KUNGL TEKNISKA HÖGSKOLAN
Institutionen för teleinformatik
CCSlab

2G1305 Internetworking/Internetteknik
Winter 2002, Period 3

Module 3: UDP and friends + Multicasting

Lecture notes of G. Q. Maguire Jr.

Last modified: 2002.01.30:13:12

Lecture 3: Outline

- UDP
- BOOTP
- DHCP
- DNS, DDNS
- Multicast, IGMP, RSVP

Transport layer protocols

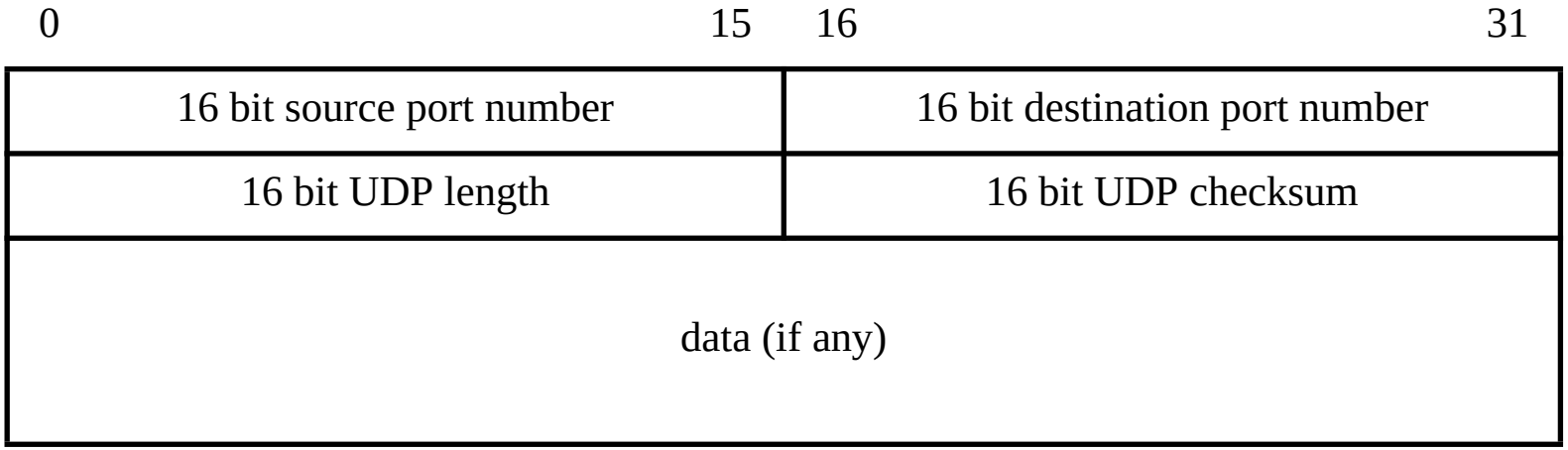
- Connectionless unreliable service (UDP) <<< today's topic
- Reliable stream service (TCP)

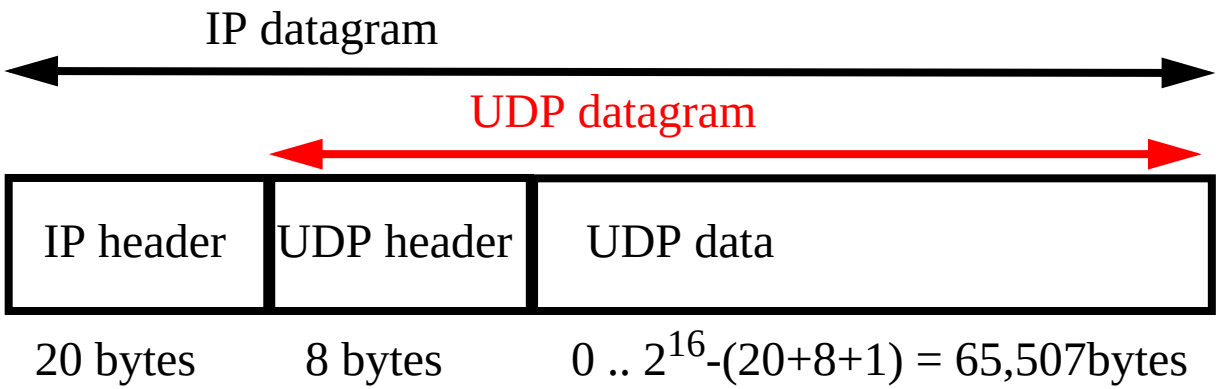
User Datagram Protocol (UDP)

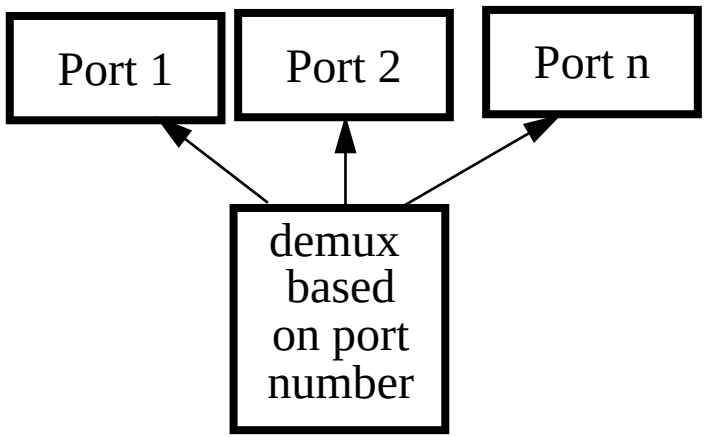
- Datagram-oriented transport layer protocol.
- Provides connectionless unreliable service.
- No reliability guarantee.
- Checksum covers both header and data, end-to-end, but optional.
  - if you care about your data you should be doing end-to-end checksums or using an even stronger error detection (e.g., MD5).
- An UDP datagram is silently discarded if checksum errors. No error message is generated.
- Lots of UDP traffic is only sent locally
  - - thus the reliability is comparable to the error rate on the local links. (see Stevens, Vol. 1, figure 11.5, pg. 147 for comparison of Ethernet, IP, UDP, and TCP checksum errors).
- Each output operation results in one UDP datagram, which causes one IP datagram to be sent.
- Applications which use UDP: DNS, TFTP, BOOTP, DHCP, SNMP, NFS, VoIP, etc.
  - An advantage of UDP is that it is a base to build your own protocols on.
  - Especially if you don’t need reliability and in order delivery of lots of data

UDP Header

8 byte header + possible data

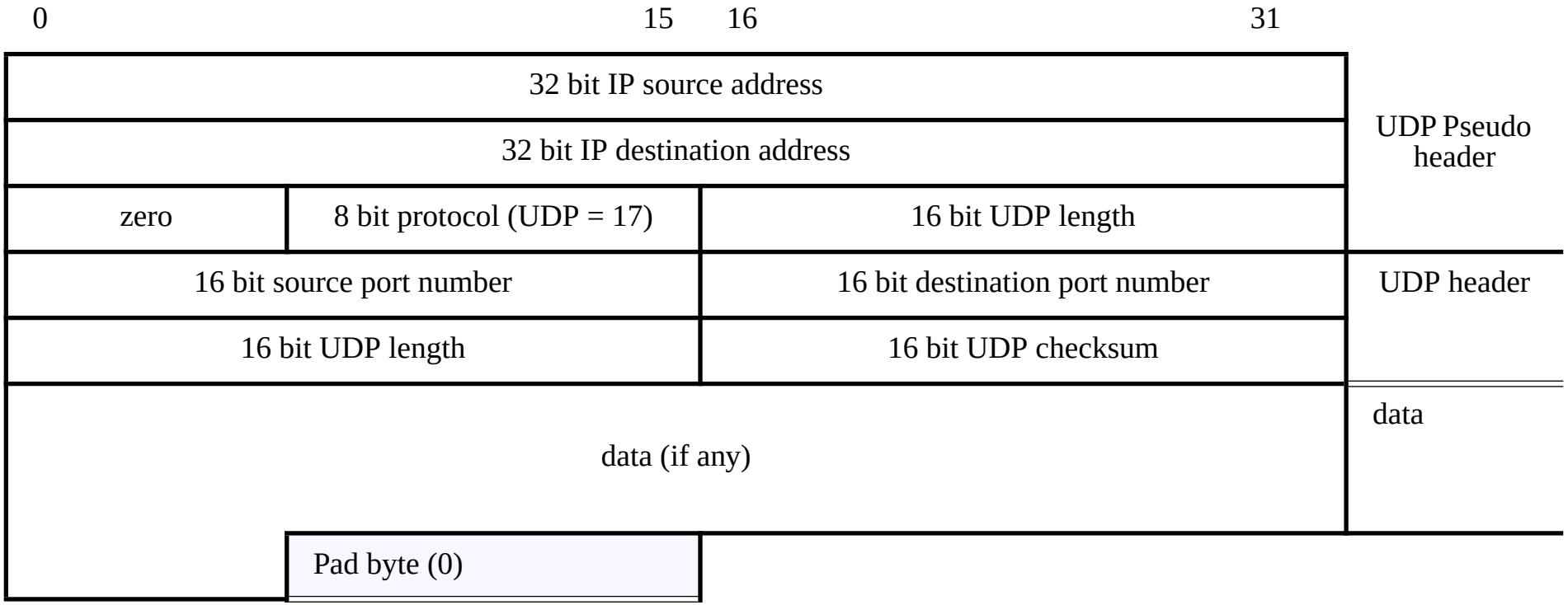






UDP Checksum and Pseudo-Header

- UDP checksum covers more info than is present in the UDP datagram alone: pseudo-header and pad byte (0) {to even number of 16 bit words}.
- Propose: to verify the UDP datagram reached its correct destination: right port number at the right IP address.
- Pseudo-header and pad byte are not transmitted with the UDP datagram, only used for checksum computation.



Reserved and Available UDP Port Numbers

	keyword	UNIX keyword	Description
0			reserved
7	ECHO	echo	Echo
9	DISCARD	discard	Discard == sink null
11	USERS	systat	Active users
13	DAYTIME	daytime	Daytime
15	-	netstat	Network status program
17	QUOTE	qotd	Quote of the day
19	CHARGEN	chargen	Character generator
37	TIME	time	Time server
39	RLP	rlp	Resource Location Protocol
42	NAMESERVER	name	Host Name Server
43	NICNAME	whois	Who is
53	DOMAIN	domain	Domain Name Server
67	BOOTPS	bootps	Bootstrap Protocol Server
68	BOOTPC	bootpc	Bootstrap Protocol Client
69	TFTP	tftp	Trivial File Transfer Protocol
88	KERBEROS	kerberos5	Kerberos v5 kdc
111	SUNRPC	sunrpc	SUN Remote Procedure Call (portmap)
123	NTP	ntp	Network Time Protocol
137	netbios_ns	netbios_ns	NetBIOS name service
138	netbios_dgm	netbios_dgm	NetBIOS Datagram Service
139	netbios_ssn	netbios_ssn	NetBIOS Session Service
161		snmp	Simple Network Management Protocol Agent
162		snmp-trap	Simple Network Management Protocol Traps
512		biff	mail notification
513		who	remote who and uptime
514		syslog	remote system logging
517		talk	conversation
518		ntalk	new talk, conversation
520		route	routing information protocol
525		timed	remote clock synchronization
533	netwall	netwall	Emergency broadcasting
750	kerberos	kerberos	Kerberos (server)
6000 + display number			X11 server
7000			X11 font server

Port numbers in three groups

Range	Purpose
0 .. 1023	System (Well-Known) Ports
1024 .. 49151	User (Registered) Ports
49152 .. 65535	Dynamic and/or Private Ports

‘For the purpose of providing services to unknown callers, a service contact port is defined. This list specifies the port used by the server process as its contact port. The contact port is sometimes called the "well-known port".’

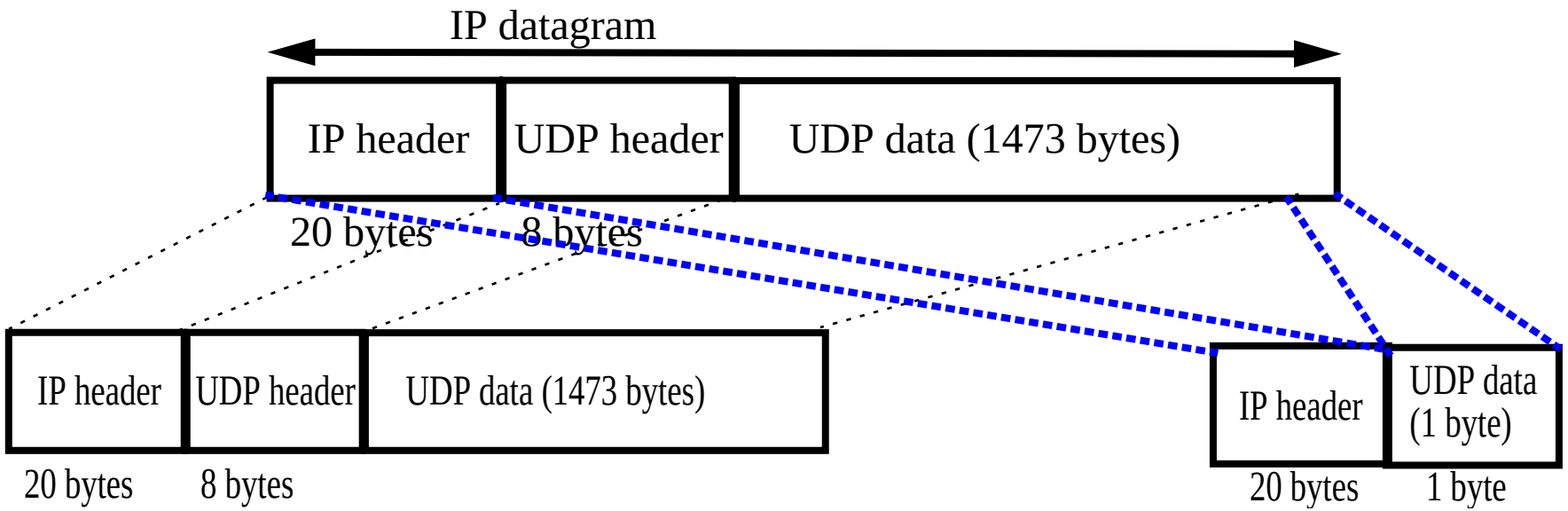
http://www.iana.org/assignments/port-numbers

MTU and Datagram Fragmentation

If datagram size > MTU, perform fragmentation.

- At sending host or at intermediate router (IPv4).
- Reassembled only at final destination.

Example: 1501 (20 + 8 + 1473 data) on Ethernet (MTU=1500):



- Note there is no UDP header in the second fragment.
- Therefore, a frequent operation is to compute the path MTU before sending anything else. (see RFC 1191 for the table of common MTUs)

Fragmentation Required

If datagram size > MTU, DF (Don’t Fragment) in IP header is on, then the router sends ICMP Unreachable Error.

Of course this can be used to find Path MTU.

Interaction between UDP and ARP

With ARP cache empty, send a UDP datagram with 8192 bytes onto an Ethernet

- 8192 bytes > ethernet MTU, therefore 6 fragments are created by IP
- if ARP cache is empty, first fragment causes ARP request to be sent
- This leads to two timing questions:

1. Are the remaining fragments sent before the ARP reply is received?
2. What does ARP do with multiple packets to the same destination while waiting for a reply?

Example under BSD

```
Bsdi% arp -a      ARP cache is empty
Bsdi% sock -u -i -n1 -w8192 svr4 discard
```

10.0		arp who-has svr4 tell bsdi
20.001234	(0.0012)	arp who-has svr4 tell bsdi
30.001941	(0.0007)	arp who-has svr4 tell bsdi
40.002775	(0.0008)	arp who-has svr4 tell bsdi
50.003495	(0.0007)	arp who-has svr4 tell bsdi
60.004319	(0.0008)	arp who-has svr4 tell bsdi
70.008772	(0.0045)	arp reply svr4 is-at 0:0:c0:c2:9b:26
80.009911	(0.0011)	arp reply svr4 is-at 0:0:c0:c2:9b:26
90.011127	(0.0012)	bsdi > svr4: (frag 10863:800@7400)
100.011255	(0.0001)	arp reply svr4 is-at 0:0:c0:c2:9b:26
110.012562	(0.0013)	arp reply svr4 is-at 0:0:c0:c2:9b:26
120.013458	(0.0009)	arp reply svr4 is-at 0:0:c0:c2:9b:26
130.014526	(0.0011)	arp reply svr4 is-at 0:0:c0:c2:9b:26
140.015583	(0.0011)	arp reply svr4 is-at 0:0:c0:c2:9b:26

- on a BSDI system:
  - each of the additional (5) fragments caused an ARP request to be generated
    - this **violates** the Host Requirements RFC - which tries to prevent ARP flooding by limiting the maximum rate to 1 per second
  - when the ARP reply is received the **last** fragment is sent
    - Host Requirements RFC says that ARP should save at least one packet and this should be the latest packet
  - unexplained anomaly: the System Vr4 system sent 7 ARP replies back!
  - no ICMP “time exceeded during reassembly” message is sent
    - BSD derived systems - never generate this error!
      It does set the timer internally and discard the fragments, but never sends an ICMP error.
    - fragment 0 (which contains the UDP header) was not received - so there is no way to know which process sent the fragment; thus unless fragment 0 is received - you are not required to send an ICMP “time exceeded during reassembly” error.

Is this just a fluke? (i.e., a rare event)

Not just a fluke

- The same error occurs even if you don’t have fragmentation - simply sending multiple UDP datagrams rapidly when there is no ARP entry is sufficient!
- NFS sends UDP datagrams whose length just exceeds 8192 bytes
  - NFS will timeout and resend
  - however, there will always be this behavior - if the ARP cache has no entry for this destination!

Maximum UDP Datagram size

- theoretical limit: 65,535 bytes - due to (IP's) 16-bit total length field
  - with 20 bytes of IP header + 8 bytes of UDP header ⇒ 65,507 bytes of user data
- two limits:
  - sockets API limites size of send and receive buffer; generally 8 kbytes, but you can call a routine to change this
  - TCP/IP implementation - Stevens found various limits to the sizes - even with loopback interface (see Stevens, Vol. 1, pg. 159)
- Hosts are required to handle at least 576 byte IP datagrams, thus lots of protocols limit themselves to 512 bytes or less of data:
  - DNS, TFTP, BOOTP, and SNMP

Datagram truncation

What if the application is not prepared to read the datagram of the size sent?

Implementation dependent:

- traditional Berkeley: silently truncate
- 4.3BSD and Reno: **can** notify the application that the data was truncated
- SVR4: excess data returned in subsequent reads - application is not told that this all comes from one datagram
- TLI: sets a flag that more data is available, subsequent reads return the rest of the datagram

UDP server design

Stevens, Vol, 1, pp. 162-167 discusses how to program a UDP server

You can often determine what IP address the request was sent to (i.e., the destination address):

- for example: thus ignoring datagrams sent to a broadcast address

You can limit a server to a given incoming IP address:

- thus limiting requests to a given interface

You can limit a server to a given foreign IP address and port:

- only accepting requests from a given foreign IP address and port #

Multiple recipients per port (for implementations with multicasting support)

- setting SO_REUSEADDR socket option ⇒ each process gets a copy of the incoming datagram

Note: limited size input queue to each UDP port, can result in silent discards without an ICMP message being sent back (since OS discarded, not the network!)

ICMP Source Quench Error

ICMP source quench **may** be generated if the system receives data faster than it can process it.

Note: "**may** be generated" - it is not required to generate this error

Stevens (Vol. 1, pp. 160-161) gives the example of sending 100 1024-byte datagrams from a machine on an ethernet via a router and SLIP line to another machine:

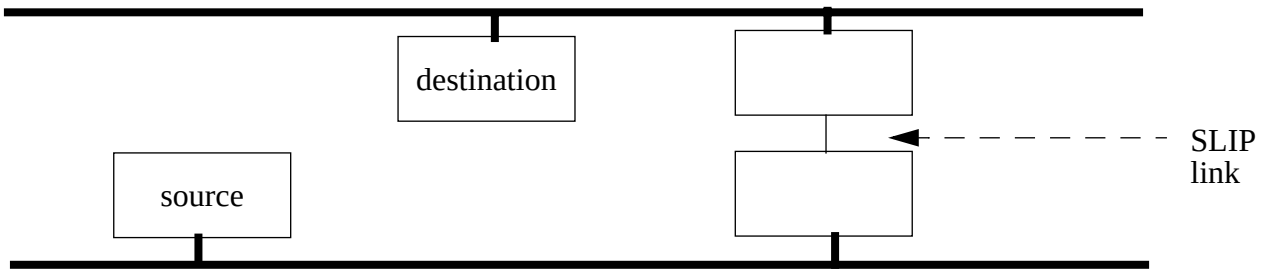


Figure 45: simplified from Stevens, Vol. 1, inside cover

- SLIP link is ~1000 times slower than the ethernet
- 26 datagrams are transmitted, then a source quench is sent for each successive datagram

- the router gets all 100 packets, before the first has been sent across the link!
  - the new Router Requirements RFC - says that routers should not generate source quench errors, since it just consumes network bandwidth and it is an ineffective and unfair fix for congestion
- In any case, the sending program never responded to the source quench errors!
  - BSD implementations ignore received source quenchs if the protocol is UDP
  - the program finished before the source quench was received!

Thus if you want reliability you have to build it in and do end-to-end flow control, error checking, and use (and thus wait for) acknowledgements.

BOOTP: Bootstrap Protocol (RFC 951)

Although you can figure out who you are, i.e., your IP address, via RARP - many machines want more information.

BOOTP requests and answer are sent via UDP (port 67 server; port 68 client)

- so it is easy to make a user space server
- the client (who wants the answer) need not have a full TCP/IP, it can simply send what **looks** like a UDP datagram with a BOOTP request[1].

Opcode (1=request, 2=reply)	hardware type (1=ethernet)	hardware address length (6 for ethernet)	hop count
transaction ID			
number of seconds		unused	
client IP address			
your IP address			
server IP address			
gateway IP address			
client hardware address (16 bytes of space)			
server hostname (64 bytes)			
Boot file name (128 bytes)			
Vendor specific information (64 bytes)			

1. see Stevens, Vol. 1, figure 16.2, pg. 216

BOOTP continued

When a request is sent as an IP datagram:

- if client does not know its IP address it uses 0.0.0.0
- if it does not know the server's address it uses 255.255.255.255
- if the client does not get a reply, it tries again in about 2 sec.

Vendor specific information (RFC 1497 and RFC1533)

- if this area is used the first 4 bytes are: IP address 99.130.83.99 this is called the "**magic cookie**"
- the rest of the area is a list of items, possibly including:
  - Pad (tag=0);
  - Subnet mask (tag=1);
  - Time offset (tag=2);
  - List of IP addresses of Gateways (tag=3);
  - Time server's IP address (tag=4);
  - Name Server (tag=5);
  - Domain Name Server (tag=6);
  - LOG server (tag=7); …
  - LPR server (tag=9); …
  - this Host's name (tag=12);
  - Boot file size (tag=13); …
  - Domain name (tag=15); …
  - End (tag=255)

DHCP: Dynamic Host Configuration Protocol (RFC 1531)

Extends the Vendor specific options area by 312 bytes.

This protocol is designed to make it easier to allocate (and reallocate) addresses for clients. DHCP defines:

- Requested IP Address - used in client request (DHCPDISCOVER) to request that a particular IP address (tag=50)
- IP Address Lease Time - used in a client request (DHCPDISCOVER or DHCPREQUEST) to request a lease time for the IP address. In a server reply (DHCPOFFER), specific lease time offered. (tag=51)
- Option Overload - used to indicate that the DHCP “sname” or “file” fields are being overloaded by using them to carry DHCP options. A DHCP server inserts this option if the returned parameters will exceed the usual space allotted for options, i.e., it uses the sname and file fields for another purpose! (tag=52)

- DHCP Message Type - the type of the DHCP message (tag=53)

Message Type	purpose
1	DHCPDISCOVER
2	DHCPOFFER
3	DHCPREQUEST
4	DHCPDECLINE
5	DHCPACK
6	DHCPNAK
7	DHCPRELEASE

- Server Identifier - used in DHCPOFFER and DHCPREQUEST (optionally in DHCPACK and DHCPNAK) messages. Servers include this in the DHCPOFFER to allow the client to distinguish **between** lease offers. DHCP clients indicate which of several lease offers is being accepted by including this in a DHCPREQUEST message. (tag=54)
- Parameter Request List - used by a DHCP client to request values for specified configuration parameters. The client **may** list options in order of preference. The DHCP server **must** try to insert the requested options in the order requested by the client. (tag=55)

- Message - used by a server to provide an error message to client in a DHCPNAK message in the event of a failure. A client may use this in a DHCPDECLINE message to indicate the reason **why** the client declined the offered parameters.(tag=56)
- Maximum DHCP Message Size - specifies the maximum length DHCP message that it is willing to accept. A client may use the maximum DHCP message size option in DHCPDISCOVER or DHCPREQUEST messages, but should not use the option in DHCPDECLINE messages. (tag=57)
- Renewal (T1) Time Value - specifies the time interval from address assignment until the client transitions to the RENEWING state. (tag=58)
- Rebinding (T2) Time Value - specifies the time interval from address assignment until the client transitions to the REBINDING state.(tag=59)
- Class-identifier - used by DHCP clients to optionally identify the type and configuration of a DHCP client. Vendors and sites may choose to define specific class identifiers to convey particular configuration or

other identification information about a client. Servers not equipped to interpret the class-specific information sent by a client **must** ignore it (although it may be reported). (tag=60)

- Client-identifier - used by DHCP clients to specify their unique identifier. DHCP servers use this value to index their database of address bindings. This value is expected to be unique for all clients in an administrative domain. (tag=61)

DHCP's importance

- allows reuse of address, which avoids having to tie up addresses for systems which are not currently connected to the Internet
- avoids user configuration of IP address (avoids mistakes and effort)
- allows recycling of an IP address when devices are scrapped
- …

How big a problem is manual configuration?

A large site (such as DuPont Co. - a large chemical company) has over 65,000 IP addressable devices; or consider what happens if each of the 815,000 Wal-Mart employees has an IP device

Address management software

Product	Vendor	URL
Network Registrar	Cisco	http://www.cisco.com
NetID	Nortel Networks	http://www.nortelnetworks.com
Meta IP 4.1	CheckPoint	http://www.metaip.checkpoint.com
QIP Enterprise 5.0	Lucent Technologies	http://qip.lucent.com

DHCP performance problems

Most implementations of DHCP do a duplicate address detection (DAAD) test after they have picked an address to assign.

An alternative approach to speed up the DHCP process does the duplicate address detection process in the background (in advance) so that you will have a set of recently tested addresses to hand out:

> Jon-Olov Vatn and Gerald Q. Maguire Jr., "The effect of using co-located care-of addresses on macro handover latency", Fourteenth Nordic Tele-traffic Seminar (NTS 14), August 18 - 20, 1998, Lyngby, Denmark.
>
> http://www.it.kth.se/~vatn/research/techrep.ps.gz
> or *http://www.it.kth.se/~vatn/research/nts14-coloc.pdf*

The result is that a DHCP request can be answered in less than 100ms.

Example of dhcpd.conf

```
### Managed by Linuxconf, you may edit by hand.
### Comments may not be fully preserved by linuxconf.
server-identifier dhcptest1;
default-lease-time 1000;
max-lease-time 2000;
option domain-name           "3ctechnologies.se";
option domain-name-servers   130.237.12.2;
option host-name             "s1.3ctechnologies.se";
option routers               130.237.12.2;
option subnet-mask           255.255.255.0;
subnet 130.237.12.0 netmask 255.255.255.0 {
          range  130.237.12.3 130.237.12.200;
          default-lease-time 1000;
          max-lease-time 2000;
}
subnet 130.237.11.0 netmask 255.255.255.0 {
           range  130.237.11.3 130.237.11.254;
           default-lease-time 1000;
           max-lease-time 2000;
}
```

DHCP and DNS

- There is no dynamic host name assignment yet.
- Interaction between DHCP and DNS is needed.

For example: once a host is assigned an IP address the DNS should be updated dynamically:

- If the host hasn't got a name: it should assign a name along the IP address assignment (no DNS update is needed).
- If the host has already a name: the DNS should be dynamically updated once the host has gotten a new IP address from DHCP.

The IETF's Dynamic Host Configuration (dhc) Working group *http://www.ietf.org/html.charters/dhc-charter.html* is working on addressing the issues concerning interaction between DHCP and DNS.

Trivial File Transfer Procotol (TFTP)

TFTP uses UDP (unlike FTP which uses TCP)

- simple and small
- requires only UDP, IP, and a device driver - easily fits in ROM
- a stop-and-wait protocol
- lost packets detected by timeout and retransmission
- Two operations:
  - Read Request (RRQ)
  - Write Request (RRQ) - for security reasons the file must already exist
- The TFTP server (“tftpd”) is generally run setrooted (i.e., it only has access to is own directory) and with a special **user** and **group** ID - since there is no password or other protection of the access to files via TFTP!
- TFTP request is sent to the well know port number (69/udp)
- TFTP server uses an unused ephemeral port for its replies
  - since a TFTP transfer can last for quite some time - it uses another port; thus freeing up the well known port for other requests

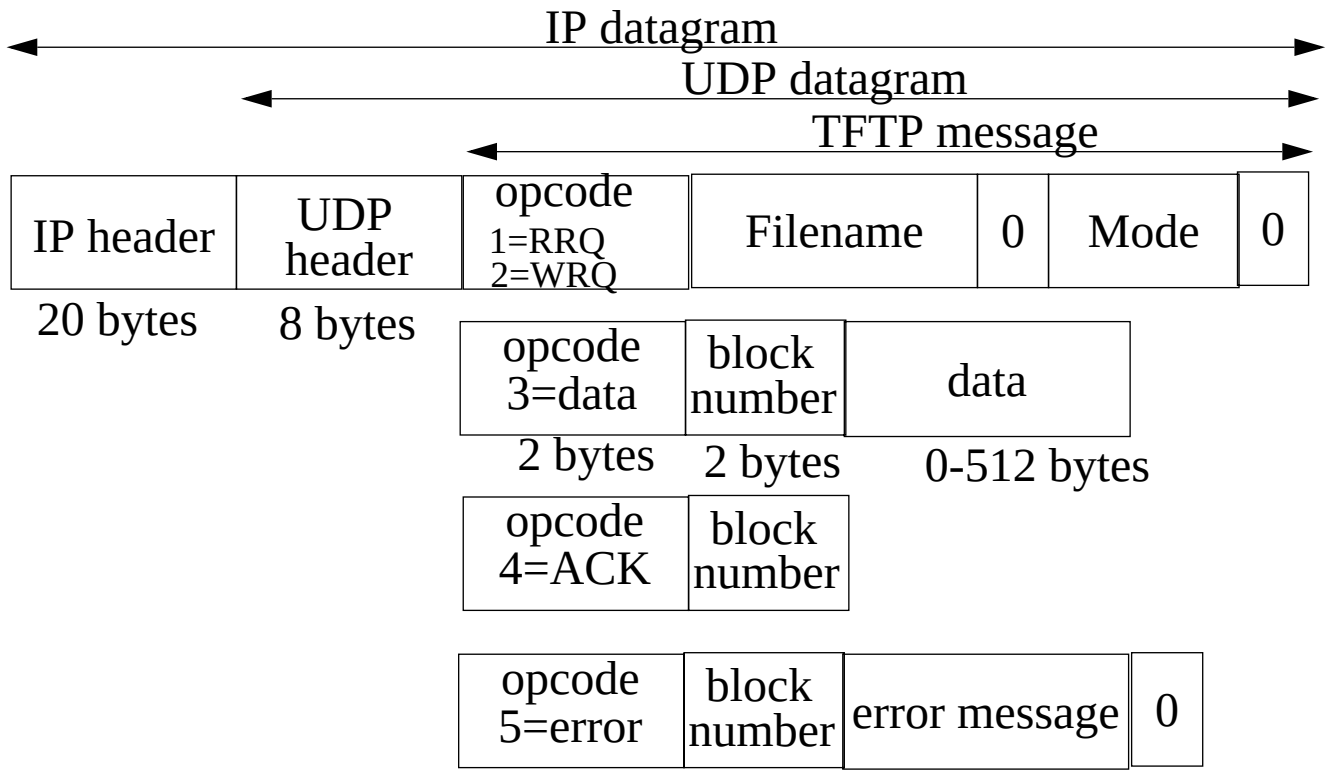


Figure 46: TFTP messages (see Stevens, Vol. 1, figure 15.1, pg. 210)

Filename and Mode (“netascii” or “octet”) are both N bytes sequences terminated by a null byte.

Widely used for bootstrapping diskless systems (such as X terminals) and for dumping the configuration of routers (this is where the write request is used)

Mapping names to IP addresses

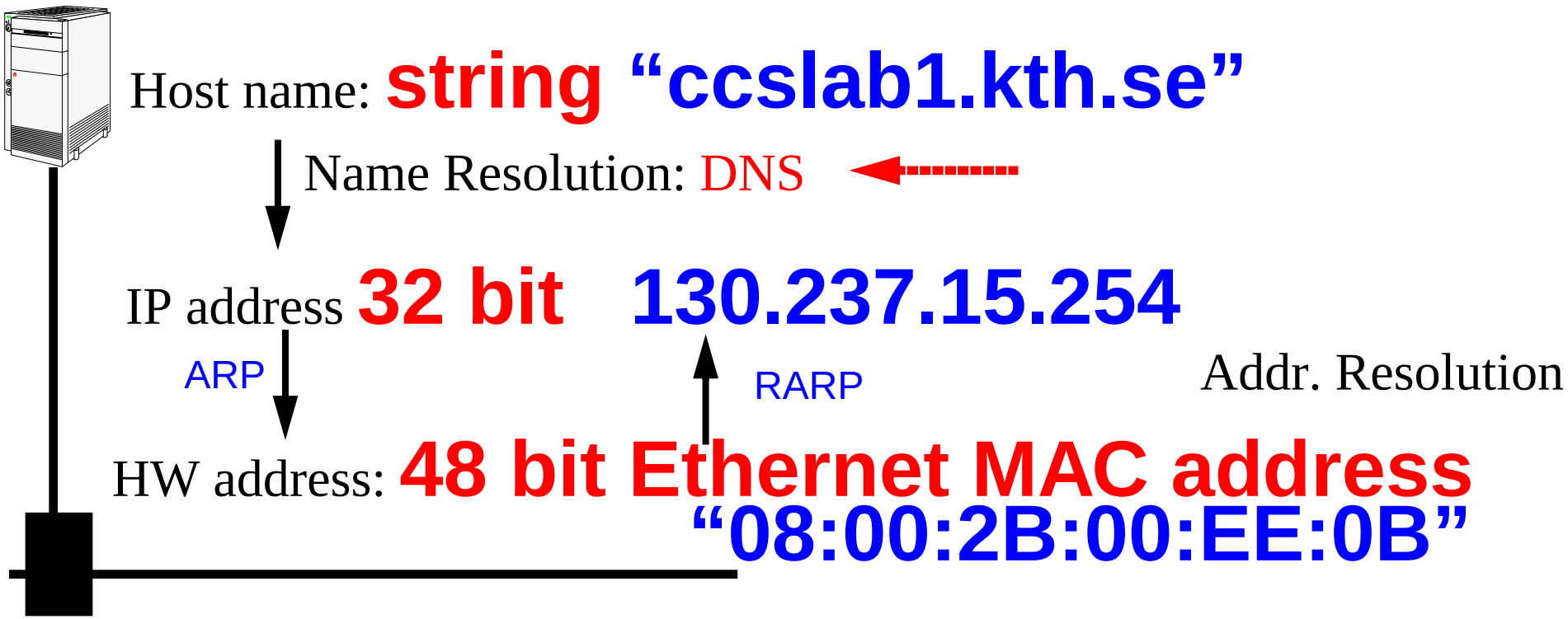


Figure 47: mapping between host names and IP address(es)

DNS: Domain Name Service (RFC 1034, RFC 1035)

- To make the network more user friendly
- Distributed database (with caching) providing:
  - hostname ⇒ IP address, IP address⇒ hostname
  - mailbox ⇒ mail server
  - …
- applications call a “resolver”
  - gethostbyname: hostname ⇒ IP address
  - gethostbyaddr: IP address⇒ hostname
- Resolver’s contact name servers (see “/etc/resolv.conf”)
- DNS names:
  - domain name: list of labels from a root, i.e., www.imit.kth.se
  - Fully Qualified Name (FQDN): a domain name ending in “.” - there are no further labels
  - leaves are managed locally through delegation of authority (to a zone) **not** centrally; this allows scaling
  - if a name server does not know the answer it asks other name servers
    - every name server **must** know how to contact a root server
- Uses UDP (for query) and TCP (zone transfer and large record query)

Zones

A zone is a subtree of the DNS tree which is managed separately.

Each zone must have multiple name servers:

- a primary name server for the zone
  - gets its data from disk files (or other stable store)
  - must know the IP address of one or more root servers
- one or more secondary name servers for the zone
  - get their data by doing a zone transfer from a primary
  - generally query their primary server every ~3 hours

To find a server you may have to walk the tree up to the root or possibly from the root down (but the later is not friendly).

DNS Message format

0	16	31

Identification	Parameters
Number of Questions	Number of Answers
Numer of authority	Number of Additional
Question section …	
Answer section …	
Additional Information section …	

Bit or Parameter field	**Meaning**
0	Operation: 0=Query, 1=Response
1-4	Query type: 0=standard, 1=Inverse
5	Set if answer is authoritative
6	Set if answer is truncate
7	Set if answer is desired
8	Set if answer is available
9-11	reserved
12-15	Response Type: 0=No error, 1=Format error in query, 2=Server failure, 3=Name does not exist

Internet's top level domains

(see Stevens, Vol. 1, figure 14.2, pg. 189)

Domain	Description
com	commercial organizations
edu	educational organizations
gov	other U.S. government organizations (see RFC 1811 for policies)
int	international organizations
mil	U.S. Military
net	networks
org	other organizations
arpa	special domain for address to name mappings, e.g., 5.215.237.130.in-addr.arpa
ae	United Arab Emerates
…	
se	Sweden
zw	Zimbabwe

Lots of interest in having subdomains of "com"

- companies registering product names, etc. - in some cases asking for 10s of addresses
- who gets to use a given name? problems with registered trade marks, who registered the name first, … [How much is a name worth?]

New top level domains

There is a proposed new set of top level domains and an increase in the number of entities which can assign domain names.

Generic Top Level Domains (gTLDs), November 16, 2000:

.aero	for the entire aviation community
.biz	for business purpose
.coop	for cooperatives
.info	unrestricted
.museum	for museums
.name	for personal names
.pro	for professionals

CORE (Council of Registrars) - operational organization composed of authorized Registrars for managing allocations under gTLDs.

WIPO provides arbitration concerning names:
http://arbiter.wipo.int/domains/gtld/newgtld.html

Domain registrars

Internet Corporation for Assigned Names and Numbers (ICANN) Accredited Registrars, the full list is at
http://www.icann.org/registrars/accredited-list.html

Even more registrars are on their way to being accredited and operating!

Country Code Top-Level Domains (CCTLDs)

http://www.iana.org/cctld/cctld-whois.htm

For Sweden (SE) SE-DOM, the NIC is: *http://www.nic-se.se/*

	Administrative contact	**Technical contact**
Address	II-Stiftelsen Sehlstedtsgatan 7 SE-115 28 Stockholm Sweden	Network Information Centre Sweden NIC-SE Box 5774 SE-114 87 Stockholm Sweden
e-mail	ii-stiftelsen@iis.se	hostmaster@nic-se.se
phone	+46 8 56849050	+46 8 54585700
fax	+46 8 50618470	+46 8 54585729

The above is from *http://www.iana.org/root-whois/se.htm*

URL for registration services: *http://www.iis.se/*

Resource Records (RR)

See Stevens, Vol. 1, figure 14.2, pg. 201 (augmented with additional entires)

Record type	Description
A	an IP address. Defined in RFC 1035
AAAA	an IPv6 address. Defined in RFC 1886
PTR	pointer record in the in-addr.arpa format. Defined in RFC 1035.
CNAME	canonical name≡ alias (in the format of a domain name). Defined in RFC 1035.
HINFO	Host information. Defined in RFC 1035.
MX	Mail eXchange record. Defined in RFC 1035.
NS	authoritative Name Server (gives authoritative name server for this domain).Defined in RFC 1035.
TXT	other attributes. Defined in RFC 1035.
AFSDB	AFS Data Base location. Defined in RFC 1183.
ISDN	ISDN. Defined in RFC 1183.
KEY	Public key. Defined in RFC 2065.
KX	Key Exchanger. Defined in RFC 2230.
LOC	Location. Defined in RFC 1876.
MG	mail group member. Defined in RFC 1035.
MINFO	mailbox or mail list information. Defined in RFC 1035.
MR	mail rename domain name. Defined in RFC 1035.
NULL	null RR. Defined in RFC 1035.
NS	Name Server. Defined in RFC 1035.

See Stevens, Vol. 1, figure 14.2, pg. 201 (augmented with additional entires)

Record type	Description
NSAP	Network service access point address. Defined in RFC 1348. Redefined in RFC 1637. Redefined in RFC 1706.
NXT	Next. Defined in RFC 2065.
PX	Pointer to X.400/RFC822 information. Defined in RFC 1664.
RP	Responsible Person. Defined in RFC 1183.
RT	Route Through. Defined in RFC 1183.
SIG	Cryptographic signature. Defined in RFC 2065.
SOA	Start Of Authority. Defined in RFC 1035.
SRV	Server. DNS Server resource record -- RFC 2052, for use with DDNS.
TXT	Text. Defined in RFC 1035.
WKS	Well-Known Service. Defined in RFC 1035.
X25	X25. Defined in RFC 1183.

Note that an number of the RR types above are for experimental use.

Name of an organization:

ISI.EDU. PTR 0.0.9.128.IN-ADDR.ARPA.

Network names

Conventions:

- it.kth.se includes all the computers in the KTH/SU IT-University
- kth.se includes all the computers at KTH
- …

As resource records:

```
> set querytype=any
> kth.se
…
Non-authoritative answer:
kth.se                  internet address = 130.237.72.201
kth.se

                        origin = kth.se
                        mail addr = hostmaster.kth.se
                        serial = 2002011500
                        refresh = 3600 (1H)
```

	retry = 600 (10M)
	expire = 604800 (1W)
	minimum ttl = 86400 (1D)
kth.se	nameserver = kth.se
kth.se	nameserver = nic.lth.se
kth.se	nameserver = ns.kth.se

Authoritative answers can be found from:

kth.se	nameserver = kth.se
kth.se	nameserver = nic.lth.se
kth.se	nameserver = ns.kth.se
kth.se	internet address = 130.237.x.y
nic.lth.se	internet address = 130.235.z.w
ns.kth.se	internet address = 130.237.m.n

ARPANET.ARPA. PTR 0.0.0.10.IN-ADDR.ARPA.

isi-net.isi.edu. PTR 0.0.9.128.IN-ADDR.ARPA.

Example:

```
$ORIGIN it.kth.se.
@                       1D IN SOA       bbbb hostmaster (
                                                2002012001      ; serial
                                                8H              ; refresh
                                                2H              ; retry
                                                2W              ; expiry
                                                8H )            ; minimum

                1D      IN NS           ns.ele.kth.se.
                1D      IN NS           ns.kth.se.
                1D      IN MX           0 mail
                1D      IN A            130.237.x.y
                1D      IN AFSDB        1 xxxx
                1D      IN AFSDB        1 yyyy
                1D      IN AFSDB        1 zzzz
```

MX information

```
> set querytype=MX
> kth.se
…
kth.se                  preference = 0, mail exchanger = mail1.kth.se
kth.se                  nameserver = kth.se
kth.se                  nameserver = nic.lth.se
kth.se                  nameserver = ns.kth.se
mail1.kth.se            internet address = 130.237.32.62
kth.se                  internet address = 130.237.72.201
nic.lth.se              internet address = 130.235.20.3
ns.kth.se               internet address = 130.237.72.200
```

Another examine in MX RR format:

```
1D      IN MX       10 xxx.e.kth.se.
1D      IN MX       20 yyy.e.kth.se.
```

Host names and info

How to give your host a name?

see RFC 1178: Choosing a Name for Your Computer

Internet Addresses: A second address for your host?

- to have multiple addresses for you computer, see section on “ifconfig”

Hostinfo (HINFO)

```
> set querytype=HINFO
> kth.se
…
kth.se                          CPU = sun-4/60          OS = unix
```

Entry	owner	clas	TTL	RR type	value	comment
xxxx			1D	IN HINFO	"PC" "FLINUX"	; “CPU” “OS”
			1D	IN A	130.237.x.y	

Storing other attributes - TXT records

The general syntax is:

<owner> <class> <ttl> TXT “<attribute name>=<attribute value>”

Examples:

host.widgets.com IN TXT “printer=lpr5”

sam.widgets.com IN TXT “favorite drink=orange juice”

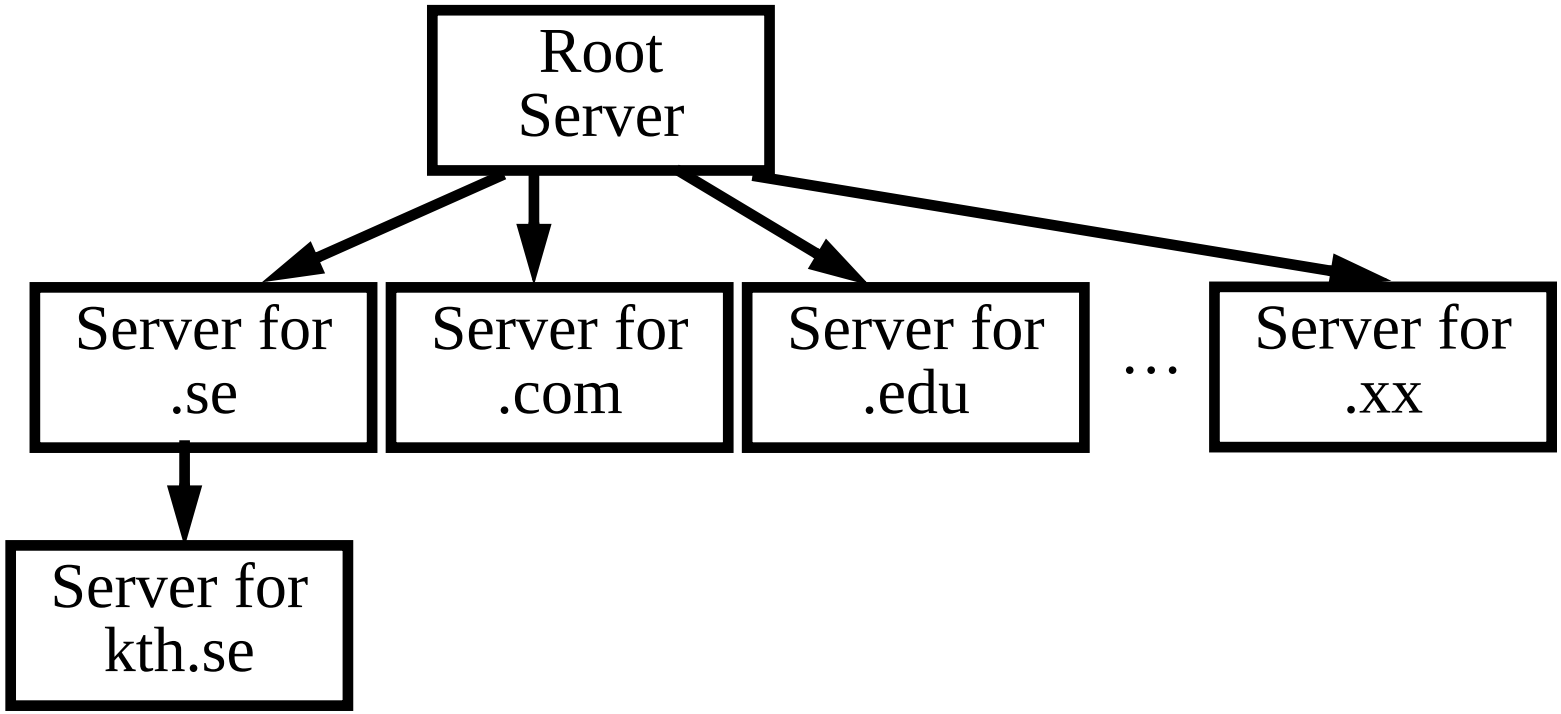
For more information see:

RFC 1464: *Using the Domain Name System To Store Arbitrary String Attributes*

Configuring DNS

- Configuring the BIND resolver
  - /etc/resolv.config
- Configuring the BIND nameserver (named)
  - /etc/named.boot or /etc/named.config
- Configuring the nameserver database files (zone files)
  - named.hosts the zone file that maps hostnames to IP addresses
  - named.rev the zone file that maps IP addresses to hostnames

Dynamic Domain Name System (DDNS)



Host Name	IP-address
host_a	130.237.x.1
host_b	130.237.x.2
host_c	130.237.x.3
host_d	130.237.x.4
mobile1	c/o address <<< we can update this dynamically

DDNS

RFC 2136: Dynamic Updates in the Domain Name System (DNS UPDATE)

- add or delete resource records

RFC 2052: A DNS RR for specifying the location of services (DNS SRV)

- When a SRV-cognizant web-browser wants to retrieve

  http://www.asdf.com/

  it does a lookup of

  http.tcp.www.asdf.com

RFC 2535: Domain Name System Security Extensions (DNSSec)

Multicast and IGMP

Broadcast and Multicast

Traditionally the Internet was designed for unicast communication (one sender and one receiver) communication.

Increasing use of multimedia (video and audio) on the Internet

- O**ne-to-many** and **many-to-many** communication is increasing
- In order to support these uses in a *scalable* fashion we use the technical method of **multicasting**.
- Replicating UDP packets where paths diverge (i.e., split)

Mbone is an experimental multicast network which has been operating for a number of years.

Multicasting is useful for:

- Delivery to multiple recipients
  - reduces traffic, otherwise each would have to be sent its own copy
- Solicitation of service (service/server discovery)
  - Not doing a broadcast saves interruptings many clients

Filtering up the protocol stack

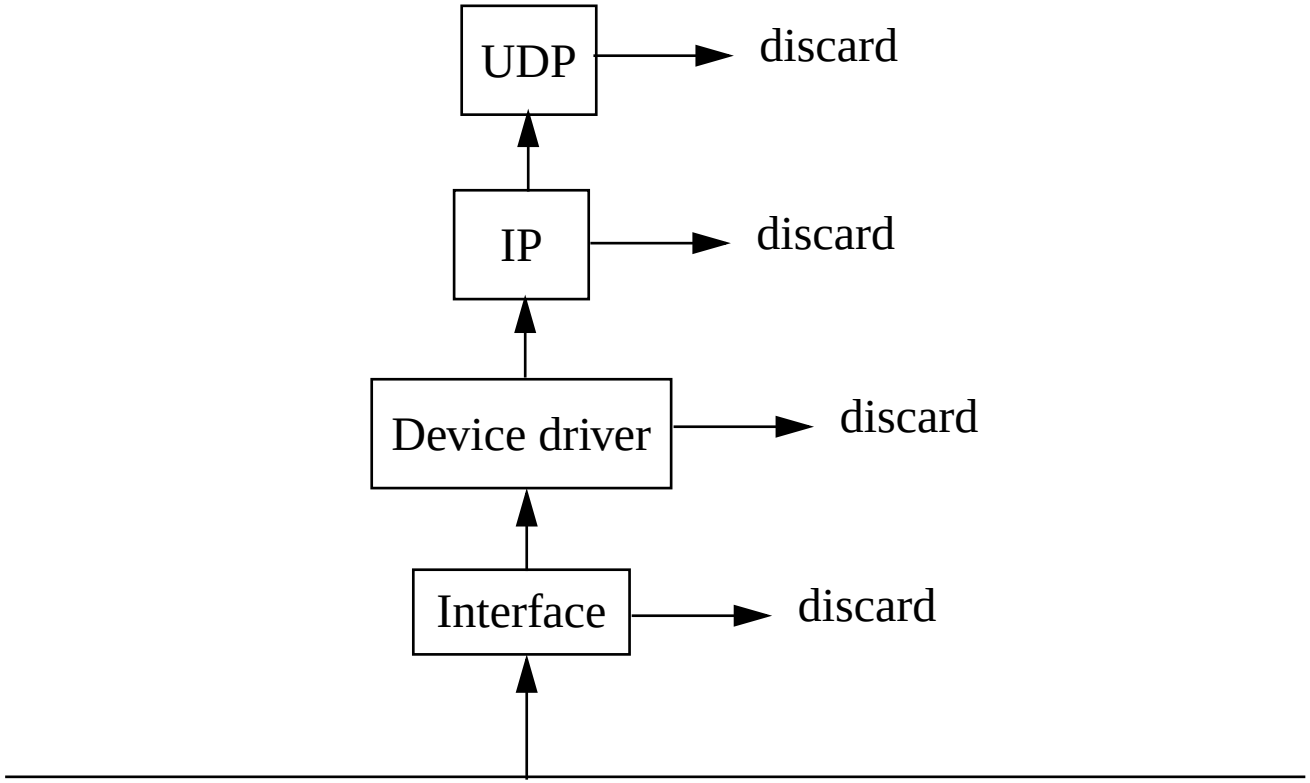


Figure 48: Filtering which takes place as you go up the TCP/IP stack
(see Stevens, Vol. 1, figure 12.1, pg. 170)

We would like to filter as soon as possible to avoid load on the machine.

Broadcasting

- Limited Broadcast
  - IP address: 255.255.255.255
  - never forwarded by routers
  - What if you are multihomed? (i.e., attached to several networks)
    - Most BSD systems just send on first configured interface
    - routed and rwhod - determine all interfaces on host and send a copy on each (which is capable of broadcasting)
- Net-directed Broadcast
  - IP address: netid.255.255.255 or net.id.255.255 or net.i.d.255 (depending on the class of the network)
  - routers must forward
- Subnet-Directed Broadcast
  - IP address: netid | subnetid | hostID, where hostID = all ones
- All-subnets-directed Broadcast
  - IP address: netid | subnetid | hostID, where hostID = all ones and subnetID = all ones
  - generally regarded as obsolete!

To send a UDP datagram to a broadcast address set SO_BROADCAST

Other approaches to One-to-Many and Many-to-Many communication

Connection oriented approaches have problems:

- large user burden
- have to know other participants
- have to order links in advance
- poor scaling, worst case $O(N^2)$

Alternative centralized model

CU-SeeME uses another model - a Reflector (a centralized model)

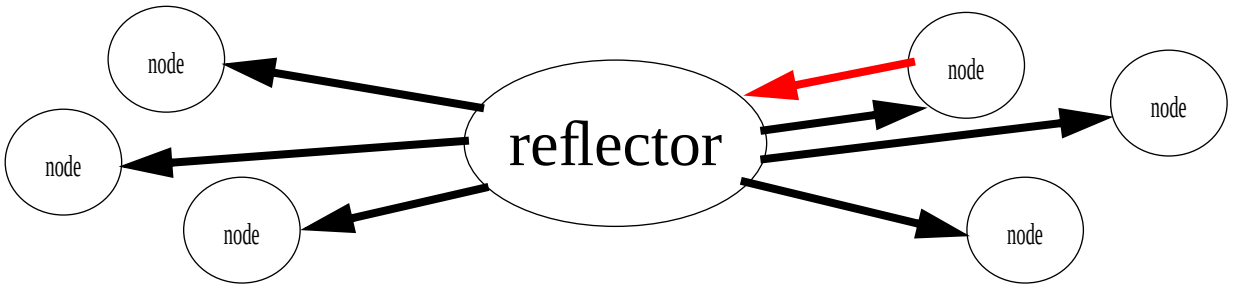


Figure 49: Reflector

- All sites send to one site (the reflector) overcomes the N^2 problems
- The reflector sends copies to all sites

Problems:

- Does not scale well
- Multiple copies sent over the same link
- Central site must know all who participate

Behavior could be changed by explicitly building a tree of reflectors - but then you are moving over to Steve Deering's model.

Multicast Backbone (MBONE)

Expanding multicasting across WANs

World-wide, IP-based, real-time conferencing over the Internet (via the MBONE) in daily use for several years with more than 20,000 users in more than 1,500 networks in events carrier to 30 countries.

See “Introduction to the MBone” at *http://www-itg.lbl.gov/mbone/*

For a nice paper examining multicast traffic see: “Measurements and Observations of IP Multicast Traffic” by Bruce A. Mah <bmah@CS.Berkeley.EDU>, The Tenet Group, University of California at Berkeley,and International Computer Science Institute, CSD-94-858, 1994 ,12 pages:
http://www.employees.org/~bmah/Papers/Ipmcast-TechReport.pdf

IP Multicast scales well

- End-nodes know nothing about topology
  - Dynamically changes of topology possible
- Routers know nothing about “conversations”
  - changes can be done without global coordination
  - no end-to-end state to move around

Participants view of Multicast

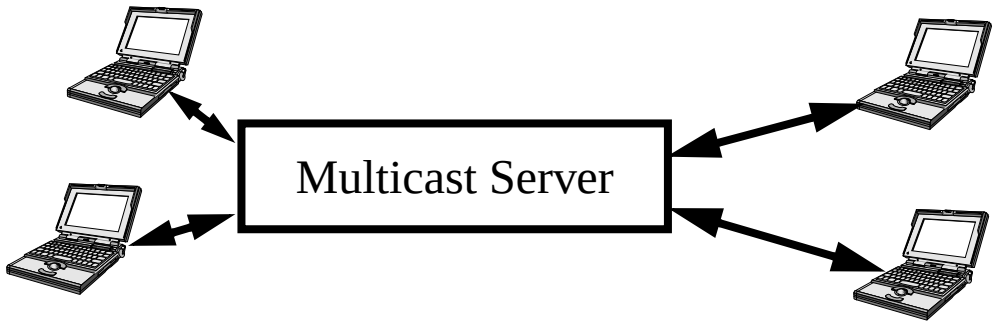


Figure 50: MBONE behaves as if there were a multicast server, but this functionality is distributed not centralized.

Core Problem

How to do efficient multipoint distribution (i.e., at most one copy of a packet crossing any particular link) without exposing topology to end-nodes

Applications

- Conference calls (without sending N copies sent for N recipients)
- Dissemination of information (stock prices, "radio stations", …)
- Dissemination of one result for many similar requests (boot information, video)
- Unix tools (see *http://nic.merit.edu/~mbone/output.html*):
  - nv - network video
  - vat - visual audio tool
  - wb - whiteboard
  - sd - session directory
  - …

Steve Deering's Multicast

Dynamically constructs efficient delivery trees from sender(s) to receiver(s)

- Key is to compute a spanning tree of multicast routers

Simple service model:

- receivers announce interest in some multicast address
- senders just send to that address
- routers conspire to deliver sender's data to all interested receivers
  - so the real work falls once again to the routers, not the end nodes
  - Note that the assumption here is that it is worth loading the routers with this extra work, because it reduces the traffic which has to be carried.

IP WAN Multicast Requirements

- Convention for recognizing IP multicast
- Convention for mapping IP to LAN address
- Protocol for end nodes to inform their adjacent routers,
- Protocol for routers to inform neighbor routers
- Algorithm to calculate a spanning tree for message flow
- Transmit data packets along this tree

Multicasting IP addresses

Multicast Group Addresses - “Class D” IP address

- High 4 bits are 0x1110; which corresponds to the range 224.0.0.0 through 239.255.255.255
- host group ≡ set of hosts listening to a given address
  - membership is dynamic - hosts can enter and leave at will
  - no restriction on the number of hosts in a host group
  - a host need not belong in order to send to a given host group
  - permanent host groups - assigned well know addresses by IANA
    - 224.0.0.1 - all systems on this subnet
    - 224.0.0.2 - all routers on this subnet
    - 224.0.1.1 - Network Time Protocol (NTP) - see RFC 1305 and RFC 1769 (SNTP)
    - 224.0.0.9 - RIP-2
    - 224.0.1.2 - SGI’s dogfight application

Internet Multicast Addresses

http://www.iana.org/assignments/multicast-addresses listed in DNS under MCAST.NET and 224.IN-ADDR.ARPA.

- 224.0.0.0 - 224.0.0.255 (224.0.0/24) Local Network Control Block
- 224.0.1.0 - 224.0.1.255 (224.0.1/24) Internetwork Control Block
- 224.0.2.0 - 224.0.255.0 AD-HOC Block
- 224.1.0.0 - 224.1.255.255 (224.1/16) ST Multicast Groups
- 224.2.0.0 - 224.2.255.255 (224.2/16) SDP/SAP Block
- 224.3.0.0 - 224.251.255.255 Reserved
- 239.000.000.000-239.255.255.255 Administratively Scoped
  - 239.000.000.000-239.063.255.255 Reserved
  - 239.064.000.000-239.127.255.255 Reserved
  - 239.128.000.000-239.191.255.255 Reserved
  - 239.192.000.000-239.251.255.255 Organization-Local Scope
  - 239.252.000.000-239.252.255.255 Site-Local Scope (reserved)
  - 239.253.000.000-239.253.255.255 Site-Local Scope (reserved)
  - 239.254.000.000-239.254.255.255 Site-Local Scope (reserved)
  - 239.255.000.000-239.255.255.255 Site-Local Scope
  - 239.255.002.002 rasadv

Converting Multicast Group to Ethernet Address

Could have been a simple mapping of the 28 bits of multicast group to 28 bits of Ethernet multicast space (which is 2^{27} in size), but this would have meant that IEEE would have to allocate multiple blocks of MAC addresses to this purpose, but:

- they didn't want to allocate multiple blocks to one organization
- a block of 2^{24} addresses costs $1,000 ==> $16K for 2^{27} addresses

Mapping Multicast (Class D) adddress to Ethernet MAC Address

Solution IANA has one block of ethernet addresses 00:00:5e as the high 24 bits

- they decided to give 1/2 this address space to multicast -- thus multicast has the address range: 00:00:5e:00:00:00 to 00:00:5e:7f:ff:ff
- since the first bit of an ethernet multicast has a low order 1 bit (which is the first bit transmitted in link layer order), the addresses are 01:00:5e:00:00:00 to 01:00:5e:7f:ff:ff
- thus there are 23 bits available for use by the 28 bits of the multicast group ID; we just use the bottom 23 bits
  - therefore 32 different multicast group addresses map to the same ethernet address
  - the IP layer will have to sort these 32 out
  - thus although the filtering is not complete, it is very significant

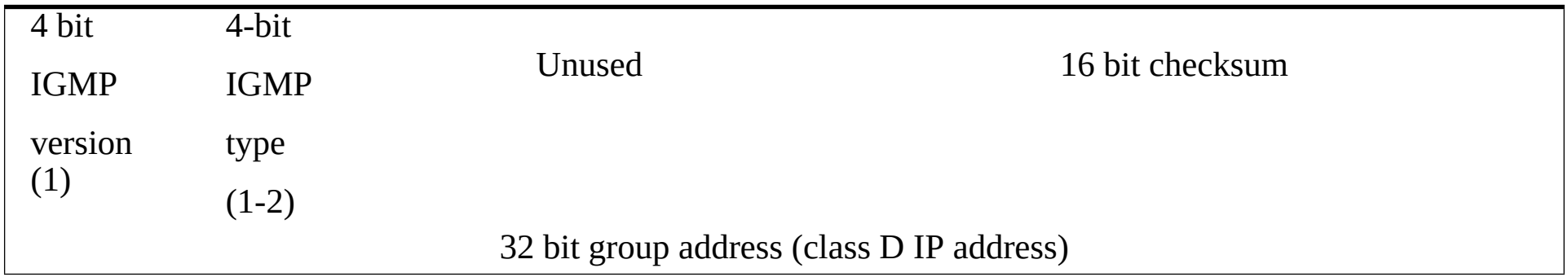
The multicast datagrams are delivered to all processes that belong to the same multicast group.

To extend beyond a single subnet we use IGMP.

IGMP: Internet Group Management Protocol (RFC 1112)

- Used by hosts and routers to know which hosts currently belong to which multicast groups.
- multicast routers have to know which interface to forward datagrams to
- IGMP like ICMP is part of the IP layer and is transmitted using IP datagrams (protocol = 2) I

Figure 51: Encapsulation of IGMP message in IP datagram(see Stevens, Vol. 1, figure 13.1, pg. 179)



- type =1 ⇒ query sent by a router, type =2 ⇒ response sent by a host

Joining a Multicast Group

- a process joins a multicast group on a given interface
- host keeps a table of all groups which have a reference count ≥ 1

IGMP Reports and Queries

- Hosts sends a report when first process joins a given group
- Nothing is sent when processes leave (not even when the last leaves), but the host will no longer send a report for this group
- IGMP router sends queries periodically (one out each interface), the group address in the query is 0

In response to a query, a host sends a IGMP report for every group with at least one process

Routers

- Note that routers have to listen to all 2^{23} link layer multicast addresses!
- Hence they listen promiscuously to all LAN multicast traffic

IGMP Implementation Details

In order to improve its efficiency there are several clever features:

- Since initial reports could be lost, they are resent after a random time [0, 10 sec]
- Reponse to queries are also delayed randomly - but if a node hears someone else report membership in a group it is interested in, its response is cancelled

Note: multicast routers don't care which host is a member of which group; only that *someone* attached to the subnet on a given interface is!

Time to Live

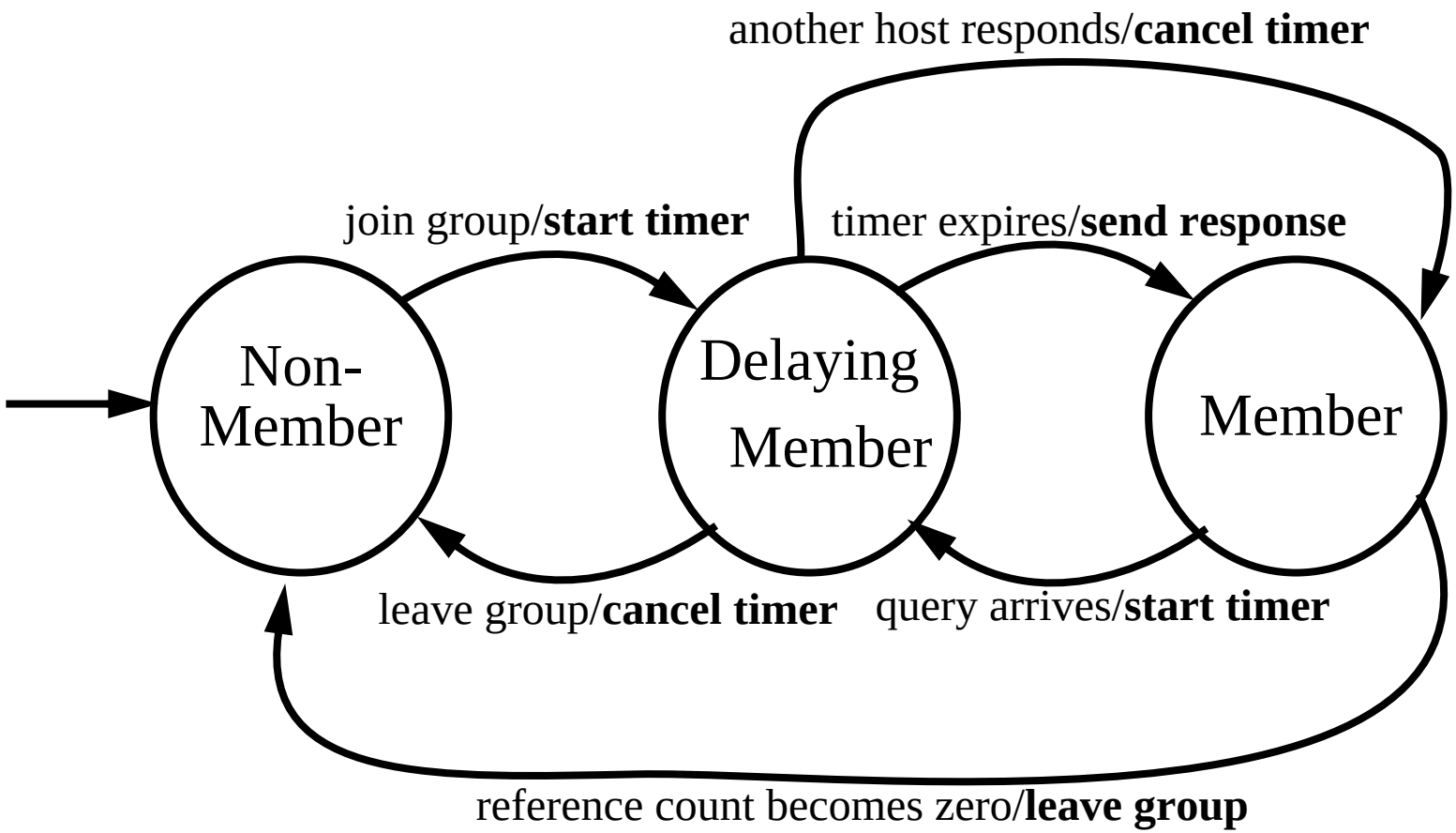
- TTL generally set to 1, but you can perform an expanding ring search for a server by increasing the value
- Addresses in the special range 224.0.0.0 through 224.0.0.255 - should never be forwarded by routers - regardless of the TTL value

All-Hosts Group

- all-hosts group address 224.0.0.1 - consists of all multicast capable hosts and routers on a given physical network; membership is *never* reported

Group membership State Transitions

adapted from Comer figure 17.4 pg. 330



IGMP Version 2

Allows a host to send a message when they want to explicitly leave a group -- after this message the router sends a *group-specific* query to ask if there is anyone still interested in listening to this group.

- however, the router may have to ask multiple times because this query could be lost
- hence the leave is not immediate -- even if there had been only one member (since the router can’t know this)

IGMP Version 3

- Joining a multicast group, but with a specified set of sender -- so that a client can limit the set of senders which it is interested in hearing from.
- all IGMP replies are now set to a single layer 2 multicast address
  - because most LANs are now *switched* rather than shared media -- it uses less bandwidth o not forward all IGMP replies to all ports
  - most switches now support IGMP snooping -- i.e., the switch is IGMP aware and knows which ports are part of which multicast group (this requires the switch to know which ports other switches and routers are on -- so it can forward IGMP replies to them)
    - switches can listen to this specific layer 2 multicast address - rather than having to listen to all multicast addresses
  - it is thought that rather than have end nodes figure out if all the multicast senders which it is interested in have been replied to - simply make the switch do this work.

Telesys class was multicasted over MBONE

Already in Period 2, 1994/1995 "Telesys, gk" was multicast over the internet and to several sites in and near Stockholm.

Established ports for each of the data streams:

- electronic whiteboard
- video stream
- audio stream

The technology works - but it is very important to get the audio packets delivered with modest delay and loss rate. Poor audio quality is perceived a major problem.

NASA and several other organizations regularily multicast their audio and video “programs”.

Benefits for Conferencing

- IP Multicast is efficient, simple, robust
- Users can join a conference without enumerating (or even knowing) other participants
- User can join and leave at any time
- Dynamic membership

MBONE Chronology

Nov. 1988	Small group proposes testbed net to DARPA. This becomes DARTNET
Nov. 1990	Routers and T1 lines start to work
Feb. 1991	First packet audio conference (using ISI's vt)
Apr. 1991	First multicast audio conference
Sept. 1991	First audio+video conference (hardware codec)
Mar. 1992	Deering & Casner broadcast San Diego IETF to 32 sites in 4 countries
Dec. 1992	Washington DC IETF - four channels of audio and video to 195 watchers in 12 countries
Jan. 1993	MBONE events go from one every 4 months to several a day
1994/1995	Telesys gk -- multicast from KTH/IT in Stockholm
July 1995	KTH/IT uses MBONE to multicast two parallel sessions from IETF meeting in Stockholm
...	
today	lots of users and "multicasters"

IETF meetings are *now* regularily multicast - so the number of participants that can attend is not limited by physical space or travel budgets.

MBONE growth

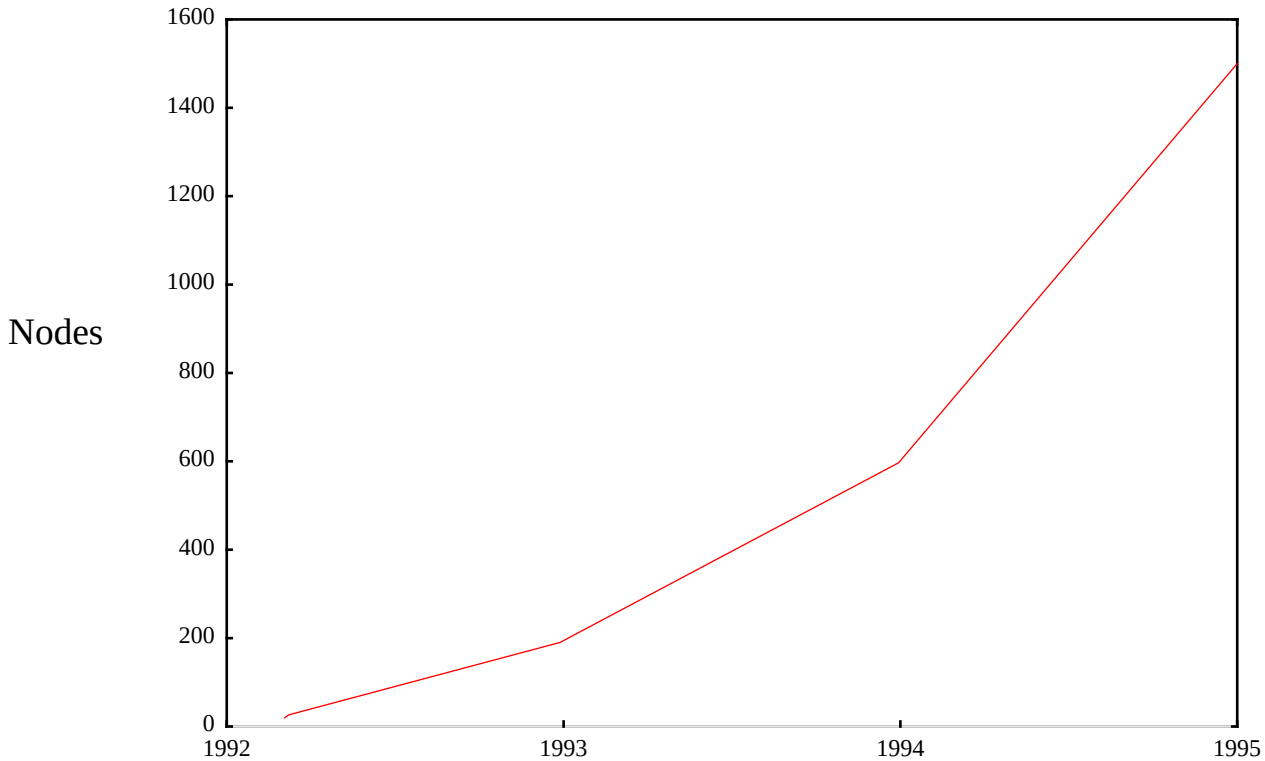


Figure 52: MBONE Growth - Doubling time ~8 months

For the latest statistics see: *http://www.caida.org/tools/measurement/mantra/* as of 01/21/2002,11:30 PST (Pacific Standard Time) there was an average of 1002 groups with on average 4 members per group (in 2000 this was 330 active groups)

MBONE connections

MBONE is an “overlay” on the Internet

- multicast routers were distinct from normal, unicast routers - but increasingly routers support multicasting
- it is not trivial to get hooked up
- requires cooperation from local and regional people

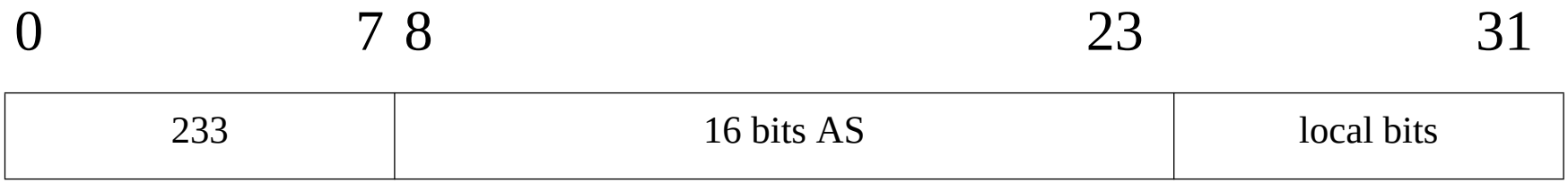
MBONE is changing:

- Most router vendors now support IP multicast
- MBONE will go away as a distinct entity once ubiquitous multicast is supported throughout the Internet.
- Anyone hooked up to the Internet can partipate in conferences

GLOP addressing

Traditionally multicast address allocation has been dynamic and done with the help of applications like SDR that use Session Announcement Protocol (SAP).

GLOP is an example of a policy for allocating multicast addresses (it is still experimental in nature). It allocated the 233/8 range of multicast addresses amongst different ASes such that each AS is statically allocated a /24 block of multicast addresses. See "*RFC 3180*: GLOP Addressing in 233/8" by D. Meyer, P. Lothberg. September 2001.



mrouted

mrouted UNIX deamon

tunneling to other MBONE routers

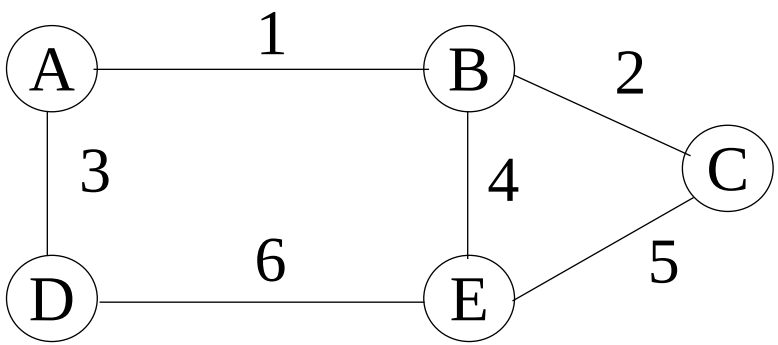
See: “Linux-Mrouted-MiniHOWTO: How to set up Linux for multicast routing” by Bart Trojanowski <bart@jukie.net>, v0.1, 30 October 1999
http://jukie.net/~bart/multicast/Linux-Mrouted-MiniHOWTO.html

and *http://www.linuxdoc.org/HOWTO/Multicast-HOWTO-5.html*

Multicasting

Example: Transmitting a file from C to A, B, and D.

- ✘ Using point-to-point transfer, some links will be used more than once to send the same file



- ✔ Using Multicast

Point-to-point						
Link	A	B	E	D	Total	Multicast
1	1				1	1
2	1	1			2	1
5			1	1	2	1
6				1	1	1
	2	1	1	2		4

Multicast Routing - Flooding

- maintaining a list of recently seen packets (last 2 minutes), if it has been seen before, then delete it, otherwise copy to a cache/database and send a copy on all (except the incoming) interface.

✘ Disadvantages:

- ◆ Maintaining a list of “last-seen” packets. This list can be fairly long in high speed networks
- ◆ The “last-seen” lists guarantee that a router will not forward the same packet twice, but it certainly does not guarantee that the router will receive a packet only once.

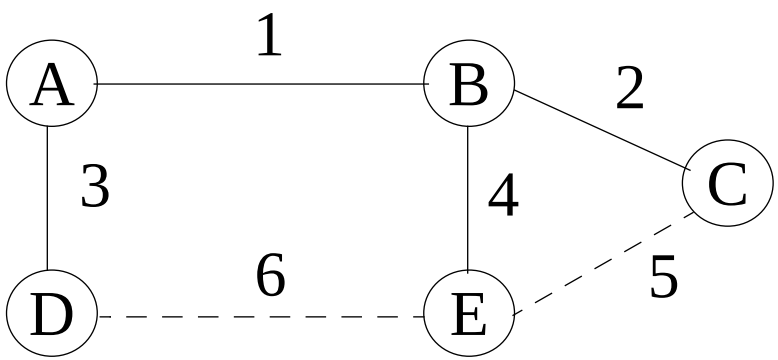
✔ Advantages

- ◆ Robustness
- ◆ It does not depend on any routing tables.

Multicast Routing - Spanning Trees

The “spanning tree” technique is used by “media-access-control (MAC) bridges”.

- Simply build up an “overlay” network by marking some links as “part of the tree” and other links as “unused” (produces a loopless graph).



Drawbacks

- ✘ It does not take into account group membership
- ✘ It concentrates all traffic into a small subset of the network links.

Multicast Routing - Reverse -PAth Forwarding (RPF)

RPF algorithm takes advantage of a routing table to “orientate” the network and to compute an implicit tree per network source.

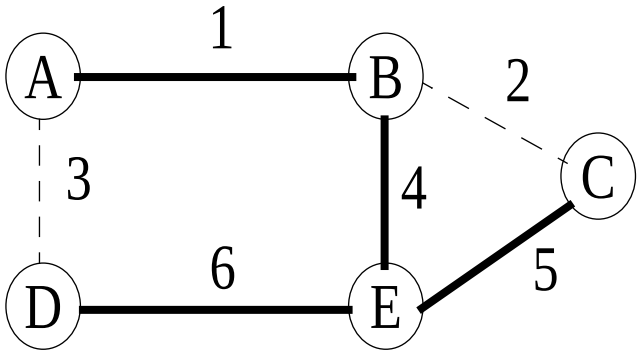
Procedure

1.When a multicast packet is received, note source (S) and interface (I)

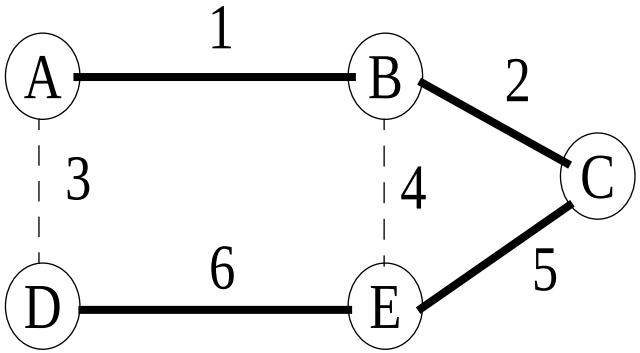
2.If I belongs to the shortest path toward S, forward to all interfaces except I.

- Compute shortest path **from** the **source** to the node rather than from the node to the source.
- Check whether the local router is on the shortest path between a neighbor and the source before forwarding a packet to that neighbor. If this is not the case, there is no point in forwarding a packet that will immediately be dropped by the next router.

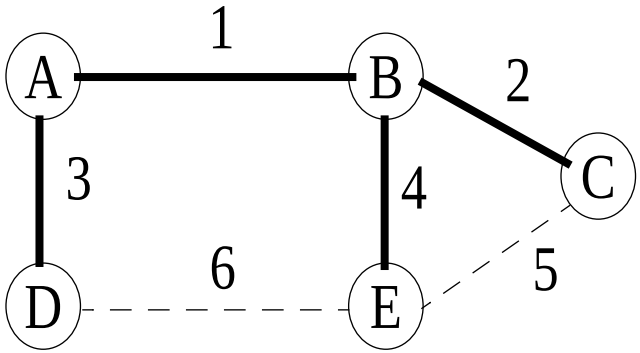
- RPF results in a different spanning tree for each source.



RPF tree from E



RPF tree from C



RPF tree from A

These trees have two interesting properties:

- They guarantee the fastest possible delivery, as multicasting follows the shortest path from source to destination
- Better network utilization, since the packets are spread over multiple links.

Drawback

✘Group membership is not taken into account when building the tree.

Multicast Routing - RPF and Prunes

Improves algorithm with a “flood and prune” option. When a multicast transmission starts from source S, the first packet is propagated to all the network nodes, this is the flooding. Leaf nodes will all receive the first multicast packet. If there is a group member attached to the leaf node, everything is fine, but if there is a group member that does not want to receive further packets, it will send back a “prune message” to the router that sent it this packet - saying effectively “don’t send further packets from source S to group G on this interface I.”

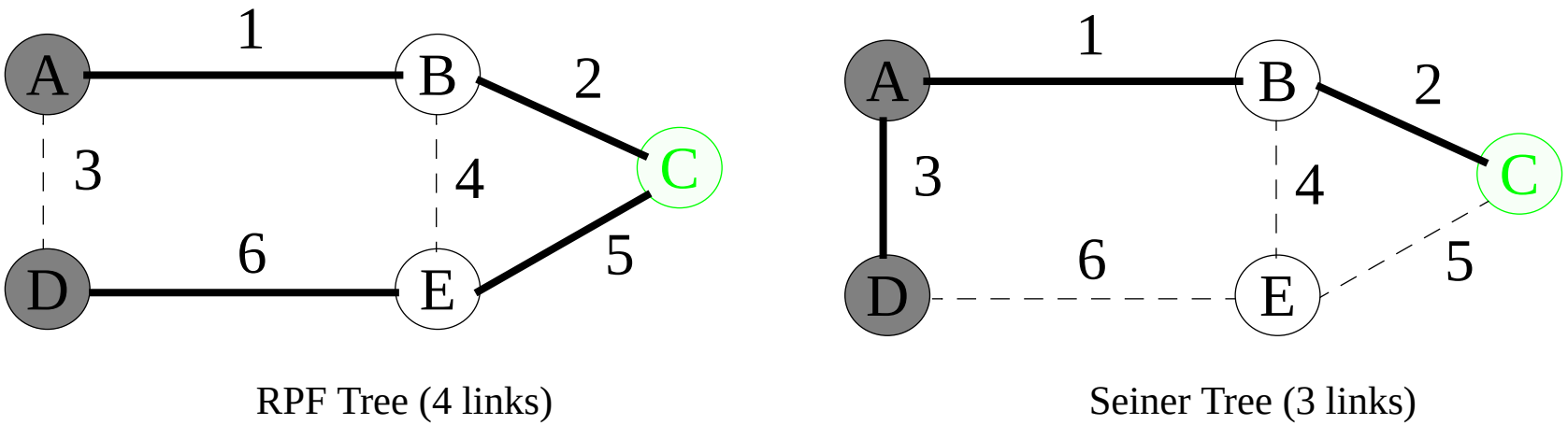
There are two obvious drawback in the flood and prune algorithm:

- The first packet is flooded to the whole network
- The routers must keep states per group and source.

Flood and prune was acceptable in the experimental MBONE which had only a few tens of thousands of nodes. But over the Internet both the number of sources and the number of groups becomes very large, thus there is a risk of exhausting the memory resources in network routers.

Multicast Routing - Steiner Tree's

Assume source C and the recipients are A and D.



RPF Tree (4 links) Seiner Tree (3 links)

Figure 53: RPF vs. Steiner Tree

- Steiner tree uses less resources (links), but are *very hard* to compute (N-P complete)
- In Steiner trees the routing changes widely if a new member joins the group, this leads to instability. Thus the Steiner tree is more a mathematical construct that a practical tool.

Multicast Routing - Core-Based Trees (CBT)

A fixed point in the network will be the center of the multicast group, i.e., “core”. The recipients will then send “join” commands toward the core. These commands will be processed by all intermediate routers, which will mark the interface on which they received the command as belonging to the group’s tree. The routes need to keep one piece of state information per group, listing all the interface that belong to the tree. If the router that receives a join command is already a member of the tree, it will mark only one more interface as belong to the group. If this is the first join command that the router receives, it will forward the command one step further toward the core.

Advantages

- CBT limits the expansion of multicast transmissions to precisely the set of all recipients. This is in contrast with RPF where the first packet is sent to the whole network.
- The amount of state is less; it depends only on the number of the groups, not the number of pairs of sources and groups.
- Since routing is based on a spanning tree, CBT does not depend on multicast or unicast routing tables.

Disadvantages

- The path between some sources and some receivers may be suboptimal.

Multicast Routing - Protocol-Independent Multicast (PIM)

PIM-dense mode and

PIM-sparse mode.

The adjectives “dense” and “sparse: refer to the density of group members in the Internet.

A group is send to be **dense** if the probability is high that the area contains at least one group member. It is send to be **sparse** if that probability is low.

Dense mode is an implementation of RPF and prune strategy.

Sparse mode is an implementation of CBT where join points are called “rendezvous points”.

Scheduling algorithms

Predictable delay is required for interactive real-time applications: Alternatives:

1.use a network which guarantees fixed delays

2.use a packet scheduling algorithm

3.retime traffic at destination

Since queueing at routers, hosts, etc. has traditionally been simply FIFO; which does not provide guaranteed end-to-end delay both the 2nd and 3rd method use alternative algorithms to maintain a predictable dealy.

Algorithms such as: Weighted Fair Queueing (WFQ)

These algorithms normally emulate a fluid flow model.

As it is very hard to provide fixed delays in a network, hence we will examine the 2nd and 3rd methods.

RSVP: Resource Reservation Setup Protocol

- RSVP is a network control protocol that will deal with resource reservations for certain Internet applications.
- RSVP is a component of “Integrated services” Internet, and can provide both best-effort and QoS.
  - Applications request a specific quality of service for a data stream
- RSVP delivers QoS requests to each router along the path.
  - Maintains router and host state along the data stream during the requested service.
  - Hosts and routers deliver these request along the path(s) of the data stream
  - At each node along the path RSVP passes a new resource reservation request to an admission control routine

RSVP is a signalling protocol carrying no application data

- First a host sends IGMP messages to join a group
- Second a host invokes RSVP to reserve QoS

Functionality

- RSVP is receiver oriented protocol.
  The receiver is responsible for requesting reservations.
- RSVP handles heterogeneous receivers.
  Hosts in the same multicast tree may have different capabilities and hence need different QoS.
- RSVP adapts to changing group membership and changing routes.
  RSVP maintains “Soft state” in routers. The only permanent state is in the end systems. Each end system sends their RSVP control messages to refresh the router state.
  In the absence of refresh message, RSVP state in the routers will time-out and be deleted.
- RSVP is **not** a routing protocol.
  A host sends IGMP messages to join a multicast group, but it uses RSVP to reserve resources along the delivery path(s) from that group.

Resource Reservation

- Interarrival variance reduction / jitter
- Capacity assignment / admission control
- Resource allocation (who gets the bandwidth?)

Jitter Control

- if network has enough capacity
  average departure rate = receiver arrival rate
- Then jitter is caused by queue waits due to competing traffic
- Queue waits should be at most the amount of competing traffic in transit, total amount of in transit data should be at most round trip propagation time
  (100 ms for transcontinental path)
  (64 kbit/sec => buffer = 8 kb/s*0.1 sec = 800 bytes)

See: Jonathan Rosenberg,Lili Qiu, and Henning Schulzrinne, “Integrating Packet FEC into Adaptive Voice Playout Buffer Algorithms on the Internet”,INFOCOM, (3), 2000, pp. 1705-1714.

See also *http://citeseer.nj.nec.com/rosenberg00integrating.html*

Capacity Assigment

- end-nodes ask network for bandwidth.
- Can get “yes” or “no” (busy signal)
- Used to control available transmission capacity

RSVP Protocol Mechanism

- Sender sends RSVP PATH message which records path
- Receiver sends RSVP RESV message backwards along the path indicating desired QoS
- In case of failure a RSVP error message is returned

Figure 54:

RSVP Soft State

- “soft state” in hosts and routers
- create by PATH and RESV messages
- refreshed by PATH and RESV messages
- Time-outs clean up reservations
- Removed by explicit “tear-down” messages

RSVP operation

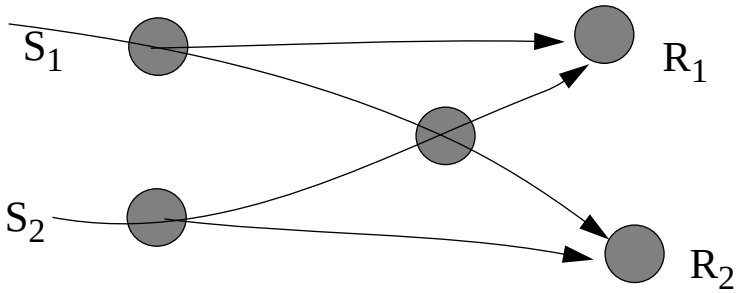


Figure 55:

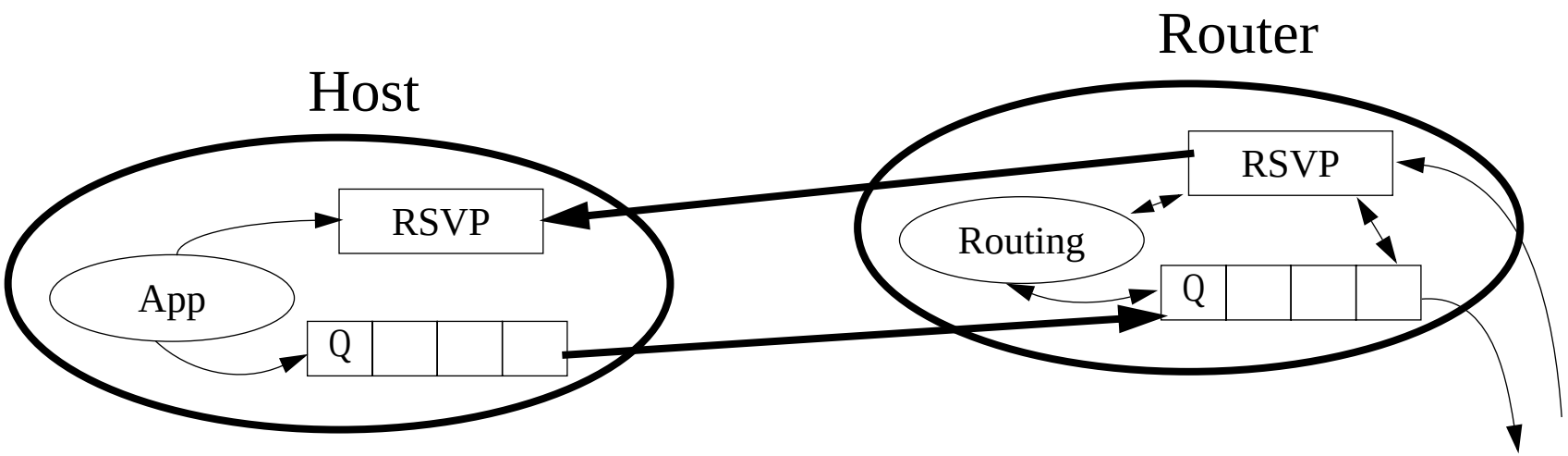


Figure 56:

RSVP operations (continued)

- At each node, RSVP applies a local decision procedure “admission control” to the QoS request. If the admission control succeeds, it set the parameters to the classifies and the packet schedule to obtain the desired QoS. If admission control fails at any node, RSVP returns an error indication to the application.
- Each router in the path capable of resource reservation will pass incoming data packets to a packet classifier and then queue these packet in the packet scheduler. The packet classifier determines the route and the QoS class for each packet. The schedule allocates a particular outgoing link for packet transmission.
- The packet schedule is responsible for negotiation with the link layer to obtain the QoS requested by RSVP. The scheduler may also negotiate a “CPU time”.

RSVP Summary

- RSVP supports multicast and unicast data delivery
- RSVP adapts to changing group membership and routes
- RSVP reserves resources for simplex data streams
- RSVP is receiver oriented, i.e., the receiver is responsible for the initiation and maintenance of a flow
- RSVP maintains a “soft-state” in routers, enabling them to support gracefully dynamic memeberships and automaticallt adapt to routing changes
- RSVP provides several reservation models
- RSVP is transparent for routers that do not provide it

Argument against Reservation

Given, the US has 126 million phones:

- Each conversation uses 64 kbit/sec per phone
- Therefore the total demand is: 8×10^{12} b/s (1 Tbyte/s)

One optical fiber has a bandwidth of ~25×10^{12} b /s

There are well over 1000 transcontinental fibers!

Why should bandwidth be a problem ?

Tools for managing multicast

“Managing IP Multicast Traffic” A White Paper from the IP Multicast Initiative (IPMI) and Stardust Forums for the benefit of attendees of the 3rd Annual IP Multicast Summit, February 7-9, 1999

http://techsup.vcon.com/whtpprs/Managing%20IP%20Multicast%20Traffic.pdf

Mrinfo	shows the multicast tunnels and routes for a router/mrouted.
Mtrace	traces the multicast path between two hosts.
RTPmon	displays receiver loss collected from RTCP messages.
Mhealth	monitors tree topology and loss statistics.
Multimon	monitors multicast traffic on a local area network.
Mlisten	captures multicast group membership information.
Dr. Watson	collects information about protocol operation.

Mantra (Monitor and Analysis of Traffic in Multicast Routers)
http://www.caida.org/tools/measurement/mantra/

SNMP-based tools and multicast related MIBs

Management Information Bases (MIBs) for multicast:

RTP MIB	designed to be used by either host running RTP applications or intermediate systems acting as RTP monitors; has tables for each type of user; collect statistical data about RTP sessions.
Basic Multicast Routing MIB	includes only general data about multicast routing. such as multicast group and source pairs; next hop routing state, forwarding state for each of a router's interfaces, and information about multicast routing boundaries.

Protocol-Specific Multicast Routing MIBs

Provide information specific to a particular routing protocol

PIM MIB	list of PIM interfaces that are configured; the router's PIM neighbors; the set of of rendezvous points and an association for the multicast address prefixes; the list of groups for which this particular router should advertise itself as the candidate rendezvous point; the reverse path table for active multicast groups; and component table with an entry per domain that the router is connected to.
CBT MIB:	configuration of the router including interface configuration; router statistics for multicast groups; state about the set of group cores, either generated by automatic bootstrapping or by static mappings; and configuration information for border routers.
DVMRP MIB	interface configuration and statistics; peer router configuration states and statistics; the state of the DVMRP (Distance-Vector Multicat Routing Protocl) routing table; and information about key management for DVMRP routes.
Tunnel MIB	lists tunnels that might be supported by a router or host. The table supports tunnel types including Generic Routing Encapsulation (GRE) tunnels, IP-in-IP tunnels, minimal encapsulation tunnels, layer two tunnels (LTTP), and point-to-point tunnels (PPTP).
IGMP MIB	only deals with determining if packets should be forwarded over a particular leaf router interface; contains information about the set of router interfaces that are listening for IGMP messages, and a table with information about which interfaces currently have members listening to particular multicast groups.

SNMP tools for working with multicast MIBs

Merit SNMP-Based Management Project has release two freeware tools which work with multicast MIBs:

Mstat	queries a router or SNMP-capable mrouted to generate various tables of information including routing tables, interface configurations, cache contents, etc.
Mview	"application for visualizing and managing the MBone",allows user to display and interact with the topology, collect and monitor performance statistics on routers and links

HP Laboratories researchers investigating IP multicast network management are building a prototype integrated with HP OpenView -- intended for use by the network operators who are not experts in IP multicast; provides discovery, monitoring and fault detection capabilities.

Further reading

IETF *Routing Area,* especially:

- Inter-Domain Multicast Routing (*idmr*)
- Multicast Extensions to OSPF (*mospf*)

IETF *Transport Area* especially:

- Differentiated Services (*diffserv*)
- RSVP Admission Policy (*rap*)
- Multicast-Address Allocation (*malloc*)

With lots of traditional broadcasters and others discovering multicast -- it is going to be an exciting area for the next few years.

Summary

This lecture we have discussed:

- UDP
- BOOTP
- DHCP
- DNS, DDNS
- Multicast, IGMP, RSVP