CS 457 – Lecture 8
Switching and Forwarding

Fall 2011
Course So Far

• Can communicate over a point to point link
  – Encode bits on the wire (NRZ, Manchester, etc)
  – Make frames (header + data)
  – Check for errors (CRC, parity bits)
  – Reliably retransmit any lost or corrupt packets

• Can communicate over multi-access
  – Shared wire (Ethernet)
  – Shared wireless (Wi-Fi)

• But Internet is clearly not a single Ethernet or single Wi-Fi network…
Switches and Forwarding
Switches: Traffic Isolation

- Switch breaks subnet into LAN segments
- Switch filters packets
  - Frame only forwarded to the necessary segments
  - Segments become separate collision domains
  - **Bridge**: a switch that connects two LAN segments
Motivation For Self Learning

• Switches forward frames selectively
  – Forward frames only on segments that need them

• Switch table
  – Maps destination MAC address to outgoing interface
  – Goal: construct the switch table automatically
Self Learning: Building the Table

• When a frame arrives
  – Inspect the source MAC address
  – Associate the address with the incoming interface
  – Store the mapping in the switch table
  – Use a time-to-live field to eventually forget the mapping

• Switch learns how to reach A.
Self Learning: Handling Misses

• When frame arrives with unfamiliar destination
  – Forward the frame out all of the interfaces
  – … except for the one where the frame arrived
  – Hopefully, this case won’t happen very often

• When in doubt, shout!
Switch Filtering/Forwarding

When switch receives a frame:

index switch table using MAC dest address

if entry found for destination
then{
  if dest on segment from which frame arrived
  then drop the frame
  else forward the frame on interface indicated
}
else flood

• forward on all but the interface
• on which the frame arrived
Switch Example

Suppose C sends frame to D

- Switch receives frame from from C
  - notes in bridge table that C is on interface 1
  - because D is not in table, switch forwards frame into interfaces 2 and 3
- Frame received by D
Switch Example

Suppose D replies back with frame to C.

- Switch receives frame from from D
  - notes in bridge table that D is on interface 2
  - because C is in table, switch forwards frame only to interface 1
- Frame received by C
Flooding Can Lead to Loops

- Switches sometimes need to broadcast frames
  - Upon receiving a frame with an unfamiliar destination
  - Upon receiving a frame sent to the broadcast address
- Broadcasting is implemented by flooding
  - Transmitting frame out every interface
  - … except the one where the frame arrived
- Flooding can lead to forwarding loops
  - E.g., if the network contains a cycle of switches
  - Either accidentally, or by design for higher reliability
Solution: Spanning Trees

• Ensure the topology has no loops
  – Avoid using some of the links when flooding
  – … to avoid forming a loop

• Spanning tree
  – Sub-graph that covers all vertices but contains no cycles
  – Links not in the spanning tree do not forward frames
Constructing a Spanning Tree

• Need a distributed algorithm
  – Switches cooperate to build the spanning tree
  – … and adapt automatically when failures occur

• Key ingredients of the algorithm
  – Switches need to elect a “root”
    • The switch with the smallest identifier
  – Each switch identifies if its interface is on the shortest path from the root
    • And exclude it from the tree if not
  – Messages (Y, d, X)
    • From node X
    • Claiming Y is the root
    • And the distance is d
Steps in Spanning Tree Algorithm

• Initially, each switch thinks it is the root
  – Switch sends a message out every interface
  – … identifying itself as the root with distance 0
  – Example: switch X announces (X, 0, X)

• Switches update their view of the root
  – Upon receiving a message, check the root ID
  – If the new id is smaller, start viewing that switch as root

• Switches compute their distance from the root
  – Add 1 to the distance received from a neighbor
  – Identify interfaces not on a shortest path to the root
  – … and exclude them from the spanning tree
Example From Switch #4’s Viewpoint

• Switch #4 thinks it is the root
  – Sends (4, 0, 4) message to 2 and 7
• Then, switch #4 hears from #2
  – Receives (2, 0, 2) message from 2
  – … and thinks that #2 is the root
  – And realizes it is just one hop away
• Then, switch #4 hears from #7
  – Receives (2, 1, 7) from 7
  – And realizes this is a longer path
  – So, prefers its own one-hop path
  – And removes 4-7 link from the tree
Example From Switch #4’s Viewpoint

- Switch #2 hears about switch #1
  - Switch 2 hears (1, 1, 3) from 3
  - Switch 2 starts treating 1 as root
  - And sends (1, 2, 2) to neighbors

- Switch #4 hears from switch #2
  - Switch 4 starts treating 1 as root
  - And sends (1, 3, 4) to neighbors

- Switch #4 hears from switch #7
  - Switch 4 receives (1, 3, 7) from 7
  - And realizes this is a longer path
  - So, prefers its own three-hop path
  - And removes 4-7 link from the tree
Robust Spanning Tree Algorithm

- Algorithm must react to failures
  - Failure of the root node
    - Need to elect a new root, with the next lowest identifier
  - Failure of other switches and links
    - Need to re-compute the spanning tree
- Root switch continues sending messages
  - Periodically re-announcing itself as the root (1, 0, 1)
  - Other switches continue forwarding messages
- Detecting failures through timeout (soft state!)
  - Switch waits to hear from others
  - Eventually times out and claims to be the root

See Section 3.2.2 in the textbook for details and another example
Evolution Toward Virtual LANs

• In the olden days…
  – Thick cables snaked through cable ducts in buildings
  – Every computer they passed was plugged in
  – All people in adjacent offices were put on the same LAN
  – Independent of whether they belonged together or not

• More recently…
  – Hubs and switches changed all that
  – Every office connected to central wiring closets
  – Often multiple LANs (k hubs) connected by switches
  – Flexibility in mapping offices to different LANs

• Group users based on organizational structure, rather than the physical layout of the building.
Why Group by Organizational Structure?

• Security
  – Ethernet is a shared media
  – Any interface card can be put into “promiscuous” mode
  – … and get a copy of all of the traffic (e.g., midterm exam)
  – So, isolating traffic on separate LANs improves security

• Load
  – Some LAN segments are more heavily used than others
  – E.g., researchers running experiments get out of hand
  – … can saturate their own segment and not the others
  – Plus, there may be natural locality of communication
  – E.g., traffic between people in the same research group
People Move, and Roles Change

• Organizational changes are frequent
  – E.g., faculty office becomes a grad-student office
  – E.g., graduate student becomes a faculty member

• Physical rewiring is a major pain
  – Requires unplugging the cable from one port
  – … and plugging it into another
  – … and hoping the cable is long enough to reach
  – … and hoping you don’t make a mistake

• Would like to “rewire” the building in software
  – The resulting concept is a Virtual LAN (VLAN)
Example: Two Virtual LANs

- Red VLAN and Orange VLAN
- Bridges forward traffic as needed
Example: Two Virtual LANs

- Red VLAN and Orange VLAN
- Switches forward traffic as needed
Making VLANs Work

- Bridges/switches need configuration tables
  - Saying which VLANs are accessible via which interfaces

- Approaches to mapping to VLANs
  - Each interface has a VLAN color
    - Only works if all hosts on same segment belong to same VLAN
  - Each MAC address has a VLAN color
    - Useful when hosts on same segment belong to different VLANs
    - Useful when hosts move from one physical location to another

- Changing the Ethernet header
  - Adding a field for a VLAN tag
  - Implemented on the bridges/switches
  - ... but can still interoperate with old Ethernet cards
What’s Next

• Read Chapter 1 and 2
• Next Lecture Topics from Chapter 3.1 and 3.2
  – Switching and Forwarding
• Homework
  – Due Thursday
• Project 1
  – Due tonight 11:45pm
  – Submit your tar file on RamCT