Transparent (spanning tree) bridge.

Objective: Plug and play (no mods to software necessary).

Operate in promiscuous modes on all links
Source LAN = Dest LAN ==> Discard
Source LAN != Dest LAN ==> Forward
Dest unknown ==> Flood.

Backward learning used to know where to forward.

Spanning tree algorithm needed because of possible cycles
Tree constructed every few seconds via flooding device ID.

Spanning tree alg:
1 – Elect a root bridge
2 – For each *bridge* calculate distance to root bridge
3 – For each *lan* select the *designated* bridge = closest to the root bridge
4 – For each *bridge* select the *root* port = port closest to root bridge
5 – Select remaining ports to be included in spanning tree = those for which "self" is a designated bridge

Root bridge election

Configuration messages:
(Root ID, Transmitting Bridge ID, Cost, Port ID)

Given two messages C1 and C2, C1 is better than C2 if:

Root ID in C1 is numerically smaller than root ID in C2
If Root ID's are equal break ties on cost
If costs are also equal break ties on Transmitting bridge ID
If Transmitting bridge ID's are equal break ties with Port ID.
(Happens only if transmitter has 2 ports on same Lan)

Each bridge starts by sending a message with itself as root
If it receives a better message *on one of its attached LAN's* it ceases transmitting configuration messages *on that LAN.*
When algorithm stabilizes only one bridge per LAN transmits.
That bridge is called the designated bridge for that LAN

Calculating cost to Root

There will be a "wining" configuration message received on each port
That message will be transmitted by the designated bridge for that LAN.
The lowest (numerically) root id received is the root bridge.
The cost from *this* bridge to the root is computed by adding 1 to that cost.
The designated bridge for a LAN is the one with the lowest cost to the root.

Calculating Root port

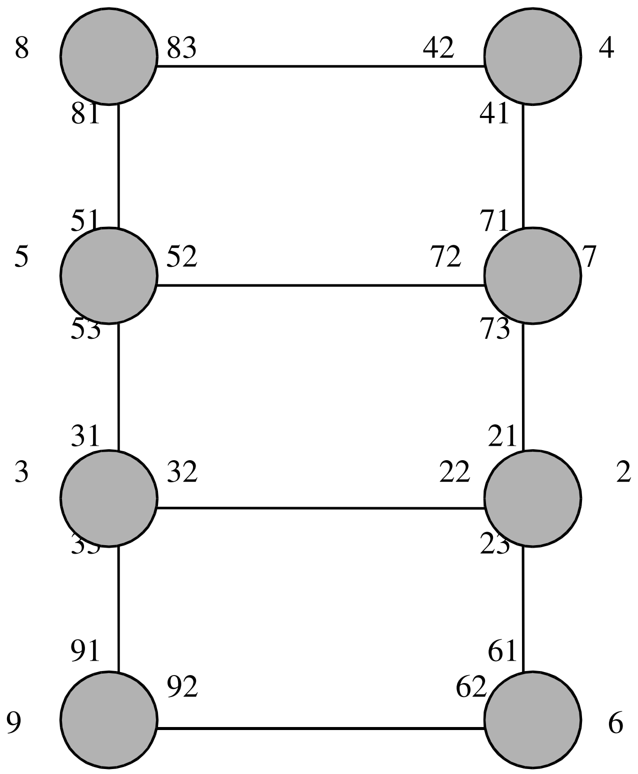
Root port is the port with the lowest cost path to the root (subject to usual tie breakers.)

Selecting designated bridge

If my message is the best message transmitted on the LAN then I am the designated bridge for that LAN

Forwarding rules

Forward all packets on from ports in spanning tree to other ports on spanning tree.
Discard all data packets received on ports not in tree.
Continue to run spanning tree alg on *all* ports



I think that I shall never see
A graph more lovely than a tree.

A tree whose crucial property
is loop−free connectivity.

A tree that must be sure to span
So packets can reach every LAN.

First, the root must be selected.
By, ID, it is elected.

Least cost paths from root are traced.
In the tree, these paths are placed.

A mesh is made by folks like me,
Then bridges find a spanning tree.

– Radia Perlman

Source routing bridge

Each LAN has a 12 bit sequence number
Each Bridge has a 4 bit sequence number
Sender node specifies exact route each packet is to take.
Route looks like B2–L4–B3–L5–B2–L6
If route is unknown a *discovery* packet is flooded through the net.

Summary of bridge differences:

Issue	**Transparent**	**Source routing**
Orientation	Connection less	Connection oriented (SNA)
Transparency	Fully transparent	End user transparent
Configuration	Automatic	Net manager at install
Routing	Spanning tree	Optimal
Route identification	Backward learning	Flooding
Failures	Handled in net	Host software
Complexity	Bridges	Bridges + Hosts

Examples of the Network Layer

Systems Network Architecture

Connectivity Elements

- The *transmission group* (TG) is the basic unit of connectivity.
  - TG is one or more links connecting two IMPS.
  - 0 – 255 TG's between any node pair.
  - Routing affinity is TG based.
- TG activation and deactivation:
  - First active link makes TG active.
  - Last link to fail makes TG inactive.
- Assigning links to TGs.
  - Many TG's with few links ==> better workload partitioning.
  - Few TG's with many links ==> greater the reliability (and the resequence problem.)

Routing Elements

- *Explicit route* (ER) is the basic end to end routing unit.
- An ER must be invertible to be usable.
- An ER is uniquely identified by a (Dest IMP#, ER #)
- The ER must be defined by net admin to each imp through which it passes.
- ER numbers range from 0, to 15

Example routing table for node F:

Dest IMP	**ER #**	**TG_ID**
A	4	2
	5	3
B	3	2
	4	1
C	1	1
	2	2
	3	4

Note that it is the combination of (DEST, ER#) that actually identifies the Route.
Duplicate ER#'s are supported at each node.