключений ведется в лексикографическом порядке. Также определяется, равно ли значение статуса подключения *off* (отключенное) или нет.

В отсортированном виде информация по подключениям абонентов передается ОПО. В зависимости от очередности поступления информации о подключении ОПО определяет статус подключения (*pri* — первичное или *sec* — вторичное).

Также сенсор *abplata* занимается анализом информации и статистики по аренде FTP/WWW-пространства. Этот анализ выполняется на основании статистики по использованию дискового пространства и категории пользователей.

Особым образом обрабатывается случай, когда организация арендует несколько каталогов и один из них (A) является подкаталогом другого (B): при расчете из объема каталога (B) необходимо вычесть объем каталога (A), чтобы не взимать плату дважды за одно и то же пространство.

В случае, если сенсор должен выдать для ОПО информацию за предыдущие дни, то используются ретроспективные возможности базы данных системы Nadmin (подробнее см. раздел 5.1.4).

Сенсор ipphone. Сенсор *ipphone* обращается к web-сайту фирмы, предоставляющей СТ “Ботик” услуги по IP-телефонии, и берет с него информацию о произведенных звон-

ках. После подсчета суммарной стоимости звонков для каждого пользователя эта информация передается ОПО.

Сенсор ip_stat. Сенсор IP-статистики производит анализ выходных файлов утилиты `tcpdump` и выдает слайсы статистики установленного для всех сенсоров системы Nadmin формата.

Подробнее о работе сенсора `ip_stat` см. в приложении А.1.

5.1.4 База данных

Основой системы является база данных (БД) по абонентам сети и предоставляемым им услугам.

5.1.4.1 Объекты базы данных в системе Nadmin

`Org` — данные об абоненте сети (организации или физическом лице, с которым заключен договор на обслуживание);

`Account` — данные о лицевом счете абонента;

`Connect` — данные об одиночном подключении (одном компьютере) абонента, получающего одну или несколько из следующих услуг: `MAIL`, `DIALUP`, `LAN`;

`Net` — данные о групповом подключении (подключения локальной сети) абонента;

`Ftp` — данные о FTP-входе с правом записи, предоставленном абоненту для доступа к арендуемому им FTP/WWW-пространству;

`IPphone` — данные об абоненте IP-телефонии;

`WWUserif` — данные о пользователе Web-интерфейса;

`Subnet` — данные о подсетях (интервалах IP-адресов в СТ “Ботик”).

5.1.4.2 Реализация базы данных системы Nadmin

База данных реализована как дерево каталогов с текстовыми плоскими файлами под управлением системы RCS⁷. За счет использования RCS автоматически реализуются два важных оригинальных свойства базы данных:

- возможность получения не только текущего состояния объекта, но и его состояния на любой момент со времени его создания (ретроспективность базы данных);

⁷RCS — система управления версиями файлов.

- хранение информации о дате и авторе каждого изменения объектов в базе данных.

Каждому корневому каталогу соответствует сущность базы данных (объекты `Org`, `Connect`, `Ftp`, `Net` и т.д.). Экземпляры сущностей представлены в виде файлов под управлением системы RCS, хранящиеся в соответствующих каталогах. Именем файла является индекс (номер экземпляра соответствующего объекта в БД) — `ID`. Идентификатор `ID` является сквозным для всех каталогов-сущностей, поэтому для сохранения значения последнего использованного значения индекса был заведён файл `last.id`.

Помимо каталогов с файлами база данных включает в себя также набор индексных файлов. Индексные файлы содержат в себе основную информацию из базы данных на текущий момент времени, необходимую для быстрой работы системы.

Структура каталогов и файлов представлена на рис. 5.6.

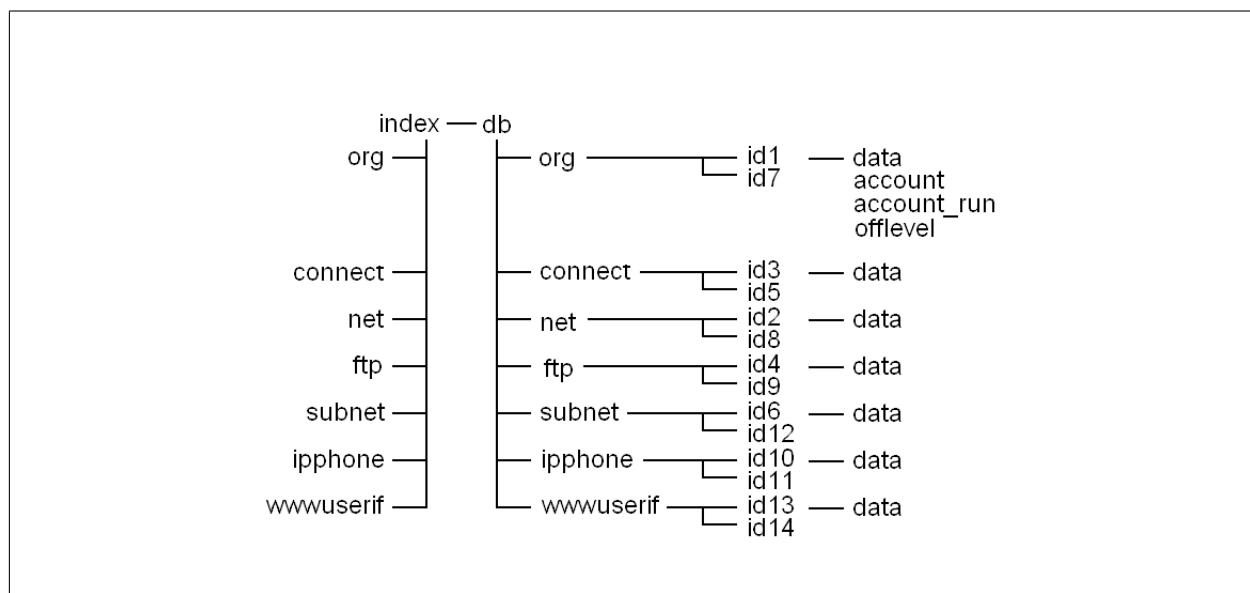


Рис. 5.6: База данных: структура каталогов и файлов

5.1.4.3 Perl-интерфейс базы данных

Perl-интерфейс базы данных системы Nadmin представляет собой слой ПО, обеспечивающий доступ к базе данных для расчетной подсистемы и Web-интерфейсов администратора и пользователя. Интерфейс построен как иерархия Perl-классов (см. рис. 5.7). Использование объектно-ориентированного подхода позволяет сделать реализацию компактной и удобной.

Класс `Nadmin::Field` представляет собой описатель одиночного поля базы данных.

Из объектов класса `Nadmin::Field` строятся таблицы описателей полей объектов БД, которые используются для реализации массовых операций с полями.

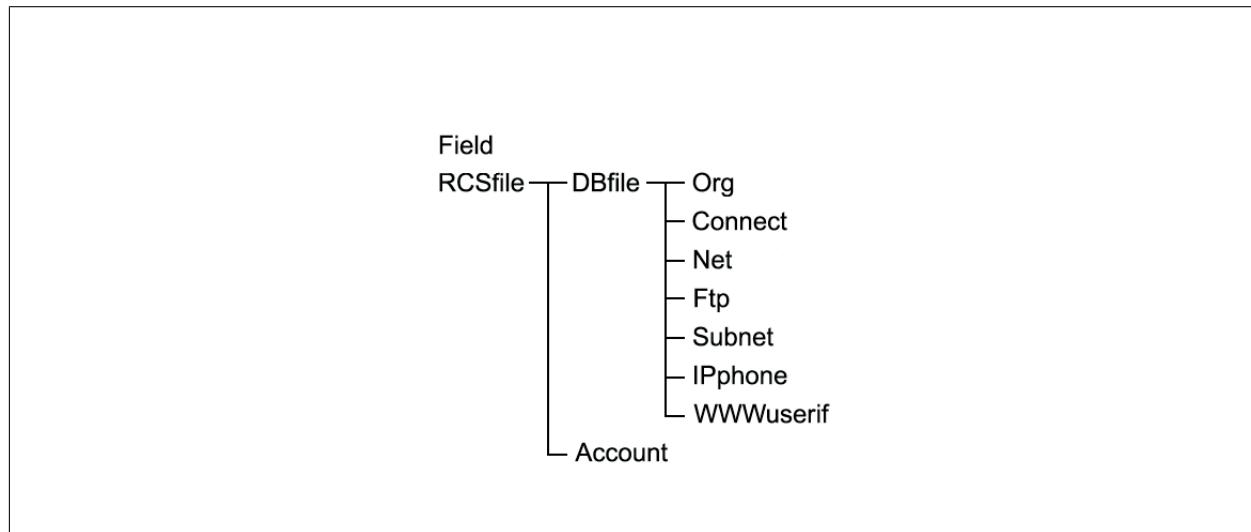


Рис. 5.7: Иерархия классов Perl-интерфейса базы данных

Класс `Nadmin::RCSfile` реализует набор базовых операций с файлами, находящимися под управлением системы контроля версий RCS.

Класс `Nadmin::DBfile` является производным от класса `Nadmin::RCSfile`. Самостоятельно не используется, только как базовый класс для конкретных объектов (`Nadmin::Org`, `Nadmin::Connect` и др.). Реализует функции, общие для этих объектов базы данных.

Классы `Nadmin::Org`, `Nadmin::Connect`, `Nadmin::Ftp` и др. являются производными классами от класса `Nadmin::DBfile`, представляющего объекты базы данных конкретных типов.

Класс `Nadmin::Account` используется для доступа к лицевым счетам абонентов. Реализует функции просмотра, анализа, модификации лицевых счетов абонентов.

Класс `Nadmin::Field`. При реализации базы данных в виде текстовых файлов необходимо отдельно определять поля объектов. Для этого используется класс `Nadmin::Field`, представляющий описатели одиночного поля базы данных. При определении поля объекта задается его имя и набор описателей. Существуют следующие описатели:

`Nadmin::Field::descr()` — описание данного поля в текстовом формате;

`Nadmin::Field::example()` — образец значения для данного поля объекта, в случае образца значения по умолчанию равен `undef`;

`Nadmin::Field::default()` — значение по умолчанию для данного поля объекта, в случае отсутствия значения по умолчанию равен `undef`;

`Nadmin::Field::invisible()` — описатель для определения скрытого поля объекта;

`Nadmin::Field::indexable()` — описатель для определения индексируемости данного поля объекта, в случае поля, не включаемого в индексный файл равен 0;

`Nadmin::Field::optional()` — описатель для необязательного поля, в случае обязательного поля равен 0;

`Nadmin::Field::id()` — описатель ключевого поля объекта, в случае, если поле не ключевое, равен 0;

`Nadmin::Field::readonly()` — описатель поля объекта, доступного только для чтения, в случае поля, доступного для изменений, равен 0;

`Nadmin::Field::condition()` — условие, задаваемое для поля объекта, в случае отсутствия условия равен 1.

Также класс `Nadmin::Field` содержит методы для работы с полями объекта:

`Nadmin::Field::form()` — создает форму для web-страницы для изменения значения данного поля объекта;

`Nadmin::Field::check()` — производит проверку значения поля объекта, в случае успешной проверки возвращает `undef`, в противном случае список ошибок;

`Nadmin::Field::untaint()` — производит лексическую проверку значения поля объекта, в случае успешной проверки возвращает `undef`, в противном случае список ошибок;

`Nadmin::Field::generate()` — производит генерацию нового значения для генерируемого поля объекта (например, поле `password`).

Класс `Nadmin::RCSfile`. Для работы с файлами, находящимися под управлением системы контроля версий RCS, необходим набор функций-методов, реализуемых в данном классе.

`Nadmin::RCSfile::new($file)` — создать новый `Nadmin::RCSfile`-объект;

`Nadmin::RCSfile::write()` — записать RCS-файл;

`Nadmin::RCSfile::read($date)` — прочитать RCS-файл на заданную дату. В случае объекта, данные о котором на заданную дату содержатся в индексном файле, ревизии не читаются, в случае отсутствия данных в индексном файле вызывается метод `Nadmin::RCSfile::really_read($date)`;

`Nadmin::RCSfile::really_read($date)` — прочитать RCS-файл на заданную дату;

`Nadmin::RCSfile::checkout()` — извлекает последнюю ревизию с блокировкой (lock) файла;

`Nadmin::RCSfile::checkin()` — сохраняет ревизию в файл;

Класс Nadmin::DBfile. Для работы с объектами базы данных системы Nadmin создан класс `Nadmin::DBfile`, в котором реализованы общие методы для объектов. Класс наследуется от класса `Nadmin::RCSfile`.

`Nadmin::DBfile::new($file, $date)` — создать новый `Nadmin::DBfile`-объект;

`Nadmin::DBfile::create()` — создать новый DBfile-файл;

`Nadmin::DBfile::index()` — создать индексный файл для объекта базы данных;

`Nadmin::DBfile::reindex()` — произвести переиндексацию;

`Nadmin::DBfile::write()` — записать DBfile-файл;

`Nadmin::DBfile::show()` — создает таблицу для web-страницы для отображения значений полей объекта;

`Nadmin::DBfile::form()` — создает форму для web-страницы для отображения или/и изменения значений полей объекта;

`Nadmin::DBfile::list(@cond)` — создает список объектов, удовлетворяющих условиям `@cond`;

`Nadmin::DBfile::checkin()` — сохраняет ревизию в файл;

Классы для реализации объектов базы данных системы Nadmin `Nadmin::Org`, `Nadmin::Connect`, `Nadmin::Forward`, `Nadmin::Net`, `Nadmin::Subnet`, `Nadmin::IPphone`, `Nadmin::WWWuserif`, `Nadmin::Ftp`, `Nadmin::Hosted`, `Nadmin::Box` реализуют объекты базы данных. Каждый класс содержит список полей объекта, методы для работы с классом. В соответствующем каждому классу модуле `Nadmin::имя_классаFields` содержится описание каждого поля объекта. Классы наследуются от класса `Nadmin::DBfile`. Классы содержат следующие методы:

`Nadmin::имя_класса::new()` — создать новый `Nadmin::имя_класса`-объект;

`Nadmin::имя_класса::create()` — создать новый `имя_класса`-файл;

`Nadmin::имя_класса::service_name()` — данный метод возвращает имя объекта из класса `Nadmin::имя_класса`;

`Nadmin::имя_класса::ур_ids()` — данный метод добавляет в справочник данные о `Nadmin::имя_класса`-объекте;

5.1.4.4 Лицевой счет абонента и класс `Nadmin::Account`

Лицевой счет абонента представлен объектом `Nadmin::Account` в базе данных системы `Nadmin`. Для каждого абонента имеется как минимум один набор из двух файлов:

- `account` — содержит записи платежей абонентов и его суммарных расходов за расчетные периоды⁸,
- `account_run` — содержит детальную статистику расходов абонента за текущий расчетный период.

Файл `account_run` в базе данных содержит расходы только за текущий расчетный период. Детальные расходы за прошлые расчетные периоды хранятся в архиве.

В системе `Nadmin` поддержана возможность наличия нескольких счетов у одного абонента. Такая ситуация решается введением нумерации лицевых счетов и соответствующих им файлов: `account`, `account_run`, `account2`, `account2_run` и т. д.

Объект `Nadmin::Account` описан в модуле `Nadmin::Account.pm`.

Все действия с лицевым счетом абонента делятся на несколько групп:

- занесение текущих расходов в run-файл лицевого счета;
- занесение на лицевой счет расходов за расчетный период и платежей;
- работа с уровнем отключения абонента;
- просмотр состояния лицевого счета.

Подробнее о функциональности и методах класса `Nadmin::Account` см. в приложении А.2.

⁸Расчетный период в СТ “Ботик” составляет 15 дней.

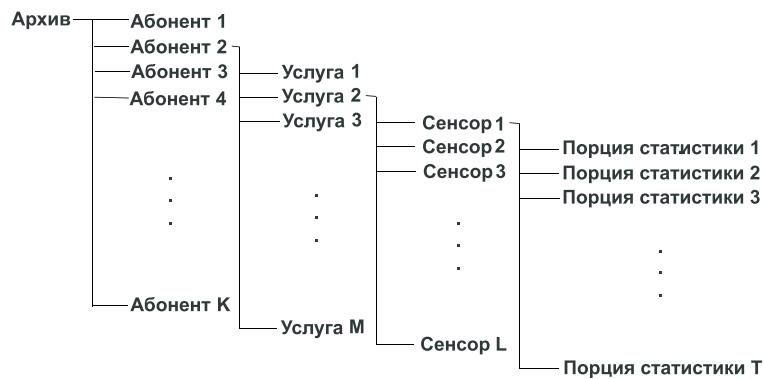


Рис. 5.8: Структура архива порций статистики

5.1.5 Архивация данных

Для аргументированного решения вопросов абонентов относительно происхождения трафика необходимо хранение статистической информации за весь расчетный период с достаточной степенью детализации. При возникновении претензий администратор должен иметь возможность предъявить абоненту данные, свидетельствующие о потреблении ресурса, со временем и указанием типа сервиса.

При работе системы в течение дня может накапливаться очень большой объем статистической информации. Поэтому необходимо иметь средство, позволяющее архивировать эту информацию с уменьшением объема. Система Nadmin включает в себя такое средство — архив системы — позволяющее компактно хранить в себе статистическую информацию.

5.1.5.1 Структура архива

Архив представляет собой дерево каталогов с файлами, содержащими в себе порции статистики, полученные от сенсоров (см. рис. 5.8). На первом уровне архива это список абонентов. В каталог с названием абонента записываются все предоставляемые ему услуги, например подключения, IP-телефонные входы и т. д. В каталогах с названиями услуг записываются названия сенсоров, от которых могут поступать порции статистики. В каталоги с названиями сенсоров складываются порции статистики.

5.1.5.2 Архивация данных

Архивация данных производится при помощи модуля `Nadmin::Archive`. Этот класс реализует следующий набор функций:

`Nadmin::Archive::new($path)` — создать новый `Nadmin::Archive`-объект, здесь пере-

менная `$path` — путь до архива;

`Nadmin::Archive::put($date1, $date2, @data)` — сохранить данные `@data` в файл; название файла имеет вид: `date1-date2`, где указанные даты означают, что в данном файле содержится информация за период времени с `date1` по `date2`;

`@data = Nadmin::Archive::get($date1, $date2)` — получить данные `@data` из архива за интервал `date1-date2`;

`Nadmin::Archive::compact()` — сжать данные в `tar.bz2`-архив, где название файла имеет вид: `date1-date2.tar.bz2`;

`Nadmin::Archive::cleanup($period)` — удаление файлов из архива за указанный период `$period`.

5.1.5.3 Сообщения об ошибках

`transposed date interval: [date1, date2]`. Данное сообщение может появиться при сохранении данных в архив за интервал `date1-date2` в случае `date1 > date2`.

`ERROR SLICE: slice is too old [slice date: date1] < [archive date: date2]`.

Данное сообщение может быть выдано при сохранении данных в архив в случае, если дата в полученном слайсе `date1` меньше, чем дата последнего успешно сохраненного слайса в архиве (`date2`).

`Invalid period spec.` Данное сообщение может быть выдано при удалении файлов из архива в случае неправильно заданного периода.

5.1.6 Расчетная подсистема

В данном разделе рассматривается реализация расчетной части системы Nadmin. Она включает в себя:

1. основной процесс-обработчик (ОПО);
2. механизм описания биллинговых функций;
3. механизм расчета стоимости мегабайта;
4. программу перерасчета статистики (в случае сбоя или ошибки).

5.1.6.1 Основной процесс-обработчик

Архитектура ОПО.

Основной процесс-обработчик является ядром расчетной подсистемы. В его обязанности входит получение порций статистики от сенсоров и запуск соответствующих расчетных (билинговых) функций, которые производят расчет стоимости потребления некоторого ресурса⁹. Схема работы ОПО приведена на рисунке 5.9.

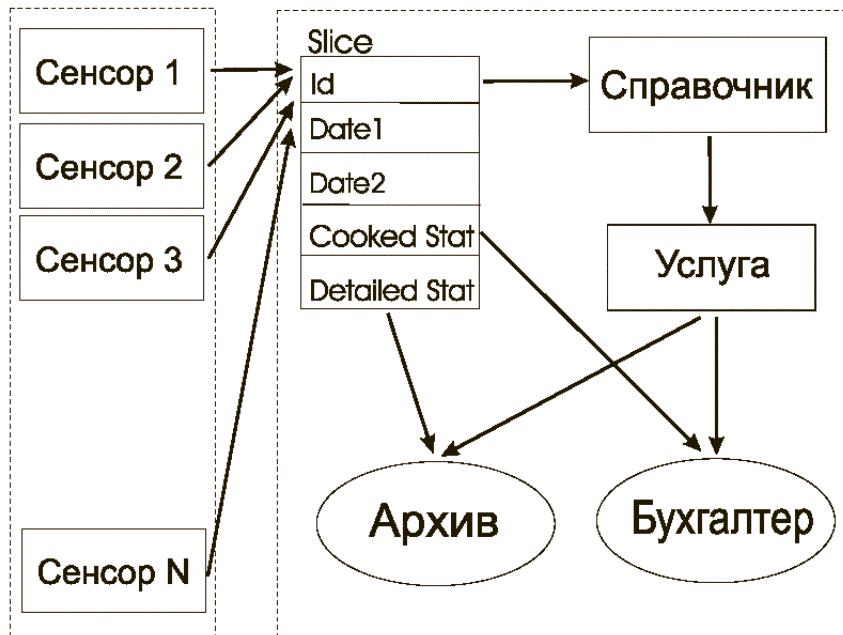


Рис. 5.9: Основной процесс-обработчик

Принцип работы.

Процесс организован следующим образом: в самом начале работы ОПО происходит инициализация. Она заключается в построении справочника (подробнее см. раздел 5.1.6.4), кэшировании необходимой для работы системы части базы данных, которая не попадает в справочник, и запуске сенсоров.

При запуске сенсоров каждый из них обязательно сообщает ОПО свой тип. Это необходимо для того, чтобы ОПО с самого начала имел информацию, какой биллинговой функцией (подробнее см. раздел 5.1.6.5) должны обрабатываться порции статистики, приходящие от данного сенсора. Кроме того, каждая связь с каждым сенсором запоминается в памяти ОПО для того, чтобы можно было вовремя выявить и, если нужно, перезапустить неработоспособный сенсор.

⁹Ресурс — например, внешний трафик, IP-телефония, арендуемое ftp-пространство и т. д.

После завершения инициализации ОПО начинает принимать и обрабатывать порции статистики, приходящие от сенсоров.

Процесс обработки порций статистики организован следующим образом: каждый сенсор связан с ОПО с помощью канала (*pipe*). Для каждого открытого канала работает функция `select`¹⁰. Обнаруженные на каком-либо канале данные складываются в буфер. Затем к содержимому буфера применяется парсер порций статистики.

Парсер просматривает буфер и пытается выбрать из него готовые порции статистики. Парсер забирает из буфера готовые порции статистики до тех пор, пока в буфере ничего не останется или же в буфере останется только одна незаконченная порция. Затем парсер разбивает каждую из готовых порций по полям и возвращает массив порций статистики, представленных с помощью хэша¹¹, где ключами являются названия полей, а данными являются соответствующие значения полей. Структура порции статистики приведена на рисунке 5.10.

Обработка каждой готовой порции статистики состоит из следующих шагов:

1. *Определение услуги*: услуга — объект базы данных, непосредственно связанный с абонентом, характеризующий род ресурса. Объект *услуга* необходим биллинговым функциям, так как в нем хранится информация, необходимая для вычисления стоимости потребленного ресурса. Услуга определяется по идентификатору (*Id*) из порции статистики с помощью справочника (подробнее см. раздел 5.1.6.4).
2. *Запуск биллинговой функции*: каждый из объектов типа “Услуга” наследуется от базового класса с одноимённым названием (класс `Nadmin::Service`). Данный класс содержит метод `Nadmin::Service::billing($date1, $date2, $resource, $id, %cooked_stat)`, который запускается при обработке каждой порции статистики. Методу на вход подаётся порция статистики `%cooked_stat`, информация об ее источнике (ресурс `$resource`) и идентификатор потребителя `$id`. В зависимости от значения ресурса запускается соответствующая данному ресурсу биллинговая функция, которой передаётся информация, хранящаяся в заголовке порции статистики. Все биллинговые функции описываются в конфигурационном файле `Nadmin`¹². Биллинговая функция возвращает сумму (оценку стоимости потреблён-

¹⁰Select — системная функция, позволяющая определять, на каких *pipe*'ах есть данные. Она необходима для того, чтобы избежать засыпаний процесса при попытках обращения к *pipe*'ам, где нет готовых данных.

¹¹Хэш — тип данных языка Perl, напоминающий массивы, только в качестве ключей используются не целые числа, а произвольные строки.

¹²Конфигурационный файл `Nadmin` — специальный файл, в котором описаны функции и переменные, специфичные для сети, в которой эксплуатируется система `Nadmin`.

```

Date1: <дата 1>
Date2: <дата 2>
Id: <идентификатор>
<название поля 1>: <значение поля 1>
<название поля 2>: <значение поля 2>
...
<название поля n>: <значение поля n>

<необработанная статистика>
<необработанная статистика>
<необработанная статистика>
...

```

Рис. 5.10: Структура порции статистики

ного ресурса), которая заносится в виде расхода на счет организации, потребившей данный ресурс.

3. *Архивация порции статистики*: после того, как подсчитана стоимость и проведены все бухгалтерские транзакции, порция статистики архивируется с помощью специального класса `Nadmin::Archive` (подробнее см. раздел 5.1.5).

С некоторой периодичностью, примерно раз в пять секунд, происходит вызов специальной функции, в обязанности которой входит:

1. проверка работоспособности подсистем ОПО (в частности, работоспособность сенсоров). В случае неработоспособности какого-либо из них, происходит перезапуск этого сенсора;
2. проверка релевантности информации, хранящейся в памяти ОПО. Средствами `Eventlog`'а проверяются все изменения, которые проходили в базе данных, и в случае критичных для ОПО изменений запускаются функции обновления справочника и перепостроения кэшей.

5.1.6.2 Параметры запуска ОПО

Запуск ОПО производится командой `main_process.pl [OPTION]`. Получить информацию о параметрах запуска ОПО (`OPTION`) можно при помощи ключа `-h` (`--help`).

Доступны следующие ключи запуска:

- v — режим запуска с подробными уведомлениями о работе программы (verbose mode);
- disable-archiving — режим запуска без использования архива; слайсы не сохраняются в архив (*disables archiving of splices*);
- yp_lodate=YMDHMS — режим запуска с построением справочника от заданной даты в формате YMDHMS (см. раздел 5.1.12); по умолчанию берется дата за 16 дней до настоящего времени (*YellowPages lodate (default now → prev(16))*);
- yp_hidate=YMDHMS — режим запуска с построением справочника до заданной даты в формате YMDHMS; по умолчанию берется настоящее время (*YellowPages hidate (default now)*);
- h, --help — выводится информация о параметрах запуска; программа завершается (*display help and exit*).

В случае неверно заданного ключа программа выдаст сообщение об этом с ссылкой на ключ -h (--help):

```
main-process: bad argument arg
Usage: main_process [OPTIONS]
run with --help to learn options.
```

Другие параметры работы ОПО задаются в начале программы в виде констант и при необходимости могут быть изменены. В качестве основных констант можно выделить следующие:

- pid file — путь к файлу с номером ОПО-процесса;
- log file — путь к log-файлу ОПО;
- pipe length — размер pipe-буфера;

5.1.6.3 Система оповещения ОПО

Log-файлы ОПО. Для отслеживания работы ОПО и происходящих ошибок программа ведет собственные log-файлы.

Config::mainlogfile — log-файл для записи сообщений при запуске программы, сообщений в случае получения внешних сигналов. По умолчанию расположен по адресу /var/log/nadmin/main_process.log;

`main_process.log` — log-файл для записи сообщений о работе программы (информация о расчете, сообщения от сенсоров и т. п.). По умолчанию расположен по адресу `/var/log/nadmin/main_process.log`;

`main_process.fail.log.rebill` — log-файл для записи сообщений о критичных ошибках, возникающих в ходе работы программы (информация о пропущенных слайсах, об ошибках при сохранении статистики в архив и т. п.). По умолчанию расположен по адресу `/var/log/nadmin/main_process.fail.log.rebill`.

Почтовые оповещения ОПО. Часть информации о работе ОПО отправляет по электронной почте (e-mail): информация о сбоях в работе сенсоров, информация о процессах, работающих по расписанию (механизм `cron`) и т. п.

Информацию, записываемую в log-файлы ОПО, можно перенаправить на e-mail, или наоборот, информацию, поступающую на e-mail, можно записывать в log-файлы ОПО, а также при желании дублировать информацию, переписав функции оповещения `inform_the_master()`, `inform_the_master1()`, входящие в состав ОПО.

Сообщения ОПО: описание событий, ошибки. При запуске ОПО в режиме оповещения `verbose-mode` выдаются сообщения, комментирующие работу программы и происходящие события. Данные сообщения (см. рис. 5.11) информируют о следующих событиях, происходящих при запуске ОПО:

- построение справочника (в справочник были сохранены `k` организаций, `m` лицевых счетов, `n` объектов БД, `p` modemных подключений);
- построение списка маршрутизаторов подсетей;
- построение списков почтовых рассылок и перенаправлений;
- создание пространства имен для биллинговых функций;
- запуск сенсоров.

Сообщения об ошибках. При работе ОПО возможны следующие сообщения об ошибках:

- Ошибка при сохранении слайса в архив, более подробное описание записывается в log-файл `main_process.fail.log.rebill` или отправляется по e-mail:

`Archive tried to die. Check mail.`

```
Starting main-process...

Info: running without archiving (при активации ключа --disable-archiving)

Building YellowPages
Caching Orgs...
Cached k orgs.
Caching Accounts...
Cached m accounts.
Caching listings...
Cached n objects.
Building listings...
Building migrating...
Precaching retrospective dialup connections...
Precached p dialup connections...

Theese TO connects couldn't be processed: con_1, con_2, ... con_n
Converting dialup precache to main (yp->{YP})...

Building system services...
These gateways were already found in YP: IP_1, IP_2, ... IP_n

Building Mailing Lists...

Sorting YP...
Building of YellowPages done.

Initializing namespace for config.pm's billing functions

Running sensors
Sensor_1 [OK] || Sensor_1 [Failed]
Sensor_2 [OK] || Sensor_2 [Failed]
...
Sensor_n [OK] || Sensor_n [Failed]

Starting the main loop
...
House keeping
House keeping
...
```

Рис. 5.11: Сообщения ОПО при запуске в режиме verbose-mode

- Ошибка при обсчете слайса, более подробное описание записывается в log-файл `main_process.fail.log.rebill` или отправляется по e-mail:

```
Billing tried to die. Check mail.
```

- Ошибка при получении слайса от сенсора, более подробное описание записывается в log-файл `main_process.fail.log.rebill` или отправляется по e-mail:

```
Bad... very bad buffer...
```

- Ошибка в слайсе: неверный интервал дат:

```
=====
main-proc: Bad dates in slice
-----
Sensor: sensor-name Org: org-name
Wrong dates (DATE2<=DATE1 || just bad) found
while processing slice: slice
-----
```

- Ошибка при сохранении слайса в архив:

```
=====
main-proc: Archive failure
-----
Tried to die while archive
(sensor-name, slice-ID)->put(DATE1, DATE2)
ORG: org-name on slice: slice
-----
```

- Ошибка при обсчете слайса:

```
=====
main-proc: Billing failure
-----
Tried to die while billing
(DATE1, DATE2, sensor-name, slice-ID, some cooked stat)
ORG: org-name on slice: slice
-----
```

- Ошибка при получении слайса от сенсора (очень большой слайс):

```
=====
main-proc: bad/incomplete slice
-----
Slice is too big or it's in wrong format,
trying to correct it: slice
-----
```

- Ошибка при анализе слайса (неправильный формат слайса):

```
=====
main-proc: bad slice... died
-----
pipe buffer for sensor-name was unparsable! :(
slice
Dumped: string-before-slice)
-----
```

- Ошибка сенсора при попытке получения слайса от сенсора:

```
=====
main-proc: stopping sensor sensor-name
-----
Critical problems while reading from sensor-name.
Stopping the sensor.
-----
```

- Попытка перезапуска неработающего сенсора:

```
=====
main-proc: rerunning sensors
-----
There are dead sensors.
Rerunning them
-----
```

5.1.6.4 Справочник

Справочник является вспомогательной частью ОПО, хотя может использоваться самостоятельно. Справочник реализует:

1. *Построение ретроспективного кэша*: поскольку все услуги (в общем случае это объекты базы данных) хранятся вместе со всей историей их изменений (это достигнуто за счёт использования системы RCS), для нас важно включить историю изменений в кэш. При построении кэша пользователем задаётся (или задаётся по умолчанию) глубина просмотра истории. Состояния кэшируемых объектов базы данных “откатываются” во времени на заданную глубину, и в кэш заносятся все состояния объекта до момента построения кэша. Ключом каждого состояния объекта становится дата и некоторый уникальный идентификатор, который строится из одного или нескольких полей объекта (это специфично для каждого типа объектов).
2. *Определение услуги по идентификатору и дате*: после построения кэша к справочнику начинают поступать запросы на определение услуги по идентификатору и дате. Для этого просматривается кэш и выбирается объект с совпадающим с входным идентификатором и датой, ближайшей к входной.
3. *Обновления кэша*: во время работы справочника в базе данных могут происходить изменения (добавление, удаление или изменение данных/объектов). Все подобные изменения обязательно фиксируются в Eventlog'е. При получении от Eventlog'a сообщений о каких-либо изменениях, ОПО выбирает только те сообщения, которые сигнализируют о существенных для него изменениях и направляет массив с этими сообщениями функции обновления справочника. Функция обновления справочника, в свою очередь, определяет, на что влияют эти изменения и обновляет только те части кэша, на которые непосредственно повлияли изменения, описанные в полученных сообщениях. После этого ОПО продолжает свою работу.

Класс Nadmin::YellowPages. Класс Nadmin::YellowPages включает в себя следующие методы:

`Nadmin::YellowPages::new()` — создать новый Nadmin::YP-объект;

`Nadmin::YellowPages::cache_orgs()` — сохранить в память справочника организации из индексного файла; в режиме `verbose-mode` появится сообщение о количестве запомненных организаций;

`Nadmin::YellowPages::cache_accounts()` — сохранить в память справочника лицевые счета организаций; в режиме `verbose-mode` появиться сообщение о количестве запомненных счетов;

`Nadmin::YellowPages::build(date1, date2)` — построить справочник для временного интервала `date1-date2`;

`Nadmin::YellowPages::build_listings(date1, date2)` — сохранить в память справочника объекты БД (`Connect`, `IPphone`, `Hosted`, `Ftp`, `Net`, `Forward`) для временного интервала `date1-date2`;

`Nadmin::YellowPages::build_mlists(date1, date2)` — сохранить в память справочника списки рассылки для временного интервала `date1-date2`;

`Nadmin::YellowPages::build_migrating(date1, date2)` — сохранить в память справочника модемные подключения для временного интервала `date1-date2`;

`Nadmin::YellowPages::build_system_services(date1, date2)` — сохранить в память справочника системные сервисы для временного интервала `date1-date2`;

`Nadmin::YellowPages::lookup(id, date)` — найти организацию по `id` на заданную дату `date`;

`Nadmin::YellowPages::update(@events)` — обновить справочник.

5.1.6.5 Биллинговые функции

Биллинговые функции являются специфичным механизмом определения стоимости оказанной услуги для каждого провайдера. По этому они вынесены в конфигурационный файл. С одной стороны, язык их описания должен быть достаточно простым, с другой, автор биллинговых функций не должен быть ограничен специализированным языком. Исходя из того, что авторами биллинговых функций чаще всего являются системные программисты, было решено использовать язык Perl для их написания.

Для удобства написания биллинговых функций разработано специальное пространство имен (`namespace`) и поддерживается соглашение, что перед вызовом любой биллинговой функции все эти переменные правильно проинициализированы. В это пространство имён входят все переменные, которые необходимы для определения стоимости оказанной услуги. Имена переменных совпадают с названиями полей из заголовка порции статистики. Такой подход значительно упрощает процесс написания биллинговых функций.

Переменные, которые автоматически инициализируются в пространстве имен ОПО:

- `$self` — ссылка на услугу, которая в данный момент обрабатывается;
- `$date1, $date2` — даты из порции статистики, обозначающие интервал, во время которого проходило использование ресурса;
- `$id` — идентификатор потребителя.

В пространство имен ОПО могут быть добавлены любые другие дополнительные переменные, которые необходимы для расчета стоимости потребленного ресурса на усмотрение системного администратора.

Класс `Nadmin::Service`. Класс `Nadmin::Service` включает в себя следующие методы:

`Nadmin::Service::init_namespace()` — инициализировать пространство имен ОПО;

`Nadmin::Service::archive(resourse, id, path)` — определить/создать директорию в архиве для сохранения слайса;

`Nadmin::Service::billing(date1, date2, resource, id, cooked_stat)` — произвести обсчет статистики из слайса, используя биллинговые функции из конфигурационного файла `Config.pm` (подробнее см. раздел 5.1.10).

5.1.6.6 Перерасчет

Во время расчетов могут происходить ошибки (в результате сбоя базы данных или еще по каким-то причинам). Для выявления и исправления таких ошибок используется перерасчет за заданный период времени.

Имеется две стадии перерасчета:

1. перерасчет всей статистики за заданный период и формирование отчетов;
2. подстановка перерасчитанной статистики вместо старой (неверной) статистики.

Первая стадия может выполняться в двух режимах:

1. перерасчет всей статистики без пересоздания порций статистики;
2. перерасчет всей статистики с пересозданием порций статистики.

В первом режиме порции статистики за заданный период целиком берутся из архива и целиком отдаются ОПО на пересчет. Во втором режиме порции статистики сначала извлекаются из архива и перед тем как пойти на пересчет ОПО, они отдаются функциям построения порций статистики в сенсоры, и только после этого они поступают в ОПО.

Таким образом, первый режим позволяет исправить ошибки, допущенные на уровне сенсоров, а второй режим исправляет ошибки, допущенные на уровне ОПО, например, в результате сбоя справочника.

5.1.7 Процесс обновления системной конфигурации

Называть систему Nadmin *административной* позволяет ее способность автоматически обновлять конфигурацию системных сервисов как на центральных серверах, так и на маршрутизаторах сети. В данной главе описан принцип и алгоритмы процесса обновления системной конфигурации (ПОСК).

В предыдущей версии системы Nadmin использовался следующий подход: при изменении какого-либо объекта системы вызывалась специальная функция, которая проверяла важность этих изменений, и в случае необходимости производила редактирование конфигурационных файлов соответствующих сервисов. За счет того, что редактировались только те настройки, которые относились к измененному объекту, процесс проходил относительно быстро, хотя в некоторых случаях, при нестабильной работе сети, этот процесс мог затягиваться надолго. Данных подход является достаточно сложным для реализации, потому что приходится учитывать множество нюансов, и сам процесс автоматического редактирования текстовых конфигурационных файлов не прост, что порой приводило к ошибкам.

В новой версии системы был использован кардинально другой подход: после любого изменения объектов системы в Eventlog заносится соответствующая запись об этом. На этом для человека (или процесса), который произвел эти изменения, процедура изменения заканчивается. Сразу после поступления соответствующей записи в Eventlog начинает работать ПОСК. Вся его работа происходит в фоновом режиме, то есть она никак не мешает ни ОПО, ни каким либо другим подсистемам Nadmin. ПОСК не пытается определить, на что повлияли прошедшие изменения, а просто строит всю системную конфигурацию заново, а затем ее инсталлирует.

Данный подход позволяет значительно ускорить процесс изменения каких-либо объектов системы, потому что нам не приходится заботиться об обновлении системной конфигурации. Об этом за нас позаботится ПОСК.

В СТ “Ботик” используются следующие сервисы, услуги и средства администрирования сети:

- UUCP — сервер почтовой службы. Для него генерируются файлы:
 - /etc/uucp/sys: файл с учетными записями.
 - /etc/uucp/passwd: файл с паролями.
- DNS сервер (named). Для этого сервера предоставляется информация о соответствии IP-адресов и имён компьютеров в сети.

- Сервер аренды FTP-пространства. Составляется список учетных записей и паролей для доступа к арендуемому пространству.
- Модемный доступ. Для этого сервиса также предоставляется список учетных записей и паролей для доступа к модемному пулу.
- Ключ доступа в сеть BotikKey. Для BotikKey составляется список паролей на доступ к тому или иному уровню доступа.

Процесс организован следующим образом: на центральном сервере непрерывно работает программа `sysconfd`, которая получает сообщения от `Eventlog`, и когда встречает сообщения об изменениях базы данных, по очереди запускает программы:

1. `sysconf-build-all` — программа, которая строит файлы с текущей конфигурации для всех сервисов;
2. `sysconf-inst-all` — программа, которая инсталлирует свежие конфигурационные файлы, построенные программой `sysconf-build-all`, в используемые сервисы.

5.1.8 Программы синхронизации баз данных систем Nadmin

Синхронизация баз данных двух версий системы Nadmin применяется в следующей ситуации:

- в штатном режиме работает *стабильная версия* системы *Nadmin*, с которой происходит взаимодействие администраторов и абонентов СТ “Ботик”;
- одновременно тестовую эксплуатацию на другом компьютере проходит *новая версия* системы *Nadmin* с измененной и/или повышенной функциональностью;
- будем эти две работающие системы называть соответственно *стабильной* и *отлаживаемой*;
- в стабильную и отлаживаемую системы Nadmin поступает статистика потребления абонентами услуг и трафика;
- в обеих системах работают механизмы расчета с абонентами.

Программы синхронизации решают следующие задачи:

- ввод в базу данных отлаживаемой системы Nadmin изменений из базы данных стабильной системы Nadmin, сделанных с интерфейса администратора или абонентов;
- сравнение баз данных двух систем для выявления расхождения в функционировании стабильной и отлаживаемой систем Nadmin.

Таким образом, на синхронизации баз данных базируется отладка новой версии системы Nadmin.

Синхронизация баз данных (БД отлаживаемой системы Nadmin с БД стабильной системы Nadmin) выполняется при помощи двух программ: сканера `db_scanner.pl` и синхронизатора `db_synchronizer.pl`.

5.1.8.1 Сканер

Сканер `db_scanner.pl` запускается на компьютере, где размещена база данных стабильной системы Nadmin.

Команда запуска сканера выглядит следующим образом:

```
ssh adm@host db-scanner.pl [-v] [-d] lodate hidate | less
```

Здесь `adm@host` задает имя компьютера (`host`), где работает стабильная система Nadmin и имя пользователя (`adm`), имеющего соответствующие права.

При запуске сканер получает два обязательных параметра:

- начальную `lodate` и конечную `hidate` даты, для определения периода времени, за который нужно искать версии rcs-файлов базы данных.

Кроме того, в команде запуска могут быть указаны следующие ключи:

- `-d` — режим просмотра версий удаленных объектов базы данных — объектов, которые в момент синхронизации не существуют в базе данных, но существовали с период с `lodate` по `hidate`;
- `-v` — режим выдачи сообщений о прохождении этапов сканирования.

После запуска сканер:

- Строит массив всех версий всех rcs-файлов базы данных, а также массив всех версий файла `db_filelist`, созданных в период с `lodate` по `hidate`.

Файл `db_filelist` содержит список файлов базы данных системы Nadmin. Каждая строка `db_filelist` включает относительный путь к какому-либо файлу и

название этого файла. Новая версия `db_filelist` появляется, если за сутки произошло изменение в структуре базы данных.

- Анализирует каждую версию из массива и выводит в стандартный поток (`stdout`) информацию об изменении объекта в одном из двух следующих вариантов форматов:

— изменение лицевого счета:

```
event date Account name author comment
- string
+ string
```

— изменение других объектов:

```
event date Object name author comment
- field = value
+ field = value
```

Здесь *event* — событие, в результате которого изменился объект базы данных (`CREATED`, `DELETED`, ...), *date* — дата занесения новой rcs-версии, *name* — имя объекта, *author* — автор rcs-версии, *comment* — комментарий rcs-версии, *string* — строка файла `account` лицевого счета, *Object* — название класса DBfile-объекта (`Org`, `Connect`, ...), *field* — название свойства объекта, *value* — значение свойства объекта.

Знак “+” предшествует новой строке, которая появилась в rcs-файле. Знак “-” предшествует строке, которая была удалена из rcs-файла. Данные о разных объектах при выводе в поток разделяются пустой строкой.

Состав строк выводимых данных об объекте зависит от события, которое с ним произошло. Бывают следующие виды событий: создание, изменение, перемещение, удаление, восстановление.

Текущая версия сканера учитывает особенности архитектуры базы данных первой версии системы Nadmin.

Так как класс `Nadmin::Subnet` в первой версии системы Nadmin не предполагает хранение своих экземпляров в отдельных файлах — есть один rcs-файл `subnets` для описания всех подсетей, для анализа событий, произошедших с подсетями, написаны отдельные функции.

5.1.8.2 Синхронизатор

Программа `db_synchronizer.pl` запускается на компьютере, на котором находится база данных отлаживаемой системы Nadmin.

Команда запуска синхронизатора выглядит следующим образом:

```
db_synchronizer.pl [-v] [-d] [-c] lodate hidate
```

При запуске синхронизатор может получить два параметра:

- начальную *lodate* и конечную *hidate* даты (они предназначены для передачи сканеру).

Если *lodate* не передана синхронизатору, то она берется из марки. Если *hidate* не передана синхронизатору, то вместо нее берется текущее время.

Кроме того, в команде запуска могут быть указаны следующие ключи:

- `-d` — режим вызова сканера (из синхронизатора) с ключом `-d`;
- `-v` — режим выдачи сообщений о прохождении этапов синхронизации (этот же ключ передается вызываемому сканеру);
- `-c` — режим внесения строк расходов за расчетные периоды при синхронизации объектов `Nadmin::Account`¹³.

После запуска синхронизатор:

- получает *lodate* из марки (если это необходимо) и записывает в марку *hidate*;
- запускает сканер на машине *host* от имени пользователя *adm*;
- с помощью потока (`pipe`) принимает от сканера информацию об изменении базы данных;
- изменяет базу данных отлаживаемой системы Nadmin.

Синхронизатор может работать в одном из двух режимов — режиме конвертора или режиме синхронизатора.

¹³Строки расходов из лицевых счетов стабильной системы Nadmin переносятся в лицевые счета отлаживаемой системы Nadmin, если отлаживаемая система Nadmin не производит расчеты за расчетные периоды.

5.1.8.3 Режим конвертора

Работа синхронизатора в режиме конвертора предполагает воссоздание базы данных стабильной системы Nadmin в отлаживаемой системе Nadmin в новом формате.

Для этого `db_synchronizer.pl` должен обязательно быть запущен с ключом `-c` и без ключа `-d`.

5.1.8.4 Режим синхронизатора

Работа в режиме синхронизатора предполагает, что:

- база данных стабильной системы Nadmin уже ранее воссоздана в отлаживаемой системе;
- в файлы `account_run` отлаживаемой системы Nadmin поступают текущие расходы;
- в базу данных отлаживаемой системы Nadmin внесены изменения различными подсистемами (web-интерфейс, обработчики журнала событий и т. п.).

В данном режиме синхронизатор должен быть запущен с ключом `-d` и без ключа `-c`.

Получая данные о каком-либо объекте от сканера, синхронизатор делает несколько проверок.

- Если объект уже существует в базе данных отлаживаемой версии системы Nadmin и дата его последней rcs-версии больше даты rcs-версии от сканера, то в качестве даты новой rcs-версии берется текущее время.
- Если версия объекта `Nadmin::Account` из базы данных стабильной версии системы Nadmin связана с внесением расхода, то она не заносится в базу данных отлаживаемой системы Nadmin.

5.1.9 Журнал событий и класс `Nadmin::Eventlog`

Журнал событий является связующим звеном всей системы Nadmin, позволяющим координировать и синхронизировать ее работу. Основными функциями журнала являются запись и чтение коротких сообщений. Такими сообщениями могут быть описания изменений объектов базы данных (подключений, лицевых счетов), конфигурации системы через web-интерфейс и т. п.

Журнал событий является объектом класса `Nadmin::Eventlog`.

Каждому журналу соответствует файл (например, `eventlog`), куда производится запись и откуда производится чтение текстовых строк — описаний событий. Вот примеры таких описаний событий:

```
1080836596 DBfileCheckin Nadmin::WWWuserif 3155  
1081162910.25819 DBfileCheckin Nadmin::Org 3157  
1082005511.84747 Account_register_payment MultiProc 100
```

Методы класса `Nadmin::Eventlog`:

- `my $eventlog = new Nadmin::Eventlog(@args)` — данный метод возвращает объект `$eventlog`. Массив `@args` может содержать имя (идентификатор) файла для объекта `$eventlog` и/или значение свойства `WATCHER_ID` (имя процесса, использующего объект `$eventlog`).
- `my $marker = $eventlog->marker()` — данный метод возвращает объект класса `Nadmin::Mark` `$marker`, который содержит последнюю прочитанную строку журнала для данного процесса.

В механизме класса `Nadmin::Eventlog` поддержана возможность использования разных закладок для разных процессов, которая позволяет независимым процессам “помнить” последнее сообщение, которое они получали, и обработать все необработанные события.

Для определения новых записей в журнале `$eventlog` используются закладки (объект `Nadmin::Mark`). В закладке хранится строка, которая была прочитана в последний раз. При вызове методов `get`, `try_get` файл с сообщениями просматривается до строки, совпадающей со строкой, хранящейся в закладке. В качестве результата вызова функций возвращается строка, следующая за совпавшей строкой. При этом закладка обновляется, то есть в нее сохраняется возвращаемая строка. Таким образом, при вызове методов `get`, `try_get` каждый раз будет браться очередная строка из файла с сообщениями.

- `$eventlog->put($data)` — данный метод записывает новую строку `$data` в журнал `$eventlog`.

Процессы, которые производят любые изменения в системе, должны своевременно делать соответствующие записи в журнал. Для этого используется метод `put`, которому передается строка, содержащая некоторое описание произведенного изменения. В начало этой строки добавляется дата (с точностью до миллисекунд), после чего строка записывается в файл.

- `my $date = $eventlog->get()` — данный метод возвращает новую запись `$date` из `$eventlog`.

Это позволяет процессам, которым необходимо своевременно отслеживать какие-либо изменения БД, использовать журнал для получения сообщений.

Особенностью метода `get` является то, что если при его вызове в файле, где хранятся все сообщения, нет ни одной новой строки, то процесс, вызвавший его, засыпает до того момента, пока не появится новая строка.

- `my $date = $eventlog->try_get()` — данный метод возвращает новую запись `$date` из `$eventlog` для процесса, который не должен засыпать при вызове метода `get`.
- `$eventlog->set_mark_at_end()` — данный метод создает закладку на последнюю строчку файла журнала `$eventlog`.

Существуют некоторые процессы, работа которых не зависит от того, что происходило в то время, пока они не были запущены, то есть они зависят только от событий, которые происходят во время их работы. Специально для таких процессов был реализован метод `set_mark_at_end`. Этот метод устанавливает закладку на последнюю строчку файла с сообщениями, и поэтому пропускает все события, о которых было оповещено в `Eventlog` до момента вызова `set_mark_at_end`.

Чтобы избежать переполнения файла с сообщениями, его необходимо периодически архивировать. Для этого необходимо переместить его в другое место, а на его месте оставить новый, пустой файл. Старый файл компрессируется (в `bzip2`) и перемещается, а все новые сообщения будут попадать в новый файл. Этот процесс автоматизирован, что избавляет системного администратора от рутинных операций.

5.1.10 Конфигурационный файл config

Конфигурационный файл системы Nadmin `config` необходим для записи данных, значение которых может быть различным в зависимости от особенностей региональной компьютерной сети, расчета с абонентами и т. п.

Файл `config` содержит описание переменных и функций, которые принадлежат классу `Nadmin::Config`, например:

- `%path` — содержит пути к различным объектам, например, указание о том, что база данных системы Nadmin находится в директории `/var/lib/nadmin/db`, выглядит следующим образом:

```
%path =
(
  ...
  db => "/var/lib/nadmin/db",
  ...
);
```

- **%resources** — описывает ресурсы статистики трафика, например, здесь:

```
%resources = (
  ip      => ["ip:slav", "ip:post"] ,
  proxy   => ["squid"] ,
);
```

задано, что ip-трафик считают сенсоры `sensor_ip` и `sensor_tproxy` с ПК-маршрутизаторов `post` и `slav`, а proxy-трафик считает сенсор `sensor_squid`.

- **@mydomains** — задает список доменных имен внутри сети, например, здесь:

```
@mydomains = qw( pereslavl.ru
                  botik.ru
                  slavich.botik.ru
                  ...
);
```

указано, что к доменам региональной сети, управляемой системой Nadmin, относятся домены `pereslavl.ru`, `botik.ru` и другие.

Кроме того, файл `config` содержит описания биллинговых функций таких, как `bill_squid`, `bill_ip` и других. Они необходимы для обсчета пользовательского трафика от различных источников и используются методами класса `Nadmin::Service`. Пример описания функции:

```
sub bill_ip
{
  my $megabytes = ($Count_in + $Count_out)/(1024*1024);
  my $res = $megabytes * $Nadmin::Config::megabyte_price;
  warn "billing[$resource]($orgname)($$date1-$$date2): $res\n";
  return ($res, $Count_in, $Count_out);
}
```

Другие примеры можно найти в файле `config`, который входит в дистрибутив системы Nadmin.

5.1.11 Вспомогательный класс `Nadmin::Lock`

Класс `Nadmin::Lock` служит для установления блокировки на различные объекты (например, файлов), обрабатываемых одним процессом, чтобы другой процесс в то же время не смог их изменить.

Для этого блокируемый объект должен иметь идентификатор (`$id` — строковая переменная), по которому будет создаваться объект класса `Nadmin::Lock`. Установление блокировки происходит так — создается файл в директории

```
$Nadmin::Config::path{"lock"}
```

с именем `$id` и выполняется системный вызов `flock` этого файла.

Методы класса `Nadmin::Lock`:

- `my $lock = new Nadmin::Lock ($id)` — данный метод возвращает объект `$lock` класса `Nadmin::Lock` для идентификатора `$id`.
- `$lock->lock()` — установка блокировки на объект `$lock`.
- `my $res = $lock->try_lock()` — данный метод устанавливает блокировку на объект `$lock`. В случае успеха метод возвращает единицу, иначе метод возвращает нуль — блокировка не произошла. Повторное применение этого метода позволяет процессу, которому необходимо установить блокировку на объект `$lock`, установить ее сразу же, когда это становится возможным (например, когда другой процесс снимает блокировку с этого объекта).
- `$lock->unlock()` — снятие блокировки с объекта.
- `my $res = $lock->is_locked()` — если на объект `$lock` установлена блокировка, переменная `$res` равна нулю, иначе равна единице.
- `$lock->set_locker_info()` — данный метод записывает в файл объекта `$lock` название процесса, установившего блокировку на этот объект. Во время снятия блокировки эта запись удаляется.
- `my $pid = $lock->locker()` — данный метод возвращает `$pid` — pid процесса, который в данный момент использует блокировку объекта `$lock`.

5.1.12 Вспомогательный класс Nadmin::YMDHMS

Объект класса `Nadmin::YMDHMS` представляет из себя запись даты из четырнадцати цифр в формате *yyyyMMddhhmmss*.

Здесь и во всем разделе четыре цифры *yyyy* обозначают год, две цифры *yy* — две последние цифры года, две цифры *MM* — месяц, две цифры *dd* — день месяца, две цифры *hh* — час, две цифры *mm* — минуты, две цифры *ss* — секунды, число *seconds* — количество секунд с начала эпохи¹⁴, *Mon* — три первые буквы названия месяца на английском языке (например, *Jan*, *Feb* и т. д.).

Например, запись `20041110123014` обозначает дату — 10 ноября 2004 года, 12 часов, 30 минут, 14 секунд.

Методы класса `Nadmin::YMDHMS`:

- `my $date = new Nadmin::YMDHMS($ini, $ini2)` — данный метод возвращает объект класса `Nadmin::YMDHMS`. Стока `$ini` должна быть задана в одном из следующих форматов:

- *yyyyMMddhhmmss*;
- *yyyyMMdd*;
- *yyMMdd*;
- *seconds*;
- *Mon dd hh:mm:ss* (при этом для определения года обязательно должен быть задана переменная `$ini2` — объект `Nadmin::YMDHMS`);
- *dd Mon yyyy*;
- *yyyy/MM/dd hh:mm:ss*;
- *yyyy-MM-dd hh:mm:ss*.

Если `$ini` — строка другого формата, функция выдает сообщение об ошибке

```
carp "Cannot construct Nadmin::YMDHMS from $ini";
```

- `$date->checked()` — данный метод проверяет правильность даты `$date` (например, проверяется, что число, обозначающее месяц, не больше 12).

¹⁴В операционной системе Unix за начало эпохи берется дата 1 января 1970 года 00:00:00.

- my \$d = \$date->ymdhms() — данный метод возвращают строковое представление \$d объекта \$date.

Аналогично работают такие методы: ymd, yyyyymmdd, yyyyymmddhh, yyyyymm, yy, yyyy, mm, dd, hms, hhmmss, min, sec.

Они выводят строковое представление даты в следующем формате:

Метод	Формат результата
ymd	yyyyMMdd
yyyyymmdd	yyyyMMdd
yyyyymmddhh	yyyyMMddhh
yyyymm	yyyyMM
yyyy	yyyy
yy	yy
mm	MM

Метод	Формат результата
dd	dd
hh	hh
hms	hhmmss
hhmmss	hhmmss
min	mm
sec	ss

- my \$mon = \$date->mon(@language) — данный метод предназначен для создания строки из первых трех букв названия месяца даты \$date. Если @language содержит

- строку *Мая* — сокращенное название месяца будет на русском языке (*May* будет в родительном падеже), например: *Mar*, *Apr*, *Мая* и т. д. ;
- строку *Май* — сокращенное название месяца будет на русском языке (*May* будет в именительном падеже), например: *Mar*, *Apr*, *Май* и т. д. .

Иначе название месяца будет на английском языке, например: *Mar*, *Apr*, *May* и т. д.

- my \$month = \$date->month(@language) — аналог mon, но названия месяцев выводятся целиком. Если @language содержит

- строку *Мая* — название месяца будет на русском языке, падеж будет родительным, например: *марта*, *апреля*, *мая* и т. д.
- строку *Май* — название месяца будет на русском языке, падеж будет именительным, например: *март*, *апрель*, *май* и т. д.

Иначе название месяца будет на английском языке, например: *March*, *April*, *May* и т. д.

- my \$date2 = \$date1->next(\$ndays) — данный метод возвращает объект \$date2, содержащий дату, которая позже даты \$date1 на \$ndays дней.
- my \$date1 = \$date2->prev(\$ndays) — данный метод возвращает объект \$date2, содержащий дату, которая раньше даты \$date1 на \$ndays дней.

Описание всех методов класса `Nadmin::YMDHMS` можно найти в модуле `YMDHMS.pm`, входящем в дистрибутив системы `Nadmin`.

5.2 Руководство администратора гражданской сети

5.2.1 Web-интерфейс администратора

Для удобного выполнения административных функций существует web-интерфейс администратора системы `Nadmin`¹⁵. Чтобы получить доступ к web-интерфейсу администратора, необходимо на главной странице web-интерфейса правильно заполнить три поля (Ваша организация, Ваш логин, Ваш пароль). В случае отсутствия у пользователя привилегий `SYSADM` доступ к web-интерфейсу администратора будет запрещен.

Если вы получили доступ к web-интерфейсу администратора, то вы имеет полный доступ к базе данных `Nadmin` (см. рис. 5.12).



Рис. 5.12: Web-интерфейс администратора: главная страница

Страница “Начало”. На этой странице расположены ссылки на различные другие

¹⁵ В рамках данного проекта web-интерфейс администратора системы `Nadmin` доступен по адресу: <https://tux.botik.ru/~nadmin/nadmin/>

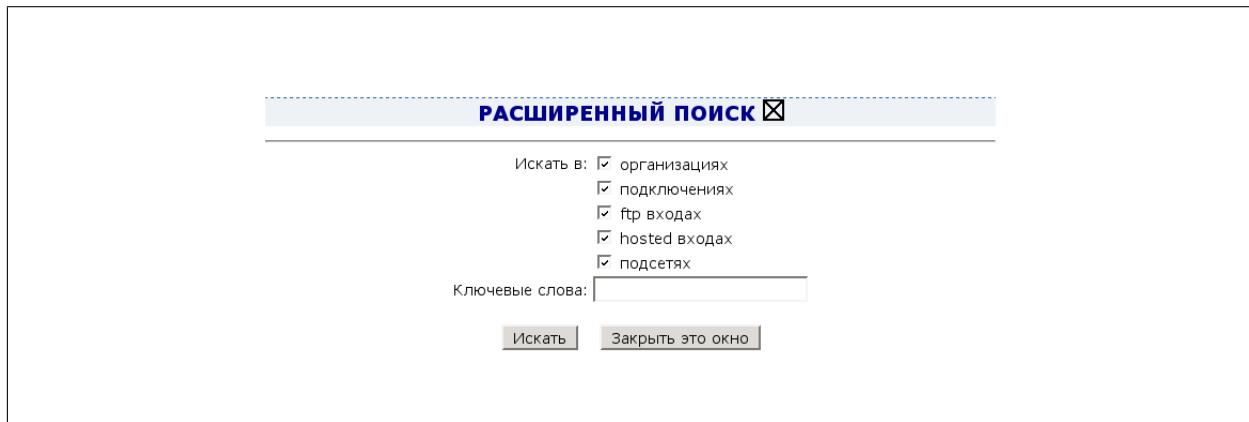


Рис. 5.13: Web-интерфейс администратора: страница поиска

страницы Web-интерфейса, такие как: абоненты, подключения, подсети, FTP-входы и т.д.

Страница “Поиск”. Одна из удобных особенностей web-интерфейса в новой системе Nadmin заключается в том, что администратор может осуществлять поиск в базе данных. Поиск происходит по организациям, подключениям, FTP-входам, hosted входам и подсетях (см. рис. 5.13).

Страница “Абоненты”. Данная страница содержит ссылки на страницы: создать нового абонента, расширенный поиск, список абонентов с описанием, список абонентов со статусом, список абонентов с датами подписания прейскурантов, светофор и расширенная выборка (см. рис. 5.14).

Страница “Создать нового абонента”. На этой странице администратор имеет возможность создать нового абонента.

Страница “Абонент::Расширенный поиск”. На этой странице администратор может осуществить поиск в базе данных Nadmin.

Страница “Список абонентов с описанием”. На этой странице отображается полный список абонентов в сети. Администратор может перейти на web-интерфейс абонента: для этого нужно щелкнуть указателем мыши на черную стрелку. Также можно посмотреть контактный телефон абонента, щелкнув указателем мыши на телефон. Чтобы посмотреть полную информацию о абоненте, необходимо щелкнуть указателем мыши на имя абонента.

Страница “Список абонентов со статусом”. На этой странице отображается список абонентов со статусом. Статус показывает, отключен абонент от сети или нет.

Страница “Список абонентов с датами подписания прейскуранта”. Данная

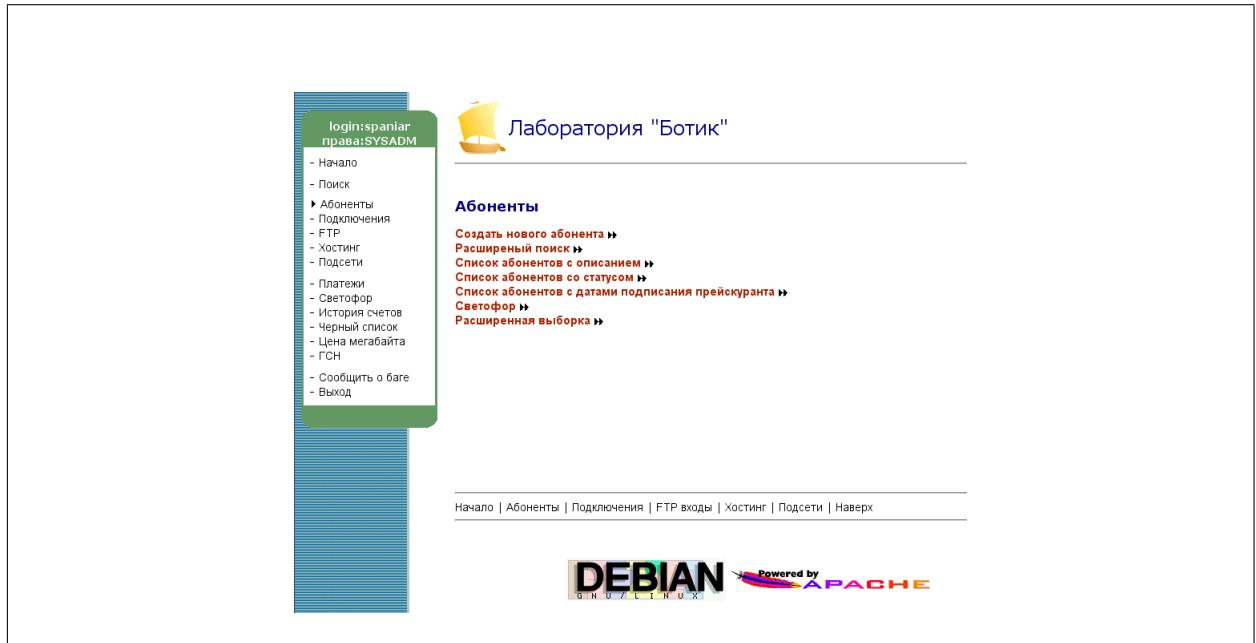


Рис. 5.14: Web-интерфейс администратора: страница “Абоненты”

страница отличается от страницы “Список абонентов с описанием” только тем, что на ней указывается дата подписания прейскуранта.

Страница “Светофор”. На этой странице отображается список абонентов с указанием значения лицевого счета. Список отсортирован по значениям лицевого счета и фон каждой строки определяется значением лицевого счета:

- Лицевой счет меньше нуля — фон красный;
- Лицевой счет равен нулю — фон желтый;
- Лицевой счет больше нуля — фон зеленый;

Страница “Абонент::Расширенная выборка”. На этой странице администратор может выбрать различные сведения об абоненте, которые ему необходимо узнать.

Страница “Подключения”. На данной странице находятся ссылки на страницы: создать новое подключение, расширенный поиск, полный список подключений и расширенная выборка.

Страница “Создать новое подключение”. На этой странице администратор может создать новое подключение у абонента.

Страница “Подключение::Расширенный поиск”. На этой странице администратор может осуществить поиск в базе данных Nadmin.

Страница “Полный список подключений”. Данная страница содержит полный список подключений. Также отображается тип подключения, имя абонента, к которому относится подключение, и подсеть, в которую входит подключение.

Страница “Подключение::Расширенная выборка”. На этой странице администратор может выбрать различные сведения о подключении, которые ему требуются.

Страница “Создать новый FTP-вход”. На этой странице администратор может создать FTP-вход у абонента.

Страница “FTP-вход::Расширенный поиск”. На этой странице администратор может осуществить поиск в базе данных Nadmin. По умолчанию в FTP-входах.

Страница “Полный список FTP-входов”. Данная страница показывает полный список FTP-входов. Также показывается имя абонента, к которому относится FTP-вход. Если щелкнуть курсором мышки на один из FTP-входов, то отобразится страница с полными сведениями о данном FTP-входе.

Страница “FTP-вход::Расширенная выборка”. На этой странице администратор может выбрать различные сведения о FTP-входе, которые ему необходимо узнать.

Страница “Создать новый вход”. На этой странице администратор может создать hosted вход.

Страница “Hosted вход::Расширенный поиск”. На этой странице администратор может осуществить поиск в базе данных Nadmin. По умолчанию поиск производится в hosted входах.

Страница “Полный список hosted входов”. Данная страница показывает полный список hosted входов. Также показывается имя абонента, к которому относится hosted вход. Если щелкнуть курсором мышки на один из hosted входов, то отобразится страница с полными сведениями о данном hosted входе.

Страница “Создать новую подсеть”. На этой странице администратор может создать новую подсеть. Также указывается имя абонента и префикс подсети¹⁶.

Страница “Подсети::Расширенный поиск”. На этой странице администратор может осуществить поиск в базе данных Nadmin. По умолчанию поиск производится в подсетях.

Страница “Полный список подсетей”. Данная страница отображает полный список подсетей. Также отображается имя абонента, к которому относится подсеть, префикс и

¹⁶Префикс подсети — уникальное пространство адресов подсети.

шлюз¹⁷. Если щелкнуть курсором мышки на одну из подсетей, то отобразится страница с полными сведениями о данной подсети.

Страница “ГСН”. На этой странице можно посмотреть сведения, нужные для отчетов для Управления Госсвязьнадзор.

Страница “Выход”. Щелкнув указателем мыши на эту ссылку, вы безопасно выходите с web-интерфейса администратора.

5.3 Руководство абонента

5.3.1 Web-интерфейс абонента

Абонент может воспользоваться рядом услуг, которые поддержаны в административной системе Nadmin:

- ip-телефония;
- учет трафика в реальном времени;
- генерация паролей BotikKey;
- управление уровнем отключения;

Для доступа к конфигурированию услуг (подключение/отключение услуги, определение параметров) в системе Nadmin был создан web-интерфейс абонента¹⁸.

Для доступа к web-интерфейсу абонент должен заполнить три поля (Ваша организация, Ваш логин, Ваш пароль). В случае успешной верификации введенных данных абонент получает доступ к своему web-интерфейсу. В противном случае ему будет предложено повторить ввод данных (см. рис. 5.15).

Web-интерфейс абонента поддерживает аутентификацию браузера, что позволяет абонентам не вводить постоянно данные (имя организации, логин абонента, пароль абонента) для доступа к последующим страницам web-интерфейса.

Абонентом региональной сети может являться организация со многими пользователями услуг сети. Это верно и для “домашних” подключений (абонент — семья, включающая несколько пользователей: мама, папа, дети). Не все пользователи должны иметь

¹⁷Шлюз (gateway) — сетевое устройство или компьютер, осуществляющие связь между двумя различными (использующими разные коммуникационные протоколы) компьютерными сетями. Кроме передачи данных, могут выполнять их фильтрацию.

¹⁸В рамках данного проекта web-интерфейс пользователя системы Nadmin доступен по адресу <https://tux.botik.ru/~nadmin/user.cgi>

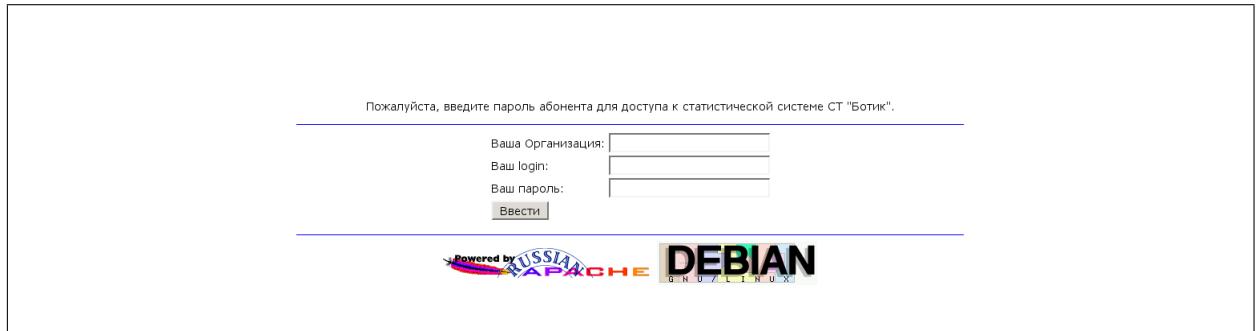


Рис. 5.15: Web-интерфейс абонента: проверка прав доступа

доступ к интерфейсу абонента — только зарегистрированные. Среди последних могут быть пользователи, имеющие разные права доступа (администраторы организаций и обычные пользователи).

На web-интерфейсе абонента поддерживается разграничение прав пользователей. В данный момент на web-интерфейсе существует два вида пользователей (администратор организации и обычный пользователь). Администратор организации может выполнять все функции обычного пользователя и, кроме того, создавать новых пользователей, удалять пользователей, менять пароль пользователям, разрешать пользователям использовать услугу IP-телефонии и строить графики расходов подключений для своей организации.

Главная страница. На странице “Главная страница” описаны услуги, предоставляемые абонентам (см. рис. 5.16).

IP-телефония. На странице “IP-телефония” абонент может ознакомиться с услугой IP-телефонии. Абонент-администратор имеет право разрешить использование услуги IP-телефонии пользователям web-интерфейса. Также абонентам предоставляется страница для просмотра своих pin-кодов и паролей для использования IP-телефонии.

Лицевой счет. На странице “Лицевой счет” абонент может посмотреть состояние своего лицевого счета (см. рис. 5.17).

Также абоненту предоставляется возможность более детально посмотреть расходы за абонентскую плату, внешний трафик и IP-телефонию. Для этого необходимо кликнуть ссылки “Абонентская плата”, “Внешний трафик” и “IP-телефония”. Расходы за внешний трафик могут быть представлены в виде графика. Чтобы посмотреть графики расходов, нужно кликнуть ссылку “График расходов”. Абонент-администратор имеет право посмотреть расходы всей организации, в то время как обычный пользователь может посмотреть расходы только своего подключения. График расходов может суммировать все виды трафика (ip-трафик, гроху-трафик, почтовый трафик) или отобразить

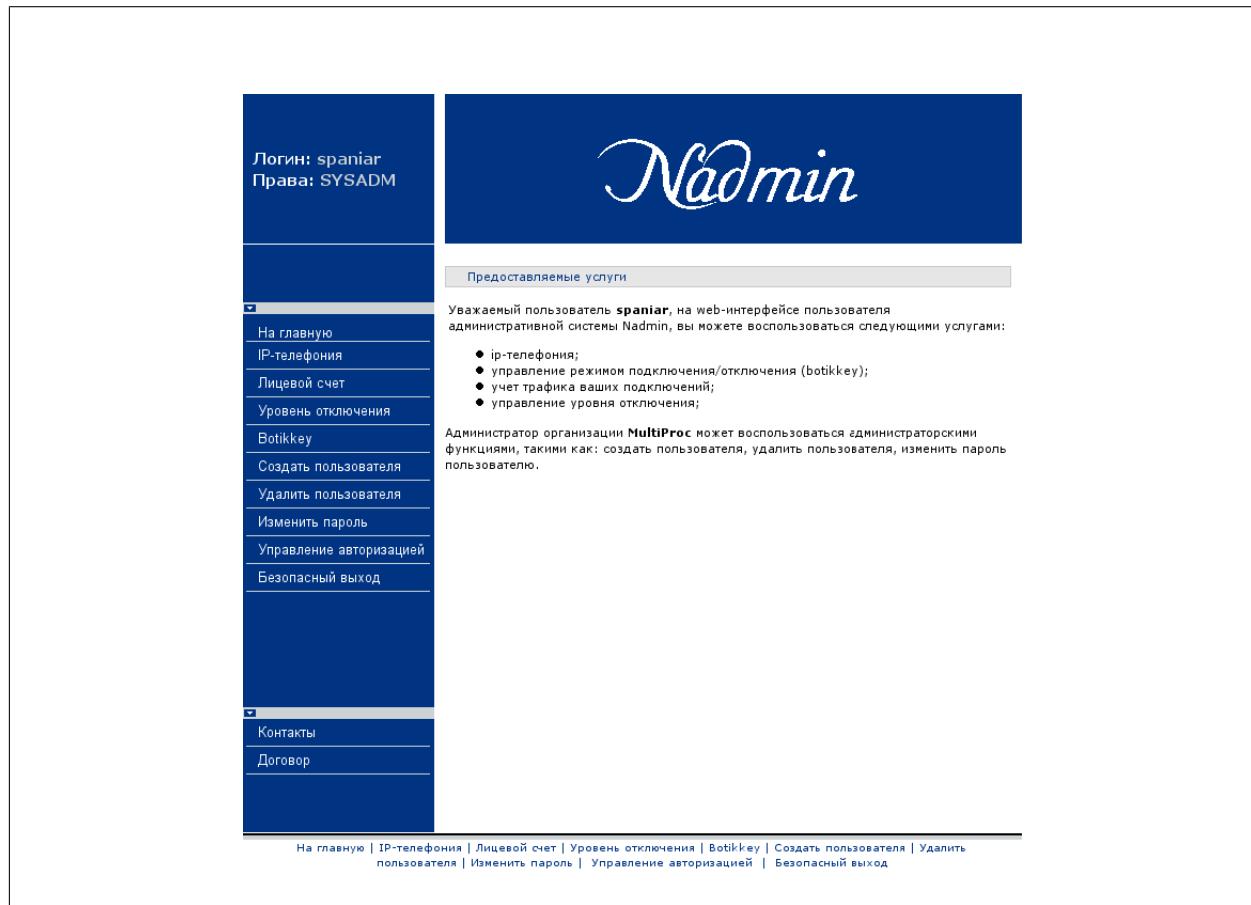


Рис. 5.16: Web-интерфейса абонента: главная страница

отдельно каждый вид трафика. График может быть построен как по всей организации, так и по отдельным подключениям.

Красный цвет на графике означает почтовый трафик, серый цвет — proxy-трафик, голубой цвет — ip-трафик. Первоначальный график строится за текущий месяц. Абонент может детализировать показ расходов после того, как построен первоначальный график. Для этого необходимо щелкнуть указателем мыши на один из столбцов, означающих день месяца. В результате этого действия построится график за выбранный день месяца. После этого абонент может посмотреть детальную статистику, щелкнув указателем мыши на один из столбцов, означающих час дня. После выполнения этого действия абоненту будет предложено сохранить детальную статистику за выбранный час дня в файле Nadmin.

Детальная статистика требуется для ответа на вопрос, какое количество трафика с каких сайтов было потреблено.

Абонент также может посмотреть состояние своего лицевого счета за предыдущие расчетные периоды. Для этого нужно кликнуть по ссылке “Архив лицевого счета”.

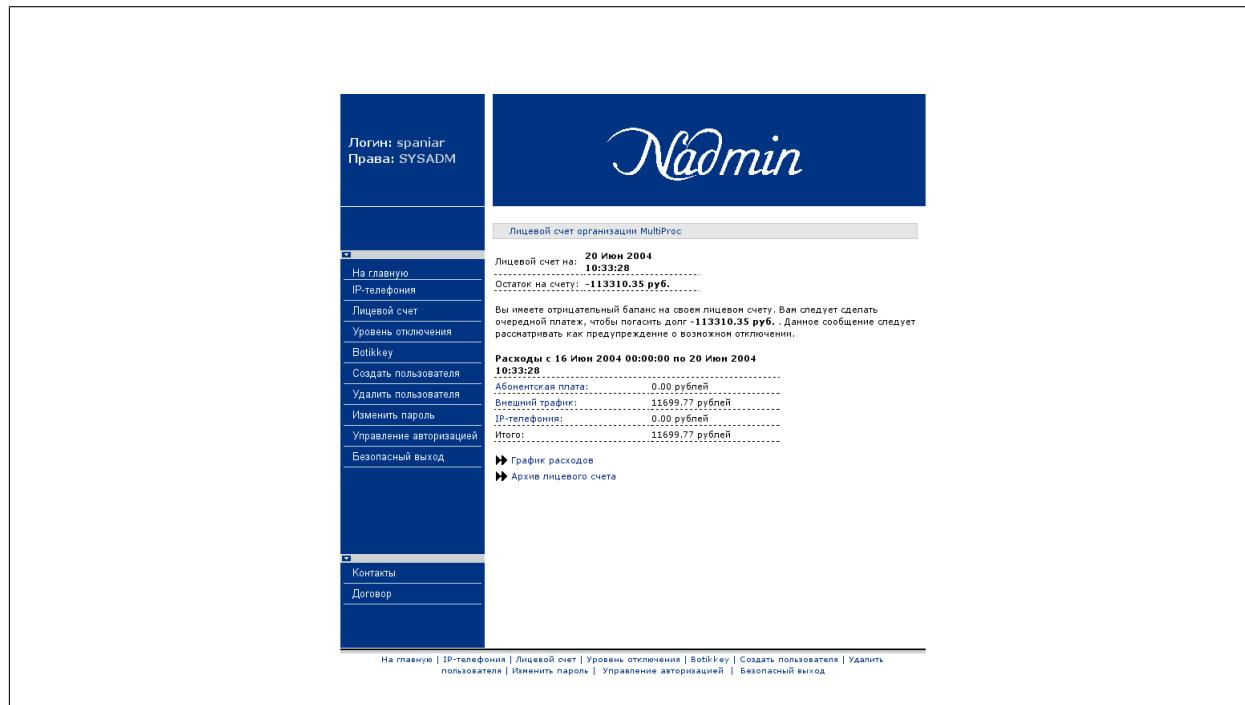


Рис. 5.17: Web-интерфейса абонента: лицевой счет

Уровень отключения. Страница “Уровень отключения” позволяет абонентам устанавливать уровень отключения. Это означает, что отключение абонента будет осуществляться автоматически при достижении установленного абонентом уровня отключения (см. раздел А.2.1.3).

На странице абоненту показана сумма, ниже которой уровень отключения не может быть установлен.

BotikKey. Страница “Botikkey” позволяет абоненту создавать, удалять, изменять пароли системы BotikKey. Данная система дает возможность абонентам отключаться/подключаться к внешней/локальной сети, обеспечивает дополнительную безопасность работы абонентов в сети¹⁹.

Создать пользователя. Страница “Создать пользователя” доступна только абоненту-администратору. На ней абонент-администратор может создавать пользователей web-интерфейса.

Удалить пользователя. Страница “Удалить пользователя” доступна только абоненту-администратору. На ней абонент-администратор может удалять пользователей web-интерфейса.

¹⁹http://www.botik.ru/_botik/tools/

Изменить пароль. Страница “Изменить пароль” доступна только абоненту-администратору. На ней абонент-администратор может изменять пароль пользователей web-интерфейса.

Управление авторизацией. На странице “Управление авторизацией” абонент может установить время жизни cookie. Данный параметр определяет период времени, в течение которого абонент может заходить на web-интерфейс без повторного ввода пароля.

Безопасный выход. Эта ссылка обеспечивает безопасный выход с web-интерфейса. После этого действия абоненту необходимо снова ввести все данные (имя организации, логин и пароль) для доступа к web-интерфейсу.

Контакты. На данной странице абонент может написать письмо техническому персоналу.

Договор. На этой странице абонент может посмотреть данные контракта, заключенного с СТ “Ботик”.

Глава 6

Специализированная гео-информационная система **BotikMap**

6.1 Общее описание

Сегодня для управления СТ “Ботик” г. Переславля-Залесского используется административно-информационная система Nadmin. Эта система обладает свойством расширяемости, позволяющей добавлять к ней различные компоненты. В их число входит подсистема MON, которая занимается мониторингом состояния сетевых узлов СТ “Ботик”. Она осуществляет проверку работоспособности узлов, и может предоставить системе Nadmin информацию о любых неисправностях.

Обслуживающему персоналу СТ “Ботик” поступает поток сообщений от пользователей или от MON, например в виде писем, в которых сообщается о проблемах или неполадках в сети. На момент принятия решения о проведении ремонтных работ или дополнительной проверки имеется регистрационная информация: имя устройства и (возможно) учётное имя пользователя, но отсутствует географическая информация. Тем самым, мы не можем быстро определить в каком месте это произошло и куда выезжать ремонтной бригаде.

ГИС сможет предоставить географическую информацию об объектах, что позволит ремонтной бригаде своевременно отреагировать на вызов. Система будет полезна не только для обслуживающего персонала, но и для рядовых абонентов городской сети.

Абонентам можно предоставить режим доступа, при котором не раскрываются технические детали, но отображается следующая информация: какая часть СТ “Ботик” работает в штатном режиме, а какая находится в аварийном состоянии или в состоянии

ремонта. Это повысит открытость системы. Абонент будет обладать всей возможной информацией и не будет зря обращаться к персоналу СТ “Ботик”.

Разработан другой подход к ГИС, при котором:

1. на клиентской стороне делается достаточно много вычислений (что даёт малую нагрузку на сервер);
2. передача информации между клиентом и сервером выполняется не в пикельном, а в векторном формате (что также разгружает сервер).

Наша разработка ГИС делится на две основные части: клиентская и серверная. Эти части обмениваются информацией между собой и с другими системами, например, с системой Nadmin — системой администрирования и управления сетью. В состав Nadmin входит также мониторная подсистема (MON), которая осуществляет мониторинг состояний (работоспособен, неработоспособен и т. п.) узлов в сети.

Требования к программному обеспечению: клиент: ОС Windows; сервер: ОС Linux. В скором времени будет доступна версия клиента под Linux.

6.2 Руководство системного программиста

6.2.1 Сервер

Серверная часть ГИС содержит базу данных для хранения географической информации, векторной карты г. Переславля-Залесского, сетевых объектов, а также средства, позволяющие добавлять и изменять информацию в этой базе данных.

Серверная подсистема ГИС реализована в виде CGI-программы на языке Perl. Эта подсистема в свою очередь делится на несколько подсистем:

1. подсистема хранения: обеспечивает хранение в базе данных ГИС информации, введённой пользователями или работниками СТ “Ботик”. Изменять информацию в этой подсистеме могут также другие компоненты программного обеспечения. Например, подсистема MON может менять состояния (работоспособен, неработоспособен и т. п.) различных объектов.
2. подсистема разграничения прав доступа: контролирует права пользователей. Существует три типа пользователей: анонимный, простой и привилегированный. Каждому пользователю определён его уровень доступа, что подразумевает доступность тех или других операций для данного пользователя. Анонимному клиенту разрешено просмотр объектов БД (кроме слоёв “Оборудование” и “Каналы”).

Простой пользователь наследует права анонимного пользователя плюс права на изменение объектов БД. Привилегированный пользователь имеет право на просмотр и изменение всех объектов. В системе Nadmin уже реализованы механизмы авторизации в рамках Web-интерфейса пользователя и администратора к системе Nadmin. Поэтому разумно использовать в нашем проекте уже существующие механизмы авторизации, а не реализовывать свои.

3. подсистема взаимодействия с клиентом и с другими компонентами административной системы Nadmin.

Рассмотрим северную подсистему более детально. Серверная часть ГИС состоит из двух частей:

- системы хранения ГИС-объектов — это один из классов базы данных Nadmin, описанный модулем Nadmin::GisObject.pm. Также, на сегодняшний день кроме вышеупомянутой БД используется временная БД MySQL;
- сервера — специализированное Web-приложение gisobject.cgi.

6.2.2 Nadmin::GisObject.pm

Nadmin::GisObject.pm описывает ГИС-объект и методы работы с этим объектом. Nadmin::GisObject.pm наследуется от Nadmin::DBfile.

Каждому экземпляру объекта соответствует rcs-файл, который состоит из строк вида [ключ]=<значение>, где ключ — это наименование свойств ГИС-объекта, а значение — его содержательная часть.

Свойства ГИС-объекта:

- ID — уникальный номер ГИС-объекта;
- LAYER — имя категории объекта;
- SUBLAYER — имя подкатегории объекта;
- TYPE — тип;
- STATE — состояние;
- NAME — имя ГИС-объекта;
- ADDRESS — адрес;
- COMMENT — комментарий;

- BOXES — области размещения объекта;
- COORDS — координаты ГИС-объекта;
- OTHER_PARAMS — другие параметры, связанные с отображением объекта.

Методы ГИС-объекта:

- *new* — функция возвращает ГИС-объект.

Применение:

$$\text{my } \$\text{gisobject} = \text{newNadmin} :: \text{GisObject}(\$ID_gisobject)$$

$\$ID_gisobject$ — уникальный номер ГИС-объекта.

- *create* — функция позволяет создать новый ГИС-объект.

Применение:

$$\text{my } \$\text{newgisobject} = \text{Nadmin} :: \text{GisObject} -> \text{create}(\%args)$$

Хэш $\%args$ имеет элементы с ключами — свойствами ГИС-объектов. Обязательно существование элемента хэша с ключом ID.

- *list* — функция возвращает список ГИС-объектов.

Применение:

$$\text{my } @\text{list} = \text{Nadmin} :: \text{GisObject} -> \text{list}(\%args)$$

Хэш $\%args$ имеет элементы с ключами — свойствами ГИС-объектов. Если хэш не задан, строится список всех ГИС-объектов.

6.2.3 Временная БД

Временная БД хранит ГИС-объекты созданные/изменённые простым пользователем, которые хранятся там, пока не пройдут процедуру цензурирования.

Временная БД состоит из двух таблиц: GO и U_INFO.

Field	Type	Null	Key	Default	Extra
ID_MGO	int(11)		PRI	NULL	auto_increment
ID	int(11)			0	
LAYER	blob				
SUBLAYER	blob				
TYPE	blob				
STATE	blob				
NAME	blob				
ADDRESS	blob				
COMMENT	blob				
BOXES	blob				
COORDS	blob				
OTHER_PARAMS	blob				
LOCK_GO	int(11)			0	
TIME_START_GO	int(11)			0	
AUTHOR	varchar(20)			anonymous	

Таблица 6.1: Временная база данных ГИС BotikMap. Таблица GO

Field	Type	Null	Key	Default	Extra
USER_NAME	char(20)		PRI		
STATUS	int(11)			1	
PENALTY	int(11)			0	
REWARD	int(11)			5	

Таблица 6.2: Временная база данных ГИС BotikMap. Таблица U_INFO

6.2.4 Web-приложение giserver.cgi

Сервер обеспечивает обмен данными между клиентом и БД. Общение между клиентом и сервером происходит в среде https. Клиент делает запрос (обращение к giserver.cgi как к Web-приложению с параметрами), сервер отвечает. С основной БД сервер взаимодействует посредством методов ГИС-объекта, описанными выше, а с временной — с помощью набора функций, описанных в giserver.cgi. Функции для работы с временной БД:

- db_open — функция открывает соединение с базой данных, вызывая для этого

метод connect.

- db_close — функция предназначена для закрытия соединения с базой данных.
- db_error — функция обрабатывает ошибочные ситуации, возникшие при обращении к базе данных средствами интерфейса DBI.
- db_sql_do — функция позволяет выполнить команды SQL без предварительной подготовки
- db_sql_prepare — функция выполняет предварительную подготовку команд SQL, вызывая для этого метод prepare.
- db_sql_execute — функция выполняет предварительно подготовленную команду.
- db_sql_fetchrow_array — функция помогает извлечь результаты выполнения команды SELECT.

Главная функция giserver.cgi начинается с авторизации/проверки пользователя, тем самым это отсекает использование сервера неавторизованным пользователем. Методы авторизации используются те же, что и в интерфейсе администратора/пользователя к системе Nadmin. Далее, в зависимости от предыдущего шага, либо выдаётся ошибка авторизации, либо происходит проверка параметров (с которыми обратился клиент) с запуском соответствующего обработчика. Возможные параметры:

- dsize — возвращает размер БД и саму БД;
- newid — возвращает новый уникальный индекс;
- creobj — создаёт/обновляет ГИС-объект в БД;
- tmp — возвращает ГИС-объект из временной БД;
- delobj — удаляет ГИС-объект из основной БД;
- deltmp — функция отрицательного отзыва о цензурированном ГИС-объекте. Удаляет объект из временной базы и начисляет штрафное очко автору;
- modobj — функция положительного отзыва о цензурированном ГИС-объекте. Удаляет объект из временной базы, заносит его в основную базу и начисляет премиальное очко автору.

6.2.5 Клиент

Клиентская часть ГИС реализована в виде программы *BotikMap* на языке TCL/TK. В этой программе можно просматривать и редактировать объекты геоинформационной системы г. Переславля-Залесского. Все объекты ГИС разделены на категории, которые называются слоями. К числу слоёв относятся, например: “Оборудование”, “Каналы”, “Объекты”, “Дома”, и др. Среди слоёв задан порядок их отрисовки по “высоте”¹ при формировании изображения карты: кварталы расположены ниже, чем водоёмы, водоёмы лежат ниже, чем мосты, а слой “оборудование” располагается выше всех остальных. Это очень важно для правильного визуального понимания картографического материала.

Для каждого слоя определены свои подкатегории объектов, которые допустимы в этом слое. Например, слою “Водоёмы” соответствуют подкатегории: “Озеро”, “Название Водоёма”, “Река”, “Ручей”. Выбор возможных объектов для создания зависит от выбранного слоя. В программе BotikMap есть возможность переключения на тот или иной слой и создания объектов, соответствующих этому слою.

После запуска программы и успешного соединения с сервером пользователю будет предложено ввести свои регистрационные данные, либо зарегистрироваться на сервере. В зависимости от прав, которыми обладает пользователь, сервер отправит ему соответствующие объекты карты г. Переславля.

В качестве среды для взаимодействия клиент-сервер выступает [https](https://), поскольку такое решение позволяет использовать Apache как платформу для реализации ГИС-сервера. Это позволит не тратить ресурсы на создание мультисессионного сервера и повысит защищённость ГИС-сервера от внешних атак. Системой хранения информации будет единая БД Nadmin. Тем самым, мы решим проблему синхронизации данных и взаимодействия ГИС СТ “Ботик” с другими компонентами системы Nadmin.

6.2.6 Взаимодействие Nadmin и ГИС

При изменениях или добавлениях объектов на карте должна быть обеспечена своевременная передача информации между серверной частью ГИС и Nadmin. Например: если в Nadmin появилась проектируемая линия, Nadmin должен выдать информацию о её появлении для ГИС. Также при появлении проектируемых объектов в Nadmin должна производиться нумерация, следовательно, нужно из ГИС передать в Nadmin уникальный номер нового объекта.

¹Это общее понятие в системах, связанных с построением многослойных изображений. Речь идёт об очерёдности отрисовки слоёв.

6.3 Руководство администратора гражданской сети

В нашем проекте на администратора наложены три функции:

- нанесение на карту оборудования и каналов;
- добавление привилегированных пользователей — данная функция осуществляется через Web-интерфейс администратора/пользователя к системе Nadmin;
- цензурирование ГИС-объектов, добавленных/изменённых простыми пользователями.

6.4 Руководство пользователя

6.4.1 Создание объектов

Все объекты ГИС можно разделить по способу их создания на следующие классы:

1. линейные;
2. площадные;
3. текстовые;
4. точечные;
5. текстово-линейные;
6. линейно-площадные.

Рассмотрим более подробно способы создания объектов каждого класса.

1. Линейным объектом может быть как обычная, так и сглаженная ломаная линия. Такие объекты рисуются следующим образом: щелчок левой кнопкой мыши добавляет узел у линии, щелчок правой кнопкой также добавляет узел и завершает процесс создания объекта. Количество узлов может быть произвольным. Примеры объектов этого класса: канал связи, бордюр, ручей.
2. Площадной объект — это, по сути, замкнутая ломаная линия с заливкой. Рёбрами этого объекта могут быть как прямые, так и сглаженные линии. Создание площадных объектов происходит так же, как и создание линейных. Отличие лишь в том, что рисуется не линия, а замкнутый многоугольник. Примеры объектов этого класса: дом, квартал, вал, лес, кустарник, озеро, река.

3. Текстовые объекты являются обычным горизонтальным текстом, которые рисуются следующим образом: сначала левым щелчком мыши выбирается точка привязки текста, затем в появившемся окне вводится сам текст и указывается его расположение относительно точки привязки: сверху, снизу, справа, слева, по центру и т. д. После нажатия клавиши “Enter” или кнопки “OK” в выбранной точке с заданным расположением создаётся введённый текст. Примеры объектов этого класса: номер дома, название квартала.
4. Точечные объекты рисуются одним левым щелчком мыши. Пример объекта этого класса: ящик с телекоммуникационным оборудованием.
5. Текстово-линейные объекты состоят из текста, нанесённого на опорную ломаную. Ломаная рисуется так же, как и обычная линия, но по окончании её создания нужно будет ввести текст. При нажатии на клавишу “Enter” или кнопку “OK” этот текст будет расположен вдоль ломаной. Примеры объектов этого класса: название улицы и название водоёма.
6. Линейно-площадные объекты — это объекты, которые состоят из многоугольников и линий. К таким объектам относится объект “Мост”. Он рисуется путём выбора восьми точек (a b c d e f g h), которые определяют:
 - четырёхугольник (“b c f g”),
 - две линии (“a b c d” и “e f g h”).

6.4.2 Основы работы в BotikMap

6.4.2.1 Запуск

После запуска программы и успешного соединения с сервером пользователю будет предложено ввести свои регистрационные данные, либо зарегистрироваться на сервере. В зависимости от прав, которыми обладает пользователь, сервер отправит ему соответствующие объекты карты города Переславля-За/лесского.

6.4.2.2 Панели инструментов

Кнопки на верхней панели инструментов используются для выполнения следующих функций:

- *Выход из программы*: завершение работы BotikMap. Если во время работы были сделаны какие-то изменения, то будет предложено их сохранить.

- *Чтение из файла:* загрузка объектов ГИС из файла, который должен иметь специальный формат и расширение “.gis” или “.txt”.
- *Сохранение в файл:* сохранение всех изменений в файл с расширением “.gis” или “.txt”. При этом всё его старое содержимое стирается и он записывается заново.
- *Вызов справки о программе*
- *Менеджер слоёв* В нём можно выбрать верхний (рабочий) слой, изменить видимость (видимый/невидимый), и редактируемость (редактируемый/нередактируемый) слоёв. С помощью кнопок “Выбрать всё” и “Убрать всё” в окне менеджера можно изменить видимость и редактируемость сразу всех слоёв.
- *Поиск:* позволяет по заданным критериям искать объекты на карте. В качестве критериев для поиска можно указать слои, подкатегории объектов, типы, состояния, а также фрагменты имени, адреса или комментария. Набор доступных объектов зависит от выбранных слоёв, и набор доступных типов и состояний — от выбранных объектов. Имеется 2 варианта поиска: обычный (кнопка “Найти”) и радиусный (кнопка “Найти ближайшие к точке”). Если выбран 2-й вариант, то нужно будет указать точку карты, вблизи которой нужно найти объекты. Найденные объекты в этом случае будут отсортированы по возрастанию в порядке их дальности от выбранной точки. Как в первом, так и при втором режиме поиска, поиск заканчивается выдачей списка найденных объектов. После нажатия левой кнопкой мыши на какой-либо строке этого списка, окно просмотра карты автоматически переместится в область выбранного объекта. Если объект слишком большой и не помещается на экране, то масштаб будет изменён так, чтобы объект полностью уместился на экране. Мерцающая стрелка укажет расположение объекта. Как обычно, при поиске используются ограничения на количество найденных объектов: если найдено слишком много объектов, то пользователю будет предложено уточнить условия поиска.
- *Выбор масштаба:* эта кнопка-меню позволяет выбрать масштаб и является альтернативой инструменту “Изменение масштаба” на левой панели инструментов (см. ниже).
- *Выбор слоя:* эта кнопка-меню позволяет выбрать верхний (рабочий) слой, не используя при этом Менеджер слоёв (см. выше).
- *Отображение подложки:* в BotikMap есть возможность подгружать отдельные части карты Переславля. Они представляют собой изображения в пиксельном

формате. Включить или отключить их отображение можно с помощью кнопки “Ото-бра-зить/Скрыть подложку” на верхней панели инструментов. Каждый раз будет показываться только та картинка, которая соответствует текущему положению на карте. При каждом зуммировании или перемещении по карте изображение будет меняться.

- *Переход в точку*: альтернатива инструменту “Перемещение в точку” (см. ниже). После нажатия на кнопку “Перейти в точку” будет предложено ввести точные координаты точки (X и Y, в метрах), в которую нужно перенести центр экрана.
- *Выпрямление углов*: данная операция по возможности делает прямыми углы у выделенной фигуры, стараясь минимально изменить её внешний вид. Если выделенный объект — линейный или площадной, то к нему можно применить операцию выпрямления углов (кнопка “Выпрямить углы” на верхней панели инструментов).
- *Отмена выпрямления углов*: эта операция используется в том случае, если выпрямление углов привело к нежелаемому для пользователю результату.

Кнопки левой панели инструментов помогают создавать и редактировать объекты ГИС.

- *Выделение объекта*: позволяет выделить один или несколько объектов. Выделение осуществляется несколькими способами:
 - щелчком на объекте левой кнопкой мыши (выделяется только один объект);
 - удержанием клавиши “Ctrl” и щелчками левой кнопкой мыши на объектах (каждый объект добавляется к группе ранее выбранных объектов или удаляется из неё);
 - удержанием левой клавиши мыши на свободном месте карты (начальное положение) или подложке, и перемещением указателя мыши — появится пунктирный прямоугольник, охватывающий экран от начального положения до текущего положения мыши; выделенными становятся только те объекты, которые пересекаются с данным прямоугольником или касаются его.

В выделенном состоянии у объекта видны его узлы, которые можно передвигать, удалять или добавлять новые (об этом речь пойдёт ниже). Сам объект можно перемещать как с помощью указателя мыши, так и с помощью клавиш со стрелками (перемещение с небольшим фиксированным шагом), а также их комбинацией с

клавишей “Ctrl” (перемещение с более крупным фиксированным шагом). Нажатие правой кнопкой мыши на объекте открывает окно его свойств, где можно изменить его тип, состояние, имя, адрес или комментарий.

- *Изменение масштаба:* этот инструмент изменяет масштаб просмотра карты (зум). Щелчок левой кнопкой в каком-либо месте карты делает увеличение, а правой — уменьшение масштаба относительно выбранной точки (она становится центральной точкой экрана). Всего существует 7 уровней масштабирования: 1-й самый мелкий, 7-й — самый крупный. Выбрать масштаб можно также с помощью меню “Выбор масштаба” на верхней панели инструментов;
- *Перемещение в точку:* позволяет перенести центр экрана в указанную точку. Это также возможно с помощью явного указания координат (в метрах) в окне “Переход в точку” (оно вызывается кнопкой “Перейти в точку” на верхней панели инструментов);
- *Добавление/Удаление узлов:* позволяет нам добавлять новые или удалять старые узлы у площадных или линейных объектов. Чтобы создать узел на выбранном ребре, нужно щёлкнуть левой клавишей мыши на начальном узле этого ребра. При этом новый узел создаётся ровно посередине между начальным и конечным узлами выбранного ребра. Для удаления узла нужно щёлкнуть на нём правой клавишей мыши;
- *Изменение сглаживания:* этот инструмент позволяет включить или отключить сглаживание углов у выделенных линейных или площадных объектов. Сглаживание осуществляется с помощью кривых Безье;
- *Поднятие объекта:* этот инструмент позволяет вынести выделенный объект на передний план в своём слое. Объект будет казаться выше, чем все остальные объекты его слоя;
- *Опускание объекта:* этот инструмент позволяет вынести выделенный объект на задний план в своём слое. Объект будет казаться ниже, чем все остальные объекты его слоя.

Все остальные кнопки на левой панели инструментов отвечают за создание объектов ГИС. Состав этих кнопок определяется тем, какой слой выбран в качестве рабочего.

Глава 7

Система мониторинга МОН/Botik для региональных компьютерных систем

7.1 Мониторная система МОН/Botik. Описание программы

Система мониторинга технического состояния сети МОН используется СТ “Ботик” для наблюдения за состоянием городской компьютерной сети¹. Мониторинг сети системой МОН осуществлялся достаточно продолжительный промежуток времени, что позволило оценить работу мониторной системы и заметить некоторые ее недостатки, а именно:

1. недостаточная тщательность проверки оборудования либо сервиса, что приводило к ложным срабатываниям системы оповещения, либо пропуск серьезных поломок в сети;
2. избыточность сообщений об одной и той же поломке — это ослабляет бдительность обслуживающего персонала, особенно если поломка незначительная и легко устранимая;
3. неудобный формат письма о проблеме в сети — необходимо просмотреть большое количество второстепенной информации, прежде чем будут предоставлены основные данные о результатах проверки;
4. неинформативный Web-интерфейс, содержащий очень мало детальной информации о ситуации в сети.

¹ В данной работе под словами “городская компьютерная сеть” понимается система телекоммуникаций (СТ) “Ботик” города Переславля-Залесского.

Кроме того, систему MON невозможно использовать для проверки целостности каналов связи. При наличии в сети резервных линий ping-тест, который используется для проверки работоспособности маршрутизаторов, не дает достоверной информации о строении и неизменности графа коннективности.

Таким образом, были сформулированы требования к мониторной системе, которые позволили бы сделать процесс наблюдения за работой сети более качественным и удобным. Так как система MON является свободно-распространяемым продуктом, есть возможность внести в нее свои модификации и удовлетворить этим требованиям. К указанным требованиям прежде всего относятся:

1. повышение качества проверок — мониторная система должна уметь отличать серьезную поломку от незначительного сбоя в работе сети с последующим самовосстановлением;
2. оповещение о поломке в сети, а также оповещение о восстановлении работоспособности оборудования (сервиса), даже если восстановились не все компьютеры в группе, а только часть из них;
3. изменение политики оповещения — создание системы оповещения с “обратной связью”;
4. представление результатов проверок в Web по всем параметрам проверки и в любой момент времени;
5. создание дополнительных мониторных программ, которые могут быть использованы системой MON, для наблюдения за неизменностью графа коннективности для проверки сети на наличие разрывов на линиях.

Модифицированная версия системы MON должна удовлетворять всем вышеперечисленным требованиям. Целью настоящей работы была разработка такой модификации систем MON, которая осуществляет более тщательную проверку сервисов, сообщает персоналу более точную информацию о ситуации в сети и отображает результаты проверок в Web.

Полная документация по системе мониторинга MON доступна в Интернете по адресу <http://www.kernel.org/software/mon>. В данном документе описаны только те возможности MON, которые были добавлены по требованиям СТ “Ботик”.

7.1.1 Ключевые слова и термины

1. **Демон (daemon, основной процесс).** Управление работой всей системы в целом ведется отдельной большой программой-демоном, которая обрабатывает значения, полученные в результате работы отдельных частей системы, а также принимает решения о дальнейших действиях системы.
2. **Сервис (service).** Система мониторинга MON способна проверять доступность и работоспособность компьютеров в сети, а также некоторых сервисов, предоставляемых компьютерами. К таким сервисам относятся FTP, HTTP, SMTP, наличие свободного пространства на дисках, дата последней модификации файлов и многое другое.
3. **Монитор (monitor).** Для проверки одного сервиса используется отдельная программа-монитор. Данная программа запускается демоном, результаты ее работы также возвращаются демону.
4. **Алерт (alert).** Программы-алерты используются для проведения оповещения о ситуации в сети. Вызов алertов производится основным процессом по результатам работы мониторов.
5. **Web-сайт системы.** Для отображения в Web состояния сервисов сети и доступности проверяемых компьютеров используется Web-сайт системы MON. Сайт предоставляет информацию о состоянии сервисов в сети в графическом виде, в удобном для восприятия человеком.

7.1.2 Архитектура системы MON. Компоненты системы

Система MON состоит из следующих компонентов:

1. конфигурационные файлы системы MON;
2. daemon (демон) — основная часть системы, координирующая действия всех остальных ее частей;
3. набор мониторов — программ, производящих проверку сервисов и оборудования, каждый монитор может производить проверку только одного сервиса;
4. набор программ, осуществляющий оповещение;
5. журнал системы.

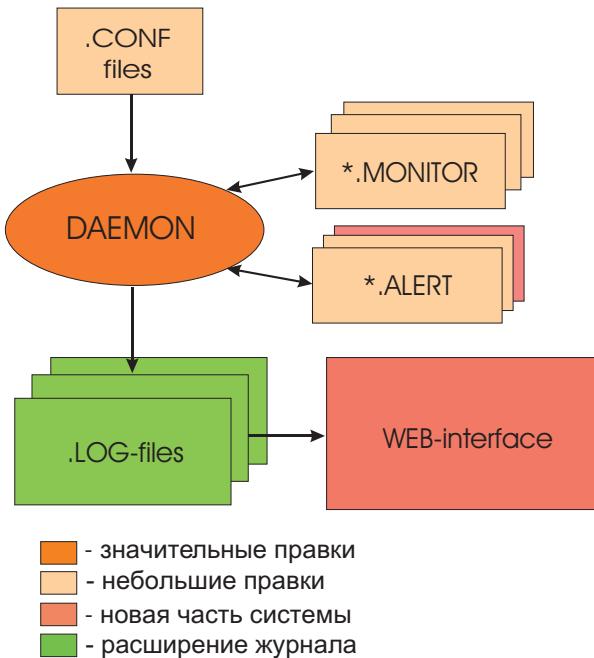


Рис. 7.1: Архитектура системы MON

Перед началом эксплуатации системы необходимо сформировать для нее конфигурационные файлы. В них указываются параметры проверок, а именно:

1. группы компьютеров (маршрутизаторов, серверов), которые должны подвергаться проверке;
2. сервисы, которые должны проверяться на указанных группах компьютеров и, если необходимо, специальные параметры проверок;
3. время проведения проверок и интервалы между проверками;
4. методы оповещения, частота посылки сообщений, адреса.

Пример конфигурационного файла показан на рисунке 7.6

7.1.3 Процесс работы системы MON

После того, как конфигурационные файлы будут написаны, можно запускать систему. Основной процесс (daemon, демон) после запуска будет работать до тех пор, пока не будет остановлен специальной linux-командой. С интервалом, определенным в конфигурационном файле, основной процесс запускает программы-мониторы для проверки

работоспособности сети. Вся информация о мониторах и проверяемых компьютерах читается один раз из конфигурационного файла и впоследствии используется демоном для запуска соответствующих мониторов.

После того, как монитор выполнил свою проверку, он возвращает основному процессу некоторое число, соответствующее результатам работы монитора. Например, если проверка показала, что все проверяемые машины работоспособны, монитор возвращает 0, в противном случае возвращаемое значение будет равно 1. Кроме числового значения основной процесс получает от монитора информацию о ходе и результатах проверки, представленную в текстовом виде. Такая информация передается основному процессу в случае если где-то найдена неполадка.

Далее основной процесс решает, нужно ли послать сообщение о поломке или оно уже было отправлено. Демон следит за количеством сообщений — оно не должно превышать указанное в конфигурационном файле. Если сообщение должно быть послано, демон вызывает соответствующую программу оповещения, передает ей информацию о получателе и текст с описанием проблемы, полученный от монитора. Программа оповещения принимает переданные ей аргументы и рассыпает сообщения.

Демон может также вызвать программу оповещения, если необходимо сообщить персоналу о восстановлении ранее неработоспособного сервиса. Это делается только если соответствующая строка (**UPALERT**) присутствует в конфигурационном файле. Сообщение о восстановлении будет отправлено только в том случае, если все компьютеры в группе работоспособны (то есть демон получил значение 0 от монитора).

По истечении интервала времени, указанного в конфигурационном файле, основной процесс снова запускает монитор и процесс повторяется.

В системе MON имеется более сорока мониторных программ. Наиболее востребованными для СТ “Ботик” являются: **fping.monitor**, проверяющий доступность компьютеров по протоколу ICMP, **ftp.monitor**, **http.monitor**, **smtp.monitor**, **pop3.monitor**, **imap.monitor**, и некоторые другие, проверяющие работоспособность соответствующих сервисов. С помощью монитора **freespace.monitor** MON предоставляет возможность отслеживать количество свободного места на диске. Некоторые мониторные программы, например, **reboot.monitor** и **process.monitor**, работают с использованием протокола SNMP.

Система MON имеет такую архитектуру, что при необходимости можно создать собственный монитор. Важно, чтобы написанный нами монитор передавал основному процессу значения, которые демон может корректно обработать. Это позволяет расширять возможности системы MON в тех направлениях, которые для нас особенно важны.

Результаты проверок и действия основного процесса фиксируются в журнале систе-

мы. Также в журнал записываются ошибки, произошедшие в процессе работы системы MON.

7.2 Руководство системного программиста

Для того, чтобы сделать процесс наблюдения за работой сети более качественным и удобным, было принято решение внести изменения в систему мониторинга MON. Результатом этих модификаций прежде всего являются:

1. оповещение в случае восстановления части компьютеров проверяемой группы;
2. возможность построения Web-сайта состояния сети на основе данных, полученных от мониторов в ходе проверок;
3. изменение параметров проверок для повышения точности результата.

7.2.1 Модификации основного процесса системы MON

7.2.1.1 Изменения функций proc_cleanup() и do_alert()

Основные изменения главного процесса мониторной системы MON затронули процедуру, занимающуюся обработкой значений, полученных от мониторных программ proc_cleanup. Именно в ней принимается решение о посылке сообщения о некотором событии в сети.

По окончании работы мониторная программа возвращает демону некоторое числовое значение, которое определяет результат проверки сервиса. Если возвращенное значение равно нулю, проверка прошла успешно, поломок нет. В противном случае демон должен оповестить о поломке. При получении результата работы монитора демон также фиксирует результат проверки в виде некоторой текстовой строки, в которой перечислены машины, непропущенные проверку. В новой версии системы MON происходит сравнение строк предыдущей проверки и той, которая только что завершилась.

Сравнение происходит с помощью функции

```
find_diff($summary, $last_summary),
```

где \$summary — строка с названиями компьютеров, непропущенных последнюю проверку, \$last_summary — строка с названиями компьютеров, непропущенных предпоследнюю проверку.

Функция `find_diff()` возвращает два значения: `$flag` и `diff`. `$flag` может принимать значения 0 или 1. Если `$flag` равен 0, то во время проверки обнаружены восстановившиеся компьютеры/сервисы. В случае когда `$flag` равен 1 монитором были обнаружены дополнительные сломавшиеся компьютеры/сервисы, либо ситуация не изменилась.

В зависимости от результата работы `find_diff()` вызывается функция `do_alert()` с аргументом `$FL_ADDIT_MONITOR` (если нет восстановившихся сервисов в группе или в случае обнаружения дополнительных поломок), либо `$FL_RESTORED` (при обнаружении некоторых восстановившихся компьютеров/сервисов в группе).

При получении флага `$FL_RESTORED` функция `do_alert()` проделывает следующее:

1. следит за тем, чтобы алERTы о восстановлении не посыпались, если предварительно не было отправлено сообщение о поломке;
2. следит, чтобы не было отправлено дублирующих сообщений.

С расширением возможностей подсистемы оповещения возникла необходимость увеличить количество аргументов, передаваемых функции `do_alert()`. Дополнительно к имеющимся аргументам необходимо передавать `id` поломки, который представляет собой время обнаружения поломки. Далее `id` поломки передается функции, отправляющей алERT, `call_alert()`. Для функции `call_alert()` этот аргумент также является новым, в отличие от оригинальной версии системы МОН.

7.2.2 Дополнительные модули для построения сайта состояния сети

7.2.2.1 Функции `all_stat()` и `write_log()`

Для генерирования данных, используемых в дальнейшем при построении сайта состояния сети, используется функция `all_stat()`. Если при запуске мониторной системы указана опция `-w`, данные для построения сайта будут генерироваться, иначе подобных действий происходить не будет.

При получении опции `-w` происходит вызов функции `minimal_interval()`. Данная функция не получает никаких аргументов, так как вся необходимая ей для работы информация имеется в глобальном хэше `%watch`. Функция возвращает некоторое число, которое является минимальным интервалом между проверками того или иного сервиса среди всех проверяемых групп. Данная процедура необходима для определения частоты запуска функции `all_stat()`.

После проведения очередной проверки сервисов запускается функции `all_stat()`, в случае если указана опция `-w` и интервал времени между предыдущим ее запуском и настоящим моментом больше минимального интервала между проверками сервисов.

Модуль `All_Stat.pm` содержит функцию `all_stat()`. Функция не получает аргументов, но использует хэши `%watch` и `%groups`, объявленные в демоне как `our %watch` и `our %groups`.

`all_stat()` проделывает следующее:

1. читает информацию о проверяемых группах и сервисах в глобальных хэшах и строит из них свою структуру;
2. производит разбор log-файлов, содержащих данные о результатах проверок;
3. строит массив из полученных данных в формате, приемлемом для JavaScript-программы и записывает его в файл с расширением `.js`.

В результате работы модуля `All_Stat.pm` имеется `.js`-файл, в котором записаны данные о состоянии сети в определенном формате. В дальнейшем этот файл используется для построения картинки состояния сети в браузере.

Функция `write_log()` используется для формирования log-файлов, которые впоследствии разбираются функцией `all_stat()`. В качестве аргументов `write_log()` получает текстовый результат работы мониторной программы, названия проверяемой группы и сервиса. Парсер разбирает текстовую часть и записывает полученные данные в файл, название которого соответствует сервису, указывая название группы, к которой относятся данные.

7.2.2.2 Модуль построения Web-сайта состояния сети

Данные для построения сайта формируются модулем `All_Stat.pm`. Формат данных, которые подаются на вход `JavaScript`-программам для построения таблицы состояния сети в браузере, показан на рисунке 7.7. Этот формат соответствует формату массива в языке `JavaScript`.

В первой строке массива указаны названия всех проверяемых компьютеров данной группы и сервиса. Далее, первый элемент в последующих строках — время проведения проверки данного сервиса, числа в остальных столбцах соответствуют состоянию компьютера, указанного в этом же столбце. Ноль означает, что данный сервис на компьютере в этот момент времени был доступен, другое число — процент недоступности сервиса.

Обработка данных, показанных на рисунке 7.7, производится скриптом `new_main.js`. Для каждой группы и по каждому сервису вызывается функция `draw_table()`, которая получает в качестве аргументов массив данных о результатах проверок, а также дату того дня, когда проводились данные проверки.

Перебирая все элементы массива, `draw_table()` передает числовые значения функции `state_color()` для определения цвета, который необходимо отобразить на соответствующем месте таблицы. После вычисления цвета ячейка таблицы прорисовывается в браузере.

Данная процедура производится для каждого сервиса всех проверяемых групп компьютеров.

7.2.3 Система оповещения с обратной связью

В системе MON расширена функциональность подсистемы оповещения, а именно имеется система оповещения с обратной связью. Данная подсистема представляет собой механизм оповещения, при котором мониторная система способна реагировать на электронные письма администратора о той или иной поломке. В случае отсутствия подтверждения администратора, что ему известно о поломке, MON приводит в действие другую программу оповещения (по желанию администратора). Как только мониторная система получает письмо от администратора цепочка дополнительных сообщений заканчивается.

7.2.3.1 Обработка ответов системой MON

Для мониторной системы MON при установке системы создается специальный пользователь `mon`. По своим возможностям он эквивалентен обычному пользователю ОС Linux. В домашней директории пользователя `mon` создается файл `.forward`, в котором указывается название программы, обрабатывающей ответы администратора. Данный механизм позволяет эффективно производить обработку писем от администратора, при этом функциональность основного процесса не перегружается дополнительными задачами. Модуль обработки ответов находится также в домашней директории пользователя `mon` и называется `Check_Mail.pl`.

С введением новой политики оповещения каждой поломке присваивается некоторый идентификатор (ID), который представляет собой время поломки, выраженное в секундах от 1 января 1970г. При обнаружении поломки в домашней директории пользователя `mon` создается файл, названием которого является идентификатор поломки. Данное действие производится в функции демона `proc_cleanup()`. При получении под-

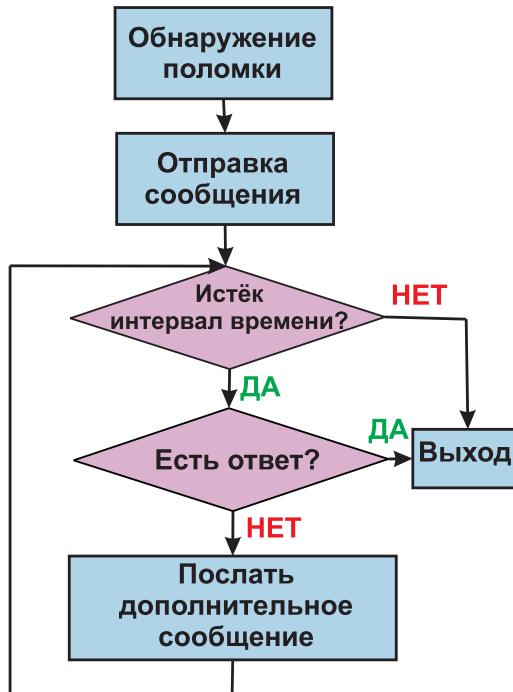


Рис. 7.2: Схема оповещения с обратной связью

тверждения на сообщение о поломке в соответствующий файл записывается о наличии подтверждения. При восстановлении сервиса файл, название которого соответствует идентификатору поломки, удаляется (функция `do_alert()`). Идентификатор поломки указывается в теле письма о проблеме, а также в поле `Subject`.

7.2.3.2 Принцип работы модуля `Check_Mail.pl`

Письмо, отправляемое администратором мониторной системе для подтверждения получения предыдущего оповещения, направляется программе `Check_Mail.pl`, путь к которой указан в файле `.forward`. Письмо подается на стандартный вход программе, складывается в массив, затем построчно разбирается с использованием регулярных выражений. В строке `Subject` выделяется поле, содержащее идентификатор поломки. Также ID поломки может быть взят из тела письма, если по каким-либо причинам его не оказалось в поле `Subject`. В случае если в письме не обнаружено идентификатора поломки, оно сохраняется в файл `errmsg`, который содержит все сообщения, не соответствующие принятому формату. Файл `errmsg` также находится в домашней директории пользователя `mon`.

```
period wd {Mon-Sun} hr {0-23}
    numalerts 1
    alert mail.alert root@host1.com
    upalert mail.alert root@host1.com
```

Рис. 7.3: Пример конструкции period

7.3 Руководство пользователя

7.3.1 Общее описание

Система MON представляет собой инструмент для наблюдения за доступностью сервисов в компьютерной сети и отправки сообщений при обнаружении неисправности. Архитектура системы позволяет использовать различные мониторные инструменты и способы оповещения, что достигается созданием единого интерфейса, который легко расширяется программами, написанными на языках Perl, C, shell и т.д.

7.3.2 Конфигурирование системы

Конфигурационный файл системы MON `mon.cf` находится в директории `/etc/mon/`. Для создания собственного конфигурационного файла можно воспользоваться файлом `example.cf`, который имеется в дистрибутиве, а также документацией к системе MON.

В конфигурационном файле помимо основных директорий, в которых располагаются составляющие системы, указываются названия компьютеров или устройств, которые впоследствии будут проверяться. Для каждой группы компьютеров (устройств) должны быть указаны сервисы, за которыми ведется наблюдение, мониторы и интервалы времени между проверками. Кроме этого в конфигурационном файле фиксируются способы оповещения, количество отправляемых сообщений и адресаты. Вследствие модификации системы мониторинга имеется возможность указать в конфигурационном файле дополнительные условия отправки сообщений.

Описание типа сообщений, частоты их отправки, интервала между сообщениями и адресатов выделено в отдельный блок конфигурационного файла. Пример описания отправки сообщения показан на рисунке 7.3.

В этой строке указывается период времени (день недели, время суток) в формате, принятом в модуле Perl 5 Патрика Райана (Patrick Ryan) `Time::Period`. (Более подробную информацию об этом модуле можно получить на странице `perldoc Time::Period`).

```
...
unacked 5h
    numalerts 1
    alert sms.alert 9101234567
    upalert sms.alert 9101234567
...
unacked 3h period wd {Tue-Sun} hr {0-23}
    numalerts 1
    alert mail.alert vera@up.botik.ru
    upalert mail.alert vera@up.botik.ru
...
period wd {Mon-Thu} hr {0-23}, wd {Mon Wed Fri} hr {8am-4pm} unacked 1m
    numalerts 1
    alert sms.alert 9102345678
    upalert sms.alert 9102345678
```

Рис. 7.4: Использование конструкции `unacked`

Перед отправкой сообщения о неисправности некоторого сервиса происходит проверка, возможно ли оповещение в данный момент времени. Кроме этого, имеется возможность использования конструкций, показанных на рисунке 7.4.

Если в конфигурационном файле присутствует запись `unacked`, предполагается использование системы оповещения с обратной связью. Запись `unacked 5h` означает, что перед посылкой дополнительного сообщения о неисправности система будет ожидать ответ администратора в течение пяти часов. Значения времени определяются как “30s”, “5m”, “1h”, “1d” и означают 30 секунд, 5 минут, 1 час и один день соответственно. Число может быть дробным, например, “1.5h”, что означает полтора часа. Остальные записи внутри конструкции `unacked` интерпретируются также, как и в конструкции `period` (см. man-страницу к системе MON).

Подробное описание принципов работы системы оповещения с обратной связью можно прочитать в главе 7.3.5.

Если в конфигурационном файле использованы одновременно конструкции `period` и `unacked`, при отправке сообщения происходит проверка, попадает ли текущий момент времени в интервал, указанный в `period`.

7.3.3 Параметры, используемые при запуске основного процесса системы

Параметры, используемые при запуске системы мониторинга MON, подробно описаны на тан-странице данной системы. Кроме этого, был введен еще один параметр **-w**, который необходимо указать в случае, если впоследствии планируется использовать возможность отображения состояния сети в WWW. Если система обнаружила этот параметр при запуске, происходит вычисление минимального интервала времени между проверками для всех групп и для всех сервисов. Предположим, в конфигурационном файле указано, что необходимо проводить проверку двух разных групп компьютеров (**group1** и **group2**), при этом у каждой группы проверяется по два сервиса (например, у группы **group1** проверяются доступность по ICMP и работоспособность FTP, а у группы **group2** – работоспособность SMTP и POP3). Искомый интервал времени между проверками будет тот промежуток, который является минимальным среди интервалов между проверками каждого из сервисов. В случае если доступность по ICMP проверяется один раз в час, проверка работоспособности FTP происходит один раз в 20 минут, а SMTP и POP3 тестируются каждые 10 минут, минимальный интервал будет равен 10 минутам. Данная величина необходима для того, чтобы было известно, с какой периодичностью необходимо осуществлять построение данных для их последующего отображения в Web.

Далее, при наличии параметра **-w** в определенный момент, а именно после проведения проверок указанных сервисов, происходит вызов модуля **All_Stat.pm**, который генерирует данные для последующего их использования программами построения сайта состояния сети. Вызов модуля **All_Stat.pm** происходит только в том случае, если интервал времени между предыдущим вызовом **All_Stat.pm** и текущим моментом времени больше минимального интервала между проверками одного из сервисов. Данная проверка позволяет осуществлять построение данных для сайта сети только в том случае, когда имеются новая информация о состоянии сети.

7.3.4 Использование мониторов

Способы использования мониторных программ подробно описаны на тан-странице системы мониторинга MON.

В результате адаптации системы MON для системы телекоммуникаций “Ботик” были добавлены некоторые дополнительные возможности, расширяющие функциональность мониторных программ. Далее перечислены параметры, которые можно использовать при эксплуатации монитора **fping.monitor** для проверки доступности компьютера.

1. **-c num** — количество пакетов, которые необходимо послать компьютеру для проверки его доступности. Если параметр не указан, количество посылаемых пакетов равно 10.
2. **-p msec** — время (в миллисекундах), которое программа ожидает перед посылкой очередного пакета при удачном прохождении предыдущего пакета. Если параметр не указан, интервал между посылками пакетов равен 1000 мсек.
3. **-b bytes** — размер посылаемых пакетов (в байтах). По умолчанию размер посылаемого пакета равен 56 байт.
4. **-l percents** — допустимое количество потерянных пакетов (в процентах). Если параметр не указан, допустимым количеством потерь считается 10% от количества отправленных пакетов.

7.3.5 Использование системы оповещения с обратной связью

Для того, чтобы стало возможным использование системы оповещения с обратной связью необходимо особым образом сконфигурировать систему. Подробное описание конфигурационного файла можно прочитать в главе 7.3.2.

Система с обратной связью работает следующим образом: мониторная программа обнаруживает неисправность и основной процесс принимает решение отправить о ней сообщение в виде электронного письма. После отправки письма мониторная система работает в прежнем режиме, но при этом система мониторинга способна принять и обработать ответ от администратора, что сообщение о неисправности получено. Если по истечении промежутка времени, указанного в конфигурационном файле, подтверждения о получении письма администратором не пришло и сервис не восстановлен, система отправляет еще одно сообщение, но теперь это может быть сообщение другого типа (например, sms). Количество дополнительных сообщений указывается администратором в конфигурационном файле. На рисунке 7.4 показан вид конфигурационного файла, в котором указана посылка двух дополнительных сообщений.

Каждой неисправности при ее обнаружении присваивается некоторый идентификатор (ID), который представляет собой время возникновения неисправности, выраженное в секундах от 1 января 1970г. При обнаружении неисправности в домашней директории пользователя `mon` создается файл, названием которого является идентификатор неисправности. При получении подтверждения на сообщение о неисправности в соответствующий файл записывается о наличии подтверждения. При восстановлении сервиса файл, название которого соответствует идентификатору неисправности, удаляется.

Идентификатор неисправности (ID) указывается в теле письма о проблеме, а также в поле `Subject`. Администратор обязан указать ID-неисправности в письме, которое является ответом на первое оповещение о проблеме.

7.3.6 Настройка `crontab` в случае использования Web-сайта и оповещения с помощью sms-сообщений

При использовании Web-сайта необходимо произвести настройку `crontab` для корректного архивирования log-файлов, содержащих данные для отображения на сайте. Log-файлы располагаются в директории `/var/log/mon`. Для сжатия log-файлов используется программа `pilelog`.

Архивация файлов может производиться в любое время на усмотрение администратора. На Web-сайте отображается состояние сети за период времени, указанный в log-файле. Рекомендуется архивировать log-файлы один раз в сутки. При использовании `pilelog` удобно использовать опцию `-d guess`, что позволяет к имени полученного архива добавить число — день, в течение которого производились записи в этот файл. Кроме того, полученный архив будет помещен в каталоги с указанием месяца и года. Информация о дате берется из первой строки log-файла, в которой указаны время и дата создания записи.

Отправка sms-сообщения о событии в сети производится следующим образом: сообщение помещается alert-модулем в файл `/var/spool/sms`, далее специальный скрипт, который запускается по `crontab`'у, проверяет, есть ли что-либо в `spool`, и если обнаруживает там некоторый файл, отправляет его по указанному номеру.

Частота проверки наличия данных с `spool` зависит от желания администратора. Рекомендуется проверять `spool` один раз в минуту. Кроме того, имеется возможность использования модуля, проверяющего работоспособность устройства отправки sms-сообщений. Программа проверки также вызывается по `cron`'у. Пример записи в `crontab` приведен на рисунке 7.5.

```
20 0 * * * /home/vera/pilelog -d guess /var/log/mon/fping_monitor.log  
  
59 */2 * * * root    /usr/local/bin/checkmodem  
0-58 * * * *   root    /usr/local/bin/sendsms
```

Рис. 7.5: Запись в `crontab` для архивирования log-файлов и использования поповещения по SMS

7.3.7 Использование Web-сайта системы MON

Файлы с данными для построения сайта состояния сети, а также скрипты, осуществляющие построение сайта находятся в каталоге `/var/www/mon`. По умолчанию сайт состояния сети можно увидеть в браузере по адресу `http://your_host.com/mon/mon.html`.

Состояние сети представлено в Web в виде таблицы. Перед таблицей указано, к какой группе и сервису она относится. В первой строке указано время проверок, в первом столбце — названия проверяемых компьютеров. На пересечении строки и столбца можно увидеть состояние интересующего компьютера в определенный момент времени.

Состояние отображается цветом. Считается, что работоспособный сервис отображен зеленым цветом, неработоспособный — красным. Цвет клетки вычисляется в зависимости от степени доступности сервиса. Шкала используемых цветов изображается перед таблицей и представляет собой градиентный переход от зеленого к красному.

```
# Пути к основным рабочим директориям

cfbasedir      = /etc/mon
alertdir       = /usr/lib/mon/alert.d
mondir         = /usr/lib/mon/mon.d

# Группы компьютеров, которые необходимо проверять.

hostgroup FirstGroup host1.domain1.com host2.domain1.com

hostgroup SecGroup server1.domain2.com server2.domain2.com

# Параметры проверок

watch FirstGroup
    service fping
    description ping-test
    interval 10m
    monitor fping.monitor
    period wd {1-4} hr {8-20}
        numalerts 2
    alert mail.alert root@host1.domain1.com
    upalert mail.alert root@host1.domain1.com

watch SecondGroup
    service http
    description http port check
    interval 1h
    monitor http.monitor
    period wd {2-6} hr {4-23}
        numalerts 5
    alert mail.alert root@host1.domain2.com
    unacked 1h
        numalerts 1
    alert sms.alert 89101234567
    upalert sms.alert 89101234567
```

Рис. 7.6: Пример конфигурационного файла системы МОН

```
all_groups=2;
group1='brick';
all_services1=2;
service1_1='http';
group1_service3_1=[['','bonjour.botik.ru','seaworld.botik.ru',
                   'brick.botik.ru','river.botik.ru',
                   'dove.pereslavl.ru'],
                  ['16:27',0,0,0,0,0],['16:07',0,0,0,0,0]];
service1_2='fping';
group4_service4_2=[['','bonjour.botik.ru','seaworld.botik.ru',
                   'brick.botik.ru','river.botik.ru',
                   'dove.pereslavl.ru'],
                  ['16:27',0,0,0,0,0],['16:23',0,0,0,0,0],
                  ['16:22',0,0,0,0,0],['16:21',0,0,0,0,0],
                  ['16:20',0,0,0,0,0],['16:19',0,0,0,0,0],
                  ['16:18',0,0,0,0,0],['16:13',0,0,0,0,0],
                  ['16:12',0,0,0,0,0]
];
group2='servers';
all_services2=2;
service2_1='ping';
group2_service2_1=[['','joker.botik.ru','up.botik.ru','tower.botik.ru'],
                  ['19:40',0,0,0],['19:30',0,0,0],['19:20',0,0,0],
                  ['19:10',0,0,0],['19:00',90,90,90],['18:50',90,90,90],
                  ['18:40',90,90,90]
];
service2_2='smtp';
group2_service2_2=[['','joker.botik.ru','up.botik.ru','tower.botik.ru'],
                  ['19:40',0,0,0],
                  ['19:20',0,0,0],
                  ['19:00',100,100,100],
                  ['18:40',100,100,100]
];
```

Рис. 7.7: Пример файла, сгенерированного для построения Web-сайта

Список использованных источников

- [1] Шевчук Ю.В. Методы построения экономически эффективных региональных компьютерных сетей / Текст диссертации на соискание ученой степени кандидата технических наук, специальность 05.13.11 — математическое и программное обеспечение вычислительных машин, комплексов, систем и сетей, ИПС РАН, 1999, доступна как <http://ftp.botik.ru/pub/local/sizif/brick/brick.ps.gz>
- [2] Абрамов С.М., Позлевич Р.В., Пономарев А.Ю., Шевчук Ю.В. Опыт использования PC-роутеров // Труды симпозиума: Международный симпозиум “Роль информатики в региональном развитии”, Переславль-Залесский, 26–29 октября 1996 г., Ред.: Айламазян А.К., М., Наука-Физматлит, 1997. С. 48–50.
- [3] Абрамов С.М., Позлевич Р.В., Пономарев А.Ю., Шевчук Ю.В. Экономически эффективные технологии построения региональных сетей для науки и высшей школы // Труды конференции: Телематика'97, СПб, 1997.
- [4] Абрамов С.М., Сеть для всех и по разумным ценам // Компьютерра, 34 (211), 25 августа 1997 г., 1997. С. 28–30.
- [5] Экономически эффективные технологии построения городских телекоммуникационных систем для науки и образования. Методические материалы, электронный ресурс, <http://www.botik.ru/tech/>. 1997.
- [6] Абрамов С.М., Котельников В.П., Пономарев А.Ю., Шевчук Ю.В. О построении высокоскоростная оптической магистрали городской компьютерной сети с учетом особенностей электропитания в районных центрах России // Труды конференции: Всероссийская научная конференция “Научный сервис в сети Интернет”, 23-28 сентября 2002 г., г. Новороссийск, М., Изд-во МГУ, 2002, С. 244–247.
- [7] Ермилова Е.В., Карлаш А.В., Нестеров А.С., Жбанов П.Г., Шевчук Ю.В., Nadmin – система администрирования для региональных сетей // Программные системы:

- теория и приложения. / Труды международной конференции “Программные системы: теория и приложения”, ИПС РАН, г.Переславль-Залесский, май 2004 / Под редакцией С.М.Абрамова. В двух томах. - М.: Физматлит, 2004. - Т. 1. С. 231–256.
- [8] Бурчу С.В. Методы хостинга веб-сайтов, применимо к системе телекоммуникаций “Ботик” //Программные системы: теория и приложения. / Труды международной конференции “Программные системы: теория и приложения”, ИПС РАН, г.Переславль-Залесский, май 2004 / Под редакцией С.М.Абрамова. В двух томах. - М.: Физматлит, 2004. - Т. 1. С. 257–264.
- [9] Парменова В.В. Адаптация мониторной системы МОН для системы коммуникаций “Ботик” //Программные системы: теория и приложения. / Труды международной конференции “Программные системы: теория и приложения”, ИПС РАН, г.Переславль-Залесский, май 2004 / Под редакцией С.М.Абрамова. В двух томах. - М.: Физматлит, 2004. - Т. 1. С. 265–276.
- [10] Кузнецов А.А. Разработка геоинформационной системы для СТ “Ботик” с использованием TCL/TK для реализации клиентской части //Программные системы: теория и приложения. / Труды международной конференции “Программные системы: теория и приложения”, ИПС РАН, г.Переславль-Залесский, май 2004 / Под редакцией С.М.Абрамова. В двух томах. - М.: Физматлит, 2004. - Т. 1. С. 277–297.
- [11] Парменова В.В. Расширение функциональности МОН-системы мониторинга компьютерной сети // Научный сервис в сети Интернет: Труды Всероссийской научной конференции, 20-25 сентября 2004 г. Новороссийск, Изд-во МГУ, М., 2004, С. 67–69.
- [12] Кузнецов А.А., Гумин М.В. Разработка геоинформационной системы для системы телекоммуникаций “Ботик” // Научный сервис в сети Интернет: Труды Всероссийской научной конференции, 20-25 сентября 2004 г. Новороссийск, Изд-во МГУ, М., 2004, С. 91-93.
- [13] Бурчу С.В., Шевчук Ю.В. Сервер аренды WWW-пространства // Научный сервис в сети Интернет: Труды Всероссийской научной конференции, 20-25 сентября 2004 г. Новороссийск, Изд-во МГУ, М., 2004, С. 220-221.
- [14] Абрамов С.М., Котельников В.П., Пономарев А.Ю., Шевчук Ю.В., Якубов Б.В. Новые аппаратные разработки для региональных компьютерных сетей от лаборатории “Ботик” // Научный сервис в сети Интернет: Труды Всероссийской научной конференции, 20-25 сентября 2004 г. Новороссийск, Изд-во МГУ, М., 2004, С. 244-246.

- [15] Карлаш А.В., Абрамов С.М., Жбанов П.Г., Нестеров А.С., Ермилова Е.В., Шевчук Е.В. Надмин — административно-расчетная система для региональных сетей // Научный сервис в сети Интернет: Труды Всероссийской научной конференции, 20-25 сентября 2004 г. Новороссийск, Изд-во МГУ, М., 2004, С. 195-200
- [16] Журнал «Радио», № 11, 2001.
- [17] Трансформаторы и дроссели для импульсных источников питания//Электронный ресурс http://www.compitech.ru/html/arhiv_s/00_01/stat_30.htm (сетевой журнал “Компоненты и технологии”)
- [18] Документация к микросхеме CS3842 фирмы ONSEMI.

Приложение А

Система администрирования Nadmin для региональных сетей

A.1 Сенсор IP-статистики

Сенсор IP-статистики, программа `sensor_ip.pl`, производит обработку выходных файлов стандартной утилиты `tcpdump`¹ и вывод IP-статистики принятого в Nadmin формата (см. раздел 5.1.3, пункт “Сенсор IP_stat”).

A.1.1 Запуск сенсора IP-статистики

При запуске программе `sensor_ip.pl` передается аргумент `$router` — имя граничного маршрутизатора (`slav`, `post` и т. п.), на котором работает утилита `tcpdump`. Список имен маршрутизаторов для сенсоров задан в конфигурационном файле следующим образом:

```
sub sensors
{
(
    "/usr/lib/nadmin/bin/sensor_ip_stat.pl post",
    "/usr/lib/nadmin/bin/sensor_ip_stat.pl slav",
    ...
);
}
```

¹Стандартная утилита `tcpdump` позволяет собирать информацию о сетевом трафике. Она способна исследовать различные типы пакетов (`packet`), проходящие через интерфейс, фильтровать пакеты и выводить их заголовки.

A.1.2 Структура иерархии каталогов для выходных файлов утилиты tcpdump

Файлы с выводом утилиты `tcpdump` расположены в каталоге, имя которого задано правилом:

```
$logdir = "/path/$router-router/outer";
```

Тем самым для каждого маршрутизатора используется свой каталог.

Путь `path` задан в конфигурационном файле системы Nadmin (см. раздел 5.1.10 Конфигурационный файл `config`).

Такой каталог содержит подкаталоги, соответствующие дням, например, с названиями 20041101, 20041102, 20041103 и т. д. Внутри этих подкаталогов находятся выходные файлы, соответствующие пятиминутным промежуткам времени, например, с названиями 09:00, 09:05, 09:10.

A.1.3 Работа сенсора IP-статистики

В начале работы сенсора происходит обработка аргументов поступивших из командной строки функцией `parse_args`.

```
parse_args ();
usage () unless defined $router;
```

Если при вызове сенсора не было указано название маршрутизатора, то выдается сообщение об ошибке.

`Usage: sensor_ip ROUTERNAME`

Далее инициализируется марка — объект `Nadmin::Mark`:

```
$marker = new Nadmin::Mark ("sensor_ip_stat_$router");
```

Если марки не существует, ее определяет функция `create_initial_mark`:

```
create_initial_mark() unless ($marker->get_mark);
```

Когда марка определена, сенсор выводит в стандартный поток (`stdout`) строку, сообщающую ОПО о работоспособности сенсора.

```
print "Sensor-Type: IP:$router\n";
```

Далее происходит обработка выходных файлов утилиты `tcpdump` и вывод в стандартный поток (`stdout`) статистики функцией `process_slice`. Если все существующие на некоторый момент выходные файлы обработаны, то происходит приостановление выполнения процесса обработки на пять минут.

```
while (1) {
    my $status = process_slice ();
    if (! $status) {
        # no unprocessed files left.
        sleep 300;
    }
}
```

A.1.4 Функции программы `sensor_ip.pl`

Программа `sensor_ip.pl` использует следующие функции для обработки выходных файлов утилиты `tcpdump` и вывода статистики.

- `my $status = process_slice()` — данная функция производит вызов функции `select_and_process_file` для обработки выходного файла утилиты `tcpdump` и выводит в стандартный поток (`stdout`) статистику установленного в Nadmin формата. Функция возвращает число равное единице, если существуют еще не обработанные выходные файлы `tcpdump`, или нулю, если такие файлы не существуют. В первом случае следующий запуск `process_slice` происходит немедленно. Во втором случае следующий запуск произойдет через пять минут.
- `my ($has_more_work, $date1,$date2) = select_and_process_file()` — данная функция создает список необработанных выходных файлов утилиты `tcpdump` (вызовом функции `list_files_from`) и передает на обработку первый файл из этого списка в функцию `Nadmin::Pcap::process_file`. Функция возвращает три аргумента: булево значение, равное единице, если существуют еще не обработанные выходные файлы утилиты `tcpdump`, или нулю, если такие файлы не существуют, и две даты — начальная и конечная даты обработанной статистики.
- `@unprocessed = list_files_from($mark)` — возвращает список всех выходных файлов утилиты `tcpdump`, созданных после даты из марки `$mark`.
- `Nadmin::Pcap::process_file($file, \&process_packet)` — данная функция производит чтение выходного файла утилиты `tcpdump` `$file` и передачу массива информации о каждом пакете IP-трафика функции `process_packet`.

Модуль `Nadmin::Pcap` — это программный интерфейс для perl-модуля `Net::Pcap`, содержащего функции сбора и обработки сетевой статистики.

Функция `process_packet` производит сбор и обработку информации о каждом пакете сетевого трафика, заголовки которого хранятся в выводном файле утилиты `tcpdump $file`, переданной функцией `Nadmin::Pcap::process_file`.

В следующих разделах некоторые функции описаны подробнее.

A.1.4.1 Функция `process_slice`

В начале работы функции `process_slice` происходит инициализация ассоциативных массивов, в которые затем будет записана обработанная IP-статистика для вывода на стандартный поток (`stdout`):

```
local (%data_in, %data_out, %count_in, %count_out);
```

`%data_in` — сырья² статистика выходящего трафика, `%data_out` — сырья статистика исходящего трафика, `%count_in` — объем входящего трафика, `%count_out` — объем исходящего трафика.

Затем инициализируется переменная `$unparsable` — счетчик пакетов, переданных не по протоколу IP_V4.

```
local $unparsable = 0;
```

Переменная `$unparsable` может получить значение, не равное нолью, в функции `process_packet`, где происходит анализ записей о переданных пакетов трафика.

Если эта переменная к концу работы `process_slice` не равна нулю, то выдается предупреждение

```
sprintf ("sensor_ip %s: %s: %.0f unparsable packets ignored",
        $router, $$date1, $unparsable);
```

После определения значения переменной `$unparsable` производится вызов функции `select_and_process_file`:

```
my ($has_more_work,$date1,$date2) = select_and_process_file();
```

²Сырая статистика для данного сенсора не является строками выходного файла. Это происходит из-за того, что выходные файлы утилиты `tcpdump` очень велики и содержат много лишней для наших целей информации, что делает их неудобными для хранения и повторной обработки. Сырая IP-статистика формируется самим сенсором `sensor_ip.pl`.

Значение `$has_more_work` показывает, есть ли еще необработанная статистика. Период от `$date1` до `$date2` — это период, за который была обработана статистика. Далее происходит вывод в стандартный поток (`stdout`) слайсов на основе информации из `%data_in`, `%data_out`, `%count_in`, `%count_out`.

```

foreach my $ip (keys %count_in) {
    print "Date1: ".$date1->ymdhms."\n";
    print "Date2: ".$date2->ymdhms."\n";
    print "Id: $ip\n";
    print "Count_in: $count_in{$ip}\n";
    print "Count_out: $count_out{$ip}\n";
    print "\n";
    foreach my $id (keys %{$data_in{$ip}}){
        print $data_in{$ip}{$id}, ":";
        print $data_out{$ip}{$id}, " ";
        $id =~ s/(^|\D)$ip(?!\\d)/$1x/;
        print "$id\n";
    }
    print "\n";
}

```

Формат слайса пояснен на следующем примере:

```

Date1: 20030606160105
Date2: 20030606160110
Id: 193.232.174.2
Count-in: 19356
Count-out: 1346

144:120 81.117.239.98.3333 x.8081 tcp
144:120 81.117.239.98.3233 x.3128 tcp
144:120 81.117.239.98.3162 x.80 tcp
144:120 81.117.239.98.3268 x.8080 tcp
144:0 81.117.239.98.2982 x.1080 tcp

```

В первой и второй строках слайса задаются соответственно даты начала (6 июня 2003, 16:01:05) и конца (6 июня 2003, 16:01:10) периода времени, за который поступал IP-трафик.

фик. В третьей задан IP-адрес компьютера (193.232.174.2), для которого был входящий и исходящий трафик за данный период времени. В четвертой и пятых задаются соответственно суммарные объемы входящего (19356 байт) и исходящего (1346 байт) трафика за этот период времени. Начиная с седьмой строки, перечислены строки сырой статистики за данный период времени.

Строка сырой статистики состоит из следующих элементов: объем входящего трафика, объем исходящего трафика, IP-адрес компьютера и порт компьютера, принимающего трафик, IP-адрес и порт компьютера, отсылающего трафик, название протокола обмена. IP-адрес компьютера, которому принадлежит слайс, заменяется символом x, чтобы сохранить дисковое пространство.

После вывода слайсов происходит запись \$date2 в марку.

```
$marker->put_mark($date2->ymdhms);
```

Функция возвращает значение \$has_more_work — показатель того, запускать ли process_slice снова сразу же или после пятиминутной задержки.

A.1.4.2 Функция process_packet

Функция получает \$h — указатель на хэш³ характеристик пакета IP-трафика.

Хэш \$h имеет следующие ключи и значения:

VERSION — версия протокола. Если версия не четвертая, то данный пакет трафика невозможно анализировать. В таком случае значение переменной \$unparseable увеличивается на единицу, затем функция process_packet возвращает значение нуль.

SRC — IP-адрес отправителя пакета (значение определяет переменную \$src).

DST — IP-адрес получателя пакета (значение определяет переменную \$dst).

SRCPORT — порт отправителя пакета (значение определяет переменную \$src).

DSTPORT — порт получателя пакета (значение определяет переменную \$dst).

TOTLEN — количество байт (значение определяет переменную \$count).

PROTONAME или PROTO — название протокола обмена (значение определяет переменную \$proto).

³Хэш — ассоциативный массив в языке программирования Perl.

Сразу после определения переменных `$src`, `$dst`, `$proto`, `$count` определяется переменная `$ip` (IP-адрес компьютера СТ “Ботик”) — это либо значение SRC, либо значение DST. IP-адрес `$ip` для ОПО является в дальнейшем идентификатором сервиса.

Далее по этим переменным заполняются хэши `%data_in`, `%data_out`, `%count_in`, `%count_out`.

Если с компьютера `$ip` выходил трафик, то в хэши записываются следующие значения:

```
my $id = "$src $dst $proto";
$data_out{$ip}{$id} += $count;
$count_out{$ip} += $count;
```

Если на компьютер `$ip` поступал трафик, то в хэши записываются следующие значения:

```
my $id = "$dst $src $proto";
$data_in{$ip}{$id} += $count;
$count_in{$ip}+=$count;
```

Хэши `%data_in`, `%data_out`, `%count_in`, `%count_out` необходимы в дальнейшем в функции `process_slice` для вывода в стандартный поток вывода (`stdout`) IP-статистики.

A.2 Класс `Nadmin::Account`

Лицевой счет абонента в административной системе Nadmin представлен объектом класса `Nadmin::Account` и описан в модуле `Nadmin::Account.pm` (см. раздел 5.1.4, пункт “Лицевой счет абонента и класс `Nadmin::Account`”).

A.2.1 Функциональность класса `Nadmin::Account`

Ниже описаны действия, производимые в системе Nadmin с лицевым счетом абонента.

A.2.1.1 Занесение текущих расходов в run-файл лицевого счета

Результат обсчета каждого фрагмента статистики потреблении абонентом услуг⁴ регистрируется в файле `account_run`. Например, строка:

⁴Услугами в системе Nadmin называется предоставление абоненту интернет-подключения (Connect), ftp-пространства (Ftp), пин-кода ip-телефонии (Ipphone) и т. п.

20040528113000 0.034 connect katta.pereslavl.ru ip:slav 144 0 0

содержит следующую информацию:

- начальная дата фрагмента статистики (20040528113000⁵);
- накопительная стоимость потребленной услуги (0.034 рубля);
- класс услуги (`connect`);
- имя услуги (`katta.pereslavl.ru`);
- тип сенсора (`ip:slav`);
- объем входящего трафика (144 байта);
- объем исходящего трафика (0 байт);
- объем локального трафика (0 байт);

В случае, если понятия входящего, исходящего, локального трафика для услуги не определены, то объемы трафика считаются равными нулю.

Накопительная стоимость услуги, представленной данным слайсом статистики, получается суммированием стоимости этой услуги (назовем ее *действительной* стоимостью услуги) и стоимости услуг из предыдущих фрагментов статистики, представленных в строках, расположенных выше данной строки.

Строки в `account_run` копятся в течение расчетного периода. В конце расчетного периода происходит регистрация расходов на лицевом счету абонента, в файле `account`.

A.2.1.2 Занесение на лицевой счет расходов за расчетный период и платежей

Строки файла `account` бывают двух видов — строки расходов и строки платежей.

Строка расхода. Для формирования строки расхода строки run-файла `account_run`, пришедшие за истекший расчетный период:

- извлекаются из файла `account_run`,
- обрабатываются для формирования строки расхода в `account`,
- записываются в файл `account` в каталоге архива данной организации (см. раздел 5.1.5).

⁵Запись 20040528113000 представляет дату 28 мая 2004 года, 11:30:00.

Строка расхода состоит из элементов, показанных на следующем примере:

20040601000000 -113.53 9.52 расход за 2004/05/16 - 2004/06/01

- дата конца расчетного периода (20040601000000⁶);
- сумму расхода (-113.53 рублей);
- остаток на счету (9.52 рублей);
- комментарий (расход за 2004/05/16 - 2004/06/01 рублей).

Комментарий имеет вид: расход за *Date1–Date2*, где *Date1* — дата конца прошлого расчетного периода, *Date2* — дата конца данного расчетного периода.

Строка платежа. Стока платежа состоит из элементов, показанных на следующем примере:

20040422140402 211.86 116.25 (250.00-38.14 НДС) в кассу

- дата регистрации платежа (20040422140402⁷);
- сумму платежа (211.86 рублей);
- остаток на счету (116.25 рублей);
- комментарий ((250.00-38.14 НДС) в кассу).

Комментарии может содержать информацию о том, кем, где был принят платеж, а также информацию о налогах.

A.2.1.3 Операции с уровнем отключения абонента

Уровень отключения — это минимальное значение суммы на лицевом счету, при котором происходит отключение абонента.

Уровень отключения абонента — это свойство объекта `Nadmin::Org`, но оно непосредственно зависит от объекта `Nadmin::Account`.

Уровень отключения формируется совместно системой и абонентом и может принимать разные значения.

Уровень отключения равен нулю. Нуль является значением уровня отключения по умолчанию.

⁶Запись 20040601000000 представляет дату 1 июня 2004 года, 00:00:00.

⁷Запись 20040422140402 представляет дату 22 апреля 2004 года, 14:04:02.

Уровень отключения больше нуля. Абонент всегда может сделать уровень своего отключения положительным (но не больше суммы на его лицевом счету). Это необходимо тогда, когда абонент ставит перед собой цель тратить в какой-либо период фиксированную сумму со своего лицевого счета.

Уровень отключения меньше нуля. Такой уровень допустим не для всех абонентов. Если расходы абонента за несколько последних расчетных периодов не опускались до уровня отключения, то это значит, что абонент в этот период кредитовал СТ “Ботик”. Поэтому представляется разумным позволить системе кредитовать данного абонента, разрешив ему иметь отрицательный уровень отключения.

Минимальный уровень отключения каждого абонента вычисляется как взвешенное среднее от сумм, на которые абонент кредитовал СТ “Ботик” в течение нескольких последних расчетных периодов.

Каждое изменение состояния лицевого счета и уровня отключения фиксируется в журнале `eventlog` (см. раздел 5.1.9). Записи в `eventlog` обрабатываются, в результате чего может произойти отключение или подключение абонента.

A.2.1.4 Просмотр состояния лицевого счета

В новой версии административной системы Nadmin можно получить следующую информацию о лицевом счете абонента:

- остаток на лицевом счету на начало текущего расчетного периода;
- сумму расходов за текущий расчетный период;
- текущий остаток на лицевом счету;
- минимальный уровень отключения абонента.

A.2.2 Методы класса Nadmin::Account

Класс `Nadmin::Account` содержит следующие методы:

- `my $account = new Nadmin::Account ($org, %args)` — возвращает объект “лицевой счет” организации с именем `$org`. Хэш `%args` может содержать элементы: с ключом `BY_ID` — индекс организации `$org` (см. раздел 5.1.4 “База данных”), с ключом `NUM` — номер лицевого счета в случае наличия у абонента нескольких лицевых счетов, с ключом `DATE` — дата версии лицевого счета `$account`. Если индекс организации задан в `%args`, то путь к rcs-файлу лицевого счета определяется сразу.

Иначе происходит поиск индекса организации с именем `$org` в индексном файле организаций, а затем определение пути к rcs-файлу.

- `my @list = list Nadmin::Account(%args)` — возвращает список `@list` объектов “лицевой счет”. Хэш `%args` имеет элементы, ключи которых являются свойствами организации, владеющей данным лицевым счетом. Если хэш не задан, строится список всех объектов “лицевой счет”.

A.2.2.1 Занесение текущих расходов в run-файл лицевого счета

Занесение текущих расходов в run-файл лицевого счета производит следующий метод:

- `$account->register_drop($date, $money, $servclass, $servname, $resource, $count_in, $count_out, $count_local)` — заносит строку платежа в run-файл лицевого счета `$account`, которая состоит из `$date` — дата регистрации расхода, `$money` — стоимость трафика, `$servclass` — вид услуги (connect, ftp и т. п.), `$servname` — название услуги (имя подключения, логин ftp-пространства и т. п.), `$resource` — вид ресурса (ip-статистика, proxy, ...), `$count_in` — объем входящего трафика, `$count_out` — объем исходящего трафика, `$count_local` — объем локального трафика.

A.2.2.2 Занесение на лицевой счет расходов за расчетный период и платежей

Занесение на лицевой счет расходов за расчетный период и платежей производят следующие методы:

- `$account->register_charge($date, $charge, $logrecord, $lodate, $hidate)`: заносит строку расхода в файл лицевого счета `$account`, состоящую из `$date` — дата регистрации, `$charge` — сумма расхода, `$logrecord` — комментарий для новой версии файла (содержит суть регистрации расхода), `$lodate` — начальная дата промежутка времени, когда был произведен расход, `$hidate` — конечная дата промежутка времени, когда был произведен расход,
- `$account->register_payment ($date, $delta, $logrecord, $comment)` — заносит строку платежа в файл лицевого счета `$account`, которая состоит из `$date` — дата регистрации, `$delta` — сумма платежа, `$logrecord` — комментарий для новой версии файла (содержит суть регистрации платежа), `$comment` — комментарий для строки платежа.

- `$account->unregister_payment ($logrecord)` — убирает последнюю строку файла лицевого счета `$account`, `$logrecord` — комментарий для новой версии файла (содержит пояснение к удалению последней строки).
- `my $sum =$account->summarize_drops ($hidate)` — метод возвращает сумму расходов на лицевом счету `$account` за текущий расчетный период, поступивших до `$hidate`.
- `$account->cut_and_archive_drops($lodate, $hidate)` — метод извлекает строки из run-файла лицевого счета `$account` за период с `$lodate` по `$hidate` и помещает их в архив.
- `$account->integrate_drops($hidate)` — извлекает из run-файла лицевого счета `$account` записи, которые поступили до даты `$hidate`, суммирует их и делает запись расхода за расчетный период.
- `my $lodate =$account->period_beginning($hidate)` — метод возвращает дату `$lodate` — конец прошлого расчетного периода для лицевого счета `$account`, если конец расчетного периода — `$hidate`.
- `Nadmin::Account::roll_account_run($file)` — в каждой строке run-файла `$file` заменяет накопительную (см. раздел А.2.1.1) стоимость трафика на действительную стоимость этого трафика.
- `Nadmin::Account::unroll_account_run($file)` — заменяет в каждой строке run-файла `$file` действительную стоимость трафика на накопительную стоимость этого трафика.

A.2.2.3 Операции с уровнем отключения абонента

Операции с уровнем отключения абонента осуществляют следующие методы:

- `my $min_offlevel = $account->min_offlevel()` — выдает уровень отключения `$min_offlevel`, который является минимальным для лицевого счета `$account`.
- `$account->check_offlevel()` — проверяет, превышает ли остаток на лицевом счету `$account` уровень отключения. Если нет, то в `eventlog` делается запись об этом.

A.2.2.4 Просмотр состояния лицевого счета

Просмотр состояния лицевого счета организации осуществляется с помощью следующих методов:

- `my @tail = $account->tail($n)` — возвращает список последних `$n` строк файла `account` для лицевого счета `$account`.
- `my $left = $account->left()` — возвращает текущий остаток на лицевом счету `$account`. Этот остаток получается вычитанием из остатка на лицевом счету на начало текущего расчетного периода (этот остаток извлекается из файла `account`) суммы расхода за текущий расчетный период (эта сумма извлекается из run-файла).
- `my $integrated_left = $account->integrated_left()` — метод выдает остаток `$integrated_left` на лицевом счету `$account` на начало текущего расчетного периода.

Приложение В

Блок 220–12

B.1 Расчеты характеристик

B.1.1 Технические требования на Блок 220–12

Мощность потребления ПК-маршрутизатора	30Вт
На зарядку АКБ	20Вт
Мощность при напряжении > 150В	50Вт
Мощность при напряжении > 100В	> 25Вт

B.1.2 Расчет импульсного блока питания для UPS 50Вт

B.1.2.1 Начальные данные

$$\begin{aligned}U_{\text{H}} &= 13.8\text{В}, \\I_{\text{H}} &= 3.6\text{А}, \\U_{\text{сети}max} &= 300\text{В}, \\U_{\text{сети}min} &= 150\text{В}(100\text{В}), \\F &= 100\text{кГц}.\end{aligned}$$

B.1.2.2 Процедура расчета

1. Определение максимального и минимального значения выпрямленного сетевого напряжения

$$U_{\text{BX}max} = \sqrt{2} \cdot 300\text{В} = 425\text{В},$$

$$\begin{aligned} U_{\text{BX}min} &= \sqrt{2} \cdot 150\text{B} - \tilde{U} = 212 - 31\text{B} = 180\text{B}, \\ U_{\text{BX}min*} &= \sqrt{2} \cdot 100\text{B} - \tilde{U} = 141 - 31\text{B} = 110\text{B} \end{aligned}$$

$U_{\text{BX}min*}$ — при снижении мощности; \tilde{U} — пульсации на входном конденсаторе.

2. Выбор диодов выпрямителя

Средний ток диода

$$\begin{aligned} I_d &= U_{out} \cdot I_{out} / (U_{\text{BX}min} \cdot \text{КПД} \cdot 2), \\ I_d &= 50 / (180 \cdot 0.8 \cdot 2) = 0.18\text{A}, \\ I_d &= 25 / (110 \cdot 0.8 \cdot 2) = 0.14\text{A}. \end{aligned}$$

Используем диоды 1N4006.

3. Расчет емкости конденсатора

$$\begin{aligned} C_{in} &= 0.5 \cdot U_H \cdot I_H / (U_{\text{сети}min} \cdot \text{КПД} \cdot 100\Gamma_{\text{Ц}} \cdot \tilde{U}), \\ C_{in} &= 0.5 \cdot 50 / (180 \cdot 0.8 \cdot 100\Gamma_{\text{Ц}} \cdot 31) = 56\text{мкФ}, \\ C_{in} &= 0.5 \cdot 25 / (110 \cdot 0.8 \cdot 100\Gamma_{\text{Ц}} \cdot 31) = 45\text{мкФ}. \end{aligned}$$

4. Расчет максимального коэффициента заполнения конденсатора

$$\begin{aligned} K_{\text{зап}} &= U_{\text{доп}} / (U_{\text{доп}} + U_{\text{вх min}} - U_{\text{си}}), \\ K_{\text{зап}} &= 145 / (145 + 180 - 2) = 0.45, \\ K_{\text{зап}} &= 145 / (145 + 110 - 2) = 0.57. \end{aligned}$$

5. Расчет трансформатора

5.1. Максимальный ток первичной обмотки

$$\begin{aligned} I_{1\text{ И}} &= 2.1 \cdot I_H \cdot U_H / (U_{\text{BX}min} \cdot K_{\text{зап}} \cdot \text{КПД}), \\ I_{1\text{ И}} &= 2.1 \cdot 50 / (180 \cdot 0.45 \cdot 0.8) = 1.62\text{A}, \\ I_{1\text{ И}} &= 2.1 \cdot 25 / (110 \cdot 0.57 \cdot 0.8) = 1.05\text{A}. \end{aligned}$$

5.2. Действующее значение тока первичной обмотки

$$\begin{aligned} I_1 &= I_{1\text{ И}} \cdot \sqrt{K_{\text{зап}}/3} = 1.62 \cdot 0.39 = 0.63\text{A}, \\ I_1 &= I_{1\text{ И}} \cdot \sqrt{K_{\text{зап}}/3} = 1.05 \cdot 0.44 = 0.46\text{A}. \end{aligned}$$

5.3. Коэффициент трансформации

$$n = W_2/W_1 = (U_H + U_{\Delta})/U_{\text{доп}} = 14.5/145 = 1/10.$$

5.4. Действующее значение тока вторичной обмотки

$$\begin{aligned} I_2 &= I_{1\text{ и}}/n \cdot \sqrt{(1 - K_{\text{зап}})/3} = 1.62 \cdot 10 \cdot 0.43 = 7.0, \\ I_2 &= I_{1\text{ и}}/n \cdot \sqrt{(1 - K_{\text{зап}})/3} = 1.05 \cdot 10 \cdot 0.38 = 4.0. \end{aligned}$$

5.5. Индуктивность первичной обмотки

$$L_1 = K_{\text{зап}} \cdot U_{\text{BX}min}/(I_{1\text{ и}} \cdot F).$$

При $F = 100\text{kГц}$,

$$L_1(100) = 0.45 \cdot 180/(1.62 \cdot 100000) = 0.5\text{мГн},$$

$$L_1(100) = 0.57 \cdot 110/(1.05 \cdot 100000) = 0.6\text{мГн}.$$

B.1.2.3 Процедура расчета

1. для ETD29 с зазором 0.5мм и с зазором 1.0мм, площадь сердечника $Sc = 0.76\text{см}^2$, объем $V = 5.35\text{см}^3$, тепловое сопротивление 28 °C/Bт .
2. для ETD34 с зазором 0.5мм, площадь сердечника $Sc = 0.97\text{см}^2$, объем $V = 7.63\text{см}^3$, тепловое сопротивление 20 °C/Bт , $f = 100\text{kГц}$

5.6. Определение количества витков для первичной обмотки

$$W_1 = \sqrt{L_1/A_l}.$$

ETD29 0.5мм ($A_l=201\text{nГн}$): $W_1 = 50$ витков,

ETD29 1.0мм ($A_l=124\text{nГн}$): $W_1 = 64$ витка,

ETD34 0.5мм ($A_l=251\text{nГн}$): $W_1 = 45$ витков,

ETD34 1.0мм ($A_l=153\text{nГн}$): $W_1 = 57$ витков.

Приращение индукции:

$$B = 10000 \cdot U_{\text{BX}min} \cdot K_{\text{зап}}/(W_1 \cdot Sc \cdot F),$$

$$B = 10000 \cdot 180 \cdot 0.45/(W_1 \cdot 0.76 \cdot F) = 1066800/(W_1 \cdot F),$$

$$B = 10000 \cdot 110 \cdot 0.57/(W_1 \cdot 0.76 \cdot F) = 825000/(W_1 \cdot F),$$

$$ETD29\ 0.5\text{мм} : B = 0.22\text{Tл},$$

$$ETD29\ 1.0\text{мм} : B = 0.17\text{Tл},$$

$$B = 10000 \cdot 180 \cdot 0.45/(W_1 \cdot 0.97 \cdot F) = 835000/(W_1 \cdot F),$$

$$ETD34\ 0.5\text{мм} : B = 0.19\text{Tл}.$$

Определим потери в сердечнике из материала. Используем графики, при этом индукцию нужно разделить пополам!

$$\text{N087 ETD29 0.5мм 100кГц: } P_v = V \cdot P_{vf} = 5.35 \cdot 0.180 = 0.96\text{Вт},$$

$$\text{N087 ETD29 1.0мм 100кГц: } P_v = V \cdot P_{vf} = 5.35 \cdot 0.100 = 0.54\text{Вт},$$

$$\text{N087 ETD34 0.5мм 100кГц: } P_v = V \cdot P_{vf} = 7.63 \cdot 0.100 = 0.76\text{Вт}.$$

5.9. Диаметр проводов.

Длина намотки первичной катушки — $21 - 3 \cdot 2 = 15\text{мм}$. Средний диаметр намотки 15мм , 50 витков — 2.4м . При использовании одинарного провода 0.5мм (погонное сопротивление с учетом скин-эффекта $\approx 0.10\Omega\text{м}$).

$$P_{w1} = I_1 \cdot I_1 \cdot R = 0.63 \cdot 0.63 \cdot 0.1 \cdot 2.4 = 0.1\text{Вт}.$$

Вторичная обмотка — 5 витков в 6 проводов ($k=6$) 0.5мм . Средний диаметр намотки 18мм , длина 34см . Потери в проводе:

$$P_{w1} = I_2 \cdot I_2 \cdot R_{\Pi} \cdot l/k = 7.0 \cdot 7.0 \cdot 0.1 \cdot 0.34/6 = 0.28\text{Вт}.$$

Рассеиваемая мощность в худшем случае: 0.92Вт , нагрев = 26°C .

6. Выбор силового транзистора

Напряжение сток-исток — 800В и минимальная емкость Миллера. Статические потери в транзисторе:

$$P_{ct} = R_{ci} \cdot I_1^2, \quad (R_{ci} \text{ берем для } 60^\circ).$$

$$\text{Для BUK456-800A: } P_{ct} = 3.5 \cdot 0.63 \cdot 0.63 = 1.39\text{Вт},$$

$$\text{Для 2SK2605: } P_{ct} = 2.7 \cdot 0.63 \cdot 0.63 = 1.07\text{Вт},$$

$$\text{Для 2SK2717: } P_{ct} = 2.5 \cdot 0.63 \cdot 0.63 = 0.99\text{Вт}.$$

7. Резистор датчика тока

$$R = 0.9\text{В}/I_{1и} = 0.9\text{В}/1.62\text{А} = 0.55\Omega,$$

(0.9В — минимальный порог датчика тока UC3842ADX). Используем три 1206 1.6Ω параллельно. Рассеиваемая мощность:

$$I_1 \cdot I_1 \cdot R = 0.63\text{А} \cdot 0.63\text{А} \cdot 0.55\Omega = 0.22\text{Вт}.$$

8. Резисторы питания UC3842ADX Fairchild

Максимальный стартовый ток — 0.4mA .

$$U = U_{bxmin} - U_{старт} = 110\text{В} - 16\text{В} = 94\text{В},$$

$$R = 94\text{В}/0.5\text{mA} = 235\text{k}\Omega.$$

Учитывая максимальное напряжение (500В), выбираем $2 \times 110\text{k}\Omega$. Рассеиваемая мощность для худшего случая, $U_{\text{сети}max} = 400\text{В}$,

$$U_{\text{вх}max} = \sqrt{2} \cdot 400\text{В} = 565\text{В}$$

Для $2 \times 110\text{k}\Omega$ — 1.45Вт.

9. Защита от перенапряжений

Максимальное рабочее напряжение Блока 220–12 — 300В. При более высоком напряжении питающей сети силовой транзистор должен переходить в выключенное состояние. Для этого используем вход усилителя ошибки (вывод 2 UC3842ADX).

$$U_{X\text{ вх}max} = \sqrt{2} \cdot 300\text{В} = 425\text{В}.$$

Пороговое напряжение $U_{\text{выб}2} = 2.42 - 2.58\text{В}$. Делитель $= 425/2.42 = 176$. Резисторы делителя имеют второе назначение — разряд входного конденсатора. Учитывая максимальное напряжение резисторов (350В), используем два 0.5Вт-ных резистора, общую мощность для надежности ограничим 0.5Вт. Получаем для $U_{\text{вх}max} = 565\text{В}$.

$$I = 0.5\text{Вт}/565\text{В} = 0.88\text{mA}, \quad R = 642\text{k}\Omega.$$

Допустимо использовать два резистора 0.5Вт от 330кОм до 1М. При сопротивлении на землю 6.8кОм — резисторы $6.8 \times 176 = 680\text{k}\Omega$.

10. Защита от пониженного напряжения

При пониженном напряжении скважность импульсов становится максимальной и происходит уменьшение выходного напряжения.

$$\text{зап} = 145/(145 + 180 - 2) = 0.45.$$

Такая скважность получается при $R_g = 680\text{Ом}$. Емкость конденсатора при этом 11нФ (для 100кГц).

11. Защитная регулировка

В случае отказа оптронной цепи обратной связи, выходное напряжение должно регулироваться по питанию микросхемы UC3842ADX, примерно 20В Для этого между питанием КА3882 и ее 3-им выводом (вход датчика тока) — стабилитрон на 20В.

12. Выбор выходного диода

Обратное напряжение на диоде:

$$U_d = U_H + (U_{\text{вх}max} + U_{\text{доп}}) \cdot W_2/W_1,$$

$$U_d = 13.8 + (425 + 145) \cdot 5/50 = 71\text{В}.$$

Статические потери в диоде:

$$P_d = U_{pd} \cdot I_2 = 0.6 \cdot 7 = 4.2 \text{ Вт.}$$

Диод 43СТQ100.

13. Выбор выходных конденсаторов

$$I_c = \sqrt{I_2^2 - I_n^2} = \sqrt{7^2 - 3.6^2} = 6 \text{ А.}$$

105°C: 3 конденсатора 3300мкФ 35В или

Такой ток допускают 3 конденсатора 4700мкФ 25В

85°C: 2 конденсатора 3300мкФ 35В или

2 конденсатора 4700мкФ 25В

Размер конденсаторов: 18×36мм.

14. Выходной фильтр

Частота $F_{lc} \approx 1/10$ частоты преобразования, то есть 10кГц. $L = 1/(2\pi \cdot 10000 \cdot 2\pi \cdot 10000 \cdot C)$. Для 220мкФ — 1.2мкГн, 4.5 витка на гантеля Ф5мм, 5.7 витков на стержне 6×25мм.

15. Цепь основной обратной связи

Единичное усиление на частоте $F_o \approx F_{lc}/5$, то есть 2кГц

Используем TL431 и оптрон 817С. Эффективность цепи регулирования: Изменение тока светодиода оптрана на 0.2mA изменяет скважность от 0 до максимума, а значит выходную мощность от 0 до 50Вт, то есть U_H от 0В до 13.8В при $R_H = 3.8\Omega$.

Выходные конденсаторы на частоте 2кГц ослабляют: $(1 + 2\pi \cdot 2\text{kГц}/W_z)/(1 + 2\pi \cdot 2\text{kГц}/W_p)$, где $W_z = 1/ESR \cdot Co$, $W_p = 1/R_H \cdot Co$ Предполагаем ESR одного = 0.1Ом, трех — 0.03.

$$W_z = 1/0.03 \cdot 0.0099 = 3367, F_z = 540\text{Гц},$$

$$W_p = 1/3.8 \cdot 0.0099 = 27, F_p = 4\text{Гц}.$$

Ослабление на частоте 2кГц: $(1 + 3.7)/(1 + 465) = 1/100$.

Для компенсации этого ослабления ток оптрана должен меняться на 0.2mA при изменении выходного напряжения на 14В/100. То есть резистор оптрана — 700Ом. Резистор ставим 1кОм, а дополнительное усиление реализуется на TL431: компенсирующая цепочка 10нФ и 13кОм.

Дополнительный полюс цепи обратной связи — на частоте $F_o \cdot 5 \approx 10\text{кГц}$, Транзистор оптрана шунтируем конденсатором. Входное сопротивление UC3842ADX по выводу — около 50кОм, компенсирующий конденсатор — около 400пФ.

16. Цепь подстроечной обратной связи

Подстройка осуществляется с выхода PWM-а. Коэффициент усиления с выхода PWM на выход Блока 220–12 примерно равен 1.

B.2 Перечень элементов

№	Обозначение	Наименование	Корпус	Кол - во	Примечание
Микросхемы					
1	DA1	UC2842BD1	SO8	1	UC2842ADX
2	DA2	TL431ID	SO8	1	
3	DA3	LP2951CM	SO8	1	LP2951ACM
4	DD1	Attiny26-8SI	SO20	1	
Транзисторы					
5	VT1	2SK2717	TO220F	1	
6	VT2	BC847C	SOT23	1	BC847A, BC847B
7	VT3	IRF7424	SO8	1	IRF7416
Диоды					
8	VD1-VD4	1N4007	DO-41	4	1N4006
9	VD5	BZX55-C22	DO-35	1	
10	VD6,VD7	UF4007	DO-41	2	
11	VD8	43CTQ100	TO220AB	1	16CTQ100
12	VD9,VD10	LL4148	SOD80C(DL35)	2	
Светодиод					
13	HL1	L-934SGD	3мм	1	
Оптрон					
14	VU1	PC817B	DIP4	1	H11A817B
Конденсаторы					
15	C1	X2 B81130 0.1мкФ 275В		1	
16	C2,C3	100мкФ×350В	18×35мм	2	22×35мм
17	C4	K73-17 22нФ×400В		1	
18	C5,C7	0.012мкФ X7R	1206	2	

№	Обозначение	Наименование	Корпус	Кол - во	Примечание
19	C6,C9,C11 C19,C21-C25	0.1мкФ X7R	1206	9	
20	C8,C18	220мкФ×35В		2	
21	C10	470пФ X7R	0805	1	
22	C12	1.0мкФ X7R	1205	1	
23	C13	Y1 DE1E3KX102M 1000пФ		1	K15-5 1000пФ3кВ
24	C16,C17	3300мкФ×35В	18 × 36ММ	2	4700мкФ×25В 18×36ММ
25	C20	10мкФ×25В		1	
Резисторы					
26	R1,R2	680кОм 5% 0.5вт		2	
27	R3,R16,R27, R35	6.8кОм 5% 0.125Вт	1206	4	
28	R4,R5	91кОм 5% 1Вт		2	
29	R6,R7	51кОм 5% 2Вт		2	
30	R8	820ом 5% 0.125Вт	1206	1	
31	R9,R23,R24	22ом 5% 0.125Вт	1206	3	
32	R10,R11,R15	1кОм 5% 0.125Вт	1206	3	
33	R12-R14	1.6ом 5% 0.125Вт	1206	3	
34	R17,R20,R21, R22,R28	13кОм 5% 0.125Вт	1206	5	
35	R18	27кОм 5% 0.125Вт	1206	1	
36	R19,R27,R35	6.8кОм 5% 0.125Вт	1206	3	
37	R25,R26, R34,R36	68кОм 5% 0.125Вт	1206	4	
38	R31-R33, R37-R41	220ом 5% 0.125Вт	1206	8	
Терморезистор					
39	RK1	SCK102 10ом 2А		1	
Трансформатор					
40	T1	B66358-G500-X167	ETD34	2	сердечник
		B66359-A2000	ETD34	2	стяжка

№	Обозначение	Наименование	Корпус	Кол - во	Примечание
		B66359-B1013	ETD34	1	каркас
Дроссели					
41	L1	82732-R2901-B30 2x27мГн 0.9А		1	
42	L2			1	
43	L3	EC24-101К 100мкГн		1	
Предохранители					
44	F1	H520PT-3A/250В		1	
45	FU2	MF-R500 5А		1	MF-R600
Кнопка					
46	S1	TS-A3PV		1	
Разъемы					
47		Mini-Fit MF-4F		1	
48		Провод сетевой с вилкой		1	
49		Диэлектрическая прокладка для ТО-220		1	
50		Стойка печатной платы РCHSN-6		4	
51		Винт M3×8 DIN965		4	

B.3 Принципиальная схема

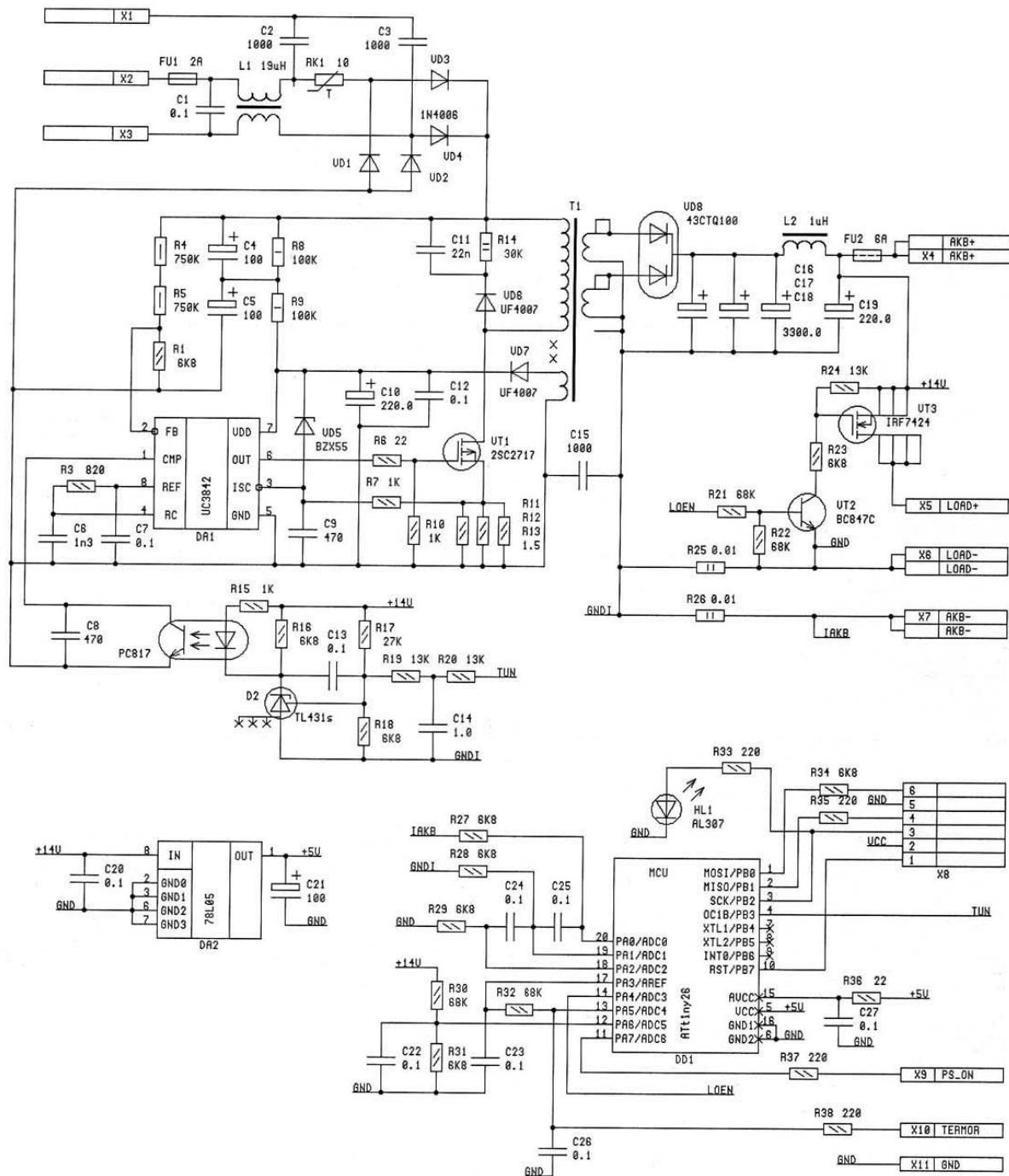


Рис. В.1: Принципиальная схема Блок 220–12

Приложение С

Программы микроконтроллера Блока 220–12

Программа управления Блоком 220–12 реализует следующие функции:

- контроль потребляемого тока, защита от перегрузки;
- подстройка напряжения зарядки аккумулятора в зависимости от температуры;
- отключение нагрузки при разряде аккумулятора;
- индикация режима работы;
- передача информации о состоянии устройства в управляющий интерфейс (обычно подключается к ПК-маршрутизатору), прием из интерфейса команд ручного управления.

Ниже каждой из перечисленных функций дается подробное описание.

C.1 Контроль потребляемого тока и защита от перегрузки

Для контроля тока нагрузки используется 10-разрядный АЦП микроконтроллера ATtiny26. Измерение тока нагрузки производится с периодом около 2 мс. В случае, если ток превышает заданное в программе значение (3А), происходит переход в режим “перегрузка”. Нагрузка отключается на 60 с, после чего подключается снова. Если причина перегрузки не устранена, в течение 2 мс произойдет возврат в режим “перегрузка” и отключение нагрузки еще на 60 с, и так далее.

C.2 Подстройка напряжения зарядки аккумулятора

Для измерения температуры аккумулятора используется термистор, подключенный к одному из входов 10-разрядного АЦП микроконтроллера ATtiny26. Измерение температуры производится с периодом около 6 мс. По результатам измерения производится коррекция выходного напряжения в соответствии с табл. С.1.

Таблица С.1: Коррекция выходного напряжения в зависимости от температуры аккумулятора

температура, °C	напряжение, В	температура, °C	напряжение, В
50	13.1	-5	14.5
40	13.2	-10	14.6
35	13.3	-15	14.75
30	13.4	-20	14.9
25	13.6	-25	15.0
20	13.7	-30	15.2
15	13.8	-35	15.35
10	14.0	-40	15.50
5	14.2	термистор не подключен	13.6
0	14.4		

При считывании с АЦП значения в диапазоне 0x3e0..0x3ff делается вывод о том, что термистор отключен. В этом случае подстройка прекращается, выходное напряжение устанавливается равным 13.6 В.

C.3 Отключение нагрузки при разряде аккумулятора

Данный раздел описывает управление нагрузкой в автоматическом режиме.

Для контроля напряжения батареи используется один из входов 10-разрядного АЦП микроконтроллера ATtiny26. Измерение напряжения батареи производится с периодом около 6 мс. В случае, если напряжение батареи в течение 5 секунд держится ниже порога отключения, производится отключение нагрузки.

Порог отключения выбирается с учетом тока нагрузки по табл. С.2.

Включение нагрузки производится не ранее чем через 60с, и при условии, что напряжение батареи не ниже 11 В.

Таблица С.2: Коррекция выходного напряжения в зависимости от тока нагрузки

Ток нагрузки	Порог отключения
$\geq 1.0 \text{ A}$	9.2 В
$\geq 0.6 \text{ A}$	9.6 В
$\geq 0.4 \text{ A}$	9.9 В
$\geq 0.2 \text{ A}$	10.2 В
$\geq 0.1 \text{ A}$	10.4 В
$< 0.1 \text{ A}$	10.6 В

C.4 Индикация режима работы Блока 220–12

Режимы работы индицируются миганием зеленого светодиода, установленного на Блоке 220–12:

- **Нормальное состояние:** питание от сети 220 В, возможно происходит подзарядка аккумулятора. Светодиод мигает с периодом 2 с и коэффициентом заполнения 1.95 (*погасает раз в 2 с на 0.1 с*).
- **Питание от батарей:** напряжение 220 В отсутствует, питание нагрузки осуществляется от аккумулятора. Светодиод мигает с периодом 2 с и коэффициентом заполнения 0.05 (*загорается раз в 2 с на 0.1 с*).
- **Батарея разряжена:** напряжение 220 В отсутствует, нагрузка отключена, чтобы предотвратить глубокий разряд батареи. Светодиод мигает с периодом 5 с и коэффициентом заполнения 0.02 (*загорается раз в 5 с на 0.1 с*).
- **Напряжение 220 В отсутствует, батарея отсутствует или разряжена до напряжения ниже 3.7...4.2 В.** При этом микроконтроллер останавливается в результате срабатывания цепи контроля питания (brown-out detector). *Светодиод не горит.*
- **Перегрузка:** нагрузка отключена. Светодиод мигает с периодом 1/7 с и коэффициентом заполнения 0.5 (*частое мигание*).

C.5 Управляющий интерфейс

Управляющий интерфейс представляет собой последовательный интерфейс (RS-232 без сигналов управления модемом), работающий на скорости 1200 бит/с.

При работе Блока 220–12 постоянно производится выдача в интерфейс текстовой строки следующего вида:

SSSSSSSS M BBB/RRR TTT LLL 000 VV v1

где

SSSSSSSS время в секундах, прошедшее с момента последнего включения блока 220–12
(с момента старта микропрограммы)

M режим:

- 0 — питание от сети (нормальное состояние);
- 1 — питание от батарей;
- 2 — батарея разряжена;
- 3 — перегрузка.

BBB напряжение батареи (данные АЦП).

В режиме 0 (питание отсети) это приложенное к батарее зарядное напряжение (выходное напряжение Блока 220–12).

В режимах 1 и 2 это напряжение батареи, по которому можно определить степень ее разряда.

RRR показания АЦП, соответствующие напряжению батареи 12В (данные калибровки, см. раздел С.6). Напряжение батареи в вольтах можно вычислить по формуле $V = 12 \times BBB / RRR$.

TTT температура аккумулятора (данные АЦП). Температуру в градусах Цельсия можно определить по таблице:

TTT	T°C	TTT	T°C	TTT	T°C
0x3b8 (952)	-40	0x26a (618)	-5	0xfc (252)	+25
0x39f (927)	-35	0x224 (548)	0	0xd3 (211)	+30
0x37f (895)	-30	0x1df (479)	+5	0xad (173)	+35
0x356 (854)	-25	0x19d (413)	+10	0x92 (146)	+40
0x326 (806)	-20	0x161 (353)	+15	0x7b (123)	+45
0x2ed (749)	-15	0x12c (300)	+20	0x66 (102)	+50
0x2ae (686)	-10				

LLL ток нагрузки (данные АЦП). Ток нагрузки в амперах можно вычислить по формуле $A = 3 \times LLL / 0xb1$. (Константа $0xb1$ получена путем калибровочных измерений, которые проводятся один раз для каждой новой партии печатных плат).

000 ток последней перегрузки: значение LLL, которое вызвало переход бустового устройства в режим “перегрузка”.

VV текущее значение широтно-импульсного модулятора, управляющего цепью обратной связи стабилизатора напряжения (п. 1.2). Зависимость обратная: большие значения VV соответствуют меньшим значениям выходного напряжения.

v индикатор ручного управления широтно-импульсным модулятором (п. 1.5.3). Может принимать следующие значения:

- = автоматическое управление напряжением;
- < ручное управление, установлено напряжение ниже, чем установил бы автомат;
- > ручное управление, установлено напряжение выше, чем установил бы автомат.

1 индикатор ручного управления подключением нагрузки. (п. 1.5.3). Может принимать следующие значения:

- a автоматическое управление подключением нагрузки (состояние нагрузки можно определить по полю M);
- + ручное управление, нагрузка включена независимо от состояния батареи;
- ручное управление, нагрузка выключена независимо от состояния батареи.

Все числа в описанном формате шестнадцатеричные.

Управляющий интерфейс позволяет управлять работой Блока 220–12. Команды управления имеют вид одиночных ASCII-символов, посылаемых Блоку 220–12 через интерфейс RS-232. Блок воспринимает следующие команды:

- < перейти в режим ручного управления выходным напряжением, понизить напряжение на один шаг по сравнению с текущим;

- > перейти в режим ручного управления выходным напряжением, повысить напряжение на один шаг по сравнению с текущим;
- = вернуться в режим автоматического управления выходным напряжением;
- + перейти в режим ручного управления подключением нагрузки, включить нагрузку вне зависимости от состояния батареи;
- перейти в режим ручного управления подключением нагрузки, выключить нагрузку вне зависимости от состояния батареи;
- а вернуться в режим автоматического управления подключением нагрузки.

C.6 Калибровка

10-разрядный АЦП микроконтроллера ATtiny26 используется для измерения следующих параметров:

- ток нагрузки
- ток заряда батареи
- температура аккумулятора
- напряжение батареи

Из них только последний параметр (напряжение батареи) требует хорошей точности измерения, и только для него используется процедура калибровки.

Калибровка также требуется для цепи обратной связи стабилизатора, через которую производится управление выходным напряжением в зависимости от температуры аккумулятора.

Процедура калибровки состоит из следующих шагов:

- подключить к Блоку 220–12 балластную нагрузку с сопротивлением 100 Ом, мощностью не менее 5 Вт;
- подключить параллельно нагрузке вольтметр;
- запустить программу калибровки командой: `./calibrate.pl -force`;
- программа запрашивает текущие показания вольтметра, и сохраняет в качестве результата калибровки в EEPROM микроконтроллера пару (V_{ref} , V_{adc}), где V_{ref} – измеренное вольтметром напряжение на нагрузке, V_{adc} – соответствующий этому напряжению 10-разрядный код АЦП.

- программа использует только что откалибранный АЦП для определения “средней точки” широтно-импульсного модулятора, управляющего цепью обратной связи стабилизатора. Производится последовательный поиск в интервале [0..255], в качестве средней точки выбирается значение ширины импульса, при котором выходное напряжение равно 13.6 В. По завершении поиска программа предлагает оператору проверить, что вольтметр показывает значение, близкое к 13.6 В (типичное отклонение $\pm 0.03\text{В}$).

Результаты калибровки сохраняются в EEPROM микроконтроллера ATtiny26, и используются при прошивке микропрограммы в микроконтроллер. Прошивка новой микропрограммы не требует повторной калибровки, используются данные калибровки, сохраненные в EEPROM.

Перед прошивкой микропрограмма перекомпилируется, поскольку необходимо заново вычислить константные выражения, использующие результаты калибровки.

Приложение D

Блок 12–АТХ

D.1 Расчеты характеристик

D.1.1 Индуктивности

Индуктивность:

$$L_{min} = (V_{in} - V_{out}) \cdot V_{out} / (V_{in} \cdot \Delta I_l \cdot F_s),$$

$$L3_{min} = 11 \cdot 5/16 \cdot 2 \cdot 200000 = 8.6\text{мкН},$$

$$L4_{min} = 13 \cdot 3/16 \cdot 2 \cdot 200000 = 6\text{мкН}.$$

Сердечник: два склеенных кольца МП-60 К19×11×4.8.

Пробная намотка: 28 витков — 49мкГн. Длина провода 0.56 — 75см; длина витка — 26.8мм, длина витка провода 0.85мм ≈ 27.7мм.

Для получения около 9мкГн нужно 12 витков; длина провода 0.85мм — 33см. Удельное сопротивление провода — 0.03Ом/м (20грЦ) 0.04Ом/м (100грЦ), Сопротивление обмотки с сдвоенным проводом — 5–7 мОм. Потери при токе 10А — 0.5–0.7 Вт ($R \cdot I$).

Вторичная обмотка:

падение на 79|12 — 2В, падение на диоде — 0.7В, таким образом, на вторичной обмотке должно быть около 15В. $15/5=3$. При 12 витках — 36, при 15 — 45.

D.1.2 Конденсаторы

Минимальная емкость фильтра выпрямителя отрицательных напряжений:

$$C = I/U_f = 0.2/(0.5 \cdot 200000) = 1\text{мкФ} .$$

Конденсаторы преобразователей 12В–5В и 12В–3.3В:

Макс.ток входных: $1.2 \cdot I_{out} \cdot U_{out} / U_{in} = 3.8\text{А}$ для 3.3В, 5.7А для 5В. Макс.ток выходных: $1.5 \cdot \Delta I_l \approx 3\text{А}$.

Конденсаторы tantalовые тип E HitachiAIC:

- 35мкФ × 25В — 0.2Ом, 1.1А (при 65°C — 2.2А),
- 100мкФ × 16В — 0.1Ом, 1.1А (при 65°C — 2.2А),
- 220мкФ × 10В — 0.1Ом, 1.1А (при 65°C — 2.2А).

Конденсаторы tantalовые тип E EPCOS:

- 68мкФ × 25В — 0.2Ом, 0.9А (при 65°C — 1.8А),
- 330мкФ × 10В — 0.1Ом, 1.3А (при 65°C — 2.6А).

Конденсаторы электролитические 10x20mm Jamicon:

- 470мкФ × 25В — 0.11Ом, 0.9А (при 65°C — 1.8А),
- 680мкФ × 16В — 0.09Ом, 0.9А (при 65°C — 1.8А).

Три конденсатора гарантированно обеспечивают входной и выходной ток. Остальные конденсаторы – типа X7R.

D.1.3 Расчет цепи коррекции, R_c, C_c

3.1 для С=220мкФ.

$$\begin{aligned} L &= 9\text{мкГн}, \\ C &= 220\text{мкФ} \cdot 3, \\ ESR &= 0.1\text{ом}/3, \\ F_s &= 200\text{кГц}, \\ gm &= 600 \cdot 10^{-6}, \\ F_{lc} &= 1/(2\pi \cdot \sqrt{LC}) = 2.07\text{кГц}, \\ F_z &= 0.75 \cdot F_{lc} = 1.55\text{кГц}, \\ F_{esr} &= 1/(2\pi \cdot ESR \cdot C) = 7.2\text{кГц}, \\ F_{esr} &< F_o < F_s/5...F_s/10, \\ F_o &= 10...20\text{кГц}. \end{aligned}$$

Для преобразователя 12В–5В:

$$\begin{aligned} R_c &= 1.25/V_{in\ max} \cdot F_o \cdot F_{esr}/F_{lc}^2 \cdot U_{out}/1.25 \cdot 1/gm, \\ R_c &= 0.078 \cdot 10...20 \cdot 1.68 \cdot 6666 = 8.7...17.5\text{kОм}, \\ C_c &= 1/(F_z \cdot 2\pi \cdot R_c) = 11700...5800\text{nФ}. \end{aligned}$$

Для преобразователя 12В–3.3В:

$$R_c = 1.25/V_{in\ max} \cdot F_o \cdot F_{esr}/F_{lc}^2 \cdot U_{out}/1.25 \cdot 1/gm,$$

$$R_c = 0.078 \cdot 10...20 \cdot 1.68 \cdot 4400 = 5.7...11.5\text{k}\Omega,$$

$$C_c = 1/(F_z \cdot 2\pi \cdot R_c) = 17700...8800\text{n}\Phi.$$

Для уменьшения номенклатуры изделий делаем компенсирующие цепочки одинаковыми: 10000пФ и 10кОм

3.2 для C=100мкФ.

$$L = 9\text{мкГн},$$

$$C = 100\text{мкФ} \cdot 3,$$

$$ESR = 0.1\text{Ом}/3,$$

$$F_s = 200\text{кГц},$$

$$gm = 600 \cdot 10^{-6},$$

$$F_{lc} = 1/(2\pi \cdot \sqrt{LC}) = 3.06\text{кГц},$$

$$F_z = 0.75 \cdot F_{lc} = 2.3\text{кГц},$$

$$F_{esr} = 1/(2\pi \cdot ESR \cdot C) = 15.9\text{кГц},$$

$$F_{esr} < F_o < F_s/5...F_s/10,$$

$$F_o = 20...40\text{кГц}.$$

Для преобразователя 12В–5В:

$$R_c = 1.25/V_{in\ max} \cdot F_o \cdot F_{esr}/F_{lc}^2 \cdot U_{out}/1.25 \cdot 1/gm,$$

$$R_c = 0.078 \cdot 20...40 \cdot 1.68 \cdot 6666 = 17.5...35\text{k}\Omega,$$

$$C_c = 1/(F_z \cdot 2\pi \cdot R_c) = 3900...1970\text{n}\Phi.$$

Для преобразователя 12В–3.3В:

$$R_c = 1.25/V_{in\ max} \cdot F_o \cdot F_{esr}/F_{lc}^2 \cdot U_{out}/1.25 \cdot 1/gm,$$

$$R_c = 0.078 \cdot 20...40 \cdot 1.68 \cdot 4400 = 11.5...23\text{k}\Omega,$$

$$C_c = 1/(F_z \cdot 2\pi \cdot R_c) = 6000...3000\text{n}\Phi.$$

Для уменьшения номенклатуры изделий делаем компенсирующие цепочки одинаковыми: 4700пФ и 15кОм.

Плавный старт. 0.1мкФ — 7.5мсек; 1.0мкФ — 75мсек.

Включение Блока 12–ATX: сопротивление ключа 0.02ом, емкость затвора 1700пФ. Задержка в затворе — 10 мксек максимум. Задержка входных конденсаторов Блока 12–ATX — $600\text{мкФ} \times 0.02\text{Ом} = 12$ мксек максимум.

D.1.4 КПД для преобразователя 12В–3.3В при токе 10А (мощность на нагрузке 33Вт)

Потери в верхнем транзисторе:

$$\begin{aligned} I_n \cdot I_n \cdot R_d \cdot U_{out}/U_{in} + & 10 \cdot 10 \cdot 0.011 \cdot 3.3/12 = 0.3 \\ I_n \cdot Q_{sw} \cdot U_{in} \cdot f/I_d + & 10 \cdot 6.7 \cdot 10^{-9} \cdot 12 \cdot 200000/0.5 = 0.32 \\ Q_g \cdot U_g \cdot f + & 14 \cdot 10^{-9} \cdot 5 \cdot 200000 = 0.02 \\ Q_{oss} \cdot U_{in} \cdot f/2 & 8 \cdot 10^{-9} \cdot 12 \cdot 100000 = 0.01, \quad \text{итого } 0.65 \text{ Вт} \end{aligned}$$

Потери в нижнем транзисторе:

$$\begin{aligned} I_n \cdot I_n \cdot R_d \cdot (U_{in} - U_{out})/U_{in} + & 10 \cdot 10 \cdot 0.007 \cdot 8.7/12 = 0.507 \\ Q_g \cdot U_g \cdot f + & 20 \cdot 10^{-9} \cdot 12 \cdot 200000 = 0.048 \\ Q_{oss} \cdot U_{in} \cdot f/2 + & 6 \cdot 10^{-9} \cdot 12 \cdot 100000 = 0.007 \\ Q_{rr} \cdot U_{in} \cdot f & 13 \cdot 10^{-9} \cdot 12 \cdot 200000 = 0.031, \quad \text{итого } 0.59 \text{ Вт} \\ \text{КПД} = 33 / (33+0.65+0.59) & = 96\% \end{aligned}$$

КПД для преобразователя 12В–3.3В при токе 3А (мощность на нагрузке 9.9Вт)

Потери в верхнем транзисторе:

$$\begin{aligned} I_n \cdot I_n \cdot R_d \cdot U_{out}/U_{in} + & 3 \cdot 3 \cdot 0.0095 \cdot 3.3/12 = 0.024 \\ I_n \cdot Q_{sw} \cdot U_{in} \cdot f/I_d + & 3 \cdot 3.7 \cdot 10^{-9} \cdot 12 \cdot 200000/0.5 = 0.054 \\ Q_g \cdot U_g \cdot f + & 10 \cdot 10^{-9} \cdot 5 \cdot 200000 = 0.01 \\ Q_{oss} \cdot U_{in} \cdot f/2 & 6 \cdot 10^{-9} \cdot 12 \cdot 100000 = 0.0072, \quad \text{итого } 0.095 \text{ Вт} \end{aligned}$$

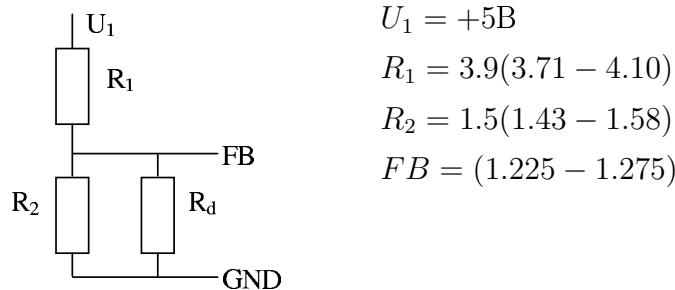
Потери в нижнем транзисторе:

$$\begin{aligned} I_n \cdot I_n \cdot R_d \cdot (U_{in} - U_{out})/U_{in} + & 3 \cdot 3 \cdot 0.007 \cdot 8.7/12 = 0.046 \\ Q_g \cdot U_g \cdot f + & 20 \cdot 10^{-9} \cdot 12 \cdot 200000 = 0.048 \\ Q_{oss} \cdot U_{in} \cdot f/2 + & 6 \cdot 10^{-9} \cdot 12 \cdot 100000 = 0.007 \\ Q_{rr} \cdot U_{in} \cdot f & 13 \cdot 10^{-9} \cdot 12 \cdot 200000 = 0.031, \quad \text{итого } 0.132 \text{ Вт} \\ \text{КПД} = 9.9 / (9.9+0.095+0.132) & = 97.7\% \end{aligned}$$

D.1.5 Расчет делителя обратной связи для IRU3037

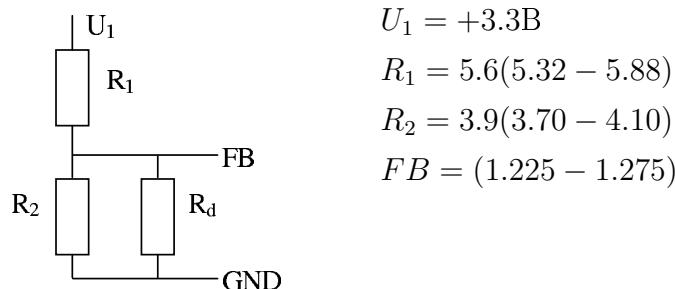
5В Расчет: при FB_{max} , 5В получается при делителе 2.922:1. Учитывая отклонения резисторов получаем минимальный делитель 2.63.

Ближе всего из стандартного ряда 3.9/1.5:



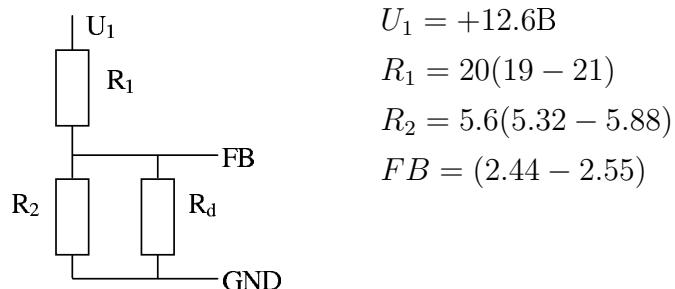
Минимальный R_{don} . При $FB = 1.275$, $3.71/1.58$, получаем 4.27В, Для получения 5В: $1.275/((5-1.275)/3.71-1.275/1.58) = 1.275/(1.00-0.81) = 6.6\text{k}\Omega$ Для общего случая $R_{don} = 1.25 \cdot 3.9/(5 - U)$. Достигимая точность — около 1%.

3.3В Расчет: при FB_{max} , 3.3В получается при делителе 1.59:1. Учитывая отклонения резисторов получаем минимальный делитель 1.44. Ближе всего 5.6/3.9:



Минимальный R_{don} . При $FB = 1.275$, $5.32/4.10$, получаем 2.93В, Для получения 3.3В: $1.275/((3.3-1.275)/5.32-1.275/4.10) = 1.275/(0.38-0.31) = 18.2\text{k}\Omega$ Для общего случая $R_{don} = 1.25 \cdot 5.6/(3.3 - U)$. Достигимая точность — около 1%.

12.6В Расчет: при FB_{max} , 12.6В получается при делителе 3.94:1. Учитывая отклонения резисторов получаем минимальный делитель 3.58. Ближе всего 20/5.6:



Минимальный $R_{\partial on}$. При $FB = 2.55, 19/5.88$, получаем 10.79В, Для получения 12.6В: $2.55/((12.6-2.55)/19-2.55/5.88)=2.55/(0.529-0.434)=26.8\text{кОм}$

Для общего случая $R_{\text{доп}} = 2.5 \cdot 20/(12.6 - U)$. Достигимая точность — около 1%.

D.1.6 Потери в верхнем транзисторе

7811AV

$I_n \cdot I_n \cdot R_d \cdot U_{out}/U_{in} +$	$3 \cdot 3 \cdot 0.011 \cdot 3.3/12 = 0.027$
$I_n \cdot Q_{sw} \cdot U_{in} \cdot f/I_d +$	$3 \cdot 6.7 \cdot 10^{-9} \cdot 12 \cdot 200000/0.5 = 0.096$
$Q_g \cdot U_g \cdot f +$	$14 \cdot 10^{-9} \cdot 5 \cdot 200000 = 0.02$
$Q_{oss} \cdot U_{in} \cdot f/2$	$8 \cdot 10^{-9} \cdot 12 \cdot 100000 = 0.01$, итого 0.153 Вт

8113

$I_n \cdot I_n \cdot R_d \cdot U_{out}/U_{in} +$	$3 \cdot 3 \cdot 0.0058 \cdot 3.3/12 = 0.014$
$I_n \cdot Q_{sw} \cdot U_{in} \cdot f/I_d +$	$3 \cdot 8.5 \cdot 10^{-9} \cdot 12 \cdot 200000/0.5 = 0.124$
$Q_g \cdot U_g \cdot f +$	$22 \cdot 10^{-9} \cdot 5 \cdot 200000 = 0.022$
$Q_{oss} \cdot U_{in} \cdot f/2$	$14 \cdot 10^{-9} \cdot 12 \cdot 100000 = 0.017$, итого 0.177 Вт

7821, 7828

$I_n \cdot I_n \cdot R_d \cdot U_{out}/U_{in} +$	$3 \cdot 3 \cdot 0.0095 \cdot 3.3/12 = 0.024$
$I_n \cdot Q_{sw} \cdot U_{in} \cdot f/I_d +$	$3 \cdot 3.7 \cdot 10^{-9} \cdot 12 \cdot 200000/0.5 = 0.054$
$Q_g \cdot U_g \cdot f +$	$10 \cdot 10^{-9} \cdot 5 \cdot 200000 = 0.01$
$Q_{oss} \cdot U_{in} \cdot f/2$	$6 \cdot 10^{-9} \cdot 12 \cdot 100000 = 0.0072$, итого 0.095 Вт

D.2 Перечень элементов

№	Обозначение	Наименование	Корпус	Кол - во	Примечание
Микросхемы					
1	DA1	78L06ACD	SO8	1	
2	DA2, DA3	IRU3037CS	SO8	2	
3	DA4	TL431CD	SO8	1	
4	DA5	79L05ACD	SO8	1	
5	DA6	79L12ACD	SO8	1	
6	DA7	MAX809TEUR	SOT23	1	MAX809TD
Транзисторы					
7	VT1-VT4	IRF7828	SO8	4	IRF7821, IRF7811AV

№	Обозначение	Наименование	Корпус	Кол - во	Примечание
8	VT5	IRLR2705	DPAK	1	IRLR2905, IRLR2905
Диоды					
9	VD1-VD7	LL4148	SOD80C(DL35)	7	
Конденсаторы					
10	C1, C23	22мкФ25В	тип D	2	22мкФ35В
11	C4-C6, C11-C13	47мкф20В lowESR	тип D	6	33мкФ25В lowESR
12	C25-27, C33-C35	100мкф16В lowESR	тип D	6	220мкФ10В lowESR
13	C7, C8, C10, C14, C15, C17-C19, C24, C28, C30, C32	1.0мкФ25В X7R	1206	12	
14	C2, C3, C20-C22, C29, C31	0.1мкФ50В X7R	1206	7	
15	C9, C16	4700пФ50В X7R	0805	2	4700пФ25В X7R
Резисторы					
16	R1, R2	15кОм 5% 0.125Вт	1206	2	
17	R3, R8, R11	1,5кОм 5% 0.125Вт	1206	3	
18	R4	20кОм 5% 0.125Вт	1206	1	
19	R6, R7, R13	5,6кОм 5% 0.125Вт	1206	3	
Самовосстанавливающийся предохранитель					
20	F1	MF-SM100		1	
Разъемы					
21		Mini-Fit MF-20F		1	
22		Mini-Fit MF-4M		1	
23		Mini-Fit MF-2F		1	
24		TH-4F		1	

№	Обозначение	Наименование	Корпус	Кол - во	Примечание
Дроссели					
25	L1	3мГн K15x7x4.8 МП-140		1	
26	L3, L4	9мГн K19x11x4.8 МП-60		2	
27	L2			1	Доп.обмотка на L3
28		провод ПЭТВ-0.85			2.5м

D.3 Принципиальная схема

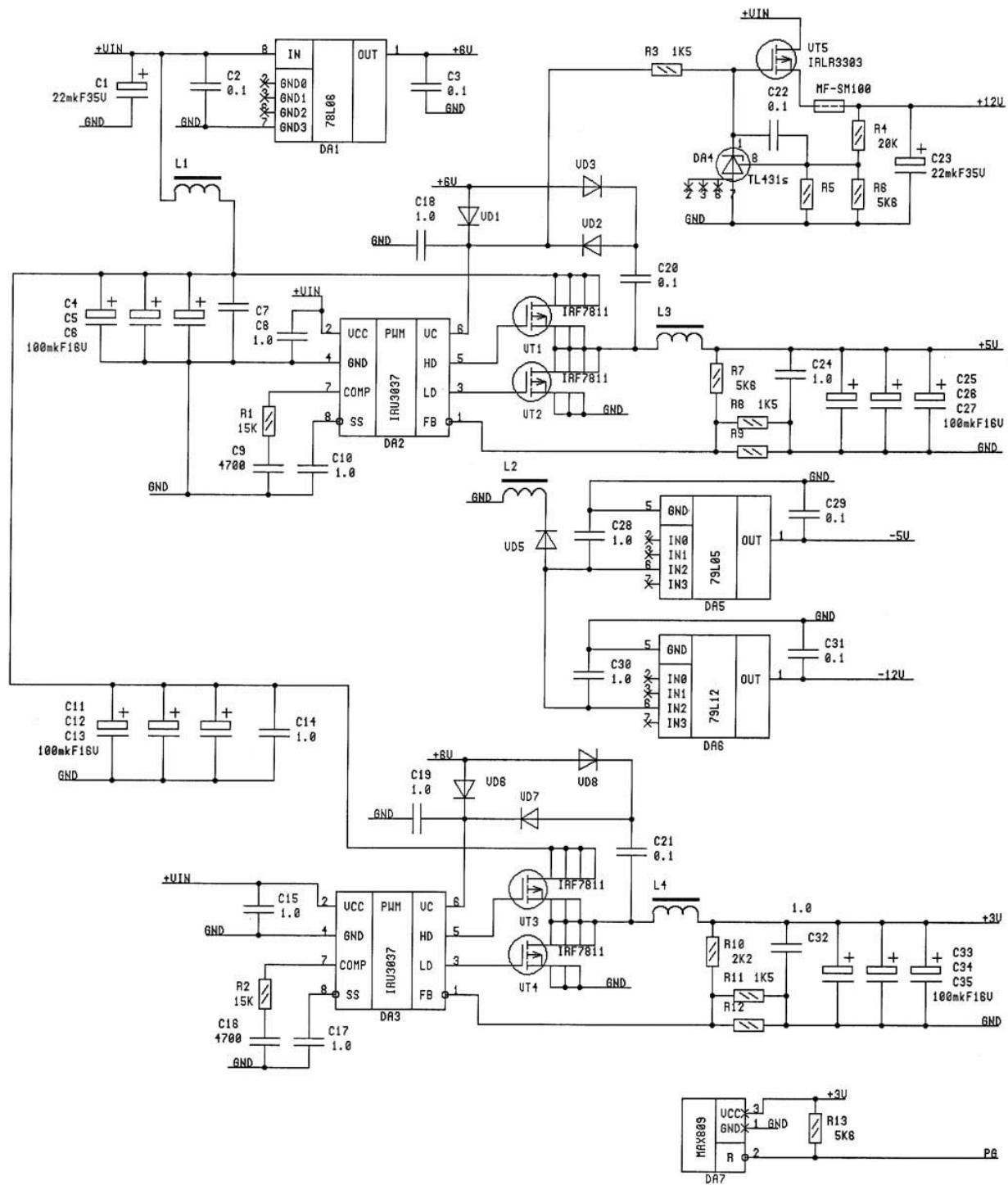


Рис. D.1: Принципиальная схема Блока 12-ATX

Приложение Е

Термоконтроллер

Термоконтроллер включает в себя следующие узлы:

- управляющий микроконтроллер;
- датчик температуры;
- датчик влажности;
- ключ подогревателя;
- ключ вентилятора;
- датчик нагрузки;
- стабилизатор напряжения питания.

Микроконтроллер. Основу термоконтроллера составляет микроконтроллер Attiny26 (DD1). Встроенный АЦП и аналоговый коммутатор микроконтроллера используются

Таблица Е.1: Технические характеристики термоконтроллера

Напряжение питания	4.75–30В
Ток потребления:	
– при отключенном подогревателе, не более	1mA
– при включенном подогревателе, не более	20mA
Температура окружающей среды	-40 / + 50°C
Точность измерения температуры в диапазоне 0+30	±1°C
Точность измерения температуры в диапазоне -40+50	±3°C
Точность измерения влажности	±2%

Таблица E.2: Предельные эксплуатационные данные термоконтроллера

Напряжение питания термоконтроллера	30В
Напряжение питания вентилятора	20В
Напряжение питания подогревателя	220В
Мощность подогревателя	200Вт
Ток вентилятора	0.5А

для измерения выходных напряжений датчиков, а выходные порты — для управления ключами подогрева и вентиляции. В качестве опорного напряжения АЦП выбрано напряжение питания его аналоговой части AVCC. Это несколько уменьшает точность АЦП (которая тем не менее остается на достаточном уровне), но при этом появляется возможность подключать датчики без внешнего делителя напряжения.

Датчик температуры. В качестве датчика температуры используется терморезистор RK1 типа B57891. Вместе с резистором R1 терморезистор образует делитель аналогового напряжения питания AVCC, подключенный к входу ADC3 встроенного в DD1 АЦП. Напряжение на этом входе меняется от $0.1 \cdot AVCC$ до $0.9 \cdot AVCC$ при изменении температуры от $+50^{\circ}\text{C}$ до -40°C . Базовая точность терморезистора в этом температурном диапазоне — $\pm 3^{\circ}\text{C}$, однако в наиболее важном диапазоне — от 0°C до $+30^{\circ}\text{C}$, ее можно повысить до $\pm 1^{\circ}\text{C}$, используя программную калибровку.

Датчик влажности. Выходное напряжение датчика влажности В1 типа НИ3610 линейно зависит от влажности воздуха, коэффициенты зависимости индивидуальны для каждого экземпляра и прилагаются в сопроводительном листке. Датчик подключен к входу ADC4 АЦП. Соответствующая корректировка осуществляется программно.

Ключ подогревателя состоит из симисторного оптрана VU1 и силового симистора VS1. В качестве силового используется симистор BT136-800, в зависимости от размеров радиатора симистора это позволяет коммутировать нагрузку от 200Вт до 1кВт. Нагрузка подключается к контактам X4 и X5, на контакты X6 и X7 подается напряжение 220В. Ключ подогревателя обеспечивает включение нагрузки при нулевом напряжении, это позволяет существенно снизить помехи коммутации. Для включения подогрева микроконтроллер DD1 выдает сигнал высокого уровня на выход PB5, светодиод оптрана начинает излучать, однакостроенная в оптрон схема переключения при нулевом напряжении позволит симистору оптрана открыться только в начале следующего полупериода сетевого напряжения. Открытый симистор оптрана подает напряжение на управляющий электрод силового симистора, который подключает нагрузку. Резистор

R14 ограничивает управляющий ток на безопасном уровне, резистор R13 обеспечивает надежное закрывание силового семистора. Цепочка C17 R17 предотвращает открывание VS1 внешней импульсной помехой.

Датчик нагрузки реализован на оптроне VS2. Резисторы R15 и R16 ограничивают ток светодиода оптрана на уровне 1-2mA на пиках сетевого напряжения, при этом фототранзистор оптрана открывается. Соответственно на вход PA1 микроконтроллера поступает сигнал частотой 100Гц. Датчик нагрузки подключен параллельно ключу подогревателя, это позволяет контролировать наличие 220В и исправность подогревателя (100Гц на входе PA0 при выключенном подогревателе) и исправность ключа подогревателя (пропадание сигнала 100Гц на входе PA0 при включении подогревателя).

Ключ вентилятора выполнен на транзисторе VT1. Вентилятор подключается к разъему X1, питание вентилятора подается на разъем X2. Для контроля тока потребления и частоты вращения вентилятора используется датчик тока на резисторах R6 и R7. Сигнал датчика тока поступает на входы встроенного в микроконтроллер компаратора, на вход PA7 непосредственно, а на вход PA6 — через интегрирующую цепочку R3 C1. На выходе встроенного компаратора выделяется переменная составляющая тока вентилятора, которая позволяет определить скорость его вращения. Кроме этого, используя встроенный в микроконтроллер АЦП, можно измерить ток потребления вентилятора и определить его исправность. Стабилитрон VD1 защищает транзистор ключа при коммутации импульсной нагрузки. Резисторы R2 и R4 защищают входы микроконтроллера при пробое ключа и замыкании в нагрузке. Резистор R5 обеспечивает закрытое состояние транзистора VT1 при инициализации и программировании микроконтроллера.

Управляющий интерфейс. Термоконтроллер имеет управ员ующий интерфейс, позволяющий считывать информацию с датчиков и управлять режимами работы. Также через разъем управляющего интерфейса осуществляется программирование и питание устройства. В разъем выведены сигналы программирования Reset, MOSI, MISO, SCK и сигналы последовательного интерфейса SCL и SDA, которые могут также использоваться как прием и передача RS-232. Резисторы R39 и R40 защищают входы от статического электричества и нормируют уровни RS-232 к стандартным логическим уровням. Для индикации режимов работы к выходу MOSI подключен светодиод HL1. При программировании он индицирует ответ микроконтроллера программатору.

Электропитание термоконтроллера. Термоконтроллер питается через управляющий интерфейс, это позволяет считывать данные с датчиков при отсутствии 220В, если используется Блок 220–12 с аккумулятором. Напряжение на контакте 6 управляющего интерфейса может находиться в пределах 4.75–16В, интегральный стабилизатор

напряжения DA1 стабилизирует напряжение питания микроконтроллера и датчиков на уровне 4.7В. Для получения такого напряжения параллельно встроенному в DA1 делителю обратной связи подключен внешний резистор R8. Конденсаторы C2 и C3 обеспечивают устойчивость стабилизатора, C4 снижает выходное сопротивление стабилизатора на высоких частотах. Фильтр L1 C5 развязывает по питанию цифровую и аналоговую части микроконтроллера.

E.1 Перечень элементов

№	Обозначение	Наименование	Корпус	Кол - во	Примечание
Микросхемы					
1	DA1	LP2951CM	SO8	1	LP2951ACM
2	DD1	Attiny26-8SI	SO20	1	
Транзистор					
3	VT1	IRLML2803	SOT23	1	
Тиристор					
4	VS1	BT136-800	TO-220	1	
Диод					
5	VD1	BZX55-C22	DO-35	1	
Светодиод					
6	HL1	L-934SGD	3мм	1	
Оптроны					
7	VU1	MOC3083	DIP6	1	
8	VU2	PC814A	DIP4	1	H11A814A
Конденсаторы					
9	C1, C2, C4, C5	0.1мкФ X7R	1206	9	
10	C6	K73-17 22нФx400В		1	
11	C4	10мкФx16В	типа	1	
Резисторы					
12	R1, R5	68кОм 5% 0.125Вт	1206	2	
13	R2, R4, R9-R13	220ом 5% 0.125Вт	1206	7	
14	R3, R8	1.3МОм 5% 0.125Вт	1206	2	
15	R6, R7	1.6ом 5% 0.125Вт	1206	2	
16	R14	330ом 5% 0.5Вт		1	

№	Обозначение	Наименование	Корпус	Кол - во	Примечание
17	R15, R16	91кОм 5% 1Вт		2	
18	R17	39ом 5% 0.5Вт		1	
Терморезистор					
19	RK1	B57891-М 223J 22кОм		1	
Датчик влажности					
20	B1	HIH3610		1	
Дроссель					
21	L3	EC24-101К 100мкГн		1	
Разъемы					
22	X1	WF-2R		1	
23	X2	Mini-Fit MF-2F		1	
24	X3	IDC BH-10		1	
25		Провод сетевой с вилкой		1	

E.2 Принципиальная схема

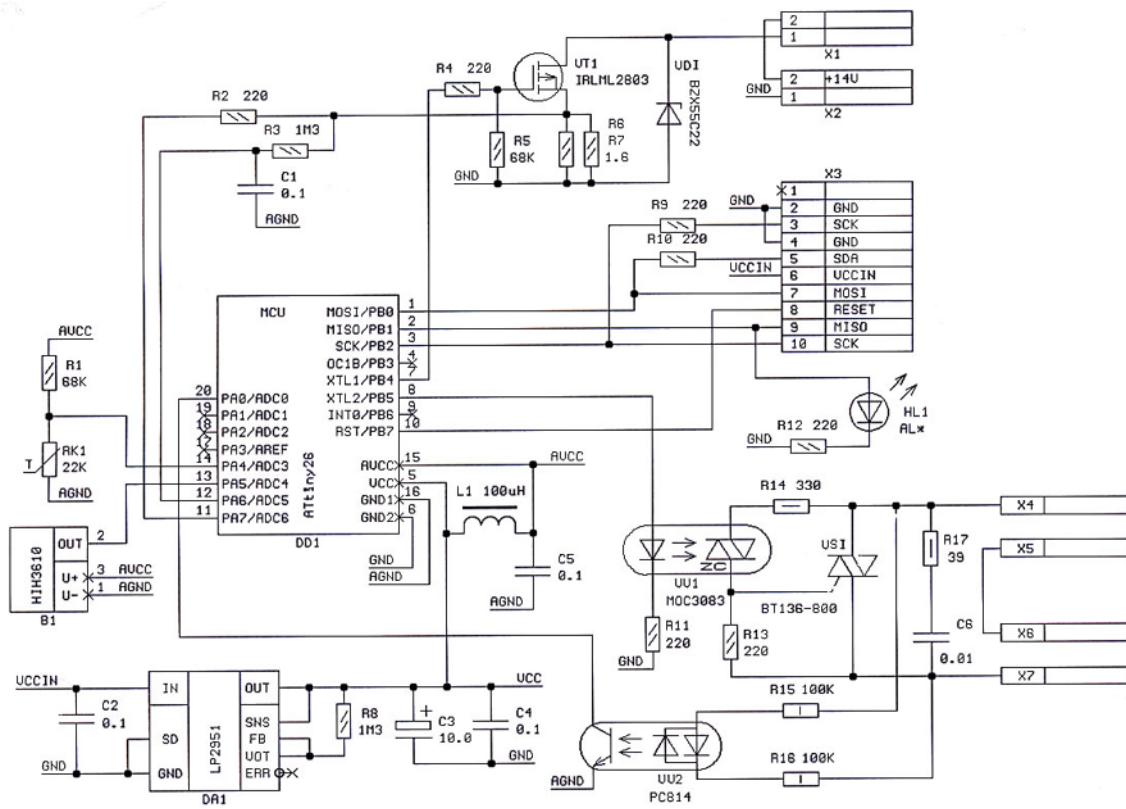


Рис. E.1: Принципиальная схема термоконтроллера