Multicast

Overview

- applications
- models
- host APIs
- LAN (IGMP, LAN switches)
- intra-domain routing
- inter-domain routing
- address allocation
- the MBONE

Additional references (some are dated!):

- Stephen A. Thomas, *IPng and the TCP/IP protocols*, Wiley, 1996.

- Christian Huitema, *Routing in the Internet*, Prentice Hall, 1995.
- Crowcroft/Handley/Wakeman, *Internetworking Multimedia*, 2000.

Partially drawn from http://www-scf.usc.edu/~dbyrne/960223.txt (D. Estrin)

Broadcast and multicast

broadcast: all hosts on (small, local) network

directed broadcast: all hosts on remote network

multicast: multiple recipients (group)

Applications for Multicast

- audio-video distribution (1-to-many) and symmetric (all-to-all)
- distributed simulation (war gaming, multi-player Doom, ...)
- resource discovery (where's the next time server?)
- file distribution (stock market quotes, new software, ...)
- network news (Usenet)

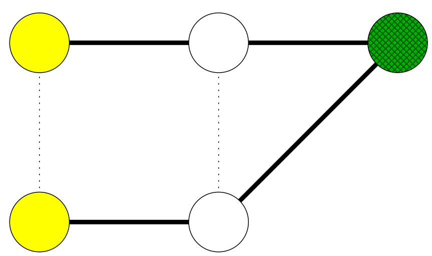
Multicast trees

spanning tree ≡ tree that connects all the vertices (hosts/routers)

shared tree: single tree for *all* sources S

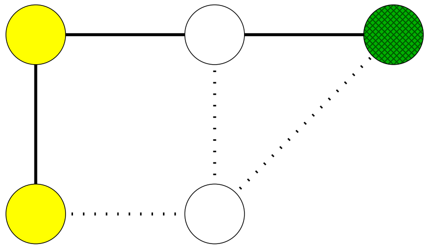
- minimum-cost spanning (MST) tree (where cost = hops, delay, \$, . . .)
- does not minimize length of S to individual destination
- all traffic concentrated on tree ➠ reservation failures

per-source tree: build independently for each source ➠ many variations!

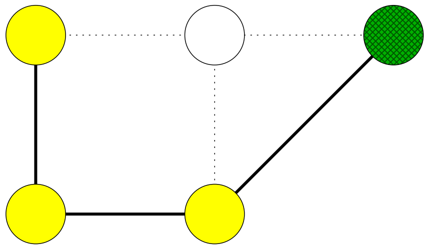


Steiner Tree

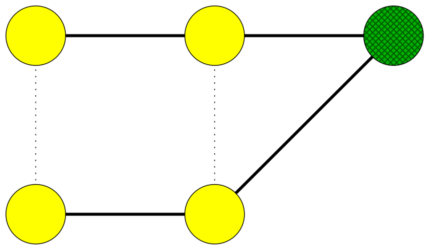
Minimizes the total number of links for all sinks



N-P complete (travelling salesman), unstable: small additions → large changes in traffic flows



Add one node:



Finding MST via Prim’s Algorithm

- centralized, finds MST for $G = (V, E)$
- U: set of vertices connected, start with one
- add lowest-cost edge (u, v) with $u \in U$ and v in $V - U$.
- $T \leftarrow T \cup (u, v)$
- $U \leftarrow U \cup v$

Connection-oriented multicast

- enumerate sources explicitly ➠ source-based trees
- examples:
  - ATM ➠ explicitly add each end point
  - ST-II ➠ enumerate end points in setup message
  - ATM, ST-II: end nodes attach themselves to tree
  - enumeration of end points in packet
- only connection-oriented (packet header size!)
- source needs to know destinations ↔ resource discovery, dynamic groups difficult
- but: natural transition from unicast to multicast

ST-II

- IEN 199: ST ➠ ST-II: RFC 1190 (1990) ➠ ST-II+: RFC 1819 (1995)
- hard state
- combines building tree with resource reservation
- first Internet resource allocation protocol
- sender-initiated tree ➠ receiver-initiated joins ST2+

Host group model

Deering, 1991:

- senders need not be members;
- groups may have any number of members;
- there are no topological restrictions on group membership;
- membership is dynamic and autonomous;
- host groups may be transient or permanent.

Local multicast

Some local networks are by nature multi/broadcast: Ethernet, Token Ring, FDDI, . . .

Ethernet, Tokenring:

- broadcast: all ones
- multicast: 01.xx.xx.xx.xx.xx
- adapter hardware can filter dynamic list of addresses

ATM: point-to-point links ➠ need ATM multicast server

IP multicast

- host-group model
- network-level; data packets same, only address changes
- need help of routers
- special IP addresses (class D): 224.0.0.0 through 239.255.255.255
- 28 bits ➠ 268 million groups (plus scope)
- 224.0.0.x: local network only ➠ 224.0.0.1: all hosts; 224.0.0.2: all routers
- some pre-assigned (224.0.1.2: SGI Dogfight)
- others dynamic (224.2.x.x for multimedia conferencing)
- map into Ethernet: 01.00.5E.00.00.00 + lower 23 bits
- ttl value limits distribution: 0=host, 1=network

Administrative Scoping

- address-based
- 239.255/16: IPv4 local scope
- 239.192/14: organization local scope
- relative addresses (from top) for common applications within scope

Multicast programming

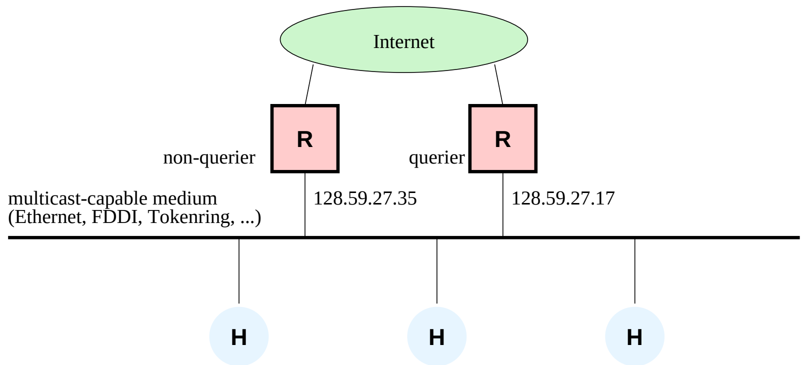
UDP, not TCP (obviously. . .)

```
struct sockaddr_in name;
struct ip_mreq imr;

sock = socket(AF_INET, SOCK_DGRAM, 0);
imr.imr_multiaddr.s_addr = htonl(groupaddr);
imr.imr_interface.s_addr = htonl(INADDR_ANY);
setsockopt(sock, IPPROTO_IP, IP_ADD_MEMBERSHIP,
  &imr, sizeof(struct ip_mreq));
name.sin_addr.s_addr = htonl(groupaddr);
name.sin_port = htons(groupport);
bind(sock, &name, sizeof(name));
recv(sock, (char *)buf, sizeof(buf), 0);
```

IGMP

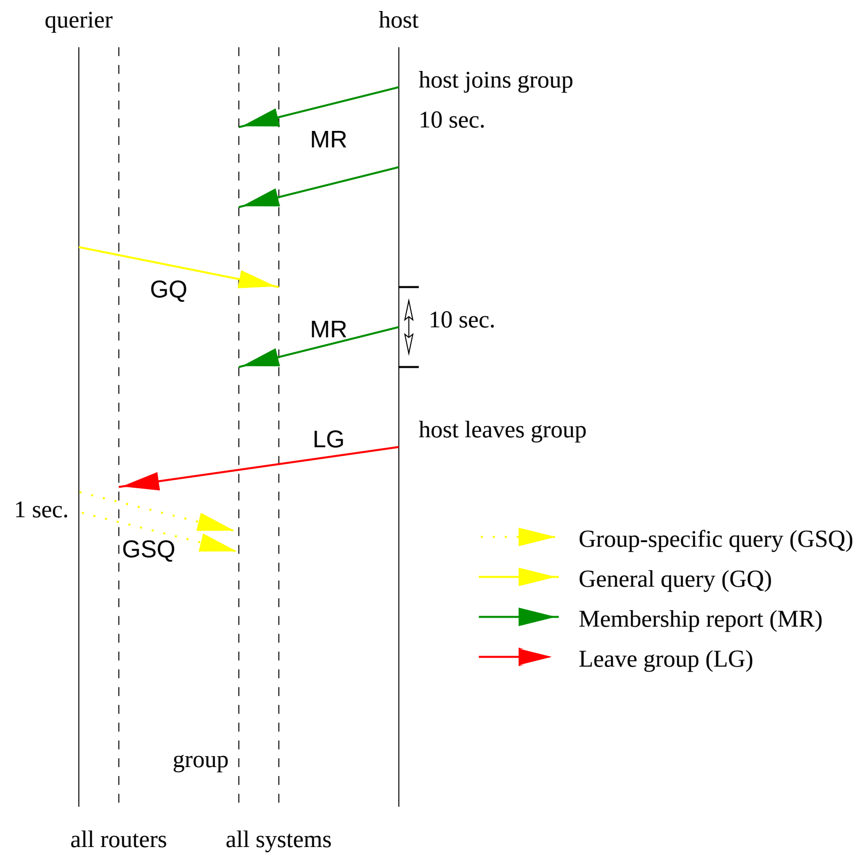
Multicast for local (*broadcast*) networks, between router and hosts



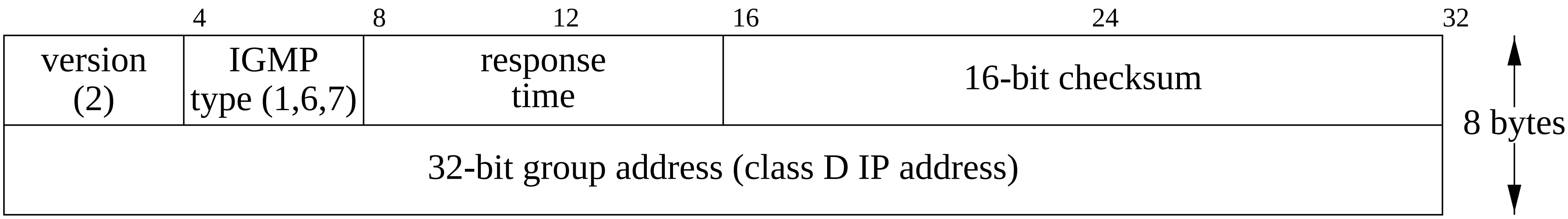
- router listens to all multicast packets on all interfaces
- hosts sends IGMP report for first process to join group to that multicast group (ttl=1), maybe repeat
- router multicasts query to all hosts (224.0.0.2) ≈ every 125 seconds or on start-up
- host waits and listens for others; if nobody else, send response for groups it's in

- if “responsible” for group, notify “all router” group ➠ querier sends group-specific query ➠ reduce bandwidth consumption
- random interval determined by router (< 10 seconds)
- really appropriate for today’s switched Ethernet?

IGMPv2 timing



IGMPv2 packet




```
$ netstat -g
Group Memberships
Interface Group                 RefCnt
--------- --------------------- ------
lo0       ALL-SYSTEMS.MCAST.NET      1
le0       224.2.127.255              1
le0       ALL-SYSTEMS.MCAST.NET      1
```

IGMPv3

- adds source filtering to IGMPv2
- Membership Report includes lists of sources to include or exclude
- Group-and-Source-Specific Query asks whether anybody cares about the group and the sources listed
- unlike IGMPv2, host no longer suppresses membership reports if it hears from another host
  - accounting
  - avoid Ethernet switches having to remove “outbound” IGMP reports to fool hosts
  - for efficiency, single membership report can list multiple groups

Note: IPv6 defines new protocol, Multicast Listener Discovery (MLD)

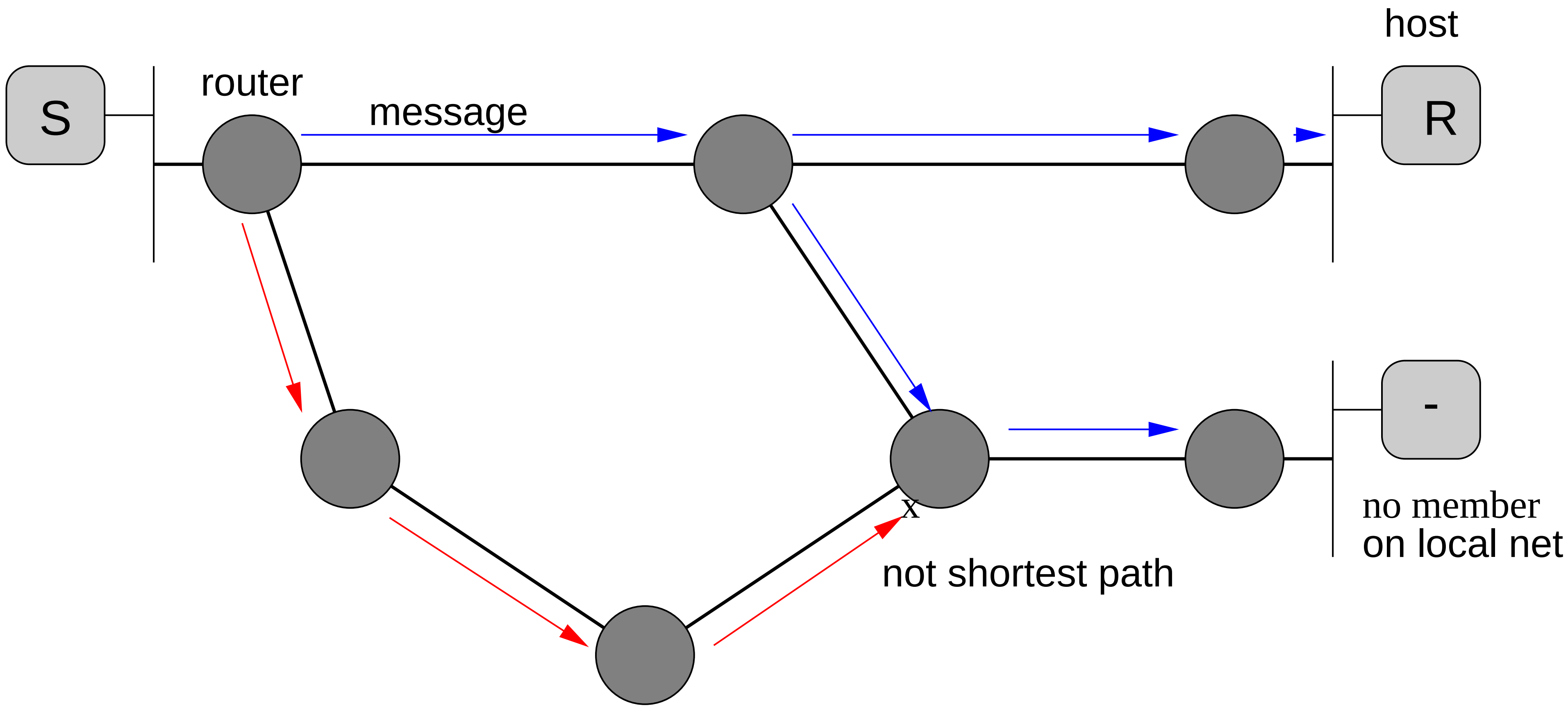
Reverse path flooding

iif: incoming interface; oif: outgoing interface

- if iif is on shortest path to source S
- forward to all other oifs (*RPF check*) towards receivers R in group G
- avoids forwarding duplicates

Multicast forwarding

First packet (truncated broadcast)



Reverse path broadcasting

- do RPF check as before
- exchange unicast routing info to establish “parentage”
- restrict oifs to child nodes

➠ reduce duplicates

Multicast routing

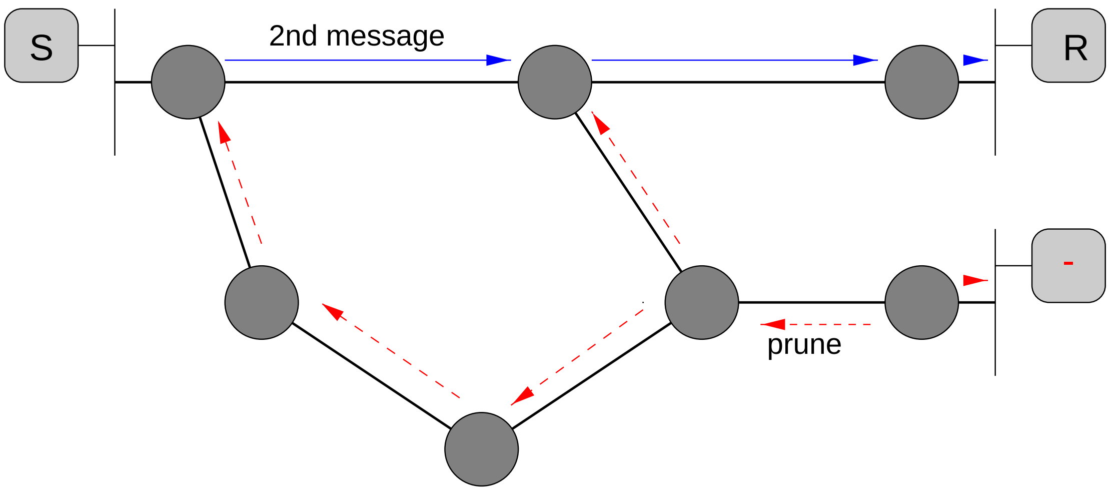
- link-state based
- dense mode
- sparse mode

Multicast forwarding with truncation

- flood with RPF check
- pruning: leaves of tree send “prune” if no members below
- receivers tell routers of membership
- routers know whether to forward to LAN or prune
- prune state must time out ➠ periodic broadcast
- trade-off: join latency $\leftrightarrow$ bandwidth
- add: explicit “graft” to cancel prune: ➠ join latency $\downarrow$
- still need occasional broadcast for topology changes

Multicast forwarding

With pruning:



➠ router needs to keep "negative" list for groups

Distance Vector Multicast Routing Protocol (DVMRP)

- flood + RPF check
- pruning: time out 1 minute
- routers may send *grafts* upstream
- only send to children
- maintain routing information (DV)
- used in → MBone

Multicast Open Shortest Path First (MOSPF)

- link-state based
- include membership info in link-state advertisements
- compute tree for each S, G pair ➠ oifs
- can create shortest-path trees even with asymmetric links
- cannot afford to recompute trees with each LS change

PIM-DM

- use unicast routing table
- DVMRP: include only oif that use this router to reach source
- PIM-DM: forward to all outgoing interfaces

Problems

- "multicast storms"
- MOSPF: broadcast of membership to off-tree areas
- DVMRP: occasional broadcast of packets ➠ bad for WANs
- prune state in routers for sparse groups
- multicast routing vs. unicast routing: reverse path with asymmetric links
- hierarchical routing?
- few "big" senders, lots of background mumbling

➠ compromise on optimal trees

Protocol Independent Multicast (PIM-SM)

- uses unicast routing
- supports SPTs and shared trees (rooted at "rendezvous point" RP), depending on traffic
  1. group-specific RP-rooted shared tree
  2. source-based tree

PIM-SM: RP election

- RP selected by hash of G
- bootstrap router (BSR) candidate sends list of candidate RPs
- candidate BSRs, configured with priority
- multicast candidacy locally (ttl = 1), then flood
- elected routers periodically sends bootstrap message with RPs
- $\{$candidate BSR$\} \approx \{$candidate RP$\}$
- candidate-RP sends message to BSR

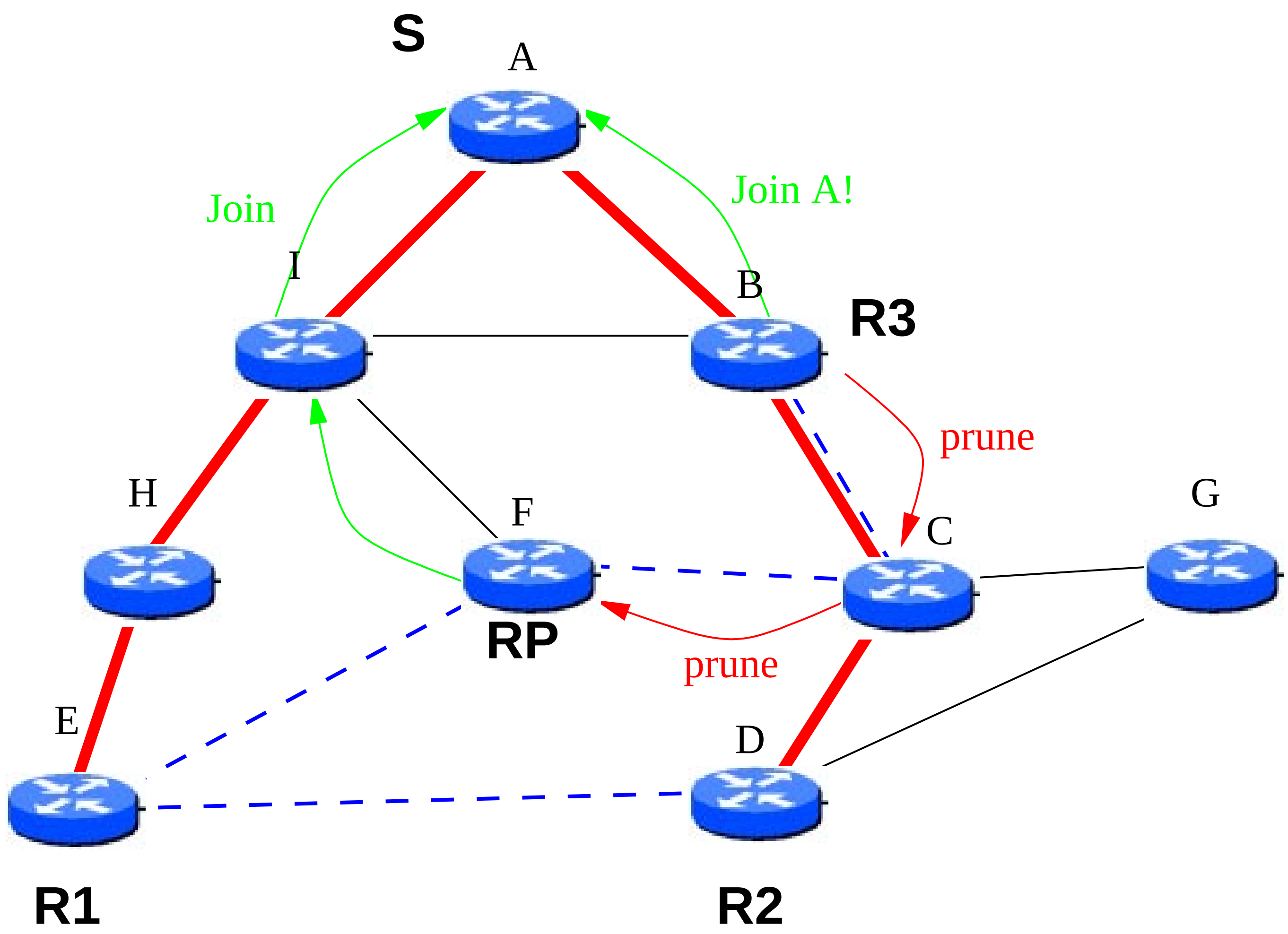
PIM-SM: shared tree

- send packet via unicast in “register” message, encapsulated, to RP
- RP forwards message down shared tree
- receivers send “join” to RP to join shared tree
- joins stop when reaching tree, install $(*, G)$ state

PIM-SM: source-specific tree

1. bypass encapsulation
   - RP sends “join” towards S
   - nodes recognize destination and forward based on G
2. receivers join
3. and prune shared tree for S

PIM-SM



Sparse Mode Problems

- single point of failure
- hot spot
- non-optimal path
- complexity

Interdomain sparse multicast routing: CBT

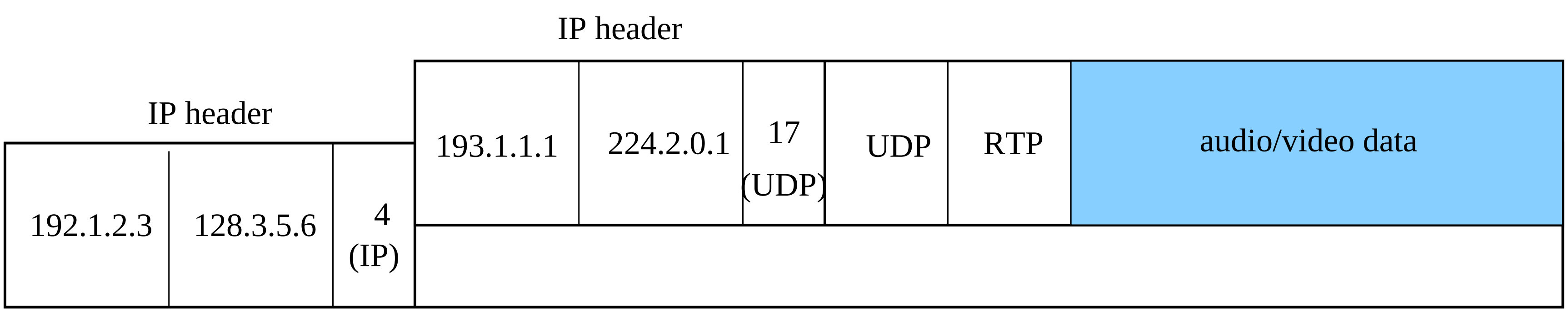
- core-based trees: bidirectional center-based shared trees routed at *core*
- receivers send join messages to core
- senders send data to core, but can be short-cut $\longrightarrow$ send to all interfaces participating in group
- no SPTs
- *hard-state* with acknowledged join from core or first on-tree router

  $+$: no source specific state

  $-$: path lengths, traffic concentration
- explicit joining (vs. implicit join and explicit prune)
  join messages from R's router to root of tree
- not much implementation

MBONE

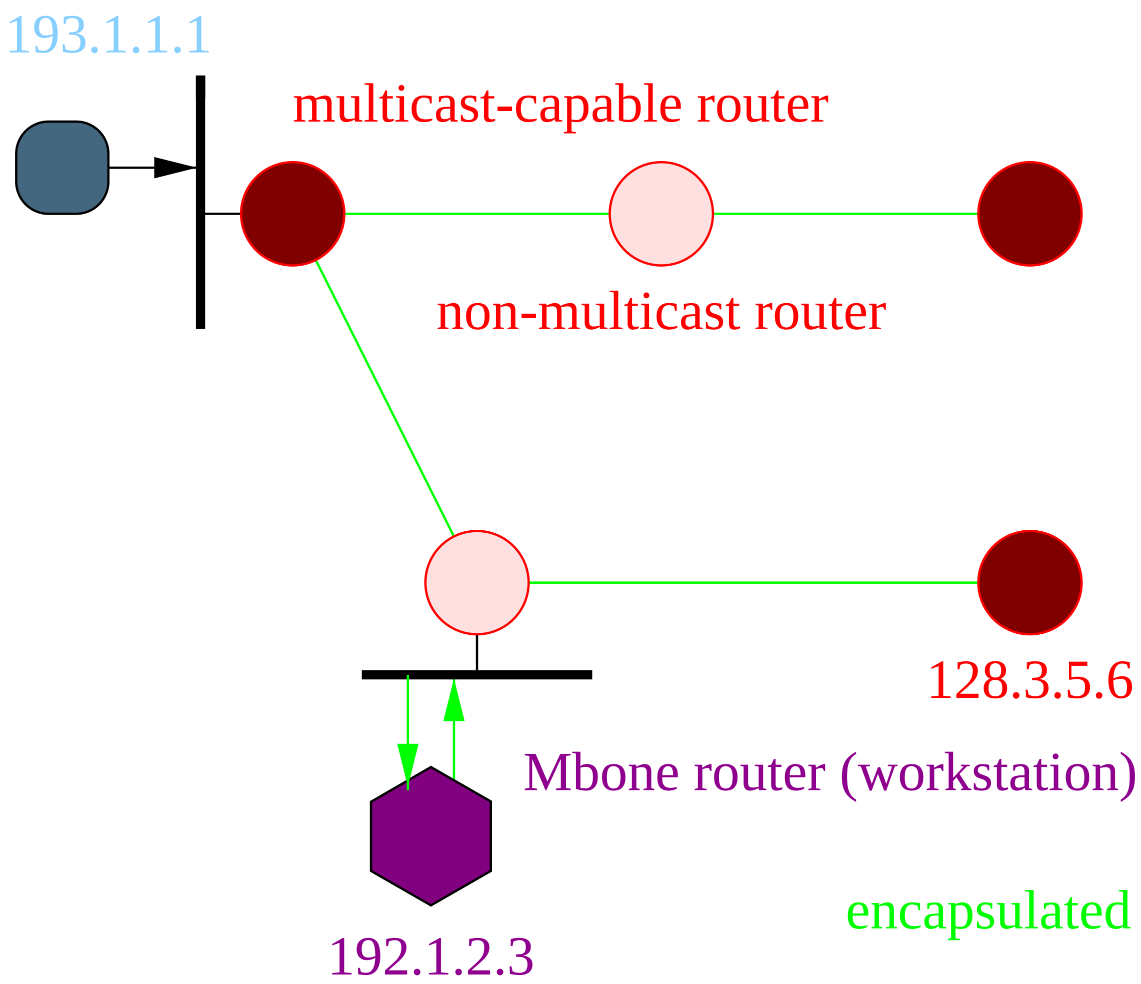
- MBONE ≡ multicast backbone
- overlay network over Internet, up to 10,000 routes
- difficulty of limiting fan-out
- needed until deployment of multicast-capable backbone routers
- IP-in-IP encapsulation ➠ *tunneling*:



source: 193.1.1.1; group: 224.2.0.1; MBONE tunnel: 192.1.2.3 to 128.3.5.6

- limited capacity, resilience

Mbone


193.1.1.1
multicast-capable router
non-multicast router
128.3.5.6
Mbone router (workstation)
encapsulated
192.1.2.3

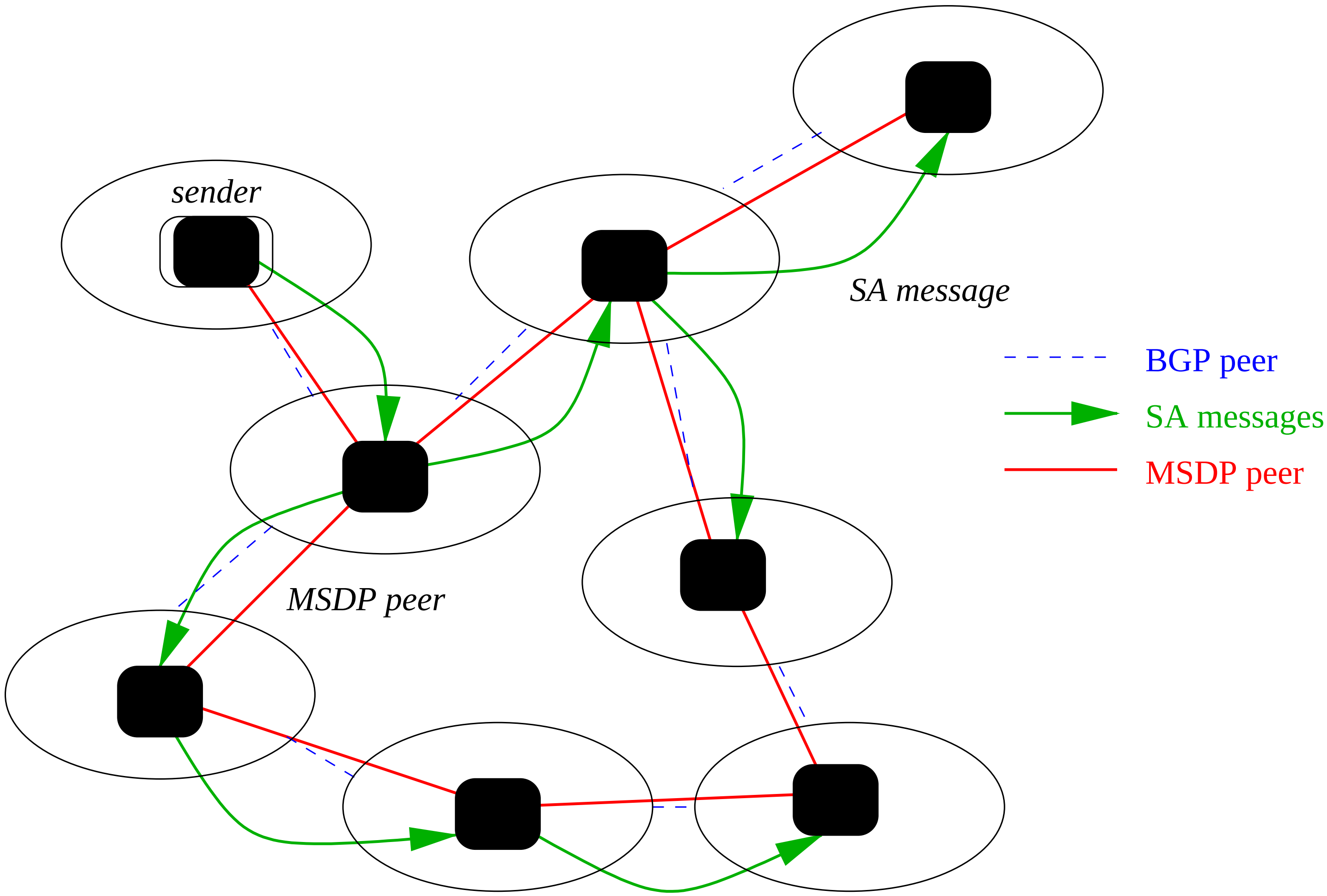
Inter-domain multicast

- one RP per AS
- Multicast Source Discovery Protocol (MSDP)
- avoid third-party dependencies
- designated party (RP) announces membership to others
- flood information to other ASs

Multicast Source Discovery Protocol (MSDP)

- join together PIM-SM regions ("AS")
- discover sources in other regions (to send them PIM "join" requests)
- peering with fellow RPs
- send "source active" to peer RPs
- flood "source active" message in BGP style
- works reasonably well only when few senders

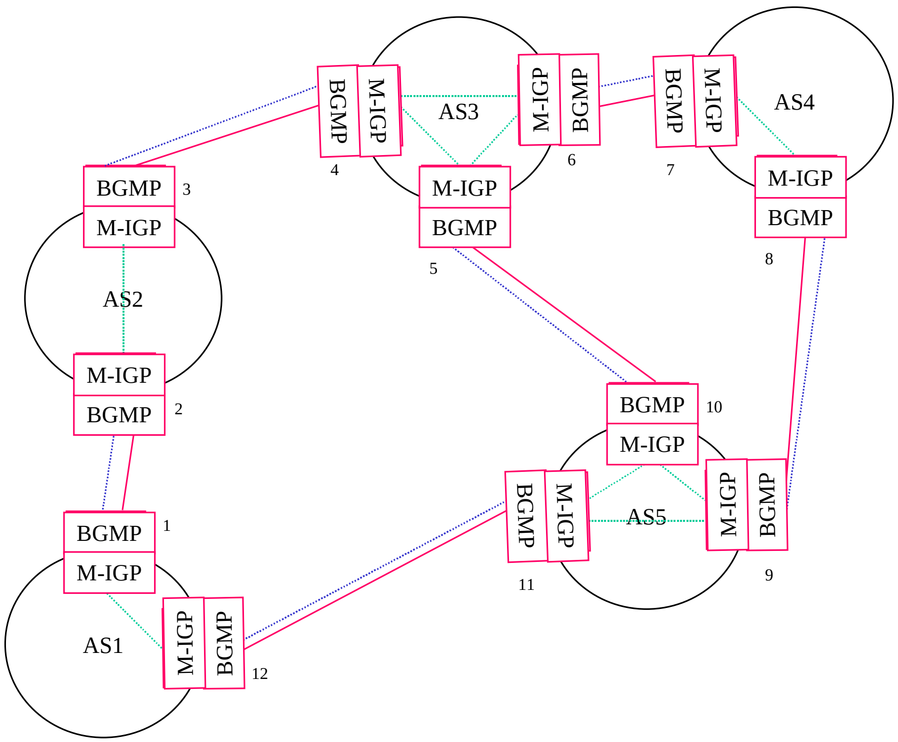
MSDP Operation



Border Gateway Multicast Routing Protocol (BGMP)

- *bidirectional* shared tree for each group
- TCP connections between routers (external BGP peers)
- root domain
- distribute "routes" to AS hosting core
- packets can bypass BGMP core
- packet forwarding similar to PIM-SM RP

BGMP



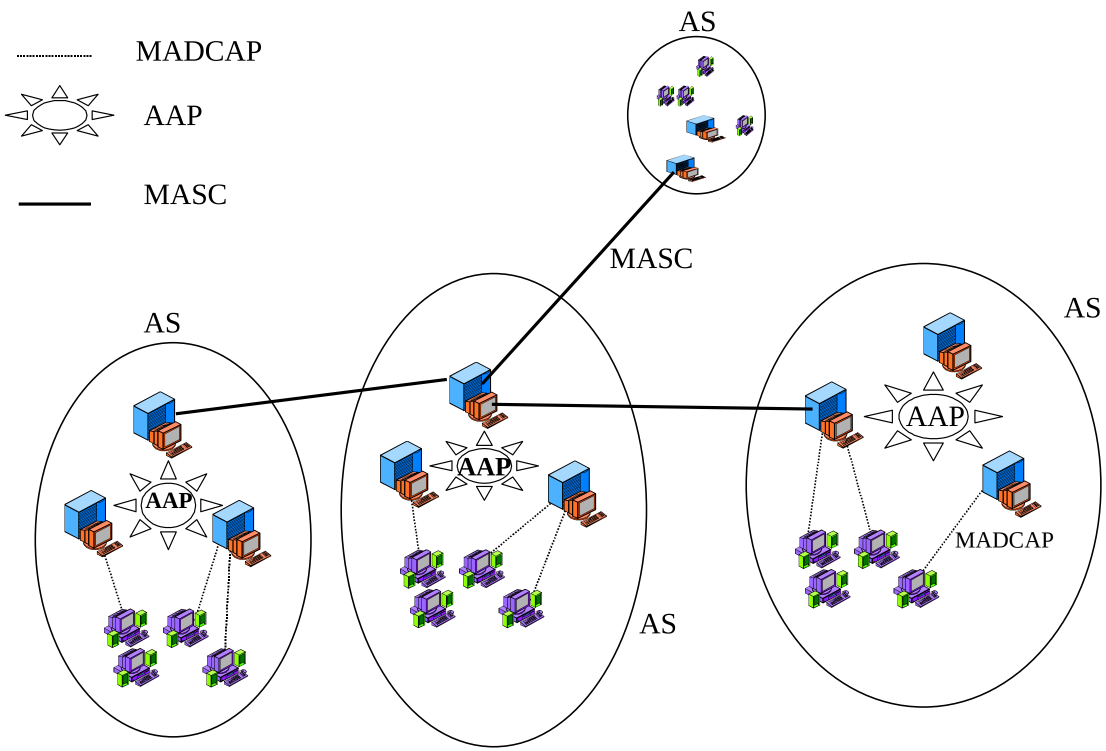
Multicast address allocation

hierarchical, with different time scales:

1. clients contact MAAS server in domain via MADCAP
2. MADCAP gets it via Multicast Address Allocation Protocol (AAP)
   - MASC routers multicast availability to the MAAS
   - multicast claims
3. MASC for inter-AS for large blocks

233/8 for per-AS static allocation

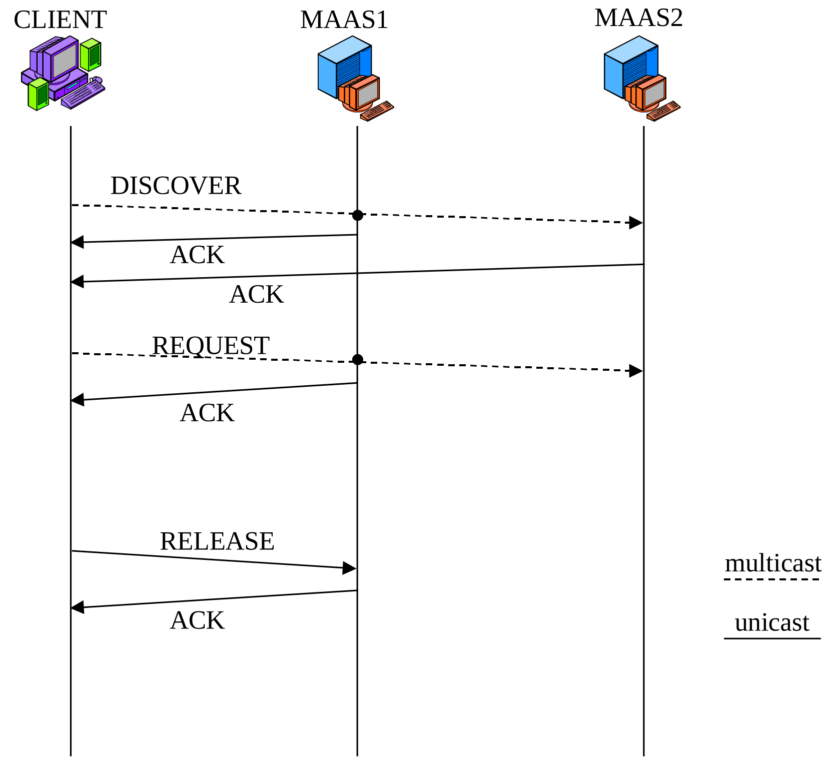
Multicast Address Allocation



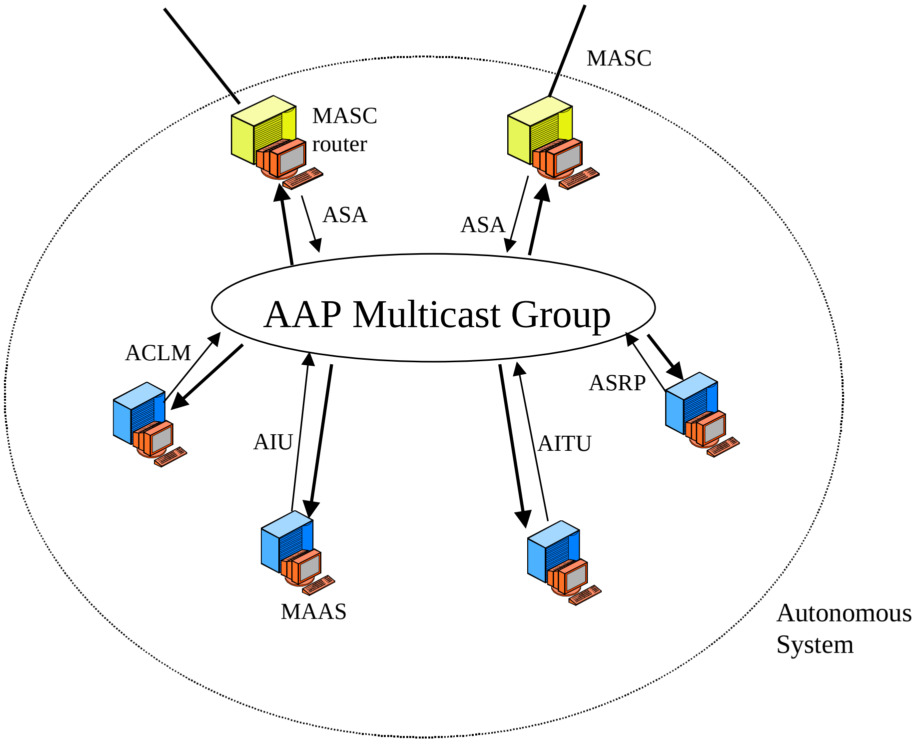
MADCAP

- UDP-based request-response (similar to DHCP)
- one or more local servers
- may request addresses in the future
- specify maximum delay
- can request specific address
- discover scopes via INFORM
- multicast request via DISCOVER
- server hands out, client confirms via REQUEST
- expires or via RELEASE

MADCAP


CLIENT
MAAS1
MAAS2
DISCOVER
ACK
ACK
REQUEST
ACK
RELEASE
ACK
multicast
unicast

AAP: Multicast Addresses within AS



AAP

- send ACLM to claim addresses
- object to claims and announce own via AIU
- MAAS can preallocate addresses (ACLM) or "Adress Intent to Use" (AITU), with reclaiming by others via ACLM
- report periodically on address space use

MASC

- top of hiearchy: inter-domain
- BGP model: TCP peering relationships
- also allows customer-provider relationships
- send time-limited claim for range, wait a few days and then use
- send “prefix managed” to children