Протоколы .

Понятие протокола.

- Современные сети построены по многоуровневому принципу. Чтобы организовать связь компьютеров, требуется сначала создать свод правил их взаимодействия, определить язык общения, т.е. определить, что означают посылаемые сигналы и т.д. Эти правила и определения называются ***протоколом***.
- Для работы сетей необходимо множество различных *протоколов*: например, управляющих физической связью, установлением связи по сети, доступом к *ресурсам* и т.д.
- Общепринятая многоуровневая структура, известная как "эталонная модель *ISO OSI*" и спроектирована с целью упростить и упорядочить это великое множество *протоколов* и отношений

Инкапсуляция

Инкапсуляция - способ упаковки данных в формате одного протокола в формат другого протокола. Например, упаковка IP-пакета в кадр Ethernet или TCP-сегмента в IP-пакет.

На каждом из уровней данные обычно инкапсулируются при помощи простого механизма: пакет состоит из заголовка и данных. Заголовок содержит метаинформацию: источник, пункт назначения и другие атрибуты, а данные представляют собой ту информацию, которая подлежит передаче. Пакет верхнего уровня инкапсулируется в данных пакета уровнем ниже. При передаче пакета обратно (от нижнего уровня к верхнему) данные восстанавливаются в том виде, в каком они должны быть представлены на данном уровне.

Наиболее часто в схеме TCP/IP используется сочетание трех протоколов: IP, TCP, UDP.

- Заголовок IP начинается с номера версии (*version number*).
- *IHL* - Поле длины заголовка.
- Поле типа услуги (*type-of-service*) -с помощью этого поля дейтаграммам могут быть назначены различные уровни значимости.
- Поле общая длина (*total length*) определяет длину всего пакета IP в байтах, включая данные и заголовок.
- Поле идентификации (*identification*) содержит целое число, обозначающее текущую дейтаграмму, используется для соединения фрагментов дейтаграммы.
- Поле флагов (*flags*) определяет, может ли быть фрагментирована данная дейтаграмма и является ли текущий фрагмент последним.
- Поле срок жизни (*time-to-live*) поддерживает счетчик, значение которого постепенно уменьшается до нуля; в этот момент дейтаграмма отвергается.
- Поле протокола (*protocol*) указывает, какой протокол высшего уровня примет входящие пакеты после завершения обработки IP.
- Поле контрольной суммы заголовка (*header checksum*) помогает обеспечивать целостность заголовка IP.
- Поля адресов источника и пункта назначения (*source and destination address*) обозначают отправляющий и принимающий узлы.
- Поле опции (*options*) позволяет IP обеспечивать факультативные возможности, такие, как защита данных.
- Поле данных (*data*) содержит информацию высших уровней.

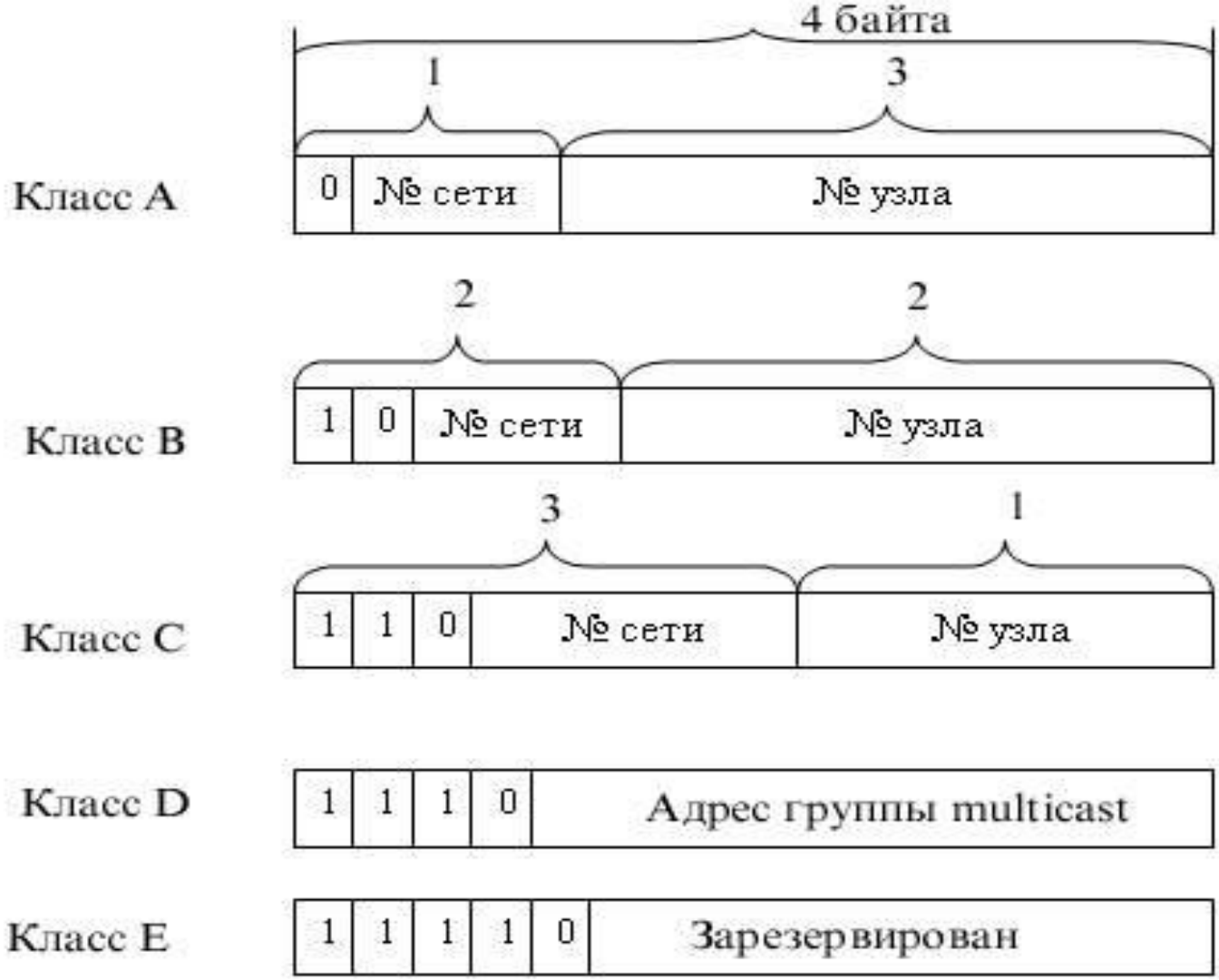
Адресация

- IP-адрес имеет длину 4 байта и обычно записывается в виде четырех чисел, представляющих значения каждого байта в десятичной форме и разделенных точками, например, 128.10.2.30 — традиционная десятичная форма представления адреса, а 10000000 00001010 00000010 00011110 - двоичная форма представления этого же адреса.
- Адрес состоит из двух логических частей — номера сети и номера узла в сети. Какая часть адреса относится к номеру сети, а какая — к номеру узла, определяется значениями первых бит адреса. Значения этих бит являются также признаками того, к какому *классу* относится тот или иной IP-адрес. Адреса подсети присутствуют только в том случае, если администратор сети принял решение о разделении сети на подсети. Длина полей адреса сети, подсети и главной вычислительной машины являются переменными величинами.

Адресация IP обеспечивает пять различных классов сети:

- *Class A* Сети класса A предназначены главным образом для использования с несколькими очень крупными сетями, т.к. они обеспечивают всего 7 битов для поля адреса сети.
- *Class B* Сети класса B выделяют 14 битов для поля адреса сети и 16 битов для поля адреса главной вычислительной машины. Этот класс адреса обеспечивает хороший компромисс между адресным пространством сети и главной вычислительной машины.
- *Class C* Сети класса C выделяют 22 бита для поля адреса сети. Однако сети класса C обеспечивают только 8 битов для поля адреса главной вычислительной машины, поэтому число главных вычислительных машин, приходящихся на сеть, может стать ограничивающим фактором.
- *Class D* Адреса класса D резервируются для групп с многопунктовой адресацией (в соответствии с официальным документом RFC 1112). В адресах класса D четыре бита наивысшего порядка устанавливаются на значения 1,1,1 и 0.
- *Class E* Адреса класса E также определены IP, но зарезервированы для использования в будущем. В адресах класса E все четыре бита наивысшего порядка устанавливаются на 1.

Адреса IP записываются в формате десятичного числа с проставленными точками, например, 34.0.0.1.



- Если адрес начинается с 0, то сеть относят к *классу А* и номер сети занимает один байт, остальные 3 байта интерпретируются как номер узла в сети. Сети класса А имеют номера в диапазоне от 1 до 126. (Номер 0 не используется, а номер 127 зарезервирован для специальных целей, о чем будет сказано ниже.) Сетей класса А немного, зато количество узлов в них может достигать 224, то есть 16 777 216 узлов.
- Если первые два бита адреса равны 10, то сеть относится к *классу В.* В сетях класса В под номер сети и под номер узла отводится по 16 бит, то есть по 2 байта. Таким образом, сеть класса В является сетью средних размеров с максимальным числом узлов 216, что составляет 65 536 узлов.
- Если адрес начинается с последовательности ПО, то это сеть *класса С.* В этом случае под номер сети отводится 24 бита, а под номер узла — 8 бит. Сети этого класса наиболее распространены, число узлов в них ограничено 28, то есть 256 узлами.
- Если адрес начинается с последовательности 1110, то он является адресом *класса D* и обозначает особый, групповой адрес — multicast. Если в пакете в качестве адреса назначения указан адрес класса D, то такой пакет должны получить все узлы, которым присвоен данный адрес.
- Если адрес начинается с последовательности 11110, то это значит, что данный адрес относится к *классу Е.* Адреса этого класса зарезервированы для будущих применений.

Диапазоны номеров сетей и максимальное число узлов, соответствующих каждому классу сетей.

Большие сети получают адреса класса A, средние — класса B, а маленькие — класса C.

Класс	Первые биты	Наименьший номер сети	Наибольший номер сети	Максимальное число узлов в сети
A	0	1.0.0.0	126.0.0.0	(2)^24
B	10	128.0.0.0	191.255.0.0	(2)^16
C	110	192.0.1.0	223.255.255.0	(2)^8
D	1110	224.0.0.0	239.255.255.255	Multicast
E	11110	240.0.0.0	247.255.255.255	Зарезервирован

Использование масок в IP-адресации

- Традиционная схема деления IP-адреса на номер сети и номер узла основана на понятии класса, который определяется значениями нескольких первых бит адреса. Именно потому, что первый байт адреса 185.23.44.206 попадает в диапазон 128-191, мы можем сказать, что этот адрес относится к классу B, а значит, номером сети являются первые два байта, дополненные двумя нулевыми байтами — 185.23.0.0, а номером узла — 0.0.44.206.

- Другой признак, с помощью которого можно было бы более гибко устанавливать границу между номером сети и номером узла.

  Маска — это число, которое используется в паре с IP-адресом; двоичная запись маски содержит единицы в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети. Поскольку номер сети является цельной частью адреса, единицы в маске также должны представлять непрерывную последовательность. Для стандартных классов сетей маски имеют следующие значения:

- **класс A - 11111111. 00000000. 00000000. 00000000 (255.0.0.0);**

- **класс B - 11111111. 11111111. 00000000. 00000000 (255.255.0.0);**

- **класс C-11111111.11111111.11111111.00000000(255.255.255.0).**

Формат пакета TCP

Source port			Destination port
Sequence number			
Acknowledgment number			
Data offset	Reserved	Flags	Window
Checksum			Urgent pointer
Options (+ padding)			
Data (variable)			

- «Порт источника» (*source port*) обозначает точку, в которой конкретный процесс высшего уровня источника принимает услуги TCP;
- «Порт пункта назначения» (*destination port*) обозначает порт процесса высшего уровня пункта назначения для услуг TCP.
- «Номер последовательности» (*sequence number*) обозначает номер, присвоенный первому байту данных в текущем сообщении. В некоторых случаях оно может также использоваться для обозначения номера исходной последовательности, который должен использоваться в предстоящей передаче.
- «Номер подтверждения» (*acknowledgement number*) содержит номер последовательности следующего байта данных, которую отправитель пакета ожидает для приема.
- «Сдвиг данных» (*data offset*) - это число 32-битовых слов в заголовке TCP.
- «Резерв» (*reserved*) зарезервировано для использования разработчиками протокола в будущем.
- «Флаги» (*flags*) содержит различную управляющую информацию.
- «Окно» (*window*) обозначает размер окна приема отправителя (буферный объем, доступный для поступающих данных).
- «Контрольная сумма» (*checksum*) указывает, был ли заголовок поврежден при транзите.
- «Указатель срочности» (*urgent pointer*) указывает на первый байт срочных данных в пакете.
- «Опции» (*options*) - различные факультативные возможности TCP.

Transmission Control Protocol(TCP)- протокол управления передачей.

TCP делит поток байт на части — сегменты и передает их ниже лежащему уровню межсетевого взаимодействия. Для этого он пользуется системой кодов, исправляющих ошибки. Простейшим примером такового служит код с добавлением к каждому *пакету* контрольной суммы (и к каждому байту бита проверки на четность). При помещении в TCP-конверт вычисляется контрольная сумма, которая записывается в TCP-заголовок. После того как эти сегменты будут доставлены средствами уровня межсетевого взаимодействия в пункт назначения, протокол TCP снова соберет их в непрерывный поток байт. Если при приеме заново вычисленная сумма не совпадает с той, что указана на конверте, значит что-то тут не то и где-то в пути произошла ошибка, так что надо переслать этот *пакет* заново, что и делается.

Протокол TCP обеспечивает гарантированную доставку с установлением логического соединения в виде байтовых потоков. Он освобождает прикладные процессы от необходимости использовать ожидания и повторные передачи для обеспечения надежности. Наиболее типичными прикладными процессами, использующими TCP, являются *www, ftp* и *telnet*.

User Datagram Protocol (UDP) *– протокол дейтаграмм пользователя.*

Дейтаграмма - это *пакет*, передаваемый через сеть независимо от других пакетов без установления логического соединения и подтверждения приема.

Протокол UDP намного проще, чем TCP; он полезен в ситуациях, когда мощные механизмы обеспечения надежности протокола TCP не обязательны.

UDP выполняет только функции связующего звена (мультиплексора) между сетевым протоколом и многочисленными службами прикладного уровня или пользовательскими процессами.
Заголовок UDP имеет всего четыре поля:

- поле порта источника (*source port*),
- поле порта пункта назначения (*destination port*),
- поле длины (*length*)
- поле контрольной суммы UDP (*checksum UDP*).

Поля порта источника и порта назначения выполняют те же функции, что и в заголовке TCP. Поле длины обозначает длину заголовка UDP и данных; поле контрольной суммы обеспечивает проверку целостности пакета. Контрольная сумма UDP является факультативной возможностью.

UDP используется для *клиентов*, которые посылают только короткие сообщения и могут просто заново послать сообщение, если отклик подтверждения не придет достаточно быстро.

В отличие от *TCP*, данные, отправляемые прикладным процессом через модуль *UDP*, достигают места назначения как единое целое. Например, если процесс-отправитель производит 3 записи в *UDP-порт*, то процесс-получатель должен будет сделать 3 чтения. Размер каждого записанного сообщения будет совпадать с размером соответствующего прочитанного. *Протокол* UDP сохраняет границы сообщений, определяемые прикладным процессом. Он никогда не объединяет несколько сообщений в одно целое и не делит одно сообщение на части.

Протокол TCP

- *Надежный* – если сегмент TCP потерян или поврежден, реализация TCP обнаружит это и повторно передаст необходимый сегмент.
- *На основе логического соединения* – перед началом передачи данных TCP устанавливает с удаленной машиной соединении, обмениваясь с ней служебной информацией. Этот процесс обычно называется квинтированием. В конце соединения аналогичное квинтирование разрывает связь.
- *С непрерывным потоком данных* – TCP обеспечивает механизм передачи, позволяющий пересылать произвольное количество байт.

Протокол UDP

- *Ненадежный* – UDP не имеет ни встроенного механизма обнаружения ошибок, ни средств повторной пересылки поврежденных или потерянных данных
- *Без установления логического соединения* – перед пересылкой данных UDP не устанавливает логического соединения. Информация пересылается в предположении, что принимающая сторона ее ожидает.
- *Основанный на сообщениях* – UDP позволяет приложениям пересылать информацию в виде сообщений, передаваемых посредством дейтаграмм, которые являются единицами передачи данных в UDP. Приложение должно самостоятельно распределить данные по отдельным дейтаграммам

Internet Control Message Protocol (ICMP) – протокол управляющих сообщений

В протоколе существует определенное количество сообщений управляющих сетью.

- Сообщает источнику об отказах маршрутизации,
- обеспечивает метод проверки способности узлов образовывать повторное эхо в объединенной сети,
- метод стимулирования более эффективной маршрутизации,
- метод информирования источника о том, что какая-то дейтаграмма превысила назначенное ей время существования в пределах данной объединенной сети и другие полезные сообщения.

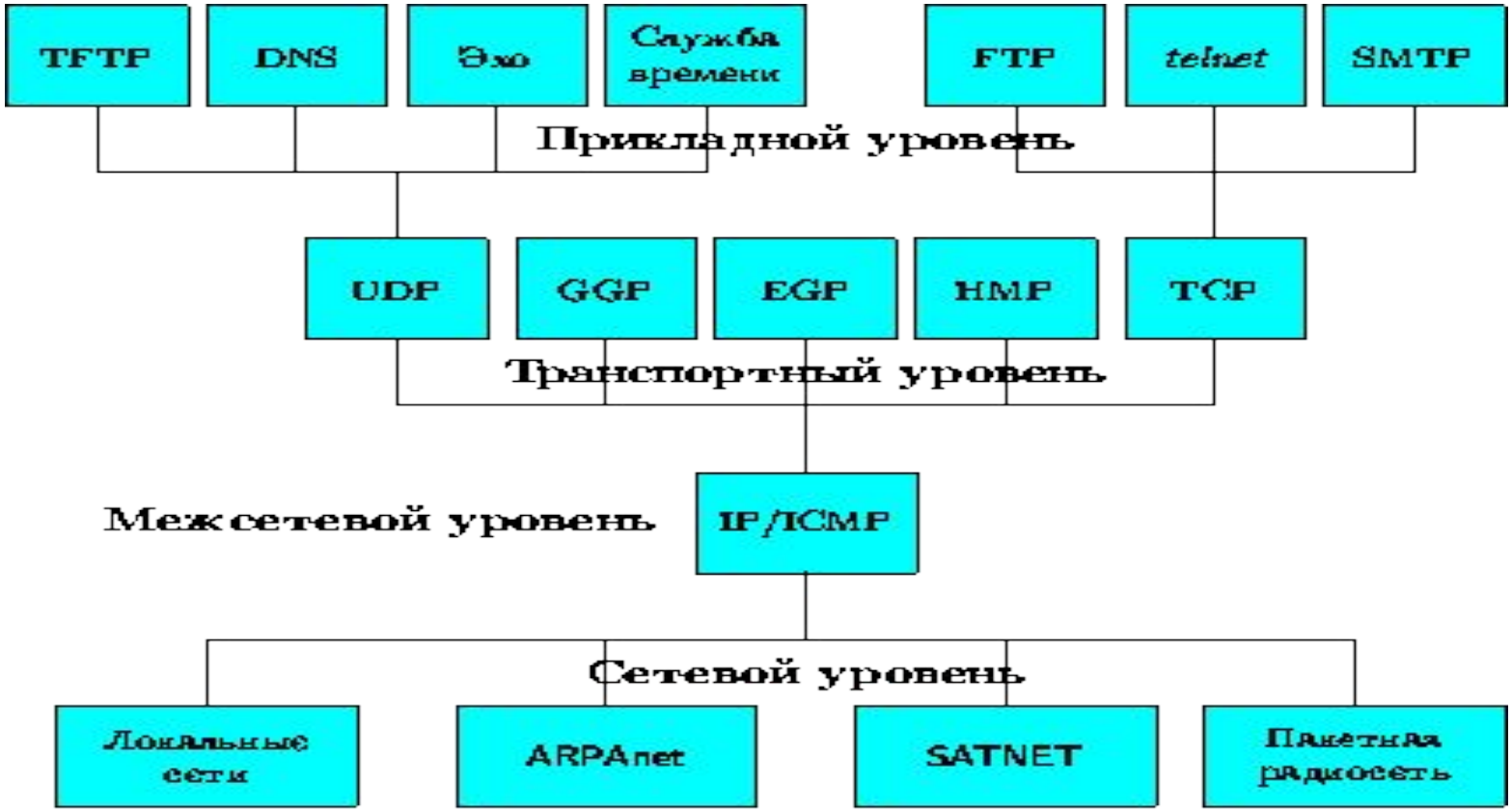
Сделанное недавно дополнение к ICMP обеспечивает для новых узлов возможность нахождения маски подсети, используемой в межсети в данный момент.

В целом, ICMP является интегральной частью любых реализаций IP.

Transmission Control Protocol - Протокол управления передачей

Термин "TCP/IP" обычно означает все, что связано с протоколами TCP и IP. Он охватывает целое семейство протоколов, прикладные программы и даже саму сеть. В состав семейства входят протоколы TCP, UDP, ICMP, telnet, FTP и многие другие.

Иерархия протоколов семейства TCP/IP



Другие протоколы(прикладного уровня)

- ***Telnet*** позволяет обслуживающей машине рассматривать все удаленные терминалы как стандартные "сетевые виртуальные терминалы" строчного типа, работающие в коде ASCII, а также обеспечивает возможность согласования более сложных функций TELNET работает на базе протокола TCP. На прикладном уровне над TELNET находится либо программа поддержки реального терминала (на стороне пользователя), либо прикладной процесс в обсуживающей машине, к которому осуществляется доступ с терминала.
- ***FTP* (File Transfer Protocol** - протокол передачи файлов) распространен также широко как TELNET. Он является одним из старейших протоколов семейства TCP/IP. Также как TELNET он пользуется транспортными услугами TCP. Существует множество реализаций для различных операционных систем, которые хорошо взаимодействуют между собой. Пользователь FTP может вызывать несколько команд, которые позволяют ему посмотреть каталог удаленной машины, перейти из одного каталога в другой, а также скопировать один или несколько файлов.
- ***Протокол SMTP (Simple Mail Transfer Protocol*** - простой протокол передачи почты) поддерживает передачу сообщений (электронной почты) между произвольными узлами сети internet. Имея механизмы промежуточного хранения почты и механизмы повышения надежности доставки, протокол SMTP допускает использование различных транспотных служб. Он может работать даже в сетях, не использующих протоколы семейства TCP/IP.

Другие протоколы(прикладного уровня)

- Протокол управления простой сетью (***Simle network management protocol - SNMP***) является протоколом управления сетью, используемым для сообщения об аномальных условиях в сети и установления значений допустимых порогов в сети.
- ***X Windows*** является популярным протоколом, который позволяет терминалу с интеллектом связываться с отдаленными компьютерами таким образом, как если бы они были непосредственно подключенными мониторами.
- Комбинация протоколов ***Network File System (NFS)*** (sys. сетевых файлов), ***External Data Representation (EDP)*** (представление внешней информации) и ***Remote Procedure Call (RPC)*** (вызов процедуры обращений к отдаленной сети) обеспечивает прозрачный доступ к ресурсам отдаленной сети.

Другие протоколы

- ***ARP (Address Resolution Protocol,*** **протокол определения адресов)**: конвертирует 32-разрядные IP-адреса в физические адреса вычислительной сети, например, в 48-разрядные адреса Ethernet.
- ***IGMP (Internet Group Management Protocol*****, протокол управления группами Internet):** позволяет IP-дейтаграммам распространяться в циркулярном режиме (multicast) среди компьютеров, которые принадлежат к соответствующим группам.
- ***RARP (Reverse Address Resolution Protocol*****, протокол обратного преобразования адресов):** преобразует физические сетевые адреса в IP-адреса.