Краткий конспект лекций по курсу

«Компьютерные сети»

Модуль 1. Введение в компьютерные сети и ОС UNIX; уровневая организация сетевых протоколов модели OSI/ISO и стека TCP/IP; организация работы протоколов «внутренних» уровней стека TCP/IP; клиент-серверная организация сетевого взаимодействия программ и программные интерфейсы транспортного уровня.

1.1. Введение в компьютерные сети и ОС UNIX

Определение (назначение) компьютерных сетей (КС)

КС - совокупность компьютеров и других интеллектуальных устройств, объединенных сетью передачи данных для совместного решения ими различных прикладных задач.

Виды интеллектуальных устройств: компьютеры, встроенные контроллеры различных технических устройств и объектов, смартфоны, планшеты и пр.

Сеть передачи данных (СПД) – *каналы передачи данных* (различной природы) возможно соединяемые через промежуточные *коммуникационные устройства* (серверы, маршрутизаторы, коммутаторы, мультиплексоры, модемы, точки радиодоступа и пр.)

Сеть передачи данных – это сеть коммутации пакетов, в отличие от сетей с коммутацией каналов телефонных сетей, существовавших на момент создания СПД (1962 год Алан Клейнрок)

Примеры сетевых прикладных задач.

Структурная организация интернета. Различие физической структуры от структуры именования

Классификация КС (по масштабу, топологии, протоколам, доступности, функциям);

базовые топологии локальных вычислительных сетей (ЛВС): шинная, звездообразная, кольцевая, P2P; .

Краткая история развития КС: доинтернетовские КС, их основной недостаток; история создания и развития Интернет; параллели в истории Интернет и истории системы UNIX

обзор основных сервисов (услуг) сети Интернет.

Обзор классов основных сетевых служб

1.2. Уровневая организация сетевых протоколов модели OSI/ISO и стека TCP/IP

Определение сетевых протоколов

Сетевой протокол – это совокупность правил, методов, стандартов и реализующих их аппаратных и программных средств, совместно обеспечивающих взаимодействие компьютеров в компьютерной сети.

Уровневая организация сетевых протоколов и принципы взаимодействия открытых систем

- уровневая организация сетевых протоколов
- основные принципы взаимодействия открытых систем

Эталонная модель сетевых протоколов OSI/ISO;

(рассматривать снизу вверх)

7. Прикладной (Application layer)
6. Уровень представления данных (Presentation layer)
5. Сеансовый уровень (Session layer)
4. Транспортный уровень (Transport layer)
3. Сетевой уровень (Network Layer)
2. Уровень каналов данных (Data link layer)
1. Физический уровень (Physical layer)

Краткая характеристика уровней

Физический уровень

- Обеспечивает процесс передачи битового потока данных
- Определяет требования
  - К среде передачи данных (разновидности кабеля, радио среда (эфир) и т.д.)
  - К способу представления цифрового сигнала сигналом используемой среды передачи
  - К соединительным разъемам и аппаратной реализации приемо-передающих устройств
- Реализуются на всех устройствах, подключенных к сети

Уровень каналов данных

- *Предоставляет средства передачи* ***пакетов*** *данных между сетевыми устройствами в пределах* одного сегмента *сети*
- Обеспечивает обнаружение ошибок
- Может поддерживать исправление ошибок (только в некоторых технологиях)
- *Обеспечивает корректность переданных данных* (путем выбрасывания пакетов, для которых ошибки исправить не удалось)

Сетевой уровень

- Обеспечивает ненадежную передачу данных между сетевыми устройствами, находящимися в одной или разных сетях
- Выполняет маршрутизацию данных между сетями
- Может выполнять фрагментацию пакетов и их сборку из фрагментов

Транспортный уровень

- Обеспечивает надежную передачу данных ***между процессами,*** работающими на одном и том же либо на разных компьютерах сети
- Выполняет контроль ошибок, управление потоком и сегментирование

Сеансовый уровень

- Предоставляет средства для установки, ***поддержания*** (восстановления после случайного разрыва) и управления соединением между программными процессами на одном или различных компьютерах

Уровень представления данных

- Обеспечивает автоматическое преобразование формата передаваемых структурированных данных (например, порядок байт в слове, кодировка символов и т.п.)

Прикладной уровень

- Состоит из протоколов, обеспечивающие работу прикладных сетевых служб, с которыми напрямую взаимодействуют пользователи

Уровни протоколов TCP/IP и их соответствие модели OSI/ISO

Вместо сетевого уровня – уровень IP (с теми же функциями)

Сеансовый уровень и уровень представления данных включаются в прикладной. Примеры – ssh и P2P (сеансовый) и ftp (представления данных).

Понятие о стеке протоколов.

Схема прохождения пакетов данных через протокольные модули стека протоколов TCP/IP (с инкапсуляцией и возможной сегментацией пакетов) при взаимодействии компьютеров в сети.

1.3. Протоколы физического и канального уровней (драйверного уровня)

Технология Ethernet

Прототип – радиосети (ether – эфир, в том числе и радиоэфир)

Архитектура ЛВС с шинной топологией Ethernet.

Адресация компьютеров на физическом уровне (MAC-адреса)

(рассказал, но нужно дополнительно отметить, что MAC-адреса «зашиты» в ПЗУ сетевых карт

Формат кадра Ethernet:

преамбула (7байт) «1010 …10», признак начала кадра (1 байт) «10101011», заголовок кадра (14 байт), данные, CRC (контрольная сумма – 4 байта)

Заголовок кадра: MAC получателя, MAC источника, длина/протокол

Длина Заголовка кадра + CRC +18 байт, длина поля данных <= 1500 байт

Метод Множественного Доступа с Контролем Носителя и Обнаружением Столкновений МДКН/ОС (Carrier Sence Multiply Access with Collision Detection – CSMA/CD)

Логика работы сетевых карт Ethernet

а) при передаче кадра в шину

при необходимости передачи пакета сетевая карта «слушает» шину и передает пакет, в случае, если шина свободна. Если шина занята – сетевая карта ожидает освобождения шины и потом передает пакет. Если освобождения шины ждали несколько сетевых карт, то в начале передачи ими своих пакетов произойдет их «столкновение» и по сети будет распространяться сигнал, не совпадающий ни с одним из отправленных пакетов. Контроль столкновений выполняется путем сравнения содержания переданного в сеть пакета с содержанием реально передаваемого средой передачи сигнала. Реакция на столкновение: немедленное прекращение передачи с выжиданием случайного промежутка времени и повторения передачи пакета (с предварительной проверкой свободности среды)

Нюанс: при контроле столкновений фактически передаваемый по шине сигнал сравнивается с *известной преамбулой*; это существенно упрощает аппаратную реализацию контроля столкновений путем сравнения передаваемого сигнала с заранее известной последовательностью единиц и нулей.

Отметить также необходимость именно случайного время ожидания после обнаружения столкновения.

б) при приеме пакета из шины

- передаваемые через шину пакеты принимают все подключенные к ней сетевые карты

- в первую очередь анализируется длина пакета и слишком короткие пакеты выбрасываются (это «осколки столкновений»)

- затем анализируется адрес получателя: если он совпадает с собственным адресом или является широковещательным (рассказать, что это такое) – пакет принимается, иначе – выбрасывается.

Рассказать о возможности программной настройки сетевой карты на прием всех пакетов вне зависимости от адреса получателя. Эта возможность полезна при создании сетевых анализаторов трафика, но предоставляет опасность прослушки (sniffing), поэтому в одном сегменте шинной сети не должно быть компьютеров, интересы владельцев которых могут не совпадать или даже конфликтовать.

Привести житейскую метафору методу МДКН/ОС: разговор лиц в противогазах в темной комнате («слушай Вася, я Петя . . .»)

Аппаратное оборудование сетей Ethernet для различных типов среды передачи сигнала, разновидности технологии

Коаксиальный кабель (тонкий, со всеми типами коннекторов и терминаторами), витая пара (с объяснением того как устроены указанные типы кабелей и для чего это делается).

Отметить, что применение хабов не меняет топологию сегмента сети с шинной на звездообразную. Логика работы хаба в точности соответствует логике работы шины. Можно представить себе хаб, как шину, сжатую до размеров небольшой коробочки

Упоминание о радиосреде, оптической кабельной среде, оптической воздушной среде

Разновидности технологии (по скорости работы)

Достоинства и недостатки изначальной технологии Ethernet

По сути – это достоинства и недостатки шинной топологии

Достоинства:

логическая простота (и дешевизна оборудования),

малый расход кабеля (дешевизна кабельной системы).

Недостатки:

невысокая надежность (любое повреждение шины или сопряжений с ней влечет неработоспособность сети из-за появления отраженного сигнала, искажающего основной сигнал)

плохая масштабируемость (средняя скорость v доступа к компьютеру обратно пропорциональна их количеству N (v=V*2/N) где V – скорость шины

Мосты и коммутаторы Ethernet

понятие о мостах (bridge) и коммутаторах (switch): обеспечивают одновременную передачу данных между любыми парами сегментов, подключенных к портам коммутатора

коммутатор – это многопортовый мост , все порты которого используют одну и ту же технологию передачи данных (но скорости портов могут различаться)

Требования к скорости работы внутреннего передающего устройства коммутатора

Сеть, построенная на основе одного коммутатора, имеет топологию «звезда»

Алгоритм работы мостов и коммутаторов Ethernet (таблицы коммутации, способ их формирования в процессе обучения коммутатора)

преимущества коммутируемых сетей;

понятие о главных дополнительных возможностях, предоставляемых управляемыми коммутаторами:

протокол STP;

построение VLAN (обязательно упомянуть протокол IEEE 802.1q , соответствующий тег и его место в кадре (перед заголовком кадра);

базовые средства QoS (приоретизации пакетов, упомянуть протокол и тег IEEE 802.1p и его место в кадре).

Введение в оптоволоконные сети

принципы передачи информации по оптическому кабелю;

многомодовый и одномодовый оптоволоконные кабели;

основные фактроры, влияющие на качество передачи оптическогосигнала

построение сети Ethernet на базе оптоволоконного кабеля.

1.4. Межсетевой (IP) уровень

Основная функция этого уровня – ***маршрутизация пакетов*** промежуточными сетевыми устройствами сети при их пересылке из одного физического сегмента сети в другой.

Коммутируемая сеть Ethernet –это один сегмент с общей (коммутируемой на 2-м уровне) средой передачи. Такой сегмент не может иметь слишком больших масштабов (в этом случае, в частности, замедлится до недопустимого скорость работы STP и произойдет ряд других неприятностей).

Маршрутизация между сегментами выполняется специальными устройствами, называемые маршрутизаторами (router) и выполняющими роль межсетевых шлюзов (gateway) с использованием специальных таблиц маршрутизации, создаваемых на каждом из промежуточных маршрутизаторов.

Функции маршрутизаторов могут выполнять компьютеры с установленным на них соответствующим ПО или так называемые аппаратные маршрутизаторы.

Основной протокол этого уровня – межсетевой протокол IP.

Межсетевые IP-адреса

Неприемлемость MAC-адресов на роль глобальных адресов в многосегментных сетях: структура адресного пространства MAC-адресов отражает лишь структуру (перечень) производителей сетевого оборудования, но никак не коррелирует со структурой сети

Корреляция структуры адресного пространства со структурой сети необходима для того, чтобы уменьшить до обозримого размер маршрутных таблиц, используемых для маршрутизации пакетов.

В простейшем случае адрес межсетевого уровня, структура которого коррелирует со структурой сложной сети, должен включать как минимум 2 поля, идентифицирующие подсеть и компьютер в этой подсети.

Поэтому в начальной версии протокола IP (IPv4) межсетевой адрес (IP адрес) – это 4 байтная структура, содержащая 2 поля, используемые в роли требуемых идентификаторов: *номер подсети* и *номер компьютера в* этой *подсети*.

Форма записи IP-адреса

4 десятичных числа (от 0 до 255), разделенные символом «.». Каждое из этих чисел – значение соответствующего по порядку байта IP-адреса

Классы IP-адресов

Но размеры этих полей не фиксированы, а различны для различных классов IP-адресов. Для различения классов сети в начале структуры создается 3-е поле: признак класса сети. Длина этого поля также различна для разных классов

Потребность во введении классов IP-адресов. При небольшой длине IP-адреса невозможно разбить этот адрес на 2 поля с фиксированной длиной так, чтобы одновременно допускать хотя бы сотни тысяч различных подсетей с сотнями тысяч компьютеров в некоторых из них.

Структура IP-адреса

Пр. класса	№ подсети	№ компьютера в подсети
1-4 бита	1-3 байта *)	3-1 байт

*) за вычетом 1-4 бит признака класса адреса

Таблица классов IP-адресов

Класс	Битовый признак класса/ диапазон значений 1-го байта	Длина поля № подсети (в байтах)	Количество подсетей	Длина поля № компьютера (в байтах)	Количество компьютеров в сети
A	0 / 0-127	1 *) **)	126	3	2^{32}-2 ***)
B	10 / 128-191	2 *)	2^{14}	2	2^{16}-2 ***)
C	100 / 192 - 223	3 *)	2^{29}	1	254 ***)
D	1110 / 224 - 239	Номер группы			
F	1111 240 -255	Служебный класс			

Сеть 127 (IP адрес 127.0.0.1)

unicast, multicast и broadcast адреса

*) за вычетом 1-4 бит признака класса адреса
**) сеть 127 зарезервирована
***) номера компьютеров, состоящие из всех нулей и всех единиц являются служебными

Маска подсети и ее назначение и формы записи

В целях экономного использования доступного адресного пространства зачастую требуется разбить подсеть некоторого класса на несколько сетей меньшего размера (каждые физический сегмент должен быть оформлен как IP подсеть, а компьютеров в сегменте может быть, например, менее десятка).

Кроме того, для уменьшения размеров таблиц маршрутизации существует обратная потребность в объединении (агрегации) нескольких смежных сетей некоторого класса в одну общую сеть.

Обе эти задачи (разбиение на подсети и агрегация подсетей) решаются с использованием маски подсети. Маска подсети – это 4-х байтное значение, позволяющее задать положение границы между полями номера сети и номера компьютера в IP-адресе с точностью до одного бита.

Значение маски подсети – это последовательность битов «1» (количество таких бит равно требуемой длине поля номера подсети (вместе с признаком подсети), остальные биты – нули.

Пусть M – маска подсети, а A – IP адрес Тогда $N_{сети} = A\&M$, а $N_{комп} = A\&(not(M))$

Формы записи маски:

1) 4 десятичных числа от 0 до 255, разделенные точками, $N=2^n$

2) «/»N, где N – количество бит со значением «1»

Простое правило вычисления последнего числа в маске при разбиении подсети класса С:

$Б_4 = 256-S$, где S – ближайшая степень двойки, превосходящая размер требуемой подсети.

Тогда маска = 255.255.255. $Б_4$

Присвоение сетевым интерфейсам IP-адреса и маски подсети

Если компьютер (маршрутизатор) имеет один или несколько сетевых интерфейсов (сетевых карт), то каждому из этих интерфейсов должны быть назначены (сетевым администратором) его IP-адрес и маска подсети.

Если компьютер (маршрутизатор) сети подключен к нескольким сегментам сети (с использованием соответствующих сетевых интерфейсам, то каждому из этих интерфейсов (а значит – и сегментам, с которыми они соединены) должны быть назначены разные номера подсетей (каждый сегмент должен быть оформлен, как отдельная IP-сеть со своим уникальным номером).

Для присвоения сетевому интерфейсу IP-адреса и маски подсети используется команда ifconfig.

Упомянуть о протоколе DHCP, предназначенном для динамического выделения и присваивания

Основные поля заголовка IP-пакета

Адрес-получателя,
адрес источника,

длина пакета (2 байта),
протокол более высокого уровня (2 байта),
TTL (1 байт),
DS-байт (включает поле TOS)

Преобразование логических IP-адресов в физические адреса с использованием протокольного модуля ARP

Схема взаимодействия модуля IP с драйверами сетевых интерфейсов включает также взаимодействие каждого из драйверов с экземпляром модуля ARP, обслуживающим этот драйвер (нарисовать картинку)

Когда модуль IP обращается к драйверу сетевого интерфейса для передачи IP-пакета, то он передает драйверу 1) пересылаемый IP-пакет 2) IP-адрес получателя пакета (возможно – промежуточного шлюза), *гарантированно* расположенного в сегменте, обслуживаемом этим драйвером. Для преобразования 2-го параметра в MAC адрес получателя драйвер обращается к своему модулю ARP, который ведет ARP таблицы (в начале – пустые), задающие соответствие IP и MAC адресов. Если требуемый IP адрес находится, то он передается драйверу. Если нет – организует рассылку специального широковещательного запроса (рассказать формат запроса, включая указание кода протокола ARP в заголовке кадра) и прием ответа (включая анализ кода протокола) на этот запрос для определения требуемого IP адреса и занесения строки в таблицу. После рассылки запроса (но еще до получения ответа) драйверу возвращается код ошибки, по которому тот выбрасывает пакет (ожидание не организуется, чтобы не усложнять реализацию) . Но следующий пакет к тому же адресату уже будет отправлен правильно.

Маршрутизация пакетов при их передаче между сетями

Межсетевые шлюзы – компьютеры (маршрутизаторы, соединяющие несколько сегментов сети.

Порядок выполнения маршрутизации протокольным модулем IP,

Структура таблицы маршрутизации, строка default.

IP-адрес подсети / маска	Признак косвенной адресации	IP-адрес шлюза	Идентификатор интерфейса	Метрика (hops)
(куда?)	прямо/ транзитом	(через какой шлюз?)	(через какой интерфейс?)	Расстояние до подсети
	0			
default	1			

Прямая и косвенная адресация: адресат находится непосредственно в одном из сегментов, подключенных к маршрутизатору (прямая) или доступен через промежуточные шлюзы.

Признак косвенной адресации может вычисляться через IP-адрес получателя и IP-адрес интерфейса шлюза (несовпадение подсетей в этих адресах), поэтому в некоторых реализациях IP этот признак в таблице маршрутизации явно не задается

Статическое (адинистратор) и динамическое (специальные протоколы) формирование маршрутов (строк таблицы).

Использование команды route для создания/удаления строк таблицы.

Агрегация соседних подсетей с использованием маски подсети.

Если маршрутизация некой подсети отлична от маршрутизации включающей ее более крупной агрегированной сети (имеющей более короткую маску), то строка для этой подсети должна быть задана раньше, чем строка для агрегированной сети (для сетей с общим префиксом номера сети те из них, что имеют более длинную маску должны указываться первыми)

Основы протокола ICMP

Предназначен для контроля работоспособности сети на уровне IP.

Включает ряд специальных пакетов icmp echo (request, reply), traceroute (допускает длину маршрута не более 8) и др.

Реализация полноценной (без ограничений на длину маршрута) команды traceroute с использованием icmp echo (циклическое echo с возрастающим (от 1) TTL)

Команды анализа состояния сети и маршрутов (ping, traceroute).

Маршрутизаторы и их функции

Основные (маршрутизация, управление маршрутизацией).

Понятие о дополнительных функциях:

фильтрация пакетов,

шейпинг,

IP-туннелирование,

трансляция адресов (NAT)

(рассказать и о внутренних («серых») IP- адресах (3 диапазона

10.0.0.0 - 10.255.255.255 (/8);
172.16.0.0- 172.31.255.255 (/12) и
192.168.0.0 –— 192.168.255.255 (/16)

<u>Назначение и основные особенности протокола IPv6</u>

Основные недостатки IPv4:

а) ограниченность (нехватка) пространства IP адресов,

б) сложности масштабирования (высокая загрузка маршрутизаторов),

в) ограниченный функционал (слабая поддержка службы QoS и средств ИБ).

Эти недостатки устранены в IPv6.

Основные особенности:

1) Новая длина (16 байт) и форма записи IP адреса
(пары байт в 16-й форме (по 4 цифры) через «:»)

2) 3 уровня иерархии (вместо1-го) подсетей (уменьшение размера таблиц маршрутизации), нашедшие отражение в формате глобального IP- адреса:

Format Prefix (тип адреса),

TLA (Top Level Aggregation),

NLA (Next Level Aggregation),

SLA(Site Level Aggregation) ,

Interface ID

3) Использование MAC адреса в качестве Interface ID, отказ от ARP (снижение нагрузки на маршрутизаторы)

4) Введение нового типа адресации anycast (поддержка source routing, уменьшающего нагрузку на маршрутизаторы)

5) Введение нового гибкого формата заголовка IP-пакета, включающего основной заголовок с дополнительными полями, ориентированными на поддержку QoS, и ряд необязательных дополнительных заголовков, поддерживающих, в частности, средства обеспечения ИБ, source routing и др. возможности

В итоге недостатки устранены

а) за счет 1

б) за счет 2-5

в) за счет 5

Возможности плавного перехода от IPv4 r IPv6.

1.5. Протоколы транспортного уровня.

В соответствии с моделью OSI/ISO функцией протоколов этого уровня является обеспечение надежной доставки пакетов

В семействе TCP/IP имеется 2 протокола транспортного уровня: TCP и UDP.

При этом только протокол TCP обеспечивает надежность доставки.

И оба этих протокола обеспечивают функцию установления соединения через сеть между парой процессов.

Виртуальные каналы и дейтаграммные соединения.

Виртуальный канал – соединение «из конца в конец», обеспечивающее надежную доставку данных с сохранением их порядка (поток-ориентированную передачу данных)

Дейтаграммное соединение – «соединение» по *пересылке пакетов* между двумя *конечными точками*, при котором часть пакетов может «теряться», и при котором порядок поступления пакетов к принимающей стороне может отличаться от порядка отправки пакетов.

Порты и их назначение

Порты – это специальные псевдоустройства (программно реализуемые над реальными устройствами ввода/вывода через сеть – сетевыми интерфейсами), предназначенные для передачи данных в сеть и приема из сети *прикладными процессами*. Порты идентифицируются своими номерами, «привязанными» к прикладным процессам.

Отправка и прием данных через порты выполняются прикладными процессами с использованием специальных программных интерфейсов транспортного уровня (будут рассмотрены в ближайших лекциях).

Протокол TCP

Протокол TCP обеспечивает надежное двунаправленное поток-ориентированное соединение типа «виртуальный канал» между парой процессов, работающих на различных или на одном и том же компьютере сети.
В поток-ориентированном соединении информация о границах между пакетами в процессе передачи теряется: читать данные можно порциями другого размера, отличного от размера отправляемых пакетов.

При установлении TCP-соединения процессы *«обмениваются рукопожатием»*. Инициатор соединения посылает партнеру пакет *запроса на установления соединения* и получает от партнера *пакет подтверждения соединения*.

При разрыве соединения его инициатор посылает партнеру *пакет разрыва соединения* и получает в ответ *соответствующее подтверждение*.

Партнеры по TCP соединению идентифицируют друг друга с указанием IP-адреса компьютера и № порта. Информация об IP-адресах и №-ах портов источника и получателя TCP-пакетов (транспортных сообщений) задается в заголовках этих пакетов, вместе с некоторой дополнительной информацией, рассматриваемой ниже.

Методы поддержания виртуального канала

Для поддержания виртуального канала требуется
1) обеспечивать *контроль потерь пакетов* и оперативную *повторную посылку потерянных пакетов*
2) обеспечить *восстановление правильного порядка следования пакетов* т.е. выдачу содержимого пакетов принимающему приложению в порядке отправки (а не приема) пакетов

Для реализации 1-й функции в ответ на каждый пакет посылается уведомление о его получении. При этом по тайм-ауту контролируется оперативная доставка каждого пакета. Если по истечении периода тайм-аута после отправки пакета уведомление о его доставке не поступило – пакет отправляется повторно.
До этого момента (или до более раннего момента получения уведомления) пакет должен сохраняться в буферной памяти на стороне отправителя пакетов.
Для установления соответствия между пакетом и уведомлением о его доставке *в заголовок этих пакетов должно входить поле*, однозначно идентифицирующее его.
В качестве значения такого поля удобно выбрать *№ байта* с момента установления соединения.

Для реализации 2-й функции каждый пакет, «обогнавший» своих предшественников необходимо сохранять в буферной памяти получателя до момента получения всех предшественников и последующего «чтения» содержания всех предшественников и самого пакета. Для определения правильного порядка следования пакетов можно воспользоваться указанным в его заголовке №-ом пакета.

Размер окна TCP

При отправке пакетов можно отравлять очередной пакет не дожидаясь получения подтверждений нескольких предыдущих пакетов. Это позволяет увеличить суммарную

скорость передачи пакетов во столько раз, сколько неподтвержденных пакетов допускается.
Суммарный размер неподтвержденных пакетов называется *размером окна TCP*

Как отмечалось выше – увеличение размера окна TCP повышает суммарную скорость передачи.
Но, с другой стороны, при увеличении размера окна TCP влечет увеличение буферной памяти как на стороне источника, так и на стороне получателя. На1-й стороне буферная память необходима для сохранения всех неподтвержденных пакетов, на 2-й для сохранения всех пакетов, обогнавших своих предшественников.

Кроме того, путем увеличения размера окна TCP нельзя повышать скорость бесконечно. Рано или поздно эта скорость «упрется» в ограниченную скорость наиболее медленного канала на маршруте от источника к получателю. Как только скорость передачи достигнет скорости работы этого канала дальнейшее увеличение окна TCP приведет к отбрасыванию части пакетов на входе этого канала.

Для автоматического регулирования размера окна TCP используется *механизм обратной связи TCP*. Суть этого механизма состоит в том, что на стороне отправителя анализируется процент потерянных пакетов. Если это процент превосходит некоторое верхнее пороговое значение – размер окна TCP автоматически уменьшается, что приводит к снижению скорости отправки пакетов. Если этот процент становится меньше некоторого нижнего порогового значения – размер окна TCP автоматически увеличивается, что влечет увеличение скорости передачи.
Таким образом, механизм обратной связи TCP позволяет установить для каждого соединения скорость передачи, адаптированную к скорости работы наиболее медленного канала маршрута (с учетом уровня его текущей нагрузки). Это позволяет, в частности, минимизировать очереди на маршрутизаторах.

Протокол UDP

Протокол UDP обеспечивает ненадежную дейтаграммную блок-ориентированную доставку пакетов от процесса-источника к процессу-получателю.

Фактически этот протокол не реализует функции обеспечения надежности, предоставляя лишь функции организации взаимодействия сетевых прикладных процессов.

Блок-ориентированный режим передачи, обеспечиваемый UDP, означает, что на стороне получателя «чтение» (прием) данных из сети должно выполняться блоками такого же размера, как и отправка пакетов их источников.
Если будет прочитан блок меньшего размера, то остаток отправленного блока будет потерян.

Если попытаться прочесть блок размера большего, чем отправленный, то в параметре «длина» будет возвращена длина реально принятого блока.

Вопрос к аудитории:
Зачем нужен «нехороший» UDP при наличии «хорошего» TCP

Достоинства и недостатки протокола TCP

Достоинство TCP – надежность и удобство соединения типа «виртуальный канал»
Недостатки:
1) большие накладные расходы для коротких (особенно – однопакетных) соединений. Дополнительно к основному пакету данных надо передать еще 5 пакетов: 2 на установление соединения с подтверждением, 1 – подтверждение получения основного пакета, 2 – разрыв соединения с подтверждением.
2) большие задержки целых *групп пакетов* при потере и повторной пересылке *единичных пакетов*, предшествующих этим группам (пакеты задерживаются в буферной памяти), что недопустимо для аудио и видео приложений

Примеры областей эффективного применения протокола UDP

UDP эффективен для тех приложений, для которых недостатки протокола TCP являются неприемлемыми.

Примеры
1) Протокол SNMP.
Управляющие станции (УС), выполняющие мониторинг состояния компьютерных сетей средствами SNMP периодически через короткие промежутки времени отправляют однопакетные запросы всем компьютерам и др. интеллектуальным устройствам контролируемой сети, в ответ на которые получают однопакетные ответы.
Устанавливать постоянные IP-соединения накладно (связано с большим объемом буферной памяти, требуемой большому числу соединений.
При установке же соединения лишь для обмена парой пакетов предаются 6 дополнительных пакетов (2 – установление соединения с подтверждением , 2 – подтверждения, 2 разрыв соединения с подтверждением). В итоге объем трафика возрастает в 4 раза.
В то же время УС может «спокойно пережить» потери отдельных пакетов. Если пакет потерян случайно, то аналогичный пакет, отправленный через короткий промежуток времени с очень высокой вероятностью не потеряется. Если потеря пакета не случайна – она свидетельствует о неполадках в сети, которые и должен обнаруживать SNMP.

2) Передача видеопотока.
Если каждый пакет видеопотока содержит инфформацию, по которой может быть вычислена позиция экрана, в которую отображается информация пакета, то при использовании UDP будет обеспечено более качественной воспроизведение изображения,

чем при использовании TCP. Это связано с тем, что потеря в UDP отдельных пакетов приводит к практически незаметным очень кратковременным искажениям незначительной части экрана. При использовании же TCP потеря каждого пакета будет приводить к задержке целой группы пакетов и, соответственно, к заметной задержке в обновлении изображения на более обширной части экрана.
Возможно также применение различных приемов, обеспечивающих приемлемый уровень качества интерполяции пропущенного цифрового сигнала. Так если в паре смежных пакетов в первом из них отправлять только четные отсчеты (нумерация с 0), а во 2-м – нечетные, то при потере одного из пакетов достаточно хорошей хорошей его интерполяцией будет замена потерянного пакета доставленным.

1.6. Общая организация прикладных служб

При взаимодействии 2-х компьютеров в сети обычно это взаимодействие не является симметричным. Как правило, один из таких компьютеров, называемый клиентским компьютером обращается к другому с запросом на выполнение некоторой услуги (service). Компьютер, предоставляющий запрошенную услугу, как правило, делает это по запросу любого клиентского компьютера сети и называется сервером. Организация предоставления серверным компьютером некоторой услуги называется службой (предоставления этой услуги). Рассмотренная схема предоставления услуг сервером произвольному множеству клиентских компьютеров называется *клиент-серверной организацией* соответствующей сетевой службы.

Подавляющее большинство сетевых служб имеет клиент-серверную организацию.

В отличие от клиентских процессов, которые запускаются и прекращаются их пользователями серверный процесс «несения своей службы» должен быть постоянно готов к приему и обработке запросов какого-то из клиентов. Это означает, что серверный процесс сетевой службы должен запускаться во время загрузки ОС сервера и постоянно быть в готовности к обработке.
Такие процессы, запускаемые при загрузке операционной системы и постоянно готовые к выполнению своей службе (предоставлении соответствующих услуг) называют *демонами (daemon).*
Демоны, «несущие» какую-либо сетевую службу называют сетевыми демонами.

Для серверов, выполняющих общеизвестные службы сети (Well Known Services-WKS), включающие такие службы как www (http), ftp, mail (SMTP) и др. номера портов таких служб должны быть известны всем клиентам сети, а значит – они должны быть фиксированными.
Соответствие между службами сети (протоколами прикладного уровня) и используемыми серверными процессами этих служб портами в системе UNIX описывается в файле /etc/services

Ввиду того, что количество WKS очень велико, держать демоны всех этих сетевых служб постоянно запущенными накладно по использованию ресурсов серверного компьютера. Поэтому для сетевых демонов придумали схему такого их запуска, при которой «постоянную вахту» несет лишь один специальный демон называемый *супердемоном*. Имя этого супердемона – inetd (сокращение от InterNET Daemon).

Супердемон запускается при загрузке операционной системы и постоянно «слушает сеть», т.е принимает все запросы на соединение, поступающие из сети. По номеру порта из пакета запроса с помощью файла /etc/services он определяет имя сетевой службы (сетевого протокола), которой направлен запрос.
Затем по имени службы с использованием файла /etc/inetd.conf определяется имя исполнимого файла с кодом демона требуемой службы, этот демон запускается и ему переадресуется принятый запрос на соединение. При разрыве соединения запущенный экземпляр демона завершается (и освобождает все выделенные ему ресурсы).

1.7. Программные интерфейсы транспортного уровня

Прикладные процессы, взаимодействуют с транспортным уровнем с использованием программных интерфейсов (API) транспортного уровня.
Существует несколько стандартов таких API.

Наиболее известными и распространенными из них являются программные интерфейсы Sockets, первоначально разработанные в составе Берклиевской Free BSD и в последствие реализованные в составе других ОС. В Частности в MS Windows эти интерфейсы реализованы в составе библиотеки WinSock.

Другим известным стандартом является TLI, реализованный в составе SystemV UNIX. Несмотря на то, что эти интерфейсы являются более наглядными, чем Sockets, они не получили широкого распространения в виду того, что к моменту их разработки соответствующая «экологическая ниша» была уже основательно занята интерфейсами Sockets.

Программные интерфейсы Sockets

Для именования этих программных интерфейсов используют также термины «гнезда» или «разъемы», являющиеся переводом исходного термина, который означает буквально «электрические разъемы/гнезда»

Используемая метафора: Прикладные сетевые процессы устанавливают сетевое соединение друг с другом через сеть с использованием API транспортного уровня, подобно тому, как различные электрические приборы соединяются через кабели,

«оконеченные» электрическими разъемами – Socket-ами (которые часто называют просто сокетами).

Socket – это специальный объект, используемый в программах для идентификации сетевых соединений между процессами в сетевых операциях, выполняемых через эти соединения, подобно тому, как дескриптор файла используется в программах ввода/вывода в файл для идентификации этого файла.
Реально этот объект имеет целочисленные значения, интерпретируемые ОС как индекс в таблице открытых соединений. Каждое соединение в этой таблице описывается определенной структурой данных, содержащей как минимум информацию об IP-адресах и номерах портов участников соединения.

Далее будут рассмотрены:
- требуемые заголовочные файлы,
- системный вызов создания сокета,
- системные вызовы для работы с сокетами,
- схема взаимодействия системных вызовов при двух типах сетевых соединений (виртруальный канал и дейтаграммное соединение),
- вспомогательные системные вызовы (gethostbyname, gethostbyaddr, getservbyname, getservbyaddr)

Заголовочные файлы, требуемые для работы с сокетами

```
#include  <sys/types.h>
#include  <sys/socket.h>
```

Создание сокета

int s = socket(***int*** domain, ***int*** type, ***int*** protocol)
Возможные значения domain: AF_UNIX, AF_INET (пояснить смысл). Указанные значения – это имена целочисленных констант, определенных в <sys/socket.h>
Возможные значения типа соединения type:
SOCK_STREAM, SOCK_DGRAM, SOCK_SEQPACKET
Protocol: TCP, UDP, 0 (0- по умолчанию для потоковых соединений выбирается TCP, для дейтаграммных – UDP.

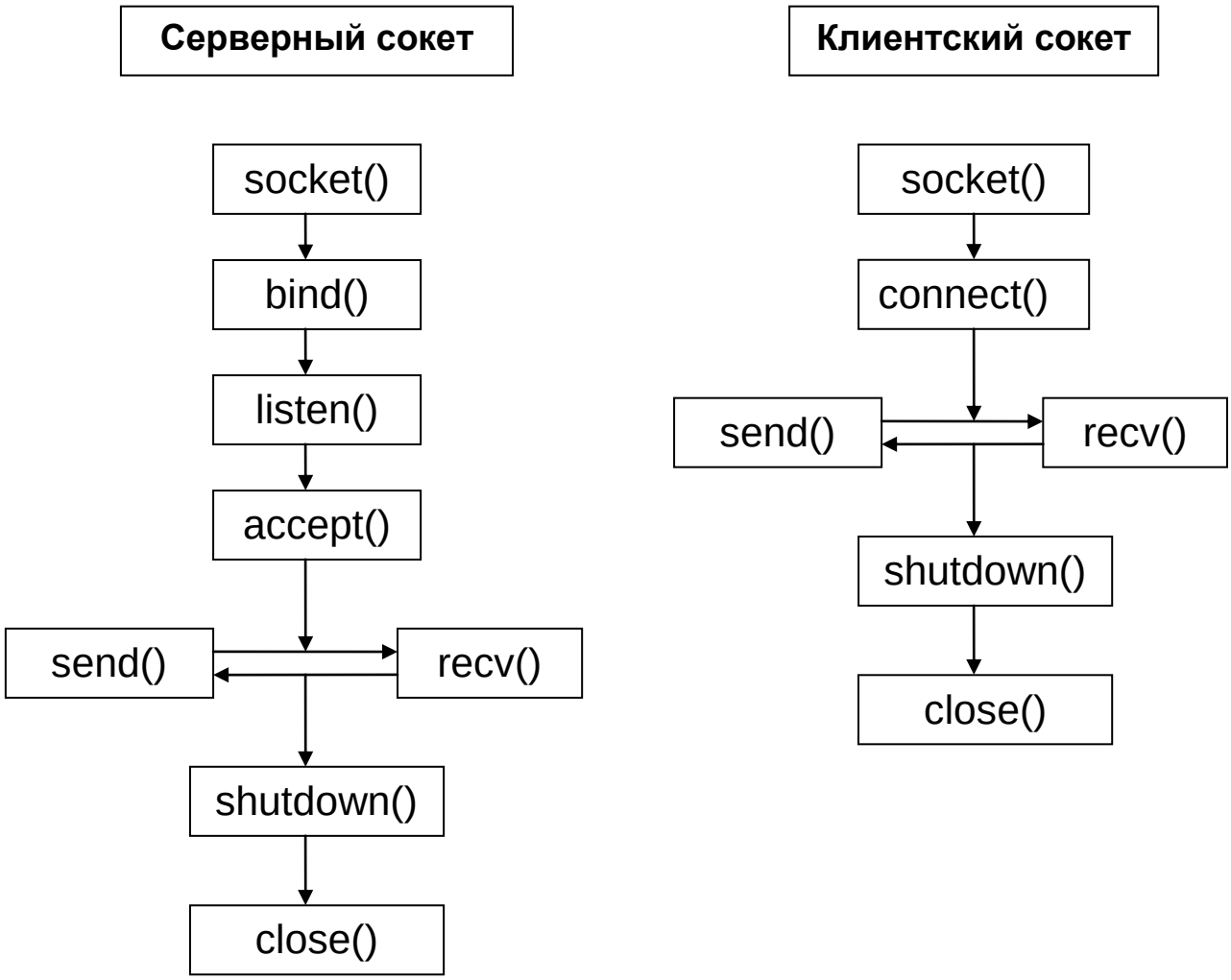
Назначение системных вызовов для работы с сокетами:

socket	– создание сокета с определенными параметрами,
bind	– связывание сокета с конкретным локальным процессом (IP-адрес, № порта) (для серверных процессов
listen	– ожидание запроса к серверному процессу и организация очереди запросов,
connect	– запрос клиентского процесса на установление соединения,
accept	– прием установления соединения серверным процессом,

send, sendto – отправка сообщения в указанный сокет,
recv, recvfrom – прием сообщения из указанного сокета,
shutdown – закрытие соединения для чтения и/или записи,
close – закрытие сокета

Схемы взаимодействия системных вызовов по работе с сокетами

Схема клиент-серверного взаимодействия сокетов через потоковое соединение



Примечание: для обеспечения параллельной обработки запросов следует после accept, создающего новый сокет, вызвать fork, который в родительском процессе возвращает управление на начало accept.

Схема взаимодействия сокетов через дейтаграммное соединение

socket

bind
sendto < === > recvfrom (дважды)
shutdown
close

Форматы системных вызовов

Функция bind
int bind (***int*** s, ***struct sockaddr**** addr-p, ***int*** len);
Функция связывет с сокетом локальный IP-адрес и номер порта серверного процесса (для AF_INET, для AF_UNIX должно быть указано путевое имя файла)

Функция listen
int listen (***int*** s, ***int*** size), возвращает код возврата 0 или -1
Создает буфер для приема не более size запросов на установление соединения

Функция connect
int connect (***int*** s, ***struct sockaddr**** addr-p, ***int*** len)
Через ***struct sockaddr**** addr-p необходимо указать адресную информацию серверного процесса, IP-адрес клиента будет вставлен автоматически, № порта клиента сгенерируется динамически (может быть запущено несколько однотипных клиентов на одном компьютере).

Функция accept
int accept (***int*** s, ***struct sockaddr**** addr-p, ***int*** len)
Через ***struct sockaddr**** addr-p серверный процесс получает информацию об адресе и порте клиента. Своя адресная информация занесена в эту структуру вызовом bind.

Функция send
int send (***int*** s, ***const char**** buf, ***int*** len, ***int*** flag) (можно использовать и write (s))
значение flag – 0 или MSG_OOB, в последнем случае сообщение должно быть передано в срочном (Out Of Band) режиме. Возвращает число преданных байт.

Функция sendto
int sendto (***int*** s, ***const char**** buf, ***int*** len, ***int*** flag, ***struct sockaddr**** addr-p, ***int*** len_p)
это комбинация connect и send.

Функция recv
int send (***int*** s, ***const char**** buf, ***int*** len, ***int*** flag) можно read (s)
Параметры – аналогично send-у

Функция recvfrom
(***int*** s, ***const char**** buf, ***int*** len, ***int*** flag, ***struct sockaddr**** addr-p, ***int*** len_p)
это комбинация accept и recv.

Функция shutdown
int shutdown (***int*** s, ***int*** mode)
mode: 0 – закрыть на чтение, 1– закрыть на запись, 2 – закрыть на чтение и на запись.

Функция close
close (***int*** s) уничтожает сокет.

Следует отметить о буферизации и использовании функции fflush(s).

2. ПРОТОКОЛЫ И СЛУЖБЫ ПРИКЛАДНОГО УРОВНЯ

2.1. Службы именования компьютеров в сети

2.1.1. Старая система именования компьютеров

Файл hosts.txt в ARPANet
хранился на сервере Стэндфордского исследовательского института SRI
и раз в сутки скачивался на все компьютеры сети

В системах UNIX – это файл /etc/hosts
Его записи имеют вид
IP-address cname [aliases (псевдонимы)] (cname – каноническое (основное) имя)
Обычно единственной строкой этого файла является строка
127.0.0.1 localhost

Недостатки: очень плохая масштабируемость (сложность обеспечения уникальности имен, большой объем работы по администрированию БД имен).

2.1.2. Доменная служба имен DNS

Доменная система имен

Структура доменного пространства имен.
Рассказать об иерархической системе имен, основанной на иерархии доменов (domain)
Домены mil, gov, edu, com, org, net
Домены стран: uk, nw, sw, fr, it, su, ru, ua
Прочие домены: например, mob
Кириллические домены РФ

Традиции именования серверов прикладных служб: давать серверу псавдоним, совпадающий с названием службы.
Интуитивный способ «вычисления» имен серверов

Служба DNS

Проект службы предложен в 1983 году Полом Мокапетрисом
BIND – Berkeley Internet Name Domain) первый демон сервера DNS, реализованный в 1984 году 4-мя студентами из Калифорнийского университета города Беркли

Общая организация DNS

Зона DNS – множество всех компьютеров, непосредственно входящих в домен

Каждую зону DNS должны обслуживать не менее 2-х *серверов DNS*.
При этом каждый из серверов может обслуживать несколько зон.

Серверы DNS могут быть классифицированы по их *роли* и их *организации*

По их роли серверы DNS могут быть *первичными* (primary) или *вторичными* (secondary)

По организации серверы DNS могут быть:
- главными (master)
- ведомыми (slave)
- только кэширующими (caching only)
- ресольверами (resolver, remote server)

Первичный (и только первичный) сервер должен быть всегда главным и единственным. Он может обслуживать несколько зон. На первичном сервере находится исходная версия полная БД информации о зоне.

Вторичных серверов может быть несколько, по организации они могут быть ведомыми (содержат копию БД первичного сервера) или только кэширующими (их БД – дисковый кэш результатов ранее выполненных (путем обращения к главному или ведомому серверу) запросов)

Ресольверы – не имеют своей БД и всегда обращаются для получения результата запроса к другим серверам DNS. Устанавливаются на клиентских компьютерах для

Зачем нужны дополнительные (сверх 2-го) вторичные серверы (в «полу изолированных» подсетях)

Программная организация DNS
1) демон named (bind или другие реализации демона имен)

2) конфигурационные файлы (БД) DNS (приведен состав БД UNIX System V, в других версиях UNIX состав и имена файлов могут отличаться)
/etc/named.boot (задает тип сервера, место расположения его БД и пр.)
Обычное место расположения БД DNS – каталог /etc/namedb/
В нем размещаются следующие файлы (имеющие общий формат записей)
root.cache – информация о серверах корневой зоны DNS
local – информация о компьютере, на котором размещен сервер DNS
hosts – информация о компьютерах обслуживаемых зон
rev – информация для преобразования IP-адресов в доменные имена (IN-ADDR.ARPA)

На ресольвере есть только файл /etc/resolve.conf

Логика работы сервера DNS

1) Проверка локальности имени (принадлежности к одной из зон ответственности) и, для локального имени – выдача ответа на запрос.
2) Для нелокального имени – обращение с рекурсивным запросом к одному из 13-ти корневых серверов Интернет с запросом на получение адреса DNS сервера, ответственного за интересующую зону 1-го уровня.
3) возможные рекурсивные запросы для получения адресов DNS серверов, ответственных за зоны 2-го и последующего уровней.
4) Обращение к серверу имен требуемой зоны для определения требуемого IP-адреса
Рассмотреть на примере www.sfedu.ru

Логика обратного преобразования (с использованием IN-ADDR.ARPA)

Формат записей файлов БД DNS

Каждый из файлов БД DNS (во всех версиях ОС)может содержать записи 2-х типов: SOA и RR

Запись SOA – *Start Of Authority* (начало зоны ответственности) находится в начале последовательного сегмента файла, все дальнейшие записи которого имеют формат RR и описывают ресурсы (компьютеры и их роли) зоны.
Запись имеет ряд параметров, одним из главных среди которых является номер версии информации о зоне. Значение этого параметра используется для синхронизации баз данных всех видов ведомых серверов с БД главного сервера

Запись RR (Resource Record – запись о ресурсе)
Формат RR
[имя] [ttl] класс-адреса тип {данные, специфичные для типа}
Правила умолчания для параметров
Пример

www	IN	A	192.15.208.01

Основные типы записей о ресурсах

NS – запись указывает, что соответствующий компьютер является сервером имен
A – address – запись задает IP-адрес компьютера
AAAA – информация об IPv6 адресе компьютера
HINFO – информация, специфичная для компьютера, например тип его ОС
CNAME – каноническое имя
PTR – указатель; объяснить использование (домен IN-ADDR.ARPA)
MX – Mail eXchanger – почтовый сервер зоны

Интерфейсы доступа к информации сервера имен

1) команда (утилита) nslookup

Использование (простейший формат)

nslookup {доменное имя | IP-адрес}

Если задано имя, перчатает IP-адрес, если

2) API gethostbyname, gethostbyaddr

по заданному имени или IP-адресу компьютера заполняют специальную структуру информацией о всех именах (каноническом и псевдонимах) и IP-адресах (их столько, сколько есть сетевых интерфейсов) компьютера.

2.2. Протоколы удаленного терминала

Назначение – обеспечение удаленного использования ресурсов удаленного вычислительного сервера путем «превращения» терминала локального компьютера в терминал удаленного сервера.

Разновидности

1) Протоколы удаленного символьного терминала (telnet, rlogin,rsh, ssh)

2) Протокол удаленного графического терминала (X11)

2.2.1. Протокол telnet

Демон – telnetd, клиент – программа (команда) telnet (другие варианты имени клиента)

Клиентская программа telnet

Формат команды вызова telnet

telnet [IP-адрес | доменное имя]

Рассмотрим вызов без параметров.

В результате ввода команды появляется приглашение для ввода команды:

telnet>

Команды telnet: '?', '? com-name', open, close, quit, **char**, **line** и др. Способ сокращения имени.

По команде «open имя-сервера»

появляется приглашение к вводу логина и пароля.

После их ввода – сообщение о подключении к серверу с указанием его доменного имени и типа ОС и сообщение

“ Escape character is ‘^[‘ ” – пояснить смысл.

После этого появляется приглашение

telnet>

Главный недостаток: пароль передается в открытом виде. Поэтому

при подключению к серверу через telnet *нельзя указывать логин root* (“use su instead”)/

Другие недостатки – внутренняя «многословность» из-за системной независимости и необходимости согласования параметров партнеров по взаимодействию.

2.2.2. Протокол rlogin

Демон – rlogind, клиент – программа (команда) rlogin (принадлежит группе r-команд)
Формат вызова
rlogin [ключи] доменнное-имя-сервера
Если не настроены специальные *файлы автологина* – запрашиваются логин и пароль, если настроены – соединение выполняется без ввода логина и пароля.
При установлении соединения на сервере вызывается экземпляр командного интерпретатора, обслуживающего удаленный терминал
Стандартная команда выхода из интерпретатора “exit” разрывает соединение.
Файлы автологина
/etc/hosts.equiv
/home/user-name/.rhosts

Ключ ‘-u’ при вызове rlogin

2.2.3. Протокол rsh

Демон – rshd, клиент – программа (команда) rsh
Формат вызова
rsh [ключи] “команда”
Должны выполняться условия автологина.

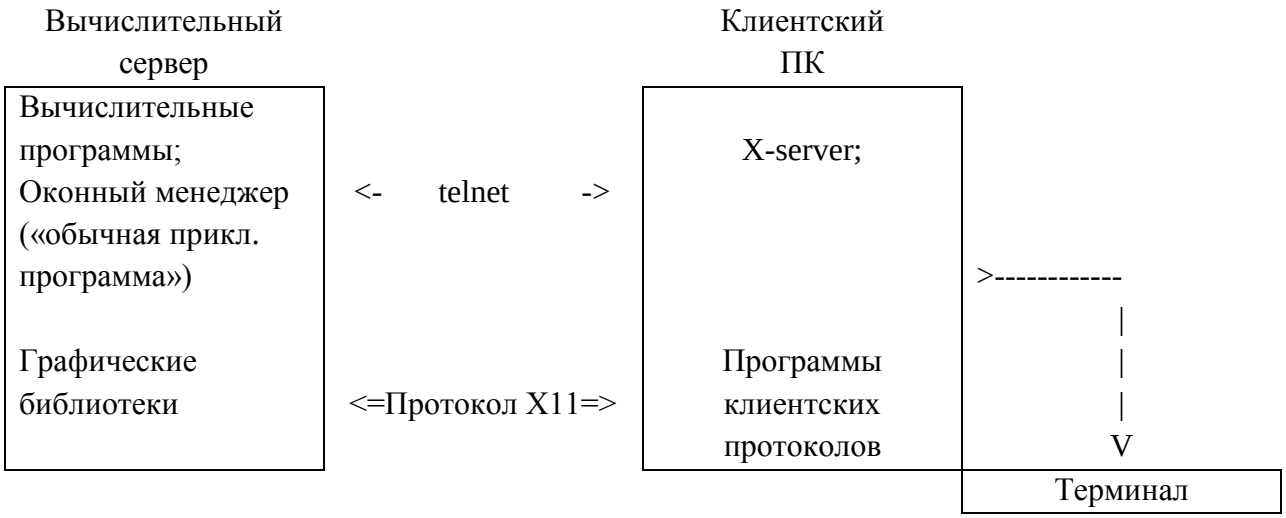
2.2.4. Протокол ssh

Защищенный аналог telnet, работающий над SSL
Пояснить принципы несимметричных схем шифрования на основе открытого и закрытого паролей.

2.2.5. Протокол удаленного терминала X11

2.2.5.1. Организация оконной системы xWindows ОС UNIX

(разработка MIT, 1984 год)



Сценарий установления графического соединения

1) установить удаленное терминальное соединение с вычислительным сервером
2) выполнить на сервере команды

Export DISPLAY; DISPLAY=имя-кл-комп:0
Export OPENWINHOME; OPENWINHOME=/usr/openwinhome
xterm – ls &
olwm &

Понятие о протоколе XDMCP

2.3. Протоколы пересылки файлов

ftp, sftp, tftp, rcp, scp

ftp

Авторизованный вход, анонимный сервер ftp (правила авторизации)

Джентльменский набор команд
Доступ к ФС с правами подключившегося пользователя

Файлы
/etc/ftpusers
/home/login/.netrc

Недостатки: передача пароля по сети в открытом виде

sftp
Построен над SSL

tftp
Упрощенный аналог ***ftp,*** работает над UDP без авторизации
Права доступа: только к общедоступным файлам и каталогам в соответствии
С правами доступа «для прочих пользователей»
Набор команд
Недостаток: большие проблемы ИБ
Современное применение – загрузка ядра ОС бездисковых компьютеров

rcp
Формат вызова
rcp [ключи] файл1 файл2

Имя каждого из файлов может иметь вид *имя-компьютера : путевое-имя-файла*

scp – rcp над SSL

2.4. R-команды

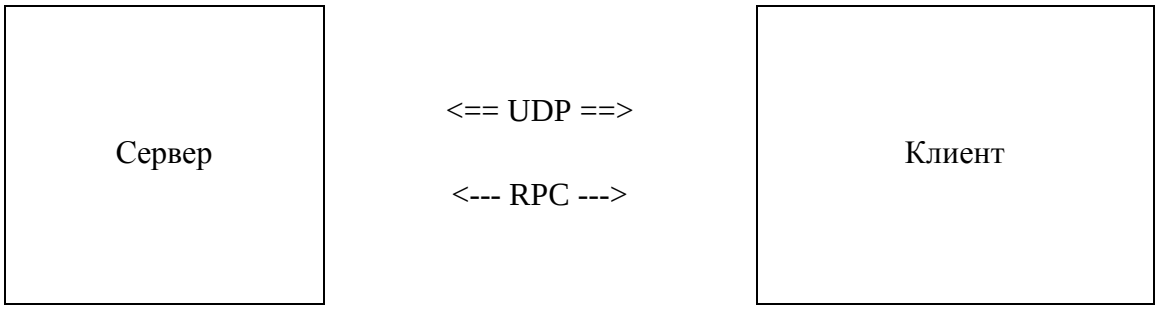
rlogin, rsh
rcp
rwho - выдает информацию о пользователях компьютеров локальной сети
/var/rwho/rwhod.*
ruptime

finger [имя-польз]@имя-компьютера

2.5. Сетевая файловая система NFS

(Разработка Sun Microsystems, 1984 год)

Обеспечивает прозрачный доступ к файлам других компьютеров путем логического «встраивания» каталогов удаленных компьютеров в дерево иерархии локальной файловой системы



Экспортирует каталоги

Монтирует каталоги

Демон: nfsd (8 &)

Демон: mountd

Конф. файл
/etc/fstab

Конф. файл
/etc/mounttab

Команда
exportfs

Команда
mount

/etc/fstab

имя-ф-спецустр каталог тип ФС режимы
/dev/sd1 /usr/bin NFS (-access:имена, -rw, -root и др.)
ufs

Команда
exportfs { -a | каталог [режимы экспорта]}

/etc/mounttab

имя-сервера каталог-сервера точка-монтажа [режимы монтажа]

Права доступа к файлам пользователей NFS - по UID
Проблема синхронизации UID
Назначение системы NIS (праобраз AD)

2.6. Организация службы электронной почты

Появилась в UNIX с момента ее создания, до реализации в UNIX протоколов TCP/IP работала над UUCP – старые сетевые протоколы UNIX. Эти протоколы потом еще много лет использовались для передачи почты по коммутируемым телефонным каналам.

Команды mail, mailx UNIX
Кодировка ASCII-7, применение uuencode и uudecode для других кодировок и двоичных файлов

Современные клиентские программы электронной почты (их возможности)

Организация взаимодействия клиентских компьютеров и почтовых серверов
при пересылке почтовых сообщений (с использованием почтового сервера отправителя и почтового сервера получателя.
Картинка
Отправка письма: от клиента своему серверу – по SMTP, между почтовыми серверами – по SMTP.
Прием – со своего почтового сервера по протоколам POP3/IMAP

Демон sendmail – демон почтового сервера
Логика работы sendmail

Формат почтового сообщения
Заголовок, тело, формат MIME, формат HTML

Почтовый СПАМ и методы борьбы с ним
Определение СПАМа
Фильтрация СПАМа на стороне сервера и клиента
Ошибки 1-го и 2-го рода
Некоторые приемы в фильтрации СПАМа

2.6. Служба всемирной информационной паутины WWW

Понятие гипертекста (гипермедиа)

URL

http – протокол пересылки гипертекста
Язык представления гипертекста html: язык описания формата гипертекстовых страниц с теговой структурой
Теговая структура: вложенные «скобочные» конструкции вида
<имя возможные-параметры> возможный-вложенный-гипертекст </имя>
Ссылочный тег <a href=”URL”> изображение-ссылки” </a href >
Понятие о редакторах html
Понятие о CMS: Drupal, Joomla)

Методы построения «активных» HTML-страниц:
Разновидности динамических эффектов:
динамические бланки, анимационные эффекты и др.

Понятие о CGI, CGI-скрипты, скриптовые языки PHP и Perl$
Мобильные языки (Java);
flash-технологии , плагин, файлы .swf (Smal Web Format)

Понятие о web-сервисах

Понятие об Интранет-технологиях
Понятие об Интранет-технологиях.
Тонкий и толстый клиенты, достоинства тонких клиентов.

Назначение и принципы функционирования поисковых серверов.
Почтовые роботы, индексатор, поисковая машина.
Ранжирование найденных ответов, рейтинг сайтов

3. СИСТЕМНЫЕ ПРОТОКОЛЫ И СЛУЖБЫ КОМПЬЮТЕРНЫХ СЕТЕЙ. ОСНОВЫ БЕЗОПАСНОСТИ КОМПЬЮТЕРНЫХ СЕТЕЙ

3.1. Протоколы управления маршрутизацией

Напомнить структуру таблицы маршрутизации.
Метрика, значение «бесконечность»

Статическая и динамическая маршрутизация

Назначение протоколов управления маршрутизацией;

Организация обмена маршрутной информацией на базе протокола RIP
Протокол класса векторов расстояний (в hop'ах)

Алгоритм Беллмана-Форда
На всех маршрутизаторах (узлах)
1) рассылать свои таблицы маршрутизации всем соседям с периодичностью 30 сек
2) при получении таблица от соседа для строк локальной таблицы маршрутизации и принятой таблицы сопоставить пары строк с одинаковым значением IP-адреса подсети, и для каждой пары сопоставившихся строк выполнить алгоритм
2.1) Если в качестве шлюза в локальной строке указан сосед,
то безусловно поменять метрику Млок:= Мсоседа+1
2.2) Иначе Если Мсоседа+1 < Млок
заменить информацию в локальной строке информацией о соседе
(Шлюз-лок :=IP-адр- соседа, Млок:= Мсоседа+1

Формально доказано что
1) алгоритм сходится
2) алгоритм сходится к оптимальным таблицам при любых начальных значениях метрик (при условии, что метрики до соседей указаны правильно)

Проблема «ложных маршрутов возрастающей длины при разрыве соединения» (картинка)
Метод борьбы –
1) никогда не верить информации о себе, полученной от соседа
Проблема сохраняется при наличии циклических маршрутов
2) конечное значение «бесконечности» (16, 32)

Недостатки протокола RIP

0) относительно большая дополнительная нагрузка на сеть (пересылаются полные таблицы маршрутизации)
1) медленная сходимость: проблема масштабирования сетей
2) невозможность распределения трафика между несколькими равноценными каналами
3) невозможность учета уровня текущей загрузки каналов, влияющего на скорость передачи данных (самый короткий, но перегруженный маршрут, может быть медленнее чем более длинный, но с низкой загрузкой)
Чтобы бороться с последним недостатком нужны *динамические метрики*

Основы протокола OSPF

OSPF – значит Open (открытый, не требующий лицензирования) SPF - таково название алгоритма Дейкстры (Shortest Path First – первым выбирать кратчайший путь)
Протокол с *динамической метрикой*, протокол *состояния связей*
Возможно несколько метрик канала. Но в каждый момент может использоваться только одна текущая метрика.
Администратор конфигурирует выбор текущей метрики
Основные метрики:

1) *Метрика скорости доставки пакета* обратно пропорциональна пропускной способности канала. Обычно значение этой скорости для одной связи (иногда называемой стоимостью порта) вычисляется как отношение некоторой эталонной пропускной способности (reference bandwidth - B_R) к пропускной способности канала B: $M = B_R / B$. При этом в коммуникационном оборудовании компании Cisco (коммутаторах и маршрутизаторах) $B_R = 10^8$.
Таким образом, скорость связи через канал Fast Ethernet составит 1, а, например, очень медленного канала 64 Кбит/с – 1562,5.
Но в оборудовании некоторых производителей $B_R = 10^9$
Значение B_R может конфигурироваться.

2) *Динамическая метрика свободной пропускной способности* вычисляется по формуле, похожей на формулу расчета предыдущей метрики $M(t) = B_R / B_F(t)$, где $B_F(t)$ - незанятая (free) в данный момент времени t пропускная способность канала. Очевидно, что при использовании этой метрики наименьшее значение метрики будет у наименее загруженных каналов, что устраняет 4-й недостаток протокола RIP. Поскольку эта метрика явно зависит от времени, она является динамической. Именно благодаря возможности использования этой метрики протокол OSPF относят к классу протоколов с динамическими метриками.

3) *Метрики параметров QoS*. В настоящее время не применяются.

Для всех типов метрик полагается, что значение метрики, равное $2^{24}-1 = 16777215$, не достижимо ни при каких условиях. Это значение используется в качестве признака недоступности канала или сети. Для любого типа метрик метрика маршрута является суммой метрик составляющий этот маршрут связей.

Логика работы протокола OSPF

Каждый маршрутизатор при каждом изменении метрик каналов рассылает *лавинообразно* своим соседям информацию об этих изменениях по сети (рассылаются не полные таблицы, а только информация об изменениях)

Каждый маршрутизатор на основании информации, полученной от других маршрутизаторов, строит размеченный граф маршрутов сети, и с использованием алгоритма Дейкстры (нахождения кратчайших маршрутов в размеченном графе) вычисляется каждая строка таблицы маршрутизации. При изменении графа – таблицы маршрутизации перевычисляются.

Отметим, что для каждой метрики строятся отдельные маршрутные таблицы, но модули IP работает с маршрутными таблицами текущей метрики.

Основные достоинства OSPF

- быстрая сходимость
- способность выбирать самые быстрые маршруты
- возможность распределения трафика между несколькими равноценными каналами
- возможность направления трафика различных прикладных протоколов по различным маршрутам (пример быстрого спутникового видео и медленного наземного telnet)
- (модернизированный вариант модуля IP, поэтому OSPF иногда относят к протоколам 2-го уровня).

Недостаток: квадратичная сложность алгоритма Дейкстры влечет определенное ограничение пределов масштабирования сетей, обслуживаемых протоколом OSPF. Метод, позволяющий частично преодолеть этот недостаток - “раздвинуть» пределы масштабирования – зонная организация OSPF-сети: вся сеть организуется в виде звезды сетей, называемых зонами. В центре звезды находится магистральная зона. Полные графы строятся только внутри зон. Пути между маршрутизаторами, входящими в различные зоны (например – периферийные) конструируются из участков путей к пограничным мапшрутизаторам соответствующих зон и пути между пограничными маршрутизаторами периферийных зон. В перифирийных зонах могут использоваться другие протоколы управления маршрутизацией.

Б'ольших пределов масштабирования не требуется, поскольку OSPF является внутренним протоколом маршрутизации.

Другие протоколы управления маршрутизацией *класса состояния связей* – протокол состояния связей **IS-IS**

Демоны протоколов управления маршрутизацией

routed, gated

Понятие о внутренних и внешних протоколах маршрутизации

Протоколы RIP, EIRGP, OSPF и IS-IS являются внутренними протоколами маршрутизации.
Они могут использоваться для управления маршрутизацией в сетях, находящихся под единым административным управлением.
Такие сети называются доменами маршрутизации
Основная особенность доменов маршрутизации состоит в том, что внутри этих доменов все маршруты являются допустимыми: не существует никакого ограничения на возможные маршруты пересылки данных.
Автономная система (AS) – домен маршрутизации, регистрируются в RIPE NCC.
Примером AS является AS 5480, принадлежащая ЮФУ
Внешней маршрутизацией является междоменная маршрутизация (маршрутизация между AS).
Управление внешней маршрутизацией выполняется на основе протоколов внешней маршрутизации BGP или EGP (устаревший протокол).

В междоменной маршрутизации не все маршруты через другие домены допустимы.
Для того, чтобы какая-то AS разрешила передачу через себя трафика других AS, руководство этих AS должно договориться об условиях передачи транзитного трафика и формально зафиксировать эти договоренности средствами протокола внешней маршрутизации BGP посредством сообщения соседним AS допустимых путей к другим AS через текущую AS (BGP – *протокол состояния путей*).

Для каждой AS определяется
1) множество IN – множество AS, смежных с данной, из которых возможен прием трафика этих AS и, возможно проходящего через них трафика более удаленных AS.
2) множество OUT – множество AS, смежных с данной, в которые возможна передача трафика данной AS и, возможно транзитного трафика более удаленных AS

Для того, чтобы разрешить соседней IN AS передавать в данную транзитный трафик других AS, данная AS должна выдать этой соседней IN AS анонсы, т.е. перечень путей к AS, трафик которых данная AS согласна принимать. При этом действует простое правило: я Вам анонсы – Вы мне трафик.
Принятый трафик может передаваться в OUT AS, если из нее поступили соответствующие анонсы. Анонсируются *пути* к AS, в которые возможна передача транзитного трафика.
Поэтому протокол BGP относят к классу *состояния путей*.
(Картинка с примерами анонсов между связными AS)

BGP работает на пограничных маршрутизаторах сетей, на которых средствами этого протокола ведутся мировые таблицы маршрутизации, Информация об изменении междоменных маршрутов рассылается лавинообразно сразу же после изменения состояния какого либо из внешних каналов и/или выставления новых анонсов. На каждом из пограничных маршрутизаторов оптимальная таблица маршрутизации строится с использованием алгоритма Дейкстры на базе графа логической связности AS.

Отметим, что протокол BGP включает 2 протокола: EBGP (Exterior – внешний) и IBGP (Interior – внутренний).
Средствами EBGP строятся таблицы маршрутизации, используемые при пересылке трафика между AS;
Средствами IBGP строятся таблицы маршрутизации, используемые при пересылке транзитного трафика через AS (между ее различными пограничными маршрутизаторами).

3.2. Основы управления сетями на базе протокола SNMP

SNMP – Simple Network Management Protocol (Простой протокол управления сетью).

Управляемая сеть – управляющие станции и агенты. Управляющих станций может быть несколько.
Если на управляющей станции установлено специальное ПО мониторинга сети, то такую станцию часто называют сервером управления сетью, хотя фактически соответствующее ПО является клиентом всех агентов (которые с точки зрения клиент-серверной организации программ являются серверами).

Агенты устанавливаются на ВСЕ интеллектуальные устройства сети. В их число наряду с компьютерами, маршрутизаторами и коммутаторами могут входить и другие интеллектуальные устройства, например, точки доступа WiFi, модемы, сетевые принтеры и даже интеллектуальные UPC, Сплит-системы и другие устройства.

Стандарт протокола SNMP состоит из 3-х основных частей: SMI, MIB и собственно протокол передачи пакетов

SMI – Structure of Management Information
Стандарт определяет
- правила идентификации объектов сети и
- базовые типы данных, используемые при работе с контролируемыми устройствами сети:
IP-Addres, Counter, TimeTicks и др.
- Составные объекты данных (группы переменных), рассматриваются ниже

MIB – Management Information Base – «База данных» контролируемого объекта, структурирована в соответствии со SMI
Основная особенность этой БД – особый способ интеграции с ядром ОС контролируемого объекта.
Каждому полю данных этой БД соответствует какой-то параметр ядра ОС или поле из какой-то структуры данных ядра ОС.
При этом
1) Изменение параметра в ядре приводит к синхронному изменению соответствующего параметра в MIB
2) изменение параметра в MIB приводит к синхронному изменению соответствующего параметра в ядре ОС.
Таким образом, ***путем (удаленного) изменения*** MIB можно ***менять структуры данных ядра ОС***

MIB, в соответствии со SMI, включает ряд стандартных поименованных групп переменных, состав которых может быть расширен при помощи специального протокола SMUX (SNMP MUltypleXing).
В число основных групп входят группы
system, interfaces, ip, icmp, arp, tcp, udp, egp, snmp и некоторые другие

Пример имени переменной группы interfaces (If) – If.Adminstatus2
Пример имени переменной группы ip – Ip.Route.NextHop.192.236.208.32

Собственно-протокол

Пакеты PDU фиксированного размера, пересылаемые по UDP через порты 161 и 162

Содержимым пакетов пересылаемых от управляющей станции могут быть
1) команды группы get (getone, getnext, getmany, snmpstat и др.)
2) команды группы put

Содержимым пакетов пересылаемых от агентов могут быть
1) ответы на команды группы get
2) сигналы прерывания trap (выключение питания, попытка несанкционированного доступа)

Команды SNMP могут посылаться либо администратором управляющей станции через интерфейс командной строки, либо (гораздо чаще), через соответствующий API из программ мониторинга сети, работающих на управляющей станции

Аутентификация в SNMP

1) Объектом аутентификации являются именованные сообщества (community) на управляющей станции, имеющей определенный IP-адрес
2) на каждом агенте явно конфигурируется, какие права доступа предоставляются команде, поступившей от «лица» (IP-адрес, имя- community), см. ниже как конфигурируются

Приведем формат простейших команд

getone IP-адрес-агента имя- community имя-переменной

putone IP-адрес-агента имя- community имя-переменной тип-значения значение-переменной

snmpstat
-r – информация о таблицах маршрутизации
-a – информация о таблицах ARP
-i – статистика по интерфейсам
-S – группа SNMP
-s – группа System
-t – информация о всех соединениях

Программы и конфигурационные файлы агентов и управляющих станций SNMP

На агенте
Программы: демон snmpd
конфигурационные файлы
/etc/snmpd.conf – задаются значения переменных группы System
etc/snmpd.comm – сообщества
IP-адрес-УС имя-community права-доступа (rw)
Файл
etc/snmpd.trap
IP-адрес-УС имя-community порт (162)

На управляющей станции
Программы: утилиты командной строки, ПО мониторинга сетью

3.3. Основы протокола DHCP

(логичнее было бы рассмотреть этот протокол при рассмотрении уровня IP)

DHCP – Dynamical Host Configuration Protocol

Протокол обеспечивает динамическое выделение IP-адреса компьютеру при его включении

Основан на использовании специальных серверов DHCP.

При загрузке компьютеров с DHCP-клиентом он посылает широковещательный запрос по протоколу UDP на подключение к серверу DHCP
(адрес источника – все 0, адрес получателя – все 1).
Серверы DHCP, получившие такой запрос сообщают о себе клиенту по его MAC-адресу.
Если клиент получил сообщения о нескольких серверах, то он по своим настройкам выбирает наиболее подходящий и обращается к нему.
Сервер DHCP присваивает адреса клиентам одним из нескольких возможных способов (способ задается при конфигурировании сервера DHCP.

3.4. Основы информационной безопасности компьютерных систем и сетей

Термин secure в зависимости ок контекста означает либо «безопасный» либо «защищенный»

Состав ИС

- Компьютер(ы) (аппаратура, программы, файлы данных)
- Внешние носители информации (распечатки, магнитные носители, CD, USB-флэшки, изображения на экране монитора . . .)
- Каналы связи (каналы передачи данных)

Определение безопасной информационной системы (ИС)

Защищенной (безопасной) называется такая компьютерная система, которая:
1) всегда делает то, что она должна делать
и
2) никогда не делает того, что она делать не должна

Свойства информации, как объекта защиты.

- Конфиденциальность
  (privacy – секретность)
- Целостность
  (integrity - сохранность, защищенность от несанкционированных изменений)
- Доступность
  (возможность оперативного доступа к информации)
- Изоляция
  (объектов защиты друг от друга)
- Предсказуемость
- Возможность аудита (напр. C2 аудит)

Угрозы информационной безопасности (ИБ)

Угроза ИБ – потенциальное событие или явление, в результате которого ИБ ИС может быть нарушена

Классификация угроз ИБ

- По виду нарушений безопасности
  нарушение различных свойств безопасной КС: конфиденциальности, целостности, доступности и пр.
- По причине нарушений:
  - аварии и стихийные бедствия
  - диверсии
  - ненадежный персонал
  - сбои и отказы оборудования
  - применение технических средств нарушения ИБ
  - непосредственный доступ к информации
  - программно-технические методы нарушения ИБ

"Непосредственные» методы нарушения ИБ

- Непосредственный доступ к носителям информации
- Внешний визуальный доступ к информации
- Внешний доступ к системе и информации через электромагнитные поля, излучаемые системой
- Непосредственный доступ к самой компьютерной системе и к работе в ней

Программно-технические методы нарушения ИБ

- Проникновение в систему через вредоносные программы (троянские программы, вирусы, интернет-черви)
- Взлом внутренней защиты программ (внутренние атаки) – **крэкерство**
- Взлом сетевой защиты (внешние атаки) - **хакерство**

Понятие о политике ИБ

- Оценка вероятности различных угроз ИБ
- Оценка финансового риска реализации различных угроз ИБ
- Оценка стоимости мер по предотвращению различных угроз безопасности
- Выработка политики ИБ как компромисса между
  - затратами на предотвращение различных угроз
  - размером ущерба, наносимого при реализации этих угроз

Классификация мер по обеспечению ИБ

- Правовые (юридические) (например, статьи УК РФ 272, 273, 274)
- Экономические
  (надбавки, страховки, возмещение убытков)
- Организационные
  (выбор месторасположения КС (и резервных площадок), подбор персонала, разработка и контроль за выполнением должностных инстр., сервисное обслуживание, взаимодействие с органами и пр.)

- Инженерно-технические
  (защищенная инфраструктура КС: электрозащита; защита от разрушений, пожаров и пр.; оптимальное размещение оборудования, защита от прослушки и визуального доступа и пр.)
- Технические
  (резервирование оборудования, данных и каналов связи, использование резервного электропитания, контроль отсутствия средств съема информации)
- Программно-технические

Меры по предотвращению непосредственных и программно-технических нарушений ИБ

Меры по предотвращению непосредственного неправомерного доступа к КС

- Ограничение доступа в помещения
- Авторизация доступа к КС (логины и пароли)
- Разграничение уровней полномочий доступа различных пользователей к различным компонентам и функциям КС
- Использование нескольких логинов для работы с разными уровнями полномочий пользователям, обладающим большими полномочиями в системе
- Система 2-х ключей от «*ядерного чемоданчика*»
- Недопустимость покидания рабочего места даже на короткий промежуток времени при открытом окне интерфейса с высоким уровнем полномочий

(!!) Правильные выбор и хранение паролей

(!!!) Пример копирования командного интерпретатора (sh, csh и др.) с установкой SetUID
(!!!) Опасность *SetUID* файлов

Методы предотвращения программно-технических угроз :

Программно-технические угрозы
- вредоносные программы,
- внутренние атаки,
- внешние (сетевые) атаки

Вредоносные программы и меры по предотвращению их проникновения

Программы, выполняющие несанкционированные действия на компьютере, по *методам проникновения* делятся на:

- Вирусы
- Троянские программы
- Сетевые (Интернет) черви

Определения вируса, трояна, червя: . . .

Разновидности вирусов
- по «среде обитания» (исполнимые программы, boot-сектор, postscript-программы, скриптовые вирусы, макровирусы, почтовые вирусы)
- по виду разрушающего воздействия

Возможные эффекты выполнения вредоносных программ
- Передача информации «взломщику» через сеть
- Разрушение файлов программ
- Разрушение файлов данных
- Нарушение работы или разрушение операционной системы или отдельных ее частей
- Форматирование дисков
- Физическое повреждение дисков
- Физическое повреждение экрана монитора
- Физическое повреждение процессора и других устройств

Минимальные меры по предотвращению проникновения вредоносных программ
- установка программ только из надежных источников
- использование и регулярное обновление антивирусного и «антишпионского» ПО

Внутренние атаки (крэкерство) и методы борьбы с ними

Внутренняя атака – это атака, произведенная зарегистрированным пользователем системы, повышенные полномочия достигаются путем «взлома» системы защиты

Основной способ взлома – это использование брешей в системе защиты

«Бреши» в защите программ – недокументированные (и не предусмотренные разработчиками программ) точки входа в эти программы, которые могут быть использованы для доступа к программам в момент исполнения их привилегированным пользователем

- Примеры:
  - использование функции (библиотечной программы) *gets()* , ввода строки символов без контроля переполнения буфера
  - использование setUID бита в правах доступа к исполнимым файлам, владельцем которых является высоко привилегированный пользователь (в среде UNIX), особенно для скриптов .

Основные меры по предотвращению и обнаружению внутренних атак

1) меры по предотвращению непосредственного использования недозволенных полномочий сотрудниками; применение средств аудита

2) Обеспечение отсутствия брешей в программах, которые могут работать от лица пользователей с высоким уровнем полномочий
- применение сканеров безопасности для обнаружения брешей.
- применение средств аудита
(в частности, встроенных в локальные системы IDS средства аудита системных журналов)

Внешние атаки (хакерство) и методы борьбы с ними

Основные программно-технические уязвимости в системе безопасности (угрозы), проявляющиеся на разных уровнях сетевых протоколов

1-2) На физическом и канальном уровнях
- Возможность прослушивания
    - прослушивание сегмента Ethernet через программы sniffer-ы
    - прослушивание по электромагнитным полям
- DoS атаки «глушения каналов» наведенными электромагнитными полями
(в электрических и радиоканалах)

3) на уровне IP
- IP-spoofing (подмена IP-адреса отправителя)
- DoS и DDoS атаки на маршрутизаторы (заваливание их чрезмерно большим количеством малых пакетов)
- прослушивание (путем зеркалирования портов маршрутизаторов)

4) На транспортном уровне
- перехват TCP-соединений
    (пример с Реестром запрещенных сайтов)
- DoS атаки портов прикладных служб (TCP Sync)

5) На прикладном уровне
- компрометация DNS (методами IP-spoofing
- «взломы» почтовых серверов
- хакерские proxy- серверы
- перехват паролей
- взломы через CGI-скрипты
- взломы через бреши в прикладных программах демонов и клиентов

- и др.

Разновидности сетевых атак

1) по цели

- на взлом системы (проникновение в систему)
- на отказ в обслуживании

2) по способу реализации

- одиночная (возможно, выполняемая через цепочку посредников)
- распределенная (обычно – для атак на отказ в обслуживании), возможно также выполняемая через цепочки посредников

Стадии развития внешних атак

- ***Разведывательная***
  - сканирование портов
  - считывание информации из памяти компьютера и внешних устройств
  - считывание с внешних носителей
  - «прослушивание» каналов сети
  - «добыча» паролей организационными методами (подкуп)
  - . . .
- ***Основная (непосредственно атакующее действие)***
  - «внедрение» и исполнение программы, реализующей функционал атаки
- ***Завершающая (Маскирующая)***
  - «заметание следов» выполненной атаки (обычно – завершающее действие вредоносной программы состоит в модернизации системных журналов)

Комплексные меры борьбы с атаками

своевременная установка патчей
- межсетевые экраны
- использование сканеров безопасности
- IDS
- применение средств аудита

(в частности, встроенных в локальные системы IDS средства аудита системных журналов)

Сканеры безопасности

- Устанавливаются на одном из компьютеров локальной сети
- Разновидности: локальный и сетевой(host based и network based)
- Может использоваться администратором этой сети для проверки уязвимости только администрируемых им компьютеров

- В своей работе среди прочих методов использует метод имитации различных сетевых атак
- Применение сканера безопасности к произвольным компьютерам сети может быть квалифицировано, как уголовно наказуемое «использование вредоносных программ для ЭВМ» (ст.273 УК РФ),

О логике работы сетевого сканера безопасности: сканирование портов с целью обнаружения уязвимостей сервисов (присущих определенным версиям их демонов) с последующей имитацией атак.
Локальный сканер безопасности может дополнительно анализировать код программ на предмет наличия в нем соответствующих уязвимостям фрагментов кода.

Системы
- XSpider – лицензирован ФСТЭК

Системы обнаружения вторжений IDS

Анализируют сетевой весь трафик в одном или нескольких (непосредственно подключенных к оборудованию, на котором установлена IDS) сегменте сети

•Сигнатурные методы обнаружения атак
-последовательность пакетов каждого сетевого соединения сопоставляется со специальным шаблоном – сигнатурой»
-база данных сигнатур и необходимость ее регулярного обновления
-достоинство: надежно и эффективно обнаруживает атаки известного типа
-недостаток – метод бессилен перед новыми типами атак

•Статистические методы: достоинства и недостатки

•Виды реакции на обнаруженную атаку

Разновидности IDS

- Встроенные средства сетевых ОС серверов
- Встроенные средства ОС маршрутизаторов
- Специальные системы с открытым кодом (например, SNORT)
- Специальные коммерческие системы (например, Real Secure)

Межсетевые экраны

•Метафора «противопожарной перегородки»

•Специальное программное или аппаратно-программное «изделие», устанавливаемое «между» КС (индивидуальным компьютером, компьютерной сетью) и внешней сетью

•Может выполнять **фильтрацию** проходящих через него в различных направлениях («из КС» и в «КС») **пакетов** на основе специальных правил, различных для разных направлений пересылки пакетов

•Фильтрация может выполняться на различных уровнях сетевых протоколов от IP до прикладного по различным параметрам, отличающимся для различных уровней

•МЭ сужает до минимально необходимого возможные направления и протоколы взаимодействия компонентов КС с внешними сетями, сужая тем самым возможные пути распространения внешних атак (изолируя от заведомо «ненужной» части сетей)

Разновидности МЭ

•Специальное ПО на клиентском компьютере

•Штатное ПО на маршрутизаторе

•Специальные программные системы

•Специальные программно-аппаратные системы – черные ящики firewall

•*Разновидности ПО, частично реализующие функции МЭ*
-NAT
-Proxy
-Wrapper