Spanning Tree

How to handle forwarding in complex Layer 2 topologies?
Each LAN segment can have many bridges

- More complex topologies can provide redundancy.
  - But can also create loops.
    - E.g. What happens when there is no table entry?
  - Multiple copies of data
    → Could crash the network → has happened often!
What is a Spanning Tree?

- Reduce our topology graph to a tree:
  - Make sure there are no loops in the topology
  - All LAN segments are still connected to the LAN and can receive messages

- Main idea: Bridges choose the ports over which they have to forward frames.
Spanning Tree Protocol

Overview

Embed a tree that provides a single unique default path to each destination:

Bridges designate ports over which they will or will not forward frames

By removing ports, extended LAN is reduced to a tree

Addresses the crashing problem; but tree is not resilient

When switch/link fails, rerun protocol to converge to new tree
Spanning Tree Algorithm

- Root of the spanning tree is elected first \( \rightarrow \) the bridge with the lowest identifier.
  - All ports are part of tree

- Bridges designate ports on which they will/ not forward

- Each bridge finds shortest path to the root.
  - Remembers port that is on the shortest path
  - Used to forward packets

- Select for each LAN a designated bridge that will forward frames to root
  - Has the shortest path to the root.
  - Identifier as tie-breaker
Spanning Tree Algorithm

• Each node sends configuration message to all neighbors.
  – Identifier of the sender
  – Id of the presumed root
  – Distance to the presumed root

• Initially each bridge thinks it is the root.
  – B5 sends (B5, B5, 0)

• When B receive a message, it decides whether the solution is better than their local solution.
  – A root with a lower identifier?
  – Same root but lower distance?
  – Same root, distance but sender has lower identifier?

• Message from bridge with smaller root ID
  – Not root; stop generating config messages, but can forward

• Message from bridge closer to root
  – Not designated bridge; stop sending any config messages on the port
  – Block port
Spanning Tree Algorithm

- Each bridge B can now select which of its ports make up the spanning tree:
  - B’s root port
  - All ports for which B is the designated bridge on the LAN

- States for ports on bridges
  - Forward state or blocked state, depending on whether the port is part of the spanning tree

- Root periodically sends configuration messages and bridges forward them over LANs they are responsible for

- Any bridge failure => Start over
Spanning Tree Algorithm Example

- B3 receives (B2, B2, 0)
  - Since 2<3 B3 accepts B2 as a root
  - B3 adds one to the distance advertised by B2(0) and thus sends (B3, B2, 1) toward B5
- Meanwhile B2 accepts B1 as the root and sends (B2, B1, 1)
- B5 accepts B1 as the root and sends (B5, B1, 1)
- B3 accepts B1 as the root and figures that B1 and B2 are closer to the root. So stops forwarding on both interfaces.
Internet Protocol

Outline
- Introduction to Internet Protocol
- Header and address formats
- ICMP
- Tools
Internet Protocol

• Runs on all hosts in the Internet and enables packets to be routed between systems
  • Key protocol for building networks (out of different LAN/ ext LAN technologies or protocols)
  • Kahn-Cerf

• Functions
  1. Datagram delivery of packets
     1. Connectionless and based on routing protocols
  2. Well defined packet format
  3. Global addressing
     1. Means for identifying Internet hosts
  4. Fragmentation and reassembly
     1. Since packets can be of varying size
  5. Error reporting

• 2 and 3 are key to ensuring global connectivity
An IP Internet – Network of Networks

Routers act as translators across different extended LAN technologies.
Protocol Stack – IP is Common to All (also key to global connectivity)
Service Model

- Connectionless (datagram-based)
- Best-effort delivery (unreliable service)
  - packets are lost
  - packets are delivered out of order
  - duplicate copies of a packet are delivered
  - packets can be delayed for a long time
# IPv4 datagram Format

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>19</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HLen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ident</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flags</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Offset</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protocol</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Checksum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SourceAddr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DestinationAddr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Options (variable)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pad (variable)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Header
Fragmentation and Reassembly

- Each network has some Maximum Transmission Unit (MTU)
  - Largest datagram that a network can carry in a frame
- Strategy
  - fragment when necessary (MTU < Datagram)
  - try to avoid fragmentation at source host
    - Due to overhead of reassembly
  - re-fragmentation is possible
  - fragments are self-contained datagrams
  - delay reassembly until destination host
    - Keep this functionality out of the network
  - do not recover from lost fragments
    - End hosts try to reassemble fragmented packets – if a fragment is lost…
- End hosts are encouraged to do MTU discovery
Example
<table>
<thead>
<tr>
<th>Start of header</th>
<th>Ident = x</th>
<th>1</th>
<th>Offset = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rest of header</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1400 data bytes</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Start of header</th>
<th>Ident = x</th>
<th>1</th>
<th>Offset = 512</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rest of header</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>512 data bytes</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Start of header</th>
<th>Ident = x</th>
<th>0</th>
<th>Offset = 1024</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rest of header</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>376 data bytes</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
IPv4 Global Addresses

- **Properties**
  - globally unique
  - hierarchical: network + host

- **Dot Notation**
  - 10.3.2.4
  - 128.96.33.81
  - 192.12.69.77

- **AS’s refer to a network type (assigned address range)**

<table>
<thead>
<tr>
<th>Network</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>A:</td>
<td>0</td>
</tr>
<tr>
<td>B:</td>
<td>1</td>
</tr>
<tr>
<td>C:</td>
<td>1</td>
</tr>
</tbody>
</table>

```
A: 7 24
   0 Network Host
B: 14 16
   1 0 Network Host
C: 21 8
   1 1 0 Network Host
```
Datagram Forwarding

• Every datagram contains destination’s address
• The “network part” of an IP address uniquely identifies a single physical network (AS)
• If directly connected to destination network, then forward to host
• If not directly connected to destination network, then forward to some router
• Forwarding table maps network number into next hop
  • Mapping is based on routing algorithm
• Each host has a default router
  • Each router maintains a forwarding table
Efficient Addressing

Outline
  - Addressing
  - Subnetting
  - Supernetting
IPV4 Global Addresses

• Properties
  • IPv4 uses 32 bit address space
  • globally unique
  • hierarchical: network + host

• Dot Notation
  • 10.3.2.4
  • 128.96.33.81
  • 192.12.69.77
Problems

• Network composed of many geographical dispersed extended LANs – sharing 1 n/w number too inefficient, giving out multiple class C or Class B (too inefficient)
• Networks with just a few more than 255 hosts
  • Class B – wastage
  • Multiple class – routing table expansion
Subnetting - 1985

- Original intent was for network to identify one physical network
  - Lots of small networks are what we actually have – how do we handle this?
- Solution: add another level to address/routing hierarchy: subnet
  - Allocate addresses to several physical networks
  - Routers in other ASs (networks) route all traffic to network as if it is a single physical network
- Subnet masks define variable partition of host part
  - 1’s identify subnet, 0’s identify hosts within the subnet
  - Mechanism for sharing a single network number among multiple networks
- Subnets visible only within a site

<table>
<thead>
<tr>
<th>Network number</th>
<th>Host number</th>
</tr>
</thead>
</table>

Class B address

| 111111111111111111111111111111 | 00000000 |

Subnet mask (255.255.255.0)

<table>
<thead>
<tr>
<th>Network number</th>
<th>Subnet ID</th>
<th>Host ID</th>
</tr>
</thead>
</table>

Subnetted address
Subnet Example

Subnet Example

Subnet mask: 255.255.255.128
Subnet number: 128.96.34.0

128.96.34.15
   H1

128.96.34.129
   R2

128.96.33.1
   H3

Subnet mask: 255.255.255.0
Subnet number: 128.96.33.0

Subnet mask: 255.255.255.128
Subnet number: 128.96.34.128

128.96.34.129
   R2

128.96.34.130

Subnet mask: 255.255.255.128
Subnet number: 128.96.34.0

128.96.34.1

Forwarding table at router R1

<table>
<thead>
<tr>
<th>Subnet Number</th>
<th>Subnet Mask</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.96.34.0</td>
<td>255.255.255.128</td>
<td>interface 0</td>
</tr>
<tr>
<td>128.96.34.128</td>
<td>255.255.255.128</td>
<td>interface 1</td>
</tr>
<tr>
<td>128.96.33.0</td>
<td>255.255.255.0</td>
<td>R2</td>
</tr>
</tbody>
</table>
Forwarding Algorithm

D = destination IP address
for each entry (SubnetNum, SubnetMask, NextHop)
   D1 = SubnetMask & D
   if D1 = SubnetNum
      if NextHop is an interface
         deliver datagram directly to D
      else
         deliver datagram to NextHop
   else
      use a default router if nothing matches
   not necessary for all 1s in subnet mask to be contiguous
   can put multiple subnets on one physical network
   subnets not visible from the rest of the Internet
Continued Problems with IPv4 Addresses

• Problem:
  • Potential exhaustion of IPv4 address space (due to inefficiency)
    – Class B network numbers are highly prized
      • Not everyone needs one
    – Lots of class C addresses but no one wants them
  • Growth of back bone routing tables
    – We don’t want lots of small networks since this causes large routing tables
    – Route calculation and management requires high computational overhead

• Solution:
  • Allow addresses assigned to a single entity to span multiple classed prefixes
  • Enhance route aggregation
Supernetting

- Assign block of contiguous network numbers to nearby networks
- Called CIDR: Classless Inter-Domain Routing
  - Breaks rigid boundaries between address classes
  - If ISP needs 16 class C addresses, make them contiguous
    - Eg. 192.4.16 to 192.4.31 enables a 20-bit network number
  - Idea is to enable network number to be any length
  - Collapse multiple addresses assigned to a single domain to one address
CIDR Addresses

• Identifying a CIDR block requires both an address and a prefix length
  • Slash notation
  • 128.211.168.0/21 for addresses 128.211.168.0 – 128.211.175.255
    – Here the /21 indicates a 21 bit mask
  • All possible CIDR masks can easily be generated
    – /8, /16, /24 correspond to traditional class A, B, C categories

• IP addresses are now arbitrary integers, not classes
• Raises interesting questions on route lookup
CIDR Implications

• Longest prefix match
  • 7 contiguous Class C’s given to network A:
    - 200.10.0.0 – 200.10.6.255
    - N/w number – 200.10.0.0/21

• 8th class C given to network B:
  - 200.10.7.0 – 200.10.7.255
  - N/w number – 200.10.7.0/24

• Packet with destination address 200.10.7.1 matches both networks
  - Must pick the most specific match!