Инфокоммуникации

Преподаватель:

Колкер Алексей Борисович,

к.т.н. доцент каф. Автоматики

Курс рассчитан на 1 семестр,форма отчетности – зачет.

тел кафедры : 3461119

e-mail: fiery@ngs.ru

Инфокоммуникации:предмет

Предмет: (информационные сети и коммуникации)

Различные аспекты организации обмена данными между информационными системами при помощи компьютерных сетей.

Основы обеспечения информационной безопасности сетевых коммуникаций.

Литература

УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ

- Олифер Н.А., Олифер В.Г. Сети. - СПб.: Питер, 2001.- 560 с.
- Костромин В. А.//Самоучитель Linux для пользователя. - СПб.: БХВ-Петербург, 2003. - 672 с.: ил
- Колесниченко Д.Н.// Linux сервер своими руками.- СПб.:Наука и Техника,2002.-576с.:ил.
- Величко В.В и др. Телекоммуникационные системы и сети.Учебное пособие в 3х томах. Горячая линия-Телеком,2005. – 592с.: ил.
- Мур М. и др. Телекоммуникации. Руководство для начинающих.СПб.:БХВ-Петербург,2005. – 624 с: ил.

Литература(продолжение)

- Олифер Н.А., Олифер В.Г. Сетевые операционные системы. - СПб.: Питер, 2001.- 560 с.
- Робачевский А.М. Операционная система UNIX. - СПб.: BHV, 1997. - 528 с.
- Руководство по Iptables Iptables Tutorial 1.1.14 http://www.linuxsecurity.com/resource_files/firewalls/IPTables-Tutorial/iptables-tutorial.html

Структура курса:

Организационная структура курса:

- Продолжительность: 1 семестр.
- 14 лекций
- 4 лабораторные работы.

  Л.Р. в 418
- Форма отчетности - зачет

Инфокоммуникации введение

Темы:

- Организация сетей
- Основные сетевые ОС и их сравнение
- ОС Linux
- TCP-IP (на базе ОС Linux)
  - Структура
  - Маршрутизация
  - Фильтрация
  - Маскирование
  - Прикладные протоколы

Инфокоммуникации

Темы(продолжение):

- Шифрование и защита информации в сетях
- Аппаратные средства обеспечения безопасности сетей D-Link
- Безопасность в WiFi сетях
- Передача данных в сетях GSM
- Обмен электронной почтой
- Протоколы цифровой телефонии

Инфокоммуникации: ЛР

Лабораторные работы:

ЛР.1 Знакомство с ОС Linux

ЛР.2 Введение в TCP-IP: маршрутизация, фильтрация, маскирование, системное программирование

ЛР.3 Шифрование и защита информации в сетях

Л.Р.4 Аппаратные средства обеспечения безопасности сетей (D-Link DFL)

Концепция сети.

Сеть: система независимых компьютеров, связанных друг с другом с целью совместного использования данных и периферийных устройств. Самая простая сеть состоит как минимум из двух компьютеров, соединенных между собой проводной или беспроводной связью.

Концепция сети

Существуют два основных типа сетей:

- ЛВС: все компьютеры сосредоточены на ограниченной территории
- ГВС: не ограничена территориально. В нее, как правило, входят устройства географически отдаленные друг от друга.Самая крупная ГВС-сеть Интернет

Конфигурация сети

Сети можно разделить на два типа:

- Одноранговые (как правило не более 10 компьютеров).
- Иерархические: на основе сервера.

Функции выделенного сервера

- Сервер приложений
- Сервер электронной почты и факсов
- Сервер учетных записей пользователей
- Файловый сервер
- Сетевой маршрутизатор и фильтр

Современные сетевые операционные системы.

- Windows
- Unix
- Linux
- Qnx
- OS/2
- Netware

Требования к современным ОС

- Расширяемость
- Переносимость
- Совместимость
- Надежность и отказоустойчивость
- Безопасность
- Производительность.

Разделение функционального назначения семейств ОС

- **серверные ОС**
  - ОС для выполнения задач (Семейство Linux,UNIX,WINDOWS,OS2)
  - ОС для как файловый сервер (NetWare)
  - ОС для управления сетью (Семейство Linux,UNIX,WINDOWS,OS2)
- **ОС для рабочих станций**
  - Windows,OS2,MAC,Linux

Рассмотрим основные характеристики современных сетевых ОС

Этот блок информации также будет вам необходим в курсе СПО (Гунько А.В.)

Семейство операционных систем UNIX

- **Общая характеристика**
  - **исключительно удачная реализация простой мультипрограммной и многопользовательской операционной системы.**
  - **создана всего двумя разработчиками Bell-Labs (AT&T) (Кен Томпсон и Деннис Ритчи) . (первая версия в 1959-60 гг)**
  - **обладает простым, но очень мощным командным языком и независимой от устройств файловой системой.**
  - **система и приложения, выполняющиеся в ней, получились легко переносимыми (мобильными), поскольку написаны на языке С.**

Цели разработки

- Стремление сохранить простоту и обойтись минимальным количеством функций.
- Общность. Одни и те же методы и механизмы должны были использоваться во многих случаях:
  - обращения к файлам, устройствам ввода/вывода и буферам межпроцессных сообщений выполняются с помощью одних и тех же примитивов;
  - одни и те же механизмы именования, присвоения альтернативных имен и защиты от несанкционированного доступа применяются к файлам с данными и директориями, и устройствам;
  - одни и те же механизмы работают в отношении программно и аппаратно инициируемых прерываний.

Цели разработки

- Создание операционной среды, в которой большие задачи можно было бы решать, комбинируя существующие небольшие программы, а не разрабатывая программы заново.
- Важным, хотя и простым с позиций его реализации, является тот факт, что система UNIX предоставляет пользователям возможность направить выход одной программы непосредственно на вход другой (речь идет о программных каналах (pipe)).

Операционная система Linux

- Linux — это современная POSIX-совместимая и UNIX-подобная операционная система для персональных компьютеров и рабочих станций.
- Как известно, Linux — это свободно распространяемая версия UNIX, которая первоначально была разработана Линусом Торвальдсом (Linus Torvalds) (torvalds@kruuna.helsinki.fi) в университете Хельсинки (Финляндия). Все компоненты системы, включая исходные тексты, распространяются с лицензией на свободное копирование и установку для неограниченного числа пользователей.

Linux

- Работы начались в конце 80х – начале 90х.
- В отличие от WIN,OS2,Mac и коммерческих UNIX-подобных систем, GNU/Linux не имеет географического центра разработки. Нет и организации, которая владела бы этой системой; нет даже единого координационного центра. Программы для GNU/Linux — результат работы тысяч проектов. Некоторые из этих проектов централизованы, некоторые сосредоточены в фирмах, но большинство объединяют разработчиков со всего света, которые знакомы только по переписке.

Операционная система Linux

- Linux был создан с помощью многих UNIX-программистов и энтузиастов из Интернета. Большинство программ Linux разработано в рамках проекта GNU из Free Software Foundation в Кембридже, Массачусетс.
- Изначально Linux создавался как «самодельная» UNIX-подобная реализация для ПК типа IBM PC с процессором i80386. В настоящее время имеется реализация этой ОС практически для всех типов процессоров и компьютеров на их основе. На базе ОС Linux создаются и встроенные системы, и суперкомпьютеры. Система поддерживает кластеризацию и большинство современных интерфейсов и технологий.

GNU

GNU – общее лицензионное соглашение о свободном программном обеспечении (июнь 1991г). Рассматриваются правовые вопросы о свободном копировании,распространении и модификации программного обеспечения в рамках данного проекта.

Основатель – Ричард Столлман

POSIX

(Portable Operating System Interface) – это стандарт на сопряжение (интерфейс) между операционной системой и прикладной программой.

термин- также Столлману

Операционная система Linux

- Linux — это полноценная многозадачная многопользовательская операционная система. Это означает, что одновременно много пользователей могут работать на одной машине, одновременно выполняя много программ. Linux достаточно хорошо совместим с рядом стандартов для UNIX на уровне исходных текстов, включая IEEE POSIX.1, System V и BSD. Такая совместимость учитывалась при его создании. Большинство свободно распространяемых по сети Интернет программ для UNIX может быть откомпилировано для LINUX практически без особых изменений.

Операционная система Linux

- Кроме того, все исходные тексты для Linux, включая ядро, драйверы устройств, библиотеки, пользовательские программы и инструментальные средства распространяются свободно. Другие специфические внутренние черты Linux включают контроль работ по стандарту POSIX (используемый оболочками, такими как csh и bash), псевдотерминалы (pty), поддержку национальных и стандартных клавиатур динамически загружаемыми драйверами клавиатур.

Операционная система Linux

- Linux поддерживает различные типы файловых систем для хранения данных. Файловая система *ext2fs,* была создана специально для Linux. Поддерживаются другие типы файловых систем, например Minix-1 и Xenix. Реализована система управления файлами на основе FAT. Поддерживается и файловая система ISO 9660 CD-ROM для работы с дисками CD-ROM. Имеются системы управления файлами и на томах с HPFS и NTFS, правда, они работают только на чтение файлов. Созданы варианты системы управления файлами и для доступа к FAT32.

Операционная система Linux

- Linux, как и все UNIX-системы, обеспечивает полный набор протоколов стека TCP/IP для сетевой работы. Это включает драйверы устройств для многих популярных сетевых адаптеров технологии Ethernet, протоколы SLIP (serial line Internet protocol, обеспечивающий доступ по TCP/IP при последовательном соединении), PPP (point-to-point protocol), NFS (network file system) и т. д. Поддерживается весь спектр клиентов и услуг TCP/IP, таких как FTP, telnet, NNTP и SMTP. Очень часто на компьютерах, работающих под управлением Linux, реализуют DNS-сервер, WWW-серверы (Apache), файерволы для защиты локальных сетей при работе в Интернете, почтовые серверы, сервер DHCP.

Операционная система OS2

Первая коммерческая версия OS2 была выпущена фирмой IBM в 1994 году.

Ядро OS2 отличает великолепная реализация вытесняющей многозадачности и достаточно развитый GUI.

К сожалению, IBM рассматривает данную ОС лишь как платформу для своих собственных проектов (банки,крупные промышленные синдикаты), поэтому она не получила широкого распространения

Эволюция ОС, ветка IBM OS2 (для серверов и раб. станций)

80e OS2 2.0

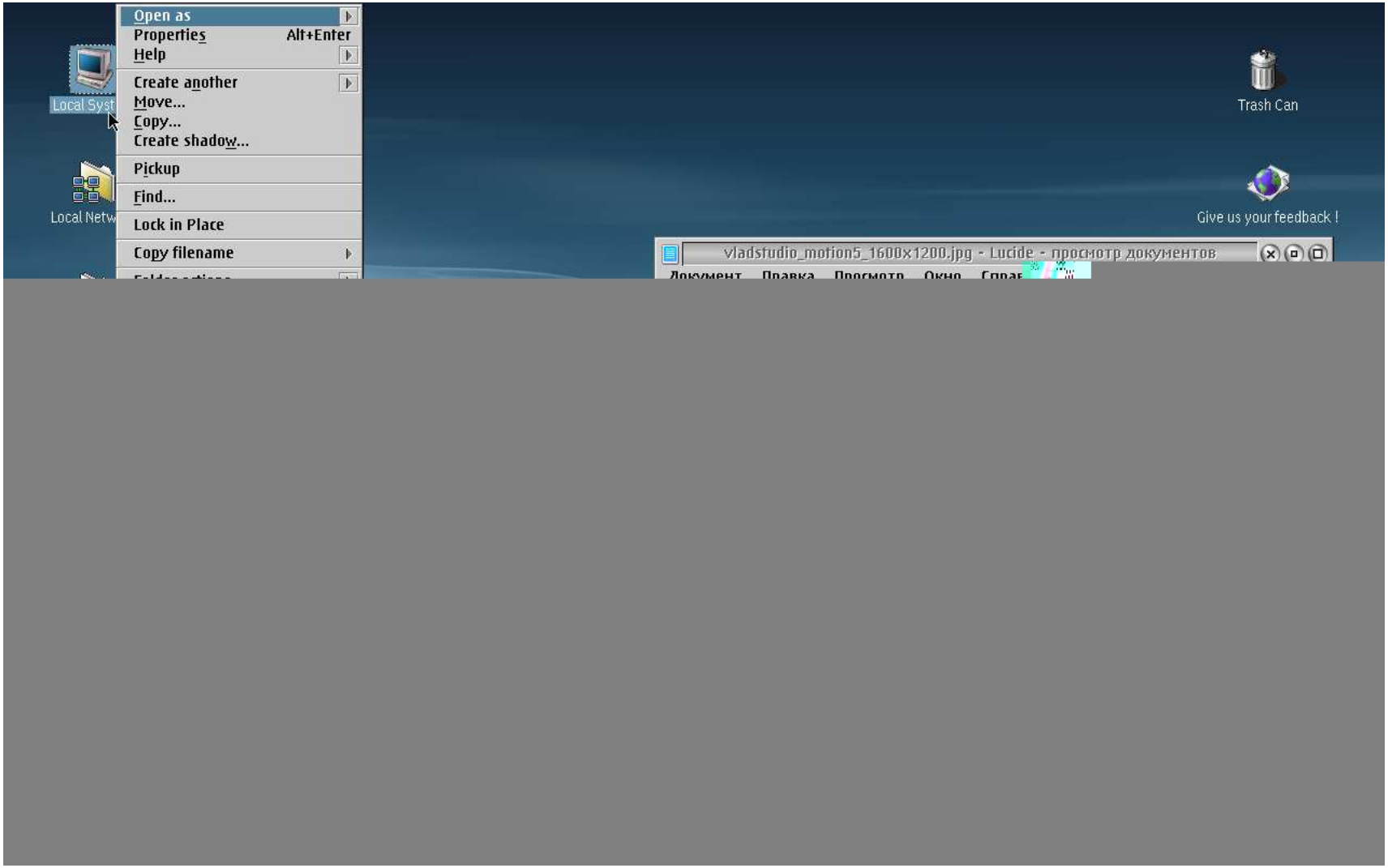
1991 OS2 3 Warp -> Lan Server

1995 OS2 4 Merlin -> Peer, Lan server

1998 OS2 Aurora

2002 OS2 EComStation

2008 OS2 EComStation 2.0


Open as
Properties Alt+Enter
Help
Create another
Move...
Copy...
Create shadow...
Pickup
Find...
Lock in Place
Copy filename
Local Syst
Local Netw
Trash Can
Give us your feedback !
vladstudio_motion5_1600x1200.jpg - Lucide - просмотр документов

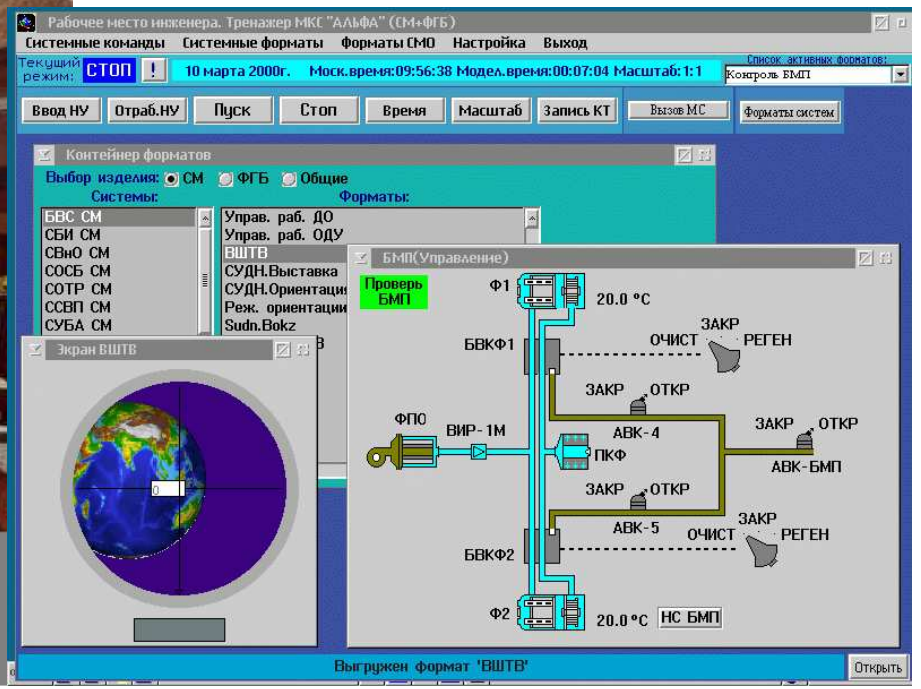
Проект OS/2 умер? Не дождутся…

- Железная дорога в Южной и Северной Италии (2008/07)
- Система сигнализации, метро Muni в Сан-Франциско
- Олимпийские игры (Сидней 2000,Атланта 1996)
- Тренажеры для нефтяников (2006)
- Промышленность
- Наука (спектроскопия)
- Космос (ЦУП)

и.т.д.

Тренажер модуля Звезда МКС

Пульт управления (OS2)

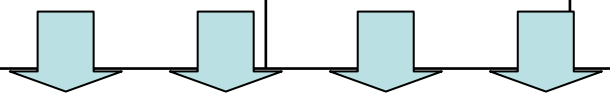


Операционная система OS2

Система поддерживает полный набор протоколов стека TCP-IP для сетевой работы. Возможно также использование в качестве файл-сервера и сервера приложений.

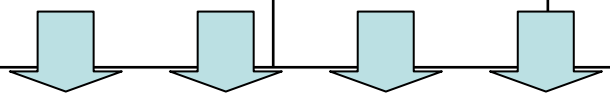
Сравнение характеристик сетевых ОС (1)

	Win	**nix	OS2	Netware
TCP-router	ЛВС	ГВС	ГВС	-
firewall	ЛВС	ГВС	ГВС	-
Файл-сервер	+	Samba	Peer	+
Контроллер домена	+	Samba3.0+ LDAP-> Домен 2000 ->~~2003~~	-	-



Сравнение характеристик сетевых ОС (2)

	Win	**nix	OS2	Netware
GUI	встроена	X win	встроена	-
Полнофункциональная работа без GUI	-	+	+	+
Открытая структура ОС	-	+	+	-



Сравнение характеристик сетевых ОС(3)

	Win	**nix	OS2	Netware
«live» версия	-	1.44Мб (64-192Mb)	1.44Мб	-
Открытый код	-	Linux	-	-
Опасность взлома	высокая	средняя	низкая	низкая

Сравнение характеристик сетевых ОС. Область применения.

Windows:

Контроллер домена,сервер ЛВС, файл сервер, сервер приложений ЛВС.

Unix/Linux:

Маршрутизатор, сетевой фильтр,сервер приложений и баз данных ГВС.

Сравнение характеристик сетевых ОС. Использование

OS/2:

Промышленные сети и системы. Сервер приложений ГВС.

Netware:

Файл-сервер

Знакомство с UNIX- подобными ОС.

Зачем?

- UNIX –подобные ОС – наиболее распространенное решение для реализации TCP-IP и других сервисов в INTERNET, в т.ч. на низком уровне.
- Платформа для одноплатных ЭВМ
- ОС для встраиваемых систем.

Dlink DSM320RD Медиацентр

Сеть

Стандарт 802.11g со скоростью соединения до 54 Мбит/с*
Порт 10/100Base-TX Fast Ethernet
Поддержка DHCP или статического IP-адреса
Сетевой протокол
TCP/IP
Протокол передачи медиа потоков
HTTP
Поддерживаемые форматы дисков
DVD, VCD, SVCD, CD-R, CD-RW, DVD-RW, CD-MP3
Поддерживаемые устройства хранения информации
SD, Compact Flash (I и II типа), MMC, Memory Stick

Поддерживаемые медиа форматы

Аудио форматы: MP3, WAV & AIFF (только PCM), WMA, Ogg Vorbis

Графические форматы: JPEG (Grayscale, RGB, только YCbCy), JPEG 2000, BMP (non-compressed), PNG, TIFF (только RGB), GIF

Видео форматы: MPEG1/2/4, XVID с MP3 или PCM,

AVI


POWERSCREEN
WARRIOR 1400
XA400

Основные понятия UNIX подобных ОС

- **Виртуальная машина**
- **Пользователь**
- **Интерфейс пользователя**
- **Привилегированный пользователь**
- **Команды и командный интерпретатор**
- **Процессы**

Виртуальная машина

- Каждому пользователю после входа в систему предоставляется виртуальный компьютер, в котором, есть все необходимые ресурсы: процессор (процессорное время выделяется на основе «карусельной» диспетчеризации (RR — round robin) и с использованием динамических приоритетов с тем, чтобы обеспечить равенство в обслуживании), память, устройства, файлы. Текущее состояние такого виртуального компьютера, предоставляемого пользователю, называется образом.

Виртуальная машина

- Можно сказать, что процесс — это выполнение образа. Образ состоит из:
  - образа памяти;
  - значений общих регистров процессора;
  - состояния открытых файлов;
  - текущего директория (каталога файлов) и другой информации.
- Образ процесса во время его выполнения размещается в основной памяти.

Пользователь

- UNIX предназначен для мультитерминальной работы. Чтобы начать работать, человек должен «войти» в систему, введя со свободного терминала свое учетное имя (account name) и, возможно, пароль (password).
- Человек, зарегистрированный в учетных файлах системы и, следовательно, имеющий учетную запись (account), называется зарегистрированным пользователем системы.
- Регистрацию новых пользователей обычно выполняет администратор системы.

Пользователь

- Пользователь не может изменить свое учетное имя, но может установить и/или изменить свой пароль. Пароли хранятся в отдельном файле в закодированном виде.
- Все пользователи ОС UNIX

Пользователь

- Каждому зарегистрированному пользователю соответствует некоторый каталог файловой системы, который называется «домашним» (home) каталогом пользователя, доступ к домашнему каталогу и всем каталогам и файлам, содержащимся в нем - неограниченный. Пользователь может создавать, удалять и модифицировать каталоги и файлы, содержащиеся в домашнем каталоге.
- Потенциально возможен доступ и ко всем другим файлам, однако он может быть ограничен, если пользователь не имеет достаточных привилегий.

Интерфейс пользователя

- Традиционный способ взаимодействия пользователя с системой UNIX - использование командных языков.
- Имеются и графические интерфейсы на базе X Window.
- После входа пользователя в систему для него запускается один из командных интерпретаторов (в зависимости от параметров, сохраняемых в файле /etc/passwd).

Интерфейс пользователя

- Обычно в системе поддерживается несколько командных интерпретаторов с похожими, но различающимися своими возможностями командными языками. Общее название для любого командного интерпретатора ОС UNIX — `shell` (оболочка), поскольку любой интерпретатор представляет внешнее окружение ядра системы.

Интерфейс пользователя

- Вызванный командный интерпретатор выдает приглашение на ввод пользователем командной строки, которая может содержать простую команду, конвейер команд или последовательность команд.
- После выполнения очередной командной строки и выдачи на экран терминала или в файл соответствующих результатов, `shell` снова выдает приглашение на ввод командной строки, и так до тех пор, пока пользователь не завершит свой сеанс работы и не выйдет из системы.

Интерфейс пользователя

- Командные языки, используемые в ОС UNIX, достаточно просты и достаточно мощны, чтобы можно было использовать их для написания сложных программ.
- Имеется механизм командных файлов (`shell scripts`), которые могут содержать произвольные последовательности командных строк. При указании имени командного файла вместо очередной команды интерпретатор читает файл строка за строкой и последовательно интерпретирует команды.

Привилегированный пользователь

- Ядро ОС UNIX идентифицирует каждого пользователя по его идентификатору (UID — user identifier), уникальному целому значению, присваиваемому пользователю при регистрации в системе.
- Кроме того, каждый пользователь относится к некоторой группе пользователей, которая также идентифицируется некоторым целым значением (GID — group identifier).

Привилегированный пользователь

- Значения UID и GID для каждого зарегистрированного пользователя сохраняются в учетных файлах системы и приписываются процессу, в котором выполняется командный интерпретатор, запущенный при входе пользователя в систему.
- Эти значения наследуются каждым новым процессом, запущенным от имени данного пользователя, и используются ядром системы для контроля правомочности доступа к файлам, выполнения программ и т. д.

Привилегированный пользователь

- Администратор системы (зарегистрированный пользователь) должен обладать большими возможностями, чем обычные пользователи.
- В ОС UNIX эта задача решается путем выделения единственного нулевого значения UID. Пользователь с таким UID называется суперпользователем (superuser) или root.
- Он имеет неограниченные права на доступ к любому файлу и на выполнение любой программы. Кроме того, он имеет возможность полного контроля над системой. Он может остановить ее и даже разрушить.

Привилегированный пользователь

- Важным отличием суперпользователя ОС UNIX является то, что на него не распространяются ограничения на используемые ресурсы.
- Для обычных пользователей устанавливаются такие ограничения, как максимальный размер файла, максимальное число сегментов разделяемой памяти, максимально допустимое пространство на Диске и т. д.
- Суперпользователь может изменять эти ограничения для других пользователей, но на него они не действуют.

Команды и командный интерпретатор

- *Оболочкой* (`shell`) в системе UNIX называют механизм взаимодействия между пользователями и системой. Это интерпретатор команд, который считывает набираемые пользователем строки и запускает выполнение запрошенных системных функций.
- Полный командный язык, интерпретируемый оболочкой, богат по возможностям и достаточно сложен, однако большинство команд просты в использовании и запомнить их не составляет труда.

Команды и командный интерпретатор

- Командная строка состоит из имени команды (то есть имени выполняемого файла), за которым следует список аргументов, разделенных пробелами. Оболочка разбивает командную строку на компоненты. Указанный в команде файл загружается, и ему обеспечивается доступ к заданным в команде аргументам.

Команды и командный интерпретатор

- Любой командный язык семейства shell фактически состоит из трех частей:
  - служебных конструкций, позволяющих манипулировать с текстовыми строками и строить сложные команды на основе простых команд;
  - встроенных команд, выполняемых непосредственно интерпретатором командного языка;
  - команд, представляемых отдельными выполняемыми файлами.
- В свою очередь, набор команд последнего вида включает стандартные команды (системные утилиты, такие как vi, cc и т. д.) и команды, созданные пользователями системы.

Лекция

Тема лекции:

Файловая система ОС Линукс:

- физическая структура ФС
- логическая структура ФС
- Защита файлов
- манипулирование учетными записями пользователей

Ввод, вывод, файловая система

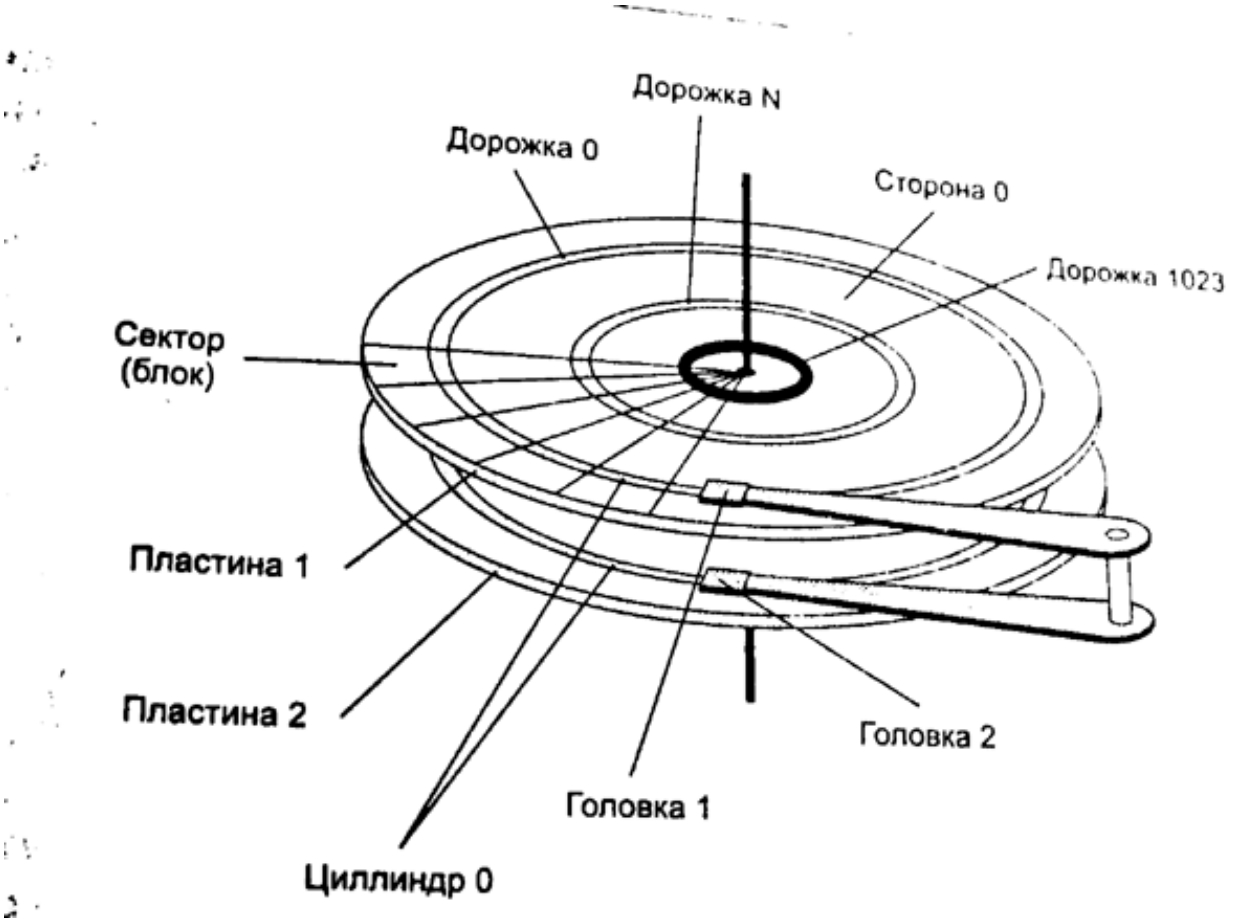
Одна из главных задач ОС- обеспечение обмена данными между приложениями, периферийными устройствами. Реализуется через подсистему ввода-вывода.

Реализация:

- Файловый обмен
- Общие области

Физическая организация файловой системы, диски

- цилиндр
- сектор
- Кластер
- MBR



Задачи ОС по управлению файлами и устройствами

- Организация параллельной работы устройств ввода-вывода
- согласование скоростей обмена и кэширование
- разделение устройств и данных между процессами
- поддержка широкого спектра драйверов с возможностью включения нового драйвера
- динамическая загрузка и выгрузка драйверов
-

ФС ОС Linux

- Minix- первая ФС. Ограничения раздела: 64Мб , имя файла <30 символов (н.в. для дискет и RAM дисков)
- Extfs,Ext2fs,ext3fs - “родная” ФС Linux(second extended filesystem),
- Ms-dos – разделы FAT16.
- Umsdos – расширение FAT16 для Linux.Добавлено:длинные имена, идент. UID/GID,разрешения в стиле POSIX,и спец. ус-ва (каналы,сокеты и.т.д.) совместимость с DOS не утеряна.

ФС ОС Linux(продолжение)

- HPFS – разделы OS/2 (r/o)
- Nfs – Сетевая ФС
- Swap – файл или раздел свопинга
- Iso9660 – CDROM (r/o)
- Vfat – раздел FAT-32
- NTFS - Разделы NT (r/o)
- Proc - используется для обращения к структурам данных ядра. Файлы этой системы не занимают дискового пространства.

Структура ФС Unix/Linux

- Физический размер 1 сектора на диске=512 байт.
- Перед созданием файловой системы диск должен быть подготовлен (разбит) на логические диски (fdisk,cfdisk)
- Разметка ФС осуществляется командой mke2fs

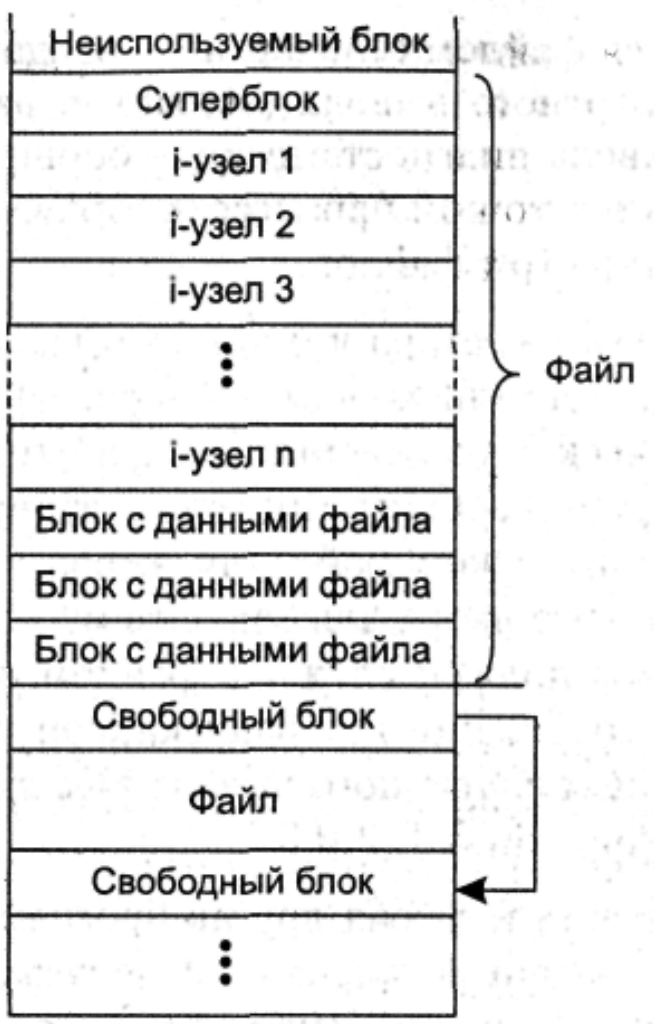
Ext2fs – логическая структура

- Макс. Размер = 4Тбайт.
- Загрузочная область.На первичном разделе содержит загрузочную запись.
- Блоки (1,2,4Кб). Блок- адресуемая единица дискового пространства. Место на диске кратно размеру блока.

Блоки объединяются в группы блоков:

Группы блоков:

1. неиспользуемый блок;
2. управляющий блок или суперблок, в котором содержится размер диска и границы других областей;
3. i-список, состоящий из описаний файлов, называемых i-узлами;
4. область для хранения содержимого файлов.



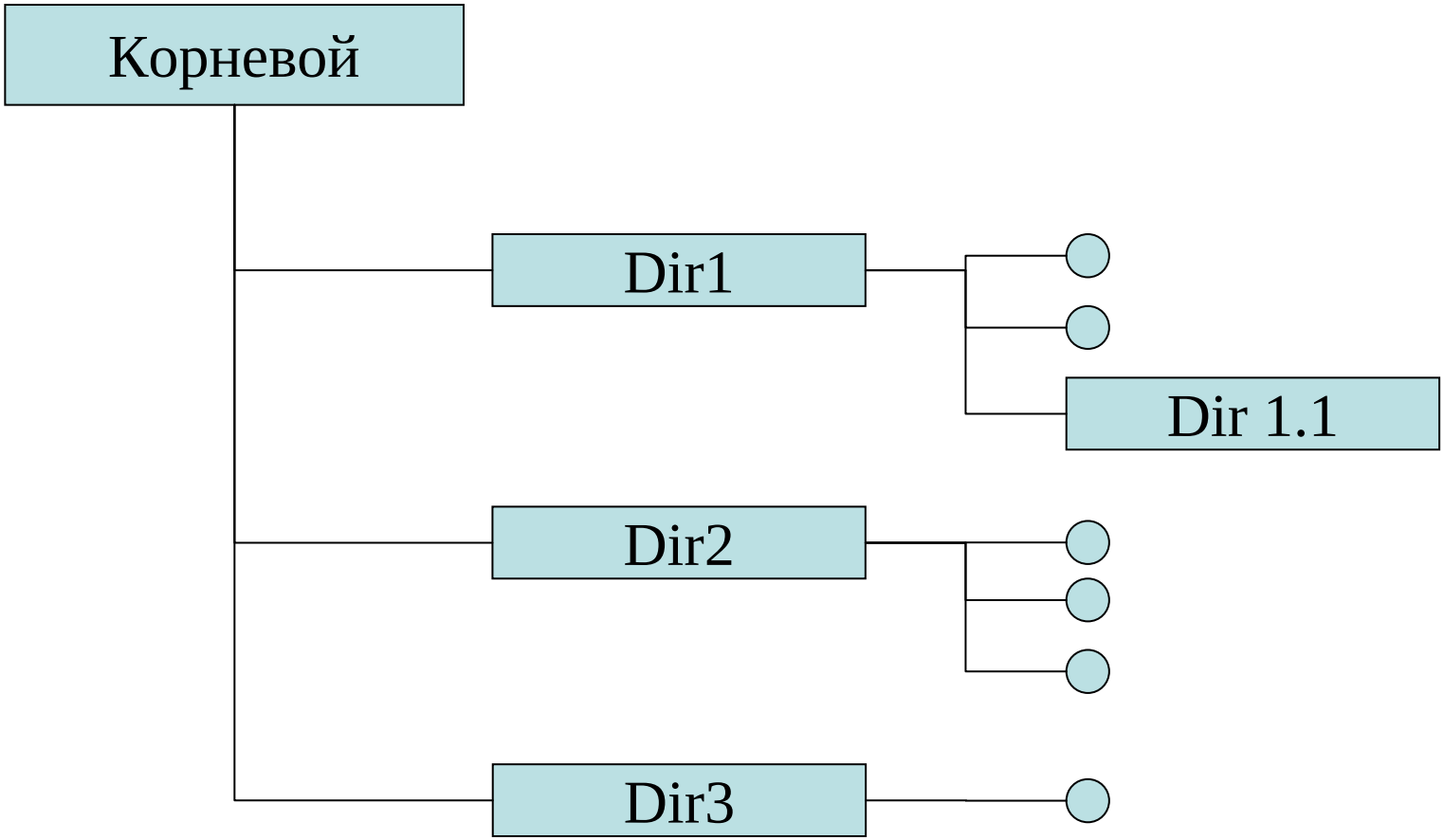
Структура i-узла

- Каждый i-узел содержит:
  - идентификацию владельца;
  - идентификацию группы владельца;
  - биты защиты(прав доступа);
  - физические адреса на диске или ленте, где находится содержимое файла;
  - размер файла;
  - время создания файла;
  - время последнего использования файла;
  - время последнего изменения атрибутов;
  - число связей-ссылок, указывающих на файл;
  - индикацию, является ли файл директорией, обычным файлом или специальным файлом.

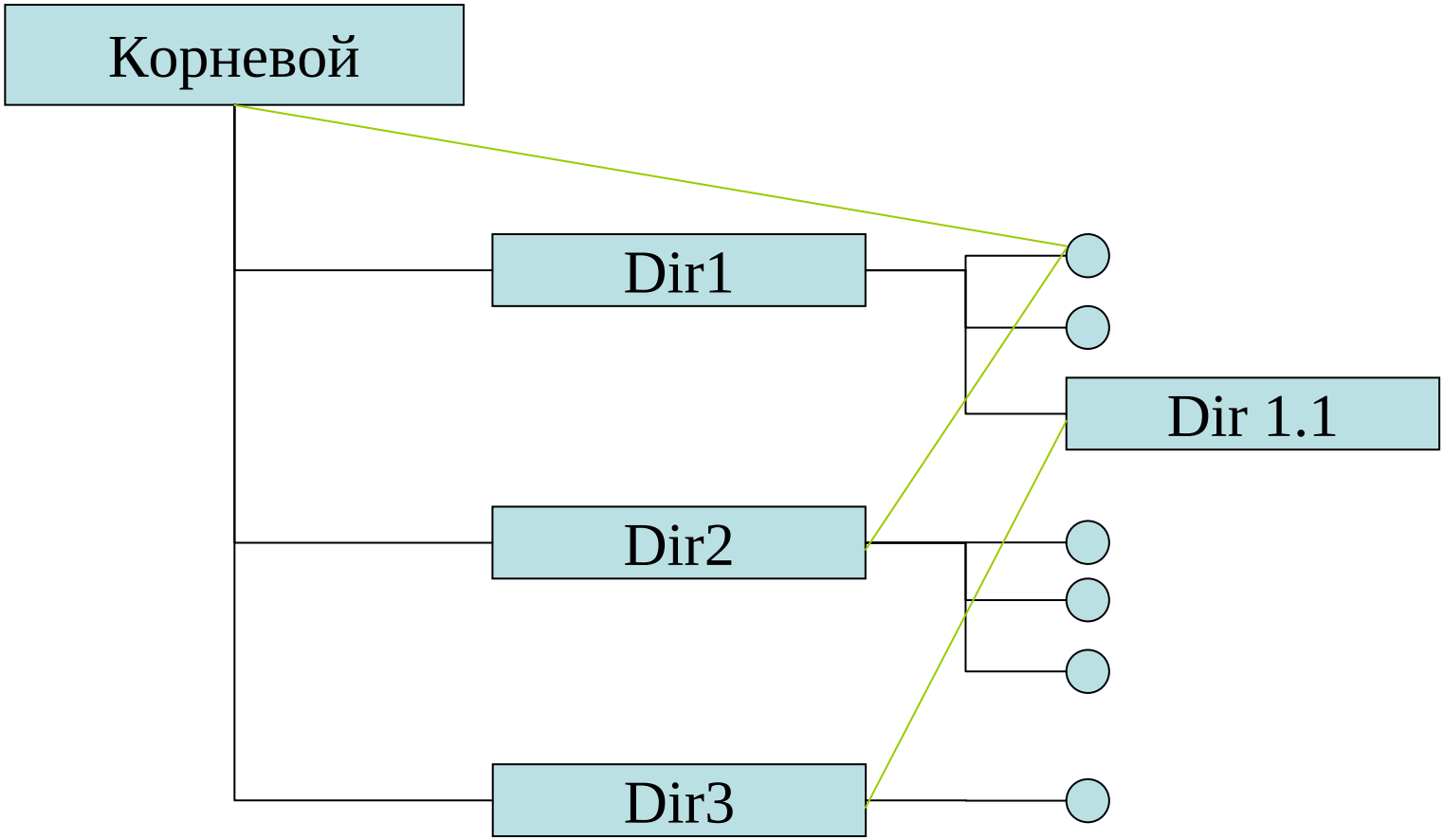
Блоки для хранения файлов

- Следом за i-списком идут блоки, предназначенные для хранения содержимого файлов. Пространство на диске, оставшееся свободным от файлов, образует связанный список свободных блоков.
- Файловая система UNIX представляет собой структуру данных, содержащую управляющий суперблок, в котором определена файловая система в целом; массив i-узлов, где определены все файлы в файловой системе; сами файлы; совокупность свободных блоков.
- Выделение пространства под данные осуществляется блоками фиксированного размера.

Типы структур - древовидная



Типы структур –древовидная сетевая



Директории

- Директории образуют древовидно-сетевую структуру. На каждый обычный файл или файл устройства могут иметься ссылки в различных узлах этой структуры. Дополнительных связей между директориями нет.
- Большое число системных директорий система UNIX использует для своих собственных нужд. Одна из них, корневая директория, является базой для всей структуры директорий.
- В других системных директориях содержатся программы и команды, предоставляемые пользователям, и файлы устройств.

Правила именования директорий

*

- Имена файлов задаются последовательностью имен директорий, разделенных косой чертой (/) и приводящих к концевому узлу (листу) некоторого дерева. Если имя файла начинается с косой черты, то поиск по дереву начинается в корневой директории. Если же имя файла не имеет в начале косой черты, то поиск начинается с текущей директории. Имена файлов, начинающиеся с «../», подразумевают начало поиска в директории, родительской по отношению к текущей, с «./» - в текущей.
- Сама по себе косая черта / обозначает корневую директорию.

Файлы устройств

- С каждым поддерживаемым системой устройством ассоциируется один или большее число специальных файлов. Операции ввода/вывода для специальных файлов осуществляются так же, как и для обычных дисковых файлов, с той лишь разницей, что эти операции активизируют соответствующие устройства.
- Специальные файлы обычно находятся в справочнике /dev. На специальные файлы могут указывать связи точно так же, как на обычные файлы.

*

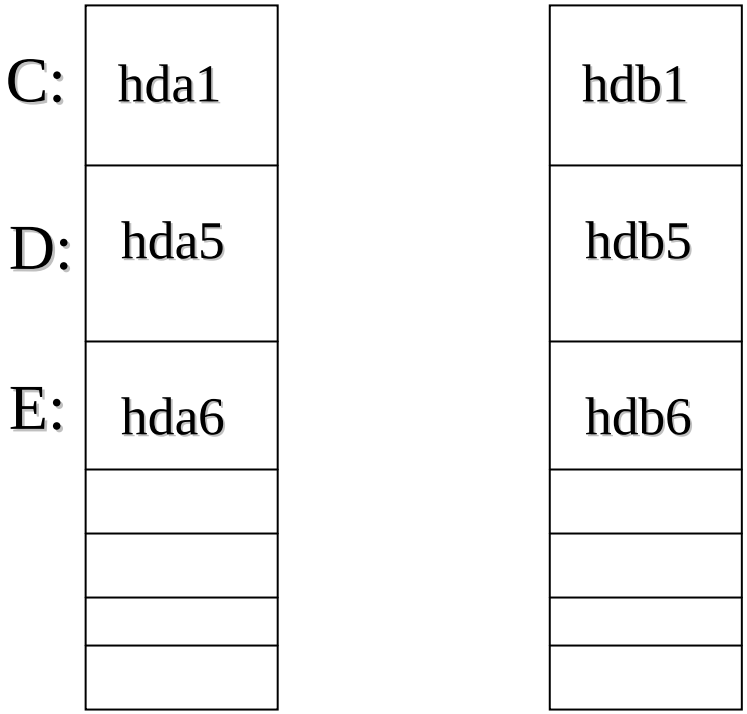
Файлы устройств

Типы устройств: блочное и символьное,сокет

- hd – жесткий диск (/dev/hda1) *(б)*
- fd – флоппи-дисковод (/dev/fd1) *(б)*
- ttyS – последовательный порт (/dev/ttyS0)
- sd – USB диск (/dev/sda,sdb,sdc)
- Tty – терминал
- Null – псевдоустройство

Правила именования логических разделов

PM=hda PS=hdb SM=hdc SS=hdd



Монтирование устройств

- От файловой системы не требуется, чтобы она вся целиком размещалась на том устройстве, где находится корень.
- Запрос от системы `mount` (на установку носителей и т. п.) позволяет встраивать в иерархию файлов файлы на сменных томах. Команда `mount` имеет несколько опций, но обязательных аргументов у стандартного варианта ее использования два: имя файла блочного устройства и имя каталога.

Монтирование устройств

- В результате выполнения этой команды файловая система, расположенная на указанном устройстве, подключается к системе таким образом, что ее содержимое заменяет собой содержимое заданного в команде каталога. Поэтому для *монтирования* соответствующего тома обычно используют пустой каталог.
- Команда `umount` выполняет обратную операцию — «отсоединяет» (размонтирует) файловую систему, после чего диск с данными можно физически извлечь из системы. Например, для записи данных на дискету необходимо ее смонтировать, а после работы — размонтировать.

Link(жесткая ссылка)

- Файл, может встречаться в различных директориях, возможно, под разными именами. Элемент в директории, относящийся к одному файлу, называется связью. В системе UNIX такие связи имеют равный статус.
- Файл «исчезает», когда удаляется последняя связь, указывающая на него.
- Биты защиты, заданные в связях, могут отличаться от битов в исходном файле. Таким образом, решается проблема избирательного ограничения доступа к файлам.

Link(жесткая ссылка)

Особенность:

- физически указывает на номер индексного дескриптора,следовательно все ссылки могут располагаться в пределах одного и того же физического раздела.
- Может указывать только на **физически существующий файл**.

Link\Создание

Ln имя_файла_или_каталога имя_ссылки

[Пример:]
Ln /home/vasya/file.txt /home/tanya/link.txt

SymLink(символическая ссылка)

Особенности:

- особый файл,который существует **независимо** от «родительского файла»
- может указывать на директорию
- может указывать на другой раздел или устройство, которое в данный момент не готово (например, нет диска в приводе)

SymLink\Создание

Ln -S имя_файла_или_каталога
имя_ссылки

[Пример:]

Ln –S /mnt/dir2 ~/FLOPPYDIR

Защита файлов\права доступа

*

- Защита файлов осуществляется при помощи номера, идентифицирующего пользователя, и установки десяти битов защиты — атрибутов доступа.
- Права доступа подразделяются на три типа: чтение (read), запись (write) и выполнение (execute). Эти типы прав доступа могут быть предоставлены трем классам пользователей: владельцу файла, группе, в которую входит владелец, и всем (прочим) пользователям.
- Девять из этих битов управляют защитой по чтению, записи и исполнению для владельца файла, других членов группы, в которую входит владелец, и всех других пользователей.

Защита файлов\владельцы

- Файл всегда связан с определенным пользователем — своим владельцем — и с определенной группой, то есть у него есть UID (user ID, идентификатор пользователя) и GID (group ID, идентификатор группы). Изменять права доступа к файлу разрешено только его владельцу. Изменить владельца файла может суперпользователь, группу — суперпользователь или владелец файла.

Защита файлов/ chown chgrp

Владелец файла устанавливается командой chown.

chown root /home/test/test.txt

Группа файла устанавливается командой chgrp

chgrp users /home/test/test.txt

Атрибуты при создании

- При создании файл получает UID, совпадающий с FSUID процесса, который его создает, и, как правило, GID, совпадающий с FSGID этого процесса.
- Атрибуты доступа определяют, что разрешено делать с данным файлом данной категории пользователей. Имеются всего три операции: чтение, запись и выполнение.
- При создании файла (или еще одного имени для уже существующего файла) модифицируется не сам файл, а каталог, в котором появляются новые ссылки на узлы. Удаление файла заключается в удалении ссылки. Таким образом, право на создание или удаление файла — это право на запись в каталог.

Атрибуты каталога

- Право на выполнение каталога интерпретируется как право на поиск в нем (прохождение через него). Оно позволяет обратиться к файлу по пути, содержащему данный каталог, даже тогда, когда каталог не разрешено читать и список всех его файлов поэтому недоступен.

Защита файлов

- Помимо трех названных основных атрибутов доступа существуют дополнительные. Атрибуты SUID и SGID существенны при запуске программы на выполнение: они требуют, чтобы программа выполнялась не от имени запустившего ее пользователя (группы), а от имени владельца (группы) того файла, в котором она находится.
- Если файл программы имеет атрибут SUID (SGID), то FSUID и EUID (FSGID и EGID) соответствующего процесса не наследуются от процесса, запустившего его, а совпадают с UID (GID) файла. Благодаря этому пользователи получают возможность запустить системную программу, которая создает свои рабочие файлы в закрытых для них каталогах.

Дополнительные атрибуты каталога *

- Кроме того, если процесс создает файл в каталоге, имеющем атрибут SGID, то файл получает GID не по FSGID процесса, а по GID каталога. Это удобно для коллективной работы: все файлы и подкаталоги в каталоге автоматически оказываются принадлежащими одной и той же группе, хотя создавать их могут разные пользователи. Есть еще один атрибут — CVTX, который теперь относится к каталогам. Он показывает, что из каталога, имеющего этот атрибут, ссылку на файл может удалить только владелец файла.

Запись прав доступа

- Существуют две стандартные формы записи прав доступа — символьная и восьмеричная. Символьная представляет собой цепочку из десяти знаков, первый из которых обозначает тип файла. Используются следующие обозначения:
  - «-» — обычный файл;
  - «d» — каталог (директория);
  - «c» — символьное устройство;
  - «b» — блочное устройство;
  - «p» — именованный канал (named pipe);
  - «s» — «гнездо» (socket[1]);
  - «l» — символическая ссылка.

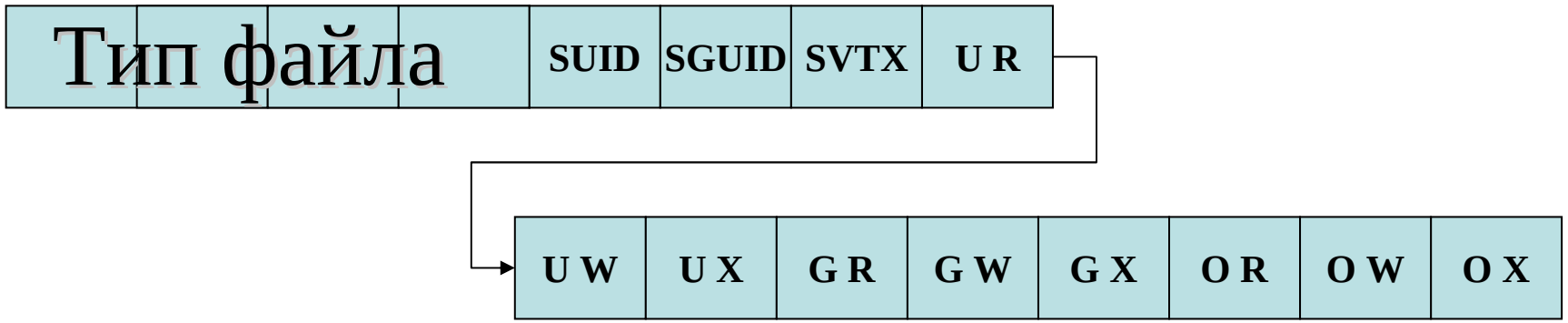
Запись прав доступа

- Далее следуют три последовательности, каждая из трех символов, соответствующие правам пользователя, группы и всех остальных. Наличие права на чтение обозначается буквой «r», на запись — «w», на выполнение — «x», отсутствие какого-либо права — знаком «-» в соответствующей позиции.
- Наличие атрибута SUID (SGID) обозначается заглавной буквой «S» в позиции права на выполнение для владельца (группы), если выполнение не разрешено, и прописной буквой «s», если разрешено.

Запись прав доступа\восьмеричная

- Восьмеричная запись — это шестизначное число, первые два знака которого обозначают тип файла и довольно часто опускаются, третья цифра — атрибуты GUID (4), SGID (2) и SVTX (1), а оставшиеся три — соответственно права владельца, группы и всех остальных. Очевидно, что право на чтение можно представить числом «4», право на запись — числом «2», а право на выполнение кодируется как «1».

Запись прав доступа\восьмеричная



SUID/SGUID – сохранять группу при выполнении файла

=> в поле X->S если разрешено выполнять, s - если запрещено.

SVTX – “Strikly bit”, удалить файл может только владелец

U – владелец, G – группа , O – прочие

R - чтение,W - запись, X – выполнение

Для каталогов атрибут X означает

Защита файлов

- Например, стандартный набор прав доступа для каталога `/tmp` в символьной форме выглядит как drwxrwxrwt, а в восьмеричной — как 041777 (это каталог; чтение, запись и поиск разрешены всем; установлен атрибут SVTX). А набор прав -r-S-xw-, или в числовом виде — 102412, будет означать, что это обычный файл, владельцу разрешается читать его, но не выполнять и не изменять; пользователям из группы файла (за исключением владельца) — выполнять (причем во время работы программа получит права владельца файла), но не читать и не изменять; а всем остальным — изменять, но не читать и не выполнять.

Защита файлов/ chmod

За изменение атрибутов файла отвечает команда chmod

2 формата: числовой формат:

Chmod <число> файл

Chmod 760 /home/test/test.txt

Защита файлов/ chmod

Chmod – символьный формат:

Chmod wXp <имя файла>

Где:

w: u,g,o

X+(предоставить) –(лишить) =(установить)

P r,w,x

Chmod ugo-rw /home/test/test.txt

Защита файлов/umask

- Большинство программ создают файлы с разрешением на чтение и запись для всех пользователей, а каталоги — с разрешением на чтение, запись и поиск для всех пользователей. Этот исходный набор атрибутов логически складывается с «пользовательской маской» — *user file-creation mask*, сокращенно `umask`, которая обычно ограничивает доступ.
- Например, следующие значения для `umask u=rwx, g=rwx, o=r-x` следует понимать так: у владельца и группы остается полный набор прав, а всем остальным запрещается запись.

Защита файлов

- В восьмеричном виде оно запишется как 002 (первая цифра — ограничения для владельца, вторая — для группы, третья — для остальных, запрещение чтения — 4, записи — 2, выполнения — 1).
- Владелец файла может изменить права доступа к нему командой `chmod`.

Вставка

Применение ссылок

- Гибкость структуры дерева файловой системы
- Переносимость версий ПО (стандартизация)
- Ссылки на специальные файлы, которые не могут быть перемещены

Создание групп пользователей

groupadd - создание группы

groupadd [–g GID] имя группы

groupmod – изменяет параметры группы

groupmod [-g GID] [–n New_Name] –o Old_name

Группы хранятся в /etc/group

Создание учетных записей

adduser - создание новой учетной записи

adduser –u UID –g <group> –G <group2>,< group3>,< group3> <имя>

Usermod – изменяет параметры учетной записи пользователя

Usermod –u UID –g <group> –G <group2>,< group3>,< group3> <имя>

Учетные записи хранятся /etc/passwd , /etc/shadow

Разделение прав доступа к объектам ФС

Задача: реализовать полные права доступа к каталогу /home/exchange

для пользователей mike,alex,andy,elena группы users:webadmin .

Решение 1.

chgrp webadmin /home/exchange

chmod g+rwx,0-rwx

Корректное решение задачи

Задача: реализовать полные права доступа к каталогу /home/exchange

для пользователей mike,alex,andy,elena группы users:webadmin .

Решение 2.

groupadd exch

chgrp exch /home/exchange

chmod g+rwx,0-rwx

usermod mike –G webadmin,exch

usermod andy –G webadmin,exch

usermod elena –G webadmin,exch

usermod alex –G webadmin,exch

Системные файлы для хранения БД пользователей

/etc/passwd

/etc/group

/etc/shadow

Для шифрования пароля применяется алгоритм DES или MD5. Для вскрытия пароля методом перебора придется проверить 2^56 (7.2 e 16) вариантов. Криптостойкость пароля часто «ухудшают» сами пользователи, используя «штампы». Кроме того, для вскрытие пароля облегчается распараллеливанием вычислений.

Назначение основных каталогов системы

- /home - домашние каталоги
- /bin,/sbin – исполняемые файлы
- /dev – файлы устройств
- /etc – файлы конфигурации
- /var, /usr, /opt – служебные файлы
- /lib файлы библиотек
- /tmp – временные файлы
- /proc – файловая система для связи с ядром

Лекция

Тема лекции

- Основы администрирования ОС Linux. Настройка,назначение конфигурационных файлов в каталоге /etc
- ОС Linux. Наиболее часто используемые команды.
- ОС Linux. Управление процессами из командной строки.

Конфигурационные файлы

Конфигурационные файлы системы сосредоточены в каталоге /etc

Иногда по умолчанию пакеты программ имеют собственный каталог с конфигурационными файлами. Однако,если программа написана согласно “правилам хорошего тона разработчика Linux”, она сначала предпринимает попытку найти конфигурацию в /etc , и использовать их, если они там найдены.

(Например, apache /etc/apache -> /usr/local/apache/conf)

Начальная загрузка ОС. Загрузчик Lilo

Загрузчик LILO создан Вернером Альмесбергером (Werner Almesberger). LILO

может загружать ядро Linux как с дискеты, так и с жесткого диска, а также

может загружать другие операционные системы: PC/MS-DOS, DR DOS, OS/2,Windows 95/98, Windows NT/XP, 386BSD, SCO UNIX, UNIXware и т. д. Может

быть задан выбор до 16 разных операционных систем на этапе загрузки.

lilo

Загрузочный сектор LILO при инсталляции системы можно разместить в:

- загрузочный сектор дискеты в формате Linux (/dev/fd0,...);
- MBR первого жесткого диска (/dev/hda, /dev/sda,...);
- загрузочный сектор первичного раздела файловой системы Linux на первом жестком диске (/dev/hdal, /dev/hda2,...);
- загрузочный сектор логического раздела в расширенном разделе первого жесткого диска (/dev/hda5,...).

lilo

Загрузочный сектор LILO при инсталляции системы <u>нельзя</u> разместить в:

- загрузочный сектор дискеты или первичного раздела, отформатированных в других файловых системах;
- в swap-разделе Linux;
- на втором жестком диске.

Lilo

загрузочный сектор LILO и файлы конфигурации должны находиться в пределах первых 1024 (логических) цилиндров на жестком диске, т. к. они должны быть доступны через BIOS.

Lilo.conf

Все параметры загрузки хранятся в файле /etc/lilo.conf

Для записи загрузочной области необходимо запустить программу lilo.

Старый загрузочный сектор будет сохранен в /boot/boot. NNNN, где NNNN соответствует номеру устройства,

Lilo vs grub

lilo

+ Стандартное средство, которое одинаково для всех Linux

+ может корректно быть вторичным загрузчиком

- Необходимо перезапускать lilo после каждого изменения параметров ядра.

Grub

- Гибкий загрузчик, инсталлируемый в MBR Основные возможности GRUB:
- Загрузка Linux, Solaris, *BSD ядер
- Передача управления другим загрузчикам (chainloading)
- Защита паролем пунктов меню
- Поддержка BOOTP и TFTP для сетевой загрузки
- Интерактивная командная строка загрузки
- Поддержка файловых систем (FFS, FAT16, FAT32, Minix, ext2, ReiserFS, JFS и XFS) и чтение файлов конфигурации, ядер, initrd и других файлов прямо с файловой системы.
- **GRUB не требует переустановки загрузчика после изменения параметров загрузки как в lilo**

Задача: клонировать сервер с установленным Grub

Предположим, что мы копируем на /dev/sdb1

- Cp –a всех «не виртуальных» каталогов (таких как /proc /sys)
- Заходим в шелл grub *grub>*
- Grub> **device (hd1) /dev/sdb** – *привязали имя*
- Grub> **root (hd1,0)** – привязали корень фс
- Grub> **setup (hd1)**
- Grub> **quit**

Задача: клонировать сервер с установленным Grub

Редактируем /boot/grub/menu.lst

Узнаем UUID ваших разделов: tune2fs -l /**dev**/**sd**?

title *Ubuntu 7.10, kernel 2.6.22.14-rtai37*

kernel */boot/vmlinuz-2.6.22.14-rtai37* ***root=/dev/sda1*** *[**root=UUID=XXXXX**] ro quiet splash*

initrd */boot/initrd.img-2.6.22.14-rtai37*

Фрагментик menu.lst из меню DrWebLiveUsb

```
title Dr.Web LiveCD
    root (hd0,0)
    kernel /drweb/vmlinuz init_opts=4
    root=/dev/ram0 dokeymap looptype=squashfs
    loop=/boot/module/white.mo usbroot slowusb
    vga=791 CONSOLE=/dev/tty1 init=/linuxrc
    splash=silent,theme:drweb
    initrd /drweb/initrd
```

Восстановление MBR

Восстановить MBR можно с помощью команды

[root:~#] lilo –u

Или с помощью fdisk дискеты DOS:

[a:] fdisk /mbr

Загрузка Ос. Процесс init

После загрузки и распаковки ядра системы, ядро монтирует корневую файловую систему и запускает процесс init. Процесс init — это программа, которая ответственна за продолжение процедуры загрузки, и перевод системы от начального состояния, возникающего после загрузки ядра, в стандартное состояние обработки запросов многих пользователей и множество других операций. Точный список этих операций зависит от **уровня выполнения** (run level).

Загрузка ОС. Уровни выполнения.

Существует 8 основных уровней выполнения системы:

- 0 – остановка системы
- 1,S – однопользовательский режим
- 2 – Многопользовательский, без nfs
- 3 – Полнофункциональный, многопользовательский режим
- 4 - Запуск в графическом режиме
- 5 – не регламентируется
- 6- перезагрузка системы

Initd inittab

В традиционных системах наследующих идеологию systemV параметры уровня загрузки находятся в конфигурационном файле inittab, который считывается после запуска демона

Строки состоят из 4 полей,разделенных двоеточиями:

Inittab

id:runlevels:action:process

- id — идентификатор строки. Это произвольная комбинация, содержащая от 1 до 4 символов.
- runlevels — уровни выполнения, на которых эта строка будет задействована.
- process — процесс, который должен запускаться на указанных уровнях.
- action – действие

Inittab/action/ключевые слова:

- **respawn** — перезапустить процесс в случае завершения его работы;
- **once** — выполнить процесс только один раз при переходе на указанный уровень;
- **wait** — процесс будет запущен один раз при переходе на указанный уровень и init будет ожидать завершения работы этого процесса, прежде, чем продолжать работу;
- **sysinit** - действия, выполняемые в процессе загрузки системы независимо от уровня выполнения

inittab/action/ключевые слова:

- boot — процесс будет запущен на этапе загрузки системы независимо от уровня выполнения;
- bootwait — процесс будет запущен на этапе загрузки системы независимо от уровня выполнения, и init будет дожидаться его завершения;
- initdefault – уровень запуска по умолчанию.Если не задан-ожидается ввод с клавиатуры.
- off — игнорировать данный элемент;

inittab/action/ключевые слова:

- powerwait — сопряжение с UPS
- ctrlaltdel – действие,которое выполняется по нажатию <ctrl>+<alt>+<del>

Список не является исчерпывающим. Подробнее – man.

Init / переход на указанный режим.

Переход на указанный режим осуществляется при помощи команды

init N , где N – номер режима.

Применить изменения – init q

Inetd/ Пример

Запуск программ пользователя с помощью inetd

Задача: Запустить процесс,который должен быть автоматически перезапущен, если произошел его останов:

Решение: inittab

my::345::respawn::/имя процесса >/dev/null

Обновление конф. Idetd : *init q*

Запуск системы. Rc

В системах Fedora Ubuntu последних сборок

наметился переход к новой системе управления процессом начальной загрузки: UpStart

- Задачи и службы запускаются и останавливаются при помощи событий
- При запуске/останове задач и служб генерируются события
- Событие может быть получено от любого процесса в системе
- Сервисы могут автоматически перезапускаться в случае их неожиданного останова
- Двунаправленная связь с демоном init, что позволяет получать больше информации в процессе работы.

Идеология настройки слегка изменилась ,но общие черты остались прежними.

Другие файлы, влияющие на процесс загрузки *

- /etc/lilo.conf
- /etc/modules.conf (или /etc/conf.modules) – подгружаемые модули
- /etc/fstab – монтируемые файловые системы
- /etc/passwd – персональные параметры пользователей
- /etc/group – параметры групп
- /etc/profile – профили,(%path и др.)*
- /etc/bashrc – конф. bash*

Пример

Задача: поприветствовать определенного пользователя по имени при входе в систему:

Решение: /etc/profile

if test $USER = test; then

echo ‘Уважаемый TEST мы рады Вас видеть!’

fi

Системный планировщик

Для того,чтобы запустить процесс в определенное время, необходимо воспользоваться системным планировщиком **crond**

Файл конфигурации системного планировщика находится в каталоге **/usr/spool/cron/чей файл**

Каждый зарегистрированный пользователь имеет свой планировщик.

Системный планировщик

Время запуска записывается через пробел:

[минута] [час] [день] [месяц] [деньнедели] [задача]

- * - каждый
- */2 – каждый второй
- 7,8 перечисление

Системный планировщик

Задача: выполнять действие каждые 3 минуты первые 20 минут часа, с 7 до 10 и с 17-19, в понедельник:

Решение: *crontab –e*

0-20/3 7-10,17-19 * mon /home/hello.pl

Основные команды системы

Получение справки

- man – справочная информация

  man lilo.conf

- info – подробная справочная информация

  info lilo.conf

Операции с файлами и каталогами

Основные команды системы

Операции с файлами и каталогами

- mkdir [-m mode 777]– создание каталога
- cat – создание файла. *cat 1.txt >2.txt*
- cp [-i/-f –r -d]– копирование файла

  cp откуда куда
- mv – перемещение файла

Основные команды системы

- rm – удалить файл
- rmdir – удалить каталог

Форматирование вывода

- more – выводить постранично

ls –l | more

- less – расширенная версия more

Основные команды системы

- find – поиск файлов

[user]$ find /usr/share/doc /usr/doc /usr/locale/doc -name instr.txt

- * - все вхождения слов, ? все вхождения символов, [a-f] список символов

-name шаблон указанного имени

-group шаблон указанной группы

-size число [c] – файлы указанного размера

-mtime число – файлы,которые в последний раз изменялись указанное число дней назад

-newer образец - файлы,которые изменялись после изменения файла,указанного в образце

-type – файлы указанного типа [cbslpd]

-or логическое или

FIND пример

Задача: удалить в каталоге /home/test все файлы, к которым пользователи не обращались(были созданы позднее чем) в течении 90 дней:

Решение:

[root]# find /meteoin/backup -type f -ctime +90 -delete

[root]# find /meteoin/backup -type f -atime +90 -delete

split – разбивает файл

разбить файл

```
[user]$ split -b400K myfile.mp3 song
```

собрать файл

```
[user]$  cat song.* > myfile.mp3
```

grep

Поиск строк ,содержащих указанный шаблон.

grep “шаблон” [где искать] > [куда положить]

Задача: найти в файлах протокола все случаи входа в систему пользователя test

Решение:

grep “test” /var/log/messages > /home/test/logs.txt

Основные команды системы

сравнение файлов

- cmp – побайтное сравнение
- diff – сравнение с выводом отчета

Задача: послать коллеге отчет об исправлениях в программе,не пересылая весь текст

Решение:

```
[user]$ diff program.c program.c.new > program.c.diff
[user]$ patch program.c program.c.diff
```

Управление процессами

ps [-опции] – вывод списка процессов

ps –e – стандартная форма

Список полей

USER — имя владельца процесса;

PID — идентификатор процесса в системе;

PPID — идентификатор родительского процесса;

%CPU — доля времени центрального процессора (в процентах

), выделен-

ного данному процессу;

%MEM — доля реальной памяти (в процентах), используемая данным процессом;

vsz — виртуальный размер процесса (в килобайтах);

RSS — размер резидентного набора (количество 1К-страниц в памяти);

STIME — время старта процесса;

ps

Список полей (продолжение)

TTY — указание на терминал, с которого запущен процесс;

S или STAT — статус процесса;

PRI — приоритет планирования;

NI — значение nice;

TIME — сколько времени центрального процессора занял данный про-

цесс;

CMD или COMMAND — командная строка запуска программы, выполняемой данным процессом.

ps

R — выполнимый процесс, ожидающий только момента, когда планировщик задач выделит ему очередной квант времени;

s — процесс "спит";

o — процесс находится в состоянии подкачки на диске;

t — остановленный процесс;

z — процесс-зомби.

Рядом с указателем статуса могут стоять дополнительные символы из следующего набора:

w — процесс не имеет резидентных страниц;

< — высокоприоритетеный процесс;

N — низкоприоритетный процесс;

L — процесс имеет страницы, заблокированные в памяти.

- top – непрерывный вывод «слепка» списка процессов

Изменение приоритета

Приоритет процесса определяется так называемым "значением nice", которое лежит в пределах от +20 (наименьший приоритет, процесс выполняется только тогда,когда ничто другое не занимает процессор), до -20 (наивысший приоритет). Значение nice устанавливается в момент порождения каждого процесса и при обычном запуске равно значению приоритета родительского процесса.

nice renice

nice [- adnice] command [args] изменяет значение nice при запуске программ

[- adnice] –20 +19

renice priority [[-p] PID] [[-g] grp] [[-u] user]

[root]# renice -1 987 -u daemon -p 32

Сигналы и команда *kill*

Сигналы — это средство, с помощью которого процессам можно передать сообщения о некоторых событиях в системе. Всего в Linux существует 63 разных сигнала, их перечень можно посмотреть по команде

```
[user]$ kill -l
```

Сигналы и команда *kill*

вызов: kill [-SIG] [PID]

Нам понадобятся следующие сигналы

-9 KILL – безусловное прерывание процесса

-15 TERM – программное завершение процесса

-3 QUIT – выход из процесса

-19 STOP

-18 CONT

Подробнее с сигналами вы познакомитесь в курсе системное ПО.

Управление процессами/другие команды

- killall [name] – прервать процесс с именем name
- *Имя процесса*& перевод после запуска в фоновый режим
- jobs – вывод списка заданий,запущенных из некущего shell
- f g – перевод процесса на передний план;
- bg – возврат процесса в фон.
- nohup команда & - отменяет прерывание процесса при закрытии shell

Команды архивирования файлов: tar

- tar – Tape Archiver. Объединяет файлы в один файл последовательного доступа

-d – находит различия между файлом и диском

-u – добавляет файлы новее,чем в архиве

tar -Mcvf /dev/fdOH1440 /каталог : сохранить на дискете

tar -Mxpvf /dev/fdOH1440 : восстановить

Команды архивирования файлов: gzip

Сжимает файл. Для того,чтобы сжать несколько файлов, используется вместе с tar (опция -z).

tar -czf имя_архива.gz шаблон_имен_файлов (или имя__каталога)

Лекция

Лекция 4.

Оболочка Bash

- Язык shell как язык программирования
- Инсталляция драйверов устройств
- Обновление ПО

Командные языки, командная строка- назначение

Командная строка:

- Управление удаленными, труднодоступными устройствами и системами в «консольном» режиме
- Работа по низкоскоростному каналу связи (от 1200 бод)

Командный язык – надстройка над консольными командами для увеличения гибкости и построения алгоритмов средней сложности для управления устройствами и их конфигурирования

Командная строка

- Абсолютно все параметры настроек Unix-подобных систем доступны для администрирования в консольном режиме.
- Подавляющее большинство «интеллектуальных» аппаратных сетевых устройств (в т.ч. построенных на собственных платформах), поддерживают администрирование при помощи командной строки.

Программирование на bash: так причем же тут сети и коммуникации?

- Большая часть интеллектуальных сетевых узлов в сети - **nix
- ->
- большая часть автоматизаций и процедур взаимодействия в **nix- командные языки
- ->
- стандартное средство для создания системных сценариев - bash

shell , как язык программирования

Язык Shell это системный язык программирования (построчный интерпретатор).Обладает всеми основными функциями : циклы, условное ветвление, подпрограммы (функции). При программировании на Shell все переменные являются системными переменными окружения среды.

Программа на Shell может содержать всего одну строчку, а может быть оформлена в виде отдельного файла.

В первой строчке обычно указывается интерпретатор (если не указано-shell по умолчанию, обычно /bin/bash)

Функции – внешние и внутренние. Внешние- любая команда системы.

**nix подобные системы

- В структуре ОС широко используются командные языки (rc.d файлы)
- Множество команд операционной системы- составные команды, написанные на SHELL (useradd и adduser)

Операторы: выполнение команд:

- ; - последовательное выполнение нескольких команд *comand1;command2*
- & - перевод команды в фон *command1&*
- && , || - выполнение команды по статусу

  command_ret_0 && command2

  command_ret_not0 || command2

Статус- это то, что вернет основная подпрограмма вызова (с – main())

Перенаправление ввода/вывода

Потоки : stdin stdout stderr

- > перенаправить вывод в файл
- >> перенаправить вывод в файл,дописать
- < перенаправить ввод из файла
- | создать программный канал

ls –l | *more*

Перенаправления вывода- часто используемые приемы

- Программы из Cron > /dev/null (Иначе- все на почту суперпользователя)
- ./myprog > /dev/tty1-6 – терминалы
- myprog > /dev/tty0 – текущий активный терминал

Программные каналы

Пример: определить PID процесса с именем crond

ps –le | grep “crond”

Пример: Определить количество строк в файле , содержащие слово “root”

grep “root” /var/spool/messages | wc -l

Переменные

Переменная с точки зрения оболочки это параметр, обозначаемый именем. Значения переменным присваиваются с помощью оператора следующего вида

[user]$ name=value

Набор всех установленных переменных оболочки с присвоенными им значениями называется окружением (environment) или средой оболочки.

Вывести набор переменных окружения можно командой *set*.

Вывести конкретную переменную: [user]$ echo $name

Специальные переменные окружения:PATH

- PATH – путь поиска файлов. по умолчанию:/usr/local/bin:/bin:/usr/bin:/usr/XHR6/bin:

  добавить новую переменную:

  [user]$ ***PATH=$ PATH:new_path***

Поиск файла- от первого к последнему каталогу в переменной PATH.

Другие полезные системные переменные

- $_ - последняя введенная команда
- $hostname – имя хоста
- $hosttype- оптимизация ядра на тип CPU
- $OSTYPE – тип ОС
- $term – тип терминала (служебные команды терминала)
- $UID - UID текущего пользователя
- $USER – имя пользователя

Специальные переменные окружения: export

Когда оболочка запускает на выполнение какую-то программу или команду,она передает им часть переменных окружения. Чтобы внести переменную в список передаваемых процессу:

[user]$ export name=value

Скрипт на языке shell

- Состоит из строк.
- В строке может быть одна или несколько команд (зависимых или независимых)
- В первой строчке скрипта необходимо указать интепретатор (если файл текстовый и он не указан будет предпринята попытка применить /bin/bash)
- Файл должен быть исполняемый для пользователя (установлен атрибут ‘x’)

Порядок раскрытия выражений при интерпретации текста программы

- раскрытие скобок (brace expansion);
- замена знака тильды (tilde expansion);
- подстановка параметров и переменных;
- подстановка команд;
- арифметические подстановки (выполняемые слева направо);
- разделение слов (word splitting),(не забывать про разделители!);
- раскрытие шаблонов имен файлов и каталогов (pathname expansion).

Раскрытие скобок

Раскрытие скобок выполняется до выполнения других видов подстановок

Поясним процесс раскрытия скобок на примере.

Задача: создать несколько подкаталогов

Решение:

[user]$ mkdir /home/test/ {a{d,c,b}e}

Замена тильды

~ заменяется на login_name

~/ заменяется переменной HOME

~имя/ заменяется HOME пользователя

~+ заменяется на полное имя текущего каталога (PWD)

~- заменяется на полное имя прошлого каталога (OLDPWD)

Подстановка параметров и переменных

$ - признак того, что следующее слово-переменная. Рекомендуется заключать имя переменной в скобки чтобы отделить ее от последующей строки

[root]$ echo “${PATH}_is my_path”

[root]$ echo “$PATH_is my_path”

Подстановка команд

Подстановка команд заключается в замене ***имени*** команды на ***результат*** ее выполнения.

- $(command) – все символы внутри скобок-команда
- 'command‘ – не выполняется интерпретация содержимого

пример

- new=$(ls –l);

Присваиваем переменной окружения new значения листинга текущего каталога.

Важно:

Переменная будет существовать до окончания текущего сеанса.

Пример-интерпретация

- comm =‘/usr/bin/lsdev’
- echo $($comm)
- echo $(“$comm”) – вывод списка уст-в
- echo $(‘$comm’) – не работает

Замена арифметики

Арифметические подстановки позволяют вычислить значение арифметического выражения и подставить вместо него результат.

$ ((expression))

[user]$ echo $((2 +3 *5))

Разделение слов

Разделение слов в команде осуществляется путем поиска символов, заданных в переменной IFS

```
[root]$ echo $IFS;
```

Раскрытие шаблонов

Условное ветвление if

Оператор проверки условия if

if {list} then {list2} else {list3} fi

=0?

test или [] – проверяет значение выражения и возвращает 0 – если истина, 1 – если ложь.

Значение, возвращаемое таким составным оператором if, совпадает со значением, выдаваемым последней командой выполняемой последовательности.

Команда проверки выражений test

Условные выражения, строятся на основе проверки файловых атрибутов, сравнения строк и обычных арифметических сравнений.

- -a –e –f file – верно, если файл существует
- -b –c –d –p -S file верно, если файл соотв. типа
- -h –l file верно, если ссылка
- -r/-w/-x file верно, если дано право на чтение/запись/исполнение
- -s верно, если размер больше нуля
- -N file верно, если с момента последнего чтения файл был изменен

Test (продолжение)

- filel -nt file2 – верно,если время модификации 1 позднее
- filel -ot file2 – верно,если 1 старше
- filel -ef file2 – верно,если файлы эквив.(inode)
- -z/-n string верно, если длина строки =/<> 0
- stringl == string2 верно,если строки совпадают
- stringl != string2 верно,если строки совпадают
- argl –ne/-le/-lt/-gt/-ge arg2

Test , условные выражения

- !(expression) “NOT”
- **expression1 -a expression2 “AND”**
- **expression1 -o expression2 “OR”**

Пример:

```
if [ “$USER”==“test” –o “$USER”==“root” ];
   then { echo ‘hello’! }
fi
```

Case – совпадение с образцом

case word in [[(] pattern [| pattern] ...) list ;;] ... esac

/etc/re.d/rc.sysinit.

```
case "$UTC" in
yes|true)
CLOCKFLAGS="$CLOCKFLAGS -u";
CLOCKDEF="$CLOCKDEF (utc)";
;;
no|false)
CLOCKFLAGS="$CLOCKFLAGS —localtime";
CLOCKDEF="$CLOCKDEF (localtime)";
;;
esac
```

Select – интерактивное взаимодействие

select name [in word;] do list ; done

введенная пользователем строка сохраняется в переменной REPLE

Пример:

```
#/bin/sh
echo “что предпочитаете на завтрак?”
select var in “Чай” “Кофе” “Сок” “Поспать подольше"; do break
done
echo "Вы бы выбрали $var"
```

Циклы: for

Оператор for при каждом проходе цикла присваивает переменной очередное значение из заданного списка слов.

for name in words do list done

Пример: создаем файлы

```
#/usr/bin/sh
for a in 1 2 3 ; do
touch /home/test/foo_$a
done
for a in seq (1 10); do
touch /home/test/foa_$a
done
```

Операторы *while* и *until*

while listl do Iist2 done

Задача протоколировать содержание директории каждые 10 секунд,пока она существует:

```
#/usr/bin/sh
while [ -d /home/test/dir ] ; do
Is -1 /home/test/dir >> /home/test/dir.log
echo -- SEPARATOR — >>  /home/test/dir.log
sleep 60
done
```

Функции

Оболочка позволяет создавать команды. Они ведут себя как обычные команды.

Конструктор: function name () { list }

Аргументы, переданные функции нумеруются $1,$2,$3.

$* - все аргументы, $# их число.

Функции (локальные)

можно задать локальный параметр внутри функции:

local name=value.

Возврат из функции выполняется командой return

Функцию можно вызывать рекурсивно.

Пример

Задача: написать функцию вычисления факториала

```
#/usr/bin/sh
fact10=fact(10);
echo “Факториал=$fact10“
fact ( )
{
if [ $1 = 0 ] ; then
echo 1;
else
{
echo $( ( $1 * $( fact $( ( $1 - 1 ) ) ) ) )
};
fi
```

Пример «боевого» скрипта

Задача: Копировать протоколы срабатывания защит трансформаторной станции на flash носитель. Flash носитель вне моментов обмена держать размонтированным.

```
#!/bin/sh

umount /mnt/sdb1;
DATE=`date +%d`
TIME=`date +%m%d%k%M%S`

echo $DATE
echo $TIME
if [ $( ps -l | grep "logger.sh" | wc -c ) == 0 ]; then exit; fi;

if [  -d /mnt/sdb1 ]; then
  echo "mnt dir exist"
else
  mkdir /mnt/sdb1
Fi

echo "flag" > /mnt/sdb1/mntflag.flg
mount /dev/sdb1 /mnt/sdb1

if [ -f /mnt/sdb1/mntflag.flg ]; then
echo "not mouted"
else
mkdir "/mnt/sdb1/$DATE"
cp -a $1 "/mnt/sdb1/$DATE/$TIME.xml"
#echo  "hello" > /mnt/sdb1/$DATE/$TIME.xml
umount /mnt/sdb1;
fi
```

Инсталляция драйверов устройств

Драйвер- это «транслятор», или связующее звено между аппаратной частью устройства и программным приложением, использующим это устройство. Драйвер позволяет обращаться к устройствам различных производителей, используя стандартизованный набор команд.

драйверы

В системе Линукс драйверы могут быть трех типов:

- код драйвера встроен в ядро (VGA,IDE)
- внешние модули ядра (lan,sound)
- функции поделены между ядром и модулем (напр. lpd)

модули/ загрузка

Список автоматически загружаемых модулей находится в */etc/modules*, опции - */etc/modules.conf*, сами модули: */lib/modules*.

Изменять файл рекомендуется при помощи скрипта *update-modules*

Модули/загрузка из командной строки

Подключить или отключить модули в работающей системе можно при помощи:

- **lsmod** – выводит список подключенных модулей
- **insmod** sound – загружает модуль из командной строки
- **rmmod** sound – выгружает модуль.

ПРИМЕЧАНИЕ! Хотя модуль имеет расширение .o в командах необходимо указывать имя без расширения.

modprobe – автоматически загружает модули.
modprobe –c – текущая конфигурация модулей

Установка ПО

Необходимость в установке новых программных пакетов под Linux возникает в двух основных случаях:

- когда появляется новая версия одного из уже установленных у вас пакетов;
- когда возникает желание или необходимость использовать какой-то пакет, еще не установленный в системе.

Два способа установки ПО

По может поставляться:

- в виде исходного текста (tar-gz)
- в виде пакета исполняемых модулей,запакованных в специально подготовленные архивы (tgz:Slack,BSD; rpm: RedHat)

Установка пакетов

Slackware, FREEBSD:

installpkg имя пакета – утилита установки

removepkg имя пакета – утилита удаления

pkgtool – утилита манипулирования пакетами

rpm2tgz – преобразование rpm в tgz

Red-Hat,ASP,Debian:

rpm - утилита манипулирования

RPM

Примеры:

rpm –qa – список установленных пакетов

rpm -qf /etc/bashrc – чей файл, откуда взялся

rpm [—install] [installoptions] [package_file] + -установка

rpm [—freshen|-F] [installoptions] [package_file] + - обновление

rpm [—uninstall|-e] [uninstalloptions] [package] + деинсталляция

При возникновении неразрешимых зависимостей выводится предупреждение,пакет не устанавливается.

Компиляция из исходных текстов

1. cd Перейти в каталог с исходным текстом
2. **./.configure** – сконфигурировать пакет
3. **make** – создать пакет
4. make check – самотестирование
5. **make install** – установить файлы
6. make clean – удалить временные файлы
7. make distciean - удалить временные файлы configure

Установка из Интернет-хранилищ (репозиториев)

apt (*advanced packaging tool*) — программа для установки, обновления и удаления программных пакетов в операционных системах Debian и основанных на них (Ubuntu, Runtu и т. п.). Способна автоматически устанавливать и настраивать программы для UNIX-подобных операционных систем как из предварительно откомпилированных пакетов, так и из исходных кодов.

Программа создана для пользователей настольных систем, привыкших к автоматизации процессов инсталляции.

Пакеты в Debian-подобных системах

- apt-get install имя пакета
- dpkg –i имя имя файла.deb
- dpkg –r имя пакета

Файл /etc/apt/sources.list

Как часть своей работы, APT использует файл, который содержит список 'источников', из которых могут быть скачаны пакеты. Это файл /etc/apt/sources.list.

Обычно этот файл имеет следующий формат:
deb http://host/debian distribution раздел1 раздел2 раздел3
deb-src http://host/debian distribution раздел1 раздел2 раздел3

Т.е. это означает,что система сама знает «где взять» ту или иную программу и как ее ставить (apt-get), избавляя пользователя от дилеммы поиска и выбора версии. Кроме того, apt-get update умеет обновлять список программ.

FAQ тут: http://www.posix.ru/distro/apt_faq/

В любом случае Вам доступны все предыдущие методы установки.

Мы познакомились с обзором ОС и Linux.

На следующей лекции-
переходим к стеку TCP-IP

Лекция

Лекция 5.

ВВЕДЕНИЕ В TCP-IP

- история
- RFC
- многоуровневая структура протоколов
- IP V4
- IP V6

Литература

- У.Ричард Стивенс// Протоколы TCP/IP Практическое руководство.-Спб.: “Невский диалект”- “БХВ-Петербург”,2003-672с
- Мур М. и др. Телекоммуникации. Руководство для начинающих. – СПб.БХВ-Петербург,2005. – 624С
- Величко и др. Телекоммуникационные системы и сети. Учебное пособие в 3х томах. М.:Горячая линия-телеком,2005.
- Microsoft TCP/IP. Учебный курс:Официальное пособие Microsoft для самостоятельной подготовки: Перевод с англ. – 2е изд.,испро.-М.: Издательско-торговый дом «Русскаяредакция», 1999-344с:ил
- [*] S Deering , R Hinden -- *Internet Protocol, Version 6 (IPv6 Specification* -- RFC 1883, 1995

Обзор TCP-IP, история

Семейство протоколов *Transmission Control Protocol/Internet Protocol(TCP/IP)* – это стандартный промышленный набор протоколов, разработанный для *глобальных вычислительных сетей (Wide Area Networks,WAN)* Семейство протоколов TCP/IP позволяет осуществлять обмен данными между любыми *микропроцессорными устройствами* – независимо от их вычислительной мощности, фирмы изготовителя или установленных операционных систем.

Обзор TCP-IP, история

Примечательно, что масштабы использования TCP/IP многократно превзошли первоначальные планы. То, что зародилось в США (в агентстве DARPA Министерства Обороны) на исходе 60х годов как обычный госбюджетный научно-исследовательский проект, переросло в 90е в самый распространенный в мире способ сетевого взаимодействия.TCP/IP лежит в основе всемирной паутины Internet,которая объединяет миллионы компьютеров и МУ и буквально опутывает весь мир.

IP протокол объединяющий множество протоколов канального уровня (Ethernet,Wify,PPP,eth)

RFC – руководящие документы

Основные руководящие документы Internet публикуются в виде так называемых RFC (Request for comment). Среди RFC можно найти как все официальные стандарты Internet, так и предложения еще не являющиеся стандартами, а также многочисленные материалы информационного плана. RFC идентифицируются их порядковыми номерами (например,RFC 1122), согласно нумерации по порядку очередности.

RFC: где найти

В электронном виде все RFC можно получить бесплатно по электронной почте или ftp. Инструкцию как запрашивать информацию можно получить по e-mail,послав запрос

to: rfc-info@ISI.EDU

subject: getting rfcs

help: ways_to_get_rfcs

При поиске нужной информации отправной точкой служит постоянно обновляемый указатель RFC (RFC index)

RFC: Особо важные документы статусы

- Assigned Numbers RFC – зарезервированные значения
- The Internet Official Protocol Standards: -официальные стандарты
  - standard
  - draft standards
  - proposed standards
  - experimental
  - informational
  - historic
- Host Requirements – требования к хосту, Router Requirements RFC
  - must обязательно
  - should желательно
  - may допустимо
  - should not нежелательно
  - must not недопустимо

RFC -5 типов

- Required
- Recommended
- Elective
- Limited use
- Not recommended

Руководство стандартами

С момента появления первого RFC функции координации выполнял один из «китов» Internet Джонотан Постиль. С 1998 года, после его кончины, эта функция возложена на некоммерческую организацию ICANN (Internet Corporation for Assigned Names and Numbers) www.icann.org

Обзор TCP-IP, история

В эволюции протокола TCP/IP можно выделить несколько важных этапов:

- 1970г Узлы сети ARPNET начали использовать протокол NCP(Network Control Protocol)
- 1972г Первая спецификация telnet оформлена как RFC
- 1973г Введен протокол File Transfer Protocol, RFC 454
- 1974г Представлена программа Transmission Control Program (TCP)
- 1981г в RFC 791 опубликован стандарт протокола IP
- 1982г протоколы TCP и IP объединены в набор TCP/IP
- 1983г сеть ARPNET переведена на протокол TCP/IP
- 1984г Введена доменная система имен DNS
- 1995г Опубликован стандарт IPV6
- 1998г Управление Internet возложено на ICANN

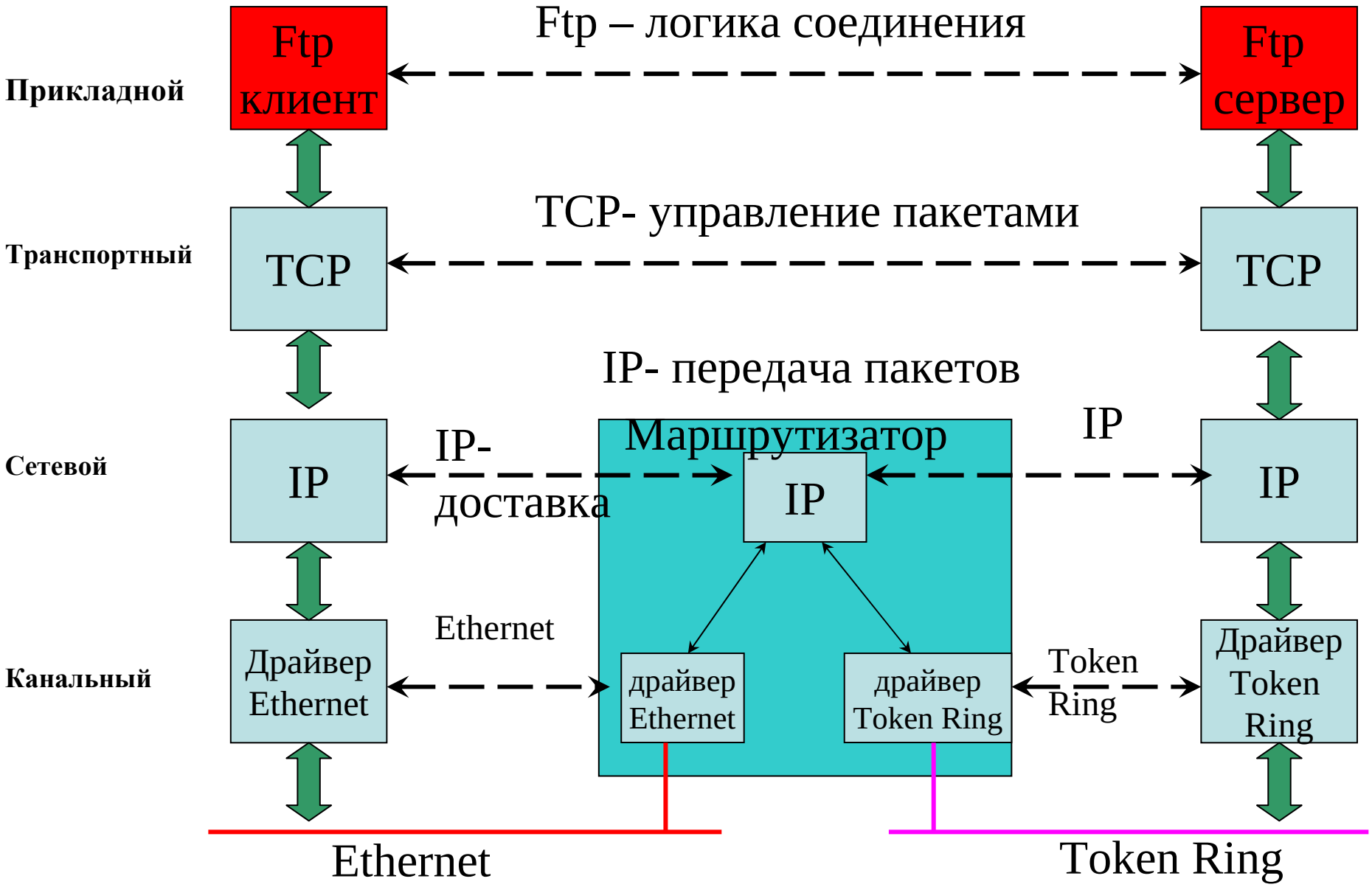
Уровни, разделение функций

- Канальный : (аппаратное устройство+драйвер в ОС).Осуществляет подключение к физической среде и управление аппаратными процессами
- Сетевой: отвечает за перемещение по тому или иному маршруту в сети (***I****nternet* ***P****rorocol,* ***I****nternet* ***C****ontrol* ***M****essage* ***P****rotocol*, ***I****nternet* ***G****roup* ***M****anagement* ***P****rotocol*)

Уровни, разделение функций

- **Транспортный уровень** организует обмен данными между двумя компьютерами в сети. Используются два принципиально различных транспортных протоколов:
  - TCP
  - UDP

- **Прикладной уровень** обеспечивает выполнение разнообразных прикладных задач. Существует определенный “классический” набор прикладных сервисов:
  - telnet
  - ftp
  - smtp
  - snmp


Прикладной
Транспортный
Сетевой
Канальный
Ftp клиент
Ftp – логика соединения
Ftp сервер
TCP
TCP- управление пакетами
TCP
IP- передача пакетов
IP
IP-
доставка
Маршрутизатор
IP
IP
IP
Ethernet
Драйвер Ethernet
драйвер Ethernet
драйвер Token Ring
Token Ring
Драйвер Token Ring
Ethernet
Token Ring

ПРОТОКОЛ IPV4

Действующий в настоящее время стандарт. Еще долгое время будеи использоваться в системах автоматики даже после перехода на IPv6

Адресация в сетях IP V4

Адрес- уникальное свойство узла сети

- Действующий стандарт адреса IP V4
- Перспективный (draft standard-готовый к внедрению) стандарт адреса IP V6

Адресация в Internet

Каждый сетевой интерфейс в сети должен иметь свой уникальный IP адрес. Ip адрес (ipv4) содержит 32 двоичных разряда (4 байта- 4’294’967’296 комбинаций). Каждый ip адрес состоит из двух частей:

- идентификатора сети(network ID).Определяет физическую сеть, одинаков для всех узлов одной сети и уникален для каждой из сетей объединенной сети
- идентификатора узла (host ID) .Уникален в рамках данной сети.

Каждый TCP/IP узел однозначно определяется по своему IP адресу.

Можно провести аналогию с мегаполисом:

- адрес сети- название улицы
- адрес хоста – номер дома.

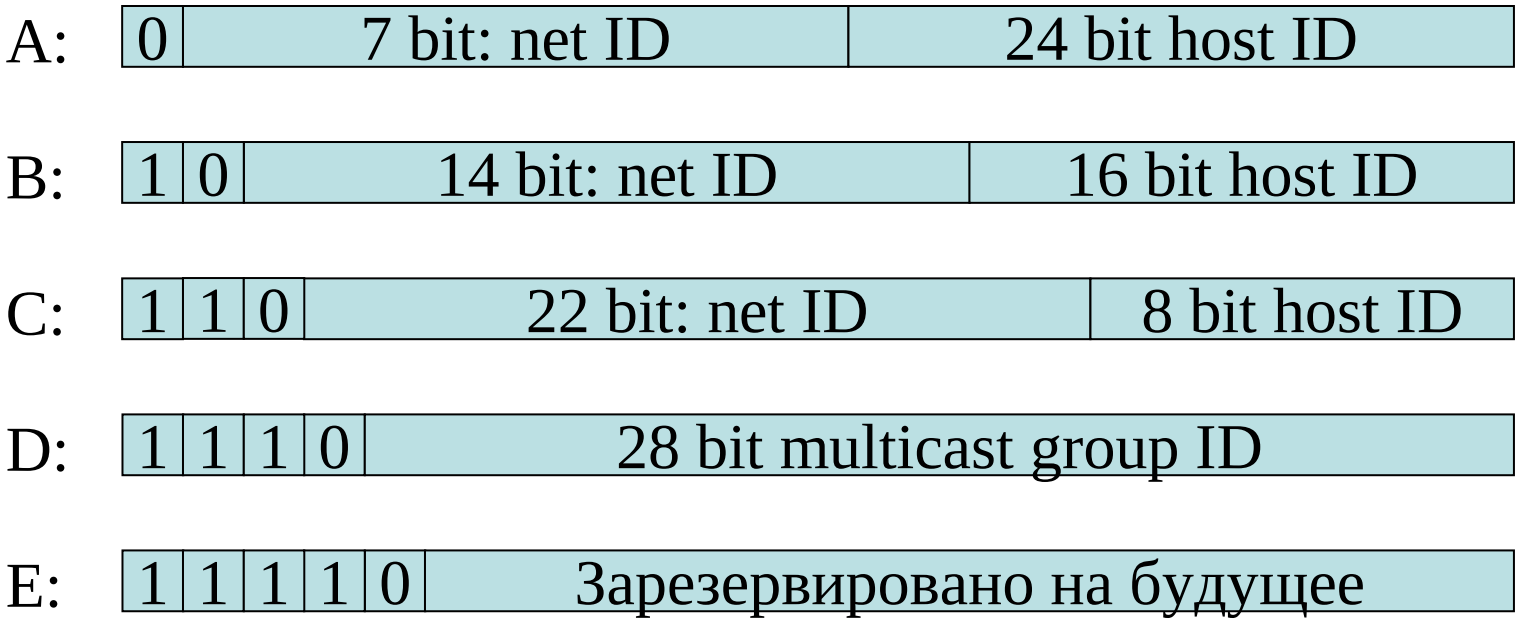
Запись IP адреса V4

Ip адрес обычно записывают в виде 4х десятичных чисел, разделенных точками, соответствующих октетам 32х разрядного двоичного числа:

10000011 01101011 00000011 00011000=>131.107.3.24

Классы сетей

Множество IP адресов(адресное пространство) структурировано: оно разбито на 5 различных классов:



Классы сетей

	Диапазон	Количество сетей	Количество узлов в сети
A	0.0.0.0 - 126.255.255.255	126	16 777 214
B	128.0.0.0 – 191.255.255.255	16 384	65 534
C	192.0.0.0 – 223.255.255.255	2 097 152	254

Маска подсети

Структурирование также возможно внутри каждой из сетей. В этом случае сеть разбивается на ***подсети (RFC950).***

Подсеть (subnet) – это физический сегмент TCP/IP сети, в котором используются адреса с общим идентификатором сети.

Использование подсетей позволяет:

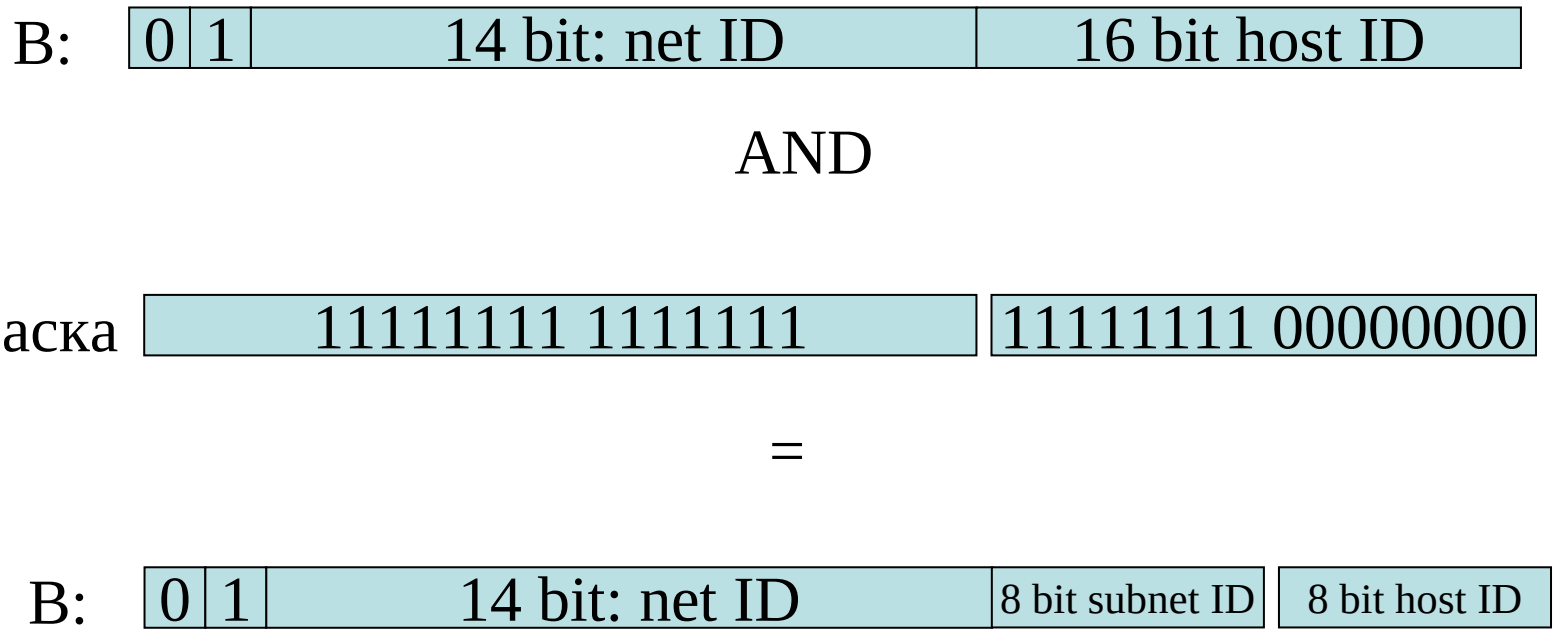
- совместно использовать различные сетевые технологии (Ethetnet,TokenRing)
- преодолеть существующие ограничения, например, на максимальное количество узлов в одном сегменте
- уменьшить нагрузку на сеть, перенаправляя сетевой трафик и сокращая число широковещательных пакетов
- Рационально использовать выделенное NIC TCP пространство адресов

Маска подсети

Маска подсети (subnet mask) – это 32х разрядное значение, используемое для выделения (маскирования) из IP-адреса из его частей: ид сети и подсети. Такая процедура необходима при выяснении относится тот или иной адрес к локальной или удаленной сети. Разбиение на подсети позволяет уменьшить размер маршрутных таблиц, поскольку внешним маршрутизаторам в этом случае достаточно знать маршрут лишь к одному узлу из сети.

В маске единицы соответствуют ИД сети, а 0 –ИД хоста. Проверка на принадлежность выполняется путем выполнения операции логического «И».

Маска подсети



Контрольные вопросы

Определите маску для различных ситуаций:

- Адрес класса A
- Адрес класса B для ЛС из 4000 узлов
- Адрес класса C для ЛС из 256 узлов
- Какие из перечисленных адресов не могут быть назначены узлам? 130.107.256.80 ; 0.10.31.20 ; 192.168.6.1 ; 1.1.1.0 ; 2.2.2.2 ; 212.20.65.90, 192.168.6.4/255.255.255.252 ; 192.168.6.2/255.255.255.0 192.168.6.1/255.255.255.255 ; 192.168.6.20/255.255.255.18

Интерфейс внутренней петли

loopback interface, lo – позволяет клиенту и серверу, действующим на одном и том же хосте обмениваться друг с другом IP пакетами,не покидающими пределов хоста.Для такого интерфейса зарезервирован ид сети класса A, равный 127 и именем localhost.

- все пакеты, адресованные интерфейсу 127.x.x.x направляются на вход того же самого IP-модуля
- все пакеты, адресованные одному из собственных IP адресов направляются на его интерфейс внутренней петли
- все широковещательные пакеты перед их отправкой в наружу копируются на интерфейс внутренней петли

Специальные разновидности IP адресов

IP адрес			Может ли быть адресом		Описание
ид. сети	ид. подсети	ид хоста	источника	назначения	
0		0	Да	Нет	«Данный» хост «данной» сети
0		host ID	Да	Нет	Указанный хост «данной» сети
127		любой	Да	Да	Адрес внутренней петли
-1		-1	Нет	Да	Местное широковещание (не подлежит транзиту)
netID		-1	Нет	Да	Широковещание на сеть netID
netID	subnetID	-1	Нет	Да	Широковещание на подсеть subnetID сети netID
netID	-1	-1	Нет	Да	Широковещание на все подсети netID

широковещание и групповая рассылка

- DHCP
- настройка маршрутизации
- доставка пакетов по нескольким адресам(UDP)
- возможность отыскивания клиентами своих серверов (WINS,NetBios Over TCP)

Специальные разновидности IP адресов и подсети

1. 192.168.x.x – выделяются для корпоративных сетей
2.192.168.6.x 255.255.255.0 -> 254 хоста
   192.168.6.0 – ИД подсети
   192.168.6.255 Широковещание

3. 192.168.6.0 255.255.255.252 -> 2 хоста
   192.168.6.0 – ИД подсети
   192.168.8.3 – Широковещание

Возможные варианты разбиения (на равные подсети)
254 хоста в 1 подсети на 256 адресов
240 хостов в 8 подсетях на 32 адреса
224 хоста в 16 подсетях на 16 адресов
192 хоста в 32 подсетях на 8 адресов
128 хостов в 64 подсетях 4 адреса

Количество IP адресов

- По сообщениям ICANN на с май 2007 года пул IPV4 адресов израсходован на 81%
- Даже при сохранении существующей динамики, адреса закончатся 2010 г.

Ipv6

В IPV6 применена принципиально иная структура пакета, не совместимая с v4.

- расширенное адресное пространство(128 бит): 3x10^38
- упрощенный формат заголовка: все что не входит в обязательной заголовок может быть размещено в необязательных расширениях.
- поддержку ориентированного на реальное время трафика
- возможность непредвиденного расширения

IPV6

- На сегодня мейнстримом в Глобальной сети является протокол IPv4, 32-битная архитектура которого позволяет адресовать до 4,3 млрд адресов в интернете.
- Новый же IPv6 имеет 128-битную структуру и способен адресовать в миллиарды раз больше интернет-узлов, максимум

340 282 366 920 938 463 463 374 607 431 768 211 456

уникальных адресов.

Каждой песчинке на планете можно будет присвоить миллион адресов и еще останется приличный буферный пул для адресации к песчинкам на луне.

Проблемы IPV6

- Адрес запомнить практически невозможно, работать человеку с ним неудобно- придется безоговорочно доверять DNS (2001:fecd:ba23:cd1f:dcb1:1010:9234:4088)
- Множество существующего прикладного ПО не работает с IPV6
- Не 100% решена проблема совместимости с IPv4
- Стандарт IPv6 опубликован более 10 лет назад, но до сих пор даже большинство вновь создаваемого ПО его не поддерживает.

Отсрочка введения IPv6 неизбежна

- Пересмотр существующих пулов
- Оптимизация существующего адресного пространства

"Еще до периода интернет-бума в начале 1990-х крупные компании, такие как HP, Apple, Ford, General Electric и другие напрямую от IANA получили примерно по 16 млн ip-адресов каждый. Адреса в таком количестве никогда этими компаниями не использовались, они просто воспользовались возможностью и получили крупные блоки IP" – отмечает глава ARIN Джон Каррен.

Адреса в IPv6

- Стандартом определено три типа адресов:

- **Индивидуальный (Unicast)** – адрес отдельного интерфейса. Пакет, отправленный по индивидуальному адресу, доставляется на интерфейс, идентифицируемый этим адресом.
- **Произвольный (Anycast)** – адрес набора интерфейсов, обычно относящихся к различным узлам. Пакет, направленный по «произвольному» адресу, доставляется на один из интерфейсов, идентифицированных этим адресом (на «ближайший», согласно метрике протокола маршрутизации).
- **Групповой (Multicast)** – адрес набора интерфейсов, обычно относящихся к различным узлам. Пакет, посланный по групповому адресу, доставляется на все интерфейсы, идентифицированные этим адресом.
- В IPv6 отсутствуют широковещательные адреса (broadcast). Вместо них используются групповые адреса.

Адреса в переходный период

- адреса IPv6, совместимые с IPv4. Такие адреса присваиваются узлам сети, осуществляющим туннелирование трафика IPv6 через инфраструктуру IPv4 (действие, необходимое в переходный период);

(Адреса первого типа представляются естественным образом — 96 нулевых бит и адрес IPv4 в младших 32-х битах)

- адреса IPv4, отображенные на IPv6. Такие адреса присваиваются узлам, поддерживающим только IPv4 (в переходный период, разумеется, будут и такие)

80 бит	16	32 бита
0000.....0000	FFFF	адрес IPv4

Адреса, локальные в пределах физической сети

- Аналог 192.168

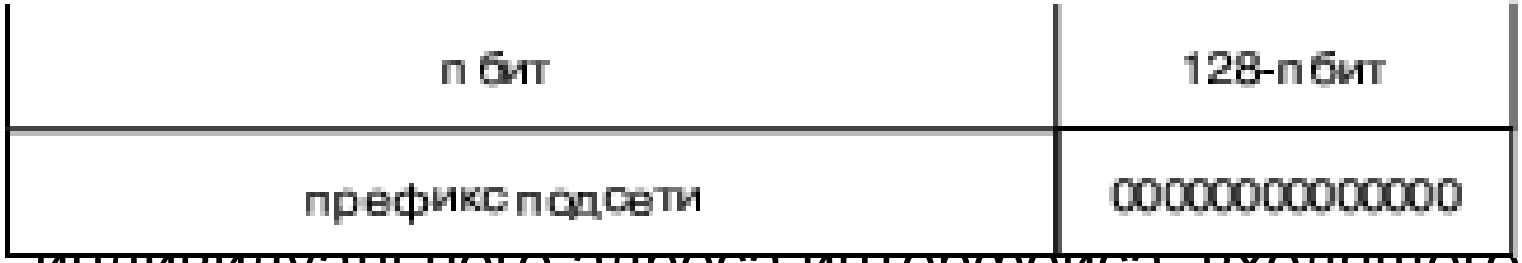
10 бит	54 бита	64 бита
1111111010	0	interface ID

Формат адресов, локальных в пределах организации

- Аналог 192.168

10 бит	38 бит	16 бит	64 бита
1111111011	0	subnetID	interface ID

Формат anycast-адреса маршрутизатора подсети



- индивидуального адреса интерфейса, входящего в данную подсеть и имеющего нулевое значение поля Interface ID. Пакет с таким адресом будет доставлен одному маршрутизатору в подсети, а понимать его должны все маршрутизаторы. Адреса данного вида могут использоваться, например, при взаимодействии мобильной системы с сервером удаленного доступа

Групповые адреса

- Спецификации IPv6 предусматривают весьма общий, практически неструктурированный формат групповых адресов Лишь бит T (единственный пока определенный элемент поля флагов) позволяет различить постоянные, общеизвестные (T=0) и временные (T=1) адреса, а 4-битное поле scop задает область их действия в соответствии со следующим перечнем:
- 1 — группа локальна в пределах узла сети;
- 2 — группа локальна в пределах физической (под)сети;
- 5 — группа локальна в пределах производственной площадки;
- 8 — группа локальна в пределах организации;
- 14 — группа является глобальной
- (остальные значения scop еще не распределены или зарезервированы).

8 бит	4	4	112 бит
1111111111	000T	scop	group ID

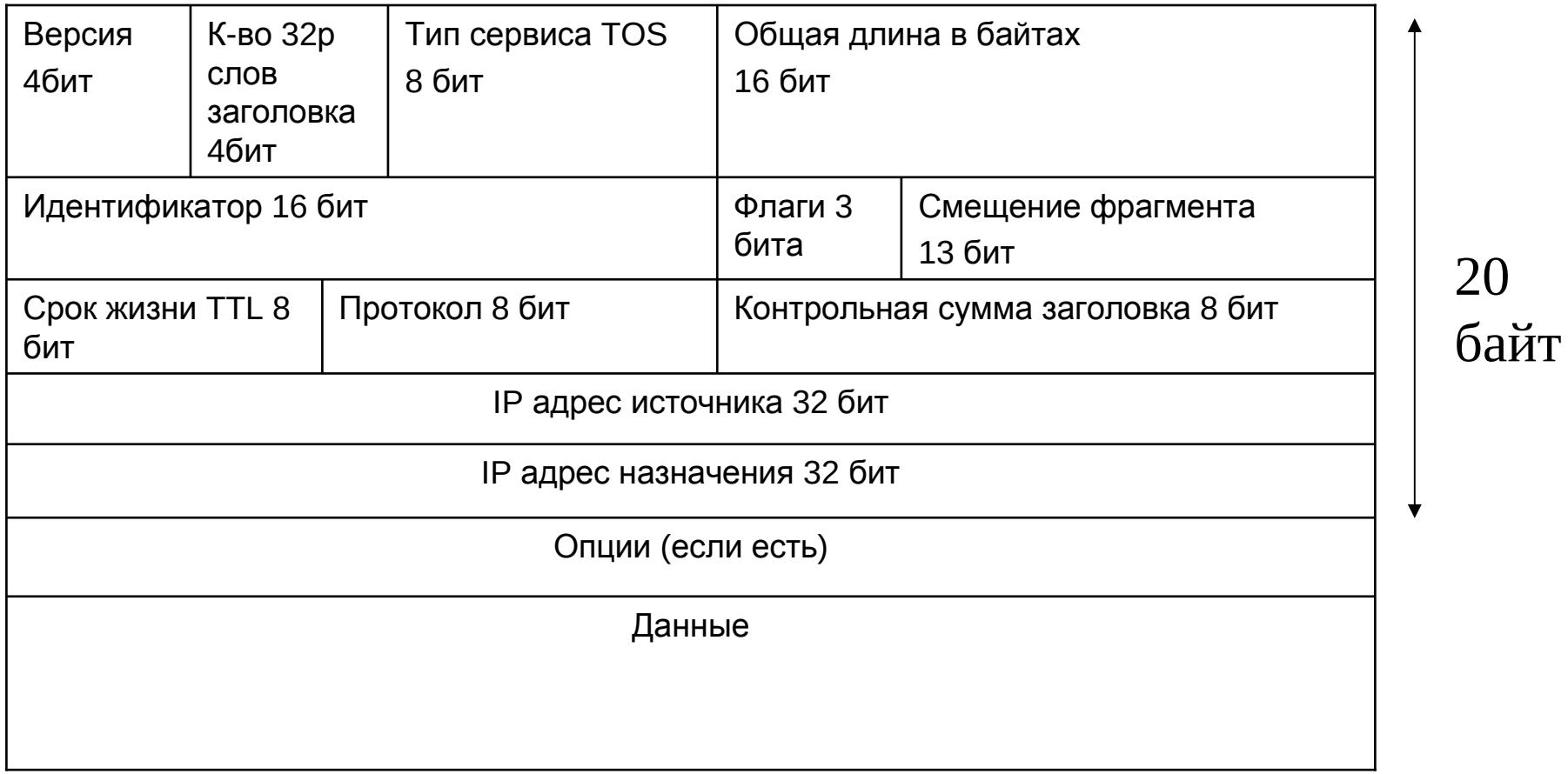
Адреса по типу

- Семантика постоянных адресов не зависит от области их действия. Например, группе "серверы NTP" (Network Time Protocol) выделен шестнадцатеричный идентификатор 101. Следовательно, адрес
- FF02:0:0:0:0:0:0:101
- (scop=2) обозначает NTP-серверы в пределах одной подсети, а
- FF0E:0:0:0:0:0:0:101
- (scop=14) — все NTP-серверы в Интернет.
- Среди предварительно распределенных групповых адресов отметим широковещательные адреса, адреса всех маршрутизаторов и адреса, затребованные узлами.

Протокол IPv4- структура

- В семействе TCP-IP протоколу IP отведена роль «рабочей лошадки». В IP пакетах (или дейтаграммах) передаются все данные TCP,UDP,ICMP,IGMP.
- IP – по определению не гарантирующий доставку сервис. Контроль доставки обеспечивают протоколы «верхнего» уровня (TCP)
- IP не поддерживающий соединение сервис. Очередность доставки пакетов может быть нарушена.

Структура пакета IPV4(RFC 791)



TOS – (type of service)

минимальная стоимость

Минимальная задержка

Максимальная производительность

Максимальная надежность

Структура IP пакета/опции

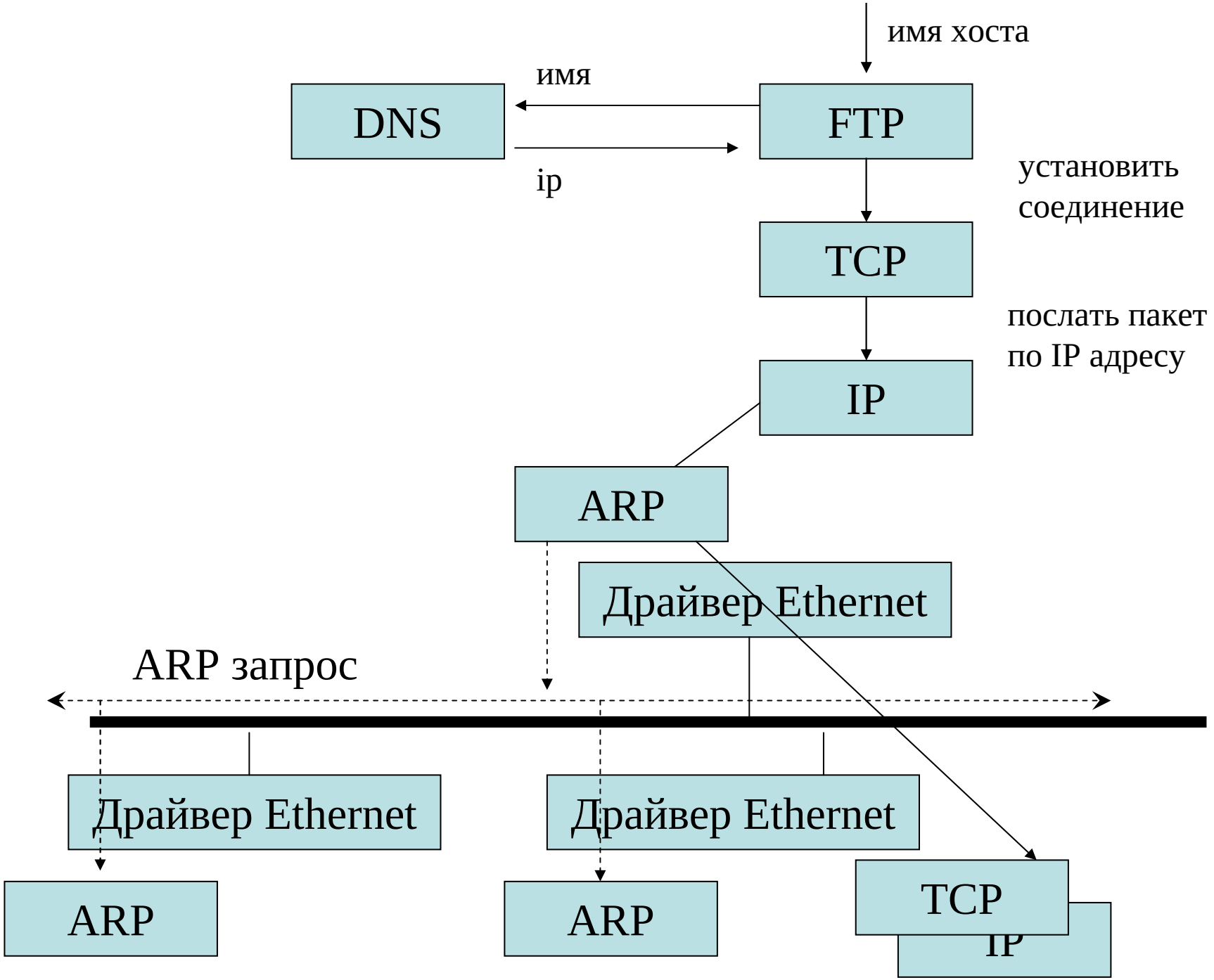
необязательные опции – дополняется до границы 32 разрядного слова

- Защита данных (секретные приложения оборонного значения)
- запись маршрута
- штемпель времени
- гибкая маршрутизация от источника
- жесткая маршрутизация от источника

Протокол отображения адресов ARP

Рассмотренные IP адреса могут восприниматься лишь на сетевом и вышестоящем уровнях TCP/IP. На канальном уровне действует иная схема адресации, например, в Ethernet- 48-разрядный MAC-address

Управляющие драйверы сетевых карт не могут воспринимать IP-адрес. Для установки соответствия между 32 разрядными IP-адресами и действующими аппаратными адресами применяется механизм привязки адресов по протоколу ARP (RFC 826)


имя хоста
имя
DNS
FTP
ip
установить
соединение
TCP
послать пакет
по IP адресу
IP
ARP
Драйвер Ethernet
ARP запрос
Драйвер Ethernet
Драйвер Ethernet
ARP
ARP
TCP
IP

ARP

Каждый хост поддерживает в кэше динамическую ARP – таблицу. Время существования записи обычно составляет 20 минут.

утилита манипулирования- *arp [ключ]*

-a вывести список всех записей

-d host удалить запись для хоста

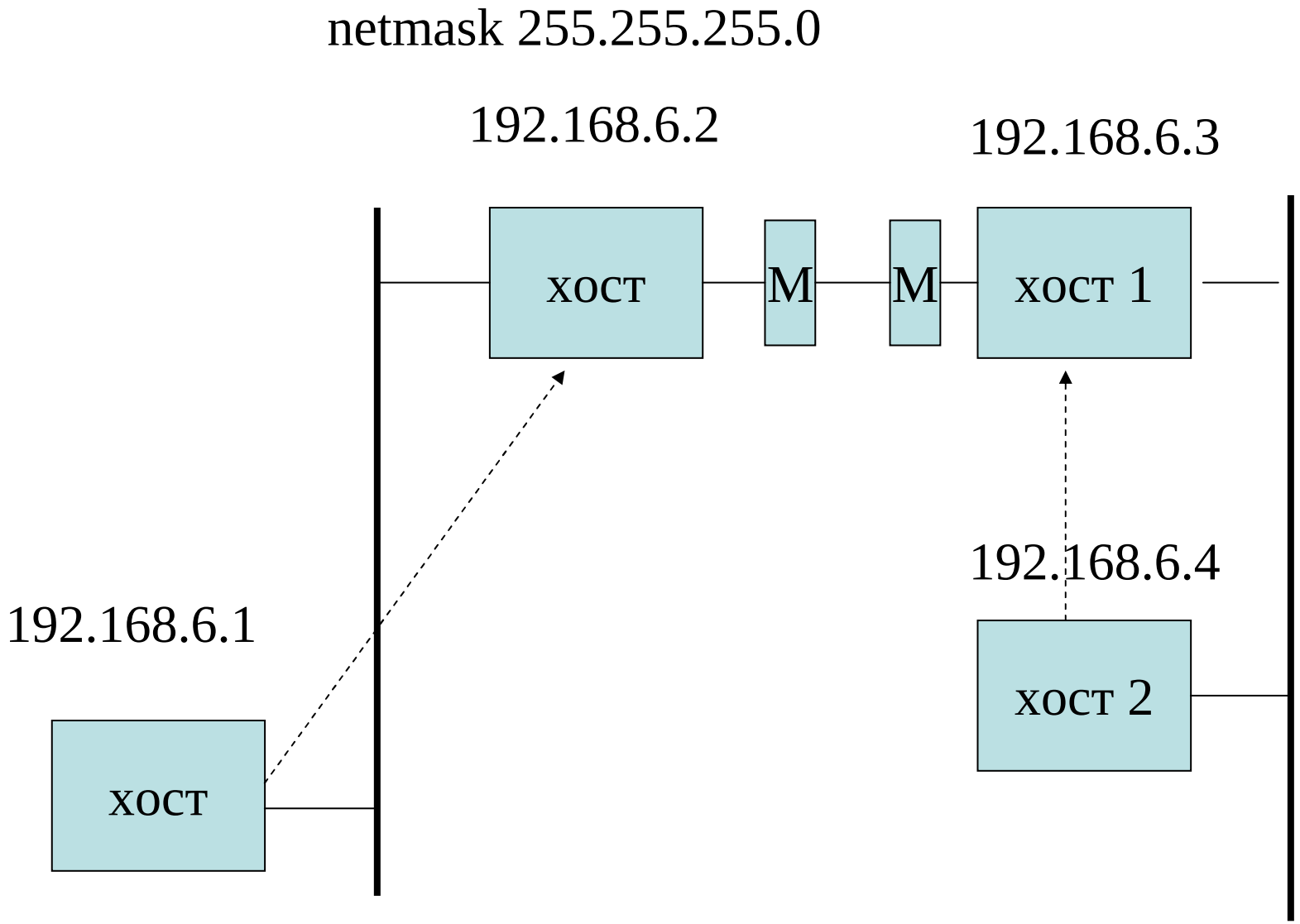
-s host hwaddr создать запись вручную

Прокси-ARP

Маршрутизатор может на уровне ARP модуля замещать хост или хосты, отвечая на предназначенные для этих хостов запросы.

Использование:

- удаленный доступ через модем
- «склеивание» двух различных физических сегмента


netmask 255.255.255.0
192.168.6.2
192.168.6.3
хост
M
M
хост 1
192.168.6.4
192.168.6.1
хост 2
хост

Протокол управляющих сообщений ICMP

(Internet Control Message Protocol) Служит для обмена сообщениями об ошибках в различных особых случаях, требующих обработки. В ICMP сообщении всегда возвращается IP заголовок и первые 8 байт пакета, признанного ошибочным.

Структура ICMP пакета

<table>
<tr><td colspan="3">IP заголовок (20 байт)</td></tr>
<tr><td>тип 8бит</td><td>код 8бит</td><td>Контрольная сумма (16 бит)</td></tr>
<tr><td colspan="3">Содержание сообщения
(зависит от кода ошибки)</td></tr>
</table>

Типы ICMP пакетов

Тип	Код	Описание	Запрос/ ответ	Ошибка
0	0	Эхо-ответ (ping)	x	
3		**Адресат недоступен:**		
	0	сеть недоступна		x
	1	хост недоступен		x
	2	протокол недоступен		x
	3	порт недоступен		x
	4	ошибка фрагментации		x
	5	маршрутизация от источника невыполнима		x
	6	сеть назначения неизвестна		x
	7	хост назначения неизвестен		x
	8	не используется		x
	9	сеть назначения административно закрыта		x
	10	хост назначения административно закрыт		x
	11	сеть недоступна для данного сервиса TOS		x
	12	хост недоступен для данного сервиса TOS		x
	13	связь административно закрыта фильтром		x
	14	нарушение старшинства хостов		x
	15	действует отключение по старшинству		x

Типы ICMP пакетов

Тип	Код	Описание	Как обрабатывается
0	0	Эхо-ответ (ping)	пользовательским процессом
3		**Адресат недоступен:**	
	0	сеть недоступна	“No route to host”
	1	хост недоступен	“No route to host”
	2	протокол недоступен	“Connection refused”
	3	порт недоступен	“Connection refused”
	4	ошибка фрагментации	“Message too long”
	5	маршрутизация от источника невыполнима	“No route to host”
	6	сеть назначения неизвестна	“No route to host”
	7	хост назначения неизвестен	“No route to host”
	8	не используется	“No route to host”
	9	сеть назначения административно закрыта	“No route to host”
	10	хост назначения административно закрыт	“No route to host”
	11	сеть недоступна для данного сервиса TOS	“No route to host”
	12	хост недоступен для данного сервиса TOS	“No route to host”
	13	связь административно закрыта фильтром	(игнорируется)
	14	нарушение старшинства хостов	(игнорируется)
	15	действует отключение по старшинству	(игнорируется)

Тип	Код	Описание	Запрос/ ответ	Ошибка
4	0	Прикрыть источник		x
5		**Перенаправление (redirect)**		
	0	перенаправить путь на сеть		x
	1	перенаправить путь на хост		x
	2	перенаправить путь на сеть для типа TOS		x
	3	перенаправить путь на хост для типа TOS		x
8	0	Эхо -запрос	x	
9	0	Объявление маршрутизатора	x	
10	06	запрос маршрутизатора	x	
11		Срок истек		
	0	Срок истек при переходе на TTL=0		x
	1	Срок истек при сборке		x
12		Нарушены параметры пакета(parametr problem)		
	0	Испорчен IP заголовок		x
	1	отсутствует необходимая опция		x
13	0	Запрос отсчета времени	x	
14	0	Отклик отсчета времени	x	
17	0	Запрос адресной маски	x	
18	0	Ответ адресной маски	x	

Тип	Код	Описание	Где и как обрабатывается
4	0	Прикрыть источник	Ядром в TCP
5	 0 1 2 3	**Перенаправление (redirect)** перенаправить путь на сеть перенаправить путь на хост перенаправить путь на сеть для типа TOS перенаправить путь на хост для типа TOS	Вызывает обновление маршрутной таблицы (ядром)
8	0	Эхо -запрос	ядром ген. эхо-отклик
9 10	0 06	Объявление маршрутизатора запрос маршрутизатора	пользовательским процессом
11	 0 1	Срок истек Срок истек при переходе на TTL=0 Срок истек при сборке	пользовательским процессом
12	 0 1	Нарушены параметры пакета(parametr problem) Испорчен IP заголовок отсутствует необходимая опция	 “Protocol not available” “Protocol not available”
13 14	0 0	Запрос отсчета времени Отклик отсчета времени	ядром ген. ответ польз. процессом
17 18	0 0	Запрос адресной маски Ответ адресной маски	ядром ген. ответ польз. процессом

ICMP

для предотвращения «широковещательных штормов» ICMP сообщение никогда не генерируется в ответ на:

- другое сообщение об ошибке (кроме запроса)
- пакет с широковещательным или групповым адресом
- на широковещательный кадр канального уровня
- на любой фрагмент пакета, кроме его первого пакета
- на пакет, в котором адрес источника не определяет конкретный хост(0,lo,группа,шир.)

Пакет IPv6

- Пакет в IPv6 включает стандартный заголовок, произвольное число дополнительных (необязательных) заголовков, а также "полезную нагрузку" — заголовки и данные протоколов более высоких уровней

Спецификация IPv6

Спецификации IPv6 (см. [*])
применительно к форматам пакетов определяют три принципиально важных новых аспекта:

- порядок заголовков;
- формат стандартного заголовка IPv6;
- форматы дополнительных заголовков

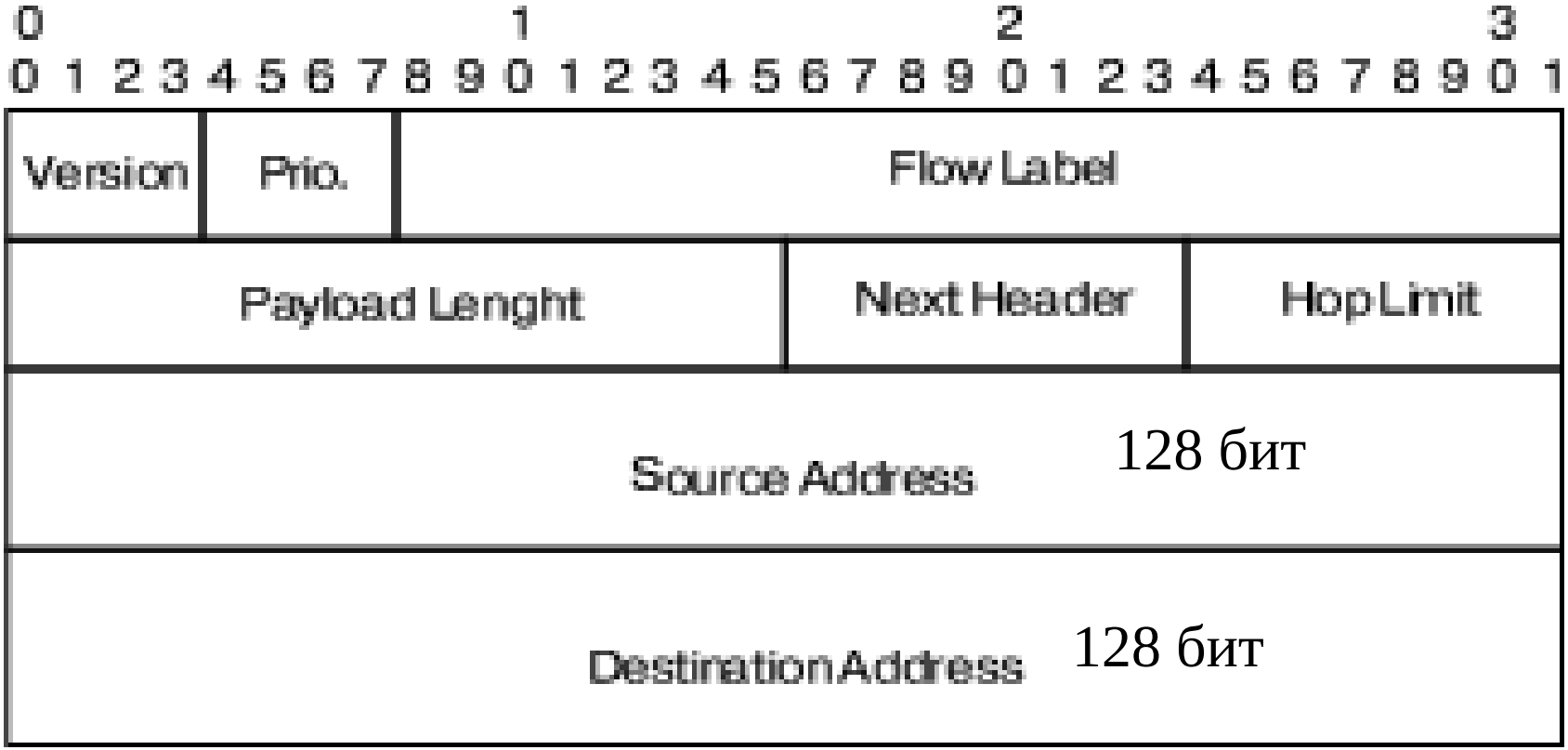
Дополнительные заголовки

- В IPv4 суммарная длина дополнительных заголовков не могла превышать 40 байт. В IPv6 это ограничение снято, дополнительные заголовки могут быть сколь угодно длинными (в пределах максимального размера пакета, разумеется) и сложными. Тем самым в IPv6 изначально заложены достаточно мощные и гибкие средства расширения.

Пакеты не могут фрагментироваться

- Обратим внимание на то, что в IPv6 **пакеты не могут фрагментироваться и собираться маршрутизаторами**. Отправитель должен заранее выяснить максимальный размер пакетов (Maximum Transmission Unit, MTU), поддерживаемый на всем пути до получателя, и, при необходимости, выполнить фрагментацию своими силами. (Оговаривается, что MTU не может быть меньше 576 байт; вероятно, в последующих версиях спецификаций IPv6 это значение возрастет до 1500 байт.) Снятие с маршрутизаторов забот о фрагментации также способствует повышению эффективности их работы, хотя и усложняет в определенной степени жизнь оконечным системам.

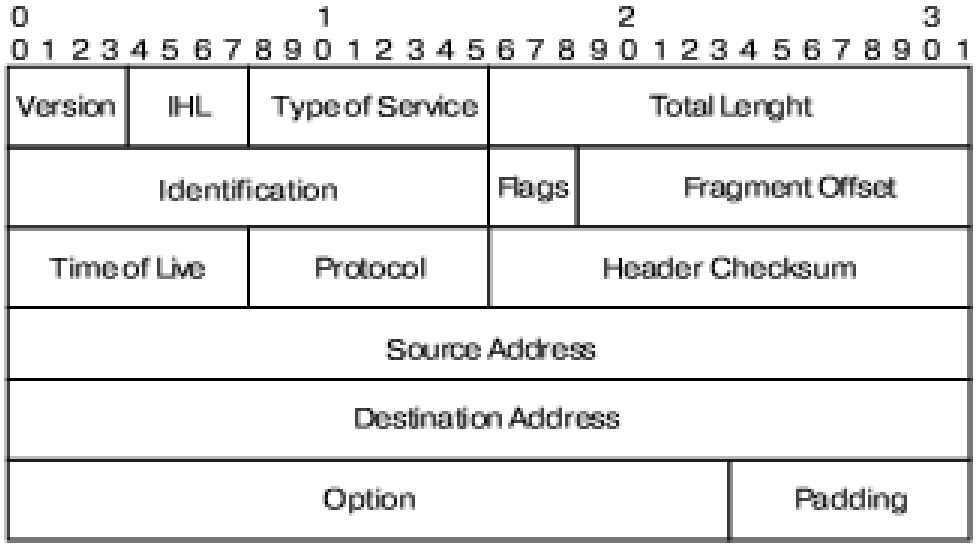
Стандартный заголовок



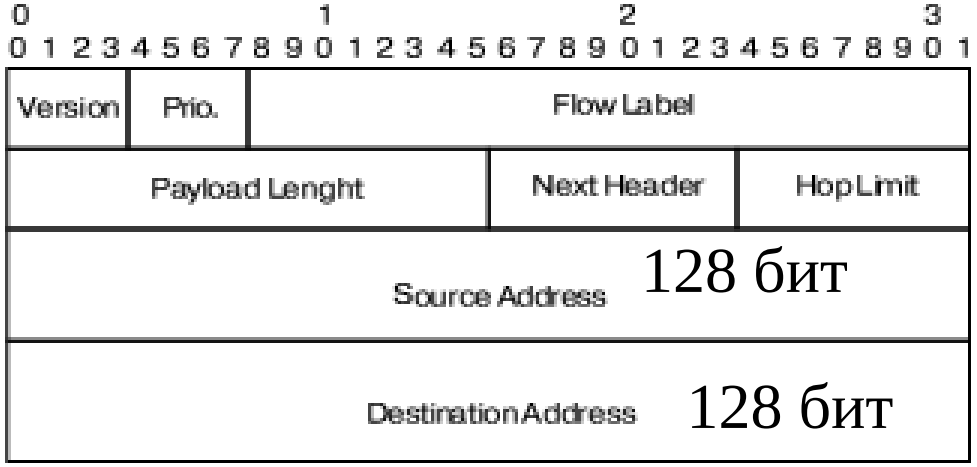
Заголовок

- **Version** — номер версии IP-протокола (6);
- **Prio**. — приоритет пакета;
- **Flow Label** — метка IP-потока.
- **Payload Length** — длина содержимого, то есть того, что следует в пакете за заголовком IPv6 (дополнительные заголовки сетевого уровня, заголовки и данные протоколов более высокого уровня);
- **Next Header** — номер (тип) следующего заголовка (дополнительного заголовка IP-уровня или заголовка транспортного уровня);
- **Hop Limit(TTL IPV4)** — максимальное число промежуточных систем на пути следования пакета. Уменьшается каждым маршрутизатором на 1. Если Hop Limit становится равным 0, пакет ликвидируется;
- **Source Address** — 128-битный исходный адрес;
- **Destination Address** — 128-битный целевой адрес.

Сравнение IPV6 и IPv4



IPV4 20-40 байт



IPV6 40 байт

Сравнение базовых заголовков:

- В IPv6 заголовок стал проще, он имеет фиксированную длину (40 байт). Хотя размер IP-адреса увеличился вчетверо (с 32 до 128 бит), длина заголовка возросла лишь вдвое (20-40)
- Часть полей, присутствовавших в IPv4, ликвидирована (помимо длины заголовка это контрольная сумма заголовка).
- Еще одна группа полей перекочевала в дополнительные заголовки. Имеется в виду все, что связано с фрагментацией, а также опции, следующие в пакете IPv4 за адресами.
- Поля времени жизни пакета (Time to Live) и протокола (Protocol) в общем и целом лишь сменили названия, соответственно, на Hop Limit и Next Header, с некоторым уточнением (и обобщением) трактовки.
- Однобайтное поле Type of Service расширилось в IPv6 до двух полей, Prio. и Flow Label, с суммарным размером 4 байта и гораздо более богатой семантикой.

Дополнительный заголовок

Дополнительные заголовки используются в IPv6 для поддержки механизмов безопасности, фрагментации, сетевого управления и т.п. Их общее количество и внутренняя сложность практически не ограничены. Мы не будем рассматривать их подробно.

Резюме.

Видим, что революции не произошло

- Наведен порядок
- Убрана избыточность
- Добавлена необходимая функциональность
- Все что не обязательно - выносится в дополнительный заголовок
- Изменились механизмы фрагментации- они перенесены на «верхний уровень»(хотя формально механизм фрагментации пакетов, превышающих MTU существует)

Лекция

Лекция 6.

TCP-IP

- протоколы канального уровня(продолжение)
- Сетевые утилиты и команды
- маршрутизация
- DNS

Ifconfig

Перед началом работы с сетью в Linux необходимо активировать («поднять») сетевой интерфейс. Анализ состояния и активация сетевого интерфейса выполняется при помощи команды ifconfig.

ifconfig *[interface] [address] [**netmask** mask] [**mtu** bytes] [arp/-arp] [promisc/-promisc] [**up** / **down**]*

Вызов ifconfig без параметров позволяет получить список активных интерфейсов и их состояния. По умолчанию сетевой интерфейс обычно конфигурируется из скрипта начальной загрузки.

Команда ifconfig распространена и в других семействах протоколов и ОС.

ifconfig

eth0 Link encap:Ethernet HWaddr 00:0C:29:BF:BD:5A

Анализатор пакетов

Прослушивает все кадры, передаваемые в физической линии. Требуется возможность переключения сетевой карты из selective mode -> promiscuous mode

tcpdump [флаги] –i interface

-c N – принять только N ракетов

-e – отключить анализ содержимого

-a/-n – пытаться/не выполнять поиск имени

Допускается создание собственных фильтров:

tcpdump tcp port 25

tdpdump ‘icmp[0] != 8 and icmp[0] !=0’

В базовой поставке ОС Windows подобной команды нет (необходима установки программы-сниффера)

Пакетный локатор ping

Разработал Mike Muuss. Термин взят из «военно-морского» лексикона-эхо локация. Программа проверяет доступность адресата, посылая ему ICMP сообщение и обеспечивает хронометраж.

Использование: ping [флаги] хост

- Параметры:
- -a Определение адресов по именам узлов.
- -n число Число отправляемых запросов.
- -l размер Размер буфера отправки.
- -f Установка флага, запрещающего фрагментацию пакета.
- -i TTL Задание срока жизни пакета (поле "Time To Live").
- -v TOS Задание типа службы (поле "Type Of Service").
- -r число Запись маршрута для указанного числа переходов.
- -s число Штамп времени для указанного числа переходов.
- -j списокУзлов Свободный выбор маршрута по списку узлов.
- -k списокУзлов Жесткий выбор маршрута по списку узлов.
- -w таймаут Таймаут каждого ответа в миллисекундах.

Ping с опцией запись маршрута

```
PING www.lenta.ru (81.19.69.28): 56 octets data
64 octets from 81.19.69.28: icmp_seq=0 ttl=55 time=68.5 ms
RR:     90.28.106.217
        250.21.106.217
        134.6.106.217
        130.6.106.217
        122.6.106.217
        162.7.106.217
        252.1.161.195
        1.64.19.81
        1.69.19.81
64 octets from 81.19.69.28: icmp_seq=1 ttl=55 time=68.0 ms      (same route)
64 octets from 81.19.69.28: icmp_seq=2 ttl=55 time=74.1 ms      (same route)
64 octets from 81.19.69.28: icmp_seq=3 ttl=55 time=67.9 ms      (same route)
```

Трассировка маршрутов

Позволяет отследить маршрут движения пакетов от одного хоста к другому.

traceroute [флаги] назначение

Для определения маршрута используется механизм TTL (time-to-life) IP заголовка. Посылается IP пакет к заведомо «бесхозному» к порту, инкрементируя TTL на 1.

В Win – *tracert*

В OS2 - tracerte

Структура пакета IPV4(RFC 791)

<table>
<tr><td>Версия
4бит</td><td>К-во 32р слов заголовка 4бит</td><td>Тип сервиса TOS
8 бит</td><td colspan="2">Общая длина в байтах
16 бит</td></tr>
<tr><td colspan="3">Идентификатор 16 бит</td><td>Флаги 3 бита</td><td>Смещение фрагмента
13 бит</td></tr>
<tr><td colspan="2">Срок жизни TTL 8 бит</td><td>Протокол 8 бит</td><td colspan="2">Контрольная сумма заголовка 8 бит</td></tr>
<tr><td colspan="5">IP адрес источника 32 бит</td></tr>
<tr><td colspan="5">IP адрес назначения 32 бит</td></tr>
<tr><td colspan="5">Опции (если есть)</td></tr>
<tr><td colspan="5">Данные</td></tr>
</table>

Пример трассировки

Трассировка маршрута к lenta.ru [81.19.69.28]
с максимальным числом прыжков 30:

```
 1  nsk-car0-fa0-1-111.rt-comm.ru (217.106.28.89)  389.504 ms  435.325 ms  508.130 ms
 2  217.106.21.254 (217.106.21.254)  574.006 ms  638.453 ms  815.510 ms
 3  kochenevo-bbn0-po3-2.rt-comm.ru (217.106.6.133)  713.968 ms  74.965 ms  66.299 ms
 4  tschelkun-bbn0-po1-1.rt-comm.ru (217.106.6.129)  59.622 ms  127.427 ms  60.574 ms
 5  shigony-bbn0-po9-3.rt-comm.ru (217.106.6.97)  68.661 ms *  110.811 ms
 6  msk-bbn1-po2-3.rt-comm.ru (217.106.6.65)  125.506 ms  219.212 ms  63.673 ms
 7  msk-dsr0-ge0-3-0-0.rt-comm.ru (217.106.7.194)  57.580 ms  60.349 ms  60.540 ms
 8  * 213.59.1.250 (213.59.1.250)  375.175 ms  237.812 ms
 9  sl6509-v27.ramtel.ru (81.19.64.13)  70.976 ms  63.825 ms  65.378 ms
10  lenta12.cust.ramtel.ru (81.19.69.28)  154.240 ms  69.338 ms  62.310 ms
```

Трассировка завершена.

Трассировка по «жесткому» или «гибкому» маршруту

(Маршрутизация от источника)

Жесткий: traceroute –G getw1 –G getw2 –G getw3 target

гибкий: traceroute –g getw1 –g getw2 –g getw3 target

Опции IP пакета:

код 1б	длина 1б	Указ. 1	адрес1 4б	адрес2 4б	…	адрес9 4б

Если невозможно: ICMP (source route not found)

IP-маршртизация

Маршрутизация - важнейшая из основных функций уровня IP. Источником пакетов, подлежащих маршрутизации на хосте может быть как сам хост, так и любой другой компьютер в сети.

За процесс маршрутизации отвечает маршрутизирующий *демон*. В Linux чаще всего используются routed и gated.

Определения

Демон - прикладной процесс, который действует на прикладном уровне, но предназначен не для обслуживания пользователя, а для реализации системных функций.

Маршрутизатор: устройство, позволяющее перенаправлять пакеты между сетями в соответствии с заданными правилами.

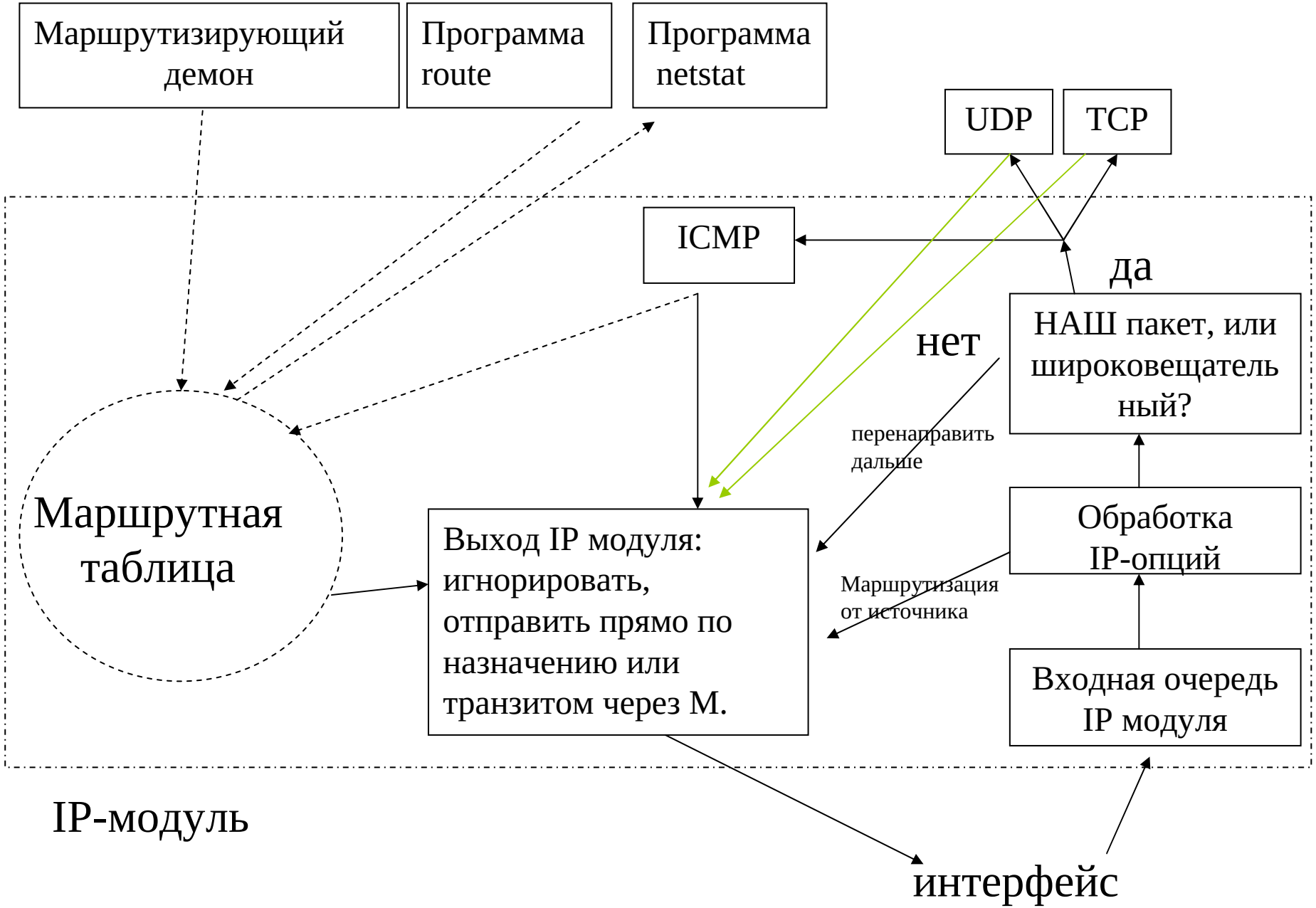
Маршрутная таблица: список правил (маршрутов), регламентирующих путь следования пакета от адресата к адресуемому

Механизм маршрутизации: процесс, когда система просматривает маршрутную таблицу и выбирает интерфейс для отправки пакета

Политика маршрутизации (routing policy): - набор правил согласно которым в таблицу вносятся изменения.

Маршрутизатор по умолчанию – хост к которому будет перенаправлен пакет, если о адресуемом хосте ничего не известно.

IP процессы на уровне хоста



Инициализация маршрутной таблицы

Прямой путь:

После активации сетевого интерфейса ifconfig в маршрутную таблицу заносится прямой путь, открытый с этого интерфейса. Эта запись недоступна для изменения.

Путь к удаленному хосту:

- статическая маршрутизация (route)
- динамическая маршрутизация (rip)

route

route - манипулирует таблицей маршрутов.

route [add / del] [-net / -host / default] target [netmask Nm] [gw router]
    [mtu NNN]

[add / del] добавить / удалить

[-net / -host / default] маршрут к сети / хосту / по умолчанию

[Target] – пункт назначения

[Netmask Nm] – сетевая маска

[gw router] – маршрутизатор

Вызов без параметров выводит текущую таблицу маршрутов.

route add –net 192.168.6.0 netmask 255.255.255.0 gw 192.168.1.1

Без параметров- выводит текущую таблицу (netstat -r)

Пример простой таблицы

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.8.5     192.168.6.10    255.255.255.255 UGH    1     0       450 eth0
192.168.8.0     192.168.6.9     255.255.255.240 UG    1      0       343 eth0
192.168.6.0     *               255.255.255.0   U     0      0        0  eth0
127.0.0.0       *               255.0.0.0       U     0      0        0  lo
default         192.168.6.5     0.0.0.0         UG    1      0       23  eth0
```

Destination – назначение
Gateway – шлюз
Genmask – маска
Flags – флаги
Metric – метрика
Ref – список активных соединений
Use – счетчик пакетов
Iface – Название интерфейса

Флаги

- U (Up) – путь открыт
- G (Gateway) – Путь через маршрутизатор
- H (Host) – Путь к конкретному хосту
- D (Directed) Путь автоматически добавлен
- M (Modified) – автоматически скорректирован

Проверка сети при помощи netstat

Netstat- мощный инструмент проверки настроек и функционирования сети.

netstat –r – отображение маршрутов

Флаги как в route

netstat –i – статистика интерфейса

Флаги:

B – нестандартное широковещание

L – локальная передача данных

M – promisc on

O – Arp off

P – PPP link

R – Interface running

U – разрешена передача данных

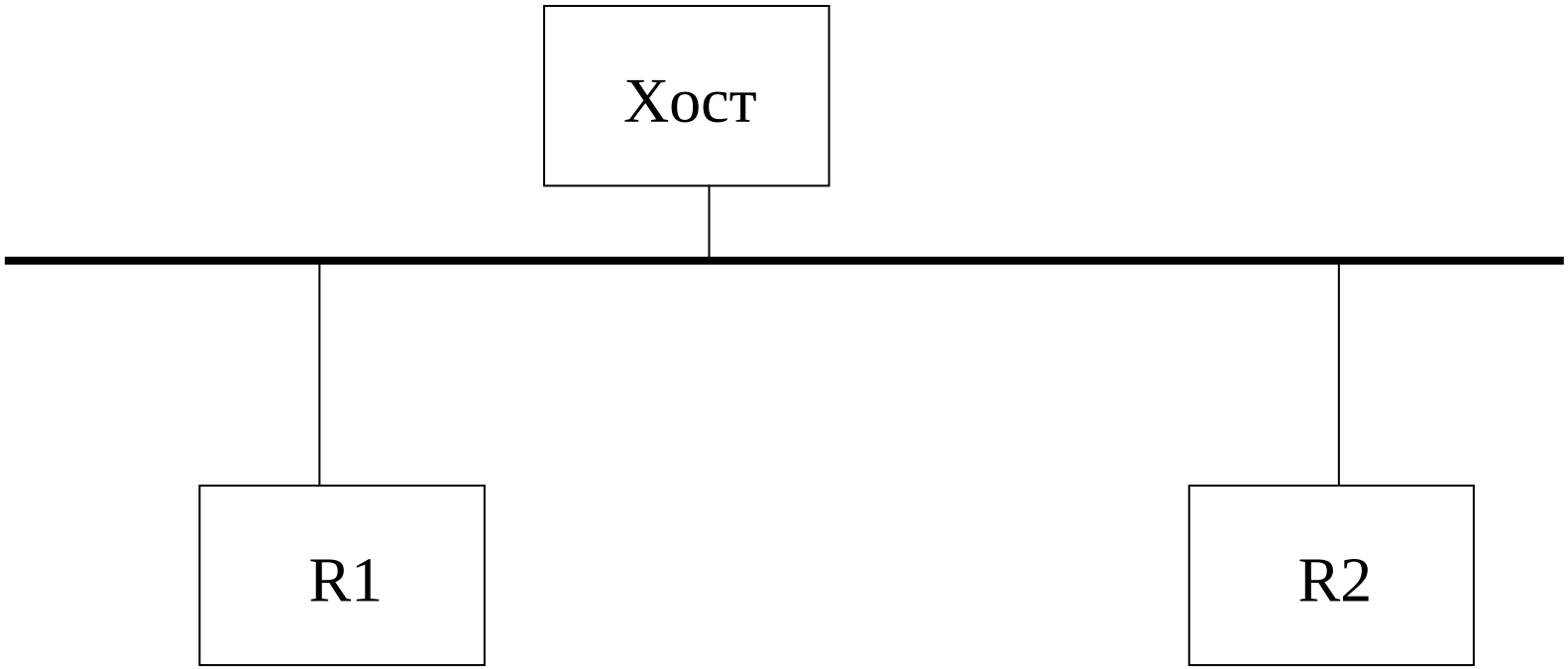
netstat –t –u –w –x отображение активных соединений.

Принципы маршрутизации

Очередность шагов, выполняемых при поиске в маршрутной таблице:

1. Поиск на совпадение среди строк с адресами хостов
2. Поиск на совпадение среди строк с адресами сетей
3. Поиск пути по умолчанию (default route)

Перенаправление пути при помощи ICMP



Перенаправления

Необходимые условия для посылки сообщения о перенаправлении:

1.Интерфейс с которого получен пакет совпадает с интерфейсом с которого он отправлен.

2. Путь по которому направлен пакет не был создан или модифицирован ICMP перенаправлениями

3. Пакет не содержит опции «маршрутизация от источника»

4. Ядро разрешает рассылку ICMP перенаправлений (ip_senddirects=1)

Перенаправления: хост

1. Рекомендованный маршрутизатор находится в одной сети
2. ICMP пакет пришел от того маршрутизатора, кому посылался пакет
3. ICMP сообщение не должно задавать в качестве адреса м. собственный адрес получившего это сообщение хоста
4. Перенаправляемый путь должен быть косвенным

ICMP-сообщения с запросами и объявлениями маршрутизаторов

Действия маршрутизатора:

После загрузки ОС маршрутизатор время от времени рассылает broadcast (multicast) пакеты с ICMP объявлениями со всех своих интерфейсов, способных к широковещанию. Интервал между объявлениями случаен (обычно 450-600с)

По умолчанию срок достоверности-30 мин.

Действия хоста:

После загрузки рассылается 3 ICMP запроса (3с) и ожидается получение первого ответа объявления маршрутизатора.

Бесклассовая междоменная маршрутизация CIDR

Обрабатываемые адреса рассматриваются как «цельные» 32х разрядные номера без разделения на сети. Принцип CIDR : объединение в «суперсети», т.е. агрегирование множества адресов сетей в группы,каждой из которых отводится одна строка в маршрутной таблице.

Агрегирование становится возможным:

- старшие биты сетей совпадают
- Маршрутные таблицы и алгоритмы маршрутизации должны быть адаптированы для обработки «маски надсети»
- Используемые протоколы маршрутизации должны сообщать не только 32-х разрядные IP адреса, но и 32х разрядные маски.

iproute Новые возможности маршрутизации в ядрах выше 2.2

Большинство дистрибутивов Linux, впрочем как и UNIX, для настройки сети и маршрутизации используют команды ifconfig, arp и route. Однако в Linux, начиная с ядра 2.2, была полностью переделана сетевая система и были добавлены новые возможности, которые ранее требовали дополнительных утилит, такие как маршрутизация на основе правил, управление трафиком и т.д. К этим возможностям предоставляет доступ пакет программ iproute2, который в настоящее время входит в большинство современных дистрибутивов.

ip

Утилита ip объединяет в себе возможности команд ifconfig, arp и route
синтаксис команды:

ip [Опции] Объект [Команда[Аргументы команды]]

Объекты:

- link - сетевое устройство
- address - IPv4 или IPv6 адрес на устройстве.
- neighbour - ARP адреса
- route - машрутизация
- rule - база данных правил машрутизации
- madress - Multicast-адреса представляют собой особый подвид широковещательных адресов, позволяющих обращаться к группе машин, которые не обязательно должны быть в той же самой подсети. Они весьма полезны при сетевых голосовых переговорах и видеоконференциях. Поддерживаются многими, но не всеми картами Ethernet.
- mroute - Multicast-пакетов.
- tunnel - туннель через IP.

Команда описывает действие над Объектом.

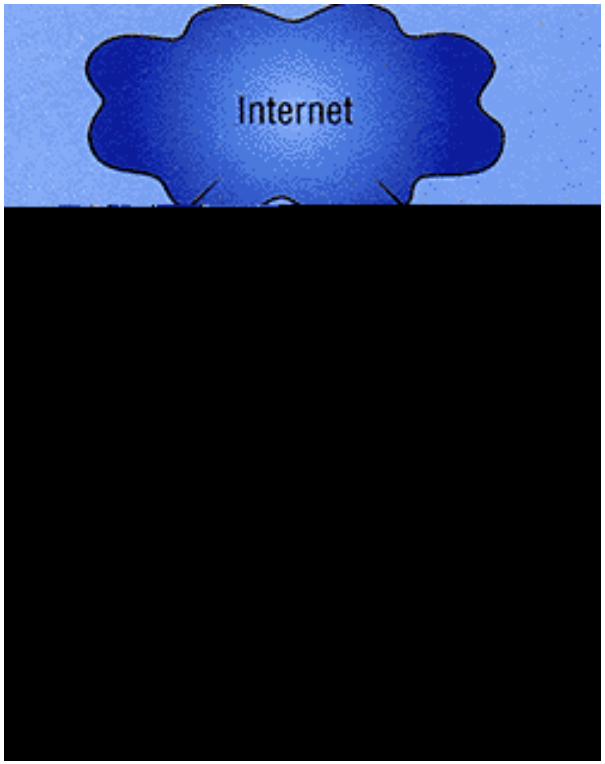
ip

Основное отличие маршрутизации: возможность создавать маршрутные таблицы ***от источника*.** Это позволяет строить сети с избыточными соединениями Internet.

Это дает возможность:

- корректно поддерживать входящие сетевые соединения от нескольких провайдеров Internet .
- обеспечивать горячее резервирование каналов
- настраивать «балансировку» трафика между каналами

Возможности команды IP




```
#Задание правил маршрутизации IP
# по источнику для DSL
ip rule add from 63.89.102.157 lookup 1
ip route add 10.0.0.0/24 via 10.0.0.1 table 1
ip route add 0/0 via 63.89.102.1 table 1
#Задание правил маршрутизации IP по источнику
#для кабельного модема
ip rule add from 65.3.17.133 lookup 2
ip route add 10.0.0.0/24 via 10.0.0.1 table 2
ip route add 0/0 via 65.3.17.1 table 2
```

- #Задание правил маршрутизации по IP-адресу источника
- ip rule add from 63.89.102.157 lookup 1
- ip rule add from 65.3.17.133 lookup 2

- #Настройка балансировки нагрузки
- ip route add default equalize
- nexthop via 63.89.102.1 dev eth1
- nexthop via 65.3.17.1 dev eth2

Документация: ftp://ftp.inr.ac.ru/ip-routing/

Система доменных имен DNS

Система доменных имен DNS (Domain Name System) – распределенная база данных, которая позволяет приложениям в TCP/IP определять взаимное соответствие между именами хостов и их IP адресами. Служба DNS основана на взаимодействии глобально рассредоточенных серверов – не существует центрального сервера, накапливающего все текущие данные о именах. Каждая отдельная организация, представленная в Сети как самостоятельно администрируемая единица, поддерживает собственную часть базы данных. За присвоение имен отвечает служба NIC

Определения

Домен- любое полное поддерево графа, начинающееся с некоторого родительского узла и содержащее дочерние узлы

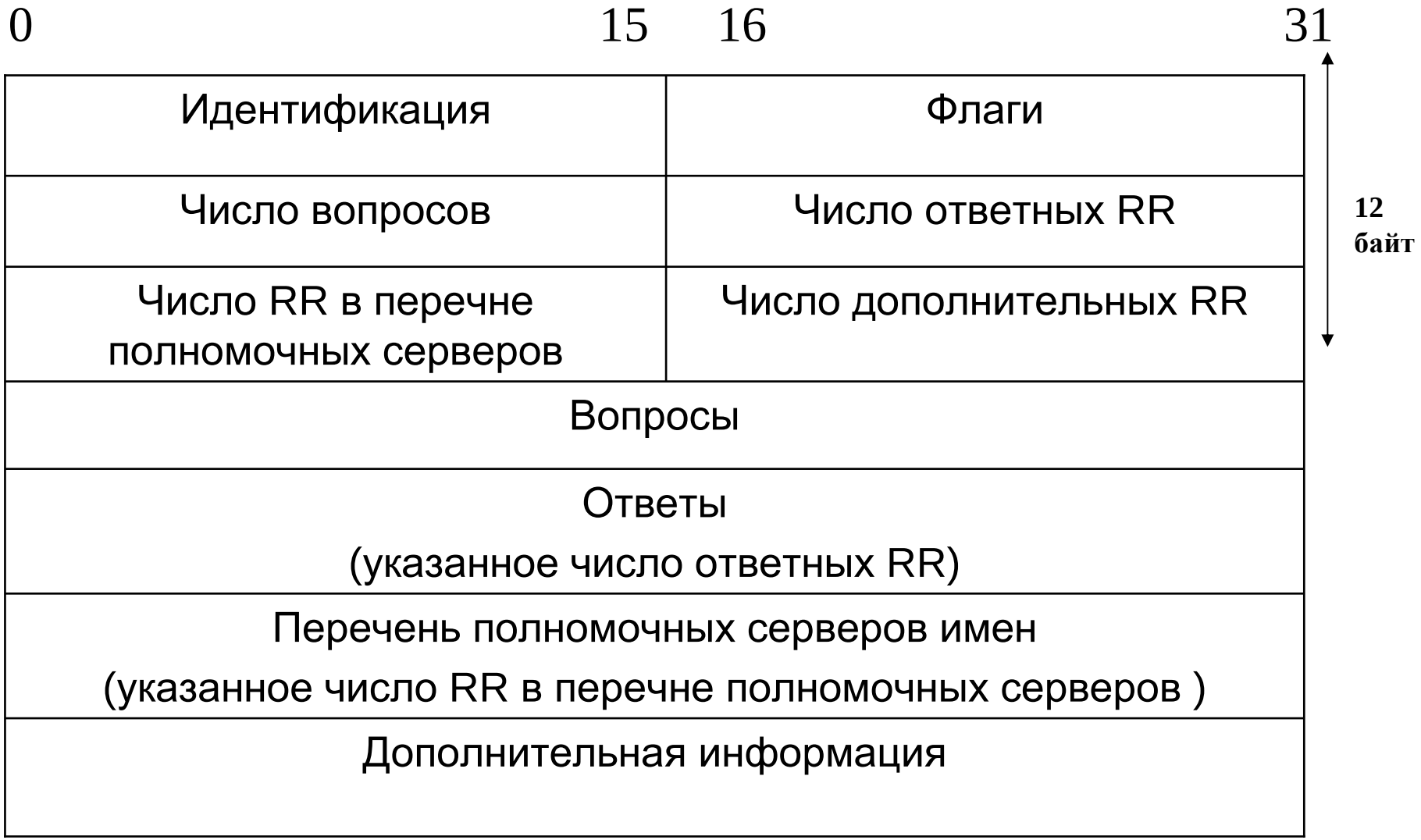
Зона – любая автономно администрируемая часть полного дерева DNS

Сервер имен: компьютер, ответственный за поддержку базы данных имен в зоне

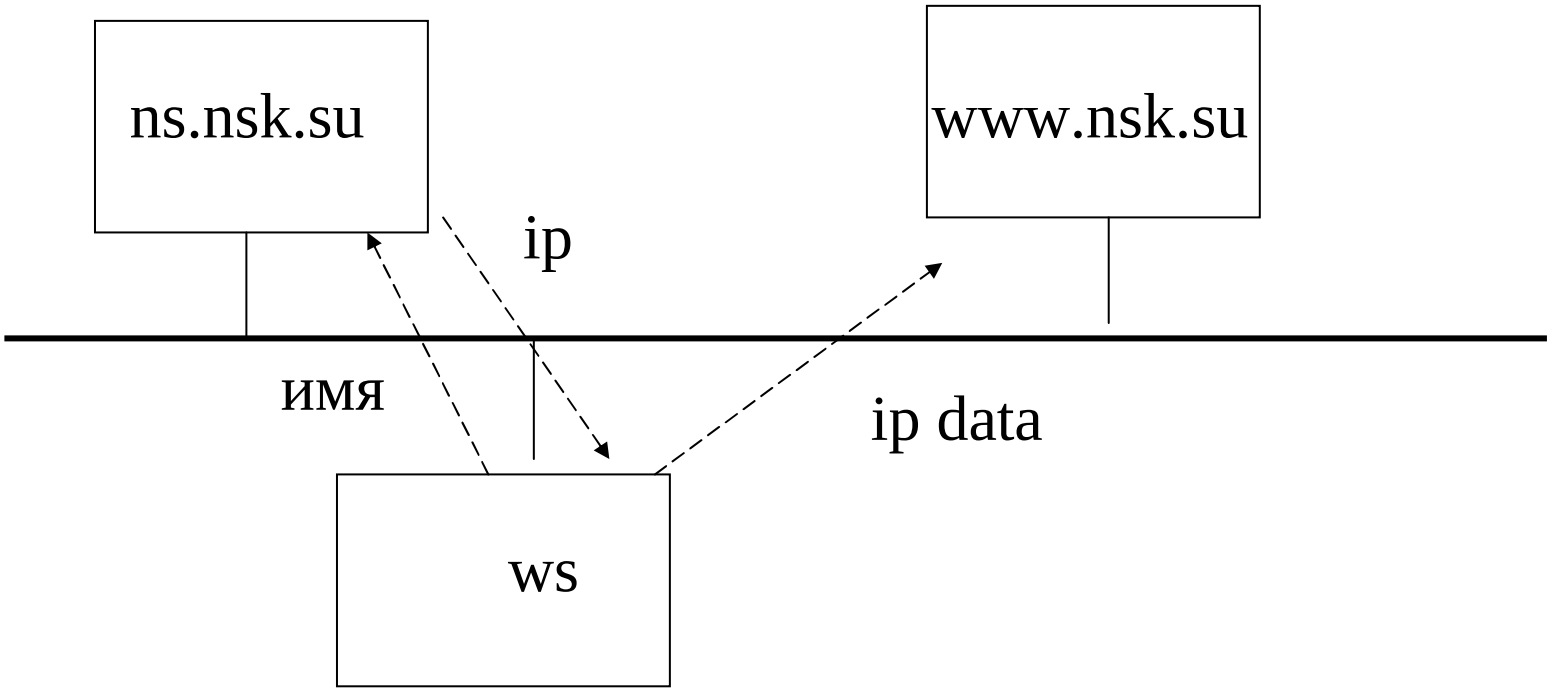
Авторитетный сервер – содержит информацию о всех именах зоны

Первичный сервер DNS – «главный сервер», который отслеживает все изменения в базе данных

Вторичный сервер DNS – ведомый сервер, получает таблицу от главного сервера.


0
15
16
31
Идентификация
Флаги
Число вопросов
Число ответных RR
Число RR в перечне полномочных серверов
Число дополнительных RR
12 байт
Вопросы
Ответы
(указанное число ответных RR)
Перечень полномочных серверов имен
(указанное число RR в перечне полномочных серверов)
Дополнительная информация

Пример обработки стандартного запроса



Утилиты

hostname – выдает или устанавливает имя компьютера

hostname имя

host – утилита DNS поиска, поддерживает инверсные запросы.

Dig @nameserver goal

Nslookup

Инверсные запросы

Служат для выяснения имени по ip-адресу. Организация, взявшая ответственность за определенную часть пространства имен несет ответственность также за соответствующее пространство в домене in-addr.arpa

Пример: пусть nao.edu -> 140.252

sun.nao.edu -> 149.252.13.33

33.13.252.140.in-addr.arpa

host 212.20.13.18 ->

18.13.20.212.in-addr.arpa

Настройка в linux DNS клиента

/etc/hosts – имена в обход DNS ()

127.0.0.1 localhost lo myhost

/etc/resolv.conf – перечисление серверов DNS

domain имя – указание своего домена

nameserver – указание сервера имен

Создаем собственную суб-доменную зону

- Вариант 1 (это не субдомен)
  - объявляем соответствие ip-адреса и доменного имени в /etc/hosts
- Вариант 2.

активизируем и настраиваем DNS-сервер (named).

  - настраиваем полнофункциональную субдоменную зону

Настройка BIND как кэширующий сервер

- Необходимо установить пакеты **bind**, **caching-nameserver**
- Избегайте запуска вашего DNS-сервера в качестве root (**/etc/rc.d/init.d/named** ->**daemon named** на **daemon named -u nobody -g nobody**).
- Версии BIND старшие 8.2.2 устанавливать крайне не рекомендуется

Что надо сделать в named.conf

- **directory (x.x.x.x ; y.y.y.y**);;
- **allow-query { 192.168.1/24; 127.0.0.1/32; };;**
-

Протоколы транспортного уровня

- TCP
- UDP

Протокол UDP

UDP – простой протокол транспортного уровня, обеспечивающий обмен отдельными пакетами. Каждая операция вывода данных для пересылки порождает единственный UDP пакет, отсылаемый в одном IP пакете.

Протокол UDP не обеспечивает надежного транспортного сервиса-он лишь оформляет порцию данных и отдает ее на уровень IP,который, как мы уже говорили не гарантирует доставку пакета по назначению.

Формат пакета UDP

IP заголовок 20 байт

Порт источника 16бит	Порт назначения 16бит
Длина UDP 16бит	Контрольная сумма 16бит
Данные	

Некоторые нюансы

Если а пункте назначения подсчитанная контрольная сумма не совпала- пакет «молча» отбрасывается

Механизм подсчета контрольной суммы необязателен и может быть выключен. Эта тактика может быть оправдана в локальном сегменте.

Если пакет оказался фрагментирован (больше MTU) и какая-либо его часть «потерялась», то все данные теряются

Вероятность ошибки

Замерялась в течении 40 суток, данные из источника см. сп. литературы, п.п 1

Протокол	Типичное число ошибок контрольной суммы	Примерное общее число пакетов
Ethernet	446	170 000 000
IP	14	170 000 000
UDP	5	170 000 000
TCP	350	30 000 000

Лекция

Лекция 7.

TCP-IP- транспортные протоколы (продолжение)

- Транспортные протоколы
- Понятие порта в сетевом приложении.
- Протокол TCP и его спецификация.
- Установление и окончание TCP-соединения.

Защита информации в сетях

- Фильтрация IP пакетов.

Понятие порта

Пакеты, поступающие на транспортный уровень, организуются операционной системой в виде множества очередей к точкам входа различных прикладных процессов. Такие точки входа называются портами. Порт однозначно определяет конкретное приложение.

Итак, в TCP/IP адрес соответствует конкретному сетевому интерфейсу, а порт – приложению,выполняющему какие-либо действия.

Назначение портов

Номера портов стандартизованы для всех операционных систем. В ОС Linux номера портов и сопоставляемые им службы перечислены файле /etc/protocols /etc/service

Централизованное присвоение сервисам номеров портов выполняется организацией *Internet Assigned Numbers Authority*. Эти номера затем закрепляются и опубликовываются в стандартах Internet.

Локальное присвоение номера порта заключается в том, что разработчик некоторого приложения просто связывает с ним любой доступный, произвольно выбранный числовой идентификатор, обращая внимание на то, чтобы он не входил в число зарезервированных номеров портов(не приветствуется но не запрещается).

Назначения портов

Соответствия портов службам перечислены в файле /etc/services

Пространство 0:1024 зарезервировано для стандартных сервисов. 1025:65535- для пользовательских сервисов.

Не рекомендуется при создании собственных сетевых приложений использовать зарезервированные порты.

Стандартные и пользовательские порты

- Стандартные порты- точки входа закрепленные за приложениями для ожидания входящих (как правило) соединений
- Пользовательские порты
  - Порты исходящих соединений инициаторов
  - Дополнительные порты при обмене данными

Протокол TCP

Соединение в протоколе TCP идентифицируется парой полных адресов обоих взаимодействующих процессов (оконечных точек). Адрес каждой из оконечных точек включает IP-адрес (номер сети и номер компьютера) и номер порта. Одна оконечная точка может участвовать в нескольких соединениях.

Надежность обеспечивается

- Разбиение потока данных на кадры
- Проверка квитанций
- Корректная, контролируемая «сборка» пакетов
- Проверка контрольной суммы
- Выбраковка дублированных пакетов
- Управление темпом передачи данных

Установление соединений

Установление соединения выполняется в следующей последовательности:

- Одна из сторон является инициатором. Она посылает запрос к на открытие порта для передачи (active open).
- После открытия порта протокол TCP на стороне процесса-инициатора посылает запрос процессу, с которым требуется установить соединение.
- Протокол TCP на приемной стороне открывает порт для приема данных (passive open) и возвращает квитанцию, подтверждающую прием запроса.
- Для того чтобы передача могла вестись в обе стороны, протокол на приемной стороне также открывает порт для передачи (active port) и также передает запрос к противоположной стороне.
- Сторона-инициатор открывает порт для приема и возвращает квитанцию. Соединение считается установленным.

Проверка корректности

В рамках соединения правильность передачи каждого сегмента должна подтверждаться квитанцией получателя. *Квитирование* - это один из традиционных методов обеспечения надежной связи.

Данные разбиваются на кадры, для каждого кадра отправитель ограниченное время ожидает получения положительной квитанции о доставке. Если положительная квитанция не получена, или получена отрицательная квитанция, кадр посылается повторно.

Формат пакета TCP

<table>
<tr><td colspan="10">IP Заголовок (20 байт)</td></tr>
<tr><td colspan="5">Порт передатчик (16 бит)</td><td colspan="5">Порт приемник (16 бит)</td></tr>
<tr><td colspan="10">Номер последовательности 32 бит</td></tr>
<tr><td colspan="10">Квитанция 32 бита</td></tr>
<tr><td>Длина заголовка 4 бит</td><td>Резерв (6 бит)</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td colspan="2">Размер окна приемника (16 бит)</td></tr>
<tr><td colspan="5">Контрольная сумма 16 бит</td><td colspan="5">Указатель границы срочных данных 16 бит</td></tr>
<tr><td colspan="10">Опции (если есть)</td></tr>
<tr><td colspan="10">Данные (если есть)</td></tr>
</table>

Флаги

SYN - синхронизация номера последовательности. Сообщает, что источник готовится передать новый поток данных.

URG - сообщает принимающему модулю TCP о том, что указатель на данные, требующие неотложной обработки, установлен.

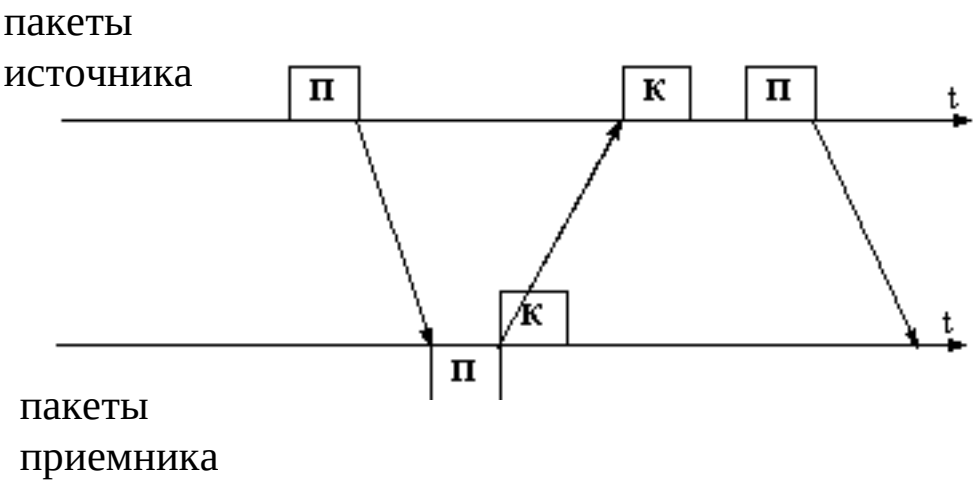
ACK – «флаг квитанции»

PSH - требование немедленно выслать принятый сегмент данных приложения получателю. Как правило, TCP буферизует данные, а затем передает приложению. Флаг PSH позволяет исключить задержки, например, при работе с Telnet для выдачи эха на экран.

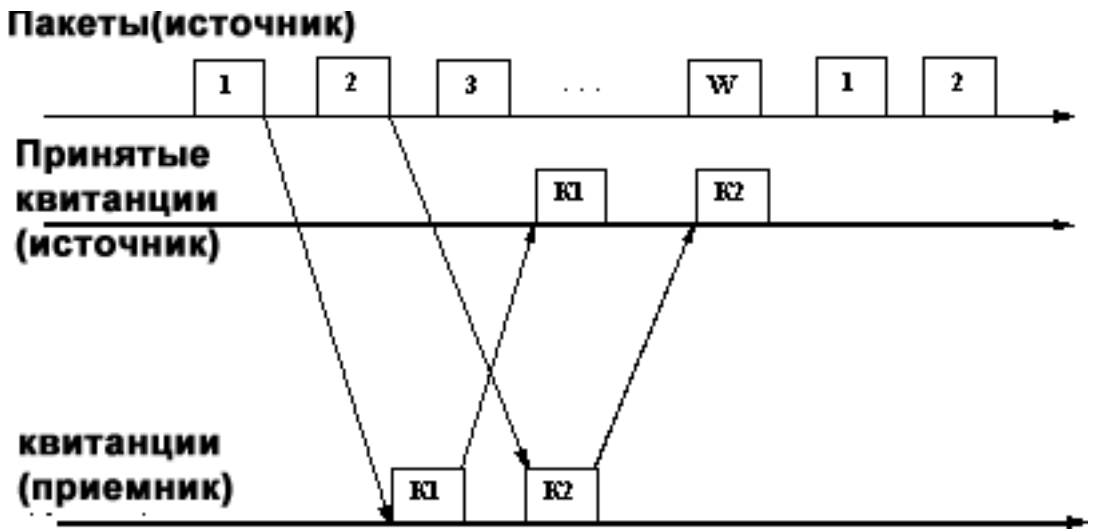
RST - запрос у принимающего модуля TCP на сброс соединения. Большинство приложений завершает работу при получении этого флага (сброс в сети, в оборудовании и т.д.)

FIN - источник закончил передавать данные. Чтобы полностью закончить соединение, принимающий компьютер должен послать также пакет с установленным FIN.

Обмен квитанциями с простоями



Обмен квитанциями по методу скользящего «окна»



Выбор времени ожидания

Выбор времени ожидания (тайм-аута) очередной квитанции является важной задачей, результат решения которой влияет на производительность протокола TCP.

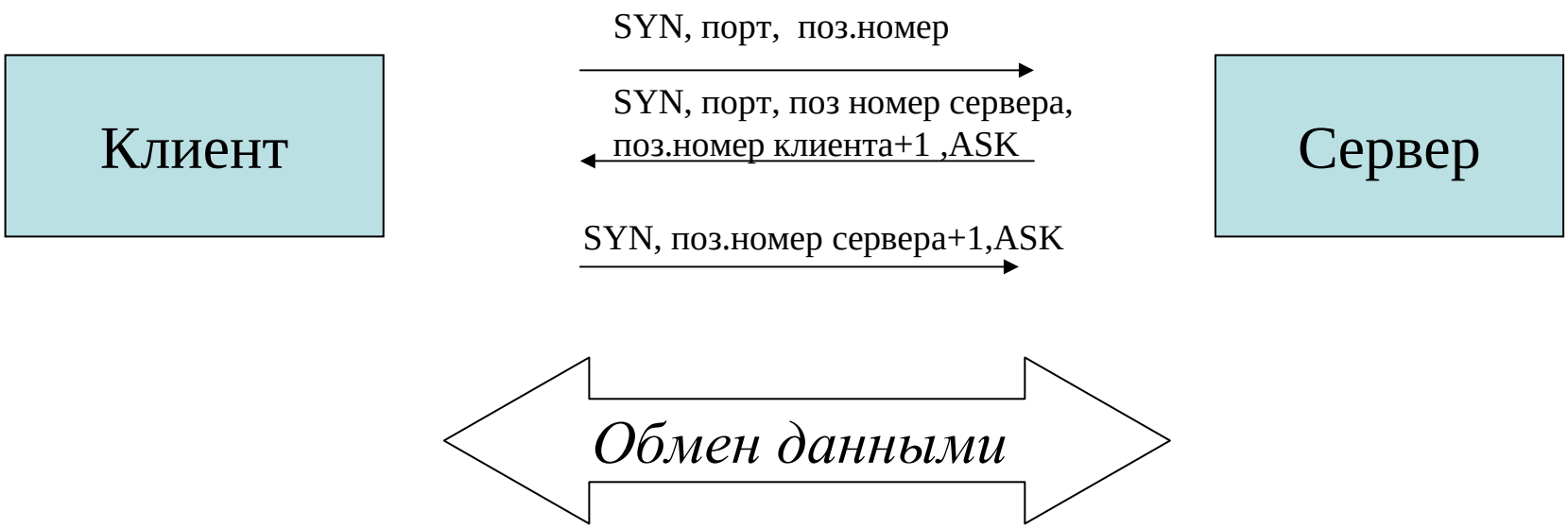
При каждой передаче засекается время от момента отправки сегмента до прихода квитанции о его приеме (время оборота). Получаемые значения времен оборота усредняются с весовыми коэффициентами, возрастающими от предыдущего замера к последующему. В качестве времени ожидания берется полученная величина, умноженная на 2.

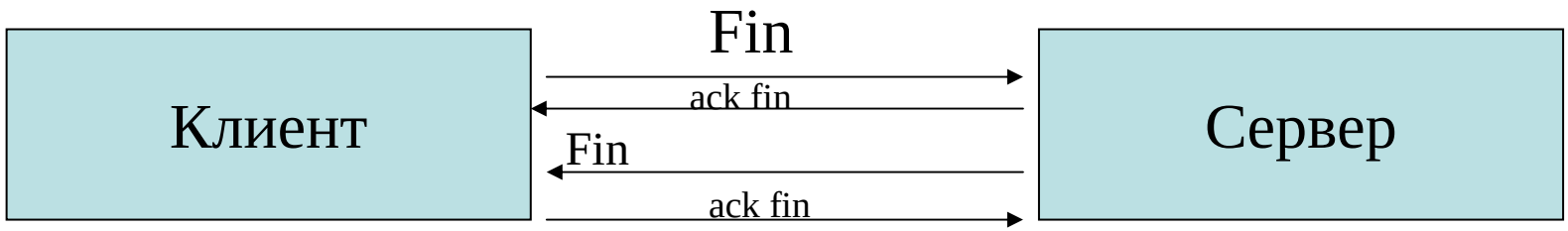
Реакция на перегрузку сети

Варьируя величину окна, можно повлиять на загрузку сети. Чем больше окно, тем большую порцию неподтвержденных данных можно послать в сеть. Если сеть не справляется с нагрузкой, то возникают очереди в промежуточных узлах-маршрутизаторах и в конечных узлах-компьютерах.

При переполнении приемного буфера конечного узла "перегруженный" протокол TCP, отправляя квитанцию, помещает в нее новый, уменьшенный размер окна. Если он совсем отказывается от приема, то в квитанции указывается окно нулевого размера. Однако даже после этого приложение может послать сообщение на отказавшийся от приема порт. Для этого, сообщение должно сопровождаться пометкой "срочно" (бит URG в запросе установлен в 1). В такой ситуации порт обязан принять сегмент, даже если для этого придется вытеснить из буфера уже находящиеся там данные

Установка/завершение соединения





Обобщение

- TCP – надежность, но большая сложность и накладные расходы. Опасность SynFlood. Появление Zombi sockets.
- UDP – простота, возможность широковещания, нет гарантии доставки и управления темпом передачи данных

Другие протоколы, основанные на IP

Мы рассмотрели 4 протокола, транспортного уровня, являющихся стандартом в Internet:

udp,tcp,icmp,igmp.

Другие протоколы, использующие IP соединения перечислены в

/etc/protocols

Обобщение

Протоколы семейства TCP/IP являются надежным средством обмена данными в сетях. В настоящее время TCP/IP – де-факто стандарт обмена данными во всемирной Сети Интернет.

Итак, лидирующая роль стека TCP/IP объясняется следующими его свойствами:

- Это наиболее завершенный стандартный и в то же время популярный стек сетевых протоколов, имеющий многолетнюю историю.
- Почти все большие сети передают основную часть своего трафика с помощью протокола TCP/IP.
- Это метод получения доступа к сети Internet.
- Этот стек служит основой для создания intranet- корпоративной сети, использующей транспортные услуги Internet и гипертекстовую технологию WWW, разработанную в Internet.
- Все современные операционные системы поддерживают стек TCP/IP.
- Это гибкая технология для соединения разнородных систем как на уровне транспортных подсистем, так и на уровне прикладных сервисов.
- Это устойчивая масштабируемая межплатформенная среда для приложений клиент-сервер.

Rfc1180 перевод

http://lib.ru/TCPBOOK/tcp1.txt

- **RFC1180 Семейство протоколов TCP/IP в переводе Брежнева и Смелянского**
- Оригинал - RFC1180 T.Socolofsky and C.Kale http://www.alcpress.com/rfc/tcpip/rfc1180.htm Перевод с английского: Брежнев А.Ф., Смелянский Р.Л.

Защита информации в сетях

- Защита от несанкционированного доступа методом отказа в доступе(свой/чужой)
  - Защита паролями и ключами доступа
  - Межсетевые экраны (firewall)
- Защита методом криптования
  - Шифрование с открытым ключом
  - Защищенные службы и приложения (ssl,ssh,pgp)

Межсетевые экраны

Предназначены для ограничения доступа посторонних к сервисам(портам). Как известно, в наше время компьютеры все чаще и чаще подвергаются сетевым атакам. Излюбленная цель хакеров - компьютеры под управлением ОС Windows, которые сравнительно несложно заразить вирусом или трояном. Однако пользователи юникс-систем также не всегда остаются в стороне. Причина большинства взломов компьютеров с установленной ОС Linux - неправильно настроенные сервисы или использование устаревших версий ПО.

Литература

- [1] Руководство по Iptables Iptables Tutorial 1.1.14 http://www.linuxsecurity.com/resource_files/firewalls/IPTables- Tutorial/iptables-tutorial.html
- [2]http://www.opennet.ru/docs/RUS/iptables/ руководство 1.1.19

Firewall

- **Программный firewall**

Программные межсетевые экраны работают на базе традиционных операционных систем, которые сами имеют слабые места, постоянно изучаемые и атакуемые хакерами. Для настройки программного межсетевого экрана требуется опытный системный администратор.

- **Аппаратный firewall**

Специальное устройство созданное для зашиты сети.

Аппаратные межсетевые экраны почти всегда построены с использованием операционных систем, специально разработанных (либо модифицированных) для этой цели. Аппаратные межсетевые экраны легки в настройке и обслуживании.

Межсетевые экраны

Во все ядра Linux, начиная с 2.0, встроено средство для фильтрации сетевых пакетов. В 2.0 это ipfwadm, в 2.2 - ipchains, а в 2.4 и 2.6 - iptables. Мы будем рассматривать последний вариант, как наиболее современный.

Принцип фильтрации: когда через ядро проходит пакет, он проверяется на совпадение с одним или несколькими правилами. При этом в зависимости от этих правил он может быть пропущен (ACCEPT), отброшен (DROP) или отклонен (REJECT). Кроме того, он может быть отправлен на проверку в следующую цепочку правил. Здесь же можно указать, что факт прохождения пакета, подходящего под определенное правило, должен быть отмечен в syslog. Правила могут включать в себя проверку адреса/порта источника/получателя, протокола, флагов TCP.

Определения

"Поток" (Stream) - под этим термином подразумевается соединение, через которое передаются и принимаются пакеты (как минимум 2 пакета).

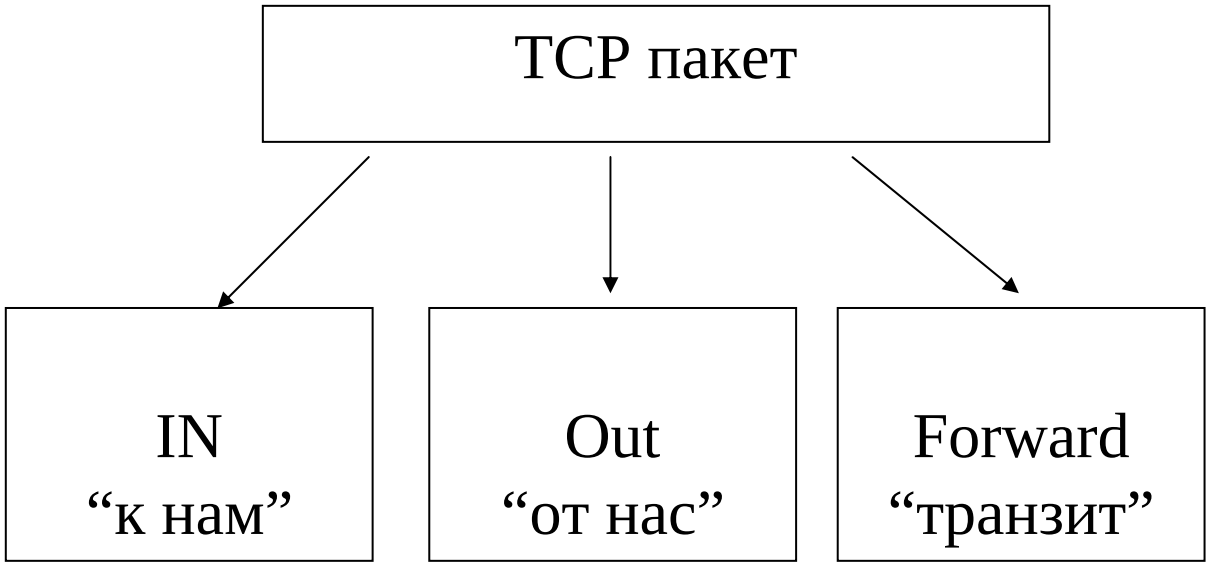
Пользовательское пространство" (User space) - под этим термином подразумевается все, что расположено за пределами ядра.

Пространство ядра (Kernel space) - в большей или меньшей степени является утверждением, обратным термину "Пользовательское пространство". Подразумевает место исполнения - в пределах ядра.

Таблица обобщает действия, которые будут осуществлены с пакетом

Цепочка – набор проверок, которым подвергается пакет в рамках обработки таблицы. По результатам проверки- пакет может перейти в другую таблицу.

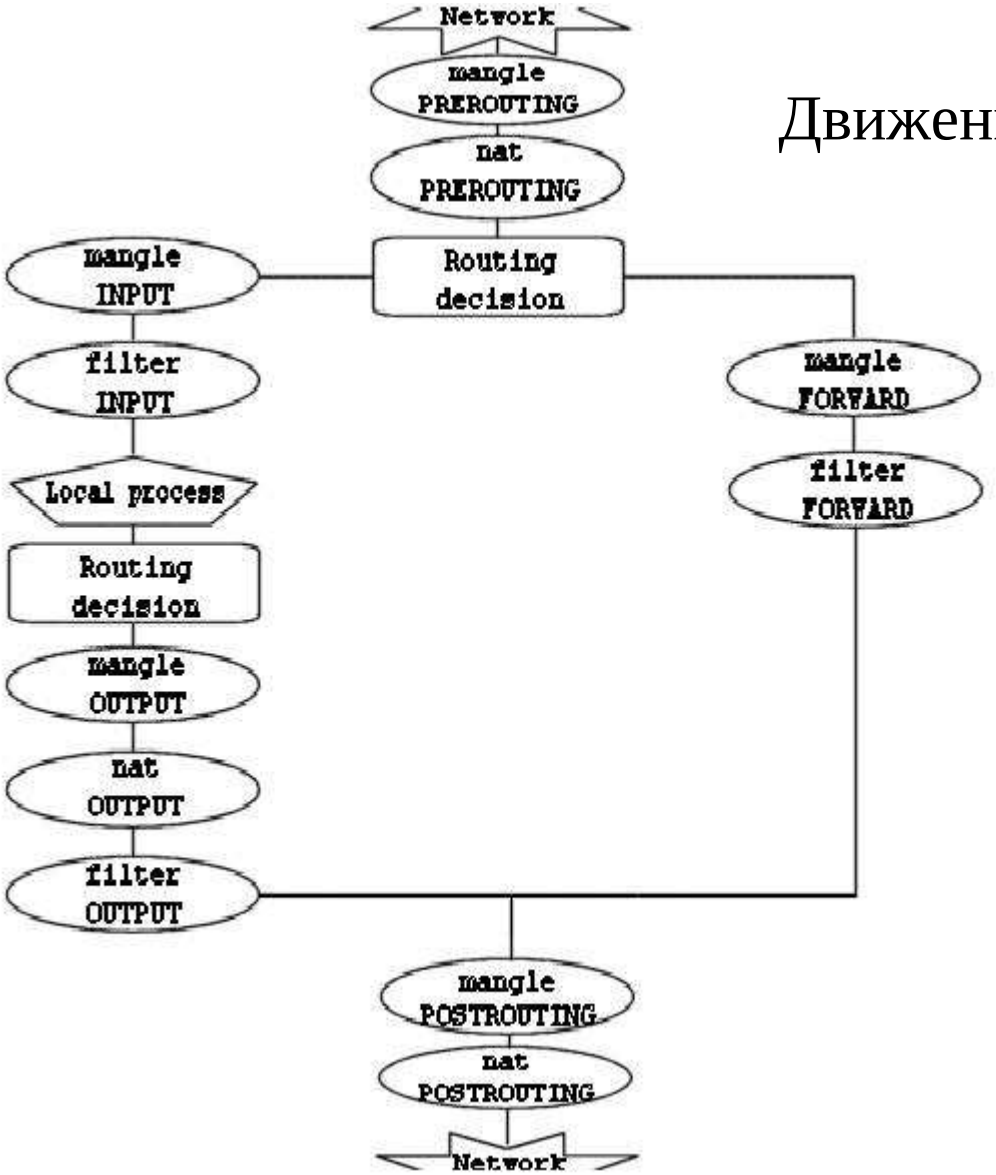
Типы пакетов



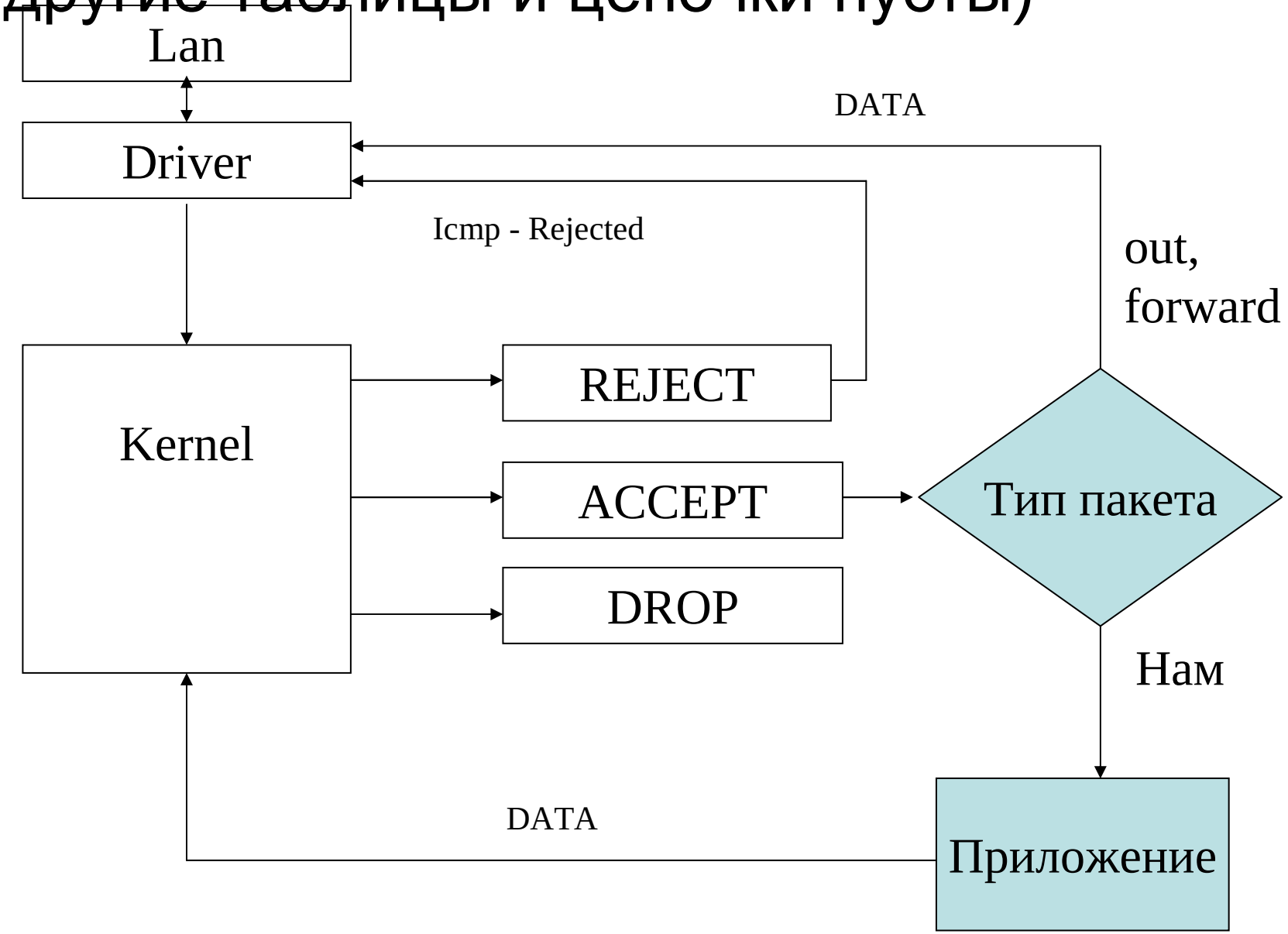
Типы таблиц

Таблица	Назначение	цепочки
nat	Network Adderss Translation	PREROUTING POSTROUTING
mangle	Изменяет заголовки пакетов	1. PREROUTING, POSTROUTING 2.INPUT,OUTPUT, FORWARD
filter	Фильтрация пакетов	INPUT,OUTPUT, FORWARD

Движение пакетов [2]



Принцип фильтрации,таблица filter (если другие таблицы и цепочки пусты)

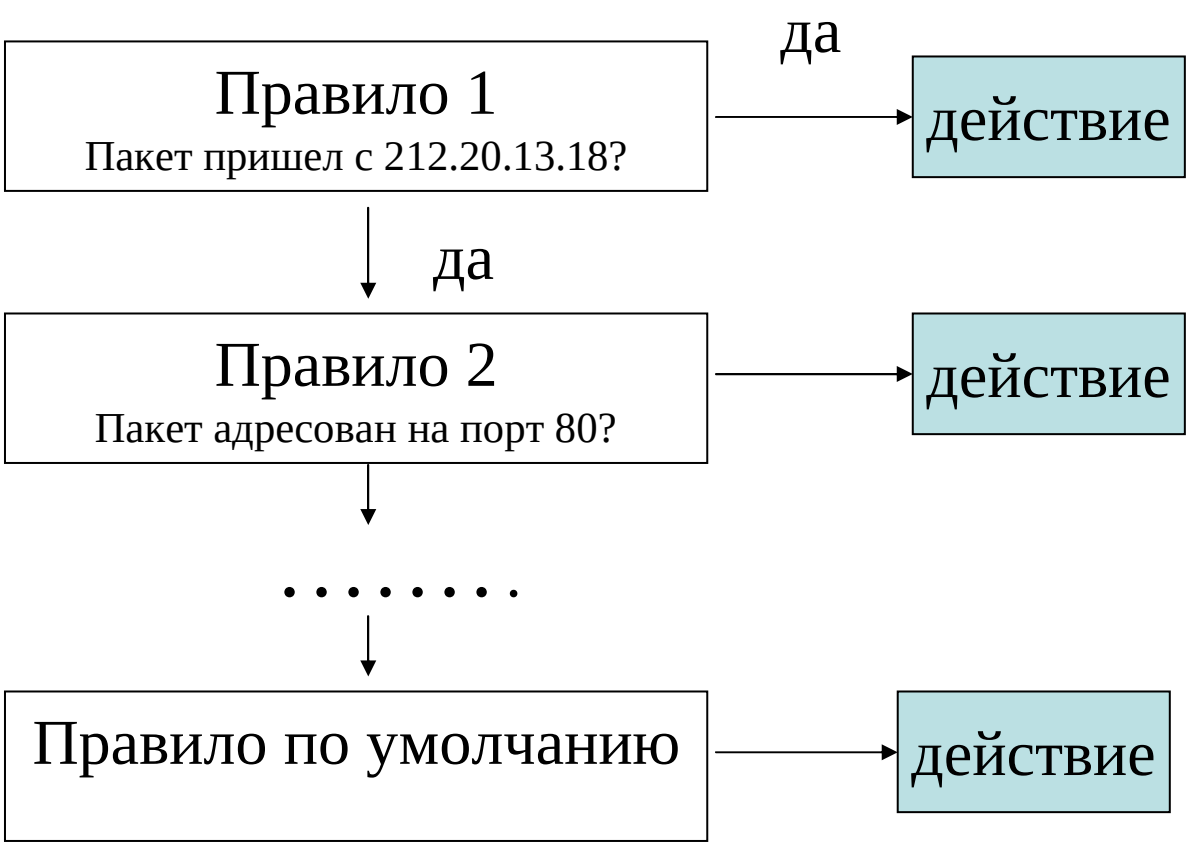


iptables

iptables [-t table] command [match] [target/jump]

- Каждая строка, которую вы вставляете в ту или иную цепочку, должна содержать отдельное правило.
- Нигде не утверждается, что описание действия (target/jump) должно стоять последним в строке, мы, однако, будем придерживаться именно такой нотации для удобочитаемости.
- то по умолчанию предполагается использование таблицы filter , иначе требуется указать имя таблицы [-t]
- за именем таблицы, должна стоять команда. Если спецификатора таблицы нет, то команда всегда должна стоять первой. Команда определяет действие iptables, например: вставить правило, или добавить правило в конец цепочки, или удалить правило и т.п.
- match задает критерии проверки.
- target указывает действие,которое должно быть выполнено при условии выполнения критериев в правиле

Цепочка


да
Правило 1
Пакет пришел с 212.20.13.18?
действие
да
Правило 2
Пакет адресован на порт 80?
действие
. . . . . .
Правило по умолчанию
действие

Iptables, команды

iptables [-t table] **command** [match] [target/jump]

команда	пример	действие
-L, **--list**	**iptables -L INPUT**	Вывод всех правил
-A, **--append**	**iptables -A INPUT**	Добавляет новое правило в цепочку
-D, **--delete**	**iptables -D INPUT --dport 80 -j DROP;** **iptables -D INPUT 1**	Удаление правила из цепочки
-R, **--replace**	**iptables -R INPUT 1 -s 192.168.0.1 -j DROP**	заменяет одно правило другим

Iptables, команды

iptables [-t table] **command** [match] [target/jump

команда	пример	действие
-I, **--insert**	**iptables -I INPUT 1 --dport 80 -j ACCEPT**	Вставляет новое правило в цепочку. Номер-перед чем вставлять
-F, **--flush**	**iptables -F INPUT**	Сброс правил (если без указания цепочки - действует на все)
-Z, **--zero**	**iptables -Z INPUT**	Обнуление всех счетчиков в заданной цепочке
-N, **--new-chain**	**iptables -N allowed**	Создается новая цепочка с заданным именем в заданной таблице

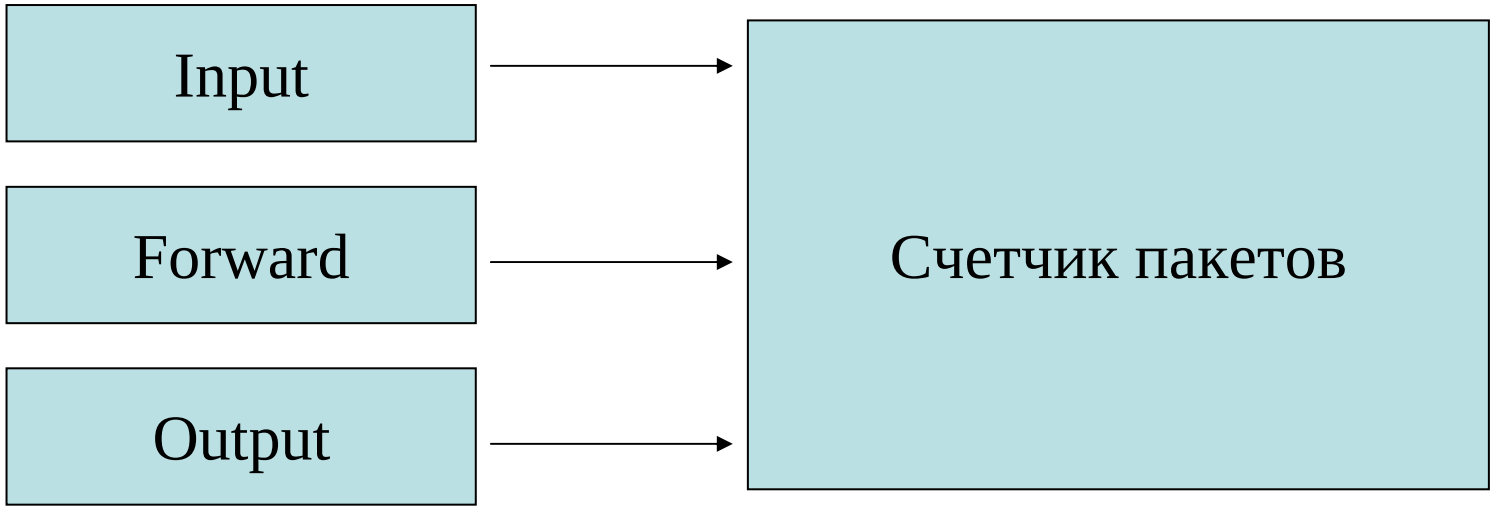
Iptables, команды

iptables [-t table] **command** [match] [target/jump

команда	пример	действие
-P, **--policy**	**iptables -P INPUT DROP**	Определяет политику по умолчанию для заданной цепочки. **DROP**, **ACCEPT** и **REJECT**.
-E, **--rename-chain**	**iptables -E allowed disallowed**	выполняет переименование пользовательской цепочки

Пример: создание счетчиков

Задача: подсчитывать все проходящие пакеты через ядро.



Цепочка подсчета пакетов

```
iptables -F
iptables -X COUNT
#
iptables -N COUNT
#
iptables -I OUTPUT -j COUNT
iptables -I FORWARD -j COUNT
iptables -I INPUT -j COUNT
#
iptables -P OUTPUT ACCEPT
#
iptables -I COUNT -j RETURN
#
iptables -L -v -x
```

Iptables ключи

Ключ	к команде	действие
--v, **--verbose**	**--list, --append, --insert, --delete, --replace**	Повышает информативность вывода
-x, **--exact**	**--list**	Выводятся точные значения (без округления)
-n, **--numeric**	**--list**	Запрещает DNS разбор
--line-numbers	--list	Выводит line numbers

Критерии

Критерии- это условие, в результате проверки которого будет выполнено какое-либо действие.

Рассмотрим 5 групп критериев:

- общие
- TCP
- UDP
- ICMP
- Специальные

Критерии

iptables [-t table] command **[match]** [target/jump]

Критерий	**-p**, **--protocol**
Пример	**iptables -A INPUT -p tcp**
Описание	Указывает тип протокола. tcp,udp,icmp,all. Логическая инверсия ‘!’
Критерий	**-s**, **--src**, **--source**
Пример	**iptables -A INPUT -s !192.168.1.1/24**
Описание	IP-адрес(а) источника пакета.
Критерий	**-d**, **--dst**, **--destination**
Пример	**iptables -A INPUT -d 192.168.1.1**
Описание	IP-адрес(а) получателя.

Критерии

iptables [-t table] command **[match]** [target/jump]

Правило распространяется на все фрагменты фрагментированного пакета, кроме первого.

Усложняем счетчик

```
…
#
iptables -I COUNT -s 192.168.6.2/32  -j RETURN
iptables -I COUNT –i lo;
iptables -I COUNT -p tcp -j
iptables -I COUNT -p udp -j
iptables -I COUNT -p icmp –j
iptables -I COUNT –j RETURN
#
iptables -L -v -x
```

TCP ОПЦИИ

Критерий	--sport, --source-port
Пример	iptables -A INPUT -p tcp --sport 22
Описание	Исходный порт, с которого был отправлен пакет. В качестве параметра может указываться номер порта или название сетевой службы, а также интервал: 22:80
Критерий	--dport, --destination-port
Описание	Порт назначения
Критерий	iptables -p tcp --syn
Описание	Блокирует попытки соединений.
Критерий	--tcp-option
Пример	iptables -p tcp --tcp-option 16
Описание	Опции пакета

UDP,icmp ОПЦИИ

Критерий	--sport, --source-port
Пример	iptables -A INPUT -p udp --sport 53
Описание	Исходный порт, с которого был отправлен пакет. В качестве параметра может указываться номер порта или название сетевой службы, а также интервал: 22: Инверсия разрешена.
Критерий	--dport, --destination-port
Описание	Порт назначения

Критерий	--icmp-type
Пример	iptables -A INPUT -p icmp --icmp-type 8
Описание	Тип сообщения ICMP Тип сообщения ICMP определяется номером или именем. Числовые значения определяются в RFC 792. Чтобы получить список имен ICMP значений выполните команду iptables --protocol icmp --help.

Усложняем счетчик

```
…
#
iptables -A COUNT -s 192.168.6.2/32 -p tcp --sport :1024
iptables -A COUNT -s 0/0 -p tcp --sport 80
iptables -A COUNT -s 0/0 -p tcp --dport 21
iptables -A COUNT -s 0/0 -p udp --dport domain
iptables -L -v -x
```

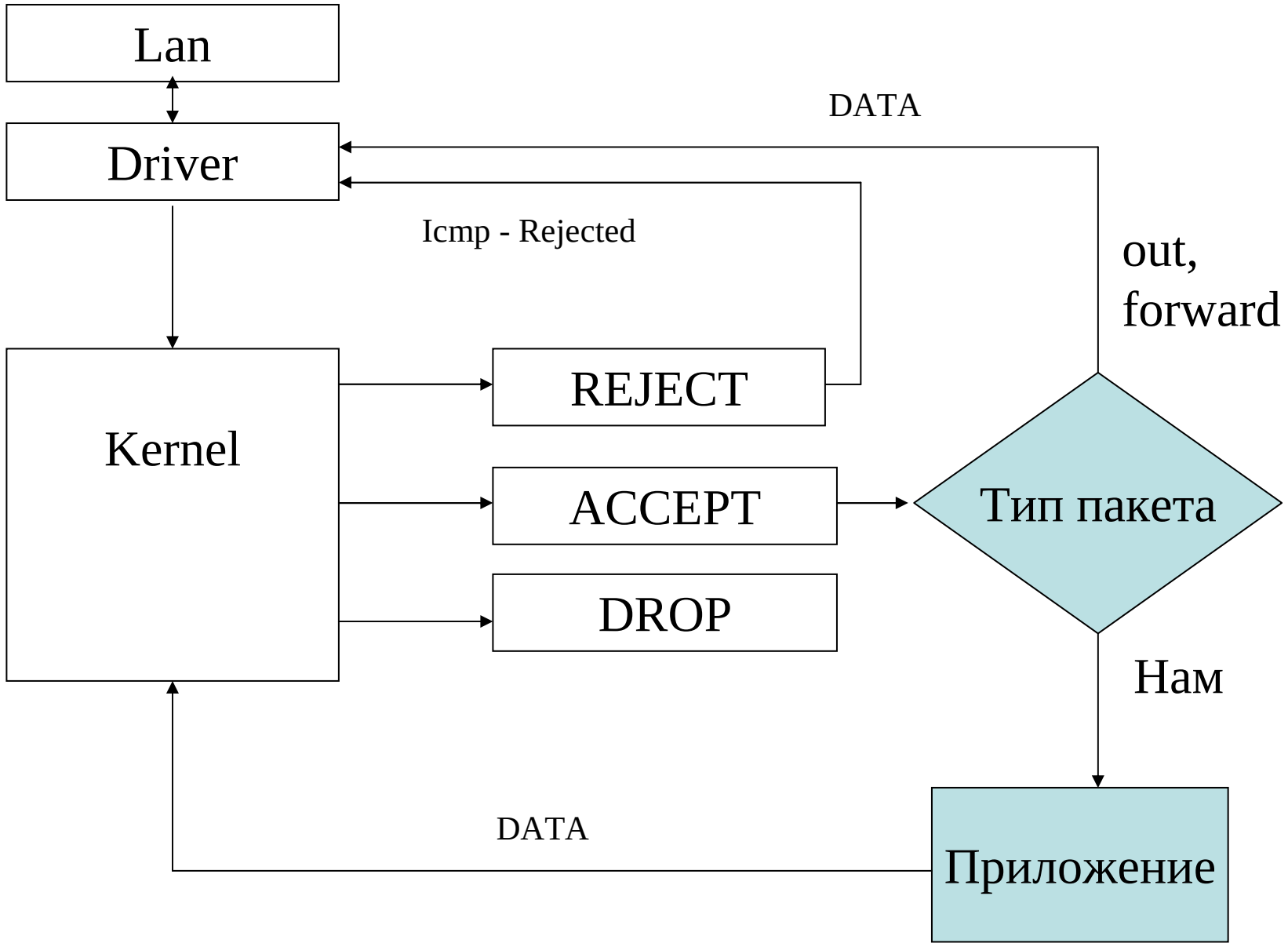
Лекция

Лекция 8. -9

Защита информации в сетях

- Фильтрация IP пакетов (продолжение)
- Управление трафиком
- Типовое решение построения фильтра пакетов для среднего предприятия.
- Введение в криптографию.Алгоритмы криптования с открытым ключом и их использование.

Принцип фильтрации,таблица filter



iptables

iptables [-t table] command [match] **[target/jump]**

- Каждая строка, которую вы вставляете в ту или иную цепочку, должна содержать отдельное правило.
- Нигде не утверждается, что описание действия (target/jump) должно стоять последним в строке, мы, однако, будем придерживаться именно такой нотации для удобочитаемости.
- то по умолчанию предполагается использование таблицы filter , иначе требуется указать имя таблицы [-t]
- за именем таблицы, должна стоять команда. Если спецификатора таблицы нет, то команда всегда должна стоять первой. Команда определяет действие iptables, например: вставить правило, или добавить правило в конец цепочки, или удалить правило и т.п.
- match задает критерии проверки.
- **target указывает действие,которое должно быть выполнено при условии выполнения критериев в правиле**

Действия

iptables [-t table] command [match] **[target/jump]**

- ACCEPT – пакет прекращает движение по данной цепочке и может быть принят. Тем не менее, он продолжает движение по другим таблицам и может быть отвергнут там. –j ACCEPT

  iptables -A INPUT –i eth0 –j ACCEPT
- DROP - пакет прекращает движение по всем таблицам и сбрасывается. Система «забывает о нем» -j DROP

  iptables –A INPUT –s 212.20.14.0/24 –j DROP
- QUEUE – пакет передается на обработку пользовательскому процессу –j QUEUE

  # modprobe iptable_filter

  # modprobe ip_queue

  # iptables -A OUTPUT -p icmp -j QUEUE
- RETURN - возврат из текущей в вызывающую цепочку

Действие REJECT

REJECT – пакет прекращает движение по таблицам, владельцу высылается ICMP сообщение об ошибке. Работает в цепочках INPUT, FORWARD, OUTPUPUT.

Опция --reject-with [ключ] задает тип ICMP сообщения:

*icmp-net-unreachable, icmp-host-unreachable, i**cmp-port-unreachable**, icmp-proto-unreachable, icmp-net-prohibited u icmp-host-prohibited*

iptables -A FORWARD -p TCP --dport 22 -j REJECT - -reject-with icmp-net-unreachable

Действие REDIRECT

Перенаправляет пакет на другой порт той же самой машины. Может использоваться только в цепочках PREROUTING и OUTPUT таблицы nat.

Пример «прозрачного» проксирования (прокси на порту 8080):

iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 8080

iptables -t nat -A PREROUTING -p tcp --dport 21 -j REDIRECT --to-ports 8080

iptables -t nat -A PREROUTING -p tcp --dport 20 -j REDIRECT --to-ports 8080

Действие LOG

служит для журналирования в syslog отдельных пакетов и событий. syslog – системный журнал событий, который ведет демон syslogd. (/etc/syslogd.conf). Используется для отладки фильтра.

iptables -A FORWARD -p tcp -j LOG –ключ

Ключ:

--log-level устанавливает подробность информации

--log-prefix “строка” префикс-комментарий

--log-tcp-options подробно о tcp заголовке

--log-ip-options подробно об ip заголовке

Действие MASQUERADE,SNAT

Выполняет подстановку IP адресов.
Допускается указывать только в цепочке POSTROUTING таблицы nat

iptables -t nat -A POSTROUTING -p TCP -j MASQUERADE --to-ports 1024-31000

iptables -t nat -A POSTROUTING --dst $HTTP_IP --dport 80 -j SNAT --to-source $LAN_IP

Задача

- Обеспечить возможность компьютерам с «локальными» IP адресами обмениваться почтой с «бесплатными почтовыми сервисами»

Решение

iptables -A FORWARD -s 192.168.6.70 -j ACCEPT

iptables -A FORWARD -d 192.168.6.70 -j ACCEPT

iptables -t nat -A POSTROUTING -o eth0 -p tcp -s 192.168.6.70 --dport 25 -j MASQUERADE

iptables -t nat -A POSTROUTING -o eth0 -p tcp -s 192.168.6.70 --dport 110 -j MASQUERADE

Управление трафиком

Достаточно часто возникает необходимость управления трафиком на граничных шлюзах, серверах или конечных станциях с использованием пороговых значений скорости.

Условие, позволяющие ограничить частоту тех или иных событий, было реализовано самым первым в модуле limit и предназначалось прежде всего для ограничения частоты записи о событиях в журнальные файл системы.

limit

- **Соответствие limit**

-m limit позволяет задать пороговое значение частоты выполнения условий, по достижении которого выполняется заданная правилом операция

- может использоваться с параметрами
- **--limit <avg>** **средняя частота событий** 5 /n /s /h /d
- **--limit-burst <burst> пик “разовой” доставки пакетов**

Механизм

Модуль работает следующим образом:

условие считается выполненным, пока значение счетчика пакетов не превысит пика **limit-burst** ;

каждый пакет, соответствующий правилу, увеличивает значение счетчика на 1;

по истечении каждого интервала 1/ **limit** значение счетчика уменьшается на 1.

Разъяснение

Аналог – бассейн, из которого через трубу выливается вода.

Параметр **limit-burst** задает объем бассейна (количество помещающихся в него пакетов), а параметр **limit** определяет скорость оттока через выходную трубу.

Пока в бассейне есть место, правило выполняется, а как только бассейн наполнится до краев, пакеты перестанут соответствовать правилу (польется через край). Очевидно, что в пустой бассейн может сразу поместиться **limit-burst** пакетов, а за счет вытекания через трубу число пакетов в бассейне уменьшается на **limit** в единицу времени. Следовательно за это время в бассейн можно поместить до **limit** новых пакетов. Если пакеты не приходят, бассейн постепенно опустошается и может принять в себя больший объем.

Умолчания:

- Limit 3 пакета в час
- **--limit-burst** 5 пакетов.

iptables -A FORWARD -m limit -j LOG

- 1е 5 пакетов
- 20мин – ничего не пишем
- 1 пакет в 20 минут

Защита от атак:

Защита от атак SYN-flood:

- **iptables -A FORWARD -p tcp --syn -m limit --limit 1/s -j ACCEPT**

Защита от хитроумных сканеров портов:

- **iptables -A FORWARD -p tcp --tcp-flags SYN,ACK,FIN,RST RST -m limit --limit 1/s -j ACCEPT**

Защита от Ping of death:

- **iptables -A FORWARD -p icmp --icmp-type echo-request -m limit --limit 1/s -j ACCEPT**

Ограничение обмена данными:

iptables -A FORWARD -s 192.168.6.70 -j ***-m limit --limit 1/s --limit-burst*** *10 ACCEPT*

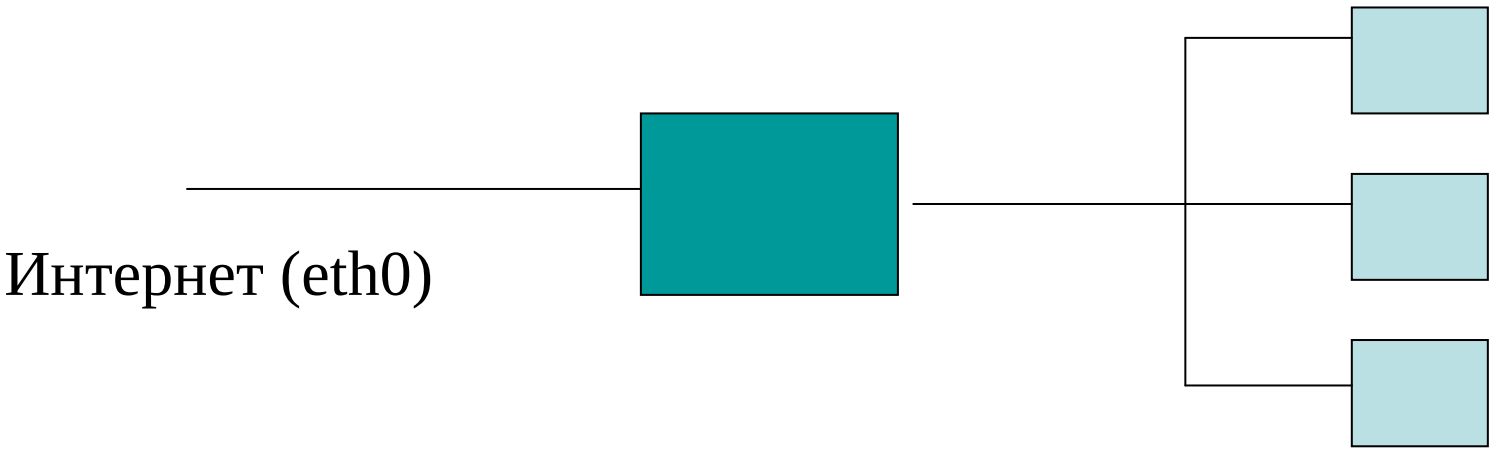
Ограничили трафик – 10 пакетов секунду

iptables -A FORWARD -s 192.168.6.70 -j ***-m limit --limit 1/h --limit-burst*** *60 ACCEPT*

Разрешили 1 раз в час сливать 90 кб почты

Пример rc.firewall

Имеется сеть:



- Http,ftp,domain,icmp,smb,pop3,smtp,proxy

```
#!/bin/sh

#Все сбросили
iptables -F
iptables -X COUNT

#Правила по умолчаниию
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# разрешаем доступ к внутренней петле
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
#считаем все входящие пакеты ($$$)
#Создаем цепочку счетчиков
iptables -N COUNT
iptables -A INPUT -i eth0 -j COUNT
```

```
# Разрешаем ходить пакетам icmp
iptables -A INPUT -p ICMP -j ACCEPT
iptables -A OUTPUT -p ICMP -j ACCEPT

#Разрешаем соединения "запрошенные" с нашего компьютера. Избавляет от некоторых
      видов атак
iptables -A INPUT -p TCP -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p UDP -m state --state ESTABLISHED,RELATED -j ACCEPT
#Разрешаем пользовательские порты
iptables -A OUTPUT -p TCP --sport 32768:65535 -j ACCEPT
iptables -A OUTPUT -p UDP --sport 32768:65535 -j ACCEPT

#Разрешаем порты SMB для доступа Windows-машин к нашему серверу
# Smb Данныые
iptables -A INPUT -i eth1 -p TCP --dport 137:139 -j ACCEPT
iptables -A INPUT -i eth1 -p UDP --dport 137:139 -j ACCEPT
iptables -A OUTPUT -o eth1 -p TCP --sport 137:139 -j ACCEPT
iptables -A OUTPUT -o eth1 -p UDP --sport 137:139 -j ACCEPT
#Smb служба имен
iptables -A INPUT -i eth1 -p UDP --sport 137 --dport 32768:65535 -j ACCEPT

#Разрешаем видеть нас как ВЕБ-сервер
iptables -A INPUT  -p TCP --dport 80 -j ACCEPT
iptables -A OUTPUT -p TCP --sport 80 -j ACCEPT
```

```
#Разрешаем видеть нас и как FTP -сервер
iptables -A INPUT  -p TCP --dport 20:21 -j ACCEPT
iptables -A OUTPUT -p TCP --sport 20:21 -j ACCEPT

#Разрешаем изнутри забирать почту по протоколу POP3
#iptables -A INPUT -i eth1 -p TCP --dport 110 -j ACCEPT
iptables -A OUTPUT -o eth1 -p TCP --sport 110 -j ACCEPT

#Разрешаем обмен почтой по протоколу smtp
iptables -A INPUT -p TCP --dport smtp -j ACCEPT
iptables -A OUTPUT -p TCP --sport smtp -j ACCEPT

#Разрешаем DNS обмен
iptables -A INPUT -p UDP --dport domain -j ACCEPT
iptables -A OUTPUT -p UDP --sport domain -j ACCEPT

#Разрешаем доступ изнутри к прокси на порт 8080
iptables -A INPUT -i eth1 -p TCP --dport 8080 -j ACCEPT
iptables -A INPUT -i eth1 -p UDP --dport 8080 -j ACCEPT
iptables -A OUTPUT -o eth1 -p TCP --sport 8080 -j ACCEPT
iptables -A OUTPUT -o eth1 -p UDP --sport 8080 -j ACCEPT
#---------------------------------
#Секция счетчиков
iptables -A COUNT   # Сколько всего пакетов
iptables -A COUNT -p icmp # Сколько ICMP пакетов
iptables -A COUNT -p tcp --dport smtp # Сколько напринимали спама по почте
iptables -A COUNT -j RETURN
#Показываем что получилось (затрет ошибки!!!)
iptables -L -v -x
```

Правила безопасности

- rc.firewall должен запускаться ДО активации сетевых интерфейсов
- первыми командами необходимо обязательно очищать старую таблицу устанавливать правила по умолчанию
- Правила по умолчанию Accept нельзя назначать цепочкам input,forward на внешнем интерфейсе
- Не открывайте без необходимости наружу “лишние” сервисы

Литература

- Руководство по Iptables: Iptables Tutorial 1.1.14 http://www.linuxsecurity.com/resource_files/firewalls/IPTables- Tutorial/iptables-tutorial.html

Лекция

Лекция 9-10.

Защита информации при информационном обмене

- Введение в криптографию.Алгоритмы криптографии с открытым ключом и их использование.
- Ключи, электронные сертификаты
- Шифрование электронной корреспонденции
- Защита беспроводных сетей.

Алгоритмы шифрования/применение

ВВЕДЕНИЕ В КРИПТОГРАФИЮ

Б.Шнайер. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си.-М.:Издательство ТРИУМФ, 2002 - 816с.:ил.)

А. Салома Криптография с открытым ключом: М.Мир 1995- 318с.

Применение криптографии:

- Защита информации:
  - Конфиденциальность данных
  - Электронная торговля и банковские операции
  - Обеспечение интеллектуальной собственности (ключи к программным продуктам)
  - Системы контроля доступа (электронные замки)

Основные результаты использования в ИС:

- Шифрование
- Проверка подлинности
- Проверка целостности
- Штамп времени
- Неотрицание авторства

Определения

- **Пароль (password)** - строка символов, уникальная для пользователя, используемая для получения доступа в систему или к информации в системе. Легко может быть воспроизведен третьими лицами
- **Ключ** – уникальный код, который создается на основе пароля и другой дополнительной информации. Создается пользователем. «Подделка» третьими лицами затруднительна.
- **Сертификат – блок информации, идентифицирующий владельца, выданный и подписанный Certificate authority**
  - **имя человека/организации, выпускающей сертификат;**
  - **Субъект сертификата (для кого был выпущен данный сертификат);**
  - **публичный ключ субъекта;**
  - **некоторые временные параметры (срок действия сертификата и т.п.).**

Определения

- **Криптография** - это наука о том, как обеспечить секретность сообщения.
- **Криптоанализ** - это наука о том, как **вскрыть** шифрованное сообщение, то есть как извлечь открытый текст не зная ключа.

Определения

- **Зашифровать информацию** – преобразовать его при помощи ключа (пароля) так, чтобы максимально затруднить ее чтение третьим лицами.
- **Создать электронную подпись** – добавить к информации«служебный» блок, позволяющий отследить изменения, определить авторство и время создания.

Алгоритмы шифрования/классификация

- Симметричные алгоритмы
  - потоковые алгоритмы
  - блочные алгоритмы (64,128 бит)
- Алгоритмы с открытым ключом

Правило безопасности шифрования: стоимость данных ниже стоимости взлома алгоритма.

Определения

Безусловно безопасный- независимо от объемов шифротекста информации для вскрытия шифра недостаточно.

Вычислительно безопасный алгоритм – не может быть вскрыт с использованием доступных ресурсов сейчас или в обозримом будущем.

Экскурс в историю

Самые известные шифры

Подстановочные и перестановочные шифры (шифр Цезаря)

Перестановочные шифры – роторная машина Энигма и машина для взлома шифра "Bombes"

Шифр Цезаря

- Выполнялась тривиальная подстановка латинского алфавита. В свое время была серьезным средством для связи с войсками и колониями.

Роторная машина

- Изобретена Артур Шребиус,Арвид Герхард Гамм.
- По принципу действия шифратор Enigma напоминал автомобильный одометр: три съемных зубчатых ротора (шифродиска) со сквозными электрическими контактами располагались друг за другом. Когда оператор нажимал клавишу с буквой открытого текста, сигнал проходил через контакты на трех шифродисках, после чего попадал на перемычку рефлектора и отправлялся в обратном направлении (уже по другому «электрическому пути»). Затем первый диск Затем первый диск поворачивался на одну позицию и закон кодирования менялся.

Роторная машина Энигма

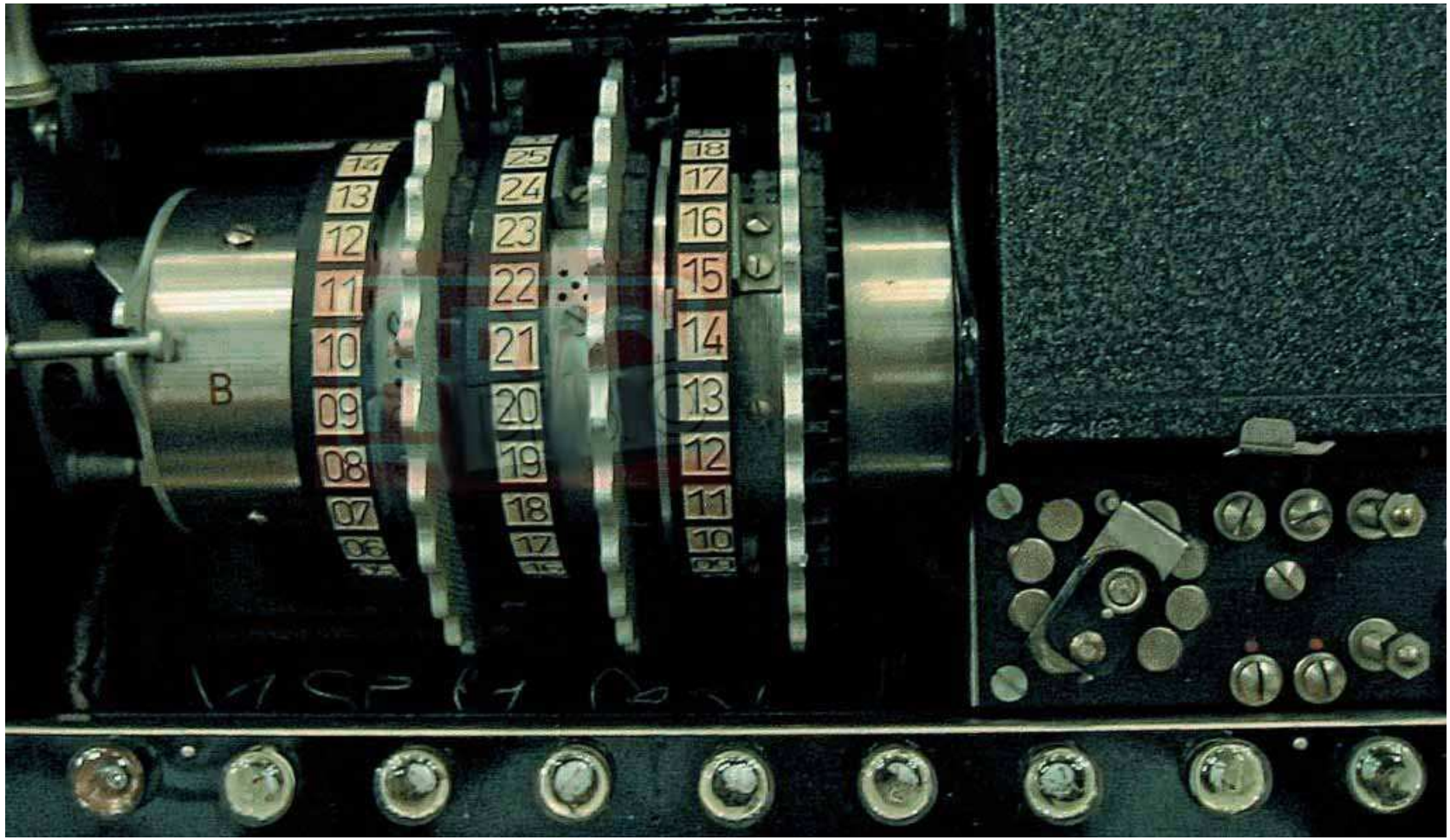
- Как только оператор вводил через клавиатуру 26 знаков, первый диск возвращался в исходное положение, а вот второй проворачивался на позицию вперед.
- Чтобы быстро зашифровать и передать текст с помощью Энигмы, требовалась бригада из четырех человек: один зачитывал вслух открытый текст, второй набивал его на клавиатуре, третий считывал зашифрованную информацию с индикаторов, а четвертый передавал ее в телефонную или телеграфную линию.


Zur Beachtung!
Q W E R T Z U I O
A S D F G H J K
P Y X C V B N M L
Q W E R T Z U I O
A S D F G H J K
P Y X C V B N M L

Период машины

- Ключами к шифрмашине Enigma служили начальное расположение роторов и электрическая коммутация цепей (набор различных роторов)
- период для роторной машины равен 26^n

Роторная машина Enigma



Роторная машина

Считалось, что если Enigma используется правильно, то вскрыть зашифрованную информацию невозможно. Однако при невнимательности связистов шифр терял свою стойкость. Первым, кому удалось «расколоть» Энигму, используя человеческий фактор, стал английский криптоаналитик Алан Тьюринг и группа польских криптоаналитиков.

Машина для вскрытия Bombies

- Для вскрытия кодов Enigm’ы ,была построена машина Bombies. Однако, для она не давала 100% гарантии вскрытия шифра.
- Когда в Берлине поняли,что шифр вскрыт, то машина была срочно модифицирована и в ней появился четвертый ротор. Однако в этот момент битва за Атлантику была проиграна.

Шифрование одноразовым блокнотом

Major Joseph Mayborgne , Gilbert Vernam 1917 AT&T

Используется большая неповторяющееся последовательность символов ключа распределенных случайным образом. Каждый символ текста складывается по модулю 26 с символом ключа, затем использованная часть ключа уничтожается. Расшифровав сообщение, получатель уничтожает свою использованную часть ленты.

Пример:

Сообщение: ONETIMEPAD

Ключ: TBFRGFARFM

Шифрограмма: IPKLPSFHGQ

При использовании нераскрытого абсолютно случайного ключа – дешифрация невозможна.

Компьютерные алгоритмы

Наиболее часто используются:

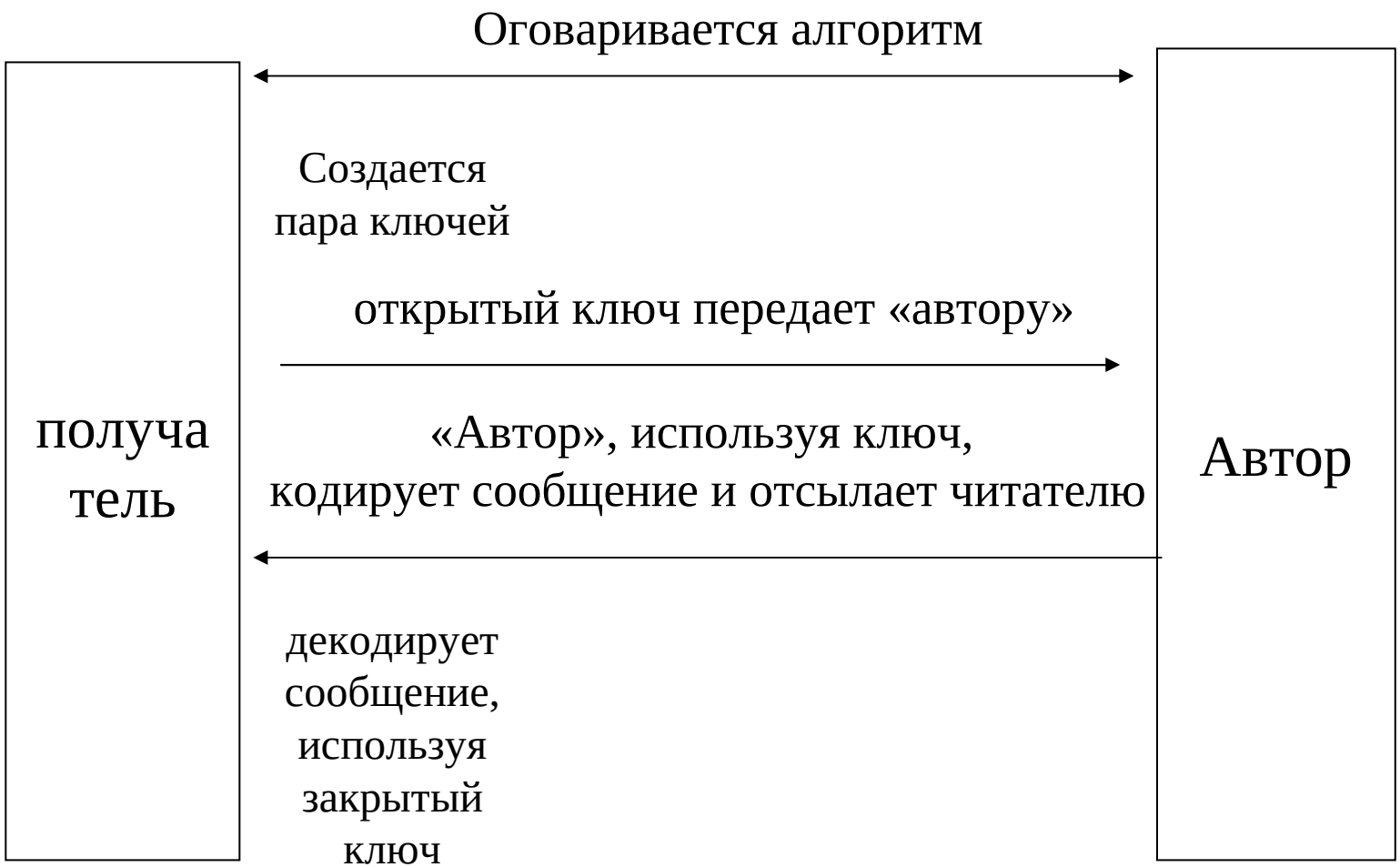
- DES (Data Encryption Standard): самый популярный стандарт шифрования, является международным стандартом-симметричный шифр
- RSA (Rivest Shamir Aldeman) – самый популярный алгоритм шифрования с открытым ключом и цифровой подписи.
- DSA (Digital Signature Algorithm) алгоритм с отурытым ключом, используется только для цифровой подписи

Шифрование с открытым ключом

Пара ключей: открытый и закрытый.

- Кто угодно может зашифровать сообщение, используя открытый ключ.
- Дешифрация сообщения при помощи открытого ключа невозможна
- Получить закрытый ключ путем преобразования открытого невозможно.

Процесс обмена данными



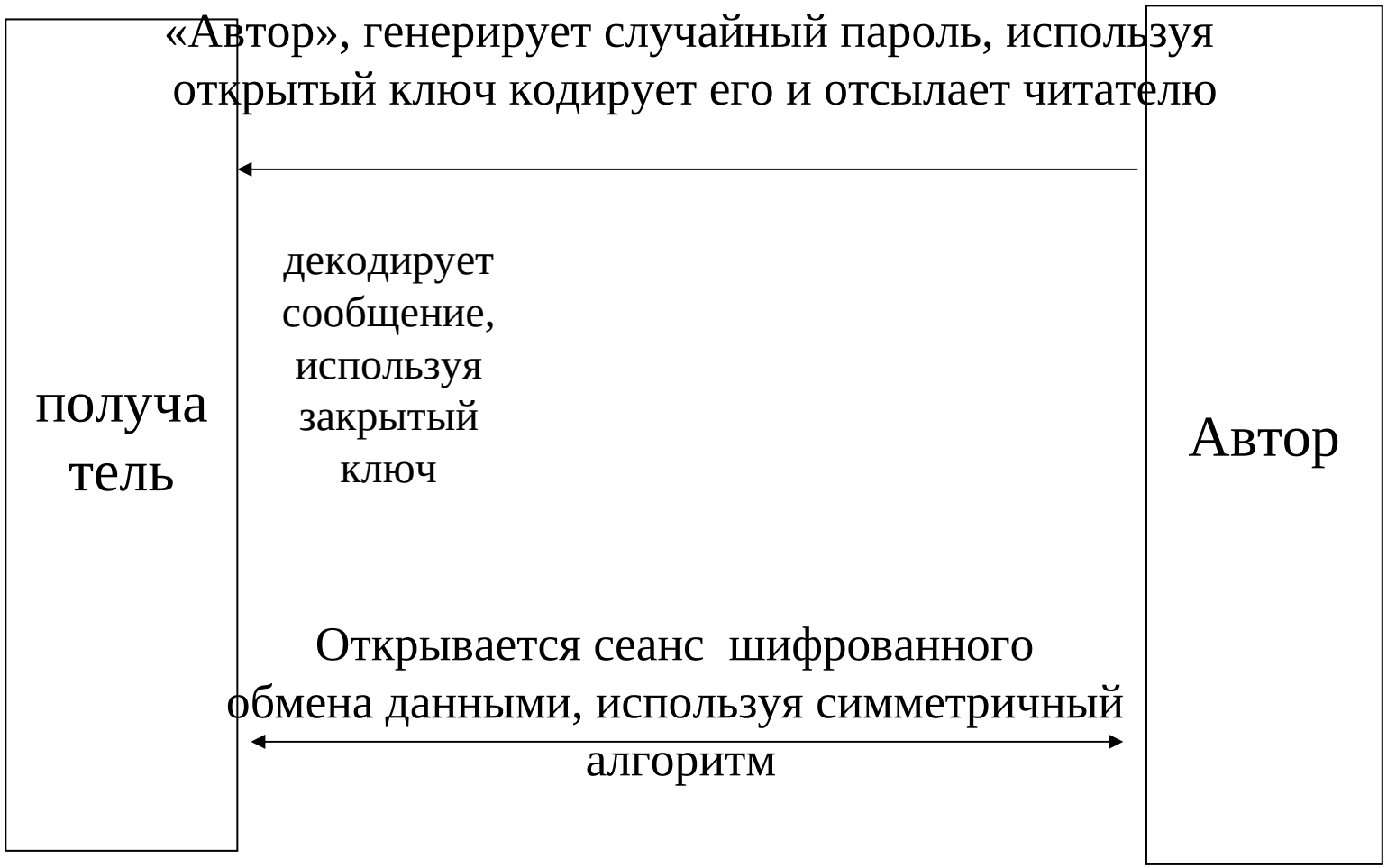
Процесс обмена данными

Недостатки применения в устройствах телемеханики:

- Алгоритмы шифрования с открытым ключом работают медленно (по крайней мере в 1000 раз медленней симметричных алгоритмов)
- криптосистемы с открытым ключом уязвимы по отношению к вскрытию с выбранным открытым текстом.

Решение: использовать криптования с открытым ключом для передачи сеансового ключа. Далее вести сеанс, используя симметричный алгоритм.

Процесс обмена данными



Генерация псевдослучайных последовательностей

Генераторы случайных чисел, встроенные в компиляторы далеко не случайны и не могут гарантировать надежность криптования. Генератор псевдослучайных чисел нам подходит, если:

1. Проходит все известные статистические и математические тесты на случайность.
2. Последовательность непредсказуема
3. Последовательность не может быть уверенно воспроизведена

DES

DES- симметричный блочный шифр, кодирует данные 64-битными блоками. Длина ключа: 56 бит. Алгоритм DES основан на 16 этапах из двух шагов: подстановки и перестановки, зависящих от ключа.

Алгоритм использует только стандартную арифметику 64-битных чисел и логические операции, поэтому идеально подходит для аппаратной реализации.

устойчивость к вскрытию DES грубой силой:

С течением времени с ростом производительности ЭВМ DES становится все менее и менее безопасным. Уже в 1993 году при помощи СуперЭвм (1 миллион $) DES возможно было вскрыть за 3.5 часа.

Обычный современный компьютер позволяет вскрыть 64 битный DES менее чем за несколько часов, суперкомпьютер-за несколько минут. Для решения этой проблемы введен новый стандарт: 3DES, Blowfish и IDEA имеют ключ – 128 бит. Для подбора прямым методом потребуется время большее чем возраст вселенной.

Алгоритмы с открытым ключом

Идея криптографии с открытым ключом связана с вычислением односторонней функции:

Пусть y=F(x). Функцию F назовем односторонней, если по заданному аргументу x легко вычислить значение F(x), однако по значению F(x) значение x трудновычислимо.

В алгортмах шифрования используются такие функции «с лазейкой»

Например, легко вычислить произведение двух простых чисел, но трудно разложить это произведение на множители. Но если известно одно из простых чисел, в примере выше, то, зная произведение, Вы легко найдете второе число.

RSA

Безопасность RSA основана на трудности разложения на множители больших чисел. Открытый и закрытый ключи являются функциями больших (128 разрядов) простых чисел.

Восстановление исходного текста по шифротексту и открытому ключу эквивалентно задаче разложения на простые множители двух больших чисел.

Безопасность RSA

Нигде математически не доказано, что проблема разложения на простые множители больших чисел неразрешима.

Проблема подбора ключей упрощается, если известен фрагмент текста

Цифровая подпись (DSA)

- Использует также дилемму разложения на простые множители и вычисления дискретных логарифмов .
- Применяется для проверки целостности принятых данных и личности отправителя.

Создание и проверка подписей зеркально отличается от зашифрования/расшифрования. При подписи документа используется закрытый ключ подписывающего, а проверяется подпись с использованием его открытого ключа.

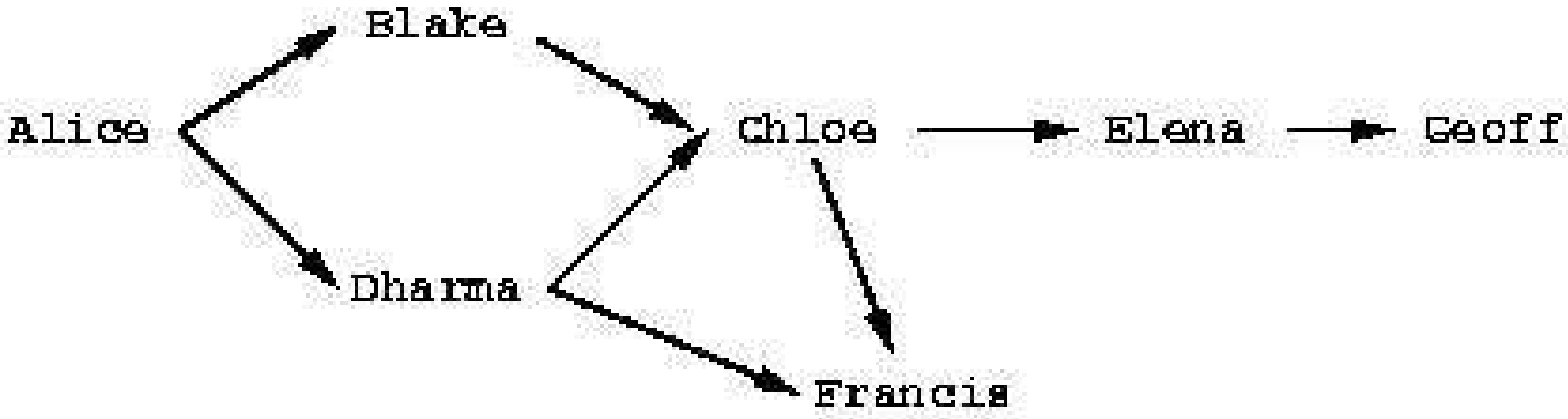
Сети доверия

- Сеть доверия» (Web of trust) механизм подтверждения достоверности ключа. Основа сети доверия – в том, что люди обмениваются подписями открытых ключей с теми, кому они доверяют. Главная идея – ключ автоматически считается достоверным, если выполняются 2 условия:
- 1) Он подписан достаточным количеством ключей, то есть:
  – Либо вы сами подписали его,
  – Либо он подписан одним из полностью достоверных ключей,
  – Либо он подписан тремя частично достоверными ключами.
  2) Количество подписанных ключей в цепочке от этого ключа до вашего ключа – 5 или меньше

5 уровней доверия

- unknown (неизвестное) присваивается по умолчанию ключам в Вашем наборе
- none (нет) Известна недобросовестность этого лица при подписи чужих ключей
- marginal (граничное) :Понимает смысл подписания ключей и проверяет их достоверность перед тем, как подписывать
- full (полное): Владелец очень разборчив в подписи ключей
- ultimate (безоговорочное) Владельцу данного ключа Вы верите как самому себе

Граф достоверности



Gnupg

GnuPG (GNU Privacy Guard, «Страж безопасности GNU», или просто GPG) – это открытый эквивалент PGP (Pretty Good Privacy), известной и широкоиспользуемой программы для Windows, DOS и других операционных систем. Он распространяется в открытых исходниках и имеет те же самые функции, что и PGP, основанные на стандарте OpenPGP. У GnuPG есть несколько применений – он может использоваться либо для шифрования писем и файлов, либо для их цифровой подписи.

GnuPG

GnuPG - инструмент для защиты коммуникаций. Подобно PGP, GnuPG использует смешанные шифры. Использует алгоритм DSA для подписи и алгоритм ElGamal для шифрования.

Применяется смешанное шифрование: сеансовый ключ, зашифрованный шифром с открытым ключом, и сообщение, зашифрованное симметричным шифром, автоматически объединяются вместе. Получатель использует свой секретный ключ для расшифровки сеансового ключа и, затем, использует полученный сеансовый ключ для расшифровки сообщения.

Настройка:

- выбор размера Вашей пары ключей,
- генерация пары ключей
- защита Вашего секретного ключа,
- выбор сроков действия ключей
- управление Вашей сетью доверия.

Генерация ключа

[user@mdk]$ gpg --gen-key
тип: «DSA and ElGamal»,
размер: рекомендовано 1024 бит
срок действия ключа: за Вами
личная информация
пароль
генерация ключей

вывод списка ключей
gpg –list-key

Импорт/экспорт ключей

После создания пары ключей необходимо экспортировать их в файл. Рекомендуется сохранять ключи в текстовом виде

gpg.exe -armor --export Fiery@ngs.ru > myopen.asc

Открытый ключ может быть добавлен к Вашей связке при помощи команды import:

gpg --import newkey.asc

Достоверность импортированного ключа должна быть подтверждена при помощи отпечатка:

gpg –fingerprint fiery@ngs.ru

Отпечатки ключа проверяются его владельцем.

можно сразу поместиить ключ на сервер:

gpg --send-keys --keyserver wwwkeys.pgp.net fiery@ngs.ru

Подписывание ключей

Если Вы уверены, что ключ который Вы импортировали достоверен, его можно подписать. Для подписи ключа необходимо перейти в режим редактирования ключа при помощи команды --edit-key

gpg --edit-key new@mail.ru

sign

потребуется: выбрать степень доверия

После подписания необходимо отправить ключ владельцу

gpg --export -a newr@mail.ru > newsined.asc

Посмотреть все подписи: gpg --list-sigs.

Шифрование/дешифрование файла

gpg -ea –r fiery@ngs.ru test.file - зашифровать

gpg -d test.file.asc >test.file расшифровать

Заключение

GPG может сначала показаться сложным, и, действительно, иногда при работе с ним могут встретиться сложные ситуации. Но в основном работа с GPG проста и понятна. Самое главное – помнить о том, что чужие ключи нужно подписывать с максимальной подозрительностью, потому что от этого зависит ваша репутация. Но подписывать ключи все же надо, потому что это – главный принцип построения сети доверия. А сеть доверия – это очень важная и чрезвычайно нужная вещь...

Подписывание файла

gpg --clearsign -a test.file - создает подпись

gpg --verify test.file.asc - проверяет подпись.

Обобщение- безопасность алгоритмов шифрования

Как мы убедились выше, криптостойкость современных алгоритмов шифрования связана с трудностью решения обратных задач.

Самыми распространенными задачами для построения алгоритмов криптования являются:

- Разложение на простые множители больших чисел
- Вычисление дискретных логарифмов

Безопасность алгоритмов

- Однако нигде математически не доказано, что подобные задачи не имеют другого, более простого решения.
- Кроме того, для устройств автоматики, использующих ключи важную роль имеет защита от промышленного шпионажа.

Квантовый компьютер

- Международная группа ученых-физиков пытается реализовать идею квантового компьютера. Международный негосударственный проект реализуется в Германии. В этом компьютере элементарная ячейка- атом, значение- орбита электрона.
- Предполагается, что при помощи специфических свойств этого компьютера (квантовая физика суперпроводимости при низких температурах) возможного будет мгновенно решать обратные задачи, одной из которых является задача разложения на простые множители

Квантовый компьютер

- В настоящее время идея квантового компьютера в силу технических причин весьма далека от воплощения и работоспособный образец по-видимому появится еще нескоро. Однако, это один из самых серьезных «подкопов» под бастион современных алгоритмов криптографии.

Примеры применения алгоритмов криптографии

Рассмотрим несколько примеров «из жизни» и проблемы в связи с этим возникающих.

Как мы уже отмечали, алгоритмы шифрования и электронной подписи прочно вошли в нашу жизнь.

Рассмотрим два примера:

- автомобильные сигнализации
- Беспроводные сети

Автосигнализации

- На заре развития автомобильных сигнализаций кодом являлась частота и характеристики радиосигнала для связи базового блока и брелока
- Однако, такой метод обеспечивал весьма слабую защиту и не обладал никакой криптостойкостью.

Системы автомобильной безопасности

- Появление автосигнализаций с динамическим кодом позволило решить эту проблему
- До сих пор множество автосигнализаций среднего и низшего ценового диапазона оснащены чипами с системой кодирования KeyLog, разработанный в середине 80х годов южноафриканской фирмой Nanoteq

Keylog

- Последовательность его посылки генерируется на основе
  - 28-битного серийного номера (постоянная часть кода),
  - 64-битного ключа (ключ зашит в микросхему,в канале не присутсвует)
  - и 16-битного счетчика синхронизации (порядковый номер посылки).

Каждая радиопосылка брелока используется только один раз, так что записывать и воспроизводить ее заново бесполезно — код становится устаревшим уже в момент передачи.


Передаваемые данные
Секретный ключ
Счетчик синхронизации
Серийный номер
Массив памяти кодера
Алгоритм шифрования KeeLog
Динамическая часть (32 бита)
Серийный номер
Номер кнопки

Посылка Keylog

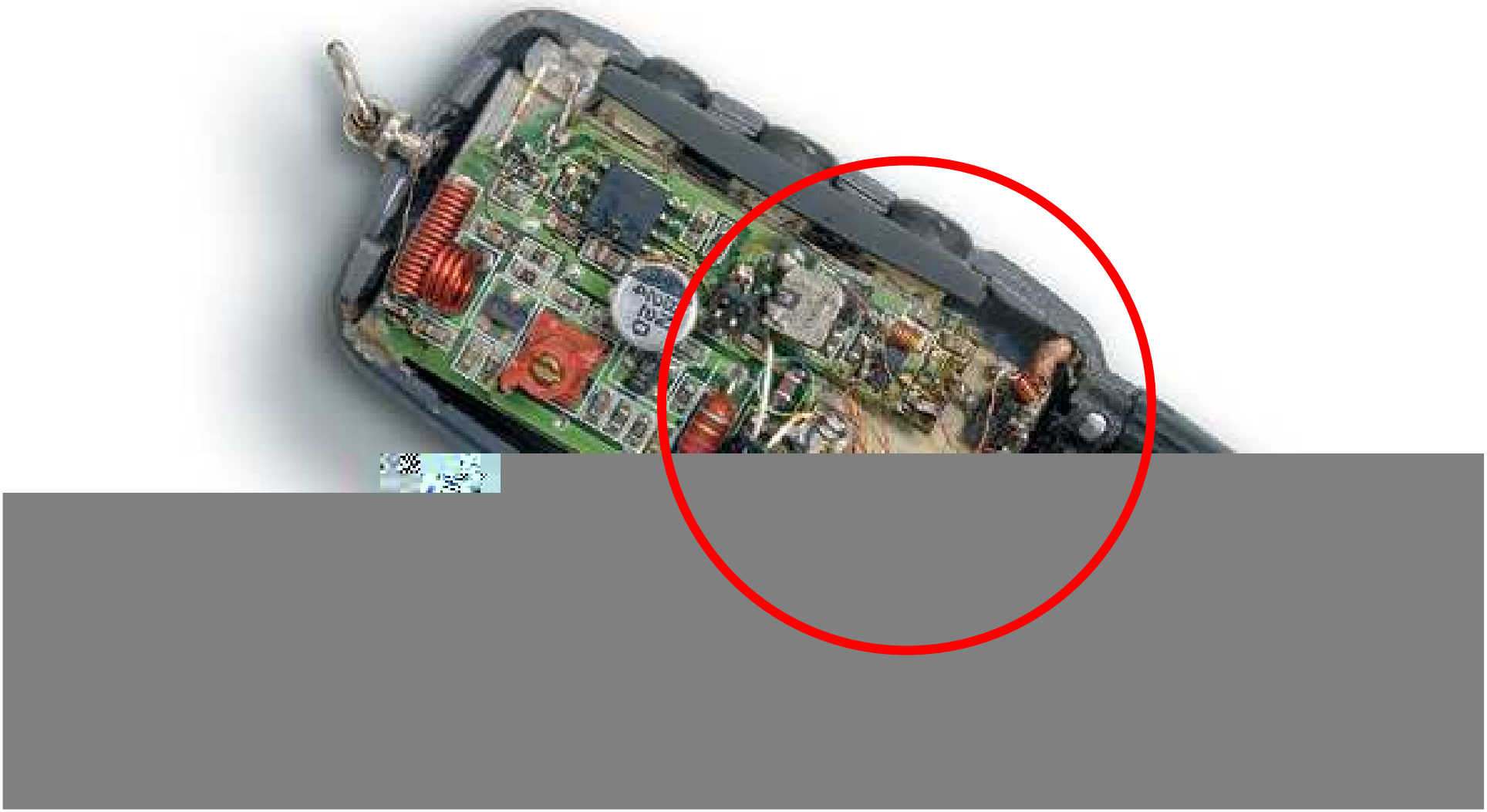
Нестойкость кодов

- К сожалению, для Key-log в настоящее время существуют как минимум два метода его взлома, а следовательно угона Вашего авто

- Рассмотрим сначала метод, с помощью которого злоумышленники могут попытаться перехватить код сигнализаций построенный и на других методах.

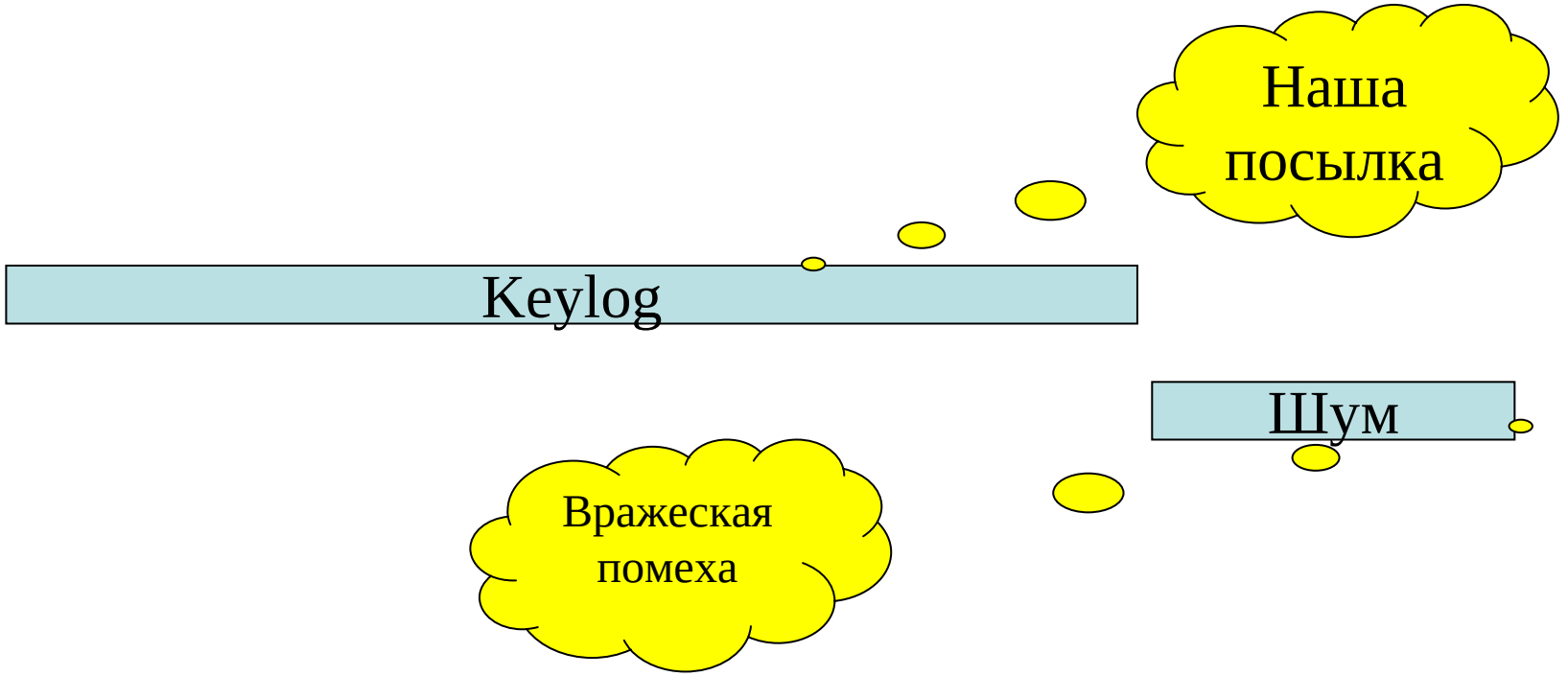
Замещающие код-грабберы

- Идея заключается в провоцировании владельца послать два сигнала на постановку на охрану и затем поменять их местами
- Таким образом, в памяти граббера остается валидный сигнал на снятие с охраны, которым можно воспользоваться только один раз (к сожалению, обычно этого бывает достаточно)



Код-грабберы

- Перехватываем посылку на постановку и портим ее помехой так, чтобы сигнализация ее «не услышала»



Код-грабберы

- Видя, что сигнализация «не сработала», владелец пытается повторить команду и нажимает кнопку еще раз.
- На многих сигнализациях постановка и снятие выполняется одной кнопкой, таким образом, злоумышленники получают код на «открытие», также портят его помехой.
- В момент передачи второй посылки на постановку, злоумышленники портят ее помехой и тут же подсовывают записанную валидную комбинацию.
- Злоумышленникам остается только дождаться пока владелец удалится на безопасное расстояние, поскольку валидная посылка снятия с у них уже имеется.

Симптомы работы код-граббера

- Сигнализация «встает на охрану» с задержкой
- Сигнализация снимается с охраны «с задержкой»

- От замещающих код - грабберов помогает диалоговый код (с перезапросом) – где команда подтверждается с центрального блока. Минус такой сигнализации- низкая помехоустойчивость.

Нестойкость key-log

- К сожалению, в 2006 году коды использующиеся в производстве чипов нынешнего владелеца прав на технологию KeeLog, американской компании Microchip

БЫЛИ УКРАДЕНЫ

Таким, образом все владельцы автосигнализаций с keylog (а таковых,к сожалению большинство), не могут чувствовать себя в безопасности.

Лекция

Лекция 11.

- Обмен электронной почтой.
- Формат заголовка и тела сообщения.
- Расширение MIME.
- smtp, pop3.
- Безопасность и борьба со спамом.

Электронная почта.

В сущности, электронное письмо – это обычный текстовый файл. Но, чтобы почтовые системы всего мира могли разобраться, кому и куда направить письмо, этот текст должен составляться по определенным правилам.

Любое электронное письмо состоит из двух частей:

- официальной (здесь указывается кто, кому, куда, когда послал письмо);
- неофициальной (вот это собственно та информация, которую один человек хочет сообщить другому человеку);

Части разделяются пустой строкой.

Формат заголовка

Формат почтового сообщения Internet определен в документе RFC-822 (Standard for ARPA Internet Text Message).

Заголовок всегда находится перед телом сообщения и отделен от него пустой строкой. RFC-822 регламентирует содержание заголовка сообщения. Заголовок состоит из полей. Поля состоят из имени поля и содержания поля. Имя поля отделено от содержания символом ":"

Электронная почта

Сеть Internet объединяет множество различных компьютеров, работающих в различных операционных системах. В каждой операционной системе есть своя почтовая служба, которая по-особому обрабатывает заголовок письма.

Чаще всего в качестве почтового сервера используется Unix-подобная ОС.

В Unix за обмен электронной почтой отвечает демон sendmail, но последнее время все чаще встречаются альтернативные демоны.

Windows используется в корпоративных сетях как сервер Exchange

Простейший заголовок

Но, чтобы пользователи сети Internet могли свободно общаться друг с другом, в заголовке письма, в соответствии с RFC, обязательно должны присутствовать такие поля:

Date:

From:

To:

Date: дата и время отправления письма; они записываются в стандартном формате - день недели, день, месяц, год (2 цифры), время, временная зона.

From: имя отправителя и его обратный адрес.

To: адрес получателя.

Если в полях адресов содержится какая-либо дополнительна информация, адрес заключается в угловые скобки.

To: “Real Name” <real@ngs.ru>

Пример

Date: Sat, 30 Apr 2005 08:50:01 +0700

To: “Dest” mail@ngs.ru, “Dest2” mail2@ngs.ru

From: “from” <letter@ngs.ru>

Hello, world!!!

другие параметры: Message-Id

Message-Id: уникальный идентификатор сообщения, который компьютер-отправитель присвоит письму. Например: это набор цифр и букв и имя машины. Этот идентификатор можно использовать для

ссылок на письмо в канцелярском деле, как исходящий номер.

другие параметры: Received

Received: отметка о прохождении письма через машину (почтовый штемпель).

Может содержать:

- имя почтового компьютера, пославшего письмо (from домен);
- имя почтового компьютера, принявшего письмо (by домен);
- физический путь следования письма (via ...);
- название протокола передачи данных (with ...);
- номер принятого сообшения (id ...);
- для кого сообщение (for адрес);
- дату прохождения письма через машину.

.

Поля заголовка (необязательные)

Reply-To: Адрес для ответа - адрес отправителя. Это позволяет при ответе на данное письмо (reply) ввести адрес автоматически.

Resent-From: Адрес человека или программы, которые переслали вам сообщение, изначально пришедшее на их адрес

Sender: имя человека или программы, прислявшего вам это

письмо. В общем случае это не то же самое, что From: .

Например, для писем из конференции From: адрес автора письма, а Sender: адрес news-сервера.

Return-Path: <evaluations@vmware.com>
Received: from [172.16.0.1] (HELO intranet.ru)
  by mx3.intranet.ru (CommuniGate Pro SMTP 4.2.4)
  with ESMTP id 29413158 for fiery@ngs.ru; Thu, 28 Apr 2005 17:21:06 +0700
Received: from mailout1.vmware.com ([65.113.40.130] verified)
  by intranet.ru (CommuniGate Pro SMTP 4.2.4)
  with ESMTP id 240354204 for fiery@ngs.ru; Thu, 28 Apr 2005 17:21:00 +0700
Received: from mailhost1.vmware.com (mailhost1.vmware.com [10.16.12.135])
        by mailout1.vmware.com (Postfix) with ESMTP id D5F324525
        for <fiery@ngs.ru>; Thu, 28 Apr 2005 03:20:26 -0700 (PDT)
Received: from script.vmware.com (unknown [10.16.19.13])
        by mailhost1.vmware.com (Postfix) with ESMTP id 32D6E6FC324
        for <fiery@ngs.ru>; Thu, 28 Apr 2005 03:20:27 -0700 (PDT)
Received: (from evaluations@localhost)
        by script.vmware.com (8.11.6/8.11.6) id j3SAKAh25571;
        Thu, 28 Apr 2005 03:20:10 -0700
Date: Thu, 28 Apr 2005 03:20:10 -0700
Message-Id: <200504281020.j3SAKAh25571@script.vmware.com>
To: fiery@ngs.ru
From: wseval@vmware.com
Subject: Don't Delay! Time's Running Out On Your VMware Workstation Evaluation
Reply-To: wseval@vmware.com

Поля заголовка(необязательные)

Return-Receipt-To: Адрес, по которому нужно отослать "увеломление о доставке".

В большинстве случаев это адрес отправителя.

X-Mailer: Программа, с помощью которой было отправлено письмо. Например, dMail, ELM, TheBat

Subject: Тема письма.

Newsgroups: название конференции или нескольких конференций через запятую.

Expires: хранить в конференции до указанного числа.

Keywords: ключевые слова, по которым можно искать статью в конференции.

Lines: количество строк в письме

Return-Path: <postmaster@intranet.ru>
Received: from [212.17.5.144] (account <postmaster@intranet.ru>)
  by intranet.ru (CommuniGate Pro WebUser 3.4.8)
  with HTTP id 143833695 for <all@ngs.ru>; Thu, 30 Sep 2004
    11:28:03 +0700
From: <support@ngs.ru>
Subject: [НГС] Уведомление
To: all@ngs.ru
X-Mailer: CommuniGate Pro Web Mailer v.3.4.8
Date: Thu, 30 Sep 2004 11:28:03 +0700
Message-ID: <web-143833695@intranet.ru>
MIME-Version: 1.0
Content-Type: text/plain; charset="KOI8-R"
Content-Transfer-Encoding: 8bit

Уважаемые пользователи почтовой службы НГС!

Проблема поддержки национальных кодировок

Первоначально электронная почта была предназначена исключительно для передачи текстовых сообщений, содержащих ASCII символы. Если же требовалось передать двоичный файл или текст на языке отличном от английского, то возникала необходимость кодирования такого файла или текста символами ASCII. Далее, закодированное сообщение передавалось с помощью обычных средств электронной почты. Принимающая сторона (пользователь) должна быть извещена о способе кодирования и должным образом декодировать сообщение. Одна из таких кодировок - UUE.

Предшественник MIME-UUE

----------> <----------

Текст в кодировке win1251.

----------> <---------- 28байт

section 1 of uuencode 4.21 of file PRIMER.TXT by R.E.M.

begin 644 PRIMER.TXT

<DJ6JX>(@HB"JKJ2HX*ZBJJ4@=VEN,3(U,2X-"BX-

`

end

sum -r/size 28678/69 section (from "begin" to "end")

sum -r/size 64566/28 entire input file

----------> <---------- 69/230 байт

Mime

Разнородность сетей и обилие не стандартизированного ПО различных производителей зачастую не позволяло пользователям "понимать" друг друга. Причины проблем:

1. Разные клиенты работали с разными кодировками.
2. Не была определена структура размещения и идентификации типа закодированных данных. Чтобы понять, что собой представляет полученная информация, ее необходимо было "вынуть" из сообщения и декодировать

Mime

С ростом популярности E-mail и multimedia возникла необходимость в одном сообщении передавать данные различных типов:

- текстовую информацию на различных языках,
- графические изображения,
- видеопоследовательности,
- голосовые сообщения (аудиоинформацию),
- и просто, бинарные файлы;

Mime

Указанные проблемы были решены путем внедрения стандарта MIME (Multipurpose Internet Mail Extention, многоцелевое расшироение интернет почты)

Стандарт не заменяет, а расширяет существующий способ формирования электронных сообщений.

MIME - новый формат представления данных, представляющий почтовому клиенту гибкий интерфейс для работы с E-mail.

Mime

Для идентификации MIME-сообщений в заголовке сообщения должны присутствовать следующие поля:

Mime-Version: - версия MIME, например, 1.0 или 1.1

Content-Type: тип/подтип – тип сообщения.

Content-Transfer-Encoding: - используемый метод кодирования для передачи.

Другие поля конкретизируют какие-либо параметры и обязательными не являются.

Кодировки

Возможные значения: **base64, quoted-printable, 8bit, 7bit, binary.**

- 8 bit – сообщения,в которых включен 8й бит. Может не поддерживаться некоторыми национальными почтовыми службами. В этом случае 8 бит отбрасывается, данные могут быть потеряны.
- quoted-printable (RFC-1341) - кодирует любые не ASCII символы, позволяет передавать их вперемешку с первыми. Символ представляет собой последовательность из знака равно ("=") и шестнадцатиричного кода символа.

“Привет” -> =CF=F0=E8=E2=E5=F2 (windows-1251)

Кодировки

base64. Наиболее распространенная кодировка для передачи файлов.

- Битовый поток разбивается по 24 бита (по 3 байта), которые в свою очередь делятся на четыре части по 6 бит.
- Каждая такая часть кодируется одним из 64 ASCII символов (отсюда название - **base64**).

0->A, 1->B ,2->C ….. 62->+ , 63->-

Структура сообщения

Каждое mime сообщение может состоять из нескольких частей (multipart), разделенных «разделителем». При этом в заголовке указывается:

Content-type: multiaprt/подтип; bound="разделитель“

Подтип:

- mixed - все части обрабатываются последовательно;
- parallel - все части обрабатываются параллельно;
- alternative - интерпретация определяется клиентом;

Заголовок rfc822
Content-Type: multipart/alternative; boundary=“---- Next”

Текст rfc822
---- Next

Content-Type: text/plain; charset="koi8-r"
Content-Transfer-Encoding: base64

DQoNCi0tLS0tT3JpZ2luYWwgTWVzc2FnZS0tLS0tDQpGcm9tOiBqb2huQHZ
0YXUtYnNkLnBzdHUu
YWMucnUgW21haWx0bzpqb2huQHZ0YXUtYnNkLnBzdHUuYWMucnVdIA
0KU2VudDogTW9uZGF5LCBG
ZWJydWFyeSAwOCwgMTk5OSAxOjM3IFBNDQpTdWJqZWN0OiANCg0K
DQrN8yDt4Oru7eXpCPcICPYt
---- Next

Содержимое блоков

В каждом блоке указывается свой Content-type содержимого, определяющий как интерпретировать информацию в данном конкретном блоке. Возможные типы:

- **Текст(гипертекст)**

text/*подтип*; charset="кодировка"

подтип:

- text (txt)- простой текст;
- html (htm, html)- форматированный текст HTML;

Содержимое блоков

- **Изображение**

image/подтип; name="имя_файла"

подтип: jpg или jpeg, gif, bmp

- **Видео**

video/подтип; name="имя_файла"

подтип: **mpeg, x-msvideo** (avi), **quicktime** (qt)

Подтипы

- **Звук**

audio/подтип; name="имя_файла" Значения подтипа: ra, wav, basic (au)

- **Общий формат**

application/подтип; name="имя_файла"

Значения подтипа

  - octet-stream - бинарный файл (исполняемый или др. файл);
  - msword (doc)- файл MS Word;
  - x-compress (z), x-compressed (tgz), x-gzip (z), z-tar (gz), x-zip-compressed (zip)- файлы в сжатых архивах

Присоединение файла

Рассмотренные выше директивы в общем случае позволяют почтовой программе отображать данные внутри тела письма. Следующая директива предназначена для присоединения файла без права отображения содержимого в письме:

Content-Disposision attachment; filename="имя_файла" - прикрепленный файл, не интерпретируется почтовым клиентом. Для просмотра файл необходимо сохранить на локальном диске.

Тем не менее, такую возможность можно разрешить, указав директиву **inline** - если почтовый агент "знает" формат файла, то он будет отображен прямо в теле сообщения.

From user@email.net Wed Feb 10 16:15:17 1999
To: ivanov@nags.com
Subject: FW: please resend it, because i can't translate it at work
Date: Wed, 10 Feb 1999 09:30:57 +0300
MIME-Version: 1.0
X-Mailer: Internet Mail Service (5.5.1960.3)
Content-Type: multipart/alternative;
boundary="---- =_NextPart_001_01BE54E2.10C07C70"

This message is in MIME format. Since your mail reader does not understand
this format, some or all of this message may not be legible.

------ =_NextPart_001_01BE54E2.10C07C70
Content-Type: text/plain; charset="koi8-r"
Content-Transfer-Encoding: base64

DQoNCi0tLS0tT3JpZ2luYWwgTWVzc2FnZS0tLS0tDQpGcm9tOiBqb2huQHZ0YXUtYnNkLnBzdHUu
YWMucnUgW21haWx0bzpqb2huQHZ0YXUtYnNkLnBzdHUuYWMucnVdIA0KU2VudDogTW9uZGF5LCBG
ZWJydWFyeSAwOCwgMTk5OSAxOjM3IFBNDQpTdWJqZWN0OiANCg0KDQrN8yDt4Oru7eXpCPcICPYt

------ =_NextPart_001_01BE54E2.10C07C70
Content-Type: application/msword
Content-Disposition: inline
Content-Transfer-Encoding: base64

PCFET0NUWVBFIEhUTUwgUFVCTElDICItLy9XM0MvL0RURCBIVE1MIDMuMi8vRU4iPg0KPEhUTUw+
DQo8SEVBRD4NCjxNRVRBIEhUVFAtRVFVSVY9IkNvbnRlbnQtVHlwZSIgQ09OVEVOVD0idGV4dC9o
dG1sOyBjaGFyc2V0PWtvaTgtciI+DQo8TUVUQSBOQU1FPSJHZW5lcmF0b3IiIENPTlRFTlQ9Ik1T

------ =_NextPart_001_01BE54E2.10C07C70-

Передача электронной почты.

Мы рассмотрели формат заголовка и тела сообщения. Рассмотрим способы передачи сообщений в сети от узла к узлу.

По умолчанию сообщение передается на узел назначения напрямую, используя IP-адрес, но исключительно по информации MX в DNS.

Чтобы объявить о предоставлении услуги обмена почтой, сетевой узел должен обладать записью MX (Mail Exchanger) в базе данных DNS.

Имя сервера ,ответственного за обработку почты, не обязано совпадать с именем почтового домена

Маршрутизация почты в Интернет

Каждая запись MX содержит параметр приоритета (preference). Параметр приоритета- положительное целое число. Почтовый агент будет пытаться переслать сообщение на сервер MX,имеющий наименьшее значение приоритета, только в случае неудачи сообщение будет передано на узел с более высоким значением приоритета.

Пример записи:

green.foolbar.com IN MX 5 mailhub.foolbar.com

Протокол SMTP

SMTP (simple mail transport protocol) – основной протокол передачи электронной почты в сети Интернет. SMTP постепенно вытесняет использовавшийся ранее протокол (UUCP). Для работы SMTP создает соединение с сервером (порт 25), затем клиент с сервером обмениваются информаций пока соединение не будет закрыто.

Самой первой процедурой является открытие канала, самой последней-закрытие.

Команды smtp

Команды SMTP указывают серверу, какую операцию хочет произвести клиент. Команды состоят из ключевых слов, за которыми следует один или более параметров. Ключевое слово состоит из 4-х символов и разделено от аргумента одним или несколькими пробелами. Каждая командная строка заканчивается символами CRLF.

Команды smtp.

- HELO <SP> <domain> <CRLF> - приветствие
- MAIL <SP> FROM:<reverse-path> <CRLF> Отправитель
- RCPT <SP> TO:<forward-path> <CRLF> Получатель
- DATA <CRLF> Данные
- RSET <CRLF> прервать текущий процесс
- SEND <SP> FROM:<reverse-path> <CRLF> доставить на терминал
- SOML <SP> FROM:<reverse-path> <CRLF> Send+Mail
- SAML <SP> FROM:<reverse-path> <CRLF> Send+Mail
- VRFY <SP> <string> <CRLF> Проверка имени пользователя
- EXPN <SP> <string> <CRLF> Проверка почтовой группы
- HELP <SP> <string> <CRLF>
- NOOP <CRLF> Пустой оператор
- QUIT <CRLF> завершить работу.

Пример smtp сесиии

C-client S-server
#Поздоровались
C: HELO 195.161.101.33
S: 250 smtp.mail.ru is ready
#сообщили адреса
C: MAIL FROM:<drozd> #указываем отправителя
S: 250 OK
C: RCPT TO:<drol@mail.ru> #указываем получателя
S: 250 OK
C: DATA сообщаем,что дальше идут данные
S: 354 Start mail input; end with <CRLF>.<CRLF>
передачу письма необходимо завершить символами CRLF.CRLF

S: 250 OK
S: QUIT
C: 221 smtp.mail.ru is closing transmission channel

SMTP в Unix

В ОС семейства Unix за реализацию SMTP отвечает демон sendmail – невероятно мощная программа. Руководство к sendmail содержит более 800 страниц текста, что способно отпугнуть даже бывалых компьютерщиков. Тем не менее, настройка по-умолчанию + минимальные конкретизирующие настройки позволяет без проблем работать с этим демоном, оставляя все тонкости и нюансы на откуп энтузиастов.

Другой популярный демон- postfix

Настройка sendmail

Все индивидуальные настройки заносятся в файл sendmail.mc , затем при помощи макропроцессора m4 создается сам конфигурационный файл sendmail.cf .

m4 sendmail.mc > sendmail.cf

sendmail.cf

```
DOMAIN(generic)
define(`confDOMAIN_NAME',`pogoda.nsk.su')
include(/usr/share/sendmail/cf/m4/cf.m4)
OSTYPE(linux)dnl
define(`ALIAS_FILE',`/etc/mail/aliases')
define(`SMART_HOST',`relay.turbosib.ru')

FEATURE(redirect)
FEATURE(local_procmail)
FEATURE(always_add_domain)
FEATURE(masquerade_entire_domain)
FEATURE(`accept_unresolvable_domains')
FEATURE(`accept_unqualified_senders')
FEATURE(allmasquerade)
FEATURE(`relay_entire_domain')

MASQUERADE_AS(pogoda.nsk.su)
MASQUERADE_DOMAIN(pogoda.nsk.su)
```

Перенаправление почты

Можно перенаправить поток почты, поступившей для одного пользователя другому пользователю или процессу (в этом случае процесс получает данные письма на stdin).

#/etc/aliases

postmaser: john,alex

robot: |/usr/bin/robot.pl,alex,control@ngs.ru

file: /root/myfile.mail

Изменения в таблице закрепляются командой newaliases.

отправка почты непосредственно из процесса

Удобна для генерации автоматических отчетов и другой информации.

1. Необходимо сгенерировать сообщение с полями согласно rfc822
2. Передать его программе sendmail с ключом . Можно открывать pipe с непосредственно в момент открытия файла.

   open MAIL,"/usr/lib/sendmail –t –oi";

Сообщение будет передано немедленно,если невозможно-будет помещено в системную очередь сообщений.

mailq – выводит системную очередь сообщений

Получение почты из ящика клиентом. Pop3

Протокол smtp предназначен для обмена сообщениями между узлами(серверами). Полученная почта накапливается в почтовых ящиках пользователей (/var/spool/mail). Для получения почты с сервера обычно используется протокол pop3 и его модификации.

pop3

Перед работой через протокол POP3 сервер прослушивает порт 110. Когда клиент хочет использовать этот протокол, он должен создать TCP соединение с сервером. Когда соединение установлено, сервер отправляет приглашение. Затем клиент и POP3 сервер обмениваются информацией пока соединение не будет закрыто или прервано.

Pop3 - правила

Команды POP3 состоят из ключевых слов, за некоторыми следует один или более аргументов. Все команды заканчиваются парой CRLF. Ключевые слова и аргументы состоят из печатаемых ASCII символов. Ключевое слово и аргументы разделены одиночным пробелом. Ключевое слово состоит от **3-х до 4-х символов**, а аргумент

может быть длиной до 40-ка символов.

Ответы в POP3 состоят из индикатора состояния и ключевого слова, за которым может следовать дополнительная информация. Ответ заканчивается парой CRLF. Существует только два индикатора состояния:

- "+OK" - положительный и
- "-ERR" - отрицательный.

Ответы из нескольких строк заканчиваются “.” и CRLF

pop3 - команды

USER [имя] – задает имя пользователя

Возможные ответы:

- * +OK name is a valid mailbox
- * -ERR never heard of mailbox name

Команда: PASS [пароль] – задает пароль пользователя

Возможные ответы:

- * +OK maildrop locked and ready
- * -ERR invalid password
- * -ERR unable to lock maildrop

STAT – выдает количество сообщений в ящике и их длину

* +OK n s

List [сообщение] – выдает информацию об указанном сообщении

- * +OK scan listing follows
- * -ERR no such message

TOP [сообщение] [n]- List + n строк сообщения

pop3 - команды

RETR [сообщение] передать тело сообщения

Возможные ответы:

- * +OK message follows
- * -ERR no such message

DELE [сообщение] сообщение помечается как удаленное, удаление по команде UPDATE

- * +OK message deleted
- * -ERR no such message

RSET – снимает метку удаленных сообщений

S: <создаём новое TCP соединение с POP3 сервером через порт 110>

S: +OK POP3 server ready
C: USER Monstr
S: +OK User Monstr is exists
C: PASS mymail
S: +OK Monsr's maildrop has 2 messages (320 octets)
C: STAT
S: +OK 2 320
C: LIST
S: +OK 2 messages (320 octets)
S: 1 120
S: 2 200
S: .
C: RETR 1
S: +OK 120 octets
S:
S: .
C: DELE 1
S: +OK message 1 deleted
C: RETR 2
S: +OK 200 octets
S:
S: .
C: DELE 2
S: +OK message 2 deleted
C: QUIT
S: +OK dewey POP3 server signing off (maildrop empty)
C: <закрываем соединение>

Спам

Если у спамера нет адреса Вашей электронной почты – то и спама Вы не получите.

Чтобы гарантировно «нарваться» на спам необходимо:

- Поместить адрес электронной почты на хорошо посещаемый сайт.
- Написать письмо (или ответить на письмо) на сайтах конференций
- Поместить пост или ответить на пост на популярном интернет-форуме.
- Зарегистрироваться или иным образом оставить свой адрес электронной почты на сайте фирмы, которая выходит из бизнеса и продает свою базу данных.
- Зарегистрироваться или иным образом оставить свой адрес электронной почты на сайте, который продает свои базы данных.
- Подписаться на порнорассылку.
- Ответить на несанкционированный e-mail.
- Дать простое имя своему адресу электронной почты. Например, director@compania.com
- Зарегистрировать доменное имя и указать там контакный адрес.
- Указать свой адрес электронной почты в интернет-чате
- Получить вирус
- Общаться с коллегами, пренебрегающими антивирусными средствами

Чем вреден спам?

1. Понижением производительности компании.
2. Случайной потерей важных сообщений при ручной чистке электронной почты.
3. Угрозой стабильности работы почтовых серверов.
4. Опасным содержанием: вирусами, троянами, запрещенными материалами.
5. В настоящее время примерно 75-80% входящих сообщений - спам. Это означает, что три четверти своего дискового пространства и процессорной мощности ваш почтовый сервис затрачивает на обслуживание бизнеса спамеров.
6. Паразитным трафиком.
   Для провайдеров - затратами на службу поддержки, конфликтами с клиентами.

Цели спамеров:

Реклама товара.

Раскрутка сайта.

Информация может быть разнообразной, но, в основном, рекламируется что-то очень хорошее и/или бесплатное. Ссылка же ведет на сайт, который совершенно не имеет отношения к этой информации. Но рейтинг сайта повышается за счет обманутых посетителей. Иногда послание может быть совершенно пустым, а страница со счетчиком программно открывается в новом окне.

Платные звонки.

Рекламируется товар и указывается номер телефона. Позвонив, Вы услышите только автоответчик, а потом Вам придет счет за соединение.

Реклама денежных пирамид.

Обещают баснословные барыши, но сначала Вы должны выслать какую-то небольшую сумму по указанному адресу.

Сбор информации.

Под видом опроса или заказа предлагают заполнить анкету и отослать Ваши данные по указанному адресу.

Засылка троянов.

Троян собирает необходимую информацию с Вашего компьютера (пароли, номера телефонов провайдера и пр.) и отсылает ее обратно.

Попытки «фишинга»(рыбалки) Подставные клоны легальных сайтов.

На войне как на войне

- Нельзя сбрасывать со счетов, что нам противостоит очень умный и квалифицированный противник, для которого спам- это его хлеб (иногда с маслицем и икоркой)
- Спам молод и хитер. Как средство активного маркетинга он возник примерно в 1997 году
- Эволюция технических видов спама на 100 процентов обусловлена эволюцией антиспамовых средств. Причем история тут развивается стремительно, по нарастающей. За последние два года в ней, по-видимому, произошло больше событий, чем за все предшествующие.

Методы

- Прямые рассылки и открытые релеи (обычные почтовые сервера, позволяющие произвольному пользователю воспользоваться сервисом отправки письма на другой сервер)
- Прокси-сервера. Socks и HTTP

обнаружилось,что некорректно настроенные прокси серверы могут использовать спамеры и для направления своего SMTP-трафика

- Взломанные машины. Стандартное ПО. Модифицированое ПО. Троянские кони.

Распространиение Интернет в Европе и Америке всеобъемлющее, квалификация администраторов в целом чрезвычайно низкая,ПО стандартное.

Борьба со спамом

- SPAM – фильтры динамическое распознавание по сигнатурам (аналог антивируса)
  - по IP
  - по заголовку
  - по телу сообщения
- Запрет неизвестных релеев
- Тайм-аут широковещательных рассылок.

Чёрно-белые списки

- Создается база данных адресов с которых прием почты запрещен. База данных периодически обновляется
- +Простота решения.
- - Большое число ложных срабатываний, спамеры редко используют одни и те же релеи.
- Результат- спам все равно приходит, письма от коллег приходить перестали….

Проблема промежуточной зоны

- Очень важная, часто недопонимаемая проблема состоит в том, что спам и не-спам пересекаются в очень большой степени.
- Рассылки, от которых трудно отписаться, но на которые вы тем не менее (кажется?) подписывались. Подписки, возникающие при регистрации, без вашего ведома. Многочисленные квитанции глупых антиспамерских и антивирусных программ. Автоответчики. Рассылки, совершаемые спамерами при помощи веб-форм из публичных, совершенно неспамерских веб-сервисов, тем не менее слабо защищенных от вторжения. Например, открытки или приглашения вступить в то или иное веб-сообщество – по тексту такого письма даже автор не может понять, спам это или нет. Вся такая корреспонденция может быть смело отнесена к «полуспаму».

Проблема «полуспама»

- Перед началом очередного этапа работ по антиспамовой фильтрации специялисты Яндекса провели исследование. Был проведен ручной анализ достаточно репрезентативной выборки из 5151 писем, пришедших на 300 адресов. Так вот, ситуации, когда проверяющий посторонний человек, используя для принятия решения всю мощь своего естественного интеллекта, отнес письмо к такой «промежуточной зоне» составляли до 40 процентов! При этом правило для такого отнесения было достаточно осторожным:
- «Полуспамовое» письмо – это письмо от известного проверяющему **реально работающего** магазина или онлайн-сервиса, в котором пользователь **скорее всего** регистрировался. …

Статья в yandex

http://company.yandex.ru/articles/spamoborona.html - том, как это решают в Яндекс

Метод получения почты с перезапросом (серый список)

Оперирование производится триплетами - IP адрес релэя, @ сендера, @ рецепиента. Когда поступет новое(по триплету) для базы письмо, сервер говорит удаленной стороне TEMPFAIL в течение 58 минут. То-есть письмо не доходит. По прошествии этого времени, открывается 4-х часовое окно, в течение которого база ждет подтверждения триплета (повторной передачи письма). Если триплет не подтверждается, запись удаляется из базы. То-есть наш сервер надо уговорить принять почту. Теоретически некто, посылающий напрямую (без релэя) на наш сервер почту раз в пять часов (или релэй с ETRN=5ч.), может никогда ее не доставить. Но такая ситуация почти невероятна, так-как релэи всегда делают ретрансмиссию по таймауту, а напрямую человек (спамер) «устанет» слать письма именно с такой периодичностью. Кроме того, фильтр имеет функции проверки почтового клиента в случае отправки напрямую. Даже если такая ситуация возникнет, есть белый список для ее обхода.

Если триплет подтверждается (resend or other mail), фильтр заносит триплет в белый список на 36 дней. Разумеется, все таймауты можно изменить на свое усмотрение.

Недостатки:

- Первое письмо приходит с некоторой задержкой (в нашем случае 58 минут)
- Метод сработает только для «прямых релеев» спамеров. Если письмо пришло через транзитный релей(или взломанный сервер) провайдера это не поможет.

Обоснование таймаутов в методе

Задержку в 58 минут не имеет смысла уменьшать потому-что:

1) Час задержки не заметит большинство пользователей и ни один почтовый сервер. Для пользователей - это происходит только первый раз для триплета. Далее задержек не будет. Для серверов - инсталляции по умолчанию сконфигурированы таким образом, что делают попытки ретрансмиссии в течение 5-7 дней!

2) Если релэй взломан, час необходим админу для обнаружения и решения этой проблемы. Если это сознательный спамерский открытый релей, час необходим чтобы кто-то занес его в блэклисты.

Однако если спамер будет настойчив, он таки "уговорит" наш MTA принимать спам. То-есть мы примем рано или поздно все, что нам послали, если оно будет посылаться с чего-то релэя вновь и вновь. Чтобы однозначно отвергать спам, сервер должен параллельно использовать другие техники защиты.

- **Вывод**: очевидно спам нельзя победить «хорошим» протоколом. Но спам можно побеждать совместными усилиями антиспамерского ПО, систем обратной связи, а также согласованных действий провайдеров.

Лекция

Лекция 12.

- PPP
- Модемы для коммутируемых линий связи, Hayes AT система команд
- Передача данных через GSM-сети, AT команды GSM устройств
- Подключение к СМСЦ, протокол SMPP.

Низкоскоростные ЛС.

Ранее мы, в основном, рассматривали вопросы обмена IP пакетами по скоростным локальным сетям Ethernet.

Рассмотрим методы организации IP сетей по низкоскоростным линиям связи.

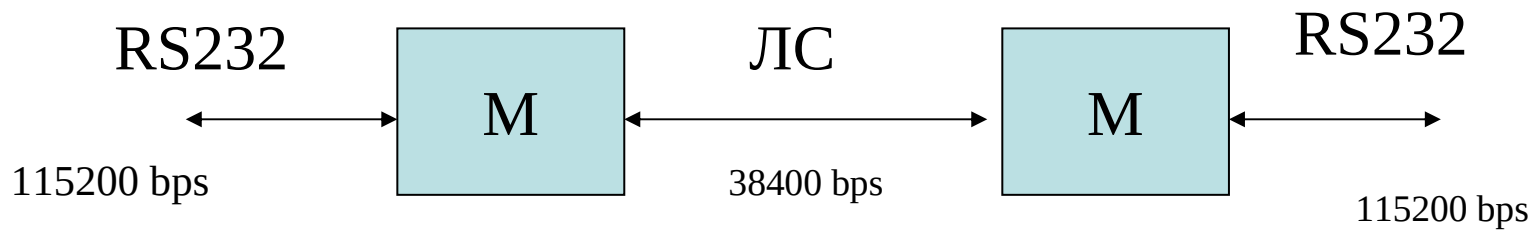
К низкоскоростным линиям отнесем:

- Выделенные и коммутируемые аналоговые телефонные ЛС (ширина полосы 3 Кгц)
- Радиомодемы
- Лазерные воздушные ЛС
- Линии связи поверх силовых линий в энергетике

Каналы точка-точка

Особенность:

- Оконечное устройство работает через порт RS232
- Байт, отправленный на вход передается на выход «партнера»
- Скорость в канале обычно не равна скорости в порту



SLIP

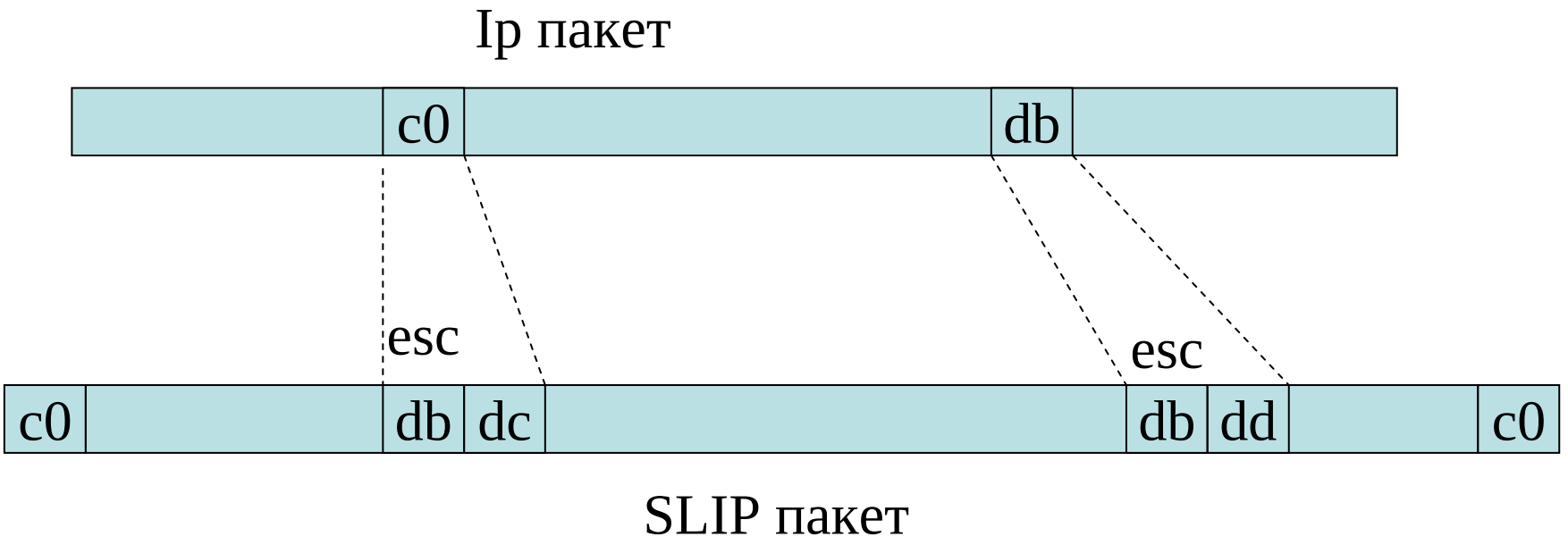
Первая реализация протокола для последовательных линий:

SLIP- Serial Line IP – межсетевой протокол для последовательных линий (RFC 1055 [Romkey,1988])

Выполняется тривиальная трансляция IP пакета в ЛС. Пакет обрамляется признаком границы

- В начале и в конце пакета передается символ END (0xc0).
- END в теле пакета заменяется 0xC0->(0xDB,0xDC : ESC 0xDC)
- ESC в теле пакета заменяется ESC->(0xdb,0xdd)

SLIP



SLIP, недостатки

- Хост должен заранее знать IP адрес партнера, возможность обмена IP адресами не заложена
- Не предусмотрена возможность дифференциации инкапсулируемых пакетов, нельзя обслуживать одновременно несколько протоколов (IP,IPX).
- Нет контроля корректности – необходимо проверять CRC на верхних уровнях
- Хост не может определить состояние линии связи, если нет обмена данными- трудно отследить обрыв связи.
- Не поддерживается синхронный побитовый режим

PPP

Протокол PPP (Point to Point Protocol RFC 1548 [Simpson 1993]) свободен от перечисленных недостатков SLIP

Протокол содержит три составляющие:

1.Метод PPP-инкапсуляции(синхр. и асинхр 8-бит режимы)

2.Протокол управления каналом LCP (Link Control Protocol). Служит для установки и тестирования соединения на канальном уровне путем переговоров между сторонами

3.Сетевые протоколы управления NCP (Network Control Protocol).

PPP

Addr ff	ctrl 03	Протокол 2 б	Данные до 1500 б	CRC 2 б	Flag 7e

Протокол 00 21	IP- пакет

Протокол c0 21	Данные управления каналом LCP

Протокол 80 21	Данные управления сетью NCP

PPP

Обработка служебных символов

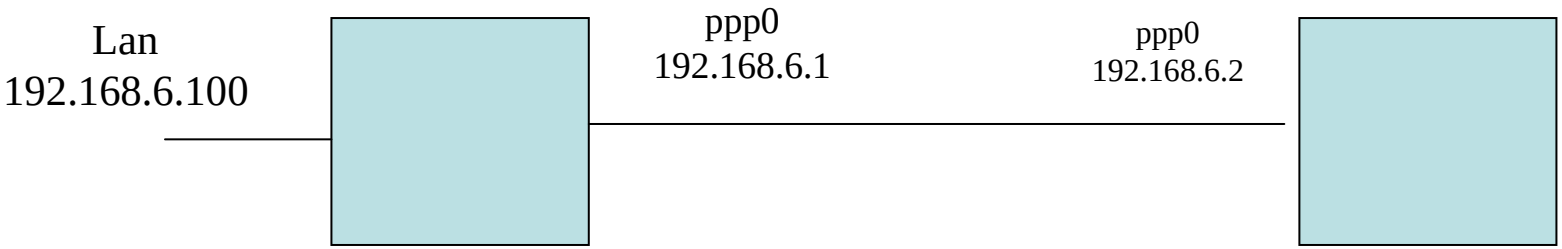
- 0x7e -> 0x7d(esc) 0x5e
- 0x7d -> 0x7d 0x5d
- все управляющие символы (код <20) передаются: esc+ код символа с инверсией старшего бита

0x01-> 0x7d,0x21

Установка соединения

В ОС Linux за установку ppp соединенния отвечает демон pppd.
Возможна работа в двух режимах:

- пассивный- демон ждет соединения



активный- демон пытается установить соединение.

- Для установки соединения потребуется пара IP фиктивных адресов, принадлежащих интерфейсам.
- Возможна организация соединения, защищенного паролем, при помощи pap-chap аутентификации
- Возможна аутентификация при помощи RADIUS сервера
- Возможен режим компрессии содержимого пакета «на лету»
- Заложена опция «Proxyarp» для адресов «за инетрфейсом»

Пример запуска pppd для последовательного порта

pppd tty_name [speed] [options]

Клиент:

#pppd /dev/ttyS0 38400 ctrscts defaultroute -detach

Сервер:

#pppd /dev/ttyS0 38400 192.168.6.1:192.168.6.2 proxyarp passive auth

Скрипты мониторинга соединения.

После установки соединения будет запущен скрипт /etc/ppp/ip-up с параметрами

/etc/ppp/ip-up iface device speed local_adr remote_adr

В этом скрипте можно, например, задавать команды маршрутизации.

После завершения соединения будет запущен скрипт

/etc/ppp/ip_down iface

PPP

PPP обеспечивает преимущества в сравнении с slip:

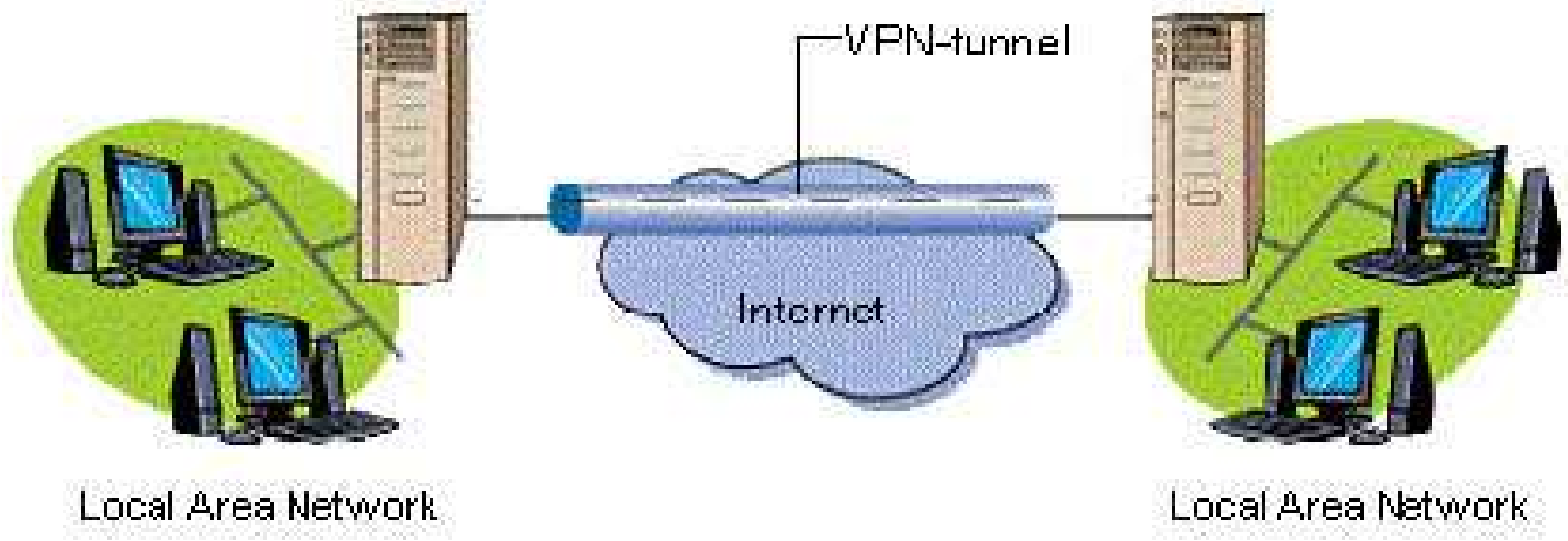
- Поддержку нескольких протоколов для одной последовательной линии
- Циклический контроль избыточности для каждого кадра
- Динамическое определение адресов партнера с использованием протокола NCP для IP
- Сжатие IP заголовков (Van Jacobson)
- Протокол управления каналом LCP для ведения переговоров об установлении тех или иных опций канального уровня
- обеспечивается непрерывный контроль наличия соединения.
- поддерживаются протоколы аутентификации PAP CHAP

Туннели

- Постепенно коммутируемые линии связи теряют свою популярность, хотя сами последовательные каналы еще долго останутся актуальными. Но протокол PPP не собирается сдавать позиции благодаря распространению туннелей и протоколов VPN.

Виртуальные частные сети - VPN

VPN представляет собой объединение отдельных машин или локальных сетей в единую виртуальную сеть, которая обеспечивает целостность и безопасность передаваемых данных. Она обладает свойствами выделенной частной сети и позволяет передавать данные через промежуточную сеть например Internet. VPN позволяет отказаться от использования выделенных линий.



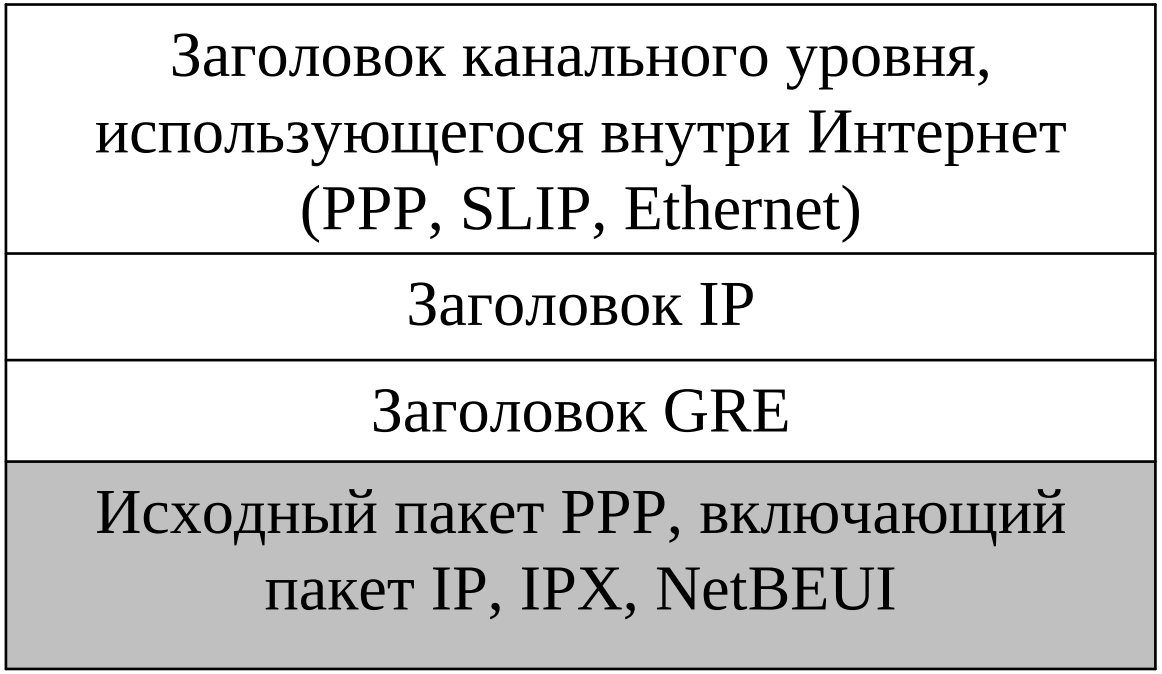
Виртуальные частные сети - VPN

Существует множество различных решений для построения виртуальных частных сетей. Наиболее известные и широко используемые это:

- **PPTP** (Point-to-Point Tunneling Protocol). Этот протокол стал достаточно популярен благодаря его включению в операционные системы фирмы Microsoft.
- **L2TP** (Layer 2 Tunneling Protocol). Этот протокол является объединением двух протоколов PPTP и L2F.
- **PPPoE** (PPP over Ethernet) — разработка RedBack Networks, RouterWare, UUNET и другие.
- **IPSec** (Internet Protocol Security) — официальный стандарт Интернет.

Виртуальные частные сети: PPTP

Пакеты, переносящие пользовательские данные в рамках сессии PPTP, инкапсулируются непосредственно в пакеты IP с помощью заголовка Generic Routing Protocol (GRE). Пакет, полученный в результате инкапсуляции, показан на рисунке:



Применение туннелей

- Доступ к частным сетям через публичные сети
- Соединение сегментов сетей через публичные сети
- Управление и мониторинг объектом к которому предъявляются повышенные требования безопасности (например, банкомат)
- Создание «контуров безопасности»

Оконечные устройства для туннелей

- Сервер с сетевой операционной системой (Win,Linux)
- Аппаратный маршрутизатор (DIR)
- Аппаратный фильтр пакетов (DFL)
- Модем DSL линии связи

Управление коммутируемыми устройствами

модемы аналоговые,
модемы GSM

Система команд коммутируемых модемов.

Прокладка или аренда специальных линий связи на большие расстояния –достаточно дорогостоящий путь построения сети. В случае, когда соединение используется не постоянно- целесообразно воспользоваться обычной телефонной сетью при помощи коммутируемого модема.

Подавляющее большинство коммутируемых и GSM модемов поддерживают разработанный фирмой Hayes набор AT-команд управления. AT-команды являются общепризнанным стандартом, принятым не только для коммутируемых модемов.

Режимы работы модема

- Режим команд. Модем воспринимает АТ команды, используется для настройки. Возможен как при установленном, так и при установленном соединениях
- Режим данных. Все байты интерпретируются как данные. Возможен только при установленном соединении
- *Режим выделенной линии. П*ытается установить соединение самостоятельно.*(Может не поддерживаться).*

Переход между режимами

команды-> данные

- при удавшейся попытке установления связи с другим модемом ;
- при выполнении модемом процедур самотестирования .

данные-> команды

- после неудачной попытки связаться с удаленным модемом ;
- при потере несущей в течение передачи данных ;
- при поступлении модему от компьютера команды в момент; набора модемом номера ;
- при передаче от компьютера модему специальной Escape-последовательности.

AT - команды

Правила:

- Все команды(кроме a/ +++) начинаются с префикса AT и заканчиваются <CRLF>

  AT команда1 команда2 команда3 <CRLF>
- Команда A/ повторяет прошлую команду
- +++ - Команда ESC-последовательность.
- «Пустая» AT команда вызывает ответ “OK”

некоторые AT команды

- ATZ[n] n-число. –сброс модема с загрузкой профиля.
- ATA – автоответ – снимает трубку.
- ATDP номер (ATDT номер) – набор номера
  ATDP 18333

Правила набора номера:

- ,-пауза
- @ ожидает 5й секундной тишины
- ! кладет трубку, снова снимает трубку
- R после набора номера переводит в режим автоответа (call-back)
- W ожидает длинный гудок (8W180)
- ^ Включает вызывающий тон 1300 Гц

ATDP 8**w**3832110101@T99223456

некоторые AT команды

- H0 H1 – снять трубку/положить трубку
- Ln – громкость динамика
- Mn – режим работы динамика
- Sn? – просмотр регистра
- Sn=v – изменение значений регистров
- V0 V1 –числовой/текстовый ответ модема
- Xn – методы набора номера
- &Fn – восстановление заводской установки
- &Dn – обработка терминала DTR
- &Mn – Синхронный/асинхронный режим работы
- &V – просмотр сохраненного профиля
- &Wn – запись профиля.
- + - расширенные команды (вне стандарта Hayes)

S-регистры

В модеме имеется набор S-регистров, позволяющих управлять различными коммуникационными параметрами , получать информацию о состоянии модема и выполнять тестовые функции. Значения регистров сохраняются в энергонезависимой памяти модема.

ATS0? – получить значение

ATS0=1 – установить значение

В некоторых моделях разрешается манипулировать битами:

ATS60.1? получить значение бит 1 регистра 60

ATS60.1=0 установить в 0 значение бит1 регистра 60

Некоторые S-регистры

S0 – количество звонков до автоответа(0-255)

S1- Счетчик звонков(только чтение)

S7 – время ожидания несущей (с)

S8- длительность паузы , (с)

S9-время реакции на несущую (0.1с)

S10-время ожидания несущей в случае потери (0.1с)

S30-максимальный таймаут(0.1с)

S37-Ограничение скорости соединения

Передача данных в GSM-сетях

В связи с широким распространением радио покрытия сотовых компаний стало возможным осуществлять информационный обмен между различными узлами и устройствами автоматики при помощи GSM-сетей.

Преимущества:

- Нет необходимости в ЛС
- Цифровая передача данных
- Сравнительно низкая стоимость организации канала

Недостатки:

- Сравнительно низкая скорость, зависящая от загрузки каналов.
- Возможный отказ в обслуживании в случае перегрузки сети.
- Сравнительно высокое энергопотреблние;
- Ограниченное радиопокрытие, наличие “мертвых зон”.

Пути обмена данными

Организация цифрового обмена данными возможна через:

- Модемный канал (выделяется один голосовой канал). 9.6 Кбит/с
- GPRS (пакетная передача сообщений) до 171.2 Кбит/с
- SMS (служба коротких сообщений).

GPRS

Для высокоскоростной передачи данных посредством существующих GSM-сетей и была разработана GPRS (General Packet Radio Service - услуга пакетной передачи данных по радиоканалу). GPRS является как бы надстройкой над стандартом GSM, неким дополнением,расширяющим функциональные возможности сотовой сети.

Обеспечивается одновременая передача по нескольким подчастотным каналам, что теоретически позволяет передавать данные до 171,2 Кбит/с

Реально сотовые операторы выставляют собственное ограничение (обычно менее 64K), причем скорость заметно падает в часы пиковой загрузки.

GPRS

Любой GPRS-модем поддерживает все базовые Hayes –совместимые команды, с помощью них же и устанавливается соединение,аналогично обычному коммутируемому модему. После набора номера требуется установить ppp-сессию.

ПРИМЕР Для NWGSM:

AT+CGDCONT=1,”ip”,”internet.sib”

OK

ATDT*99***1#

CONNECT

~я}#A!}!}#} }=}!}$}'P}"}&} }*} } }'}"}(}"}%}&}2ŭўЯ}#}%B#}%1e~~я}#A!}!}#} }=}!}$}'P}"}&} }} } }'}"}(}"}%}&}2ŭўЯ}#}%B#}%1e~~я}*

GPRS

Обычно пользователи услуг GPRS выходят в сеть через систему VPN (Virtual Private Network) и имеют "односторонний" доступ к Интернету. Это значит, что вы легко сможете попасть на любой сервер Сети, а до вашего компьютера "достучаться" будет невозможно, что накладывает свои ограничения в применении в устройствах автоматики.

При применении услуги «IP адрес» это становится возможным.

SMS

Служба коротких сообщений SMS - услуга, предоставляемая операторами цифровых стандартов мобильной связи (GSM, CDMA, DAMPS), заключающаяся в отправке коротких текстовых (и не только) сообщений на мобильный телефон .

При использовании SMS не устанавливается прямое соединение с каким-либо абонентом, поэтому не оплачивается время на линии.

Сообщения посылается службе SMS, которая отвечает за их доставку. Если абонент-адресат доступен SMS, сообщение доставляется получателю или хранится до тех пор, пока не будет доставлено или пока не истечет срок его доставки.

SMS

Как определено в стандарте ETSI (GSM 03.40 и GSM 03.38), длина SMS сообщения не может быть больше 160 символов, где каждый символ представлен только 7 битами (7-битный GSM Default алфавит).

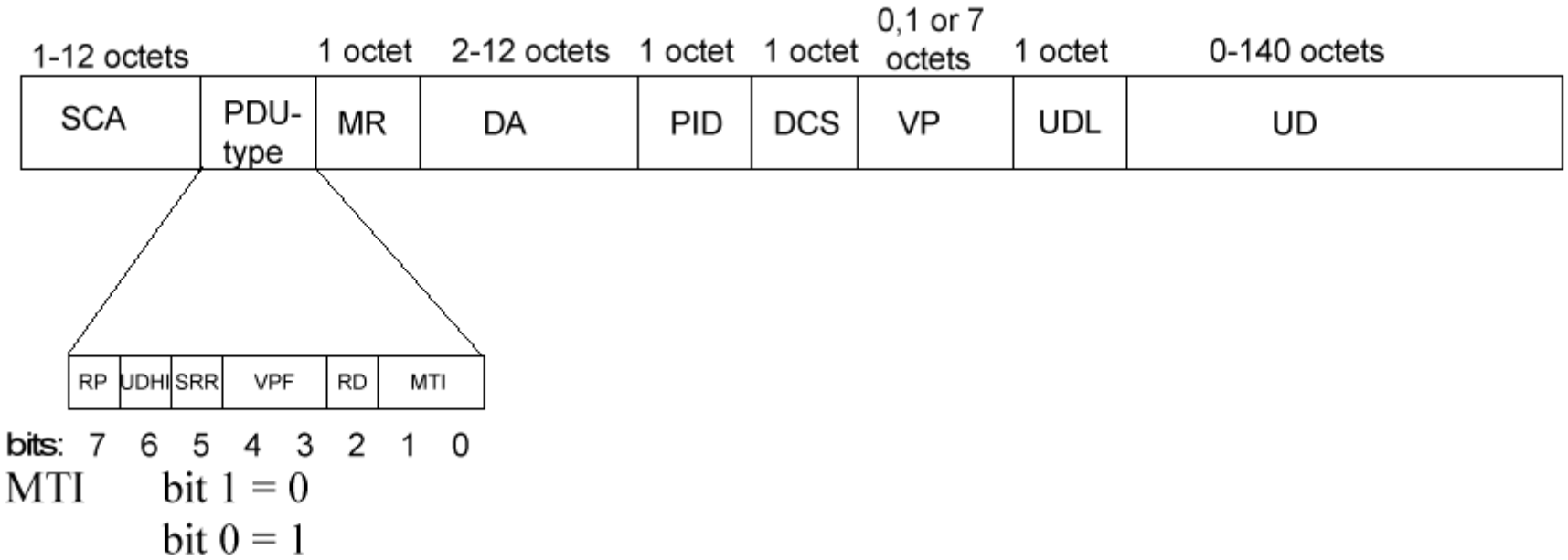
Восьмибитная кодировка (максимальная длина сообщений 140 символов) обычно предназначается для передачи не текстовых сообщений, таких как изображения, мелодии, различные OTA сервисы и редко поддерживается операторами связи.

16-битная кодировка (максимальная длина 70 символов) используются для сообщений в Unicode (UCS2) кодировке.

SMS-PDU

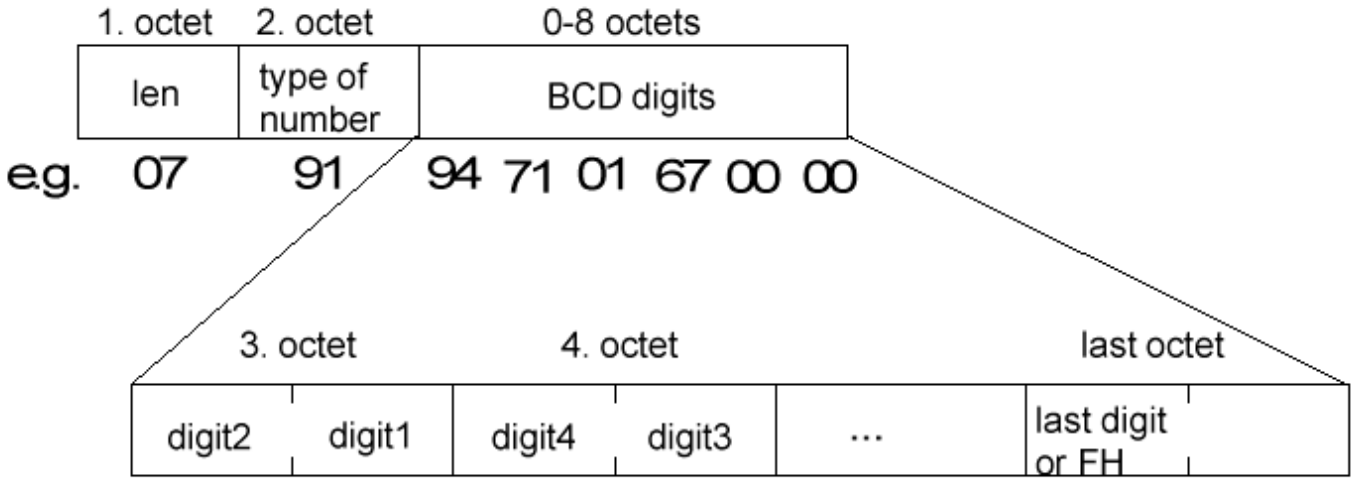
Данные упаковываются в пакет PDU,содержащий кроме тела сообщения служебную информацию. PDU записывается в ячейку на SIM карте, затем передается на СМСЦ (либо передается напрямую). Все связные операции выполняются по принципу транзакций, действие считается завершенным, если на него получено подтверждение.

Формат PDU



Sca-номер сервис-центра
PDU-type – тип сообщения
MR – ИД сообщения для генерации подтверждения
DA-адрес назначения
PID- Тип сообщения
DCS – кодировка
VP – время жизни
UDL-длина сообщения
UD- данные.

Поле номера



len – длина номера
type – тип номера:
81H – национальный
91H- международный

AT-команды мобильных устройств

Во соответствии со стандартом GSM все мобильные устройства, оборудованные модемом и имеющие возможность подключения по последовательному протоколу совместимы с набором AT-команд.

Базовый набор команд совпадает для устройств всех производителей, что обеспечивает переносимость ПО.

Расширенный набор команд как правило совместим с устройствами одного производителя.

Некоторые AT-команды (общие для многих производителей)

AT+CBC – остаточный заряд батарей +

AT+CGMI – Производитель оборудования +

AT+CMGS Send an SMS

AT+CMGW Write an SMS to the SMS memory

AT+CMSS Send an SMS from the SMS memory

—

AT+CPMS=? — выводит список возможных Memory Storage'ов для каждой группы

AT+CPMS? — выдает выбраные в каждой группе Memory Storage'ы кол-во сообщений в них и вместимость.

AT+CGSN – серийный номер устройсва

Пример 1 отправки СМС

Отправка СМС без записи в ячейку (test) - платформонезависмо

Номер центра:79139869990

Sent: AT+CMGS=18

0791**9731899699F0**11000B81**9731093427F9**0000A804**F4F29C0E**

>+CMGS: 158

Пример2

Отправка СМС с записью в ячейку. (Команды Siemens)

AT+CMGW=18

Код PDU сообщения

>

+CMGW: 4

OK

AT+CMSS=2

>

+CMGS: 158

Подключение к СМСЦ

Для непосредственного подключения к СМСЦ оператора применяется протокол SMPP. В настоящее время- используется протокол SMPPv3.4.

Существует три типа подключения:

- bind receiver - приемник
- bind transmitter - передатчик
- bind transiver – приемопередатчик

SMPP

После заключения договора, оператор выделяет IP адрес и порт, по которому надо производить соединение, а также передает следующие идентифицирующие Вашего SMPP-клиента параметры: System-ID, System-Password, System-Type.

SMPP

- Отправка сообщения осуществляется путем соединения с портом по протоколу TCP и обмена данными PDU.
- Разработаны библиотеки для C и Perl для упрощения работы с СМСЦ. Самая известная – свободная библиотека для PERL SMPP.pm

Пример подключения к используя модуль Net::SMPP.pm

Чтобы подключиться к СМСЦ для отправки коротких сообщений необходимо послать на СМСЦ команду *bind_transmitter (0x00000002)*:

```
($smpp,$err)= Net::SMPP->new_transmitter(
        "192.168.100.100",
        port=>"99999",
        system_id =>'System_Id',
        password => 'System_Password',
        system_type=>'System_Type'
      ) or die;
print"Connect-transmitter. Status=",$err->{status},"*\n";
```

Если получен ответ ОК – можно пытаться передавать сообщения

SMPP – отправка сообщений

```
$resp = $smpp_t->submit_sm(
        source_addr_ton => 0x01,
        source_addr_npi => 0x01,
        source_addr => '99999',
        dest_addr_ton => 0x01,
        dest_addr_npi => 0x01,
        destination_addr => '99999',
        data_coding => 0x04,
        short_message=> 'Тестовая СМС!!!'
        esm_class => 0x00,);
```

SMPP-прием сообщений

"Слушаем" диапазон и при поступлении СМС читаем PDU:

```
$pdu = $smpp_r->read_pdu();
```

Полученное сообщение имеет стандартный формат PDU.

Обязательно посылаем СМСЦ ответ об успешном получении СМС *deliver_sm_resp (0x80000005)*:

```
$smpp_r->deliver_sm_resp(seq=>$pdu->{seq},message_id=>"\x00");
```

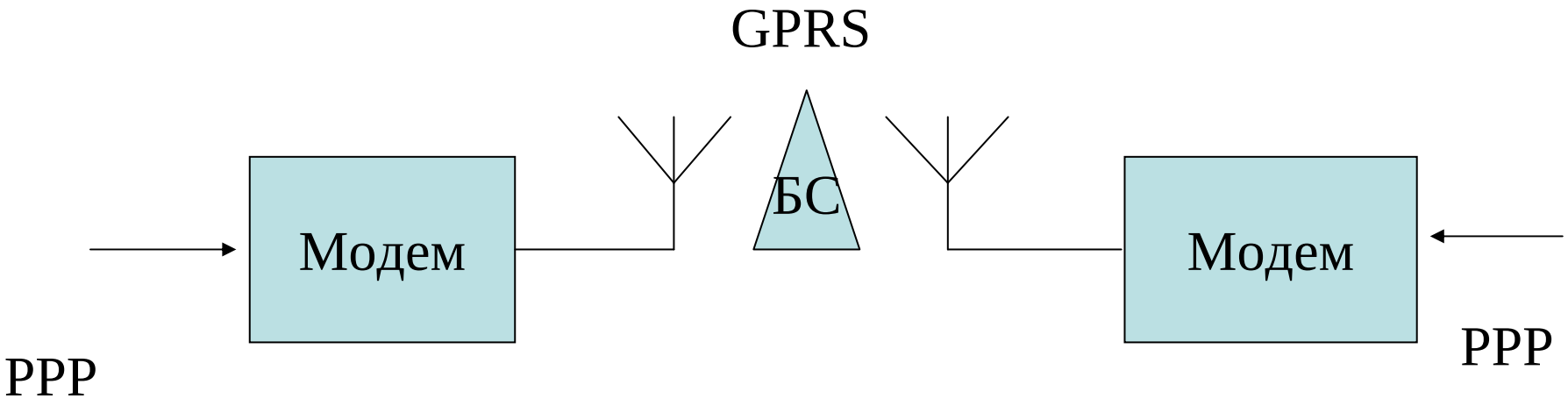
И отключаемся по окончании работы:

```
$resp = $smpp_r->unbind();
```

Кодировки сообщений

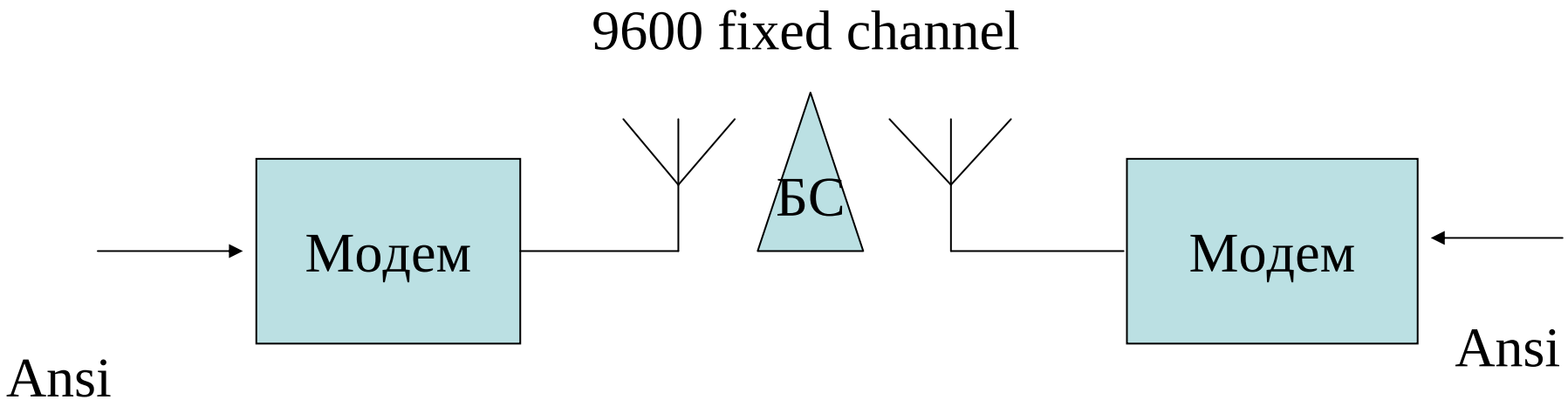
- 7 bit (по умолчанию) 160 символов
- 8 bit (поддерживается не всеми операторами и терминалами) 140 символов
- UCS2 Кодировка (16 бит на символ) – 70 символов.

Методы обмена данными через GSM-сети- GPRS



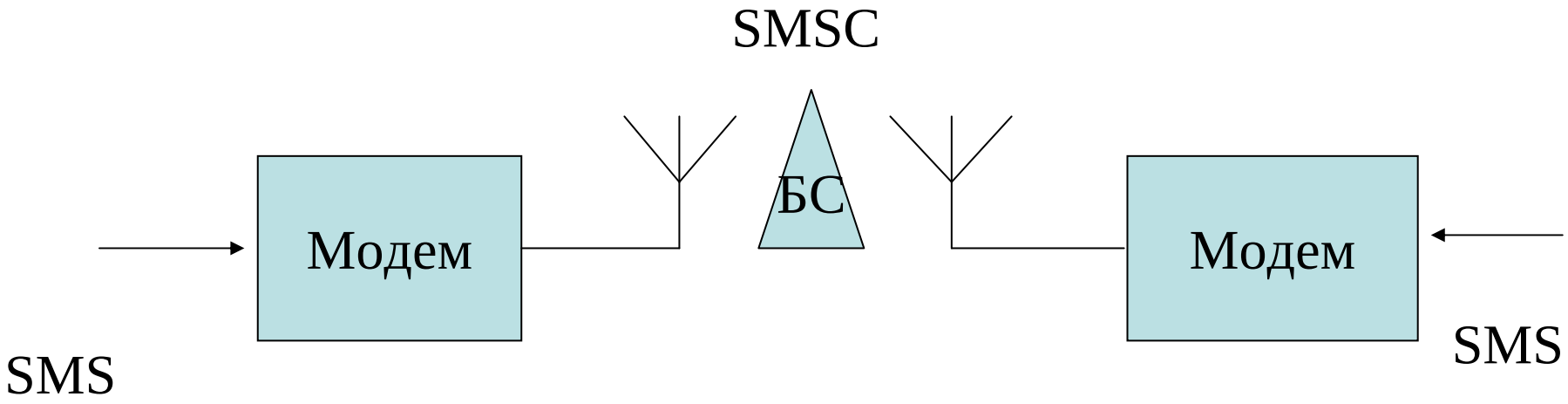
- Стоимость определяется трафиком- «тишина», не оплачивается
- Связь возможна только по PPP
- Скорость обмена не гарантируется и трудно прогнозируема

Методы обмена данными через GSM-сети- ANSI



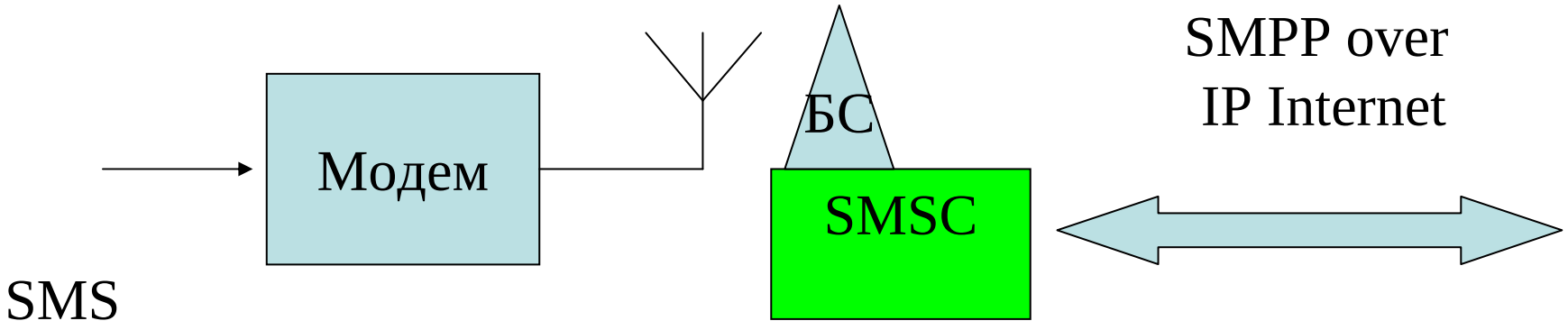
- Стоимость определяется временем соединения, «тишина» **оплачивается**
- Связь возможна и в простом терминальном режиме
- Скорость обмена **фиксирована и гарантирована 9600**
- Длительность соединения может быть ограничена оператором

Методы обмена данными через GSM-сети SMS



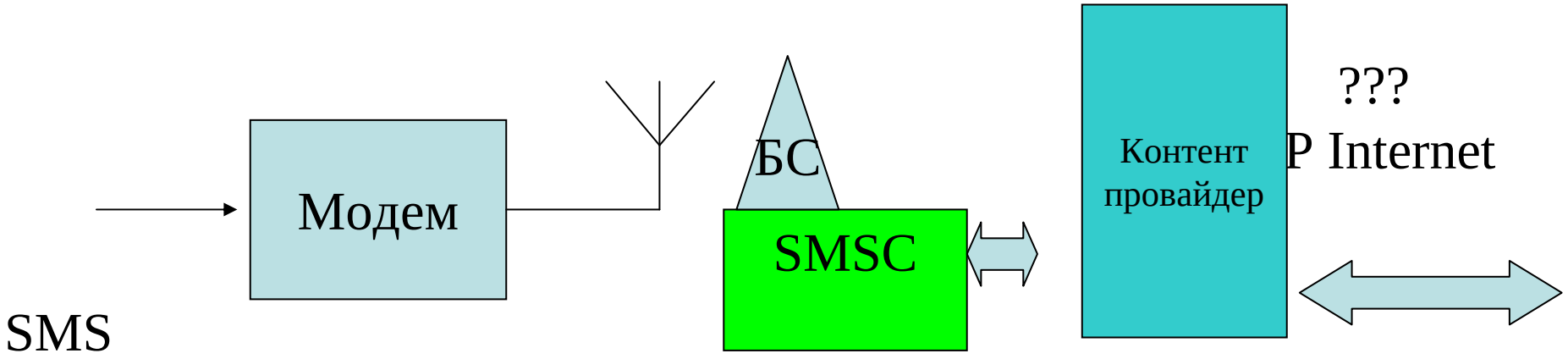
- Стоимость определяется количеством сеансов SMS
- Длина SMS 160 символов
- Скорость доставки данных зависит от SMSC
- Достоверность доставки подтверждается квитанцией

Методы обмена данными через GSM-сети SMPP



- Требует заключения договора с оператором.
- Стоимость обмена данными определяется правилами оператора для контент-провайдера и НЕ ОБЯЗАТЕЛЬНО РАВНА стоимости 1 смс

Методы обмена данными через GSM-сети SMPP



- Требует заключения договора с контент-провайдером.
- Вероятнее всего, ваша услуга будет в командах единого номера.

Лекция

Лекция 13-14

Применение криптографии

Построение защищенной беспроводной сети

WEP-WPA-WPA2

Обмен электронной почтой.

Проблематика

- Неправильная настройка беспроводных соединений– серьезная опасность для ЛВС.
- «Открытость» дает возможность для хакерства и перехвата «проездом»

Общая идеология построения защит ИС

- Стоимость алгоритма защиты не должна превышать стоимость защищаемой информации.
- Время, необходимое для вскрытия ключа или пароля должно превышать период актуальности защищаемой информации.
- Время жизни сеансового ключа не должно превышать время, необходимое для вскрытия ключа.

История стандартов защиты беспроводных сетей

- до 1997 – защита по SSID(имя сети)
- 1997- IEEE 802.11 (WEP)
- 2001- WEP опубликован алгоритм взлома
- 2002 –IEEE 802.1X (dynamic WEP)
- 2003- IEEE 802.1g (WPA)
- 2004 IEEE 802.1i (WPA2) (начало сертификации, широкое внедрение 2006/2007 актуален нв)

WEP IEEE 802.11

- Построен на базе RC4(вариация DES).
- Имеет один статический ключ.
- Ключ задается вручную.
- Не предусмотрена аутентификация пользователя или устройства.
- В стандарте предусмотрены ключи длиной 64,128,(256) бит.

WEP IEEE 802.11

- ключ=
  - 24 бит initialization vector (случайный)
  - 40 или 104 бит секретный ключ
- Процедура:
  - IC(integrity check) вычисляет контрольную сумму
  - RC4 вычисляет последовательность ключей из PASS и IV
  - передающая сторона шифрует данные и контр. сумму
  - IV передается открытым текстом
  - приемная сторона дешифрирует данные на основе известного PASS и полученного IV

WEP - уязвимость

- RC4 не стоек к взлому «грубой силой»
- Длина IV составляет 24 бит. До повтора IV необходимо перехватить всего 24 ГБ пакетов (менее 1 часа на скорости 54Мб)
- В Сети имеется множество ПО для перехвата (Линукс)
- Ключ статический- вычисление ключа: сеть взломана.
- 256 бит шифрование не увеличивает длину IV, следовательно не повышает безопасности.

Безопасность WEP, пути:

- Устаревшее оборудование, поддерживающее 802.1b не обеспечивает другой защиты кроме WEP(встречается все реже)
- Единственный путь- организация VPN-туннелей. Недостаток- необходимо дополнительное ПО.

802.1X + WEP (Dynamic WEP)

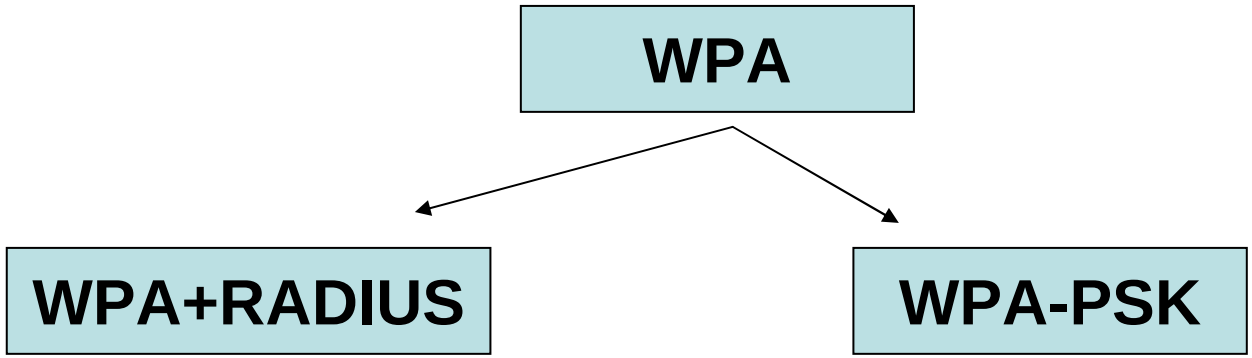
- EAP (Extensible Authentication Protocol) – протокол расширенной аутентификации пользователей или удаленных устройств (RADIUS) сервер.
- TLS (Transport Layer Security) – протокол защиты транспортного уровня, он обеспечивает целостность передачи данных между сервером и клиентом, а так же их взаимную аутентификацию;

RADIUS (Remote Authentication Dial-In User Server) – сервер аутентификации (проверки подлинности) удаленных клиентов. Он и обеспечивает аутентификацию пользователей.

Протокол WPA 802.1g

WPA (Wi-Fi protected access), анонсирован 31 октября 2002 г

Протокол действует в настоящее время и обязателен для всех сертифицируемых беспроводных устройств.



Формула WPA

- **WPA = 802.1X + EAP + TKIP + MIC**

- **EAP=**Extensible Authentication Protocol (Radius, или AP)
- **TKIP**=TKIP (Temporal Key Integrity Protocol) – реализация динамических ключей шифрования, плюс к этому, каждое устройство в сети так же получает свой Master-ключ (который тоже время от времени меняется). Ключи шифрования имеют длину 128 бит и генерируются по сложному алгоритму, а общее кол-во возможных вариантов ключей достигает сотни миллиардов, а меняются они очень часто. Тем не менее, используемый алгоритм шифрования – по-прежнему RC4.
- **MIC** (Message Integrity Check) – протокол проверки целостности пакетов. Протокол позволяет отбрасывать пакеты, которые были «вставлены» в канал третьим лицом, т.е. ушли не от валидного отправителя.

WPA-EAP

- Шаг 1. Клиент устанавливает физическое соединение с точкой доступа (AP)
- Шаг 2. AP блокирует сетевые соединения до тех пор, пока клиент не пройдет аутентификацию
- Шаг 3. Клиент посылает запрос к серверу аутентификации
  - если аутентификация не осуществлена - клиент остается «блокированным»
  - если аутентификация пройдена – процесс продолжается

- Шаг 4. Сервер аутентификации автоматически рассылает ключи шифрования точке доступа и клиенту
- Шаг 5. Доступ к сети разрешен, начинается обмен данными

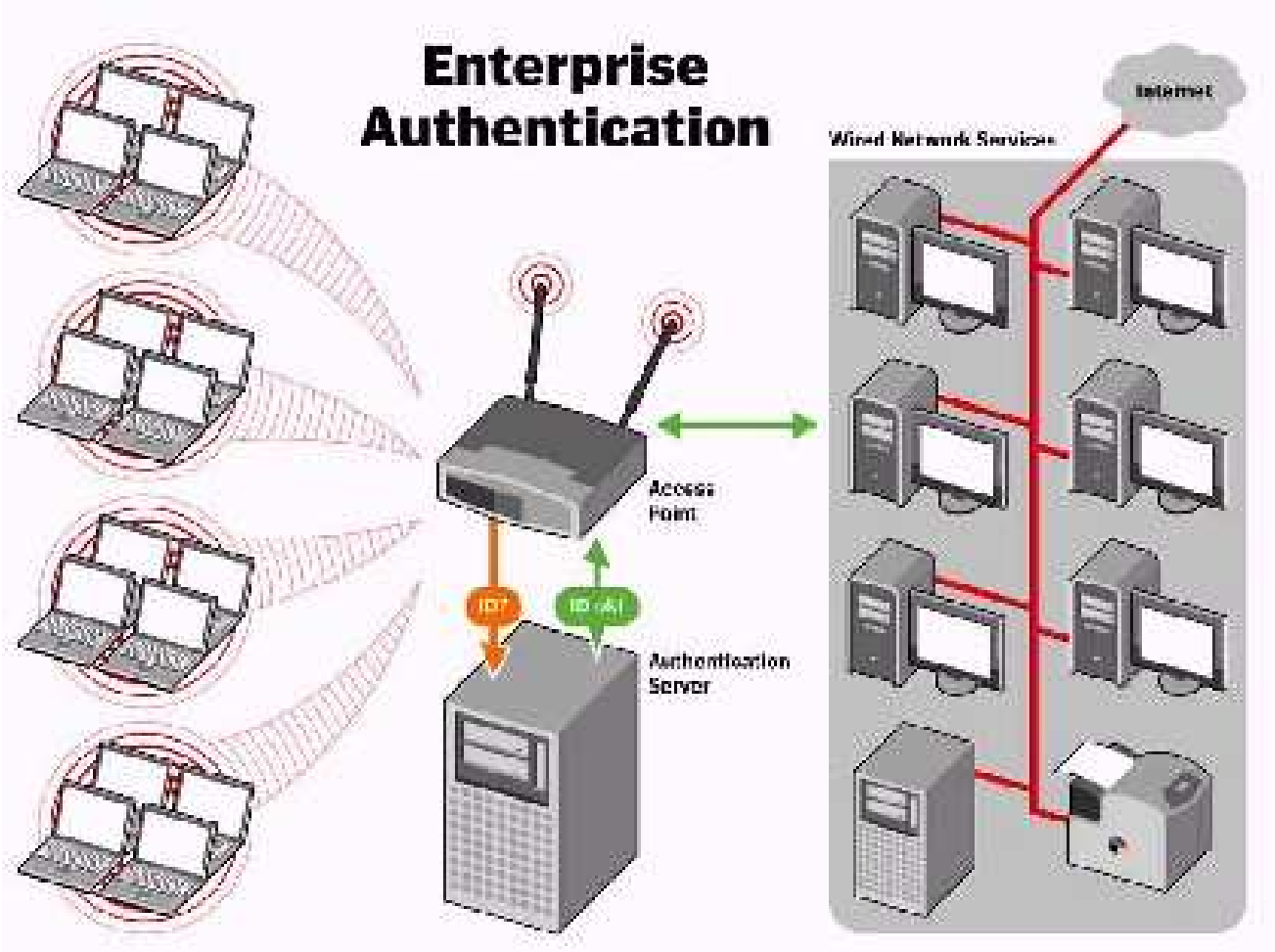


Иллюстрация Networld + Interop April 29, 2003 David Cohen Chair, Security Committee Wi-Fi Alliance

WPA-PSK

- стандарт имеет упрощённый режим, который не требует использования сложных механизмов. Этот режим называется **Pre-Shared Key** (WPA-PSK) - при его использовании необходимо ввести **один пароль** на каждый узел беспроводной сети (точки доступа, беспроводные маршрутизаторы, клиентские адаптеры, мосты). До тех пор, пока пароли совпадают, клиенту будет разрешён доступ в сеть.

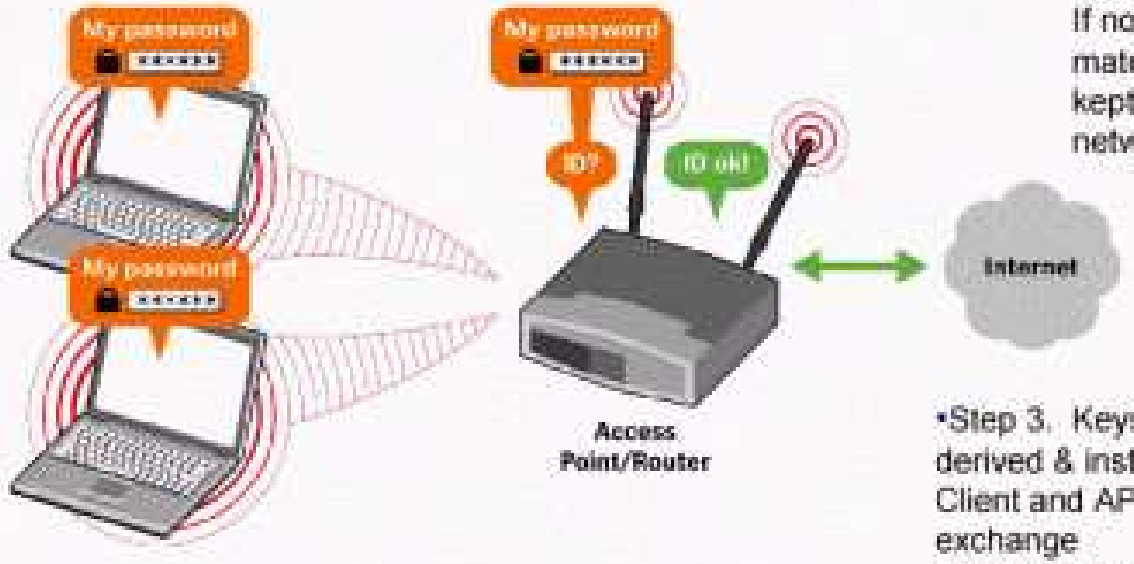
WPA-PSK отличие от WEP

- Применена аутентификация, роль «проверяющего» берет на себя точка доступа
- Не применяется механизм «нестойкого» IV
- Криптование идентично полной версии WPA

How WPA Works - SOHO

Step 1. Enter matching passwords into AP and clients.

SOHO Authentication



Step 2. AP checks client's password.
If a match, client joins network.
If not a match, client kept off network.

•Step 3. Keys derived & installed. Client and AP exchange encrypted data.

WPA-компоненты

- ***Клиент.*** (В роли клиента выступает Supplicant - программа на клиентском компьютере управляющая процессом аутентификации)
- ***Аутентификатор.*** (Это точка доступа, которая выполняет функции посредника между клиентом и сервером аутентификации (аутентификатором может быть и проводной коммутатор, т.к. 802.11 используется и в проводных сетях).
- ***Сервер аутентификации.*** (В роли сервера аутентификации выступает RADIUS-сервер).

WPA2

- задействует новый метод шифрования (получивший название CCMP — Counter-Mode with CBC-MAC Protocol), основанный на более мощном, чем RC4, алгоритме шифрования **AES** (Advanced Encryption Standard).
- WPA2 реализуется на 2м уровне OSI (WPA на 3м)

WPA2

- Подобно WPA существует два режима: Enterprise и PreShared Key — PSK –WPA2-PSK

- В режиме WPA2-Personal из введенной открытым текстом парольной фразы генерируется **256-разрядный** ключ, иногда именуемый предварительно распределяемым ключом (PreShared Key — PSK).
- Ключ PSK, а также идентификатор SSID (Service Set Identifier) и длина последнего вместе образуют математический базис для формирования главного парного ключа (Pairwise Master Key — PMK)

WPA2

- WPA2 имеется три типа ключей PTK:
  - ключ подтверждения ключа (Key Confirmation Key — KCK), применяющийся для проверки целостности кадра
  - ключ шифрования ключа (Key Encryption Key — KEK), используемый для шифрования группового временного ключа (Group Transient Key — GTK)
  - и временные ключи (Temporal Keys — TK) — для шифрования трафика.

WPA2 - роуминг

- Т.к. процессы перенесены на 2й уровень, то становится возможным реализация защищенного роумингового соединения- не было в WPA
- Предварительная аутентификация позволяет мобильному клиенту аутентифицироваться на другой, расположенной поблизости точке доступа, оставаясь “привязанным” к своей первичной точке доступа. При применении кэширования PMK клиенту, вернувшемуся с обслуживаемой роумингом территории “домой”, не нужно выполнять полную процедуру повторной аутентификации 802.1X.

AES

- В основе стандарта WPA2 лежит метод шифрования AES, пришедший на смену стандартам DES и 3DES в качестве отраслевого стандарта де-факто. Требующий большого объема вычислений, стандарт AES нуждается в аппаратной поддержке, которая не всегда имеется в старом оборудовании БЛВС .
- В процессе шифрования используется 128и разрядный вектор инициализации. Повтор вектора в рамках действующего ключа невозможен. Вектор передается также шифрованным.

Summary

	WEP	WPA	WPA2
Криптование	слабое,легко вскрывается	исправлены недостатки WEP	Перенесено на 2й уровень
Ключ,длина	40 +24 бит 104+24 бит.	128 бит Шифрование RC4	256/128/128 бит Шифрование AES
Ключ,распределение	Статический ключ. Один и тот же ключ для всей сети	динамическое распределение ключей: вплоть до один ключ на пакет	Три ключа для широковещания, распределения ключей, трафика
Ключ,управление	Ключ вводится вручную для каждого устройства	Ключ распределяется автоматически	Ключ распределяется автоматически, криптуется отдельным ключом
аутентификация	слабая, на основе WEP ключа	802.1X and EAP	802.1X and EAP

Итог:

- Никогда не используйте «настройки» по умолчанию, даже когда вы тестируете Ваше соединение.
- Старайтесь использовать WPA или WPA2 во всех случаях.
- Для обеспечения высокого уровня защиты используйте WPA(2)+ сервер аутентификации.

ОБМЕН ЭЛЕКТРОННОЙ ПОЧТОЙ

- Обмен электронной почтой.
- Формат заголовка и тела сообщения.
- Расширение MIME.
- smtp, pop3.
- Безопасность и борьба со спамом.

Электронная почта.

В сущности, электронное письмо – это обычный текстовый файл. Но, чтобы почтовые системы всего мира могли разобраться, кому и куда направить письмо, этот текст должен составляться по определенным правилам.

Любое электронное письмо состоит из двух частей:

- официальной (здесь указывается кто, кому, куда, когда послал письмо);
- неофициальной (вот это собственно та информация, которую один человек хочет сообщить другому человеку);

Части разделяются пустой строкой.

Формат заголовка

Формат почтового сообщения Internet определен в документе RFC-822 (Standard for ARPA Internet Text Message).

Заголовок всегда находится перед телом сообщения и отделен от него пустой строкой. RFC-822 регламентирует содержание заголовка сообщения. Заголовок состоит из полей. Поля состоят из имени поля и содержания поля. Имя поля отделено от содержания символом ":"

Электронная почта

Сеть Internet объединяет множество различных компьютеров, работающих в различных операционных системах. В каждой операционной системе есть своя почтовая служба, которая по-особому обрабатывает заголовок письма.

Чаще всего в качестве почтового сервера используется Unix-подобная ОС.

В Unix за обмен электронной почтой отвечает демон sendmail или postfix, но последнее время все чаще встречаются альтернативные демоны.

Windows используется в корпоративных сетях как сервер Exchange

Простейший заголовок

Но, чтобы пользователи сети Internet могли свободно общаться друг с другом, в заголовке письма, в соответствии с RFC, обязательно должны присутствовать такие поля:

Date:

From:

To:

Date: дата и время отправления письма; они записываются в стандартном формате - день недели, день, месяц, год (2 цифры), время, временная зона.

From: имя отправителя и его обратный адрес.

To: адрес получателя.

Если в полях адресов содержится какая-либо дополнительна информация, адрес заключается в угловые скобки.

To: “Real Name” <real@ngs.ru>

Пример

Date: Sat, 30 Apr 2005 08:50:01 +0700

To: “Dest” mail@ngs.ru, “Dest2” mail2@ngs.ru

From: “from” <letter@ngs.ru>

Hello, world!!!

другие параметры: Message-Id

Message-Id: уникальный идентификатор сообщения, который компьютер-отправитель присвоит письму. Например: это набор цифр и букв и имя машины. Этот идентификатор можно использовать для

ссылок на письмо в канцелярском деле, как исходящий номер.

другие параметры: Received

Received: отметка о прохождении письма через машину (почтовый штемпель).

Может содержать:

- имя почтового компьютера, пославшего письмо (from домен);
- имя почтового компьютера, принявшего письмо (by домен);
- физический путь следования письма (via ...);
- название протокола передачи данных (with ...);
- номер принятого сообшения (id ...);
- для кого сообщение (for адрес);
- дату прохождения письма через машину.

.

Поля заголовка (необязательные)

Reply-To: Адрес для ответа - адрес отправителя. Это позволяет при ответе на данное письмо (reply) ввести адрес автоматически.

Resent-From: Адрес человека или программы, которые переслали вам сообщение, изначально пришедшее на их адрес

Sender: имя человека или программы, прислявшего вам это

письмо. В общем случае это не то же самое, что From: .

Например, для писем из конференции From: адрес автора письма, а Sender: адрес news-сервера.

Return-Path: <evaluations@vmware.com>
Received: from [172.16.0.1] (HELO intranet.ru)
  by mx3.intranet.ru (CommuniGate Pro SMTP 4.2.4)
  with ESMTP id 29413158 for fiery@ngs.ru; Thu, 28 Apr 2005 17:21:06 +0700
Received: from mailout1.vmware.com ([65.113.40.130] verified)
  by intranet.ru (CommuniGate Pro SMTP 4.2.4)
  with ESMTP id 240354204 for fiery@ngs.ru; Thu, 28 Apr 2005 17:21:00 +0700
Received: from mailhost1.vmware.com (mailhost1.vmware.com [10.16.12.135])
        by mailout1.vmware.com (Postfix) with ESMTP id D5F324525
        for <fiery@ngs.ru>; Thu, 28 Apr 2005 03:20:26 -0700 (PDT)
Received: from script.vmware.com (unknown [10.16.19.13])
        by mailhost1.vmware.com (Postfix) with ESMTP id 32D6E6FC324
        for <fiery@ngs.ru>; Thu, 28 Apr 2005 03:20:27 -0700 (PDT)
Received: (from evaluations@localhost)
        by script.vmware.com (8.11.6/8.11.6) id j3SAKAh25571;
        Thu, 28 Apr 2005 03:20:10 -0700
Date: Thu, 28 Apr 2005 03:20:10 -0700
Message-Id: <200504281020.j3SAKAh25571@script.vmware.com>
To: fiery@ngs.ru
From: wseval@vmware.com
Subject: Don't Delay! Time's Running Out On Your VMware Workstation Evaluation
Reply-To: wseval@vmware.com

Поля заголовка(необязательные)

Return-Receipt-To: Адрес, по которому нужно отослать "увеломление о доставке".

В большинстве случаев это адрес отправителя.

X-Mailer: Программа, с помощью которой было отправлено письмо. Например, dMail, ELM, TheBat

Subject: Тема письма.

Newsgroups: название конференции или нескольких конференций через запятую.

Expires: хранить в конференции до указанного числа.

Keywords: ключевые слова, по которым можно искать статью в конференции.

Lines: количество строк в письме

Return-Path: <postmaster@intranet.ru>
Received: from [212.17.5.144] (account <postmaster@intranet.ru>)
  by intranet.ru (CommuniGate Pro WebUser 3.4.8)
  with HTTP id 143833695 for <all@ngs.ru>; Thu, 30 Sep 2004
    11:28:03 +0700
From: <support@ngs.ru>
Subject: [НГС] Уведомление
To: all@ngs.ru
X-Mailer: CommuniGate Pro Web Mailer v.3.4.8
Date: Thu, 30 Sep 2004 11:28:03 +0700
Message-ID: <web-143833695@intranet.ru>
MIME-Version: 1.0
Content-Type: text/plain; charset="KOI8-R"
Content-Transfer-Encoding: 8bit

Уважаемые пользователи почтовой службы НГС!

Проблема поддержки национальных кодировок

Первоначально электронная почта была предназначена исключительно для передачи текстовых сообщений, содержащих ASCII символы. Если же требовалось передать двоичный файл или текст на языке отличном от английского, то возникала необходимость кодирования такого файла или текста символами ASCII. Далее, закодированное сообщение передавалось с помощью обычных средств электронной почты. Принимающая сторона (пользователь) должна быть извещена о способе кодирования и должным образом декодировать сообщение. Одна из таких кодировок - UUE.

Предшественник MIME-UUE

----------> <----------

Текст в кодировке win1251.

----------> <---------- 28байт

section 1 of uuencode 4.21 of file PRIMER.TXT by R.E.M.

begin 644 PRIMER.TXT

<DJ6JX>(@HB"JKJ2HX*ZBJJ4@=VEN,3(U,2X-"BX-

`

end

sum -r/size 28678/69 section (from "begin" to "end")

sum -r/size 64566/28 entire input file

----------> <---------- 69/230 байт

Mime

Разнородность сетей и обилие не стандартизированного ПО различных производителей зачастую не позволяло пользователям "понимать" друг друга. Причины проблем:

1. Разные клиенты работали с разными кодировками.
2. Не была определена структура размещения и идентификации типа закодированных данных. Чтобы понять, что собой представляет полученная информация, ее необходимо было "вынуть" из сообщения и декодировать

Mime

С ростом популярности E-mail и multimedia возникла необходимость в одном сообщении передавать данные различных типов:

- текстовую информацию на различных языках,
- графические изображения,
- видеопоследовательности,
- голосовые сообщения (аудиоинформацию),
- и просто, бинарные файлы;

Mime

Указанные проблемы были решены путем внедрения стандарта MIME (Multipurpose Internet Mail Extention, многоцелевое расшироение интернет почты)

Стандарт не заменяет, а расширяет существующий способ формирования электронных сообщений.

MIME - новый формат представления данных, представляющий почтовому клиенту гибкий интерфейс для работы с E-mail.

Mime

Для идентификации MIME-сообщений в заголовке сообщения должны присутствовать следующие поля:

Mime-Version: - версия MIME, например, 1.0 или 1.1

Content-Type: тип/подтип – тип сообщения.

Content-Transfer-Encoding: - используемый метод кодирования для передачи.

Другие поля конкретизируют какие-либо параметры и обязательными не являются.

Кодировки

Возможные значения: **base64, quoted-printable, 8bit, 7bit, binary.**

- 8 bit – сообщения,в которых включен 8й бит. Может не поддерживаться некоторыми национальными почтовыми службами. В этом случае 8 бит отбрасывается, данные могут быть потеряны.
- quoted-printable (RFC-1341) - кодирует любые не ASCII символы, позволяет передавать их вперемешку с первыми. Символ представляет собой последовательность из знака равно ("=") и шестнадцатиричного кода символа.

“Привет” -> =CF=F0=E8=E2=E5=F2 (windows-1251)

Кодировки

base64. Наиболее распространенная кодировка для передачи файлов.

- Битовый поток разбивается по 24 бита (по 3 байта), которые в свою очередь делятся на четыре части по 6 бит.
- Каждая такая часть кодируется одним из 64 ASCII символов (отсюда название - **base64**).

0->A, 1->B ,2->C ….. 62->+ , 63->-

Структура сообщения

Каждое mime сообщение может состоять из нескольких частей (multipart), разделенных «разделителем». При этом в заголовке указывается:

Content-type: multiaprt/подтип; bound="разделитель“

Подтип:

- mixed - все части обрабатываются последовательно;
- parallel - все части обрабатываются параллельно;
- alternative - интерпретация определяется клиентом;

Заголовок rfc822
Content-Type: multipart/alternative; boundary=“---- Next”

Текст rfc822
---- Next

Content-Type: text/plain; charset="koi8-r"
Content-Transfer-Encoding: base64

DQoNCi0tLS0tT3JpZ2luYWwgTWVzc2FnZS0tLS0tDQpGcm9tOiBqb2huQHZ
0YXUtYnNkLnBzdHUu
YWMucnUgW21haWx0bzpqb2huQHZ0YXUtYnNkLnBzdHUuYWMucnVdIA
0KU2VudDogTW9uZGF5LCBG
ZWJydWFyeSAwOCwgMTk5OSAxOjM3IFBNDQpTdWJqZWN0OiANCg0K
DQrN8yDt4Oru7eXpCPcICPYt
---- Next

Содержимое блоков

В каждом блоке указывается свой Content-type содержимого, определяющий как интерпретировать информацию в данном конкретном блоке. Возможные типы:

- **Текст(гипертекст)**

text/*подтип*; charset="кодировка"

подтип:

- text (txt)- простой текст;
- html (htm, html)- форматированный текст HTML;

Содержимое блоков

- **Изображение**

image/подтип; name="имя_файла"

подтип: jpg или jpeg, gif, bmp

- **Видео**

video/подтип; name="имя_файла"

подтип: **mpeg, x-msvideo** (avi), **quicktime** (qt)

Подтипы

- **Звук**

audio/подтип; name="имя_файла" Значения подтипа: ra, wav, basic (au)

- **Общий формат**

application/подтип; name="имя_файла"

Значения подтипа

  - octet-stream - бинарный файл (исполняемый или др. файл);
  - msword (doc)- файл MS Word;
  - x-compress (z), x-compressed (tgz), x-gzip (z), z-tar (gz), x-zip-compressed (zip)- файлы в сжатых архивах

Присоединение файла

Рассмотренные выше директивы в общем случае позволяют почтовой программе отображать данные внутри тела письма. Следующая директива предназначена для присоединения файла без права отображения содержимого в письме:

Content-Disposision attachment; filename="имя_файла" - прикрепленный файл, не интерпретируется почтовым клиентом. Для просмотра файл необходимо сохранить на локальном диске.

Тем не менее, такую возможность можно разрешить, указав директиву **inline** - если почтовый агент "знает" формат файла, то он будет отображен прямо в теле сообщения.

From user@email.net Wed Feb 10 16:15:17 1999
To: ivanov@nags.com
Subject: FW: please resend it, because i can't translate it at work
Date: Wed, 10 Feb 1999 09:30:57 +0300
MIME-Version: 1.0
X-Mailer: Internet Mail Service (5.5.1960.3)
Content-Type: multipart/alternative;
boundary="---- =_NextPart_001_01BE54E2.10C07C70"

This message is in MIME format. Since your mail reader does not understand
this format, some or all of this message may not be legible.

------ =_NextPart_001_01BE54E2.10C07C70
Content-Type: text/plain; charset="koi8-r"
Content-Transfer-Encoding: base64

DQoNCi0tLS0tT3JpZ2luYWwgTWVzc2FnZS0tLS0tDQpGcm9tOiBqb2huQHZ0YXUtYnNkLnBzdHUu
YWMucnUgW21haWx0bzpqb2huQHZ0YXUtYnNkLnBzdHUuYWMucnVdIA0KU2VudDogTW9uZGF5LCBG
ZWJydWFyeSAwOCwgMTk5OSAxOjM3IFBNDQpTdWJqZWN0OiANCg0KDQrN8yDt4Oru7eXpCPcICPYt

------ =_NextPart_001_01BE54E2.10C07C70
Content-Type: application/msword
Content-Disposition: inline
Content-Transfer-Encoding: base64

PCFET0NUWVBFIEhUTUwgUFVCTElDICItLy9XM0MvL0RURCBIVE1MIDMuMi8vRU4iPg0KPEhUTUw+
DQo8SEVBRD4NCjxNRVRBIEhUVFAtRVFVSVY9IkNvbnRlbnQtVHlwZSIgQ09OVEVOVD0idGV4dC9o
dG1sOyBjaGFyc2V0PWtvaTgtciI+DQo8TUVUQSBOQU1FPSJHZW5lcmF0b3IiIENPTlRFTlQ9Ik1T

------ =_NextPart_001_01BE54E2.10C07C70-

Передача электронной почты.

Мы рассмотрели формат заголовка и тела сообщения. Рассмотрим способы передачи сообщений в сети от узла к узлу.

По умолчанию сообщение передается на узел назначения напрямую, используя IP-адрес. Чтобы объявить о предоставлении услуги обмена почтой, сетевой узел должен обладать записью MX (Mail Exchanger) в базе данных DNS.

Маршрутизация почты в Интернет

Каждая запись MX содержит параметр приоритета (preference). Параметр приоритета- положительное целое число. Почтовый агент будет пытаться переслать сообщение на сервер MX,имеющий наименьшее значение приоритета, только в случае неудачи сообщение будет передано на узел с более высоким значением приоритета.

Пример записи:

green.foolbar.com IN MX 5 mailhub.foolbar.com

Протокол SMTP

SMTP (simple mail transport protocol) – основной протокол передачи электронной почты в сети Интернет. SMTP постепенно вытесняет использовавшийся ранее протокол (UUCP). Для работы SMTP создает соединение с сервером (порт 25), затем клиент с сервером обмениваются информаций пока соединение не будет закрыто.

Самой первой процедурой является открытие канала, самой последней-закрытие.

Команды smtp

Команды SMTP указывают серверу, какую операцию хочет произвести клиент. Команды состоят из ключевых слов, за которыми следует один или более параметров. Ключевое слово состоит из 4-х символов и разделено от аргумента одним или несколькими пробелами. Каждая командная строка заканчивается символами CRLF.

Команды smtp.

- HELO <SP> <domain> <CRLF> - приветствие
- MAIL <SP> FROM:<reverse-path> <CRLF> Отправитель
- RCPT <SP> TO:<forward-path> <CRLF> Получатель
- DATA <CRLF> Данные
- RSET <CRLF> прервать текущий процесс
- SEND <SP> FROM:<reverse-path> <CRLF> доставить на терминал
- SOML <SP> FROM:<reverse-path> <CRLF> Send+Mail
- SAML <SP> FROM:<reverse-path> <CRLF> Send+Mail
- VRFY <SP> <string> <CRLF> Проверка имени пользователя
- EXPN <SP> <string> <CRLF> Проверка почтовой группы
- HELP <SP> <string> <CRLF>
- NOOP <CRLF> Пустой оператор
- QUIT <CRLF> завершить работу.

Пример smtp сесиии

C-client S-server
#Поздоровались
C: HELO 195.161.101.33
S: 250 smtp.mail.ru is ready
#сообщили адреса
C: MAIL FROM:<drozd> #указываем отправителя
S: 250 OK
C: RCPT TO:<drol@mail.ru> #указываем получателя
S: 250 OK
C: DATA сообщаем,что дальше идут данные
S: 354 Start mail input; end with <CRLF>.<CRLF>
передачу письма необходимо завершить символами CRLF.CRLF

S: 250 OK
S: QUIT
C: 221 smtp.mail.ru is closing transmission channel

SMTP в Unix

В ОС семейства Unix за реализацию SMTP отвечает демон sendmail – невероятно мощная программа. Руководство к sendmail содержит более 800 страниц текста, что способно отпугнуть даже бывалых компьютерщиков. Тем не менее, настройка по-умолчанию + минимальные конкретизирующие настройки позволяет без проблем работать с этим демоном, оставляя все тонкости и нюансы на откуп энтузиастов.

Настройка sendmail

Все индивидуальные настройки заносятся в файл sendmail.mc , затем при помощи макропроцессора m4 создается сам конфигурационный файл sendmail.cf .

m4 sendmail.mc > sendmail.cf

sendmail.cf

```
DOMAIN(generic)
define(`confDOMAIN_NAME',`pogoda.nsk.su')
include(/usr/share/sendmail/cf/m4/cf.m4)
OSTYPE(linux)dnl
define(`ALIAS_FILE',`/etc/mail/aliases')
define(`SMART_HOST',`relay.turbosib.ru')

FEATURE(redirect)
FEATURE(local_procmail)
FEATURE(always_add_domain)
FEATURE(masquerade_entire_domain)
FEATURE(`accept_unresolvable_domains')
FEATURE(`accept_unqualified_senders')
FEATURE(allmasquerade)
FEATURE(`relay_entire_domain')

MASQUERADE_AS(pogoda.nsk.su)
MASQUERADE_DOMAIN(pogoda.nsk.su)
```

Перенаправление почты

Можно перенаправить поток почты, поступившей для одного пользователя другому пользователю или процессу (в этом случае процесс получает данные письма на stdin).

#/etc/aliases

postmaser: john,alex

robot: |/usr/bin/robot.pl

file: /root/myfile.mail

Изменения в таблице закрепляются командой newaliases.

отправка почты непосредственно из процесса

Удобна для генерации автоматических отчетов и другой информации.

1. Необходимо сгенерировать сообщение с полями согласно rfc822
2. Передать его программе sendmail с ключом . Можно открывать pipe с непосредственно в момент открытия файла.

   open MAIL,"/usr/lib/sendmail –t –oi";

Сообщение будет передано немедленно,если невозможно-будет помещено в системную очередь сообщений.

mailq – выводит системную очередь сообщений

Получение почты из ящика клиентом. Pop3

Протокол smtp предназначен для обмена сообщениями между узлами(серверами). Полученная почта накапливается в почтовых ящиках пользователей (/var/spool/mail). Для получения почты с сервера обычно используется протокол pop3 и его модификации.

pop3

Перед работой через протокол POP3 сервер прослушивает порт 110. Когда клиент хочет использовать этот протокол, он должен создать TCP соединение с сервером. Когда соединение установлено, сервер отправляет приглашение. Затем клиент и POP3 сервер обмениваются информацией пока соединение не будет закрыто или прервано.

Pop3 - правила

Команды POP3 состоят из ключевых слов, за некоторыми следует один или более аргументов. Все команды заканчиваются парой CRLF. Ключевые слова и аргументы состоят из печатаемых ASCII символов. Ключевое слово и аргументы разделены одиночным пробелом. Ключевое слово состоит от **3-х до 4-х символов**, а аргумент

может быть длиной до 40-ка символов.

Ответы в POP3 состоят из индикатора состояния и ключевого слова, за которым может следовать дополнительная информация. Ответ заканчивается парой CRLF. Существует только два индикатора состояния:

- "+OK" - положительный и
- "-ERR" - отрицательный.

Ответы из нескольких строк заканчиваются “.” и CRLF

pop3 - команды

USER [имя] – задает имя пользователя

Возможные ответы:

- * +OK name is a valid mailbox
- * -ERR never heard of mailbox name

Команда: PASS [пароль] – задает пароль пользовтеля

Возможные ответы:

- * +OK maildrop locked and ready
- * -ERR invalid password
- * -ERR unable to lock maildrop

STAT – выдает количество сообщений в ящике и их длину

* +OK n s

List [сообщение] – выдает информацию об указанном сообщении

- * +OK scan listing follows
- * -ERR no such message

TOP [сообщение] [n]- List + n строк сообщения

pop3 - команды

RETR [сообщение] передать тело сообщения

Возможные ответы:

- * +OK message follows
- * -ERR no such message

DELE [сообщение] сообщение помечается как удаленное, удаление по команде UPDATE

- * +OK message deleted
- * -ERR no such message

RSET – снимает метку удаленных сообщений

S: <создаём новое TCP соединение с POP3 сервером через порт 110>

S: +OK POP3 server ready
C: USER Monstr
S: +OK User Monstr is exists
C: PASS mymail
S: +OK Monsr's maildrop has 2 messages (320 octets)
C: STAT
S: +OK 2 320
C: LIST
S: +OK 2 messages (320 octets)
S: 1 120
S: 2 200
S: .
C: RETR 1
S: +OK 120 octets
S:
S: .
C: DELE 1
S: +OK message 1 deleted
C: RETR 2
S: +OK 200 octets
S:
S: .
C: DELE 2
S: +OK message 2 deleted
C: QUIT
S: +OK dewey POP3 server signing off (maildrop empty)
C: <закрываем соединение>

Борьба со спамом

- SMTP – авторизация
- SPAM – фильтры

  распознавание:

  – по IP
  – по заголовку
  – по телу сообщения
- Запрет неизвестных релеев
- Тайм-аут широковещательных рассылок.

Лекция

Лекция 15.

- Аппаратные средства обеспечения безопасности сети

- Ранее в нашем курсе мы рассмотрели программное решение построения маршрутизатора и фильтра пакетов. Рассмотрим аппаратное решение на базе оборудования DLINK.

Номенклатура оборудования

- Интернет-шлюз и маршрутизатор (DI, DIR, DSL)
  - Маршрутизация
  - Тривиальная фильтрация
  - Маскирование и трансляция портов (NAT)
- Межсетевые экраны (DFL)

Интернет-шлюзы DI-604 / DI-624S

DI-804

DI-824

- 4 коммутируемых порта 10/100Base-TX для подключения к LAN
- Контроль потока 802.3 для встроенного коммутатора
- Один порт 10/100Base-TX для подключения кабельного или DSL модема
- Встроенная беспроводная точка доступа IEEE 802.11g – до 108 Мбит/с – **только для DI-824S**
- Межсетевой экран (+IDS)
- Фильтрация на основе MAC, IP, URL адресов
- DMZ
- Настройка посредством web-браузера и SNMP
- Поддержка VPN в режиме pass-through
- Клиент PPPoE, PPTP

Типичные примеры использования Интернет-шлюзов и межсетевых экранов класса SOHO:

1. Подключение к сети ETTH и доступа в Интернет небольших офисов, используя один IP адрес
2. Защита от вторжения из Интернет
3. Использование в домашних сетях
4. Объединение офисов, находящихся в разных частях города/страны и подключенных к Интернет – VPN соединения
5. Подключение устройств по защищенным соединениям IPSec

+ цена. - возможности,скорость,надежность

Типичные задачи интернет-маршрутизаторов

- Подключение небольшого офиса в Интернет
- Разделение полосы пропускания
- Трансляция IP адресов (NAT)
- Проброс портов (Port-Mapping)
- Организация VPN (PPPoE, IpSec)
- DHCP сервер

Firewall

- Межсетевые экраны
  - Сложная маршрутизация
  - Фильтрация как по типу, так и по содержимому
  - Аутентификация
  - Баланс нагрузки
  - Избыточные соединения во внешние сети
  - Поддержка большого числа VPN соединений
  - Поддержка DMZ
  - Сбор статистики
  - Встроенные средства «антиспам» и «античервь»

Фильтрация пакетов

- По адресам источника и назначения
- По IP пакетам
- По транспортному протоколу
- **По содержимому IP пакета**
- **По содержимому TCP пакета**
- **Application Level OSI**

+ управление коммутаторами для отсечения зоны

Firewall

- **Программный firewall**

Программные межсетевые экраны работают на базе традиционных операционных систем, которые сами имеют слабые места, постоянно изучаемые и атакуемые хакерами. Для настройки программного межсетевого экрана требуется опытный системный администратор.

- **Аппаратный firewall**

Специальное устройство созданное для зашиты сети.

Аппаратные межсетевые экраны почти всегда построены с использованием операционных систем, специально разработанных для этой цели. Аппаратные межсетевые экраны относительно легки в настройке и обслуживании.

DMZ

Многие организации не используют в своих сетях демилитаризованную зону, DMZ. Вместо этого они размещают свои серверы (например, Web, mail,FTP или SQL серверы) в той же внутренней сети, где находятся серверы и рабочие станции компании. Без DMZ, отделяющей общедоступные серверы от внутренней сети, последняя подвергается дополнительному риску. Когда атакующий получит возможность управления взломанным сервером, он сможет использовать его для атаки на важные ресурсы, такие как финансовые приложения и файловые серверы. Именно «когда», а не «если». Потому что независимо от того, как защищен Web-сервер, рано или поздно он подвергнется атаке. Следовательно, необходимо проектировать сеть и рабочие процессы с учетом минимизации ущерба от вторжений и гарантии их быстрого восстановления. Одной из таких стратегий является стратегия выделения рабочих зон и использование демилитаризованной зоны (DMZ)

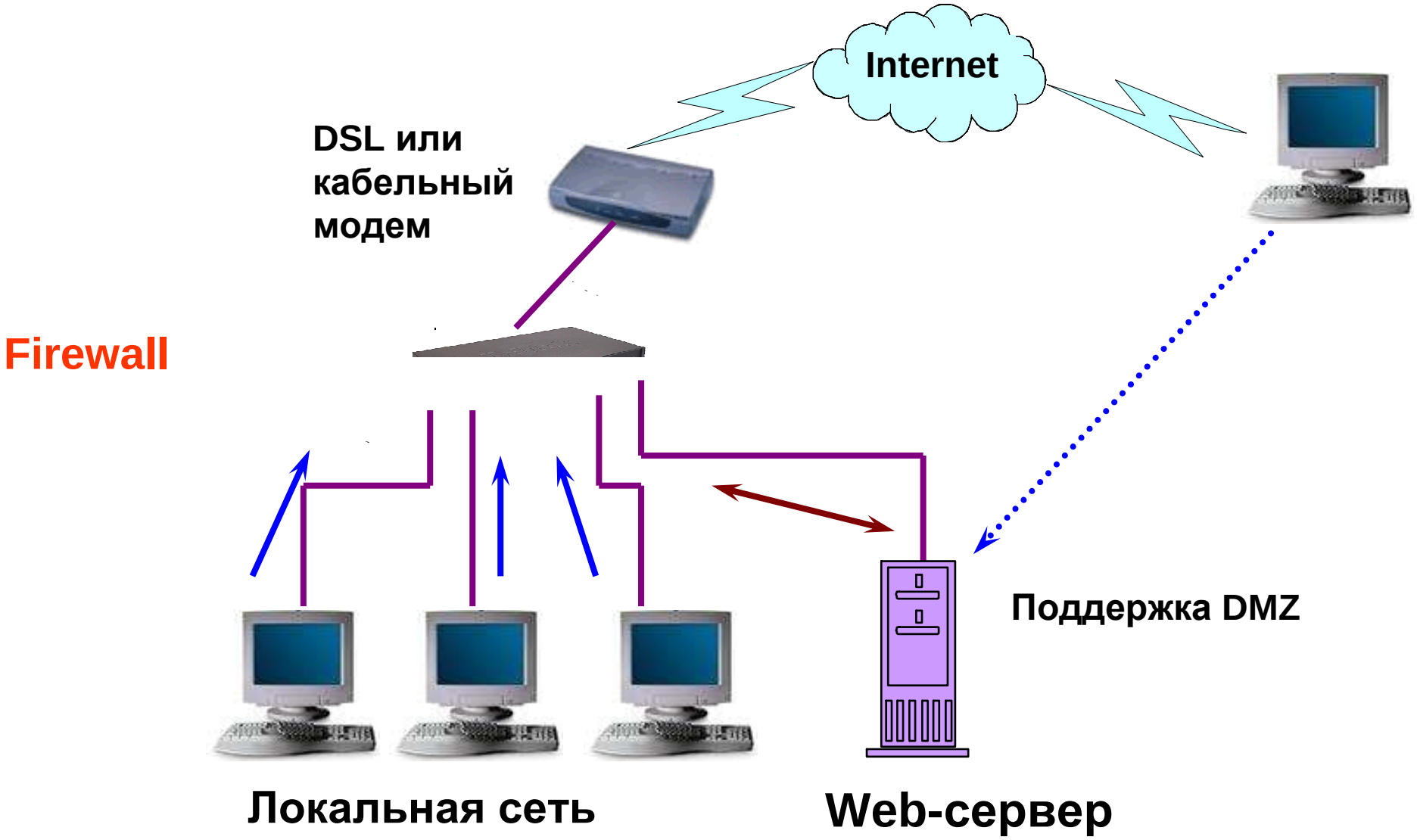
DMZ

- При формировании DMZ создается две физически разделенные сети: одна — для общедоступных серверов, другая — для внутренних серверов и рабочих станций. В зависимости от типа DMZ и числа используемых брандмауэров, применяется та или иная политика маршрутизации для каждой из сетей и жестко контролируется доступ между:
- Internet и DMZ;
- Internet и внутренней сетью;
- DMZ и внутренней сетью.

DMZ

Главное преимущество использования DMZ вместо простого брандмауэра состоит в том, что при атаке на общедоступный сервер риск компрометации внутренних серверов снижается, поскольку общедоступные и внутренние серверы отделены друг от друга. Если скомпрометированный сервер находится в DMZ, злоумышленник не сможет напрямую атаковать другие, более важные серверы, расположенные во внутренней сети. Брандмауэр блокирует любые попытки компьютеров из DMZ подсоединиться к компьютерам внутренней сети, за исключением специально разрешенных соединений. Например, можно настроить брандмауэр так, чтобы разрешить Web-серверу, находящемуся в DMZ, подсоединяться к внутренней системе с Microsoft SQL через специальный TCP-порт. Если злоумышленник захватит Web-сервер, он сможет организовать атаку на систему SQL Server через этот порт. Однако злоумышленник не сможет атаковать другие службы и порты системы с SQL Server, равно как и другие компьютеры во внутренней сети.

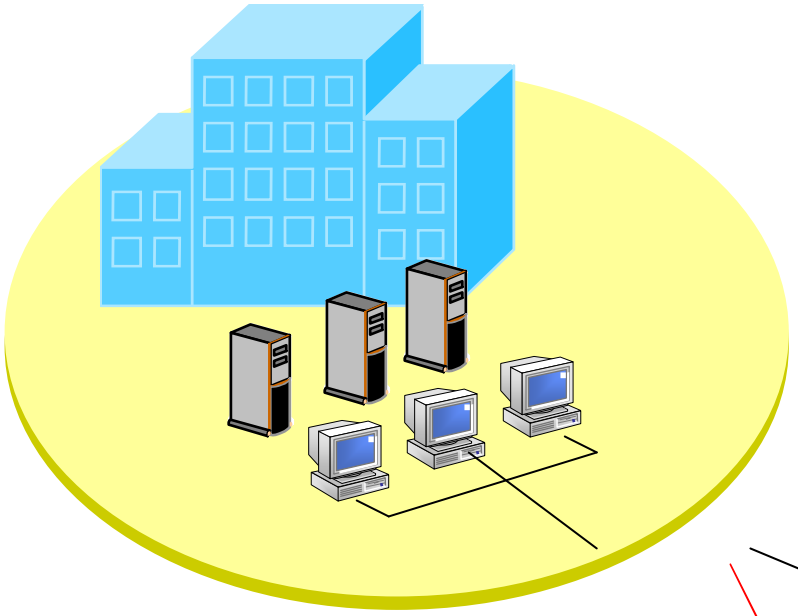
Функция DMZ и Virtual Servers

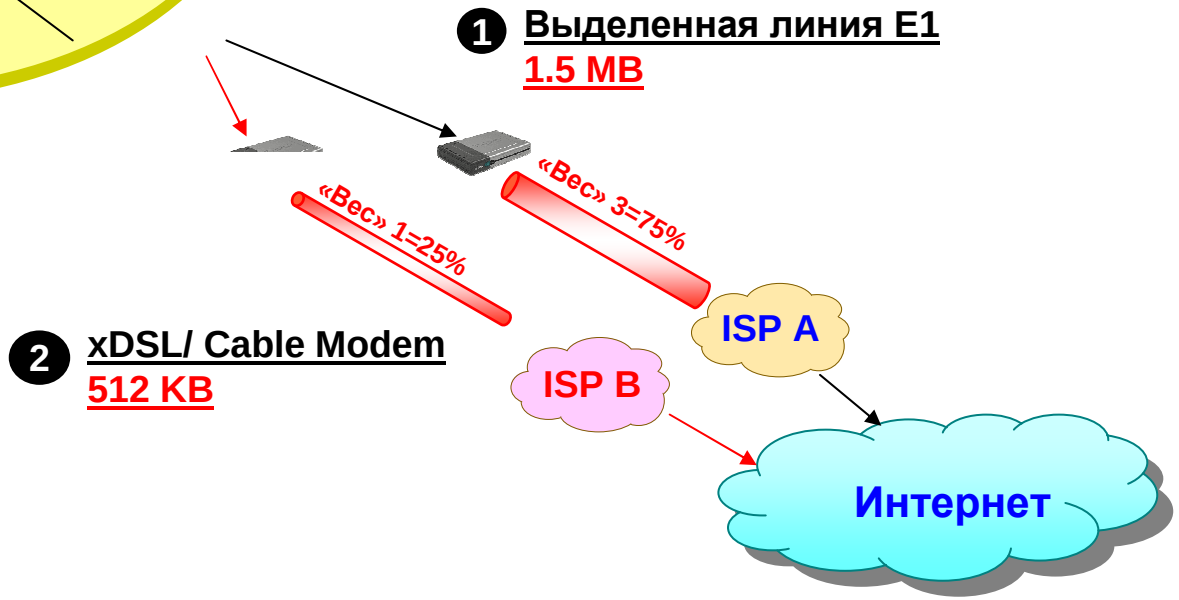


Баланс нагрузки

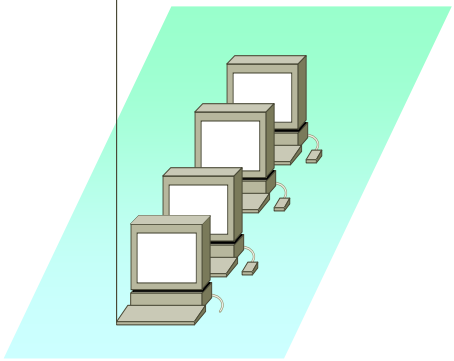
Баланс нагрузки на основе задания приоритетов – «весов»

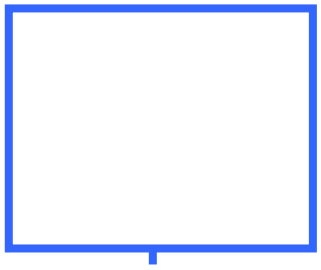
Пример: Ширина полосы пропускания выделенной линии E1 (WAN 1) в три раза больше, чем подключения по линии ADSL 512K (WAN 2). Мы можем назначит баланс нагрузки 3: 1 для WAN 1 и WAN 2.





Обеспечение отказоустойчивости





Маршрутизатор

Межсетевой экран DFL-210

- 6 портов 10/100 Мбит/с Fast Ethernet (1 порт для кабельного или DSL подключения, 1 порт DMZ и 4 порта для подключения к локальной сети)
- Межсетевой экран:
  - Stateful Packet Inspection (SPI)
  - Журнал системных событий
  - Предупреждение по email
- Фильтрация по содержимому (блокирование URL; блокирование Java/ ActiveX/Cookie/Proxy)
- Построение VPN туннелей на основе протокола IPSec
- Защита от сетевых атак.
- Поддержка PPTP и L2TP сервера
- Алгоритмы шифрования DES & 3DES (с аппаратным ускорением), RC4
- PPPoE для xDSL, PPTP для xDSL

DFL-800/1600/2500

Законченное решение в области безопасности, включающее встроенную поддержку межсетевого экрана, балансировки нагрузки, функций отказоустойчивости, фильтрации содержимого, аутентификации пользователей, блокировки «мгновенных» сообщений и приложений P2P, защиты от атак «отказ в обслуживании» DoS и виртуальных локальных сетей VPN. Межсетевые экраны поддерживают специальную функцию - Zone-Defense, представляющую собой механизм, позволяющий им работать с коммутаторами локальных сетей D-Link и обеспечивающий активную сетевую безопасность. Функция Zone-Defense автоматически изолирует инфицированные компьютеры сети и предотвращает распространение ими вредоносного трафика.

DFL-800

- **Интерфейсы**
- 2 порта 10/100Base-TX WAN,
- 1 порт 10/100Base-TX DMZ3
- 7 портов 10/100Base-TX LAN

Производительность

- Производительность межсетевого экрана 150 Мбит/с
- Производительность VPN 60 Мбит/с
- Количество параллельных сессий 25 000
- Политики 1 000

DFL-800

- **Функции межсетевого экрана**
- PPPoE
- Прозрачный режим
- NAT, PAT
- Протокол динамической маршрутизации OSPF
- Политики по расписанию
- Application Layer Gateway (ALG)
- Технология Zone-Defense

DFL-800

- **Сетевые функции**
- DHCP клиент/север
- DHCP relay
- Маршрутизация на основе политик
- IEEE 802.1Q VLAN

Виртуальные частные сети (VPN)

- Шифрование (DES / 3DES / AES / Twofish / Blowfish / CAST-128)
- 300 выделенных VPN-туннеля
- Сервер PPTP/L2TP
- Hub and Spoke
- IPSec NAT Traversal

DFL-800

-

DFL-800

- **Балансировка нагрузки**
- *Балансировка исходящего трафика (будет доступно прошивки)*
- Алгоритм балансировки: 2 типа
- Перенаправление трафика при обрыве канала

Управление полосой пропускания

- Traffic Shaping на основе политик
- Гарантированная полоса пропускания
- Максимальная полоса пропускания
- Полоса пропускания на основе приоритета

Отказоустойчивость

- Резервирование канала WAN

DFL-800

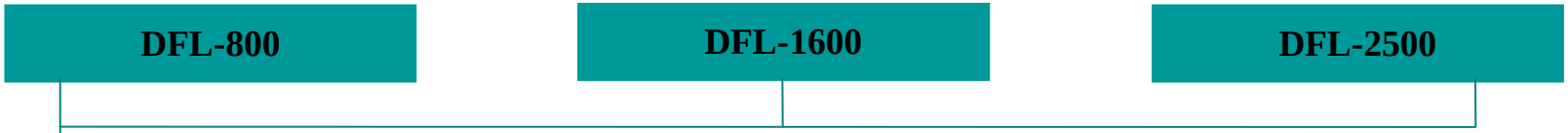
Intrusion Detection (IDS)

- Автоматическое обновление шабло
- Защита от атак DoS, DDoS ,SynFlood …
- Предупреждение об атаках по email

Фильтрация содержимого

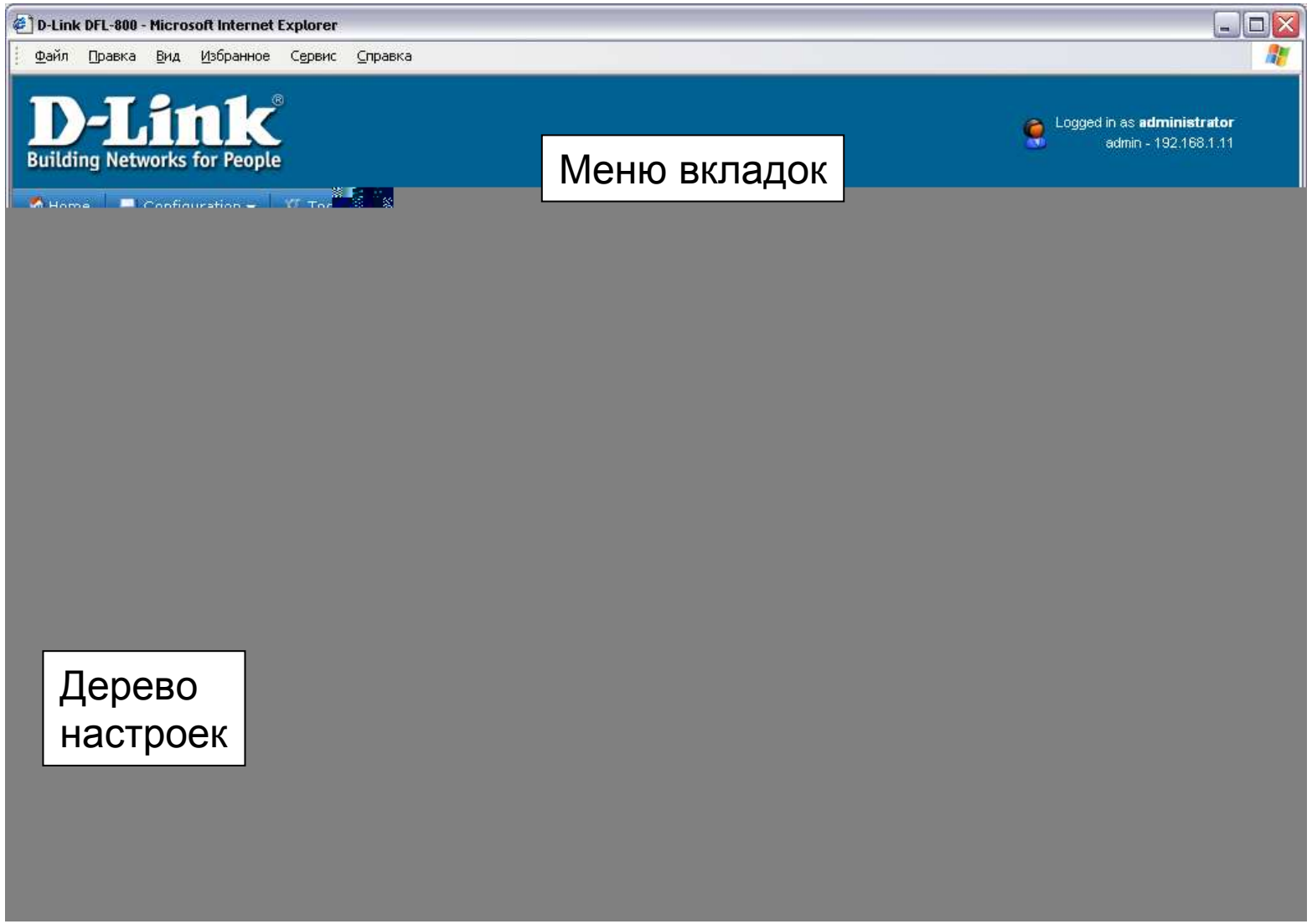
- Тип HTTP: URL, ключевые слова
- Тип скриптов: Java Cookie, ActiveX, VB
- *Тип email: «Черный» список, ключевые слова (в следующих прошивках)*

DFL-800/1600/2500



- высокая плотность портов и гигабитные порты (DFL-1600/2500)
- новый дружественный интерфейс управления
- Система Zone Defense предотвращает распространение червей внутри сети, взаимодействуя с коммутаторами D-Link (управляет блокировкой портов)
- Широкие возможности, включая “прозрачный режим”, упрощают установку в существующие системы

Конфигурация- основная панель



Список разделов WEBUI с кратким описанием

- Home – заглавная страница WebUI
- **Configuration** – меню конфигурации
  - **Save and Activate**: Записать конфигурацию и активировать изменения. В отличие от других продуктов DLINK (например, точек доступа), никакие изменения не будут приняты устройством до тех пор, пока не будут записаны в устройство из этого меню. Следует также отметить, что если изменения не будут активированы и сессия браузера будет закрыта, то они будут утеряны.
  - **Discard Changes:** отменить все изменения, произведенные в текущем сеансе.
- **Tools** – меню инструментов
  - Ping – классическая ICMP эхо-локация (указывается размер и число пакетов)
  - Backup – Архивирование и восстановление из архива текущей конфигурации
  - Reset – сброс всех настроек и возврат к заводской конфигурации.
  - Upgrade – Обновление ПО межсетевого экрана

Список разделов WEBUI с кратким описанием

- Status – мониторинг устройства и состояния
  - System – Информация о статусе системы (CPU, connections и.т.д)
  - Logging: Протоколы событий, сохраненные в памяти системы.
  - Connections: отображает активные сетевые соединения
  - Interfaces – отображает состояние сетевых интерфейсов
  - IPSec – информация о туннелях IPSec
  - Routes – информация о текущей таблице маршрутизации
  - DHCP – информация о DHSP сервисе, его статусе и состоянии пула
  - IDS: Состояние «интеллектуального анализатора информации (Intrusion Detection System)»
  - SLB Состояние сервиса Server Load Balancing – балансировка нагрузки между кластером из серверов

Объекты

Методика

Различия в общей методики конфигурирования аппаратных межсетевых экранов и традиционных программных фильтров пакетов подобно различиям в объектно-ориентированном и традиционном линейном программировании.

Если в программных фильтрах мы имели дело с конкретными портами, значениями и другими параметрами, то идеология конфигурирования аппаратного фильтра пакета подразумевает два этапа:

- Объявление логических объектов (порты,адреса,протоколы, интерфейсы)
- Объявления действий над объектами

Некоторые объекты предопределяются при начальной загрузке устройства (например, интерфейсы WAN,LAN и типичные сервисы – http,tcp,udp).

Например, для того чтобы создать правило фильтрации с участием какого-либо порта, не перечисленного в списке стандартных сервисов необходимо:

- Объявить объект «порт» и присвоить ему имя
- Объявить правило фильтрации для объекта порт

РАССМОТРИМ ПОДРОБНЕЕ ДОСТУПНЫЕ ТИПЫ ЛОГИЧЕСКИХ ОБЪЕКТОВ.

Логические объекты

Книга адресов – IP адрес
Позволяет создавать именованные объекты

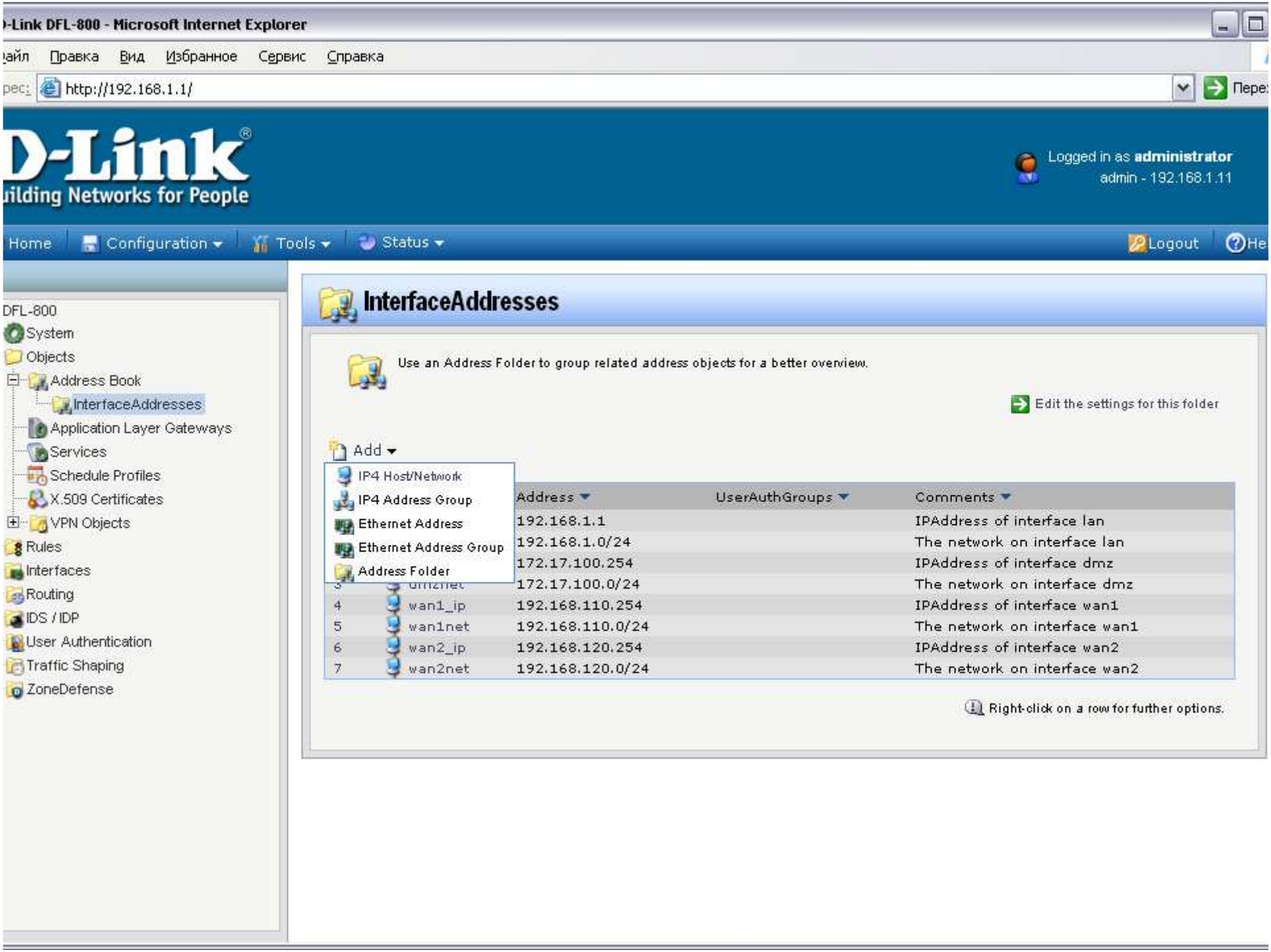
- Одиночный Ip адрес
- Сеть IP адресов
- Пара master\slave
- Группа IP адресов или интерфейсов

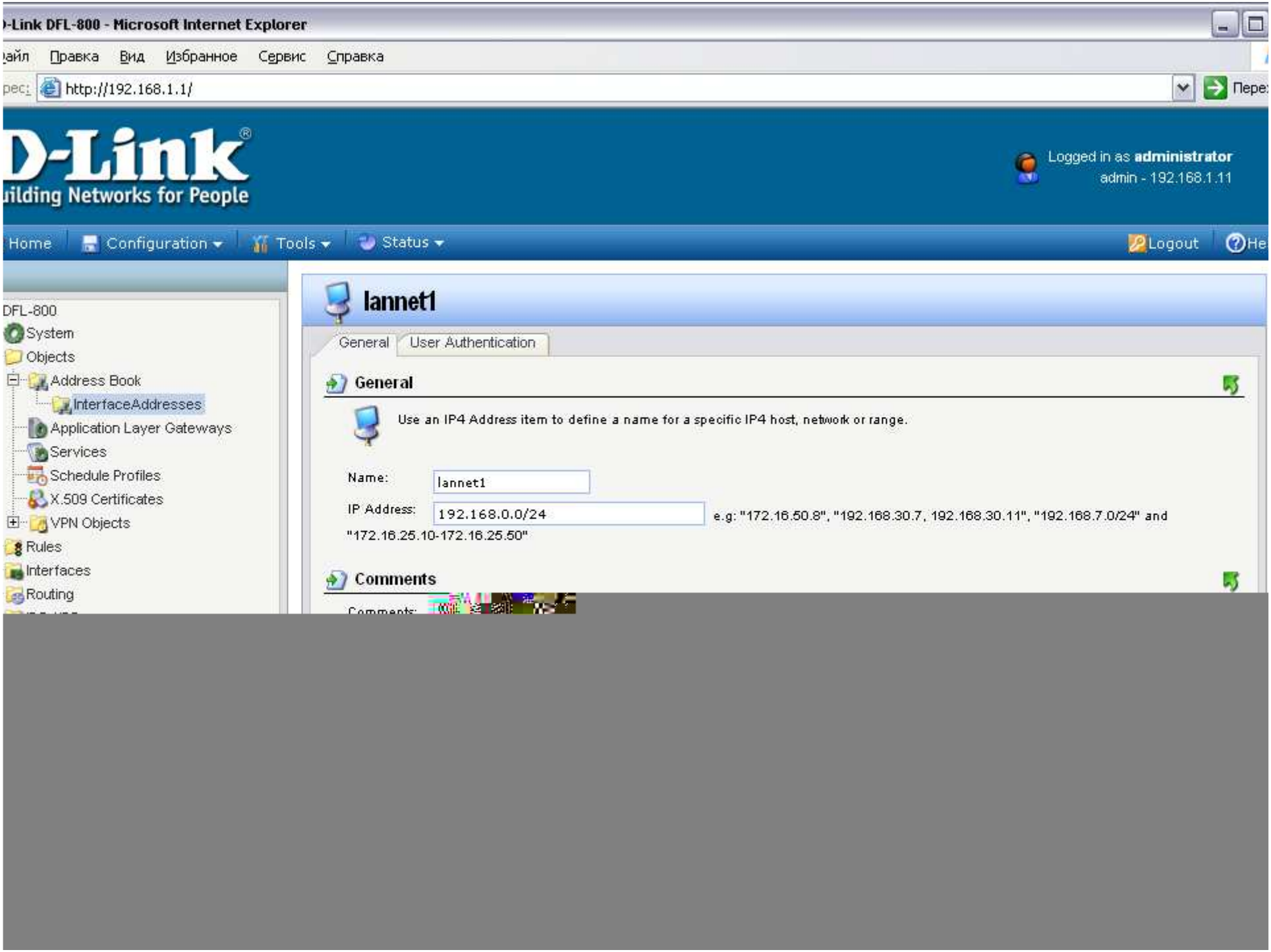
Например, 0.0.0.0/0” называется ”all-nets” и используется для обозначения всех возможных адресов

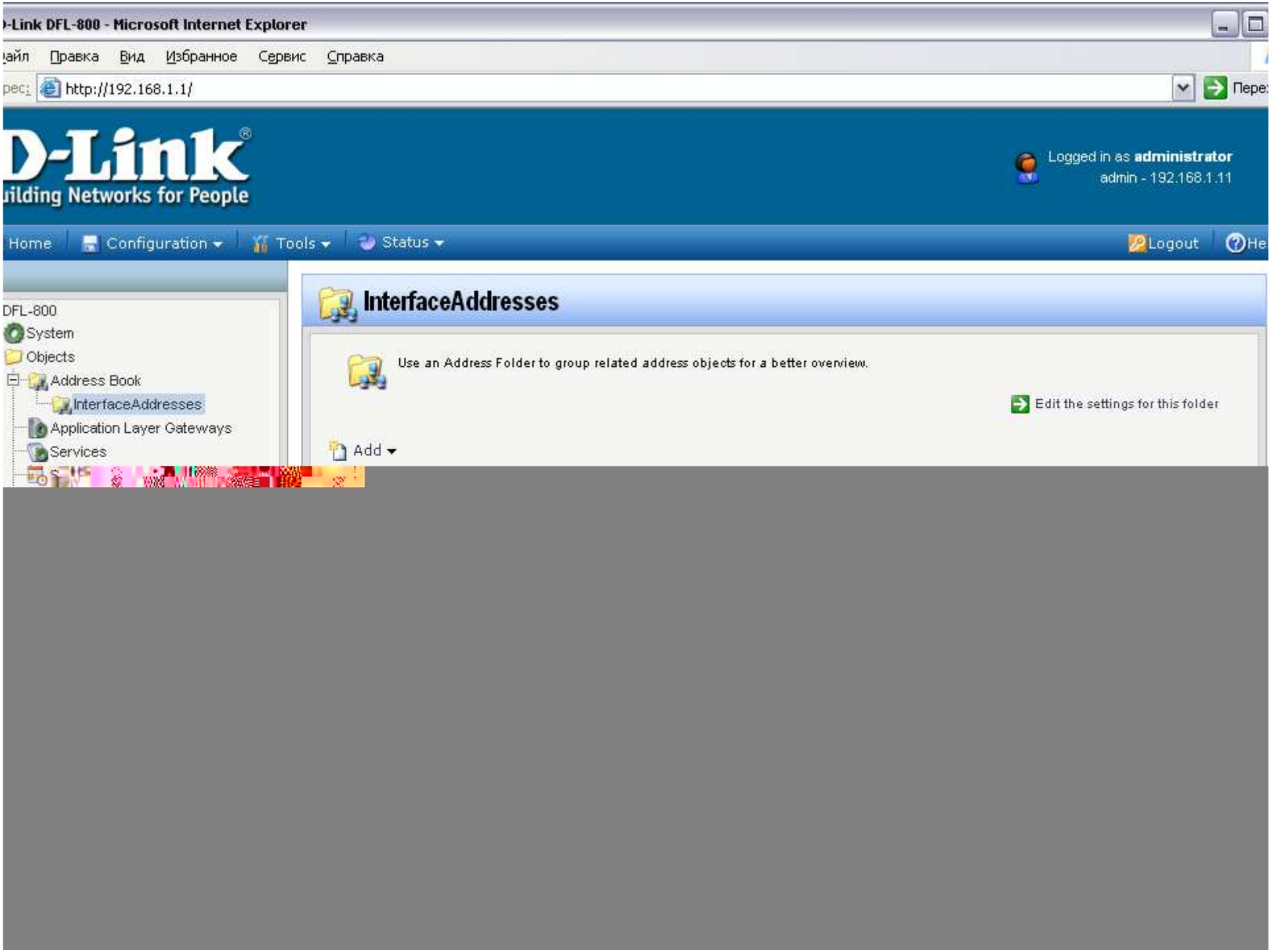
Также возможна сопоставление аутентификация пользователя и IP адреса путем установления связи между объектом из «книги адресов» и «акаунты пользователей»

Пример: локальная сеть ”192.168.0.0/24” объявлена как ”lannet”.

Objects − > Address Book − > InterfaceAddresses − > Add − >
IP4 Host/Network − > General:
Enter the following and then click OK:
Name: lannet1
IP Address: 192.168.0.0/24


-Link DFL-800 - Microsoft Internet Explorer
айл Правка Вид Избранное Сервис Справка
pec: http://192.168.1.1/
D-Link
uilding Networks for People
Logged in as administrator
admin - 192.168.1.11
Home
Configuration
Tools
Status
Logout
He
DFL-800
System
Objects
Address Book
InterfaceAddresses
Application Layer Gateways
Services
Schedule Profiles
X.509 Certificates
VPN Objects
Rules
Interfaces
Routing
IDS / IDP
User Authentication
Traffic Shaping
ZoneDefense
InterfaceAddresses
Use an Address Folder to group related address objects for a better overview.
Edit the settings for this folder
Add
IP4 Host/Network
IP4 Address Group
Ethernet Address
Ethernet Address Group
Address Folder
Address
UserAuthGroups
Comments
192.168.1.1
IPAddress of interface lan
192.168.1.0/24
The network on interface lan
172.17.100.254
IPAddress of interface dmz
172.17.100.0/24
The network on interface dmz
4 wan1_ip 192.168.110.254
IPAddress of interface wan1
5 wan1net 192.168.110.0/24
The network on interface wan1
6 wan2_ip 192.168.120.254
IPAddress of interface wan2
7 wan2net 192.168.120.0/24
The network on interface wan2
Right-click on a row for further options.


-Link DFL-800 - Microsoft Internet Explorer
айл Правка Вид Избранное Сервис Справка
pec: http://192.168.1.1/
D-Link
uilding Networks for People
Logged in as administrator
admin - 192.168.1.11
Home
Configuration
Tools
Status
Logout
He
DFL-800
System
Objects
Address Book
InterfaceAddresses
Application Layer Gateways
Services
Schedule Profiles
X.509 Certificates
VPN Objects
Rules
Interfaces
Routing
lannet1
General
User Authentication
General
Use an IP4 Address item to define a name for a specific IP4 host, network or range.
Name:
lannet1
IP Address:
192.168.0.0/24
e.g: "172.16.50.8", "192.168.30.7, 192.168.30.11", "192.168.7.0/24" and "172.16.25.10-172.16.25.50"
Comments


-Link DFL-800 - Microsoft Internet Explorer
айл Правка Вид Избранное Сервис Справка
pec: http://192.168.1.1/
D-Link
uilding Networks for People
Logged in as administrator
admin - 192.168.1.11
Home
Configuration
Tools
Status
Logout
He
DFL-800
System
Objects
Address Book
InterfaceAddresses
Application Layer Gateways
Services
InterfaceAddresses
Use an Address Folder to group related address objects for a better overview.
Edit the settings for this folder
Add

Логические объекты

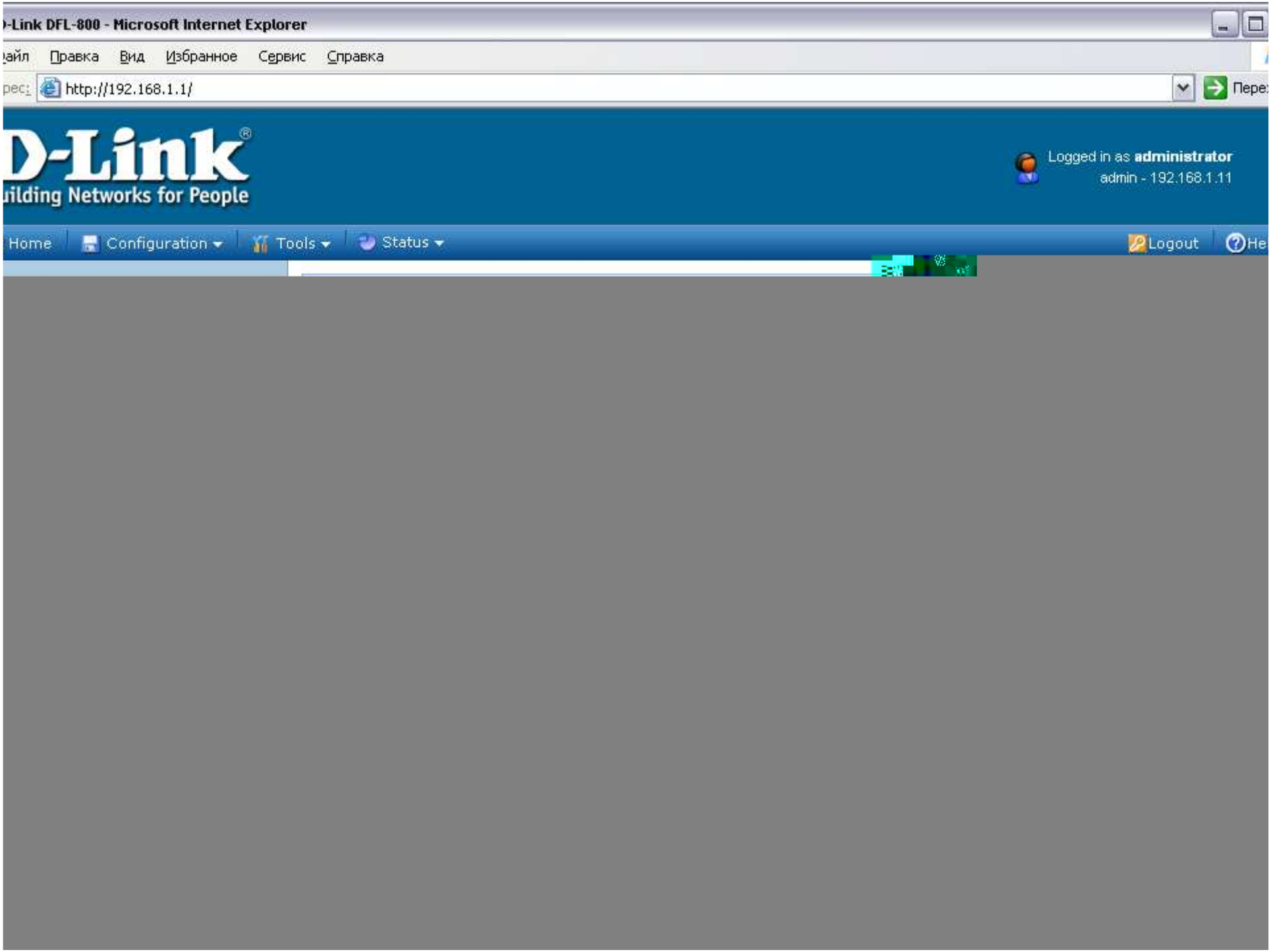
- Ethernet address – MAC адрес
- Ethernet address Group – объект группа IP адресов
- Address folder – Папка в дереве. Объект физического отображения не имеет- только группа других объектов.

Логические объекты-сервисы

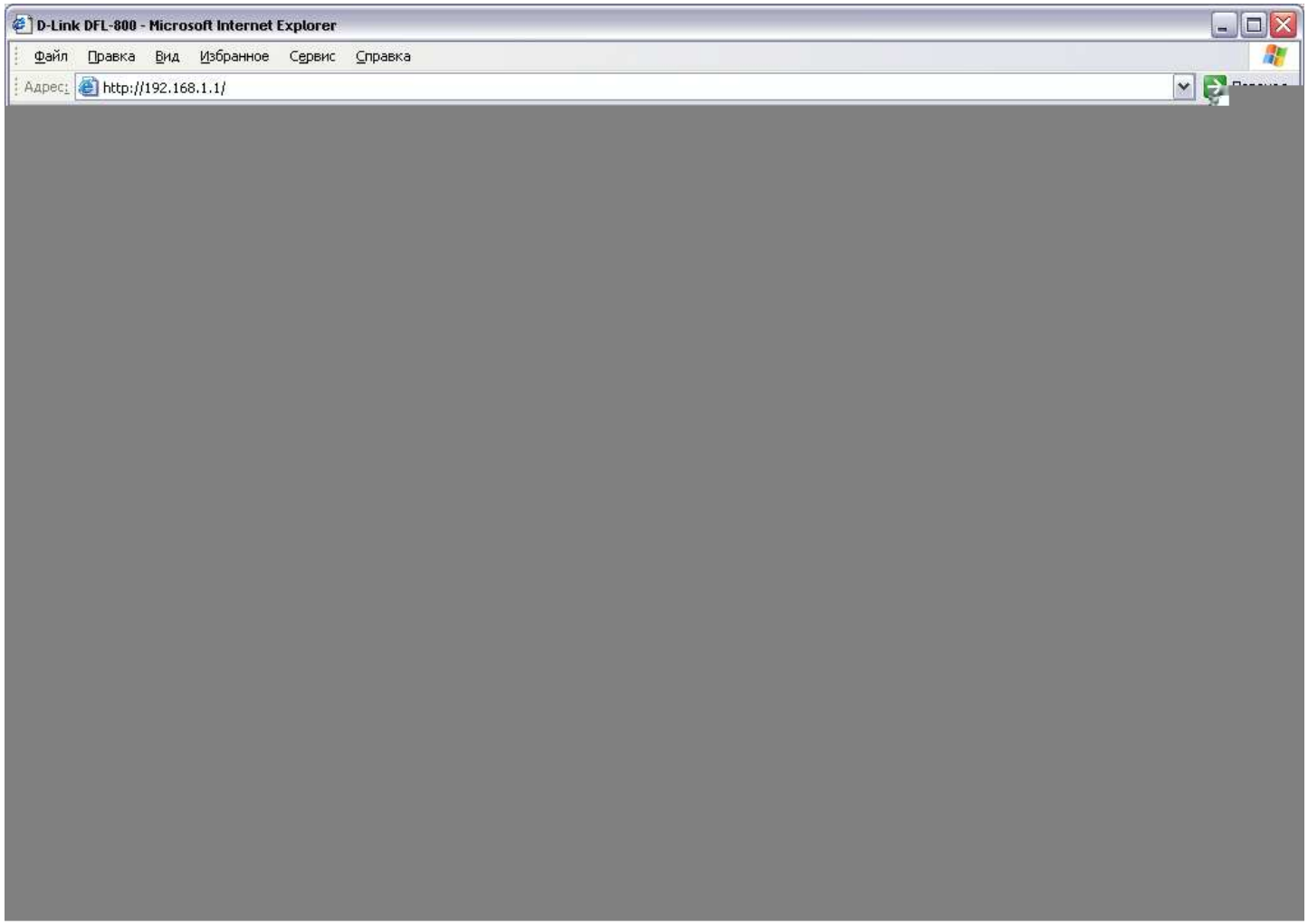
Логические объекты: сервисы – объекты высшего уровня OSI – Application.

- TCP/UDP service
- ICMP service
- IP protocol service
- Service group

Сервисы можно ассоциировать с объектом ALG (Application Layer Gateways) интеллектуальный анализатор содержимого. Рассмотрим позднее


-Link DFL-800 - Microsoft Internet Explorer
айл Правка Вид Избранное Сервис Справка
pec: http://192.168.1.1/
D-Link
uilding Networks for People
Logged in as administrator
admin - 192.168.1.11
Home
Configuration
Tools
Status
Logout
He

Логические объекты - временные интервалы

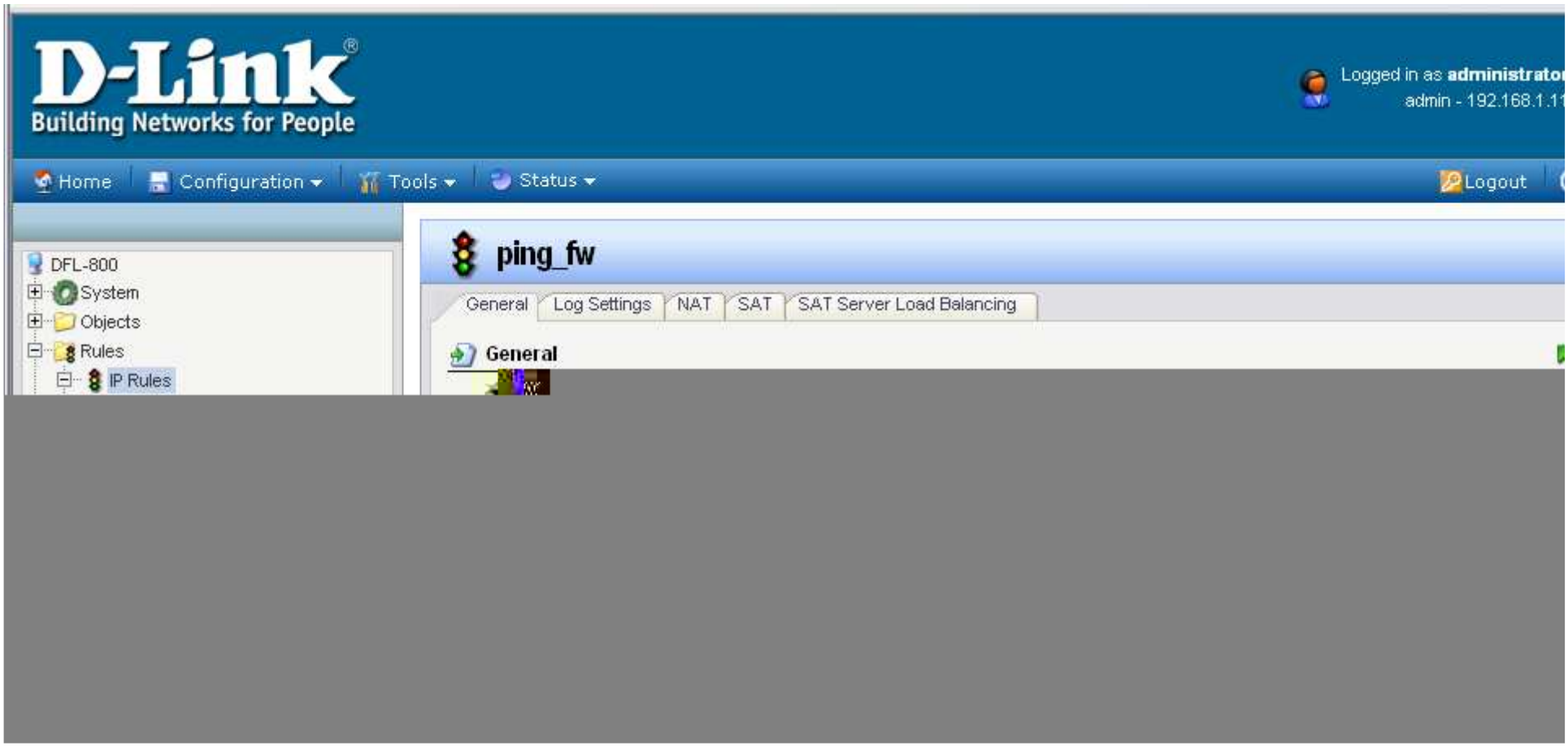


Логические объекты - временные интервалы

Логические интерфейсы

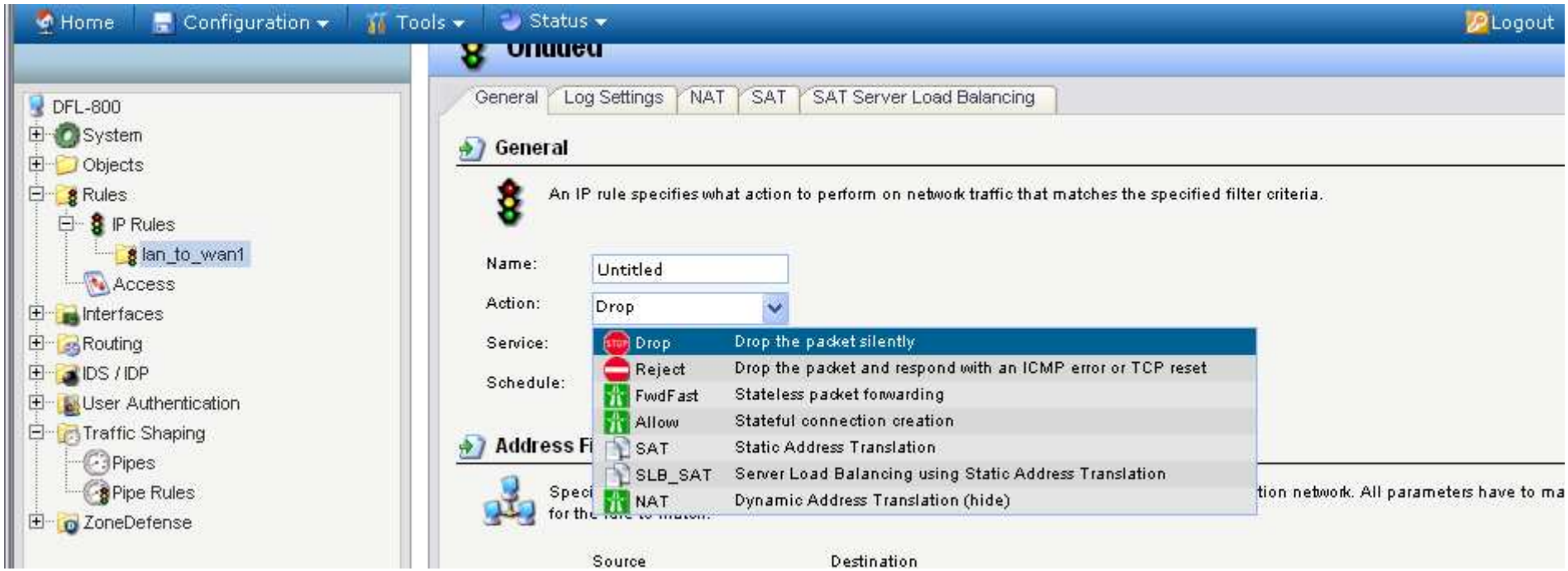
- Применительно к межсетевым экранам выделяют два логических интерфейса: **core** и **any**.
- **Core** находится «в сердце» межсетевого экрана, от физических интерфейсов весь трафик пересылается на **core** с целью управления политиками безопасности.
- **Any** – означает все возможные интерфейсы, включая **core**.

Пример Core- разрешаем ping



Операции над пакетами

Каждое действие над пакетами должно быть объявлено в Ip rules



Возможные действия

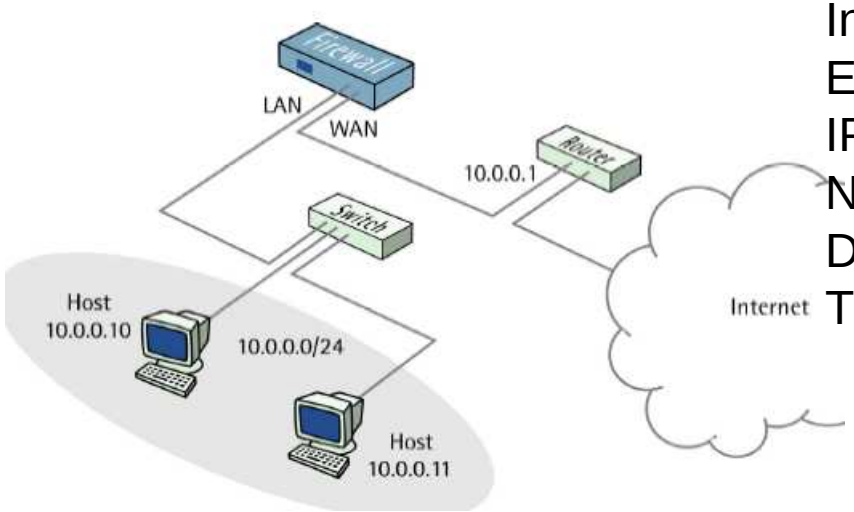
- Drop – сбросить пакет
- Reject – сбросить пакет и послать уведомление
- **FwdFast – транзит пакета**
- Allow – принять пакет
- SAT – «Проброс портов»
- SLB_Sat – Проброс портов к нескольким серверам с балансом трафика
- NAT- маскирование сетевых адресов

Настройка таблицы маршрутизации

- Принципы настройки таблицы маршрутизации подобны тому, что мы изучали в программных маршрутизаторах. Настройка в устройстве производится по уже знакомым нам «объектно ориентированным» принципам.
- Пакет, попадая в таблицу фильтрации ядра вычисляется интерфейс для перенаправления пакета. Если существуют альтернативные пути, то будет выбран путь с минимальной метрикой.
- Помимо статической маршрутизации, поддерживается динамическая маршрутизация по протоколу RIP

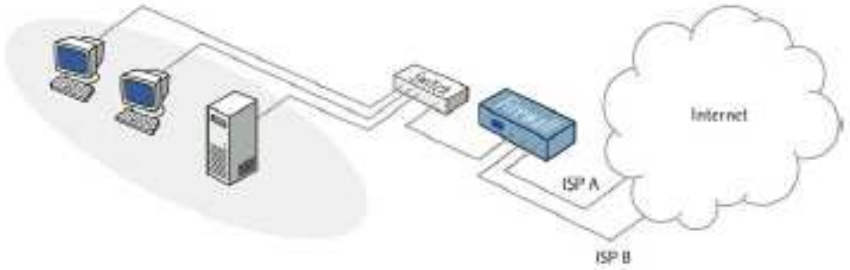
Два режима работы FIREWALL:

- Normal Mode (IP,TCP…)
- Transparent Mode (Умный OSI 3)



Interfaces − > Ethernet − > Edit (WAN):
Enter the following:
IP Address: 10.0.0.2
Network: 10.0.0.0/24
Default Gateway: 10.0.0.1
Transparent

Поддержка избыточной маршрутизации



- В случае избыточно маршрутизации в системе появятся два маршрута по умолчанию (all_nets 0.0.0.0/0), настроенных для разных интерфейсов. Различие двух маршрутов будет в метриках – первый будет иметь метрику 1,второй 2.

PBR (Policy Based routing)

PBR это расширение обычной системы маршрутизации, позволяющее:

- Осуществлять «маршрутизацию от источника» (избыточные соединения)
- Маршрутизацию «по расписанию»
- Маршрутизацию «по содержимому»

PBR состоит из двух частей:

- Несколько PBR таблиц в расширении основной
- Набор правил, определяющий какая из таблиц должна использоваться в определенный момент.

OSPF: Open Shortest Path First

Алгоритм автоматического построения маршрутной таблицы

Каждый роутер сообщает широковещательно сети, непосредсвенно подключенных к его портам. На основании этой информации вычисляется оптимальный путь пересылки пакета.

Настраивается:

- группа роутеров, которым необходимо сообщать свою OSPF данные ()
- Способ построения таблицы (построить записи динамически или взять из предопределенных)

Правила фильтрации

- IP Rules
- Access (Anti-spoofing)
- DMZ & Port Forwarding
- User Authentication

Логика построения:

- При попытке установить соединение выполняется просмотр правил «сверху вниз» пока не будет найдено правило, подходящее по критериям.
- В случае действия “Accept” соединение устанавливается и заносится в стек установленных соединений Firewall
- В случае если соединение уже установлено, для экономии ресурсов устройства не выполняет повторный поиск в правилах. Таким образом, размер маршутной таблицы будет влиять только скорость установки соединения, но не будет влиять на процесс дальнейшего обмена данными.

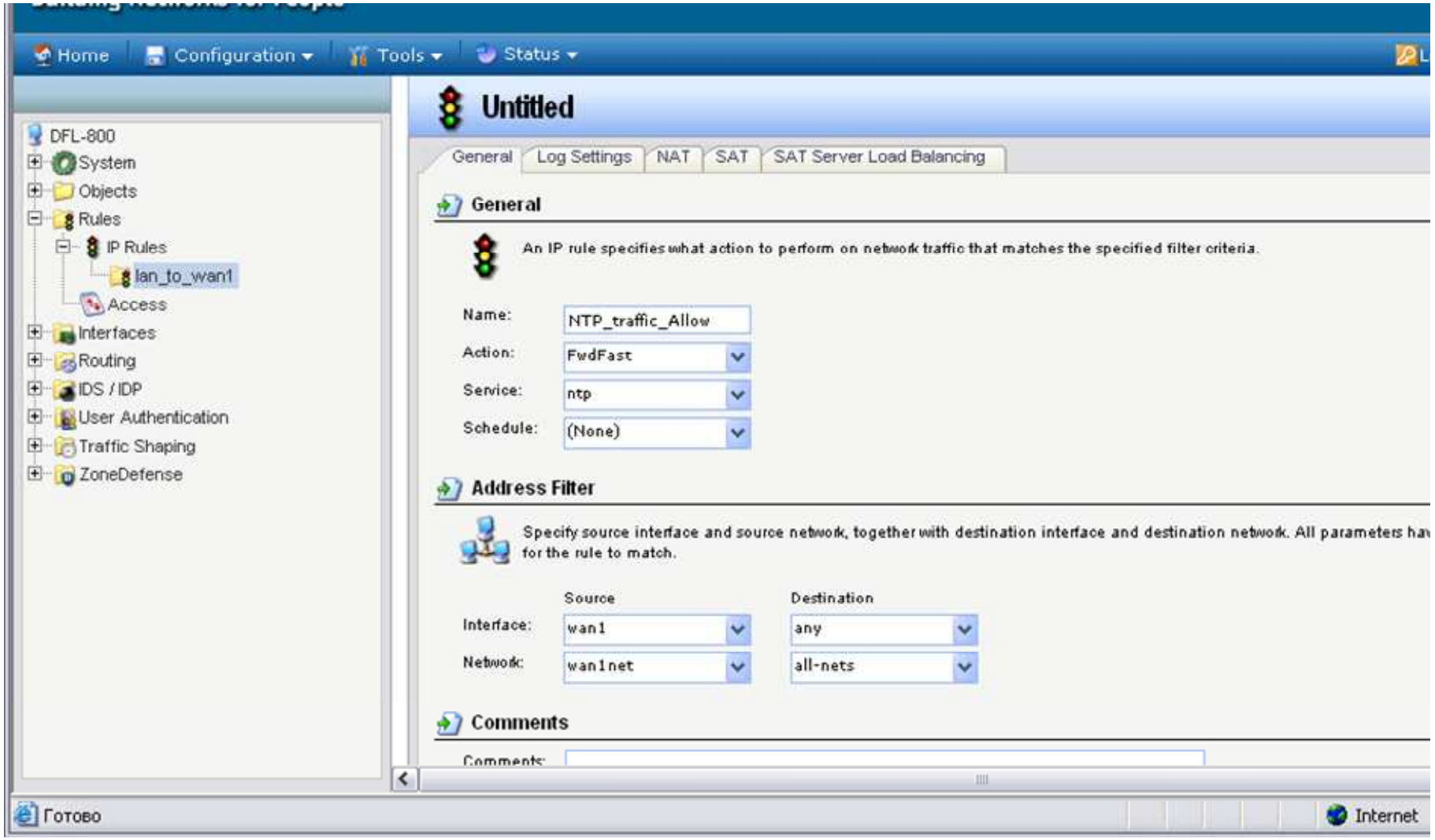
Доступные критерии

- **Service**: протокол,который будет применяться (объявляются как логические объекты).
- **Source Interface:** Один или группа интерфейсов - источник
- **Source Network:** адрес сети входящего пакета
- **Destination Interface**: один или группа интерфейсов-назначение
- **Destination Network:** сеть назначения

Действия

- Allow - открывается соединение и устройство запоминает его в своем кэш
- NAT – динамическая трансляция (маскирование) сетевых адерсов
- FwdFast – пакет принимается, но не происходит сохранение состояния соединения в кэш
- SAT – выполняется проброс порта. Требует еще одного правила (NAT или FwdFast) «ниже».
- Drop – пакет сбрасывается.
- Reject – пакет отклоняется с посылкой “ICMP–Unreachable”
- Expect: Если адрес отправителя пакета соответствует сети, указанной в данном правиле, интерфейс назначения также соответствует интерфейсу назначения- пакет принимается. Иначе - пакет отвергается. (предотвращает IP-spoofing). Присутствует только во вкладке Rules->Access

Пример- разрешаем ntp трафик



Аутентификация пользователей

Устройство поддерживает несколько режимов аутентификации пользователей. Режим аутентификации безусловно потребуется для создания PPtp сервера (сервиса удаленного доступа к локальной сети извне) и туннелей IPSec (в данном случае- в виде ключей).

Однако, возможно настроить устройство так, чтобы оно осуществляло аутентификацию даже обычных пользователей, запросивших услуги трансляции пакетов у Firewall.

Контроль содержимого

В дополнении к контролю пакетов на сетевом уровне, устройство позволяет анализировать содержимое пакетов.

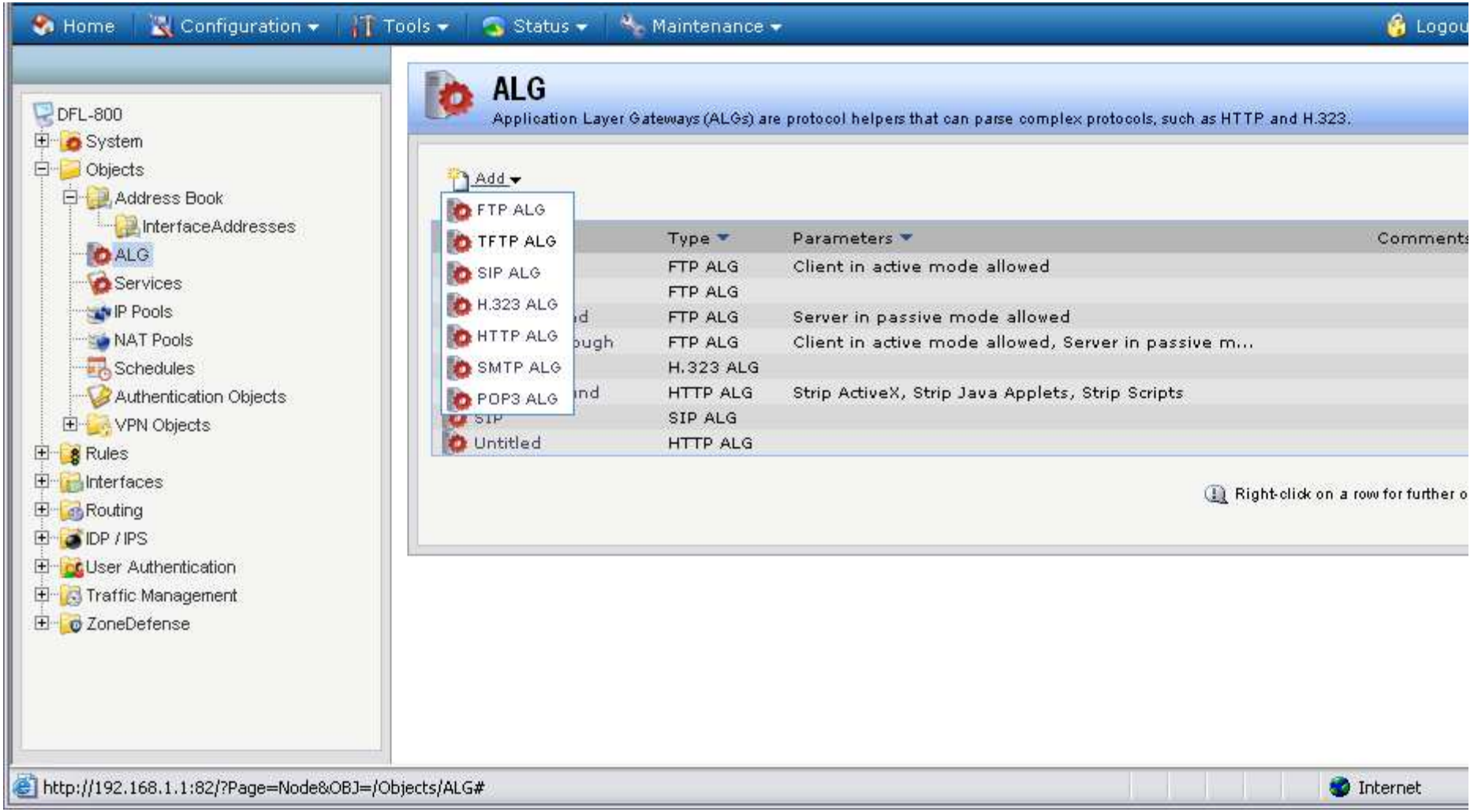
- Application Layer Gateway (ALG)
- Intrusion Detection System (IDS)

Alg

Доступны следующие компоненты:

- Ftp ALG
- Tftp
- Sip
- Smtp
- pop3
- H323 ALG
- HTTP ALG

Возможность ALG



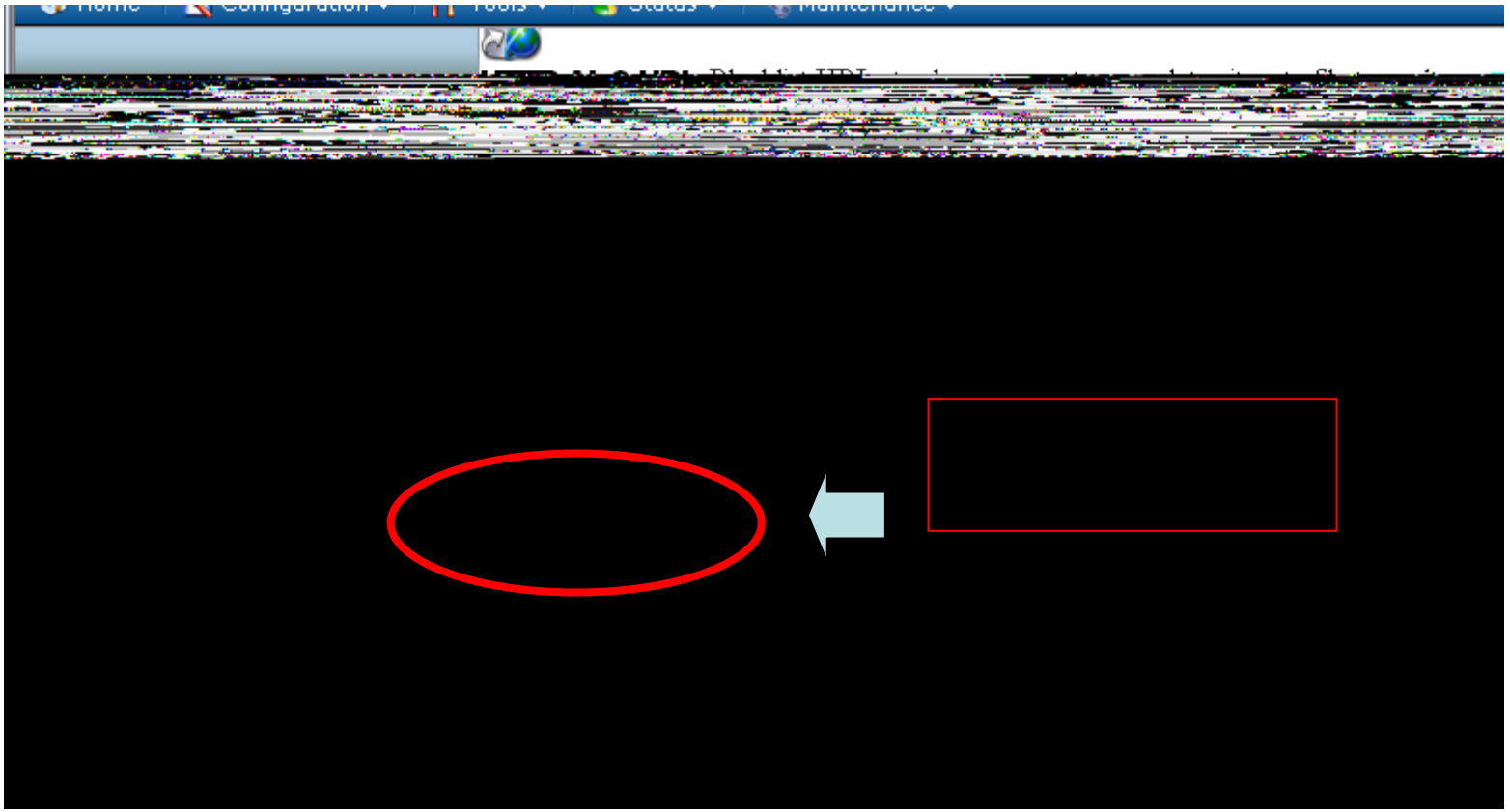
Http ALG

Позволяет удалять из содержимого HTTP трафика следующие объекты:

- Strip ActiveX objects (including Flash)
- Strip Java applets
- Strip Javascript/VBScript
- Block Cookies
- Verify that URL's does not contain invalid UTF8

Настраивается действие «да,если нашлось» или «нет, если нашлось»

А также создавать «фильтры адресов»



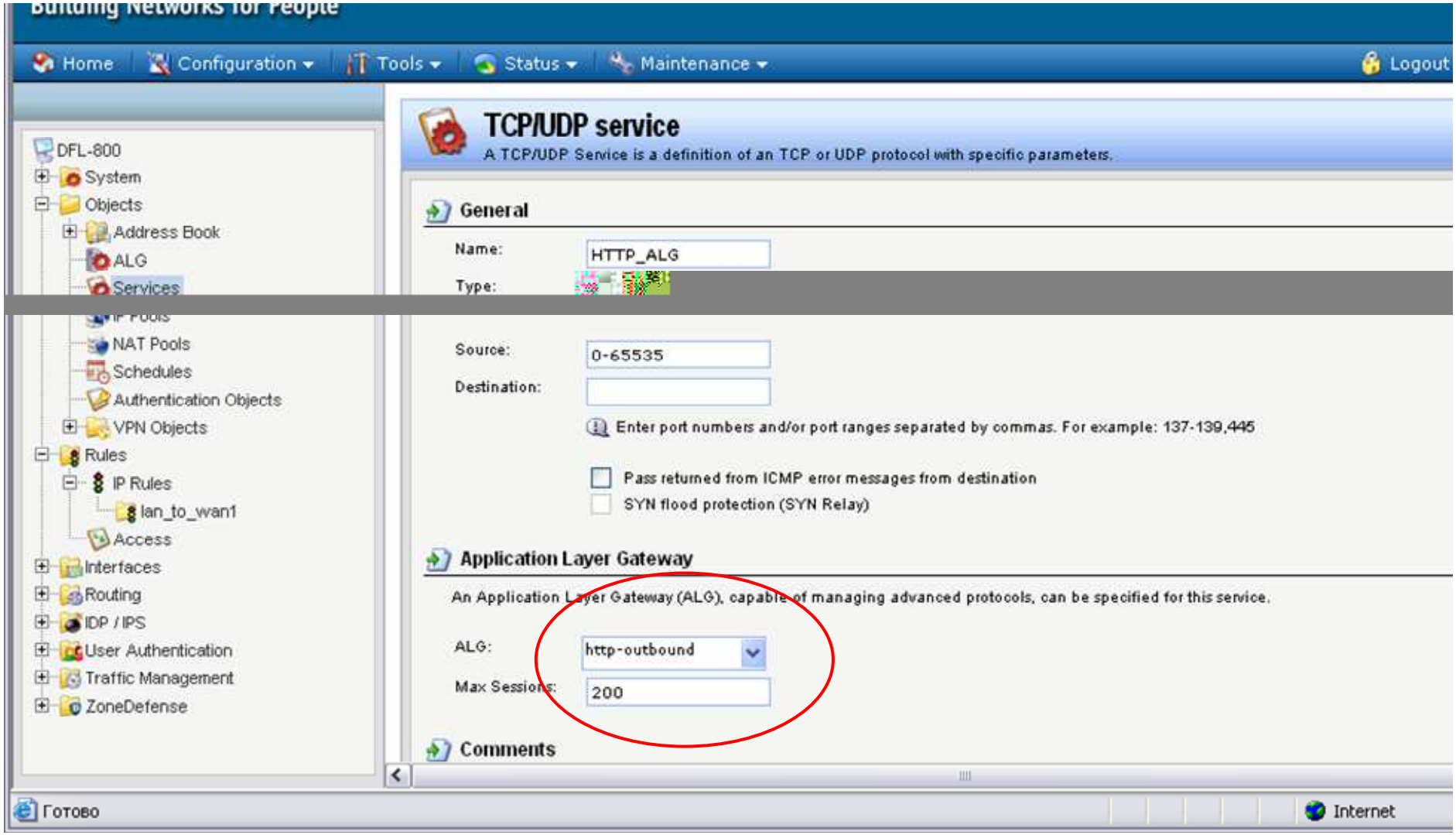
Некоторые другие фильтры:

- Smtp
  - Количество писем в минуту
  - Размер письма
  - Спам-фильтр (по заголовку,отправителю,DNS листу)
  - Анализ и болкировка

- SIP
  - Количество каналов
  - Таймауты и.т.д.
- POP3
  - Защита от некорректных команд.
  - Защита от перебора аккаунтов.
  - Анализ MIME

Возможности ALG пока не очень широки, однако производитель обещает наращивать их в следующих версиях прошивки.

Для активации ALG необходимо создать соответствующий объект «порт» и правила (Rules) для данного объекта



Intrusion Detection System

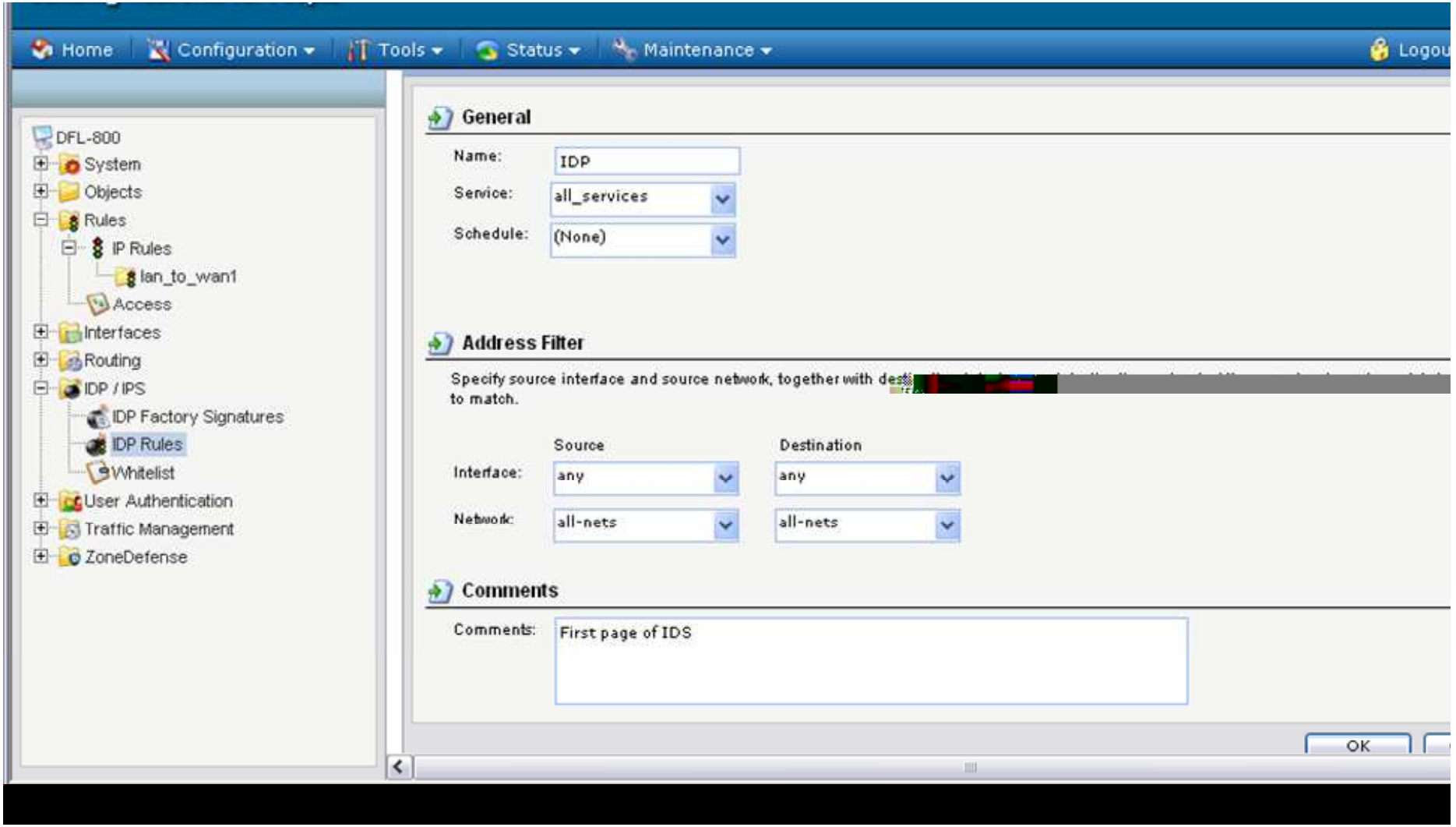
IDS анализирует сетевой трафик в поисках признаков сетевых атак, записанных в ее базе данных.

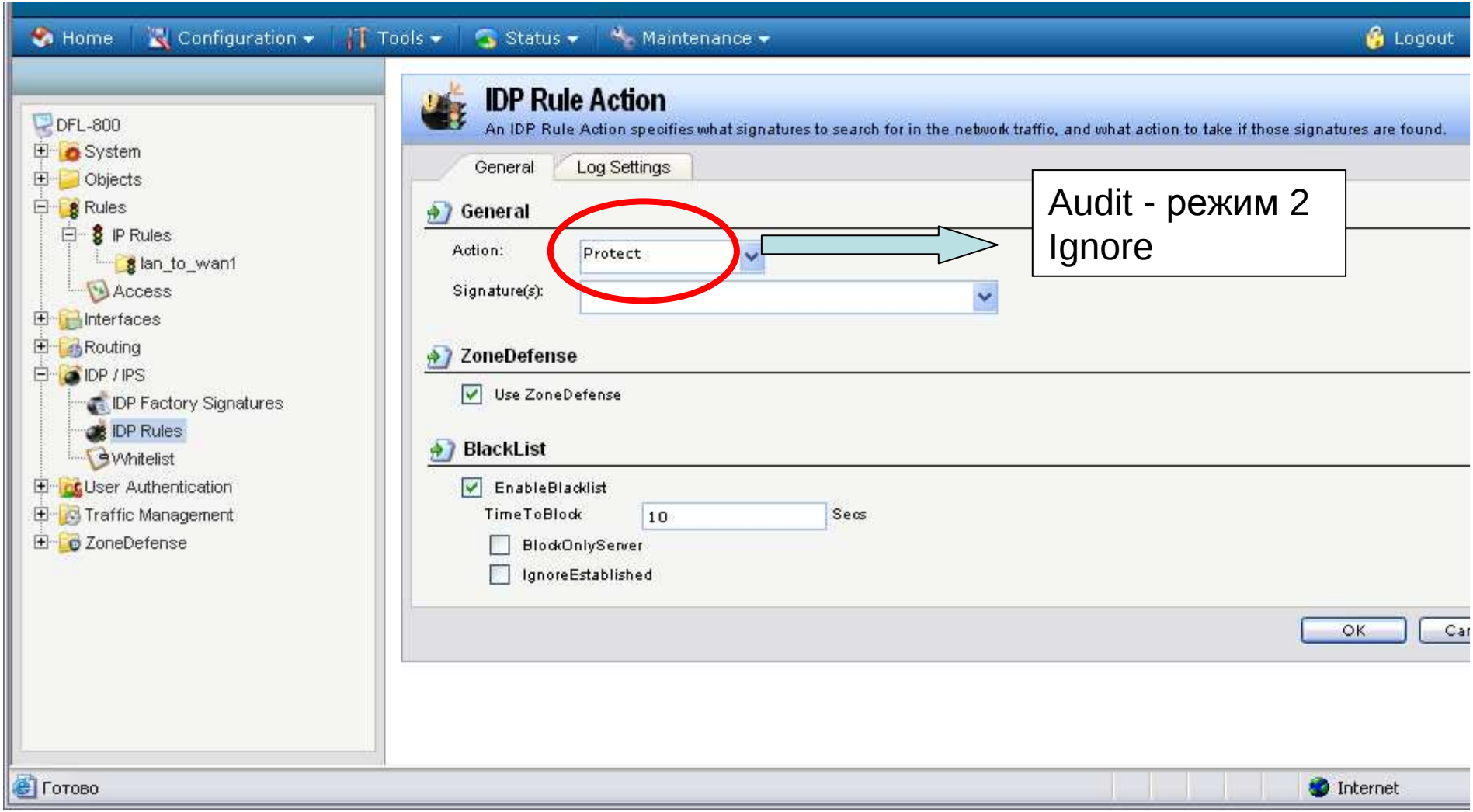
База данных автоматически обновляется через Интернет с сервера UpdateCenter.

Процесс обновления настраивается во вкладке

Mantence->UpdateCenter

- возможность настроить уведомление о атаке через E-mail системному администратору.
- возможность «отсечь» зону при помощи ZoneDefence (на коммутаторах DLINK DES 33XX и выше)
- Возможность заблокировать источник опасность при помощи blacklist на N секунд (работает в прозрачном режиме)


Home
Configuration
Tools
Status
Maintenance
Logou
DFL-800
System
Objects
Rules
IP Rules
lan_to_wan1
Access
Interfaces
Routing
IDP / IPS
IDP Factory Signatures
IDP Rules
Whitelist
User Authentication
Traffic Management
ZoneDefense
General
Name:
IDP
Service:
all_services
Schedule:
(None)
Address Filter
Specify source interface and source network, together with des
to match.
Source
Destination
Interface:
any
any
Network:
all-nets
all-nets
Comments
Comments:
First page of IDS
OK


Home
Configuration
Tools
Status
Maintenance
Logout
DFL-800
System
Objects
Rules
IP Rules
lan_to_wan1
Access
Interfaces
Routing
IDP / IPS
IDP Factory Signatures
IDP Rules
Whitelist
User Authentication
Traffic Management
ZoneDefense
IDP Rule Action
An IDP Rule Action specifies what signatures to search for in the network traffic, and what action to take if those signatures are found.
General
Log Settings
General
Action:
Protect
Signature(s):
ZoneDefense
Use ZoneDefense
BlackList
EnableBlacklist
TimeToBlock
10
Secs
BlockOnlyServer
IgnoreEstablished
OK
Audit - режим 2
Ignore
Готово
Internet

Zone Defense

Zone Defense позволяет блокировать определенные порты switch если IDP устройства обнаружило потенциальную уязвимость. Firewall загружает в коммутатор ACL (Access Control List) и коммутатор блокирует соединения с указанного порта. Порт остается в блокированном состоянии до тех пор, пока системный администратор не устранит причину блокировки и вручную не сбросит статус блокировки (через UI Firewall).

Zone Defence позволяет блокировать эпидемии сетевых червей в организации.

Блокировка по порогу

Помимо анализа IDP, решение о блокировке может быть принято и на основании задания простого порога соединений, измеряемого в соединениях в секунду. Порог может быть настроен:

- Source interface and source network.
- Destination interface and destination network.
- Service.
- Type of threshold: Host and/or network based.

TrafficManagment − > Threshold − > Add − > Threshold− > General:
Name: HTTP-Threshold
Service: HTTP
Address Filter
Source Destination
Interface: (the firewall's management interface) any
Network: 192.168.2.0/24(or the object name) all-nets − > Action:
Action: ZoneDefense
Host-based Threshold: 10

Коммутаторы, поддерживающие Zone Defense

- D-Link DES 3226S (minimum firmware: R4.02-B14)
- D-Link DES 3250TG (minimum firmware: R3.00-B09)
- D-Link DES 3326S (minimum firmware: R4.01-B39)
- D-Link DES 3350SR (minimum firmware: R1.02.035)
- D-Link DES 3526 (minimum firmware: R3.01-B23)
- D-Link DES 3550 (minimum firmware: R3.01-B23)
- D-Link DGS 3324SR (minimum firmware: R4.10-B15)

На коммутаторе необходимо корректно настроить IP адреса и разрешить SNMP управление. Для исключения конфликтов, таблицы ACL коммутаторов должны быть очищены.

Заключение

Сетевые экраны DFL-1600 2500

Отличия функциональности сетевых экранов серии 1600 и 2500 различаются, в основном, в количестве правил, быстродействии и скорости интерфейсов, однако идеологически в целом обладают той же функциональностью что и рассмотренный нами DFL-800. Рассмотрим некоторые дополнительные возможности:

- Расширение интерфейсов:
  - DFL-800 имеет 4 интерфейса : WAN1,WAN2,DMZ,LAN (10/100) (150 Мбит/с VPN 60 Мбит/с 25 000 сессий, политик 1 000)
  - DFL-1600 имеет 6 настраиваемых 1Gb интерфейсов (320 Мбит/с VPN 120 Мбит/с 400 000 сессий, политик 2500)
  - DFL-2500 имеет 8 1Gb интерфейсов (произв. 600 Мбит/с,VPN 300 Мбит/с 1 000 000 сессий)
- Расширение портов посредством технологии VLAN через 1GBit порт. Подключая управляемый коммутатор, можно добавить еще 16 интерфейсов.
- Возможность горячего резервирования (High Availability)

Обобщение

Мы рассмотрели некоторые возможности межсетевых экранов серии DFL

Перечислим их преимущества:

- Надежность
- Легкость настройки
- Достаточная гибкость
- Достаточно обширная функциональность.
- Достаточно высокий уровень защиты
- Низкое энергопотребление
- Возможность размещения в стойке коммутаторов
- Расширение функционала при совместной работе с коммутаторами DLINK
- Возможность быстрого восстановления конфигурации из резервного файла на «чистом» устройстве

Недостаток аппаратного межсетевого экрана очевиден. Расширить функциональность шире того ,что в него заложено производителем не представляется возможным. Однако функциональность сетевого экрана с лихвой покрывает все типичные запросы пользователей.