Exploring new paths for the evolution of Transparent Bridges

ARP-Path: Revisiting Backward Learning Switches

Introducing an address-port locking mechanism to set up low latency unicast paths and multicast trees (loop free)

Guillermo Ibanez GIST Netserv group Universidad de Alcalá.Madrid. Spain.

Universidad de Alcalá
Dpto. de Automática
Área de Ingeniería Telemática

UNION EUROPEA
FONDO SOCIAL EUROPEO

La Suma de Todos
Consejería de Educación
Comunidad de Madrid
www.madrid.org

Junta de Comunidades de
Castilla-La Mancha

Contents

- Introduction.
  - The need for simple bridging protocols
  - New conceptual approaches needed for evolution
- ARP-Path switches
  - Path set up and repair
  - Implementations: Linux, Openflow/NetFPGA
  - Simulation results: throughput and load distribution
    Compatibility with standard bridges
  - Applicability
- Conclusion

The well known limitations of Transparent Bridges…

- Standardized: 802.1D Rapid Spanning Tree Protocol and 802.1Q Multiple STP (MSTP).
- Lack of scalability of the spanning tree protocols
  - Limitation of spanning tree (RSTP) protocols:
    - network size, blocks all redundant links, paths are not shortest paths
  - Complexities of configuring and scaling MSTP excluded it as an alternative
- Burden of **IP addresses administration** (subnets) in campus networks
  - Importance of using a single IP subnet in campus and data center networks
    - Server/network virtualization increases the need of single IP subnet in datacenters due to dynamic server assignment
  - DHCP does not solve the problem, address administration still needed
- **Need of a *simpl*e self configuring, single IP subnet, scalable architecture.**
- **The performance potential of Ethernet Switches is underutilized**

…have generated standards based on link state routing

- TRILL . Campus and data centers
- 802.1aq Shortest Path Bridging. Provider and Interprovider Networks.
- Proposals like SEATTLE…..
- *All use link state routing in layer two*
  - *Proven, mature, standards close to completion.*
- ***But there is still room and demand for simple, efficient and compatible bridging protocols for campus and data centers***

New conceptual approaches are needed to evolve the bridge concept in the following years

- Statements:
  - *New bridge protocols preserving conceptual and architectural **coherence with the bridge concept** have not been explored enough*
  - *Evolved **pure bridging** protocols might achieve, through simplicity, high performance at low cost in network scenarios not requiring full path predictability and controllability but high availability and performance.*
- Our target in bridge research :
  - *Simple and high performance protocols for basic bridges to replace the Spanning Tree Protocol in campuses and data centers.*
    - *Low latency or near Shortest Path bridging*
    - *Load distribution capability, if simple*
    - *High reliability*
    - *IEEE 802.1D, .1Q, .1ag, … compatible*

A proposal for the conceptual evolution of transparent bridges…

ARP-Path: ARP-based, Transparent Low Latency Bridges

Guillermo Ibanez, J.A. Carral, Alberto García-Martínez (UC3M)
José M. Arco, Diego Rivera, Elisa Rojas, Arturo Azcorra (IMDEA Networks)
GIST research group. Telematic Engineering Area. Dpt. Automatica.
Universidad de Alcalá (Madrid).
Supporting projects: MEDIANET (Comunidad de Madrid) , EMARECE (C. Castilla La Mancha)
guillermo.ibanez@uah.es

Universidad de Alcalá
Dpto. de Automática
Área de Ingeniería Telemática

UNION EUROPEA
FONDO SOCIAL EUROPEO

La Suma de Todos
Consejería de Educación
Comunidad de Madrid
www.madrid.org

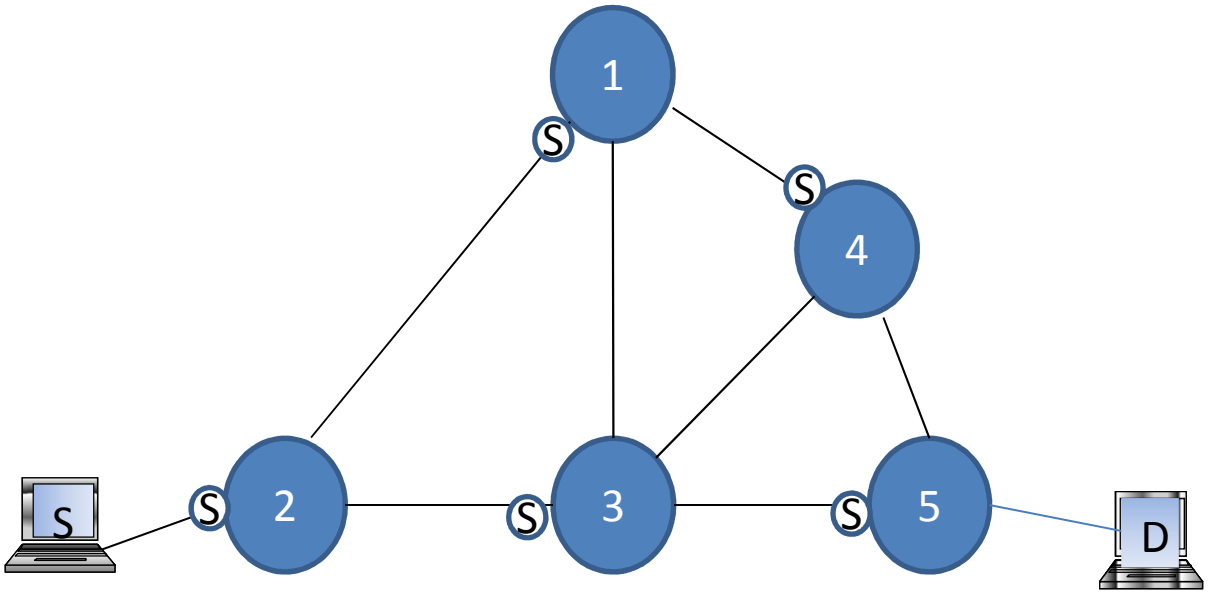
Junta de Comunidades de Castilla-La Mancha

Finding paths without routing

- *Transparent bridges with modified backward learning may achieve low latency paths*
  - *without spanning tree*
  - *without link state routing.*
  - *without ancillary loop prevention mechanisms*
- *How ?*
  - *Modify the transparent bridge mechanisms of forwarding, learning and filtering.*
  - *Flood the standard ARP Request frames issued through all network links to find the fastest path.*

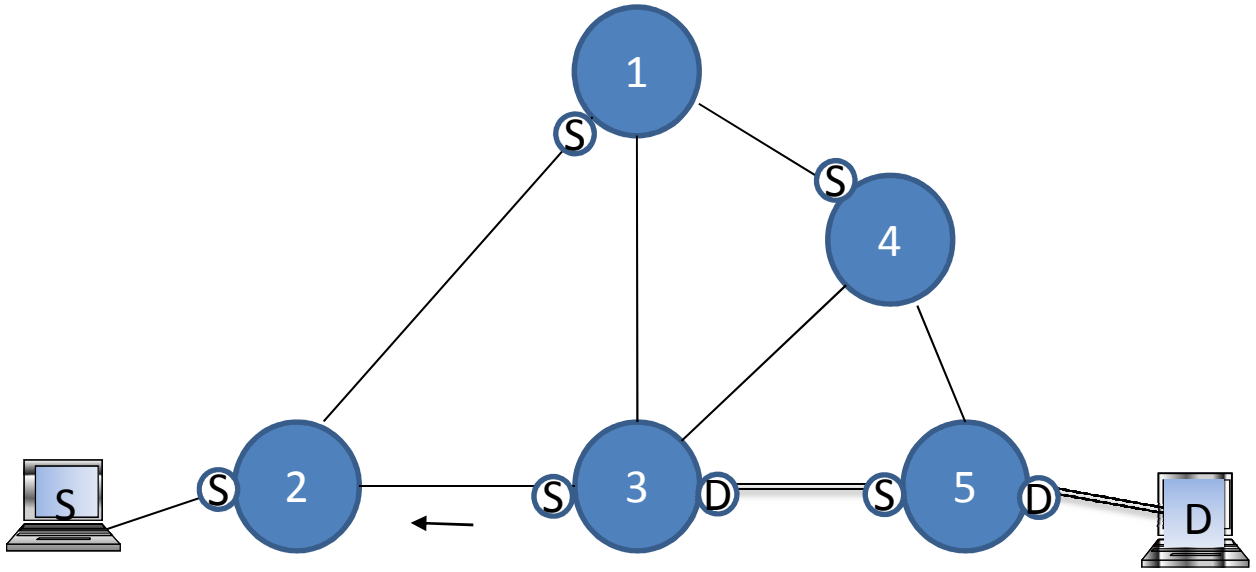
ARP Path basics

- Establish unicast paths and multicast trees just by controlled flooding of a broadcast frame e.g.: ARP Request.
- A temporary tree is established towards the source by ***learning and locking the source address to the port of the bridge that receives first the frame.***



ARP Path basics

- The path in the opposite direction (to destination host) is created by the unicast ARP Reply frame traversing the network from destination host towards source and is the symmetric path of the ARP Request path.

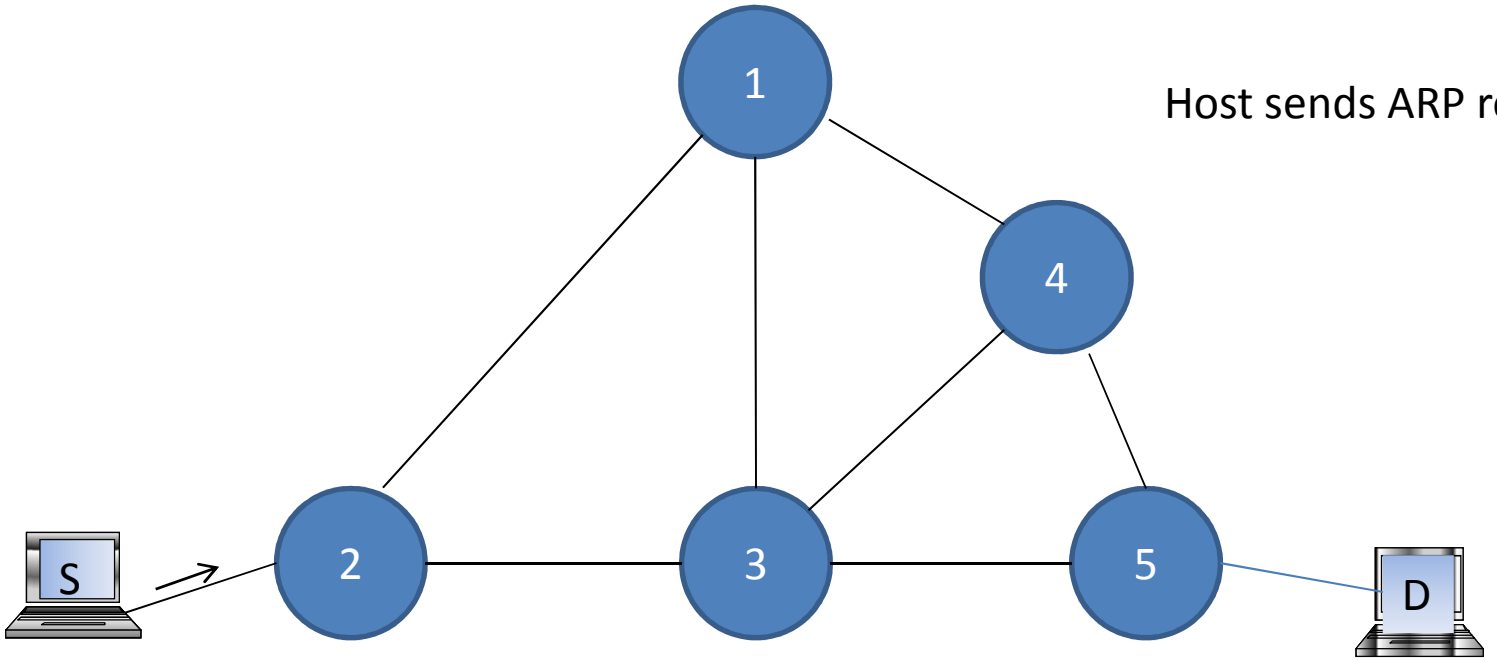


Transparent bridges with low latency paths.

- Optimizing the protocol :
  - *Use and existing broadcast frame (the standard ARP messages) to set up trees and shortest paths*
    - ***No extra message cost*** *to set up paths (besides redundant links)*
- How to avoid broadcast loops?*:*
  - ***Learning and locking the learning of the address*** *to the port of the bridge that receives first the frame.*
  - ***Discard for a short time all broadcast frames received via a different port***

Path set up from host S 1

ARP Request

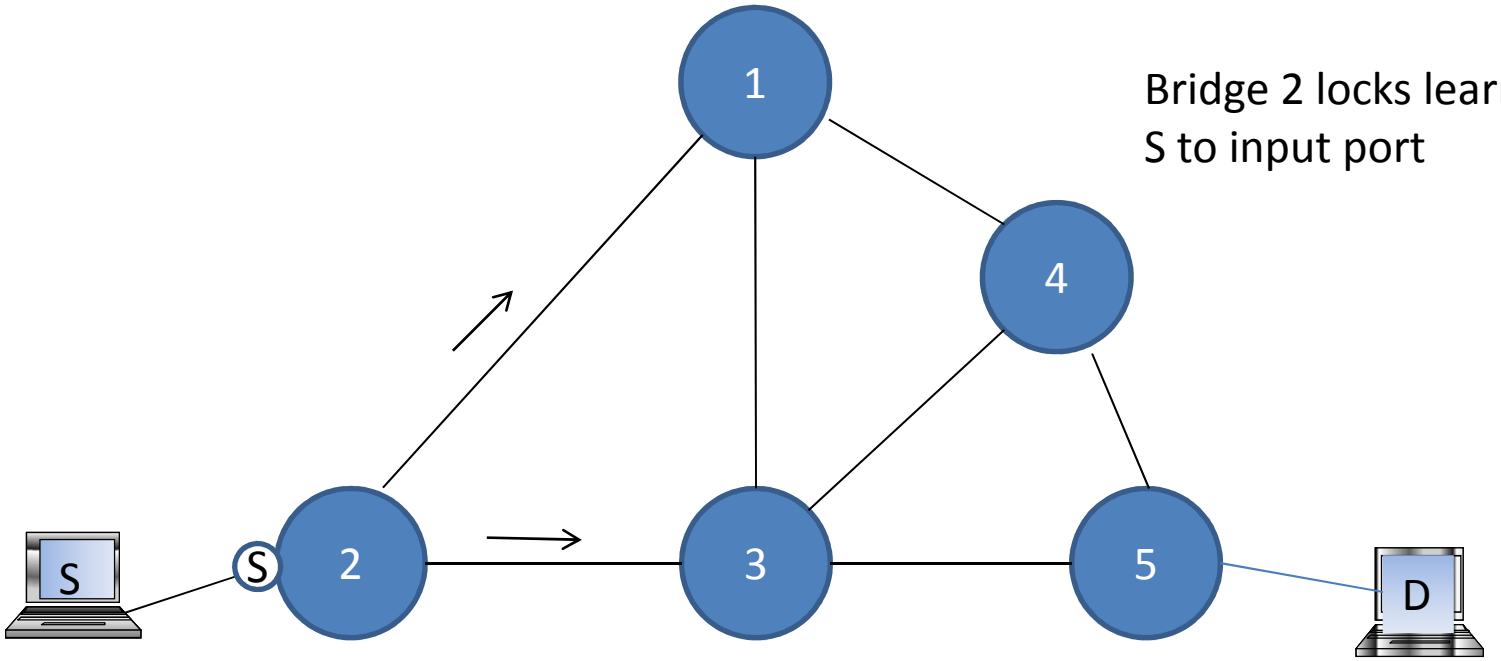


Host sends ARP request frame

ARP Request (broadcast)

Path set up 2

ARP is flooded



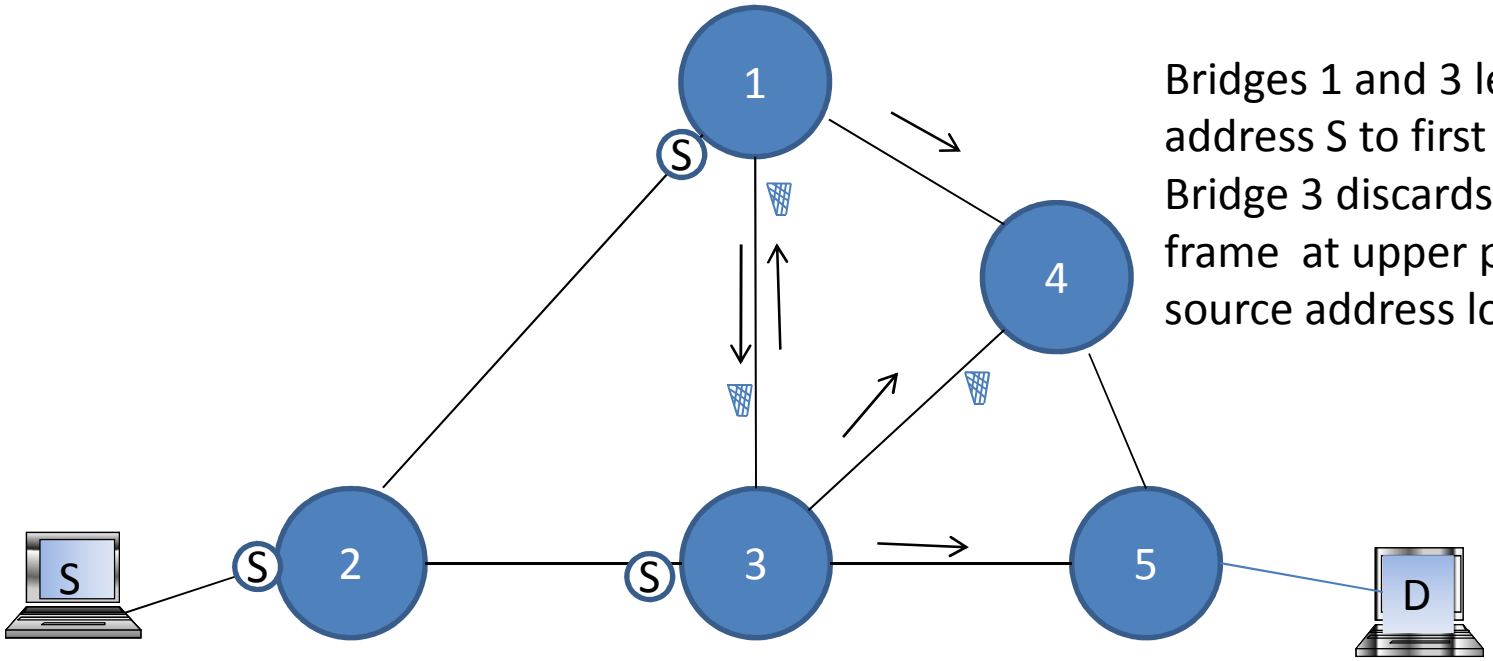
Bridge 2 locks learning of address S to input port

S *Port locked to S*

ARP Request (broadcast)

Path set up 3

ARP propagates through all links



Bridges 1 and 3 learn and lock address S to first input port (left)
Bridge 3 discards late (duplicate) frame at upper port because source address locked

S *Port locked to S*

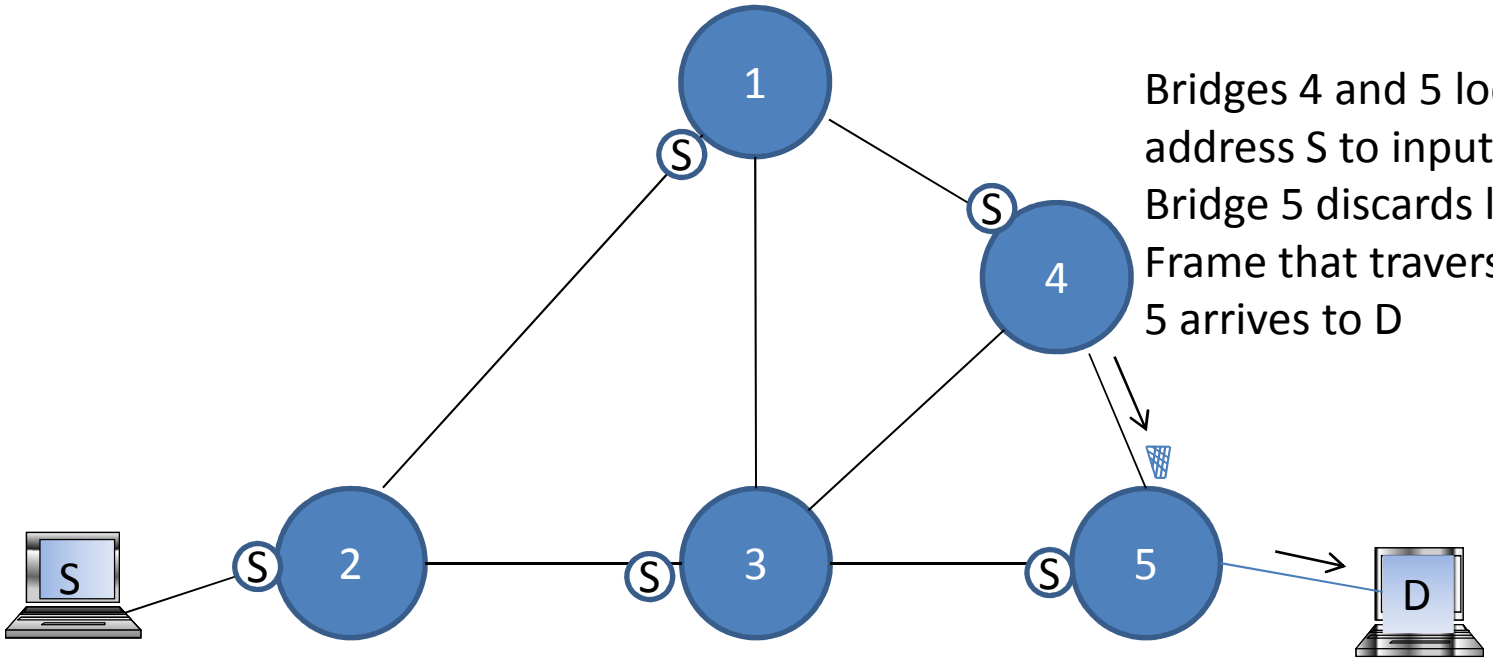
D *Port locked to D*

ARP (path) request (broadcast)

Late frame discarded

Path set up . 4

The fastest ARP Request reaches destination host



Bridges 4 and 5 lock learning of address S to input port
Bridge 5 discards late frame
Frame that traversed bridges 2-3-5 arrives to D

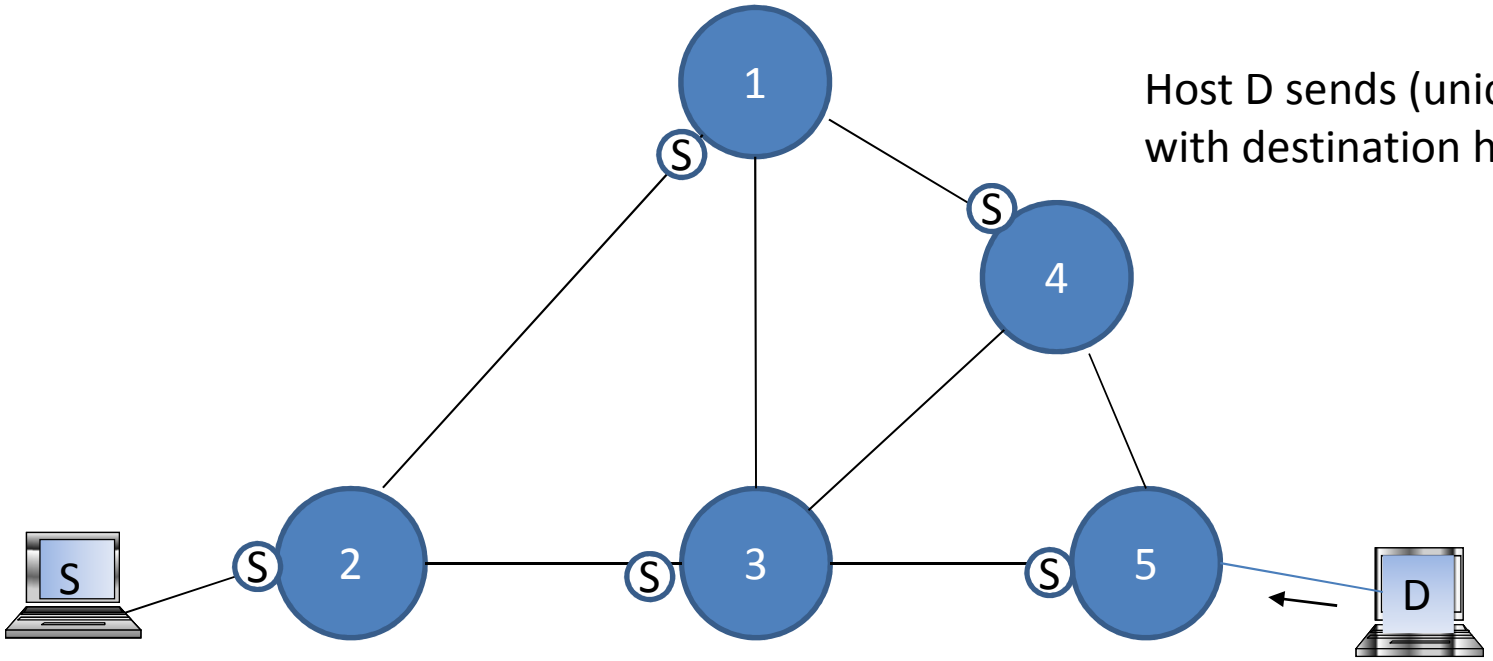
S *Port locked to S*

D *Port locked to D*

→ *ARP (path) request (broadcast)*

← *ARP (path) reply (confirm) (unicast)*

Path set up . 5
ARP Reply (unicast)



Host D sends (unicast) ARP Reply with destination host S

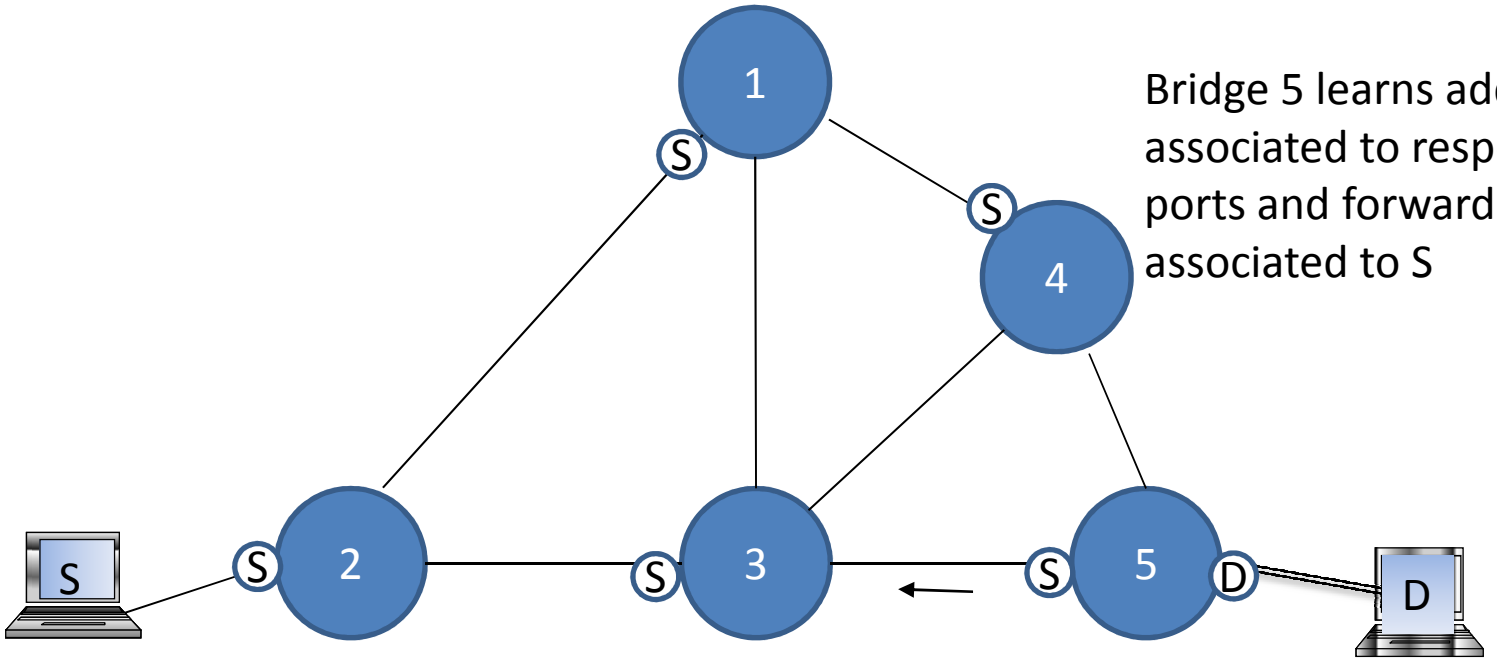
S *Port locked to S*

D *Port locked to D*

→ *ARP (path) request (broadcasted)*

← *ARP (path) reply (confirm) (unicast)*

Path set up . 6
ARP Reply



Bridge 5 learns address D associated to respective input ports and forwards via port associated to S

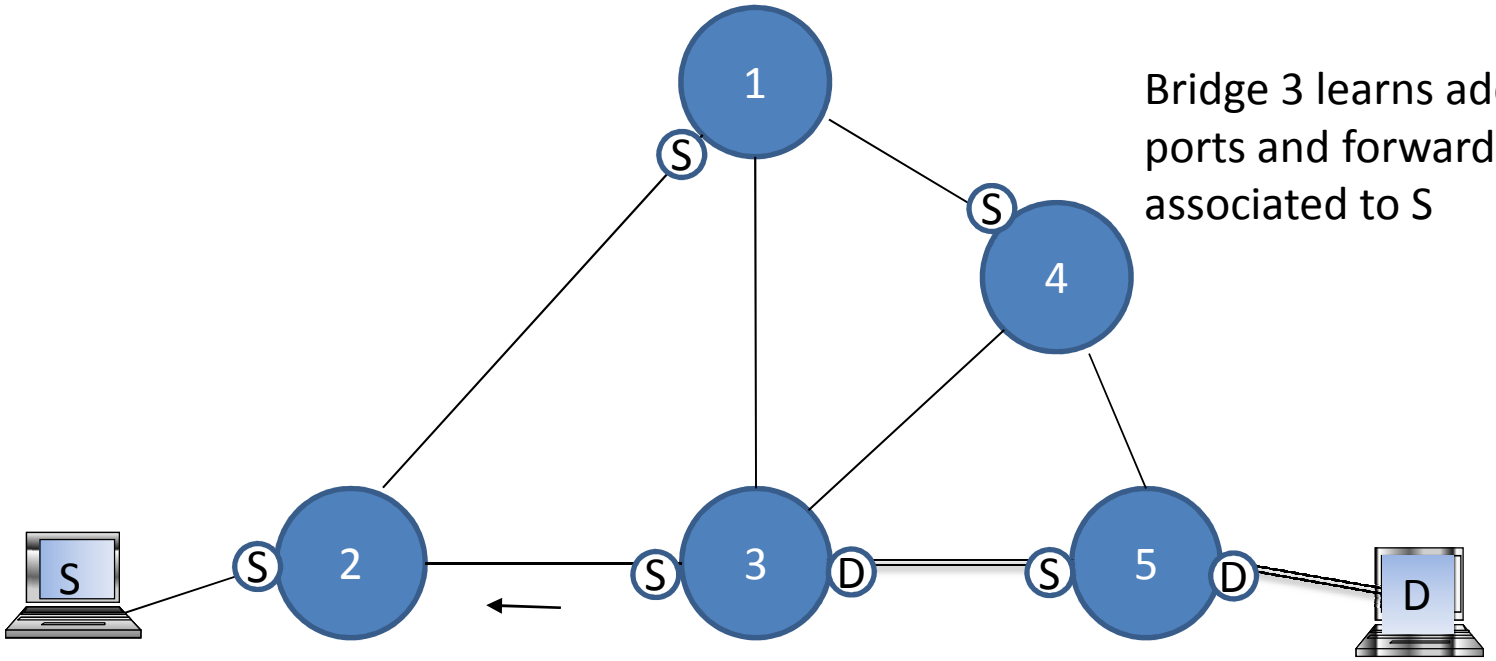
S *Port locked to S*

D *Port locked to D*

→ *ARP (path) request (broadcasted)*

← *ARP (path) reply (confirm) (unicast)*

Path set up . 7
ARP Reply



Bridge 3 learns address D at input ports and forwards via port associated to S

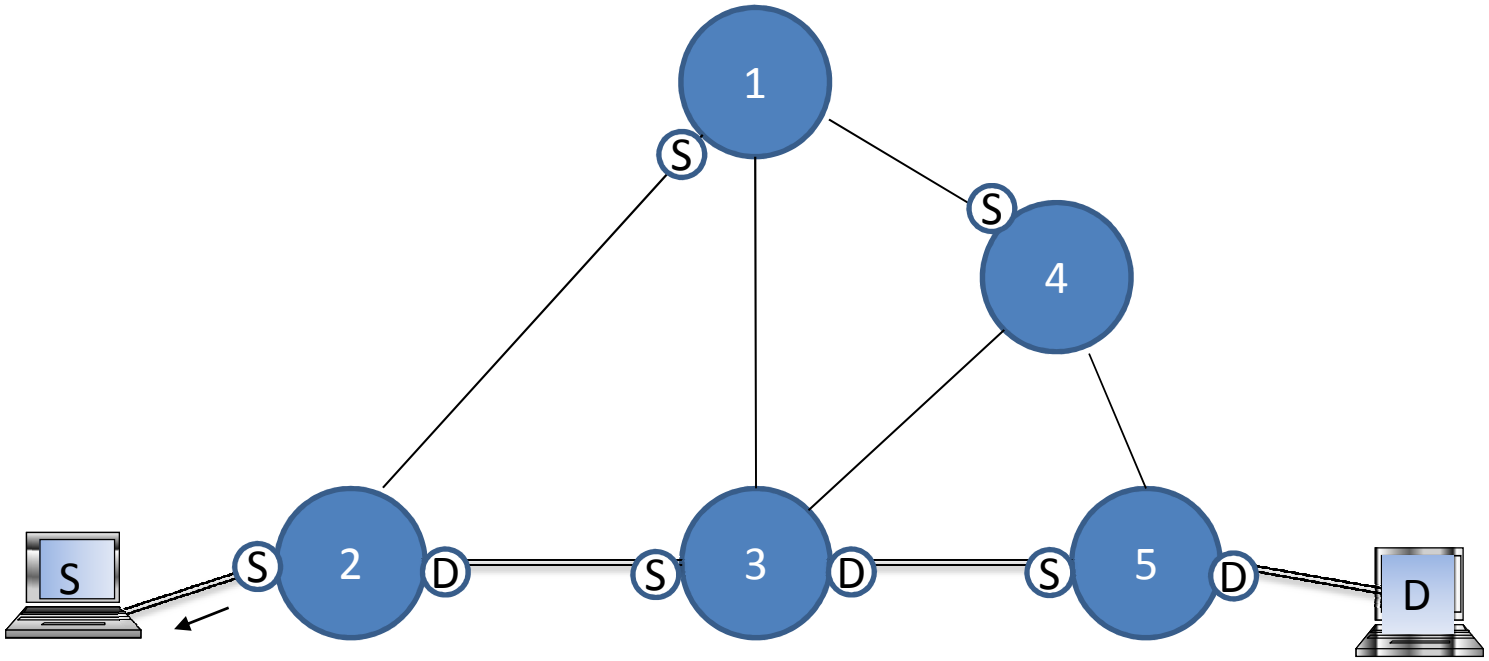
(S) *Port locked to S*

(D) *Port locked to D*

→ *ARP (path) request (broadcasted)*

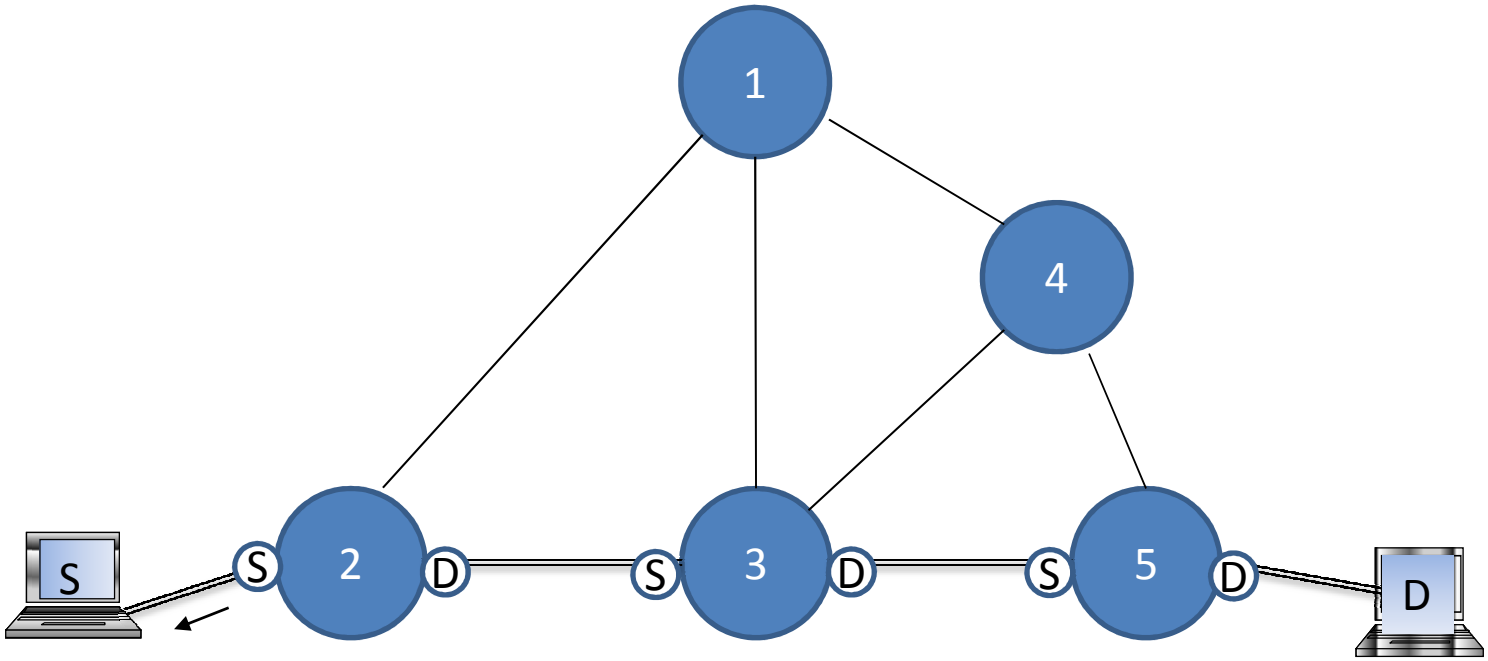
← *ARP (path) reply (confirm) (unicast)*

ARP Reply arrives at S and completes the path set up



A symmetrical path is built between S and D
A temporary tree towards S is built

Other tree branches created, but no confirmed, expire



S *Port locked to S*

D *Port locked to D*

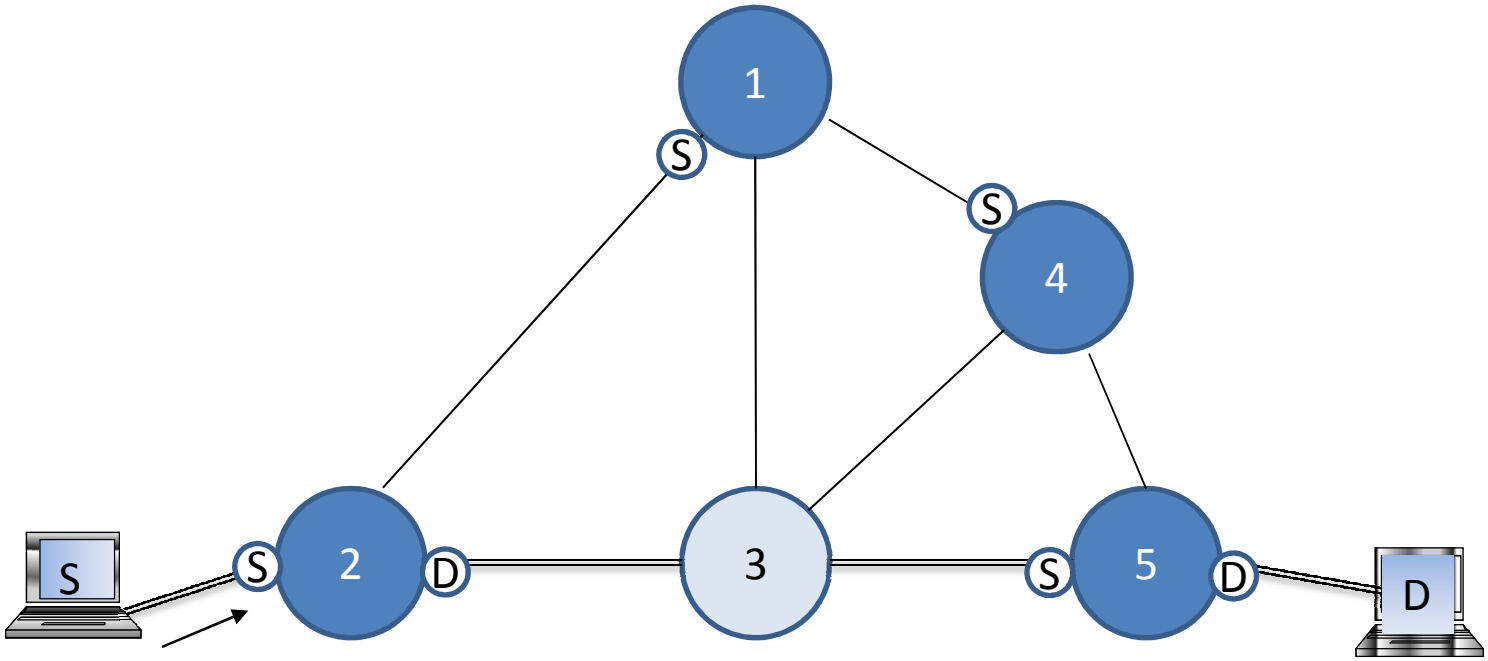
→ *ARP (path) request (broadcasted)*

← *ARP (path) reply (confirm) (unicast)*

Path repair

- A bridge reinitializes, learnt MACs are flushed.
- A unicast frame arrives at a bridge where its destination address is unknown (not associated to any port as source).
- Several variants to repair the path
  - ARP Request reissued from the bridge w/o path
    - Does not work if there are no redundant links in forward direction
  - Encapsulate frame on broadcast frame (with all ARP-Path bridges multicast destination address and return it via input port towards source bridge, who reissues ARP Request.
  - Other variants possible

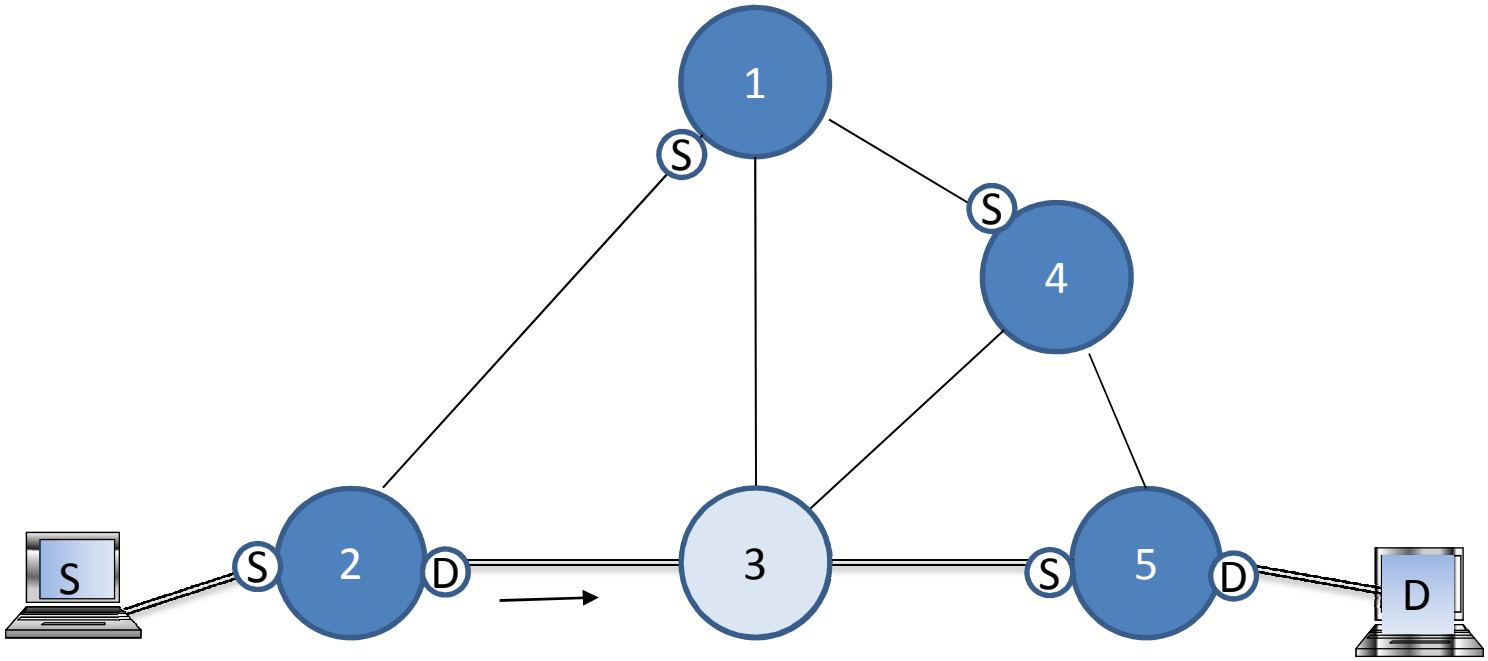
Path repair (bridge 3 flushed all its MACs by initialization after failure)



S *Port locked to S*

D *Port locked to D*

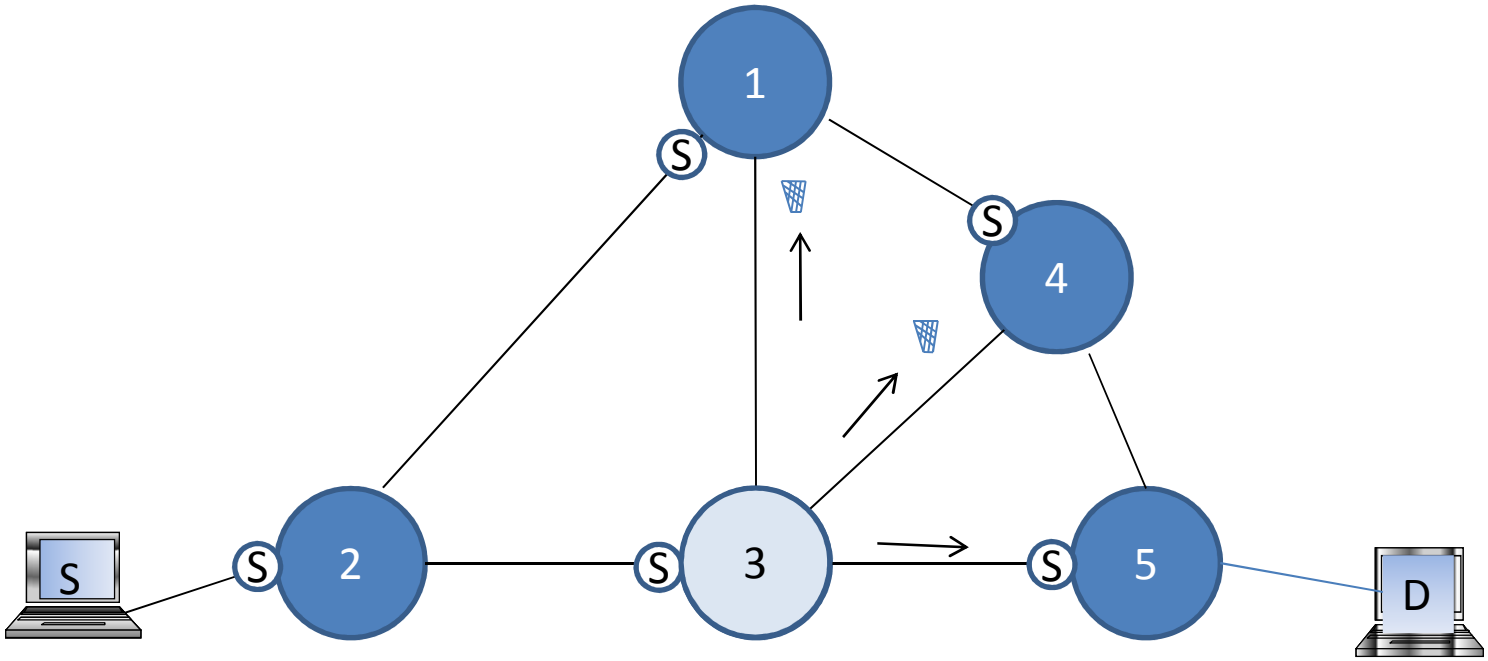
Path repair (bridge 3 had all MACs flushed by initialization)



S *Port locked to S*

D *Port locked to D*

Path repair (bridge 3 issues ARP request)



S *Port locked to S*

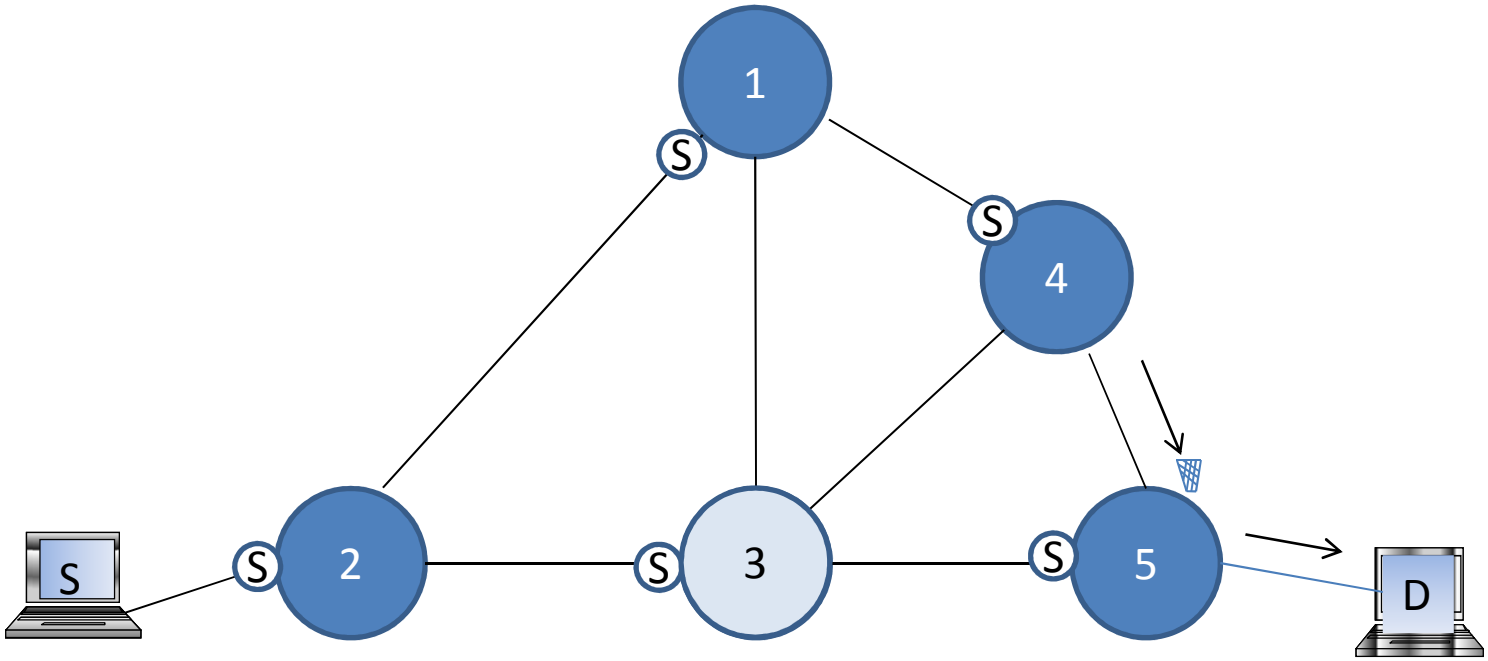
D *Port locked to D*

ARP (path) request (broadcasted)

Late frame discarded

ARP (path) reply (confirm) (unicast)

Path repair



S *Port locked to S*

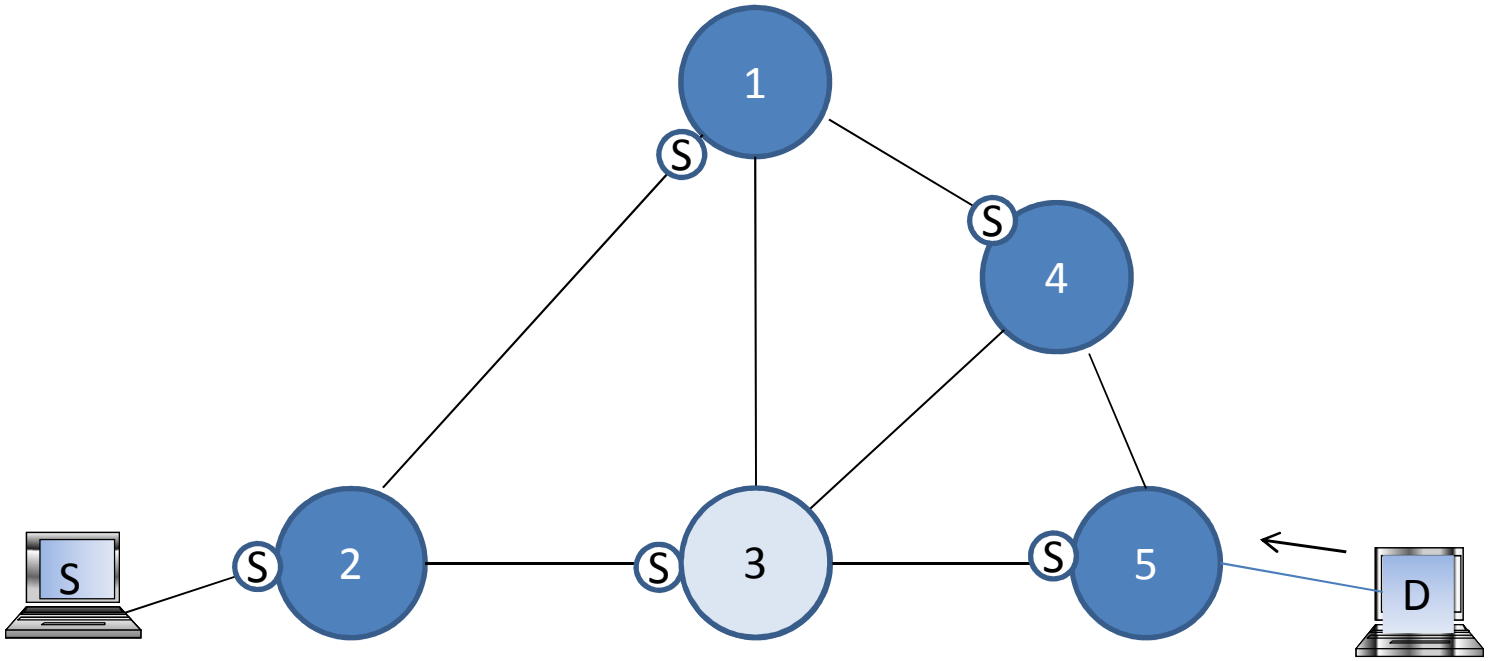
D *Port locked to D*

ARP (path) request (broadcasted)

Late frame discarded

ARP (path) reply (confirm) (unicast)

Path repair completing



S *Port locked to S*

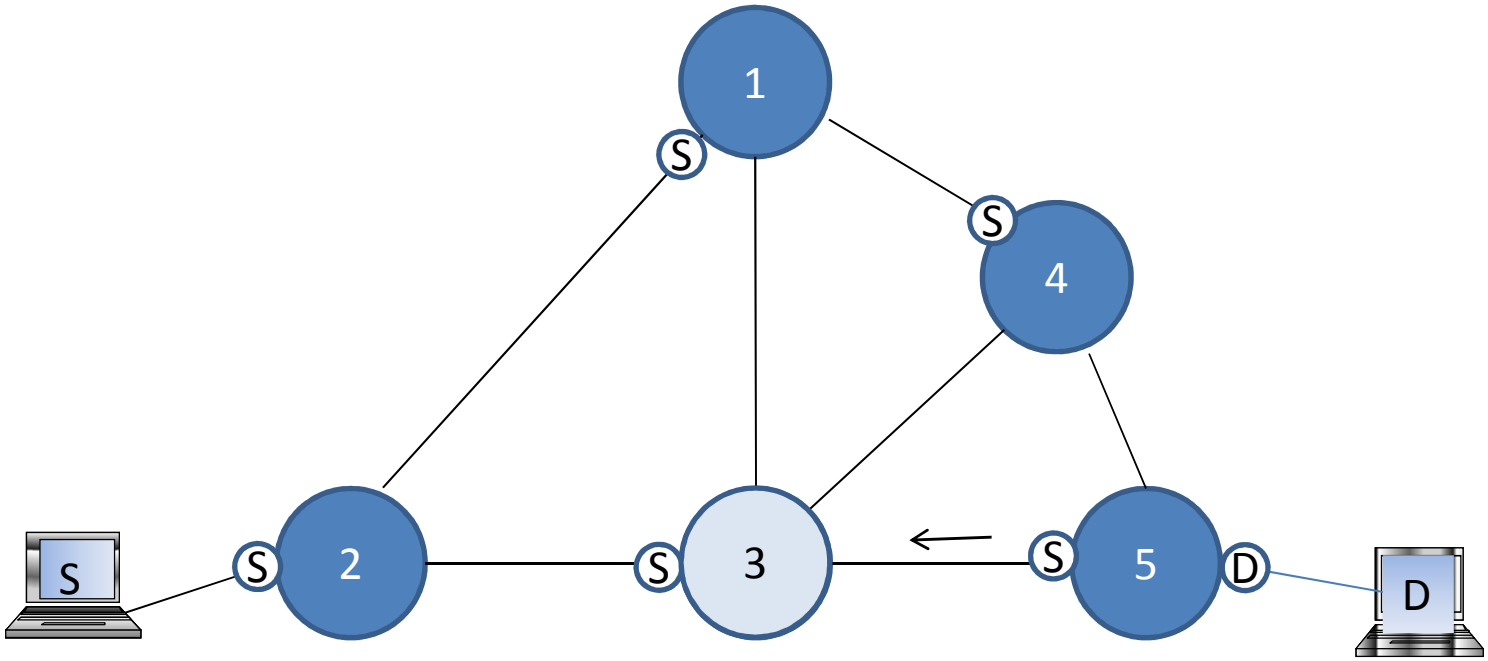
D *Port locked to D*

ARP (path) request (broadcasted)

Late frame discarded

ARP (path) reply (confirm) (unicast)

Path repair completing



S *Port locked to S*

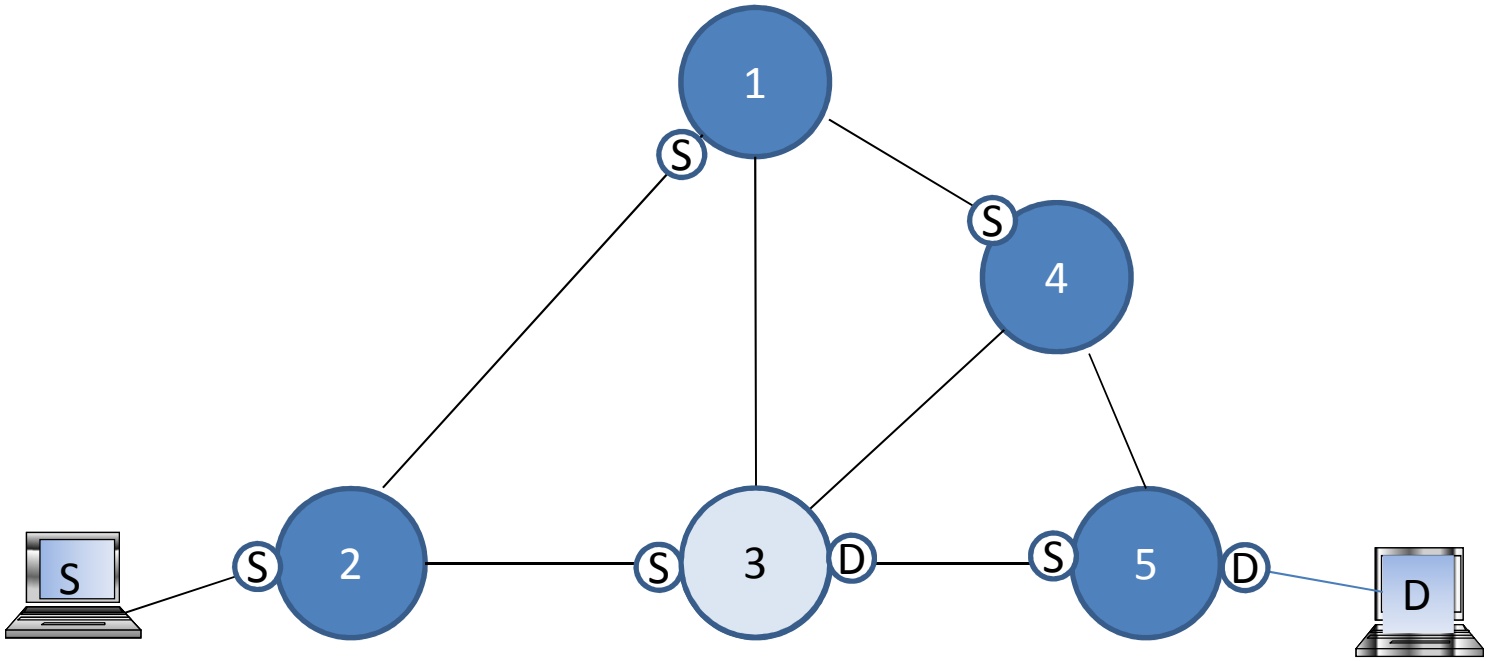
D *Port locked to D*

ARP (path) request (broadcasted)

Late frame discarded

ARP (path) reply (confirm) (unicast)

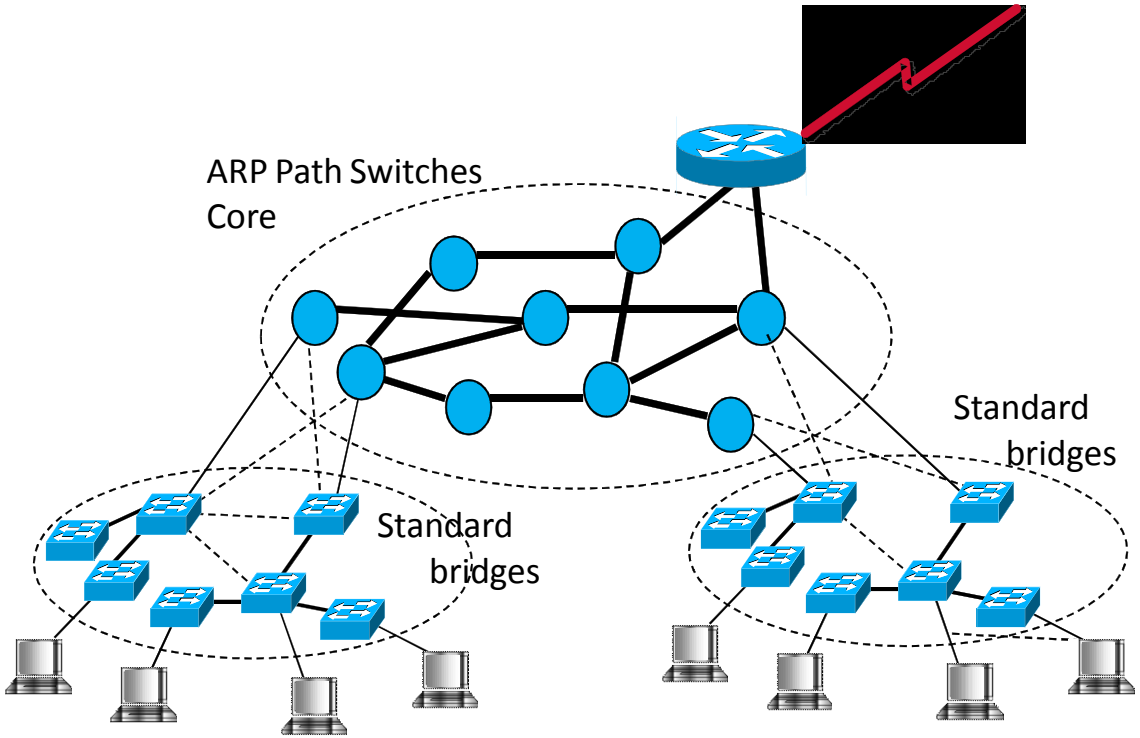
Path repair completed



S *Port locked to S*

D *Port locked to D*

ARP compatibility with standard bridges in core-island mode

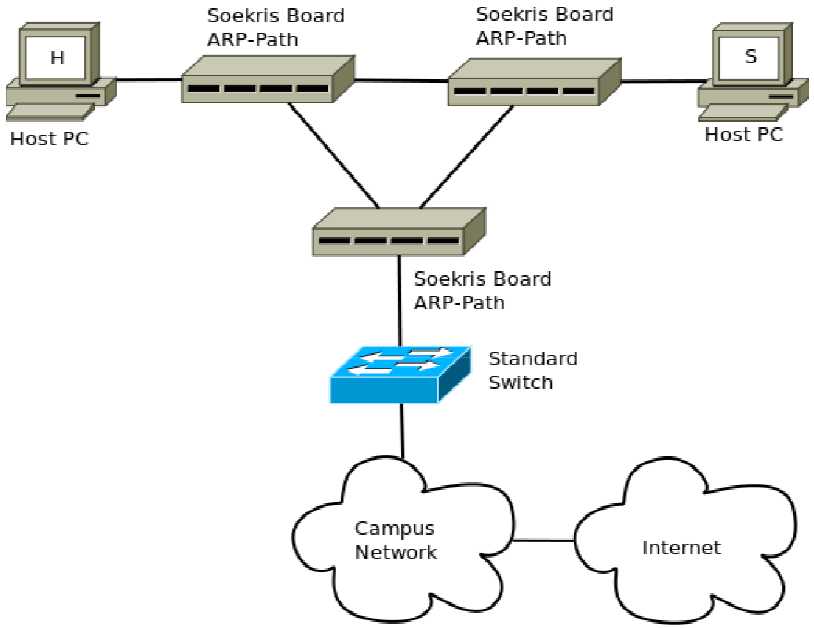


- ARP Path bridges become root of spanning trees of standard bridges (announce a high priority virtual root bridge)
- Islands split in two or more trees if connected with redundant links to core

ARP Path complexity

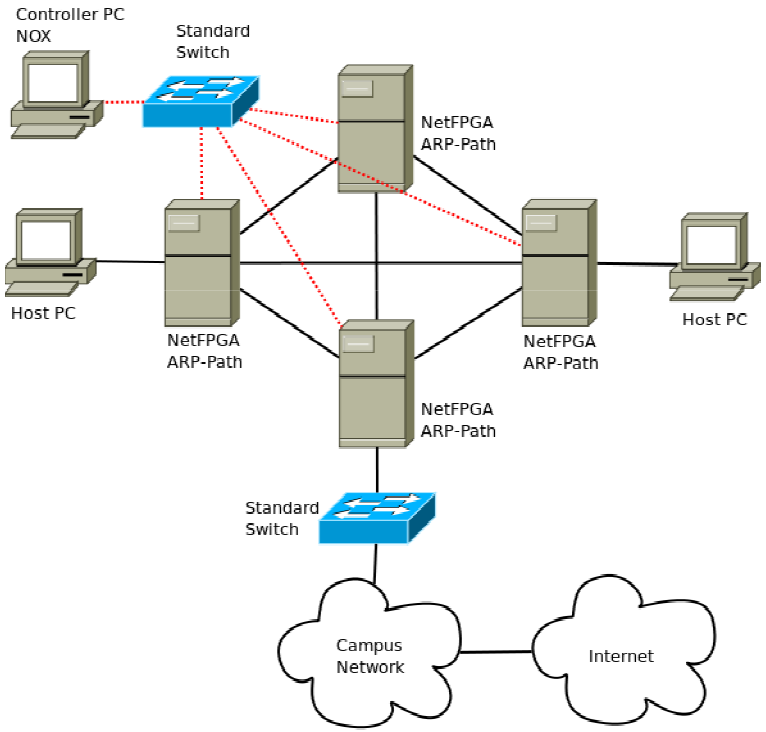
- Stored state is similar to transparent bridges: same number of MACs to learn per port, two persistence timers to process (lock-short, learn-long). Spanning tree protocols not required.
  - Extra packets received on ports not associated to source address must be discarded. Suitable for CAM-based hardware implementations or new ones.
- Reconfiguration and network availability:
  - Only the affected paths being used require path repair.
  - Path diversity (per-host, on-the-fly paths) provides robustness and high network availability.
  - Full MAC address flush at network (like RSTP) is also possible via ARP Path TCNs.

First implementation (Linux) [2]



- On kernel and user space using ebtables [5]
- Functionally simple to code and implement
- All services of campus network operate smoothly (DHCP, video streaming)
- Delays similar to hardware switches (on kernel part)

Second implementation (Openflow/NetFPGA) [2]



- 4 NetFPGA with 4*1 Gbps links
- ARP-Path protocol logic resides at NOX controller (as flow rules to ARP-Path switches)
- Functionally simple to code, implement and modify
- All services of campus network operate smoothly (DHCP, video streaming)
- Delays similar to hardware switches in normal forwarding .
- Robust and fast reconfiguration after link failure.

ARP Path: broadcast

- Extra flooded packets on redundant links: small percentage of the total of links (the highest fraction of links are the non redundant host-switch links).
  - Small fraction of added traffic to redundant links for path set up.
- Reducing ARP messages: implement ARP Proxy function (like Etherproxy [4]) on ARP Path bridges. Proxy implementation requires basically to add an IP field to ARP Path bridge table.
- Frequently used addresses (active servers) remain in ARP proxy caché.

Throughput of simulated paneuropean reference network [1]

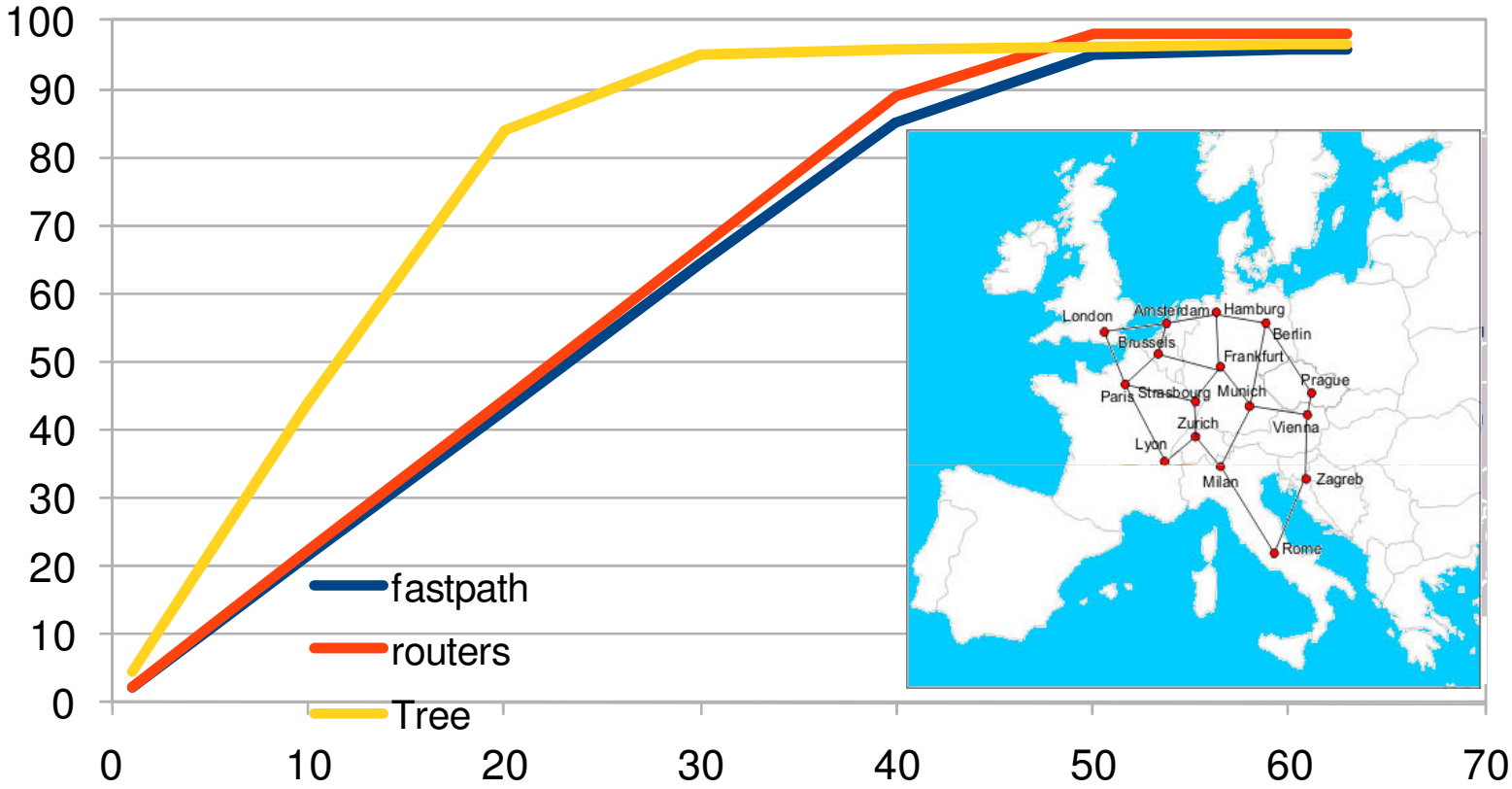
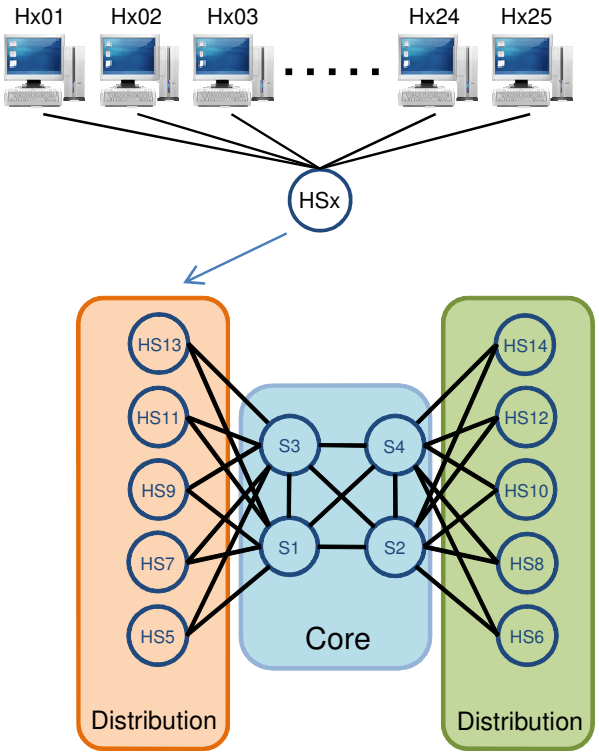
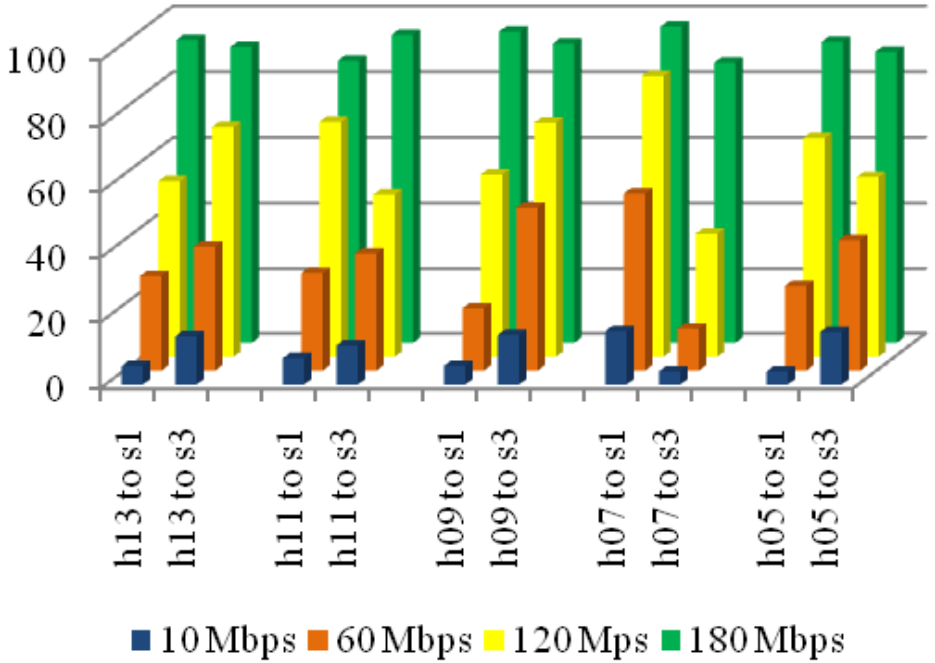


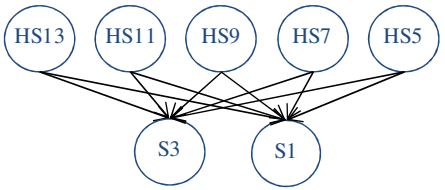
Figure 9. Throughput comparison of pan european network in % of most loaded link versus % of average traffic load applied at the sending host link [1]

Automatic load split between redundant paths of data center [6]

- Two level data center topology, 25*10 hosts
- UDP traffic from hosts on the left to hosts on the right
- Increasing load at hosts to reach link saturation
- Load is distributed among the pairs of links between distribution switch (hs13,hs11,hs9,hs5) and core (s1,s3)







Applicability

- ARP-Path is an on-demand protocol to find low latency paths between hosts.
  - Or between bridges. The same method can also used to set up paths or trees among bridges.
- Sets up low latency source multicast trees
- Applicable to campus, data center and metro networks.
- Can coexist with and combine with other standard mechanisms: e.g. port-based VLANs, RSTP, SPBV,…

Predictability, controllability, manageability

- ARP-Path is *inherently effective:* it finds, with zero added latency, the *best available path at the time it is needed*.
- ARP-Path does not provide a predictable path,
  - Predictability is not essential if reliability and performance are high. The predictability requirement is important when manual configuration is decisive and configuration errors may stop service. Some predictability features can be devised and essential controllability is kept (bridges/ports not allowed to execute ARP-Path, etc)
  - A zero configuration protocol needs “freedom” to provide high availability and performance.
  - ARP-Path is somehow “autonomic”: it finds paths and balances load according to the link loads. Autonomic protocols need autonomy.

Predictability, controllability, manageability

- Manageability: Bridges and ports can be included or excluded from the ARP Path protocol via SNMP.
- Can coexist with spanning tree protocols (separation by VLANs)
- Compatible with standard Connectivity Fault Management mechanisms.

Conclusion

- ARP-Path shows the feasibility and potential of the conceptual evolution of the Transparent Bridge concept by modifying the bridging mechanisms
- Loop free operation without ancillary mechanisms
- Similar performance to shortest path routing in terms of throughput, but lower average latency, with lower complexity.
- Native load distribution capability (path diversity), as a result of the on-demand, per host, path selection based on lowest forward latency.
- Message overhead: Extra broadcast replicas at redundant links that are automatically discarded by receiving port. Low percentage of total network links.
- ARP path broadcast reduction to hosts requires, as for other Advanced Ethernet proposals in medium large networks, an ARP Proxy function or centralized or distributed (DHT) host resolution.
- ARP proxying adds little complexity to ARP-Path switches (IP info).
- Compatible with IEEE 802.1 Bridges in core-island mode. Transparent to hosts and routers.
- Compatible and consistent with IEEE 802.1D, 802.1Q, Q-in-Q and Mac-in-Mac schemes

References

[1] Fast Path Ethernet Switching: On-demand, Efficient Transparent Bridges for Data Center and Campus Networks. Guillermo Ibanez, Juan A. Carral, Alberto García-Martínez, José M. Arco, Diego Rivera Pinto, Arturo Azcorra. IEEE LANMAN Workshop. May 2010. http://dspace.uah.es/jspui/bitstream/10017/6298/7/FastpathLANMANcamerareadyv5final.pdf

[2] A Simple, Zero Configuration Low Latency Protocol penflow/NetFPGA. Guillermo Ibáñez, Jad Naous, Elisa Rojas, Diego Rivera, Juan A. Carral, José M. Arco. Demo at Conference on Local Computer Networks. October 2010. *Best demo award*. http://dspace.uah.es/jspui/handle/10017/6770

[3]Fast Path bridges: Old and new ideas for the evolution of transparent bridges. (FYI) IEEE 802.1 Interim Meeting Sept 2009. *Primitive protocol proposal that used Up/Down.*

[4] EtherProxy: Scaling Ethernet By Suppressing Broadcast Traffic. Khaled Elmeleegy, Alan L. Cox INFOCOM 2009: 1584-1592

[5] Ebtables. http://ebtables.sourceforge.net

[6] Simulation results of ARP-Path load distribution. http://hdl.handle.net/10017/7829