The Edward S. Rogers Sr. Department
of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

ECE 361H1 F:
Computer Networks I

J.J. Garcia-Luna-Aceves

Electrical & Computer Engineering Department
University of Toronto

The Edward S. Rogers Sr. Department
of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

TRANSPARENT BRIDGES

Textbook 1, Sections 3.2.1 to 3.2.3

"Relay message"

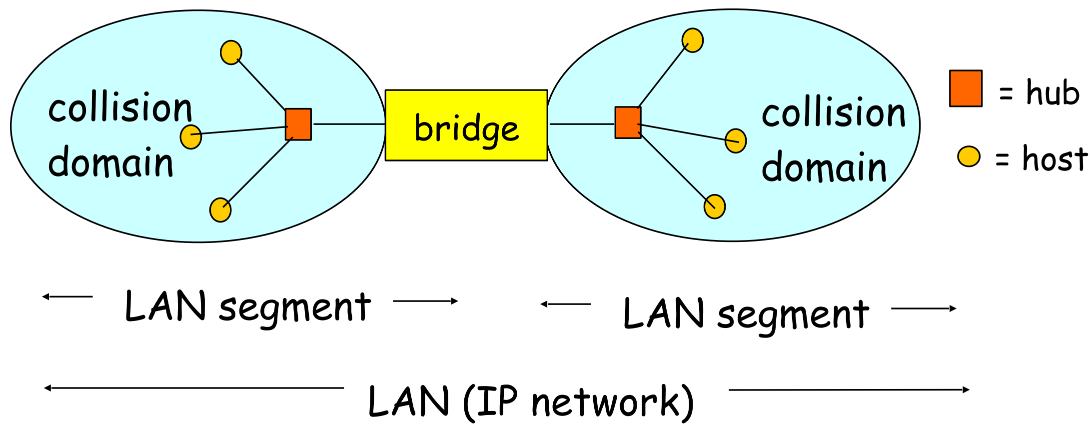
"Forward letters"

Internetworking at the Link Layer

- Bridges are used to interconnect LANs at the link layer.
- Frame forwarding from one LAN to another is based on the destination's link-level address (MAC address) without making any changes to the frame.
- A MAC address is a name, and for a bridge the address of the destination is the adjacent LAN over which the frames to the destination should be forwarded.
- Plug-and-play, self-learning
- Bridges need not be configured.

Traffic Isolation and Internetworking with Bridges

- Installing bridges breaks a large LAN into LAN segments
- Bridges **filter packets**:
  - Same-LAN-segment frames not usually forwarded onto other LAN segments
  - LAN segments become separate collision domains



- To which LAN segment should the bridge forward a frame?
- **A routing problem!**
- There are three types of bridges:
  - **Transparent bridges**

    (e.g., IEEE 802.1D, IEEE 802.1Q)
  - **Source-routing bridges**
  - **Shortest-path bridges**

    (e.g., RFC 6329 on IEEE 802.1aq)

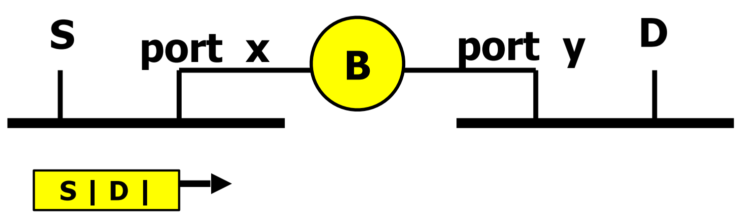
Transparent Bridges

- The purpose of transparent bridges is to keep the packet forwarding functionality transparent to the hosts.
- Transparent bridges establish and manage a spanning tree of the network to eliminate packet looping.
- The address of a station is always the LAN over which packets from that station came last; this is a dynamic process.
- If no address is known, a bridge broadcasts packets for a station over all its ports (or those in the spanning tree).

Addressing in Transparent Bridges

- Assume for now that the topology of the internet is a tree.
- A bridge listens to every packet it receives over any LAN.
- The bridge builds a **station cache** consisting of the source addresses of packets it hears and the IDs of the ports over which the packets were heard. Like **"host @ port"**
- For the bridge, the address of a station is the port over which packets from the station were received.

Bridge B assigns port x as the address of station S after hearing the packet from S.



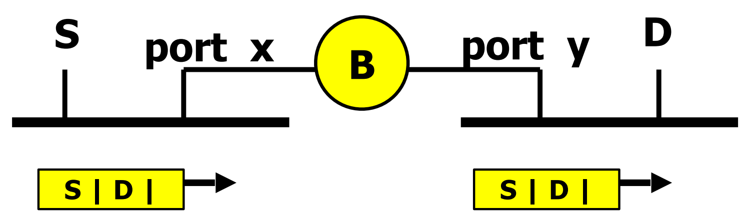
Addressing in Transparent Bridges

- Keep assuming a tree topology
- When a bridge receives a packet, it looks up its station cache for the destination MAC address in the packet.
- If match is found then:
  - If port in the cache is the same port over which packet came, the packet is filtered (dropped)
  - Otherwise, the bridge forwards the packet to the port specified in the cache.
- If no matching is found, the bridge forwards the packet over all ports other than the port from which the packet came.

B has station cache entry: D - port y

or

B does not know about D and forwards to port y (and other ports other than x)

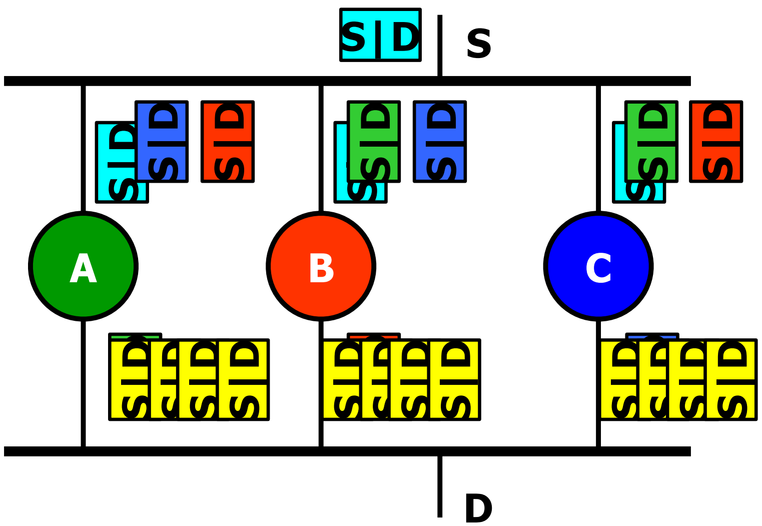


Major Problem: Looping Occurs in Mesh Topologies

- The address learning process is such that packets will traverse loops, and worse, replicas of such packets will be produced and sent over the same loops!
- **S sends a packet to D, D is silent, and bridges do not know about D.**

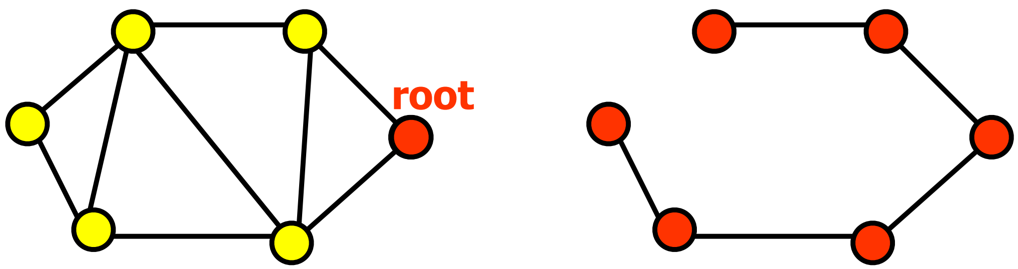
Packet is replicated three times at each LAN each time it is forwarded by the bridges!

Looping with bridges leads to traffic explosion



The Spanning Tree Algorithm (STA)

- The objective is to define a single spanning tree in the internet over which packets flow without looping.
- Basis of operation (Perlman 1992, part of IEEE standard):
  - Elect a single bridge as the **root** of the tree in a distributed manner
  - Calculate distance (in hops) on a shortest path to root
  - Elect a **designated bridge** for each LAN (e.g., closest to the root in the LAN)
  - Allow only designated bridge to forward packets to and from its LAN



A distributed election algorithm is used to build the spanning tree

STA Operation

- Each bridge has multiple MAC ID's or addresses (one per port)
- A bridge has a bridge-wide ID (one of the MAC addresses)
- HELLOs: messages used to build tree, sent to all bridges of a LAN
- **HELLO specifies:**
  - **Root ID:** The MAC address of the bridge assumed to be the root
  - **Transmitting bridge ID:** MAC address of bridge sending HELLO
  - **Cost:** Length (in hops) of path from bridge to root
- A bridge starts by considering itself the proposed root
- Bridge starts election process by sending

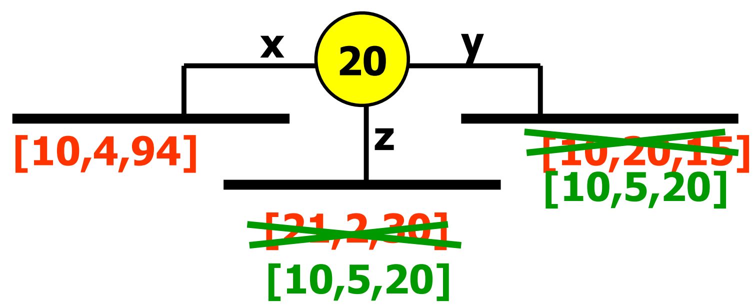
  HELLO = own ID, 0, own ID

STA Operation, Cont.

- Bridges **adopt the smallest HELLO they hear**:
  - Minimum root ID
  - Smallest distance to root
  - Minimum reporting bridge ID
- Bridge compares its own HELLO with its neighbors' HELLOs, and chooses the smallest
- Its **root port** becomes the port to neighbor bridge with smallest HELLO
- Bridge composes a new HELLO, adding 1 to the distance to adopted root

Bridge 20 must adopt HELLO from neighbor 94 over port x: smallest root ID and smallest distance to root!

Sends HELLO stating [10, 5, 20] over ports y and z

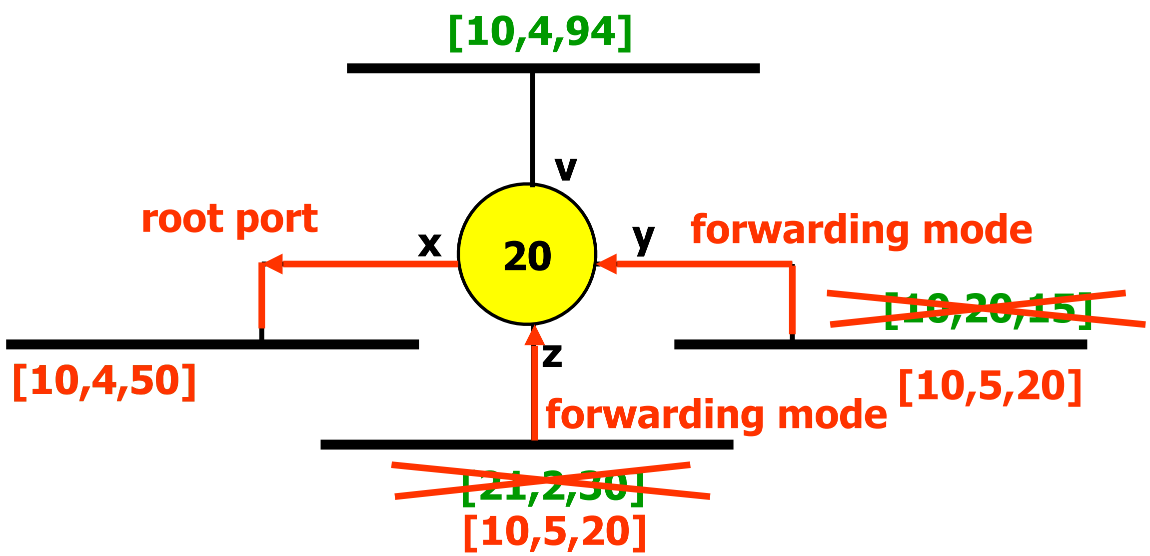


STA Operation, Conc.

- A bridge sends new HELLO over all ports from which “larger” HELLOs were received.
- Bridge knows if it is the designated bridge for a LAN if it does not hear a “smaller” HELLO than its own.
- Its root port is the port from which the smallest HELLO was received.
- Bridge puts its root port and all ports for which it is the designated bridge in **forwarding state.**
- Bridge puts all other ports in **blocking state**.
- Data packets, control packets, and learning of addresses take place only over ports in forwarding state (over the spanning tree).

Example of STA Operation

- Consider a given bridge 20:



Handling Failures in STA

Procedure to fix the tree when a failed bridge stops sending HELLOs breaking tree:

- Each HELLO has an **age field**
- Age of of HELLO is incremented over each hop and each time unit while in storage and stored HELLOs are discarded when ages reach a maximum value
- Root sends HELLO periodically with 0 age
- Bridge recomputes best HELLO (with a valid age) for port for which HELLO is deleted
- Bridge can decide to become root if it provides the best new HELLO
- Spanning tree calculation occurs when a HELLO is received from a port or a stored HELLO is discarded
- Receiving bridge forwards HELLO (with new distance and its own ID for reporting bridge) for all ports for which it is designated bridge

HELLO from root propagates through spanning tree in the absence of failures

If root or other bridge fails, bridges down-tree stop receiving HELLOs originated by the root

Temporary Loops in STA

- STA does not guarantee that the aggregate of ports in forwarding mode define a tree [or a forest] at every instant.
- **Why?** Bridges select their root port w/o any ordering constraint
- Because packets have no TTL, they loop indefinitely and multiply, until tree is correct.
- Ad hoc approach to cope with loops:
  - Make a bridge wait a long time before moving a port to forwarding mode.
  - Hold-down timer is set to twice the maximum transit time in the internet (say 30 sec).
  - This solution is slow in large internets or does not work at all, because we do not know the maximum transit time after a topology change.

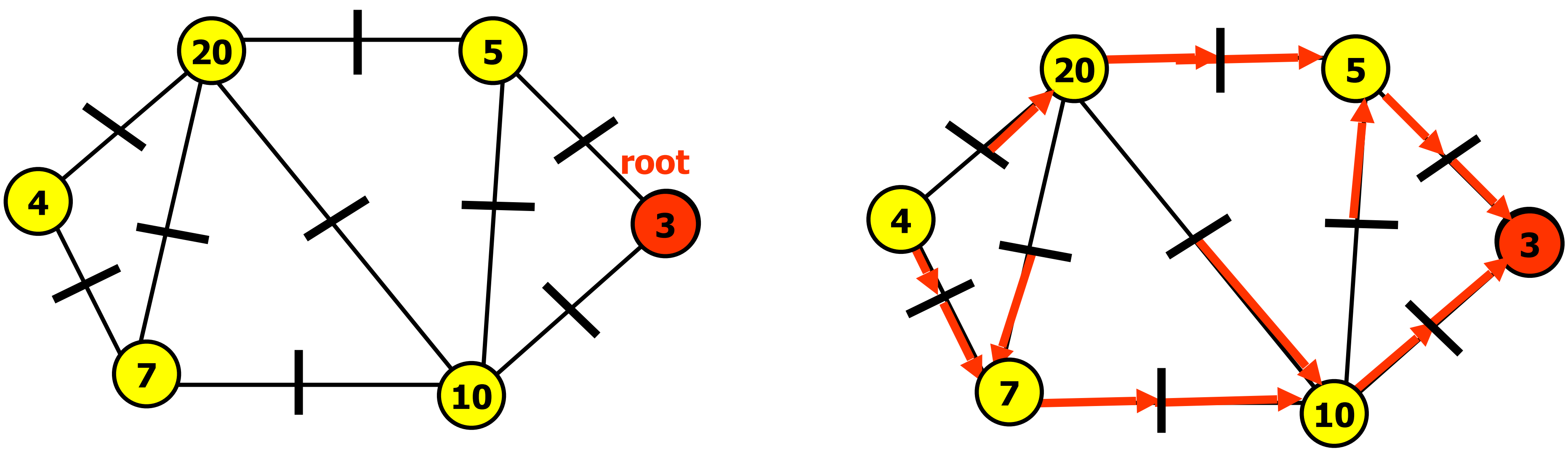
Refreshing Station Cache

- The tree can change due to link and bridge failures or stations physically moving.
- Bridges must distinguish between the two types (stations move very slowly) to decide how often to refresh station cache (so that packets are not forwarded in vain).

Approach:

- Bridge that changes the state of a port sends a topology change notification (TCN) to the root (on its root port) persistently, until the “parent” bridge ACKs (with a flag in its HELLO)
- Bridge receiving a TCN forwards it towards the root
- If root detects a topology change or receives a TCN, it sets the TCN flag in its own HELLO for a period of time (say 30 sec).
- The HELLOs with TCN set force bridges to refresh their station caches more often, until the TCN is reset.

Implications



- Limitation: Many wasted links and temporary loops may still occur
- Shortest-path bridges remedy this problem by implementing link-state routing (IS-IS protocol extension) and hence use all links as needed (e.g., **IEEE 802.1aq, RFC 6329**)
- Why do we need such a complex routing protocol?

END