Ethernet Technology Summit

Shortest Path Bridges Without Routing (All-path)

(beyond link-state routing in shortest path bridges)

***Exploring Paths instead of Computing Paths** in Data Centers, Enterprise and Audio Video Bridges*

Guillermo Ibanez (Ph.D). Elisa Rojas. (Ph.D. st.) Universidad de Alcalá. Madrid. Spain

Santa Clara, CA USA
April 2013

EUROPEAN UNION
European Social Fund

La Suma de Todos
CONSEJERÍA DE EDUCACIÓN
Comunidad de Madrid
www.madrid.org

GIST
Grupo de Ingeniería de Servicios Telemáticos

Universidad de Alcalá

Outline

- Our Research Statement (*Switches shall not route*)
- A bit of history. From transparent bridges to shortest path bridges. *(mixing up again layers 2 and 3)*
- Path Exploration versus Path Computation (SPB)
- All-Path family of protocols
  - ARP-Path, Flow-Path, Bridge-Path, Path-Moose
  - Results *(native load distribution, lower latencies)*
  - Openflow **and** All-path
- Publications and Implementations
- Conclusion

Our Research Statement

- Develop Advanced Ethernet Switched Networks
  - Simplicity as a requirement for scalability
  - Architectural consistency with bridges.
    - No Router inside the Switch
      - Avoid “hybrids” as much as possible
    - Coherence with existing transparent bridge mechanisms
  - Core-island compatibility with standard bridges is sufficient
    - Like RSTP and 802.1 protocols (point to point links required)
    - Full miscibility of advanced and new bridges adds too much complexity and departs from Ethernet (e.g. TRILL header)

Our Research Statement

- Applicable to Wired and Wireless Networks
  - But 802.1 and 802.11 architectures are too different!
    - Access Points functionality has little relation with bridge architecture
  - Now, integration is ongoing at IEEE (Klein's BSS Bridging 2012)
    - 802.1 BSS Bridging. Seamless coexistence of 802.1 and 802.11 networks.
    - The whole BSS is modeled as a distributed bridge overlaying the 802.11 protocol
    - Access Point acts as the Bridge's Control Plane
    - Stations act as Bridge Ports
  - Protocols for wired switches will likely run on hybrid networks without modification (e.g. All-path)

http://www.ieee802.org/1/files/public/docs2012/avb-phkl-80211-bss-bridging-0512-v1.pdf

Our main contributions in Advanced Ethernet Switches (2004-2013)

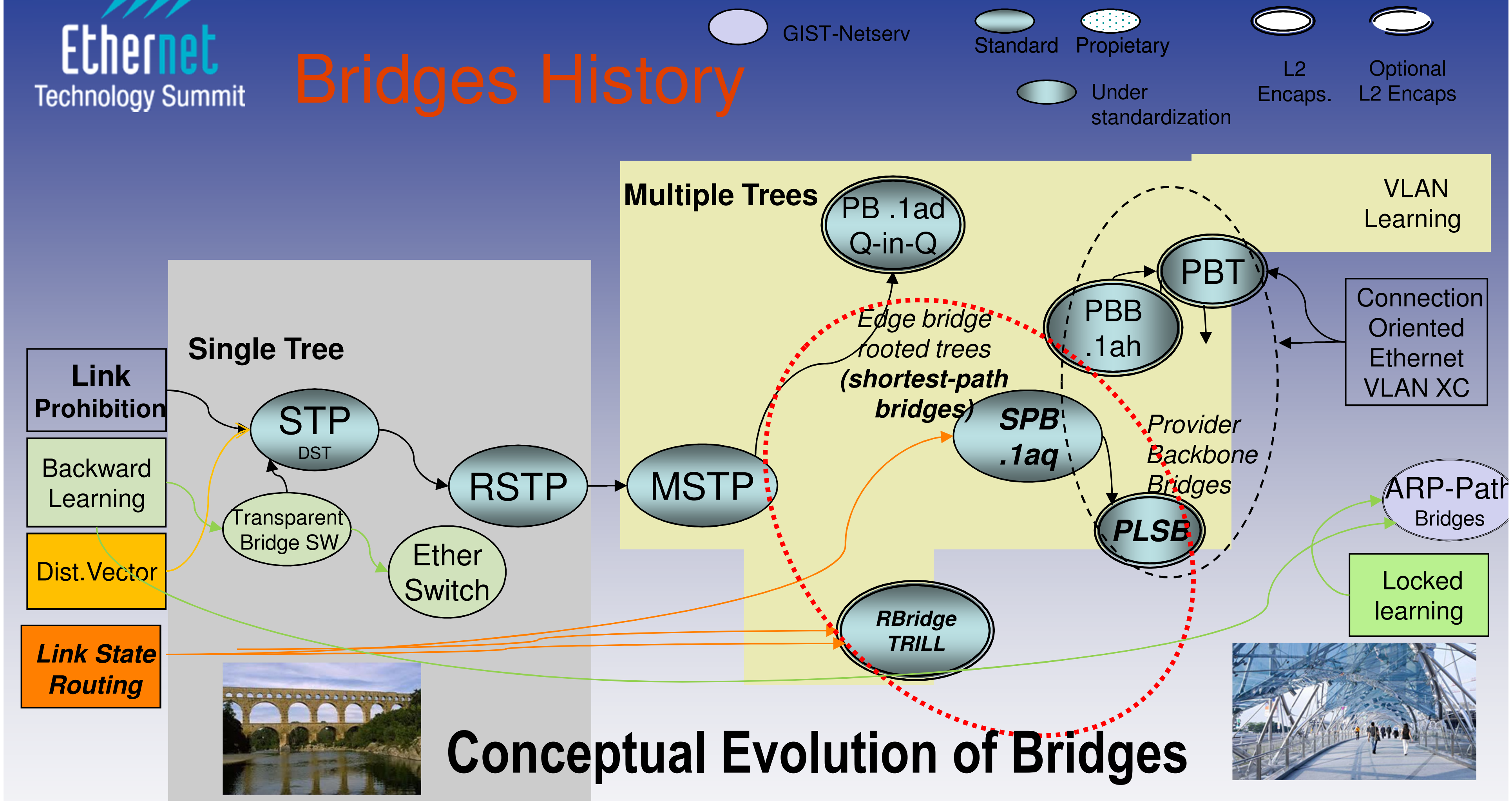
- AMSTP (2004)/Abridges. IEEE LCN 2004.
  - Supersimplified self configuring MSTP protocol.
- All-Path protocols family 2010-2013
  - *This* presentation
- Torii-HLMAC (2011-2013). (Improvement and generalization of Portland for Data Center)
  - *Presented at Forum 1A by Elisa Rojas.*
    - *Also without link-state routing*

Our approach to Shortest Path Bridges

- Look for **Pure Bridging Mechanisms**
  - *Bridging = Flooding frames + address learning*
  - Loop prevention needed (active topology of tree/s)
- If a broadcast frame is flooded over **All** links
  - We can find the shortest path to destination easily
- How do we prevent loops?
  - **Lock** source address learnt to the port of first arrival.
  - **Discard** all frames with this MAC source address received at other ports.
  - It is a kind of Reverse Path Forwarding at layer two

Ethernet Technology Summit

Bridges History

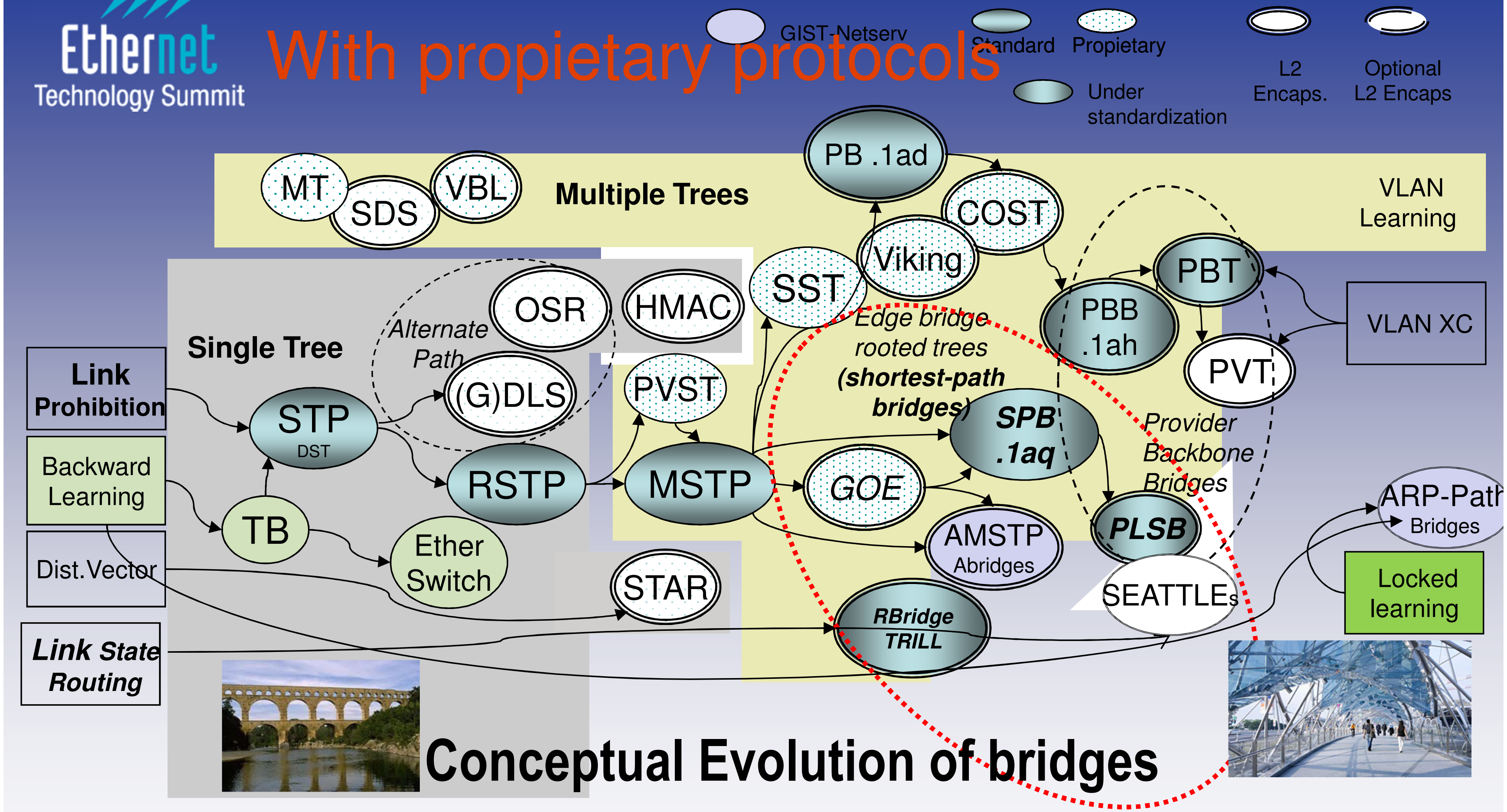


From single to multiple spanning tree protocols, then link-state computation of shortest path trees

1984 1991 1994 1998 2000 2002 2003 2004 2005 2006 2007 2008

Ethernet Technology Summit

With propietary protocols



From single to multiple spanning tree protocols, then link-state build of shortest path trees

1984 1991 1994 1998 2000 2002 2003 2004 2005 2006 2007 2008

Shortest Path Bridging history

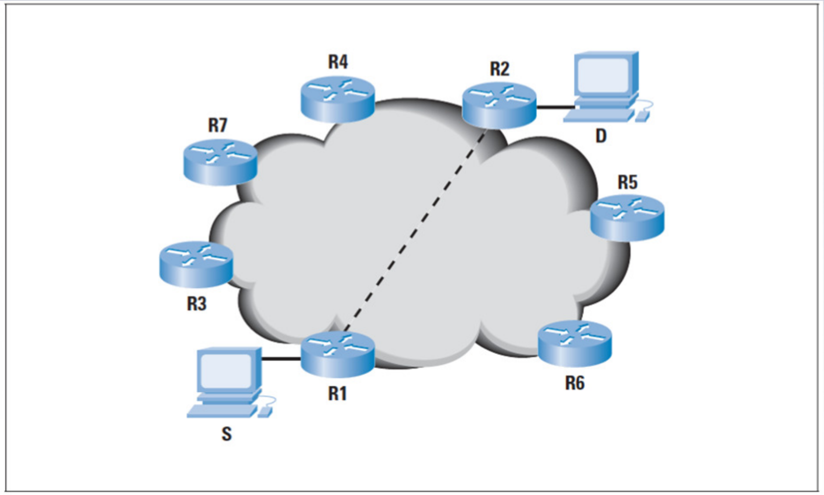
- How did Shortest Path Bridges start?
  - Need for single IP subnet at campus networks
    - Avoid IP addresses administration (even with DHCP)
    - Key for Data Centers: virtual servers move frequently
  - RSTP blocks links, MSTP is too complex to configure
- "Link-state routing protocols are fast and proven"
- Then: "Let us use them also at layer 2"
- Implicit assumption: ***" Bridges will never be as good as hybrids of bridge and router"*** ***(… unless you think outside the box!)***

Shortest Path Bridging

- Current situation: Two competing standards
- But **both use the Link-state Routing** paradigm for Shortest Path Bridging
  - – IEEE 802.1aq SPB Standard uses a IS-IS L2 variant
  - – TRILL Rbridges (IETF) also use IS-IS Layer 2 variant
- Problems
  - SPB 802.1aq focused on provider networks (tagging)
    - – Not focused in data centers and enterprise networks
    - – Interest seen on SPBM (MAC in MAC), but not on SPBV
  - TRILL is not a fully satisfying alternative
    - – Simplicity was lost to get full miscibility with std bridges(IMHO)
    - – Misalignment with existing ASICs and IEEE 802.1

Enterprise networks

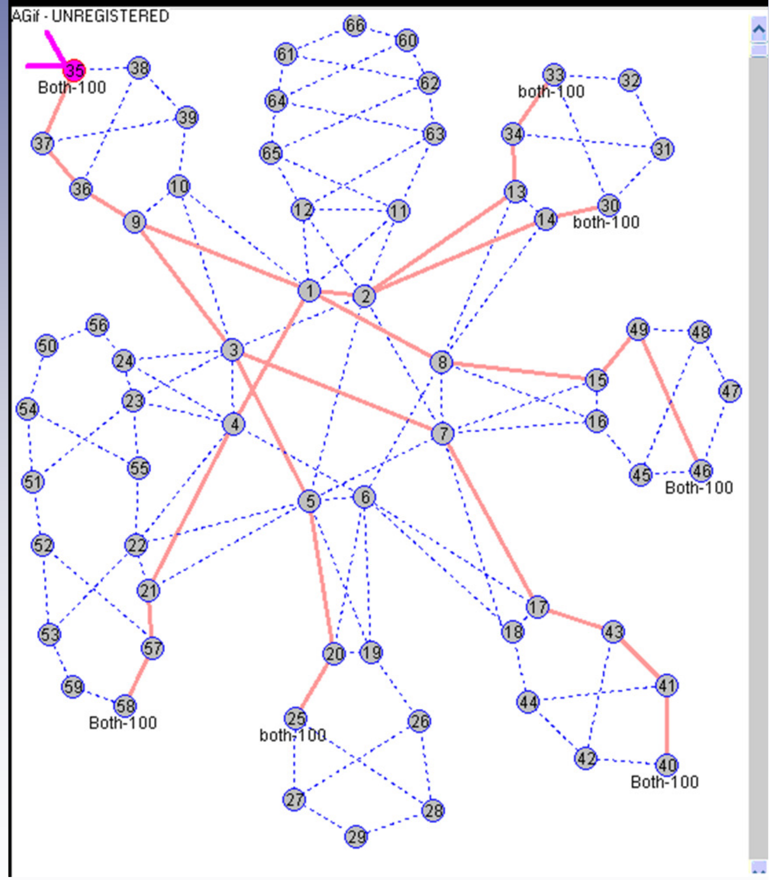
- Transparent Interconnection of Lots of Links (TRILL) – Routing Bridges (RBridges)
  - TRILL header (modified at each hop)
  - IS-IS
  - Layer 2 and ½
- Manufacturer support?



Ethernet Technology Summit

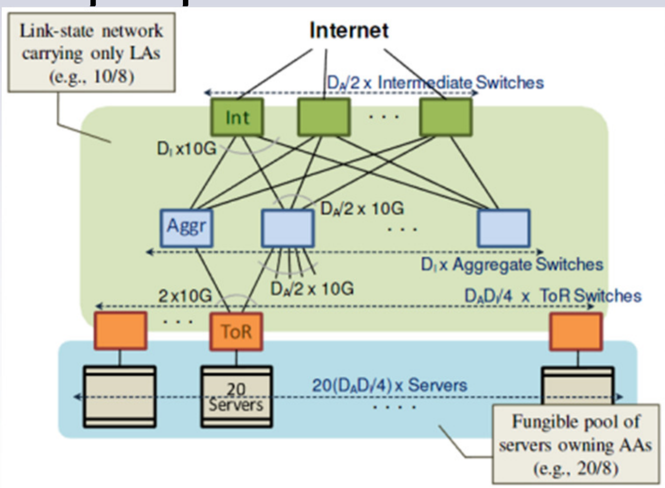
Enterprise networks

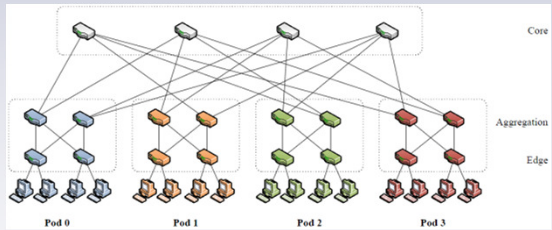
- Shortest Path Bridging (SPB)
  - Path congruency as a must
    - – Needed for loop prevention
  - IS-IS modified
    - – Path Vector
    - – Complexity escalates
      - – With multipath (N**3) and congruency requirements
    - – ECMT
  - SPBM/SPBV

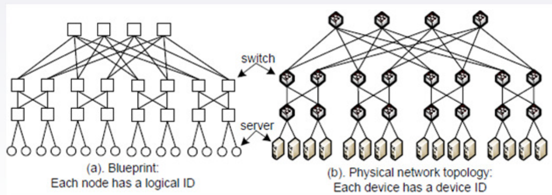


Data center networks

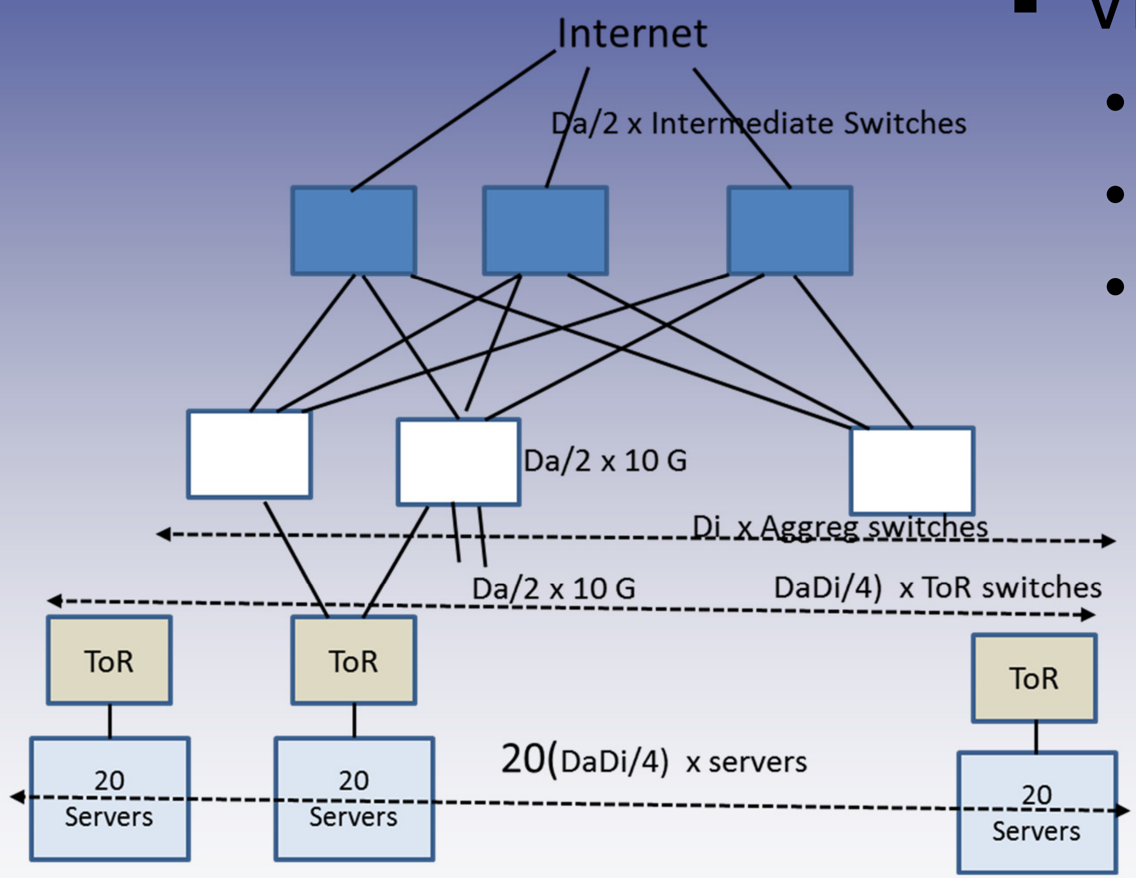
- **Data center** networks are increasingly relying on **Ethernet** and flat layer two networks
  - Due to its excellent price, performance ratio and configuration convenience
- Recent architecture proposals:
  - **VL2**
  - **PortLand**
  - **DAC**
    - Blueprint





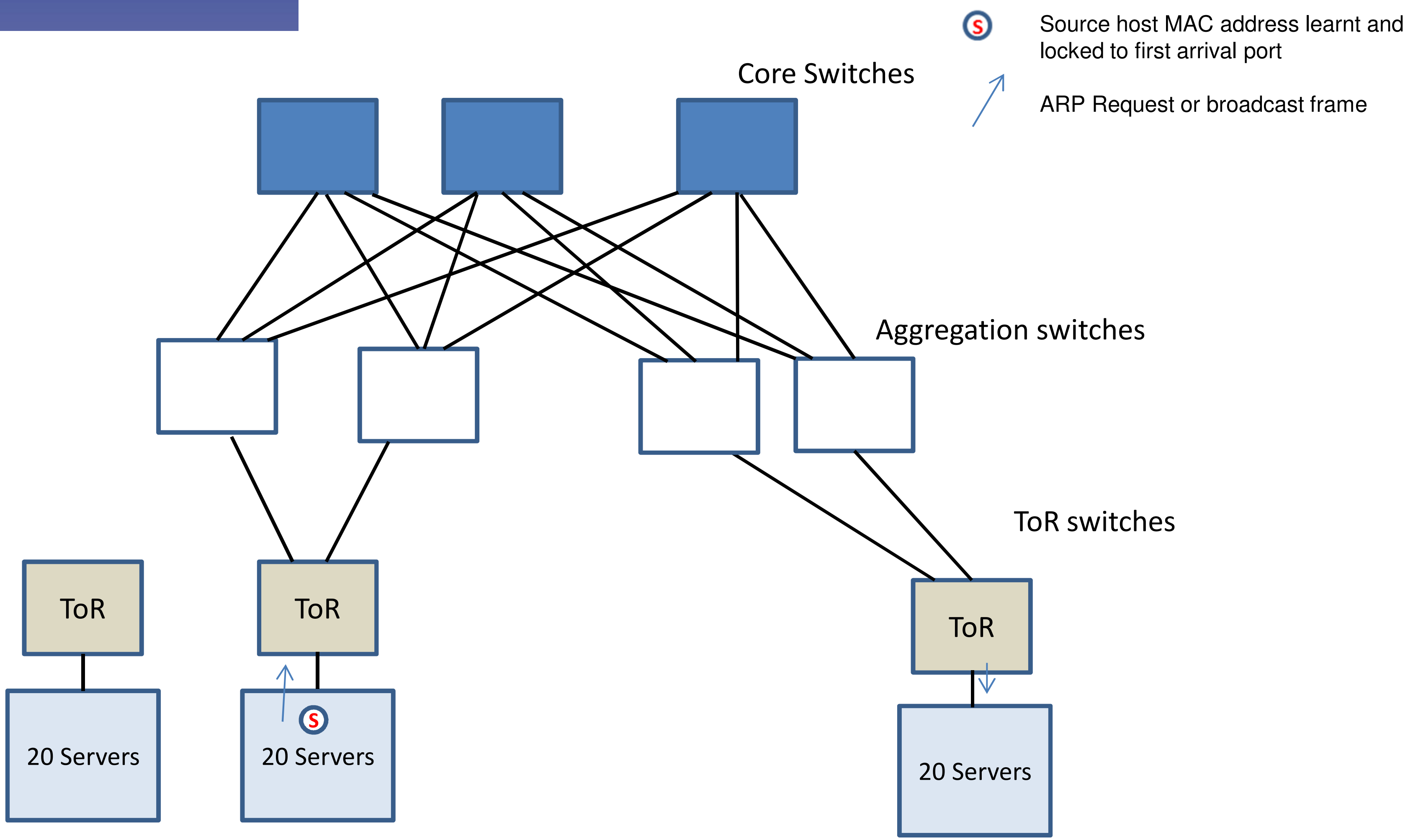


Data Center network example:VL2

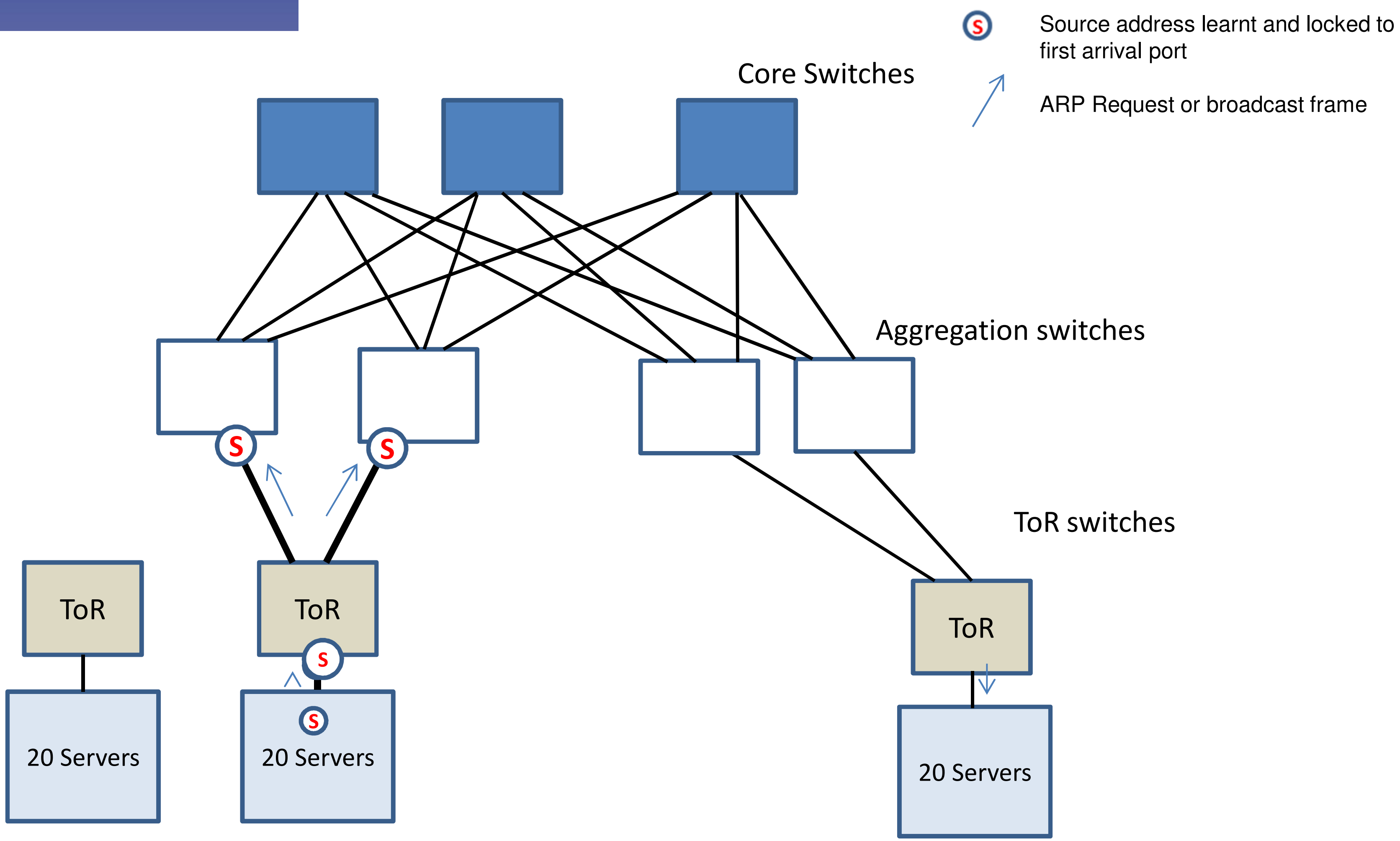


- VL2 folded Clos
  - Commodity switches
  - Valiant Load Balancing
  - Distributed directory
    - Manages IPs

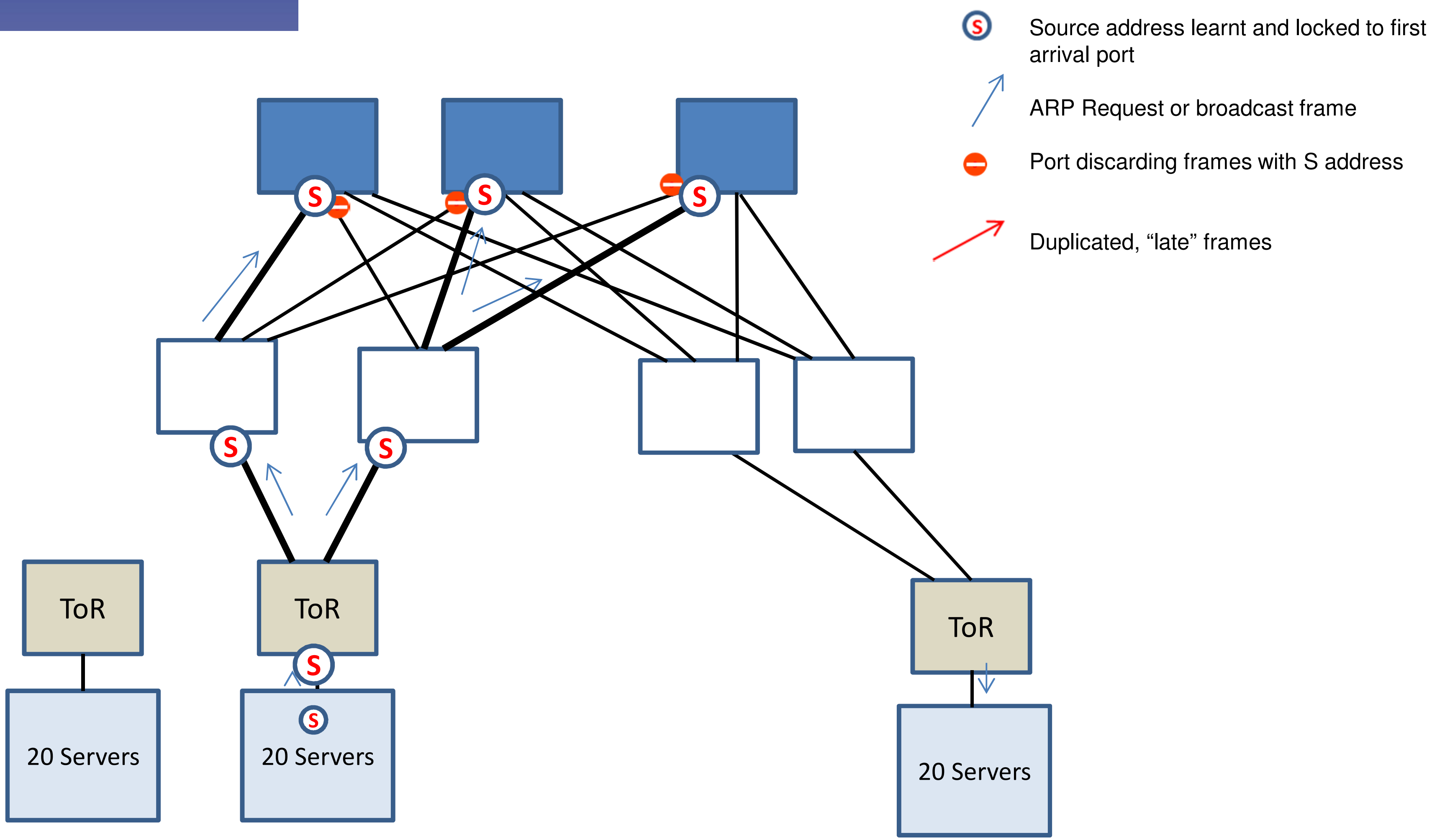
Path Exploration without loops in a VL2 Data Center network



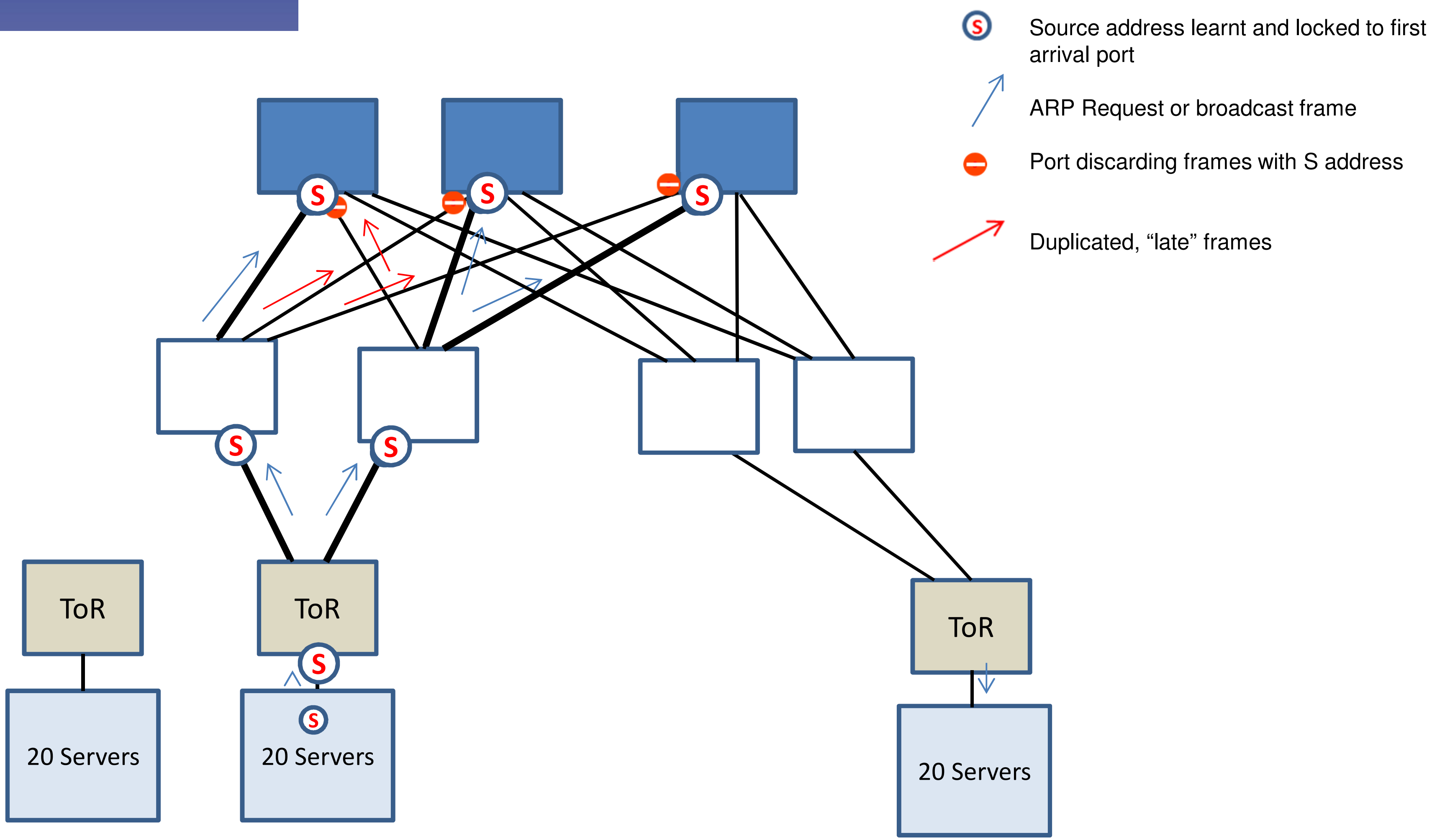
Path Exploration without loops in a VL2 Data Center network



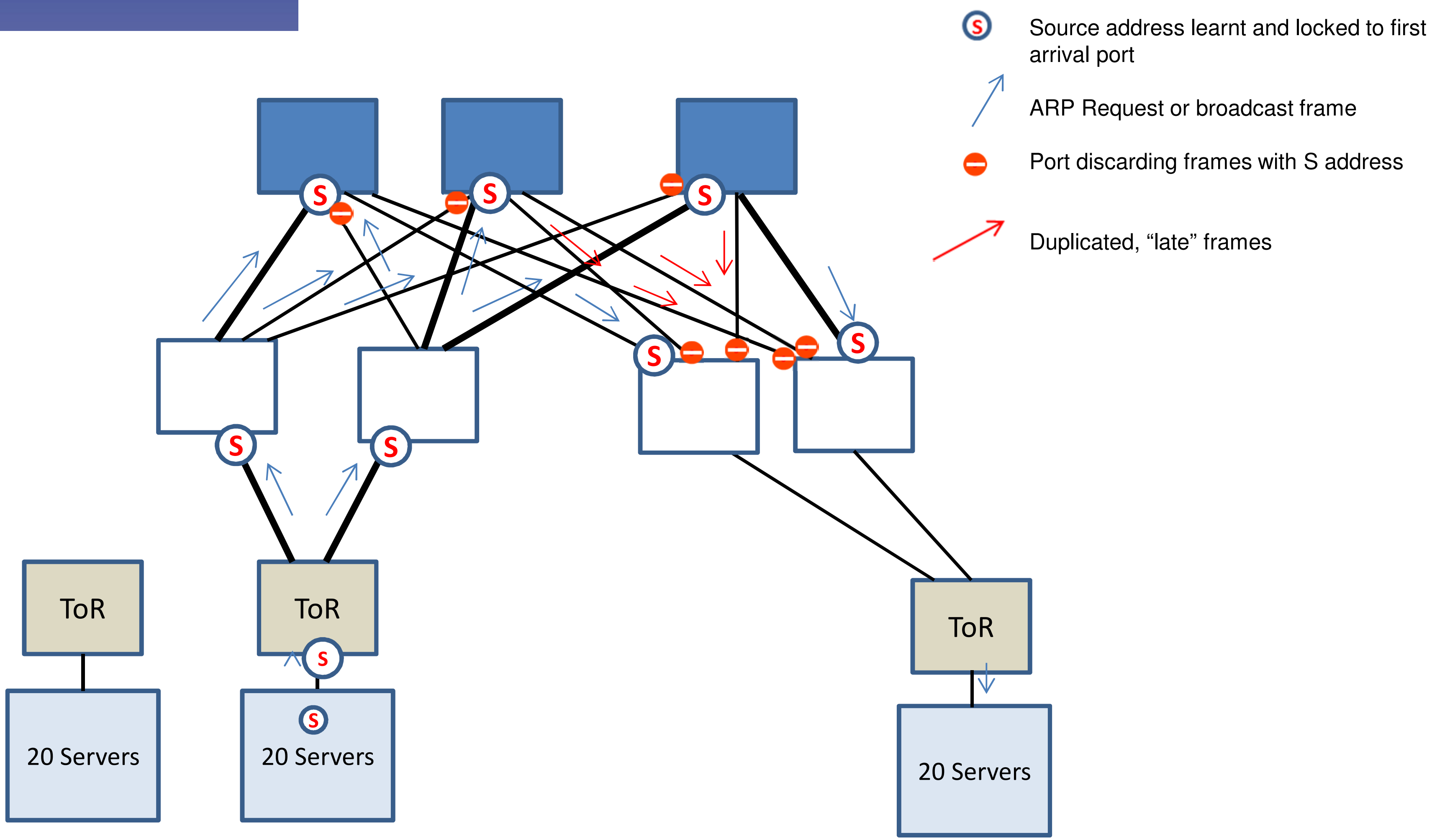
Path Exploration without loops in a VL2 Data Center network



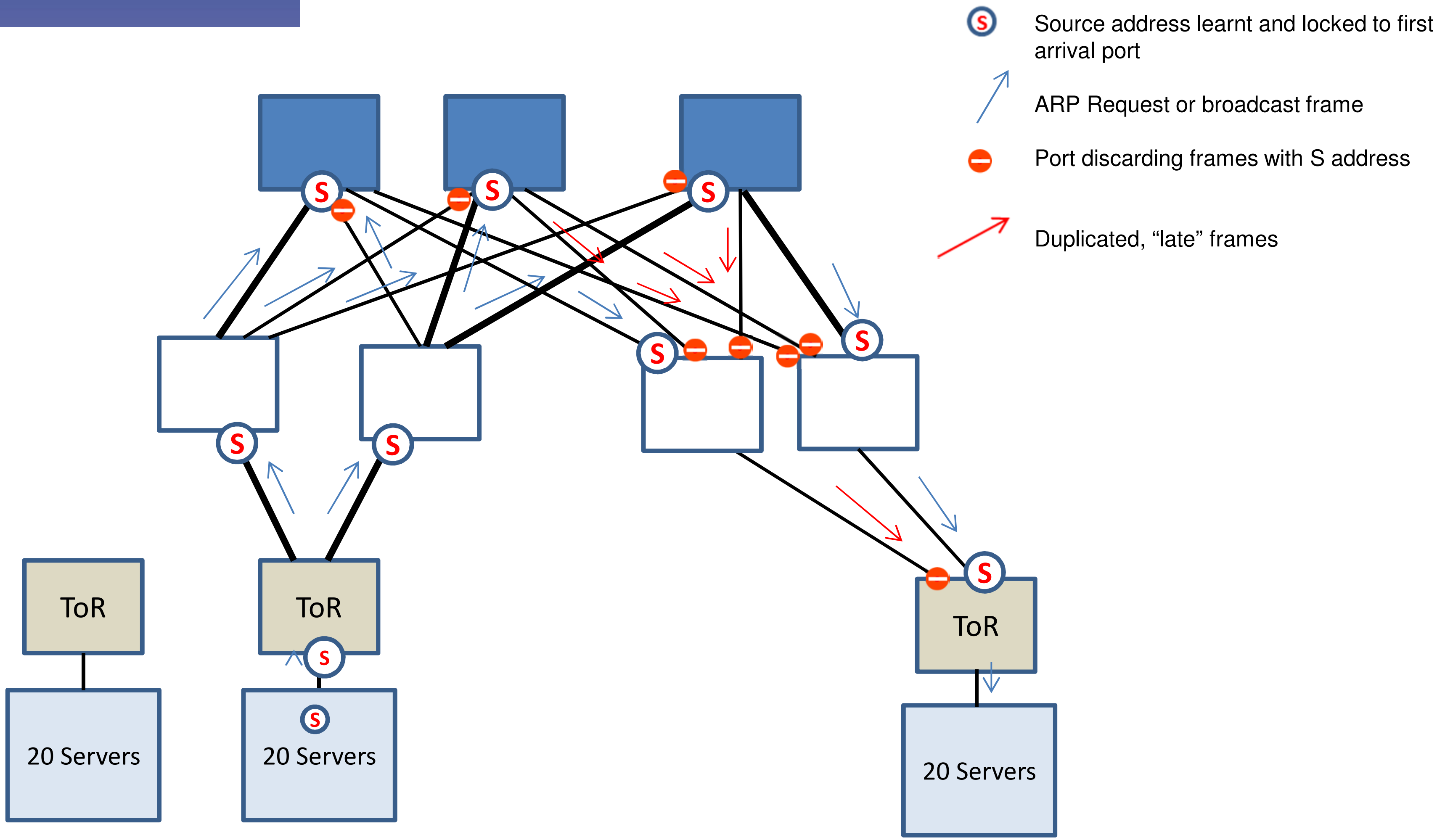
Path Exploration without loops in a VL2 Data Center network



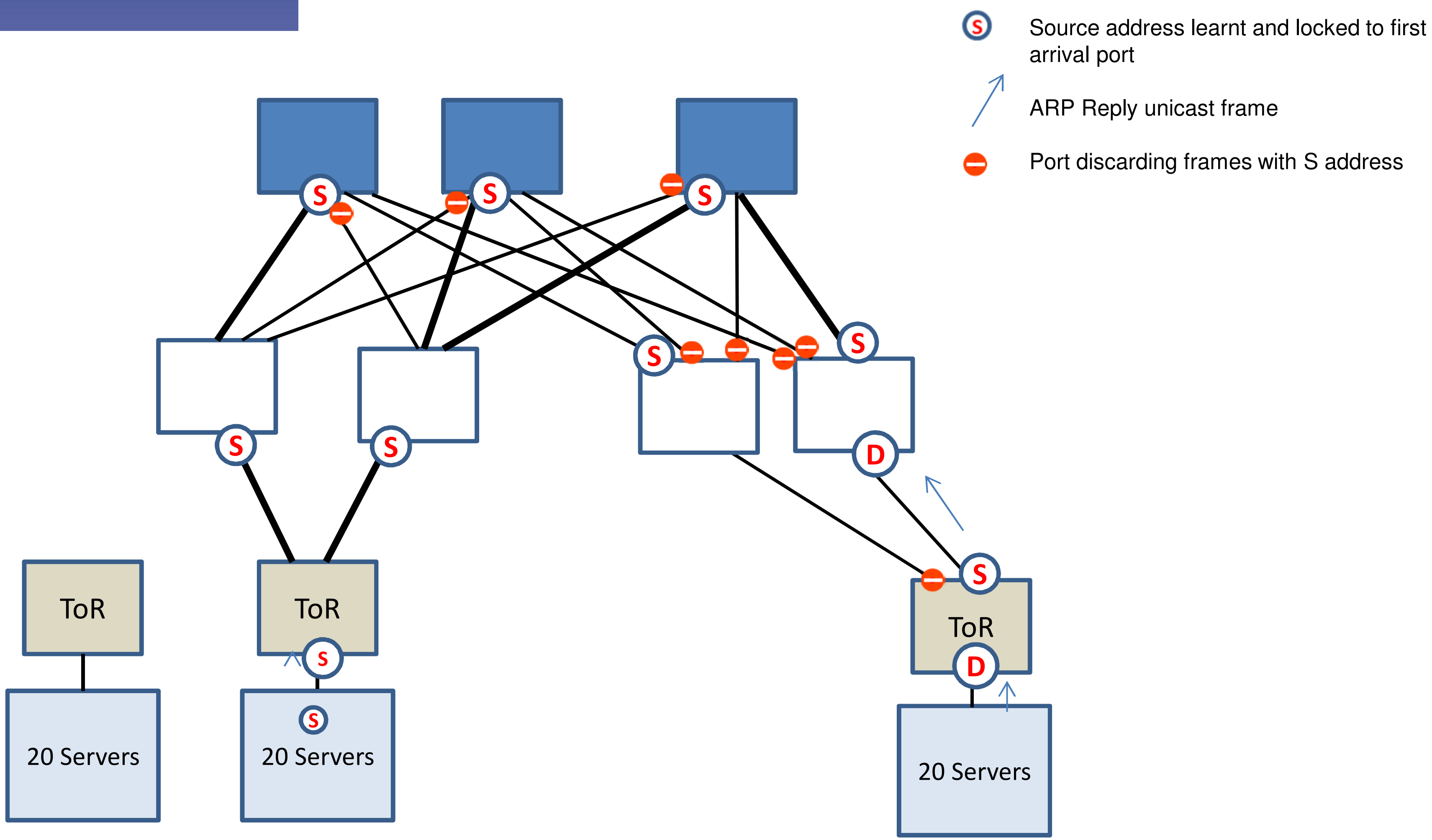
Path Exploration without loops in a VL2 Data Center network



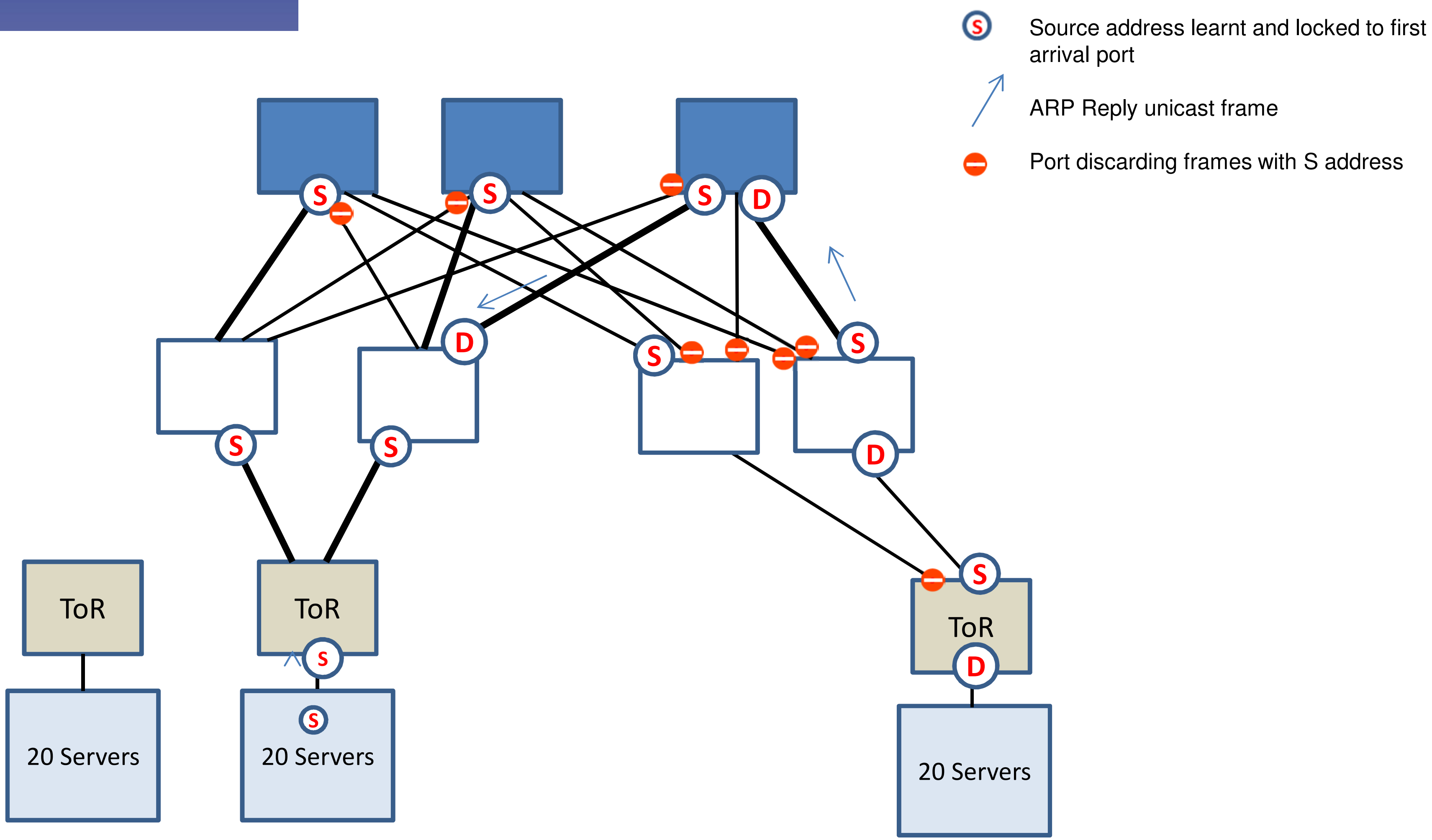
Path Exploration without loops in a VL2 Data Center network



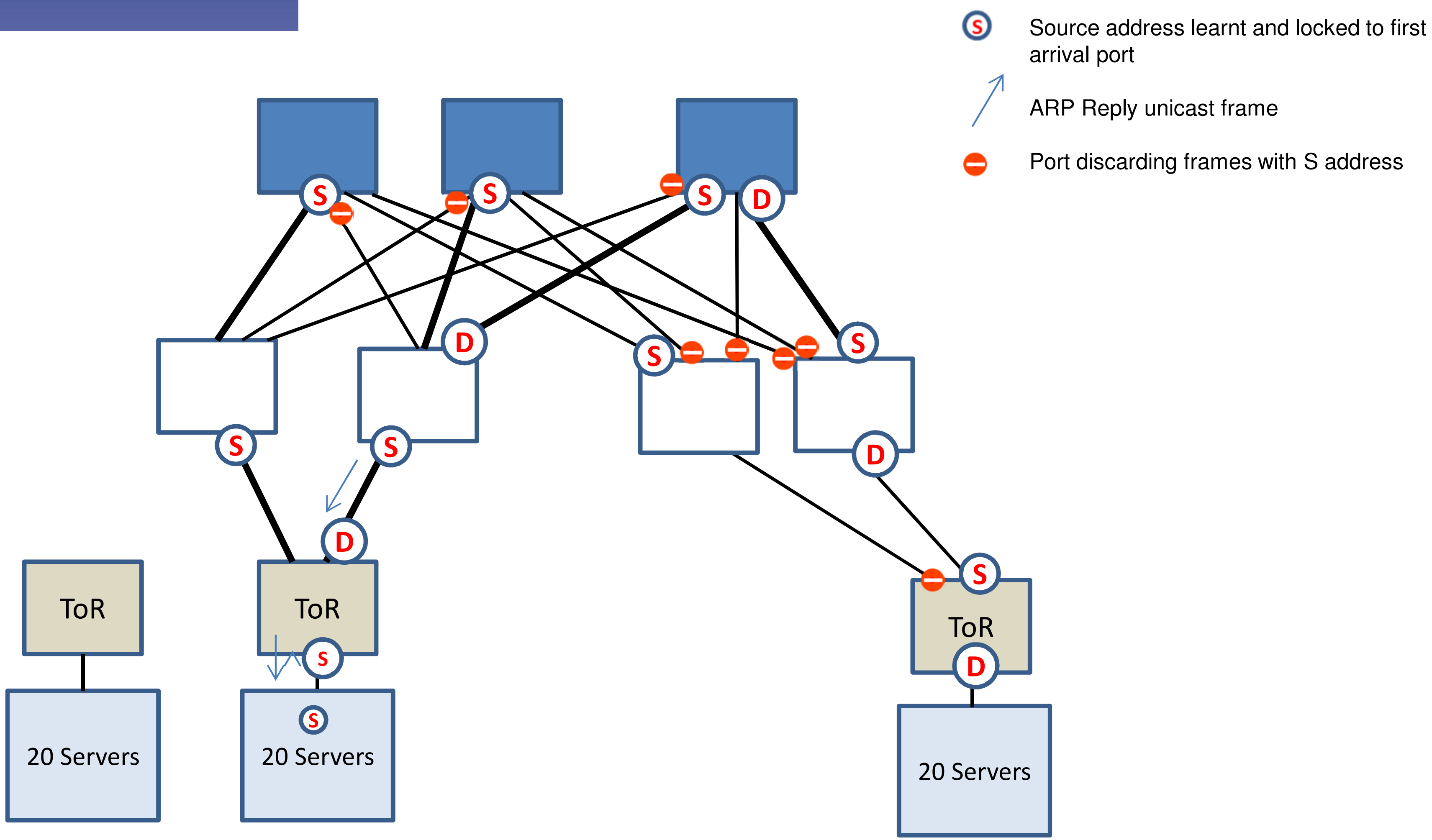
Path Exploration without loops in a VL2 Data Center network



Path Exploration without loops in a VL2 Data Center network

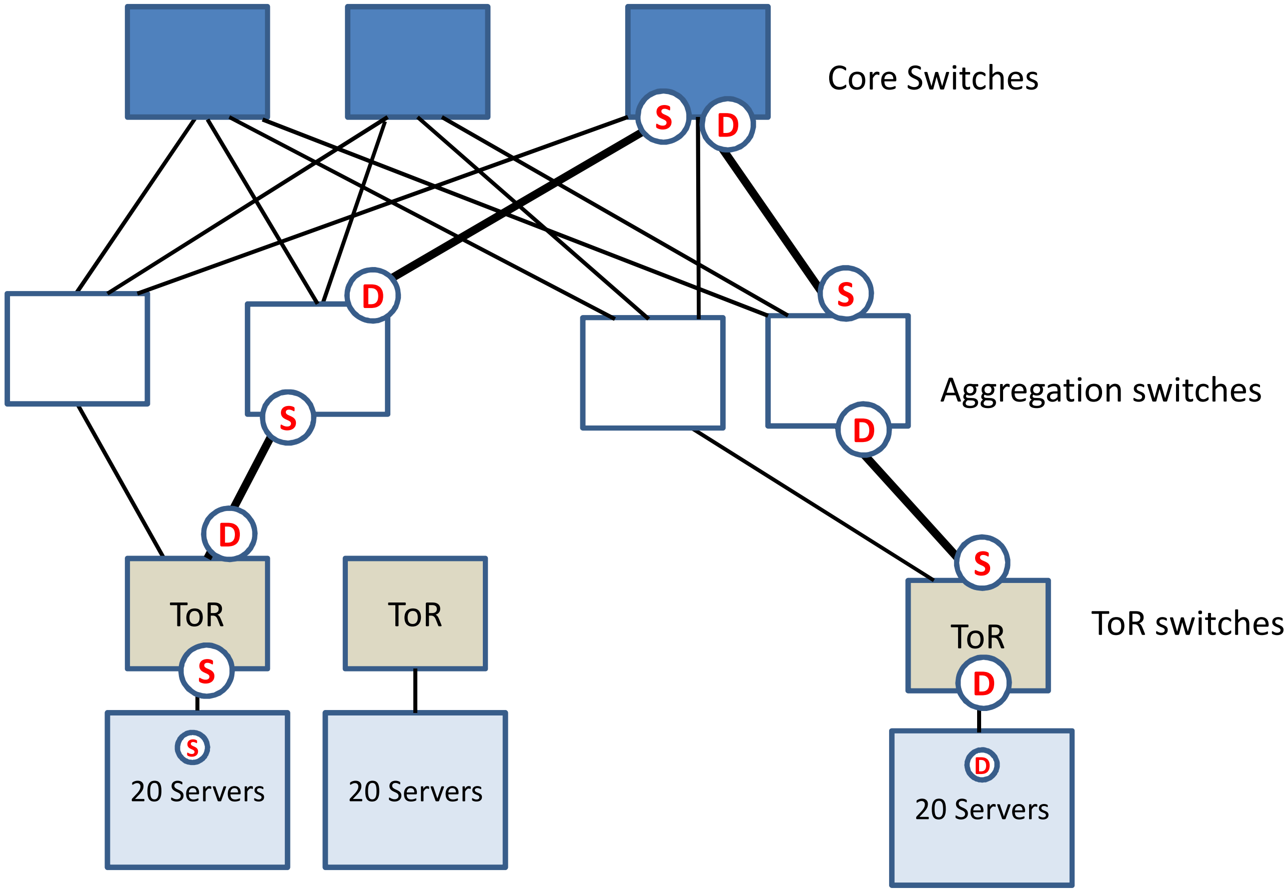


Path Exploration without loops in a VL2 Data Center network



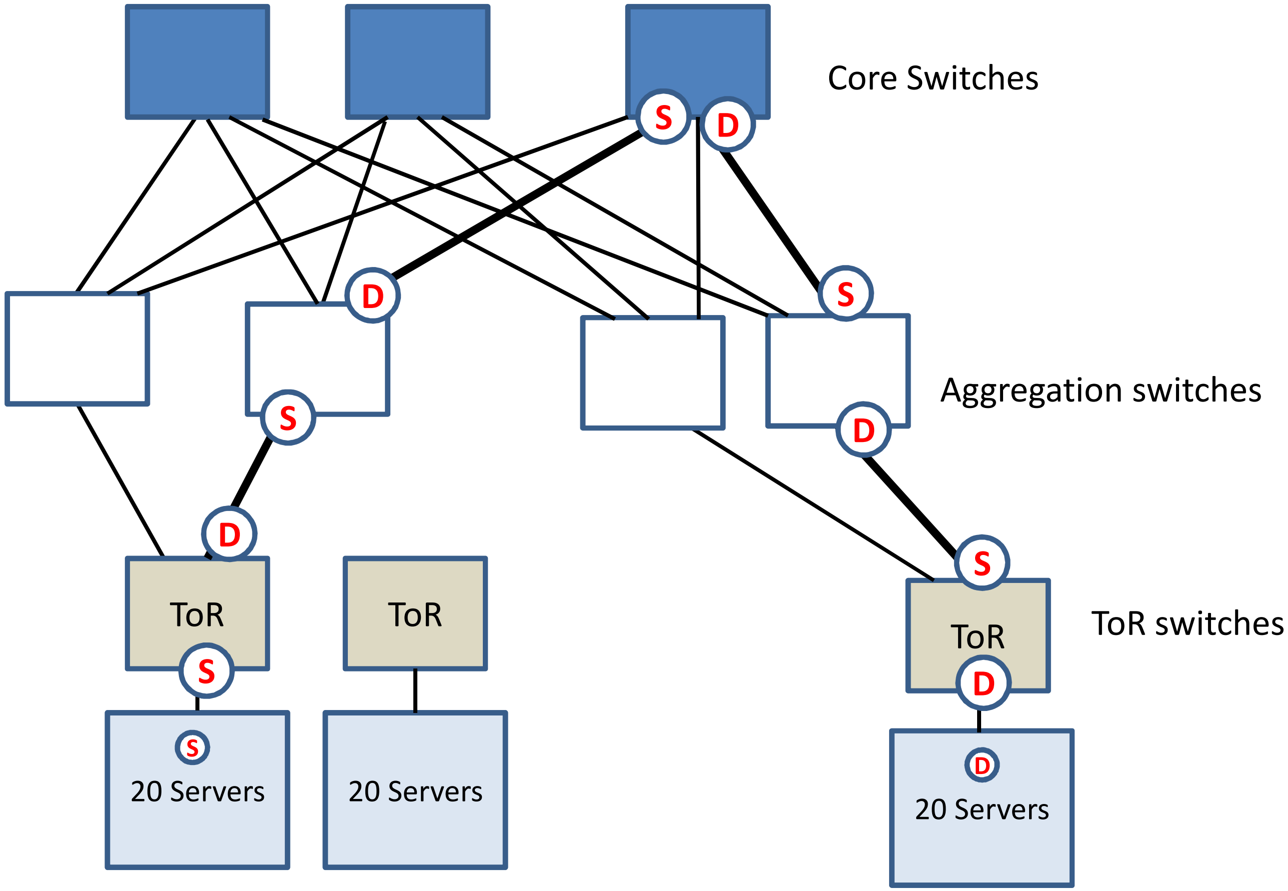
Path Exploration without loops in a VL2 Data Center network

Addresses not refreshed expire at unused branches
Path completed in both directions



Path Exploration without loops in a VL2 Data Center network

Addresses not refreshed expire at unused branches
Path completed in both directions



Path Computation vs Path Exploration

- Complex path computation
  - powerful processors avail.
- Path diversity (ECMP) increases complexity

- Multicast complexity
- VLAN IDs linked to trees

- Simple path discovery
  - Flooding and learning
- Native path diversity
  - Load adaptive routing
- Varied path granularity
  - Per host, per flow, per bridge.
- Simple multicast
  - Instantaneous building of source rooted trees
  - Straightforward IGMP snooping
- VLAN, tag independent

Path Computation vs Path Exploration

	Link State (SPB)	All Path
Forwarding state (CAM)	*O(b+h)*	*O(h)*
Routing state	*O(b*d+h)*	≤ *O(h)*
Number of messages	*O(b*E)*	Standard ARP messages + extra flood: *h*(E-N+1)*
Computational complexity	*O(b*log(b) +h)*	One CAM look-up (MAC, port)
Convergence time	*O(path length bs)*	*Negligible (extra processing of ARP at ARP Path bridges)*
Fault recovery	Messages O(2*E) Time *O(path length bs)* Recompute *O(b*log(b))*	Messages *O(2*E*hostlink)* Time *O(path length bs)* Recompute *O(b)*
Path diversity	*O(b*b*log(b))*	*O(b)*

All-Path protocols family

- ARP-Path
  - The initial protocol: host-oriented. Per host, on-demand paths. Learns source (SA) MAC addresses.
- Flow-Path
  - Follows the basics of ARP-Path but it's flow-oriented
    - Learns SA-DA address tuples
- Bridge-Path (& Path-Moose)
  - Instead of building paths between hosts (like ARP-Path and Flow-Path), they create paths/trees between edge bridges for added scalability

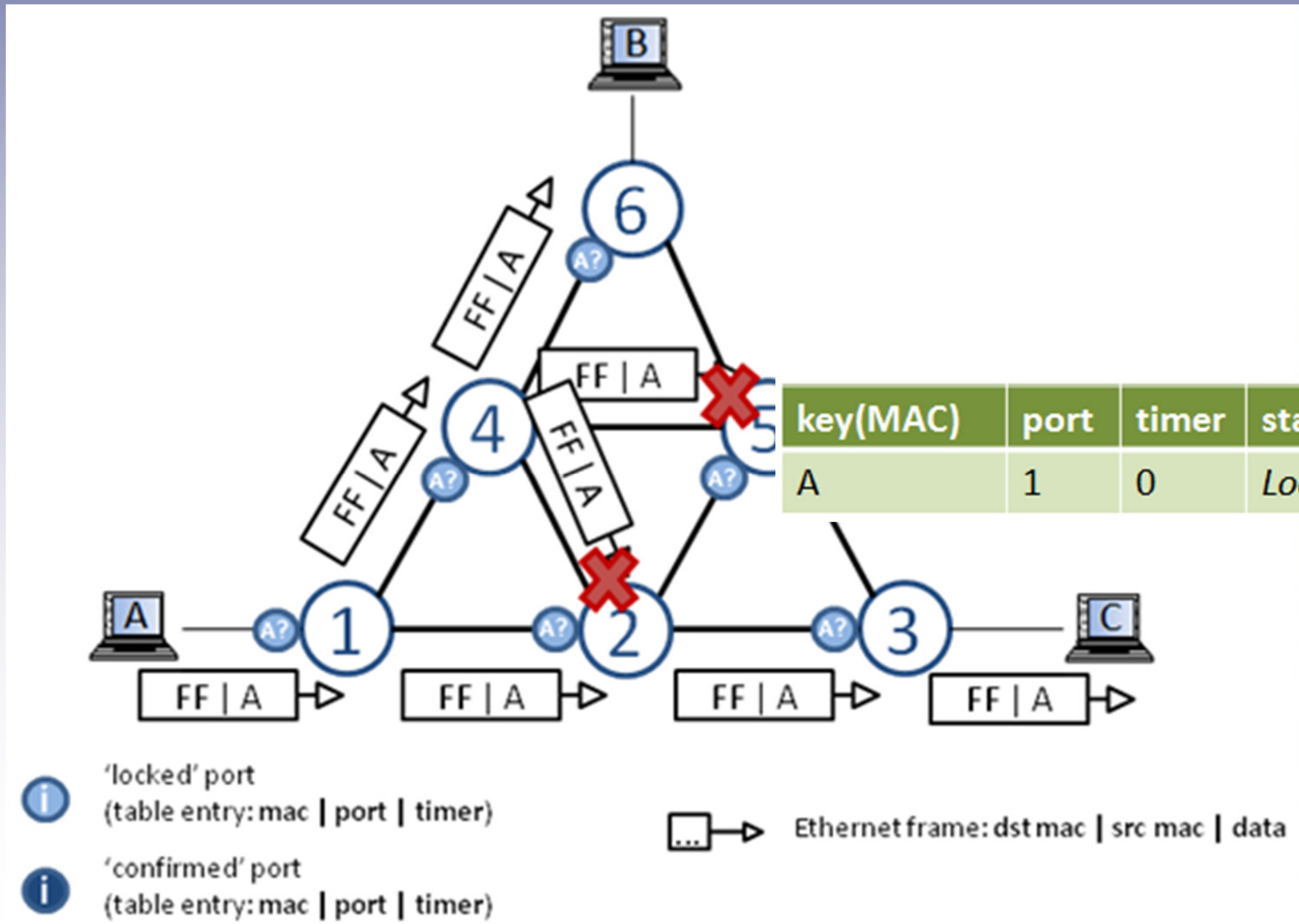
ARP-Path

- Features:
  - Zero configuration
  - Source address learning (SA)
  - Low latency paths
  - Load balancing
  - Pure bridging protocol → learns by snooping broadcast messages
  - Path symmetry not guaranteed

How does ARP-Path work ? (video)

- *(Click on link below to see the video)* https://www.youtube.com/watch?v=IhwCYAu_E7E

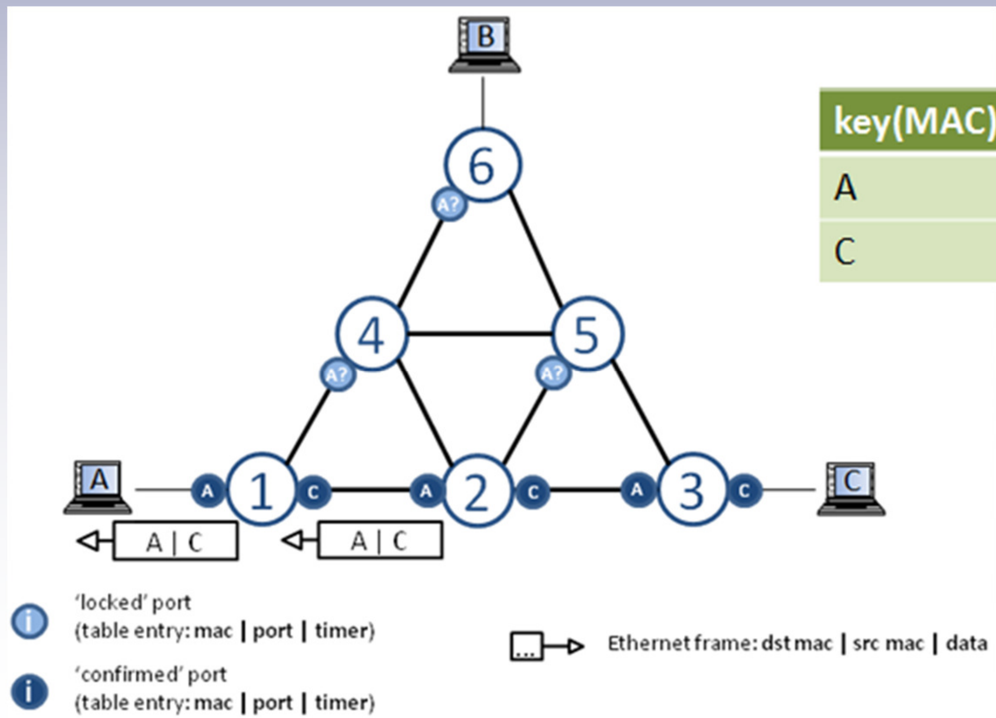
ARP-Path 1.0 (aka FastPath) Path discovery (ARP Request)



key(MAC)	port	timer	state
A	1	0	*Locked*

ARP-Path

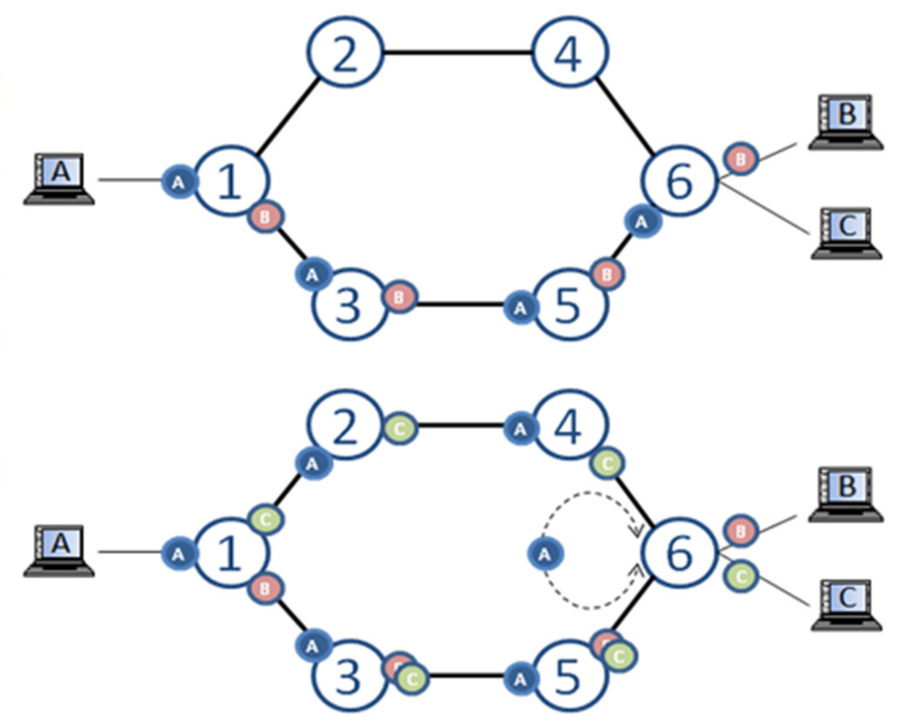
- ARP-Path 1.0 (FastPath)
  - Path confirmation (ARP Reply)



key(MAC)	port	timer	state
A	1	0	*Confirmed*
C	2	0	*Confirmed*

ARP-Path

- ARP-Path 1.0 (FastPath) has some issues
  - Backwards learning refresh → Path symmetry needed
  - Some paths oscillate
  - Confirmation → Repair flag

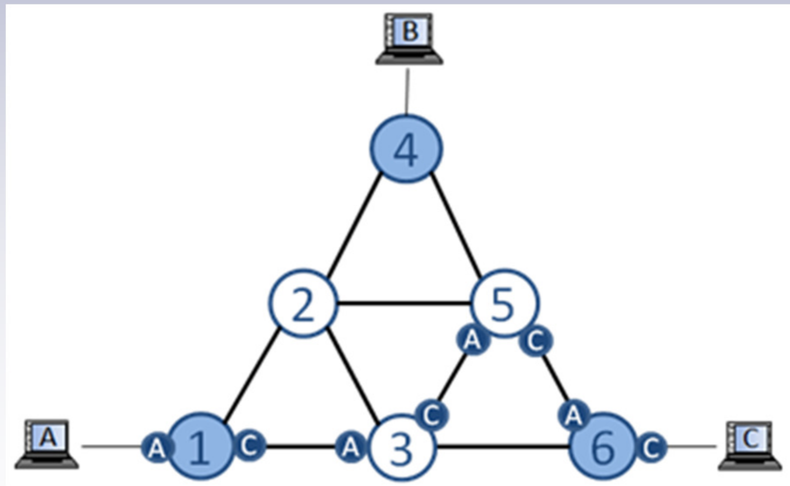


ARP-Path improved version

- ARP-Path 3.0 (alternative unidirectional)
  - Paths tend to be stable (just changed by repair)
    - No oscillations
    - No risk of transient frame disorder

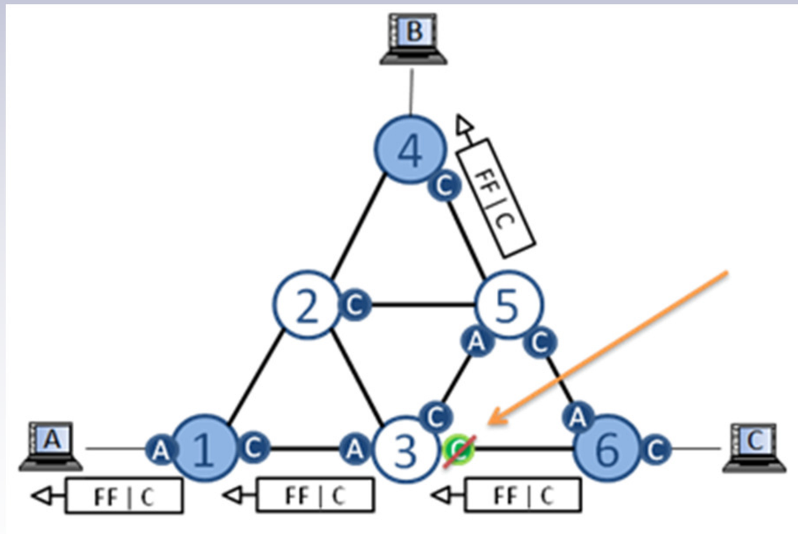
ARP-Path

- ARP-Path 3.0 (alternative unidirectional)
  - Discovery and path creation works as in 2.0 when there are no paths



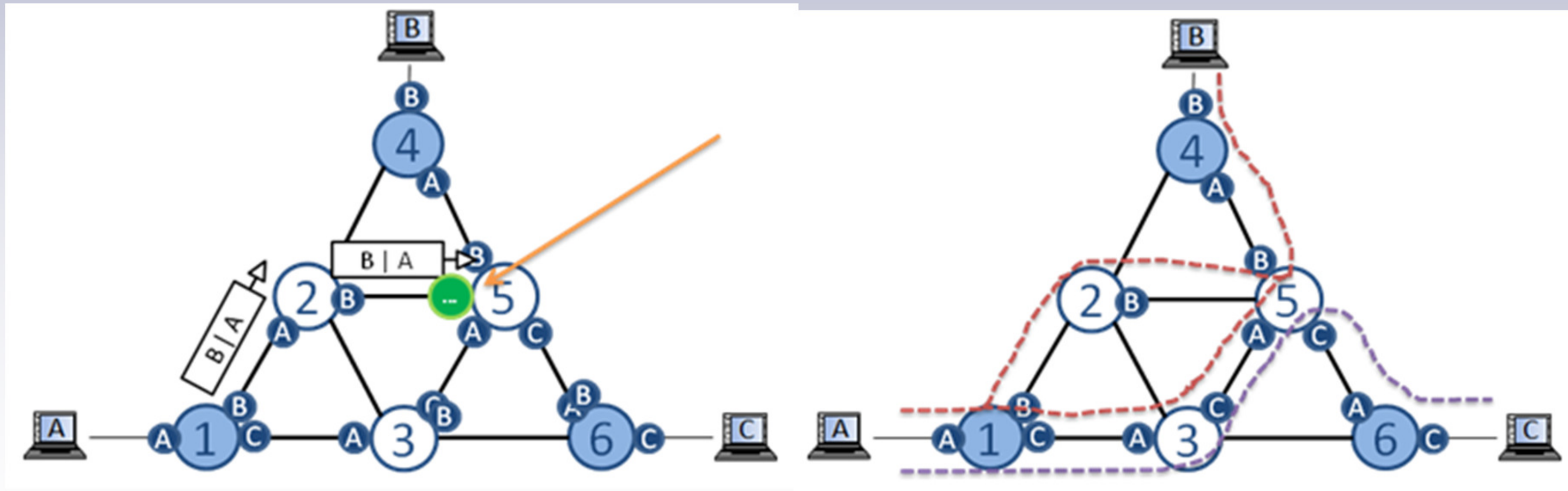
ARP-Path

- ARP-Path 3.0 (alternative unidirectional)
  - Discovery and source path creation (ARP Request)



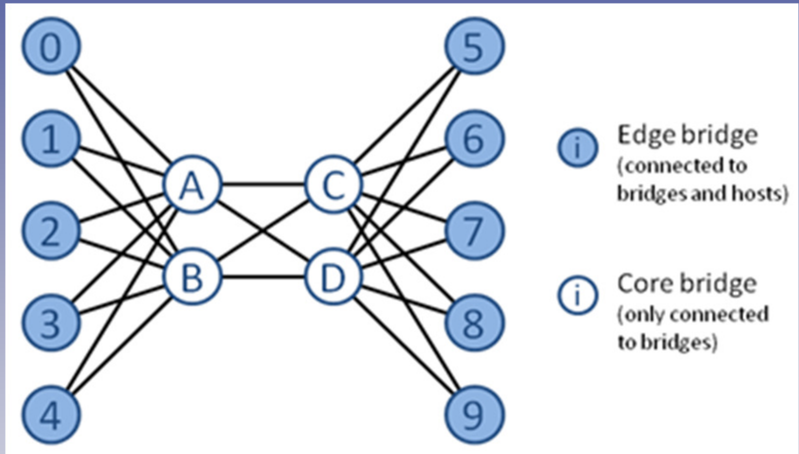
ARP-Path

- ARP-Path 3.0 (alternative unidirectional)
  - Destination path creation (ARP Reply)



ARP-Path

- Low latency paths:
  - OMNeT++

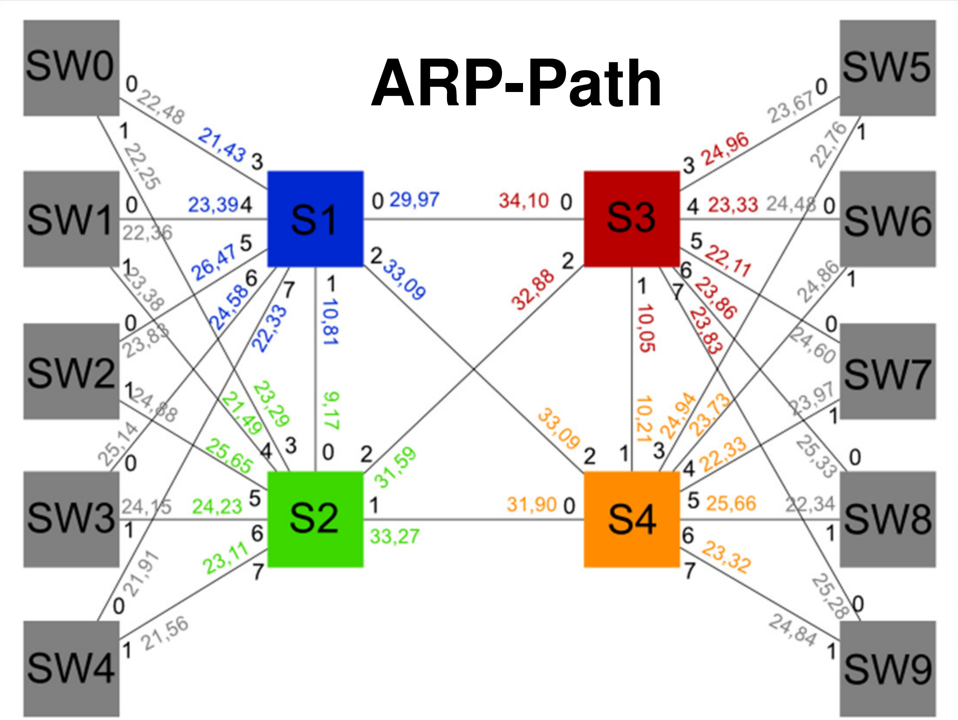


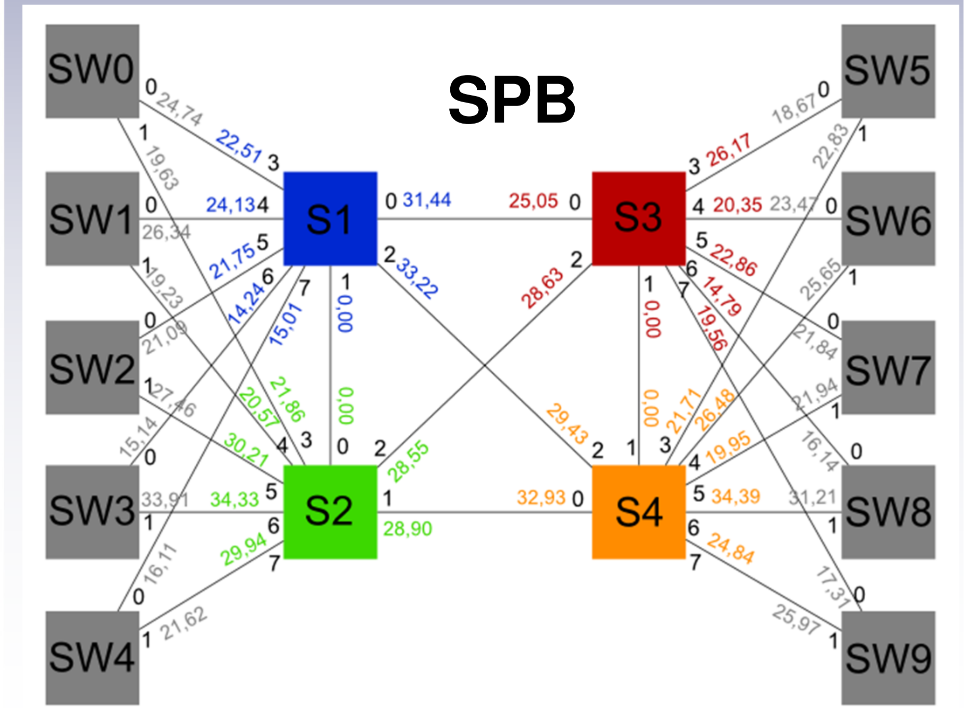
5000s/1.6s/Uniforme	Latencia máxima	Latencia mínima	Latencia media
SPB	1,884 ms	0,021 ms	0,335 ms
ARP-Path	0,811 ms	0,022 ms	0,343 ms

5000s/1.6s/No unif. (peso 100 en 1 host)	Latencia máxima	Latencia mínima	Latencia media
SPB	0,789 ms	0,038 ms	0,333 ms
ARP-Path	0,665 ms	0,041 ms	0,333 ms

ARP-Path better than plain SPB

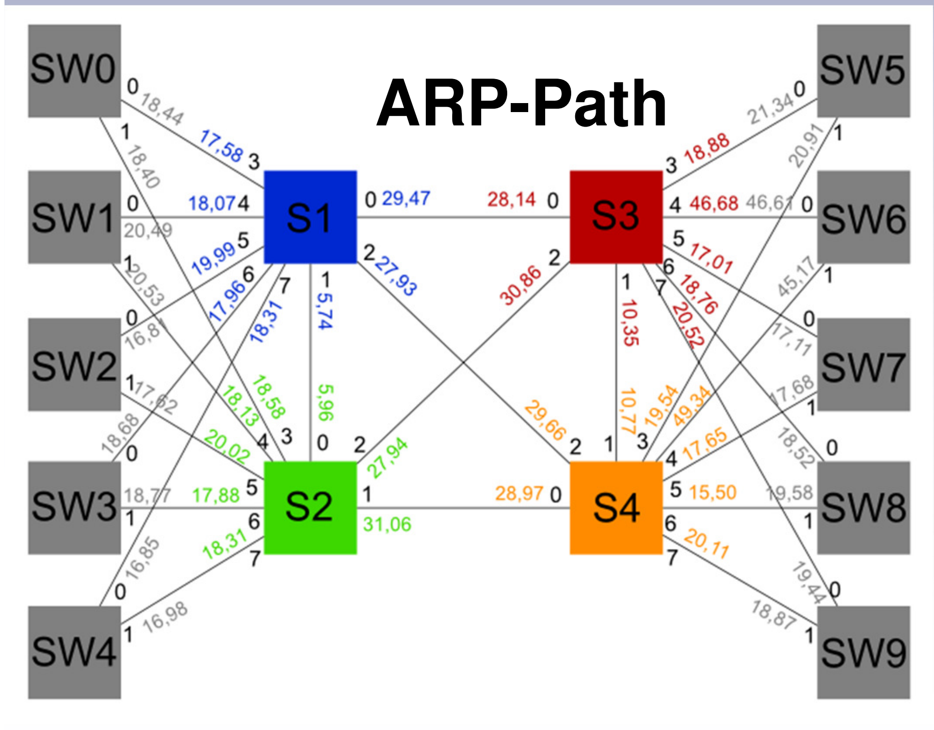
- Load balancing: ARP-path vs *plain* SPB (no ECMP) . Link utilization (%).
  - OMNeT++ (Random uniform traffic)

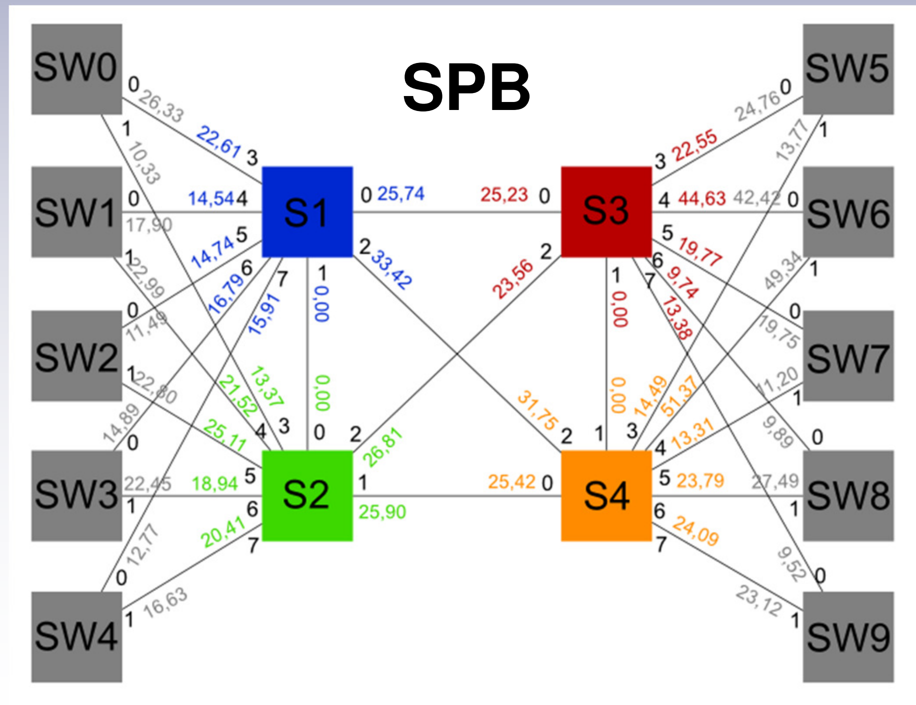




ARP-Path

- Load balancing: ARP-path vs *plain* SPB (no ECMP)
  - OMNeT++ (host at SW6 more traffic weight)





ARP-Path Latencies

- Fig. 3a) Average delay in ARP-Path and SPB is quite similar, but in some cases average delay in SPB is undesirably higher than ARP-Path.
- Fig. 3b). Maximum delays in plain SPB are much higher than those obtained in ARP-Path (more than an order of magnitude).

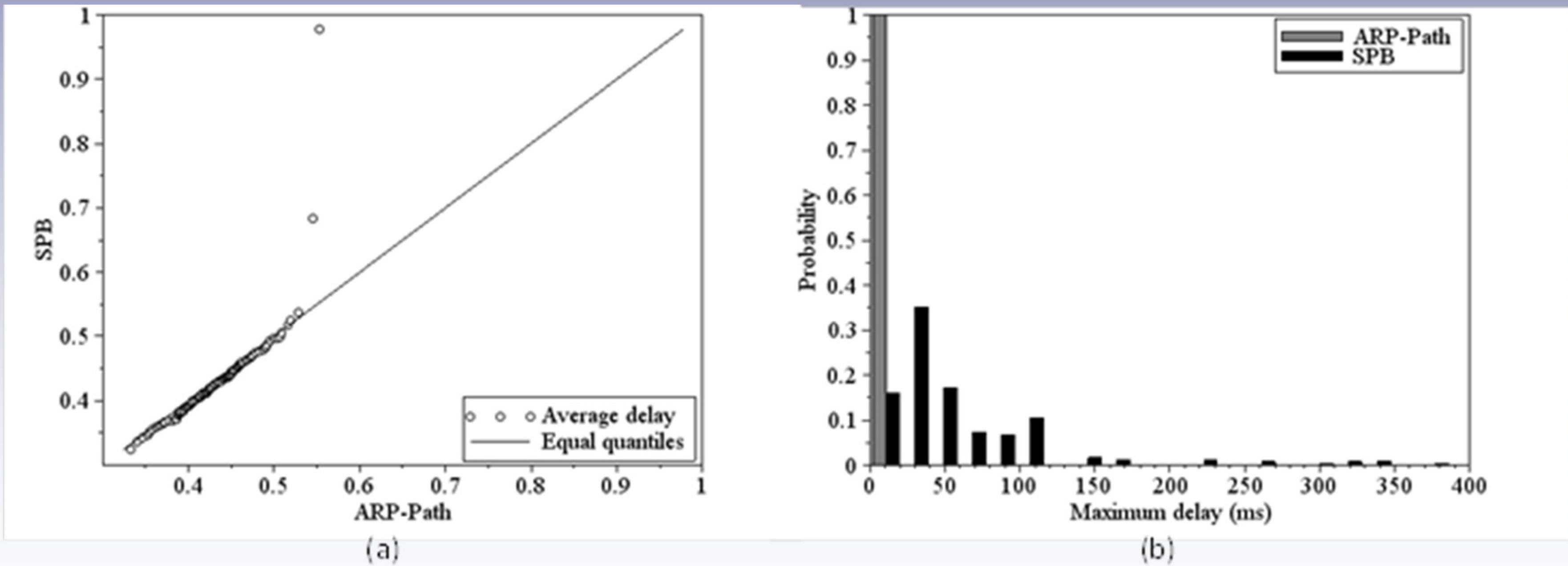


Figure 3. Packet Latencies for λ^{-1} = 0.4s. a) Q-Q plot for average delay; b) Probability density function for

ARP-Path distributes also at low loads

- Hosts conn. to left and right switches of 3x3 square mesh
- All-to-all traffic matrix
- OMNET++ simulat.
- Traffic is evenly distributed
  - Load adaptive routing

0	51.9 18.42 18.59	1	24.9 19.43 18.62	2
48.1 18.15 18.76		27.6 22.10 26.43		24.9 19.38 18.69
3	29.0 21.60 21.94	4	29.4 22.18 22.46	5
19.4 18.57 18.76		27.1 21.91 22.07		53.8 18.66 19.01
6	19.4 19.10 18.64	7	46.2 19.01 18.35	8

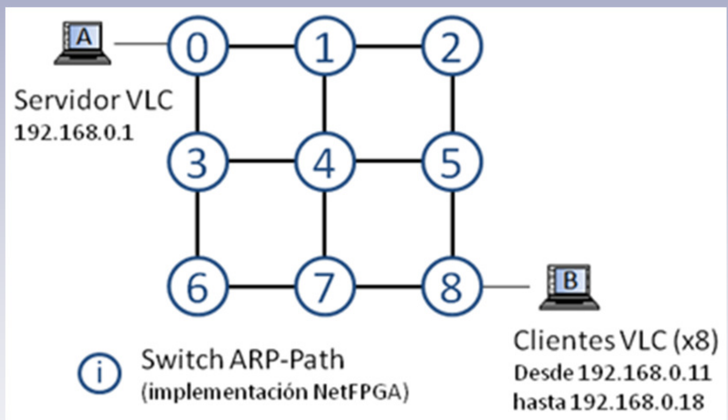
48.1 Percentage of routes between 0-8 using the link

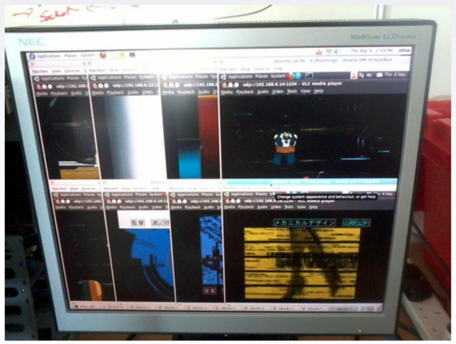
18.15 Link utilization at receiver port (average)

Ethernet Technology Summit

ARP-Path: path diversity (videos) even with very low loads

- Load balancing:
  - NetFPGA implementation





0	46,9	1	21,9	2
	0,00		0,00	
	0,00		0,00	
53,1		25,0		21,9
0,00		0,00		0,00
0,00		0,00		0,00
3	34,4	4	25,0	5
	0,00		0,00	
	0,00		0,00	
18,8		34,4		46,9
0,00		0,00		0,00
0,00		0,00		0,00
6	18,8	7	53,1	8
	0,00		0,00	
	0,00		0,00	

0,0 Percentage of routes using the link

0,0 Link utilization at receiver port (average)

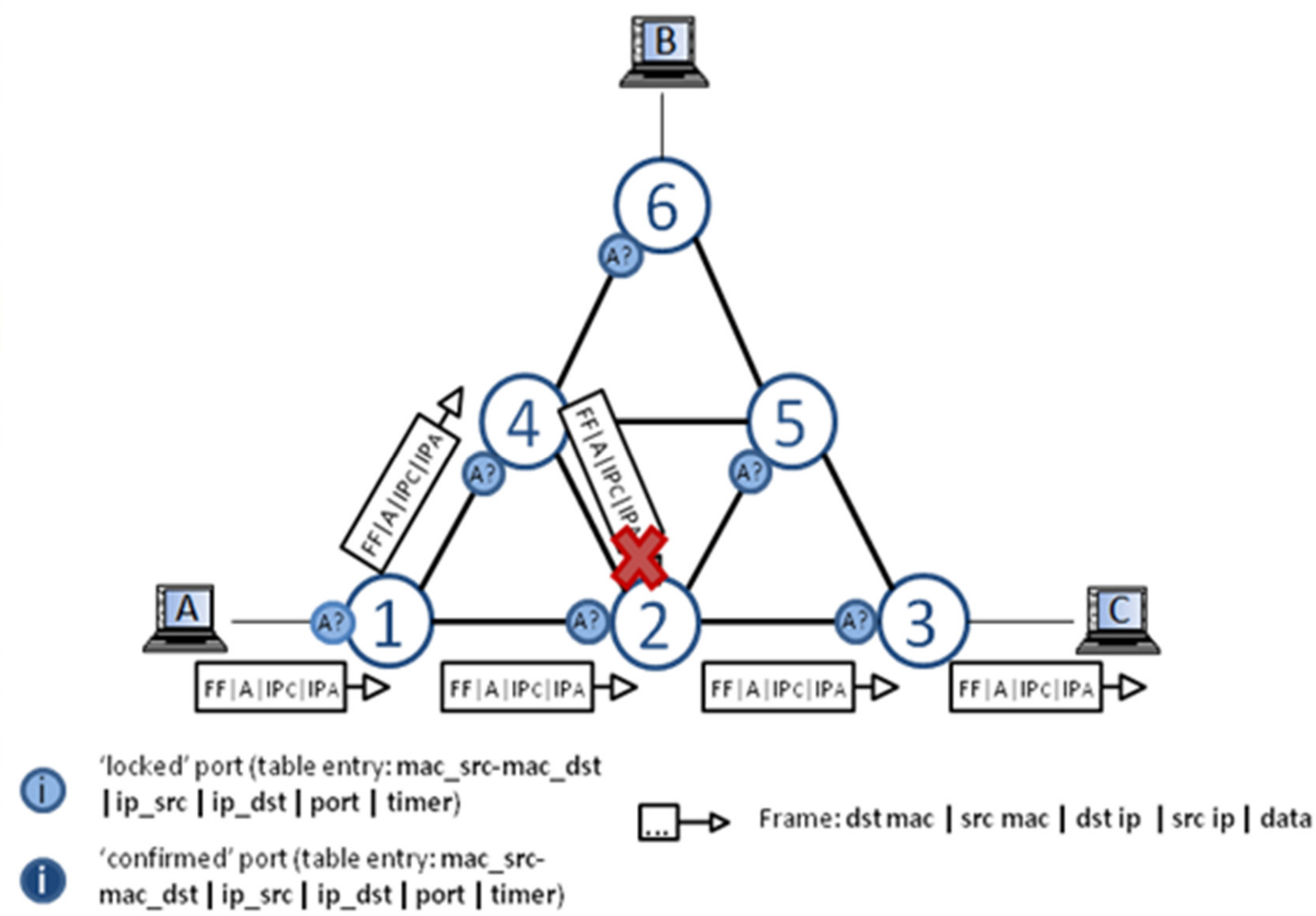
ARP-Path Implementations

- Pinging times:
  - Linux/Soekris(100Mbps): ~0,9ms
  - OpenFlow/Mininet(Virtual): ~0,05 ms
  - OpenFlow/NetFPGA(1Gbps): ~0,6 ms
  - OpenFlow/Linux(100Mbps): ~3ms
  - OpenFlow/NECswitch(1Gbps): ~0,6ms
  - NetFPGA(1Gbps): ~0,6ms

Flow-Path

key(src-dst)	srcIP	dstIP	port	timer	state
A-FF	IP_A	IP_C	1	0	*Locked*

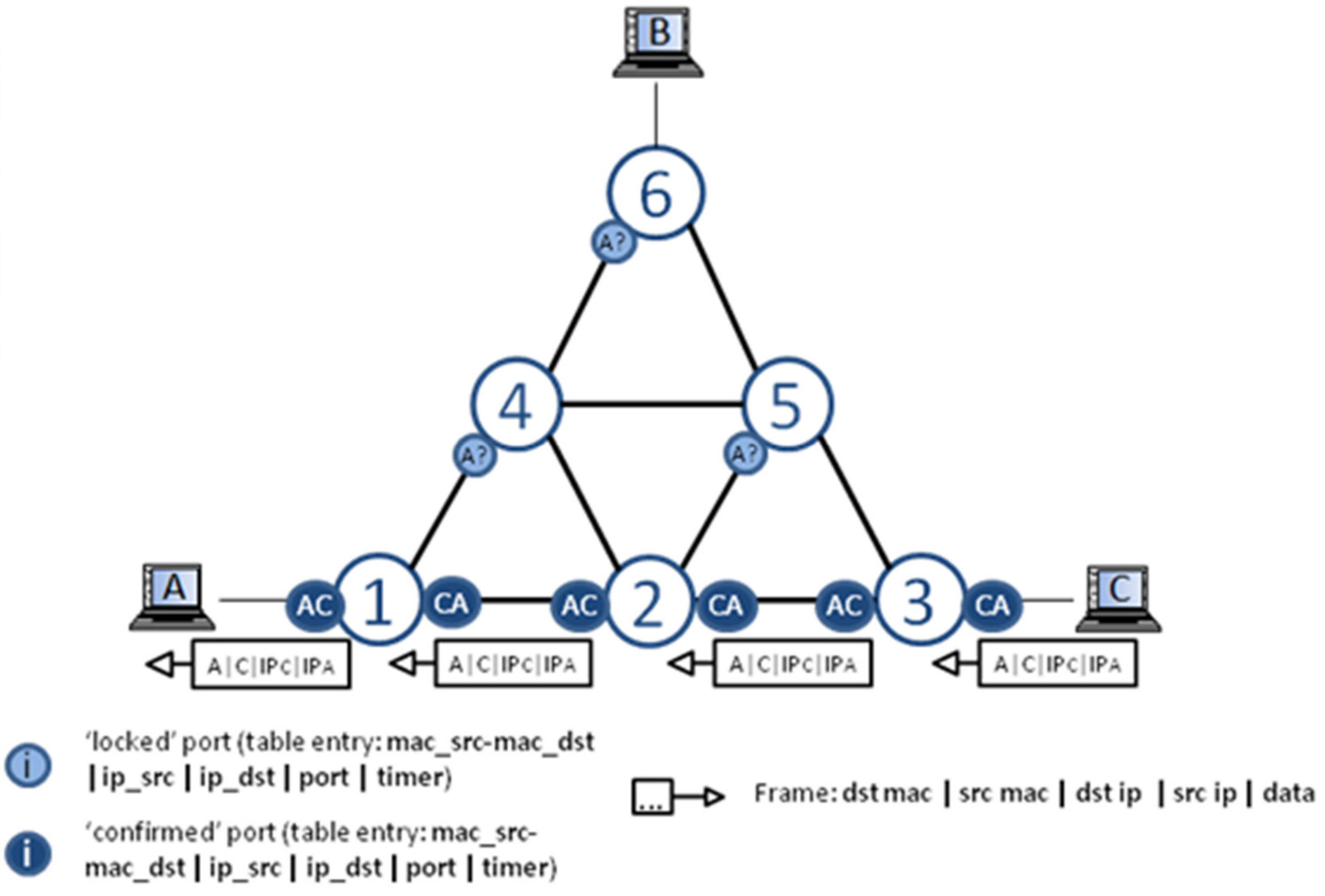
- Paths are created per flow:
- Path discovery (ARP Request)



Flow-Path

key(src-dst)	srcIP	dstIP	port	timer	state
A-C	IP_A	IP_C	1	0	*Confirmed*
C-A	IP_A	IP_C	2	0	*Confirmed*

- Paths are created per flow:
  - Path confirmation (ARP Reply)



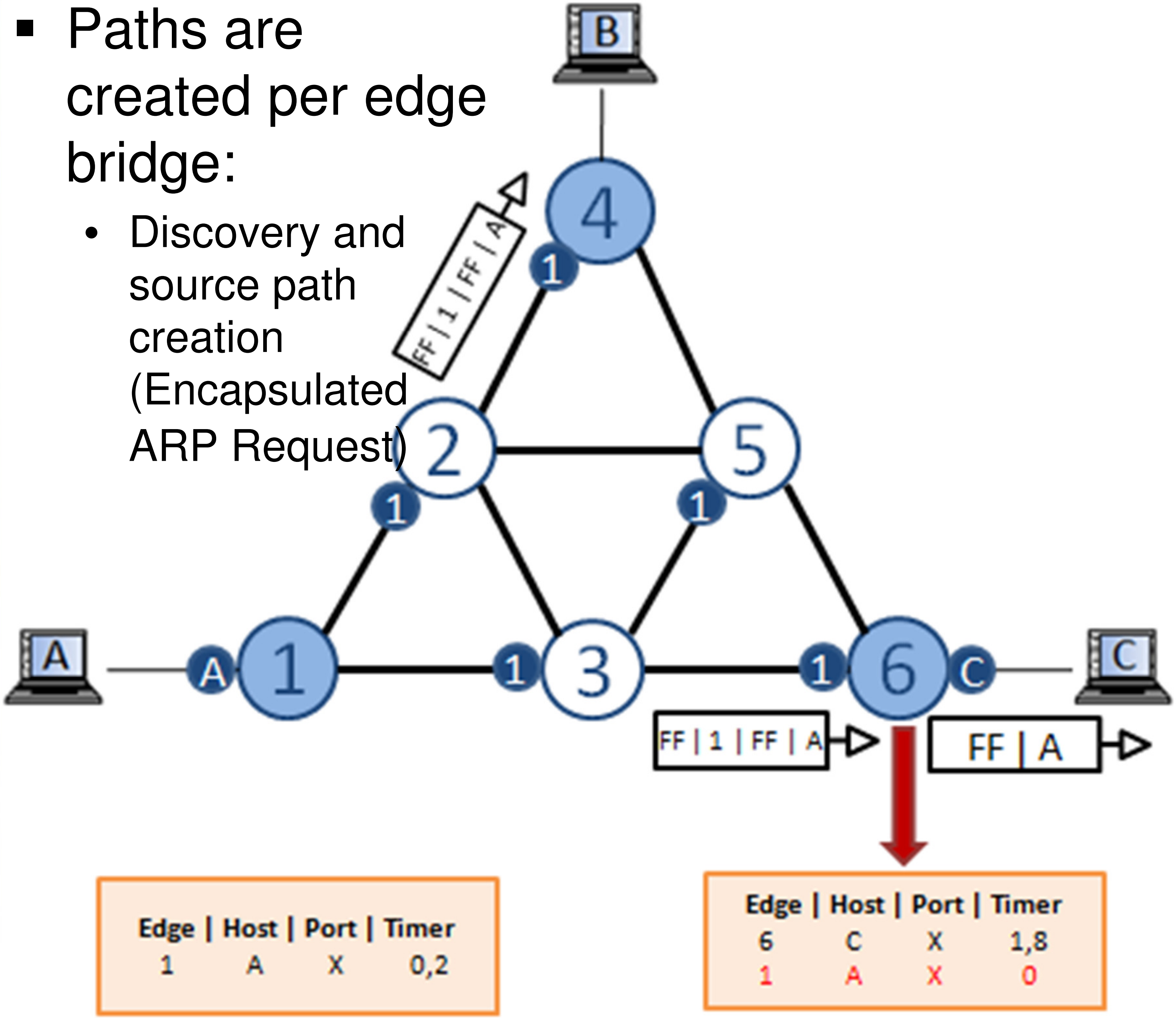
Flow-path versus ARP-path

- Flow-path Advantages
  - Guaranteed path symmetry
  - Even better load balancing: finer granularity
- Disadvantages
  - Increased stored state *O(flows)*

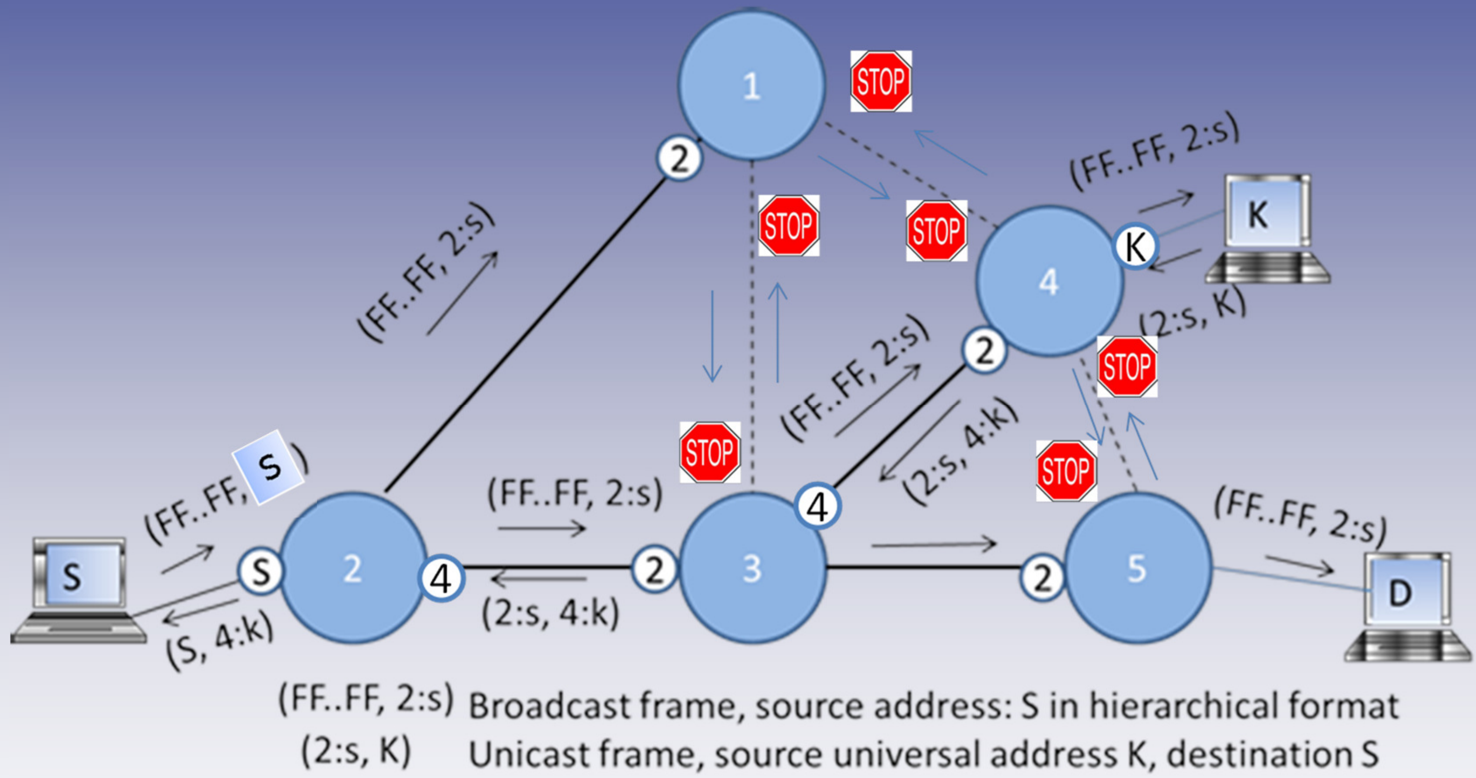
Bridge-Path (MAC in MAC, Path-Moose)

- Paths/trees are created per edge bridge
  - Less table entries → better scalability
  - Worsens path diversity and load balancing
    - Can be improved with multipath (and more table entries)
    - Using simultaneous, two-level path granularity (bridge-hosts)
- Variants:
  - ARP-Path MAC-in-MAC encapsulation
    - Also ARP-Path VLAN (like SPBV).
  - Path-Moose (NAT of MACs at edge bridges)
    - local host MAC address = *BridgeID:HostID*

- Paths are created per edge bridge:
  - Discovery and source path creation (Encapsulated ARP Request)



Path-Moose *(NAT of MACs= BridgeID:hostID, MAC(S) changed to 2:s at edge bridge 2)*



Setting a path and a tree rooted at bridge 2 with ARP Request from S to K and ARP Reply (K,S), learning BridgeID 2.

Ref.: "Path-Moose: A Scalable All-Path Bridging Protocol". G. Ibáñez et al. IEICE Transactions on Communications. March 2013.

Publications and Implementations

- 2010
  - *Fast Path Ethernet Switching: On-demand, Efficient Transparent Bridges for Data Center and Campus Networks. IEEE LANMAN Workshop May 2010.*
  - "*A Simple, Zero-configuration, Low Latency, Bridging Protocol*". LCN demos. Denver, oct. 2010. Best demo award.
    - ARP-path implementation on Openflow/NetFPGA switch implementation

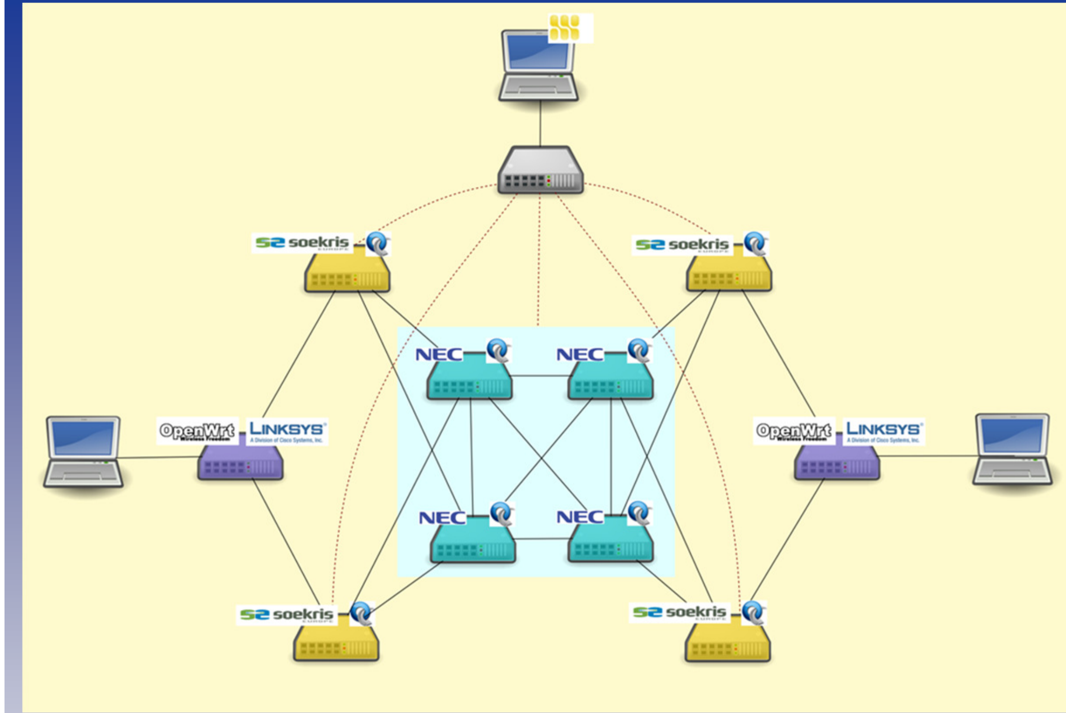
- Acknowledgement: Jad Naous (Ph.D. Stanford) for Open flow and NetFPGA implementations

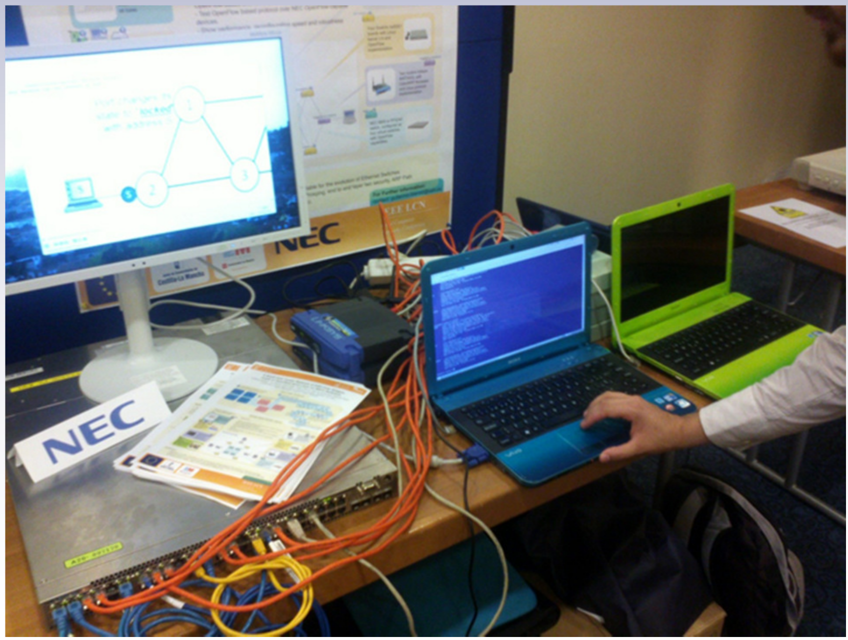
All-path on Openflow

- Early implementation of All-path protocol was in Openflow (2010), for proof-of-concept.
  - Fully operative on diverse Openflow switches
- All-path protocols best suited to either fully distributed networks or hybrids (with SDN)
  - Hybrid:
    - Basic All-path functionality in the switches for basic path exploration and forwarding
    - Complementary functions performed at SDN controller: path resiliency, additional routing, reconfiguration.

All-path on Openflow Switches

LCN 2011
(Bonn)
NEC and other Openflow Switches





- 4 NEC Openflow capable switches
- 4 Soekris boards (OF)
- 2 Open-WRT Linksys routers (Linux implementation)
- One Openflow (NOX) controller

Publications and Implementations

- 2011
  - "*Implementation of ARP-Path Low Latency Bridges in Linux and OpenFlow/NetFPGA*". HPSR 2011, Cartagena, April 2011.
  - *ARP Path: ARP-based Shortest Path Bridges. IEEE Communication Letters. July 2011*
  - "*Implementing ARP-Path Low Latency Bridges in NetFPGA*". SIGCOMM demos. Toronto (CA), August 2011.
  - "*A Small Data Center Network of ARP-Path Bridges made of Openflow Switches*". (In collaboration with NEC Europe) LCN demos. Bonn, 2011.

Ethernet Technology Summit

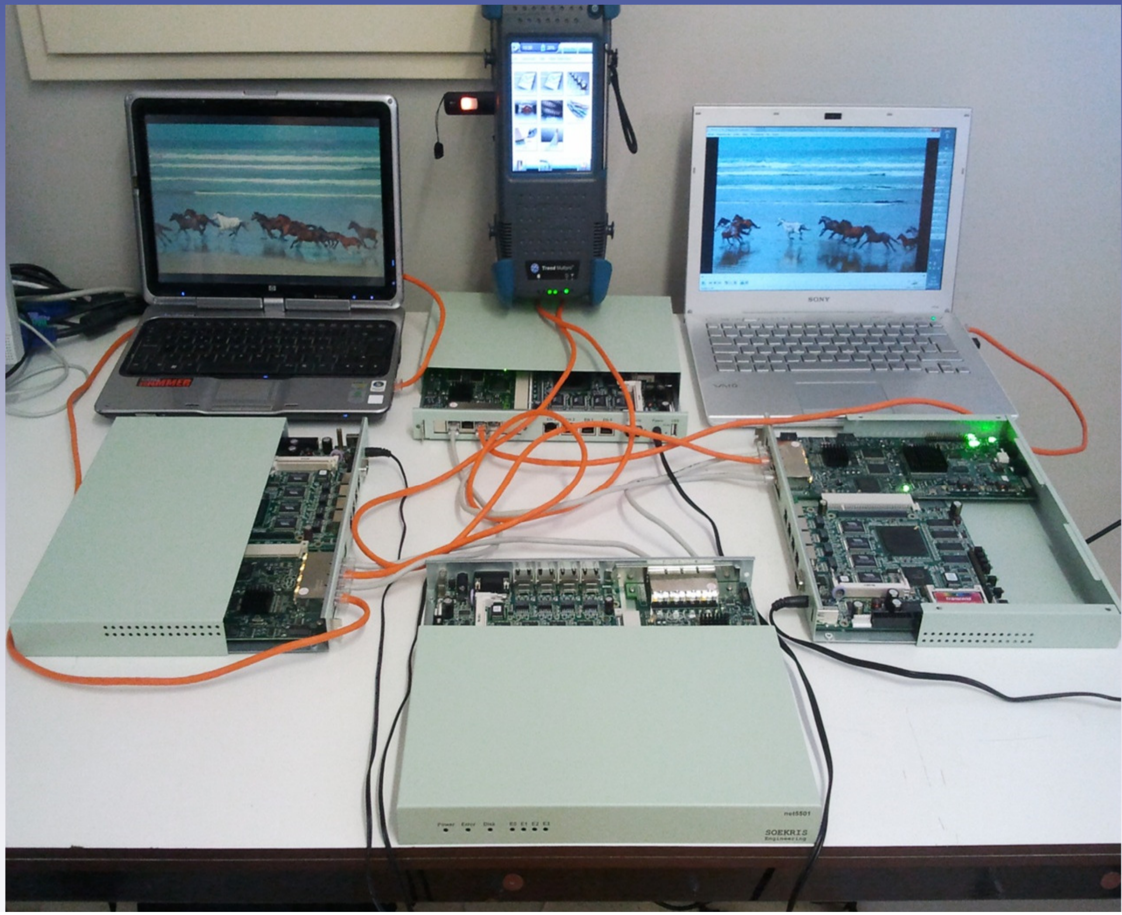
Sigcomm 2011(Toronto)

- 4 **NetFPGAs** on a PC
- Hardware: Verilog implementation
- Comparison with STP
- Robustness, path repair

Publications and Implementations

- 2012
  - *"Dynamic Load Routing/Path Diversity in a Network of ARP-Path NetFPGA Switches". LCN 2012 demo.*
  - *"Flow-Path: An AllPath Flow-based Protocol". LCN 2012.*
- 2013.
  - *"Path-Moose: A Scalable All-Path Bridging Protocol". IEICE Transactions on Communications. March 2013.*
  - *"Evaluating Native Load Distribution of ARP-Path Bridging Protocol in Mesh and Data Center". ICC 2013. Budapest, June 2013.*
  - *"All-path Bridging: Path Exploration as an Efficient Alternative to Path Computation in Bridging Standards". IEEE Workshop on Telecommunication Standards. June 2013.*

NetFPGA implementation LCN 2012



- 4 **NetFPGAs**
  - With Soekris board
- Full hardware implementation (Verilog)
- Load distribution
- Full infrastructure utilization vs STP
- Fully transparent

Future Work

- Implementation in Virtualized bridged networks and combinations with SDN (OpenvSwitch,…)
- Multicast and broadcast traffics optimization.
- Audio Video Bridges
- Bridge-path: Coexistence of different path granularities (per host,per bridge)
- Deep evaluation and adaptation to data center reqs.
- Integration on hybrid wired-wireless (802.1/.11)networks

Ethernet Technology Summit

Conclusion

- Path exploration is effective for switches:
  - **Minimizes maximum frame latencies vs. plain SPB**
  - **Load adaptive routing (native path diversity)**
    - basic protocol uses **per-host *just-in-time* path selection**
    - **Instant adaptation to traffic load**
- Path is not **predictable** (not computed) but the protocol is **resilient:** (SPB/TRILL aren't fully predictable either)
  - Even if only a path to destination remains, it will be selected
- All-Path protocols **show a way for the evolution of the transparent bridge paradigm**
  - On a pure bridging basis, proven by implementations.
- Combines well with Openflow

Ethernet Technology Summit

Thank you for your attention

Questions?

Elisa Rojas
elisa.rojas@uah.es

Guillermo Ibánez
guillermo.ibanez@uah.es