IPv6 and Multicast

Outline

• IPv6
• Multicast
IPv6 overview

• Motivation
  – Internet growth (address space depletion and routing information explosion)
  – CIDR has helped but eventually bigger address space is needed
    • ubiquitous networking, “Internet to the toaster!”
• Historical perspective
  – bigger address space ⇒ changes in IP header ⇒ new IP version
  – work initiated in IETF in early 90’s
    • name changed from IPng (next generation) to IPv6
  – “snow ball effect”: why not fix all problems at the same time!
    • added features: QoS, security, autoconfiguration, mobility, ...
    • note! most of these have in the mean time been introduced to IPv4
  – requirement: transition plan (from IPv4 to IPv6)
    • impossible to require an over night change in IP version in all routers
    • routers running only IPv4, IPv4 and IPv6, IPv6 will coexist for a long time
  – by now, most key specifications of IPv6 are Proposed or Draft Standards

IPv6 addressing

• IPv6 uses 128 bit addresses
  – enables up to $3.4 \times 10^{38}$ nodes!
  – address notation: $x:x:x:x:x:x:x$ ($x =$ hex representation of a 16 bit number)

• IPv6 address space
  – IPv6 does not use classes
  – address space subdivided based on leading bits
  – leading bits indicate different uses of the address space
IPv6 address prefixes

- **Aggregatable Unicast Address (001):**
  - most important address group, like classless IPv4 addresses but longer
  - more on these later…

- **NSAP (0000 001) and IPX (0000 010) addresses**
  - NSAP addresses used by ISO protocols; IPX for Novell networks

- **Link Local Address (1111 1110 10), Site Local Address (1111 1110 11):**
  - enables host to construct an address to be used locally on a network (or site) without having to be concerned with global uniqueness (autoconfig.)

- **Multicast Address (1111 1111)**

- **Reserved Address (0000 0000):**
  - “IPv4 compatible IPv6” and “IPv4-mapped IPv6” addresses needed during IPv4 to IPv6 transition

---

Aggregatable Global Unicast Addresses (1)

- **Aggregatable Global Unicast Addresses**
  - normal unicast addresses in IPv6
  - problem: how to assign unicast addresses effectively to ASs, networks, hosts, routers?
    - issues: new nodes added at an increasing rate, routing scalability

- **Address allocation plan**
  - Internet not just an arbitrarily connected set of ASs:
    - **subscribers** (e.g. non transit ASs) connect to **providers** (transit ASs)
    - **providers** can be **direct** (connect primarily subscribers) or **indirect** (connect other providers, backbone networks)
  - problem: how to use this hierarchy without imposing restrictive limitations?
    - subscribers may be connected to several providers
  - idea: allocate addresses to enable route information aggregation (scalability)
    - by using variable length prefixes (same as in CIDR)
    - direct provider allocated a prefix and that provider can then assign longer prefixes to its subscribers (provider based addressing)
    - thus, provider needs to advertise only one prefix to all its subscribers
  - drawback: if site changes provider, whole numbering in a site must be changed
Aggregatable Global Unicast Addresses (2)

• Is hierarchical aggregation always useful?
  – aggregation at national or continental level:
    • continental boundaries form natural aggregation points for example, all addresses in Europe have the same prefix
  – given that a provider connects to many backbones, not meaningful for providers to get their prefix from one backbone provider
  – subscriber connects to several providers:
    • if subscriber takes prefix from provider X, provider Y must advertise provider X’s networks (can not be aggregated with Y’s own prefixes)
    • if subscriber numbers its network using prefixes from X and Y and if connection to X goes down, hosts with prefix from X become unreachable
    • possible solution: provider X and Y share common prefix for all subscribers having connections with X and Y

IPv4 to IPv6 transition

• IPv6 is deployed incrementally
  – IPv4 and IPv6 routers need to coexist

• Dual stack operation and tunneling
  – dual stack
    • IPv6 node runs both IPv4 and IPv6 (Version field identifies packets)
    • node may have separate IPv4 and IPv6 addresses or an “IPv4 mapped IPv6 address”
  – tunneling
    • used to send IPv6 packets over IPv4 network
    • IPv6 packet encapsulated inside IPv4 header
    • tunneling automatic if end point has “IPv4 mapped IPv6 address”
    • otherwise, tunnel configured manually
**IPv6 packet format (1)**

- IPv6 header format simpler than IPv4
  - goal was to have simplified header processing
  - constant length 40 bytes
- Version = 6
- TrafficClass & FlowLabel related to QoS
  - lecture 10
- PayLoadLen = length of packet in bytes (without header)
- HopLimit = TTL of IPv4
- NextHeader
  - replaces Options and Protocol field of IPv4
  - if options are required they are carried in one or several extension headers following the IP header
  - if no extension headers, NextHeader identifies higher layer protocol (TCP, UDP)
  - type of extension header identified by the NextHeader field in the header preceding it

```
Version  TrafficClass  FlowLabel
  PayLoadLen  NextHeader  HopLimit
    SourceAddress
          DestinationAddress

Next header/data
```

**IPv6 packet format (2)**

- Improved Options handling in IPv6
  - IPv4: if any Options are present every router must parse the whole Options field to see if any Options are relevant; Options form an unsorted list
  - IPv6: options treated as extension headers appearing in a specific order \( \Rightarrow \) router can quickly determine which options are relevant by looking at the NextHeader field
  - no upper limit on nof options

- Some extension headers
  - fragmentation header
  - authentication header
  - routing header
    - enables source-directed routing: sender can specify nodes or topological areas that the packet should visit en route to destination
    - used also for supporting multicast and mobility
Autoconfiguration

- Traditionally, host configuration required considerable system administration expertise (IP address, subnet mask, name server)
- IPv6 provides “plug-and-play” functionality
  - DHCP can be used in IPv4
  - longer address format enables stateless autoconfiguration that does not require the use of any dedicated server
- Stateless autoconfiguration
  - each host has globally unique 48 bit LAN address (link level address)
    - LAN address used as the least significant bits for IPv6 address
  - not globally unique IPv6 address: IPv6 address prefix Link Local Address (1111 1110 10) + “70 zeros” + LAN address
    - adequate for local devices, e.g., printers, local servers
  - globally unique IPv6 address: router advertises appropriate global prefix and host uses as its address (prefix + “enough zeros” + LAN address)
    - possible because address 128 bits long!

Outline

- IPv6
- Multicast
Multicast overview (1)

• Basic problem: host wants to send same data to multiple receivers
  – on a LAN this is handled in hardware
  – In Internet, multicasting must serve hosts residing on different networks and separated by large distances

• IP multicast service model:
  – hosts wishing to receive a particular multicast transmission belong to a multicast group
    • any given host may belong to many groups simultaneously
  – a multicast group is associated with an IP multicast address
  – packet delivery: host sends one copy of a packet to a multicast address and Internet delivers it to all members of a group

Multicast overview (2)

• Here we look at how packets get distributed to the correct routers

• Group management or multicast address advertising is not considered in detail
  – hosts join/leave groups dynamically by informing their local routers using IGMP (Internet Group Management Protocol)
  – knowledge of available multicast groups handled by out-of-band means (tools exist for advertising multicast addresses in the Internet)

• Multicast packet delivery implemented by extending forwarding and routing functionality of IP routers
  – three approaches:
    • link-state
    • distance vector
    • protocol independent
Link state multicast (1)

- **Link state routing:**
  - routers flood info related to directly connected links
  - ⇒ nodes have full topology information
  - ⇒ can construct shortest paths to any given node (Dijkstra)

- **Generalization to multicast:**
  - add set of groups with members on particular network to link state info
  - hosts on a given link (LAN) announce their participation to router
  - link state flooded to all other routers and each router can compute shortest path multicast trees for all sources in all groups
  - possible because in link state nodes learn full topology
  - trigger flooding when groups appear/disappear on a link

Link state multicast (2)

- **Shortest path multicast tree:**
  - tree rooted at source that minimizes the sum of the costs of the routes between source and all destinations belonging to a multicast group (solved by Dijkstra’s algorithm)
  - Example: nodes marked with “x” belong to group G, picture shows multicast trees for nodes A, B and C

- **Problems:**
  - excessive amounts of route info: each router must keep separate shortest path multicast trees from every source to every multicast group
    - in practice, trees computed only for active source/group pairs
  - potential instability if group membership changes frequently
Distance vector multicast

- Distance vector routing:
  - neighbors exchange forwarding tables
  - that is, routers do not know full topology

- Generalization to multicast done in two steps:
  1. mechanism to broadcast packets to all networks
     - Reverse Path Broadcast, RPB
  2. pruning of networks that do not have hosts belonging to the multicast group
     - Reverse Path Multicast, RPM

- Real life example: MBone
  - overlay network on top of Internet
    - packets tunneled through Internet between MBone nodes
  - uses Distance Vector Multicast Routing Protocol
  - popular application: vic (multiparty videoconferencing)
    - IETF meetings have been broadcasted over MBone

Reverse Path Broadcast (RPB)

- Achieving flooding:
  - router forwards multicast packets of source S to all its links except on the link from where the packets were received
  - achieves flooding and packets do not loop back to S
  - creates excess traffic (router does not know if there are any group member’s to receive the packets, treated in next slide)
  - if LAN network connected to Internet via several routers, same multicast packets will be sent onto the LAN by all connected routers

- Idea: eliminate duplicate broadcasts on LANs
  - designate one router as “parent” router relative to source S
    - routers connected to same LAN hear each other’s distance vectors
    - router with shortest distance to S selected as parent (address used to break ties)
  - only parent router allowed to forward traffic from S to LAN
  - each router must maintain state for each source S/link (interface) pair if it is parent or not
Reverse Path Multicast (RPM)

- From broadcast to multicast
  - First, need to recognize if a “leaf” network has any (multicast) group members
    - network is a leaf if no other router uses it to reach source S
    - hosts on leaf network periodically announce their group memberships
      ⇒ router knows if any group members are present
  - Second, propagation of “no members of G here” information
    - distance vector info extended to include info on the set of groups from which the leaf network wants to receive multicast traffic
    - routers can decide for its links for which groups it should forward the traffic
- Problem:
  - potentially a lot of routing state info for each router
  - in practice, routers use RPB until some node becomes active, and then those nodes not interested in this particular multicast traffic speak up

Protocol Independent Multicast (PIM)

- RPB and RPM do not scale
  - routers build multicast shortest path trees for all sources in all groups
  - in particular, amount of state info too large if only small proportion of routers want to receive traffic for a certain group (=“sparse” group)
- Solution: PIM
  - use single tree for sparse groups (shared tree, saves state info in routers)
  - use shortest path trees for dense groups (lot of traffic between many nodes)
  - supports different kinds of trees
  - trees can be mixed within same group depending on traffic
- PIM-SM (PIM Sparse Mode)
  - routers send Join and Prune messages to routers that have been assigned as Rendezvous Points (RP)
  - shared and source specific trees
    - initially a shared tree rooted at RP is created and source specific trees created only if traffic warrants it
PIM operation

1. R4 and R5 join the shared tree, R2 marks interfaces with (*,G) (all traffic from group G forwarded on this interface).
2. R1 tries to send packet to G: sends packet to R1 (designated router), R1 not part of shared tree so it tunnels packet to RP, RP sends packet via shared tree.
3. RP can force R3 to know about multicast tree (removes need for encapsulation); R3 creates sender specific state ((S,G) state).
4. If data rate from R1 high enough, R4 and R5 can Join the sender specific tree (to avoid looping via RP).