Floodless in SEATTLE: A Scalable Ethernet Architecture for Large Enterprises

Chang Kim, and Jennifer Rexford
http://www.cs.princeton.edu/~chkim
Princeton University

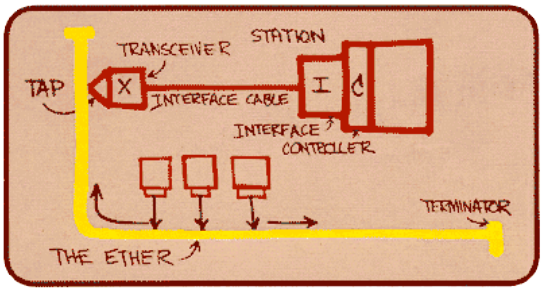
Goals of Today's Lecture

- Reviewing Ethernet bridging (Lec. 10, 11)
  - Flat addressing, and plug-and-play networking
  - Flooding, broadcasting, and spanning tree
  - VLANs
- New challenges to Ethernet
  - Control-plane scalability
    - Avoiding flooding, and reducing routing-protocol overhead
  - Data-plane efficiency
    - Enabling shortest-path forwarding and load-balancing
- SEATTLE as a solution
  - Amalgamation of various networking technologies covered so far
  - E.g., link-state routing, name resolution, encapsulation, DHT, etc.

Quick Review of Ethernet

Ethernet

- Dominant wired LAN technology
  - Covers the first IP-hop in most enterprises/campuses
- First widely used LAN technology
- Simpler, cheaper than token LANs, ATM, and IP
- Kept up with speed race: 10 Mbps – 10 Gbps

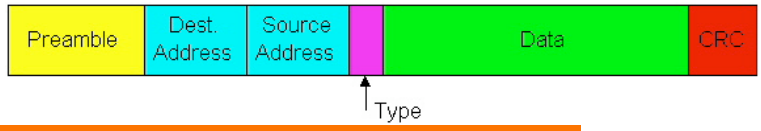


Metcalfe's Ethernet sketch

4

Ethernet Frame Structure

- Addresses: source and destination MAC addresses
  - Flat, globally unique, and permanent 48-bit value
  - Adaptor passes frame to network-level protocol
    - If destination address matches the adaptor
    - Or the destination address is the broadcast address
  - Otherwise, adapter discards frame
- Type: indicates the higher layer protocol
  - Usually IP♪



5

Ethernet Bridging: Routing at L2

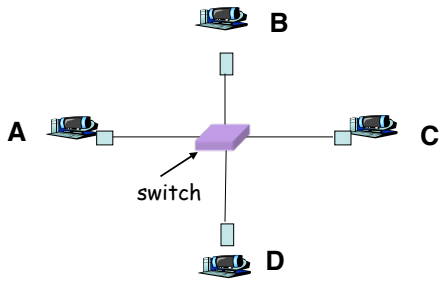
- Routing determines paths to destinations through which traffic is forwarded
- Routing takes place at any layer (including L2) where devices are reachable across multiple hops

App Layer	P2P, or CDN routing (Lec. 18)
	Overlay routing (Lec. 17)
IP Layer	IP routing (Lec. 13 ~ 15)
Link Layer	**Ethernet bridging** (Lec. 10, 11)

6

Ethernet Bridges Self-learn Host Info.

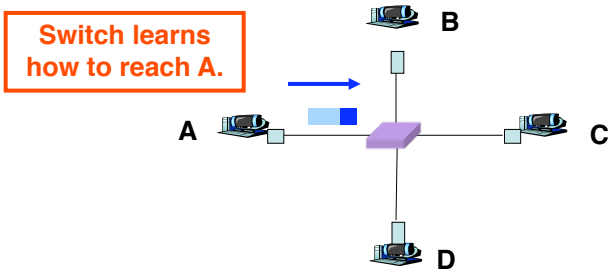
- Bridges (switches) forward frames selectively
  - Forward frames only on segments that need them
- Switch table
  - Maps destination MAC address to outgoing interface
  - Goal: construct the switch table automatically



7

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

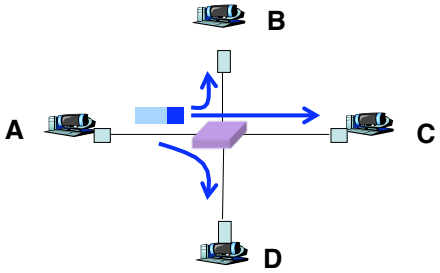


8

Self Learning: Handling Misses

- Floods when frame arrives with unfamiliar dst or broadcast address
  - Forward the frame out all of the interfaces
  - … except for the one where the frame arrived
  - Hopefully, this case won't happen very often

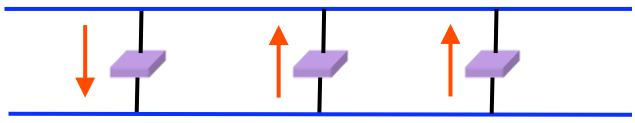
When in doubt, shout!



9

Flooding Can Lead to Loops

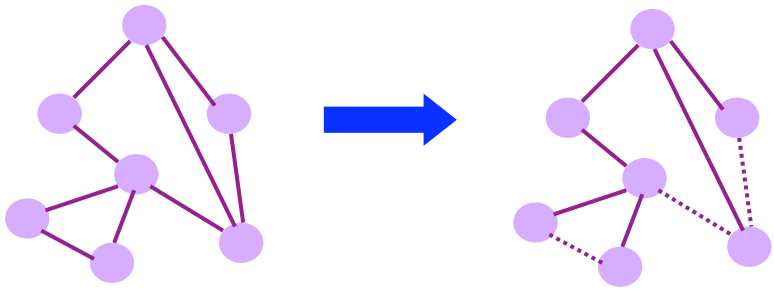
- Flooding can lead to forwarding loops, confuse bridges, and even collapse the entire network
  - E.g., if the network contains a cycle of switches
  - Either accidentally, or by design for higher reliability



10

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



11

Interaction with the Upper Layer (IP)

- Bootstrapping end hosts by automating host configuration (e.g., IP address assignment)
  - DHCP (Dynamic Host Configuration Protocol)
  - Broadcast DHCP discovery and request messages
- Bootstrapping each conversation by enabling resolution from IP to MAC addr
  - ARP (Address Resolution Protocol)
  - Broadcast ARP requests
- Both protocols work via Ethernet-layer broadcasting (i.e., shouting!)

12

Broadcast Domain and IP Subnet

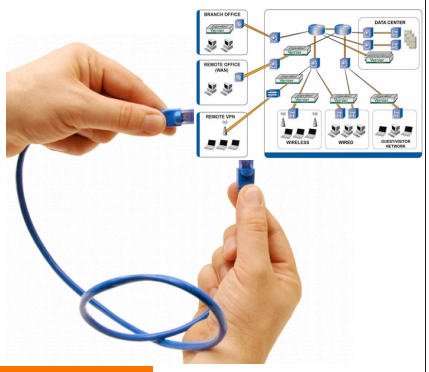
- Ethernet broadcast domain
  - A group of hosts and switches to which the same broadcast or flooded frame is delivered
  - Note: broadcast domain != collision domain
- Broadcast domain == IP subnet
  - Uses ARP to reach other hosts in the same subnet
  - Uses default gateway to reach hosts in different subnets
- Too large a broadcast domain leads to
  - Excessive flooding and broadcasting overhead
  - Insufficient security/performance isolation

13

New Challenges to Ethernet, and SEATTLE as a solution

"All-Ethernet" Enterprise Network?

- *"All-Ethernet"* makes network mgmt easier
  - Flat addressing and self-learning enables plug-and-play networking
  - Permanent and location independent addresses also simplify
    - Host mobility
    - Access-control policies
    - Network troubleshooting



15

But, Ethernet Bridging Does Not Scale

- Flooding-based delivery
  - Frames to unknown destinations are flooded

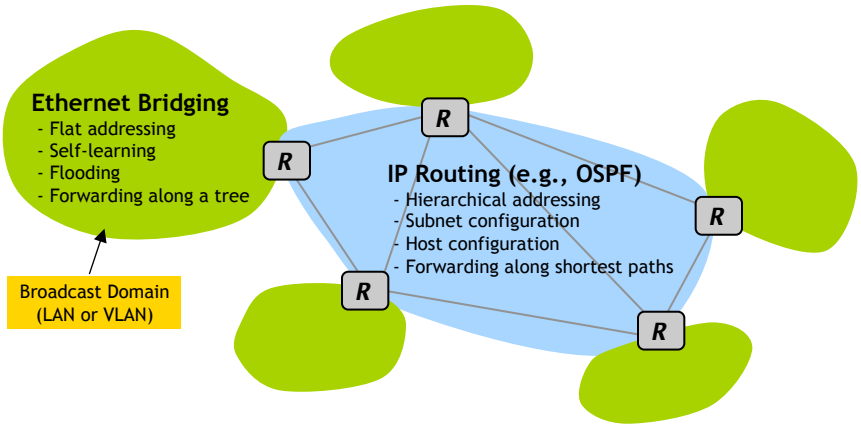
- Broadcasting for basic service
  - Bootstrapping relies on broadcasting
  - Vulnerable to resource exhaustion attacks

- Inefficient forwarding paths
  - Loops are fatal due to broadcast storms; uses the STP
  - Forwarding along a single tree leads to inefficiency and lower utilization

16

State of the Practice: A Hybrid Architecture

Enterprise networks comprised of Ethernet-based IP subnets interconnected by routers



Motivation

Neither bridging nor routing is satisfactory.

Can't we take only the best of each?

Architectures / Features	Ethernet Bridging	IP Rout	SEATTLE
Ease of configuration	👍	👎	👍
Optimality in addressing	👍	👎	👍
Host mobility	👍	👎	👍
Path efficiency	👎	👍	👍
Load distribution	👎	👍	👍
Convergence speed	👎	👍	👍
Tolerance to loop	👎	👍	👍

SEATTLE (**S**calable **E**thernet **A**rchi**T**ec**T**ure for **L**arger **E**nterprises)

Overview

- Objectives
- SEATTLE architecture
- Evaluation
- Applications and benefits
- Conclusions

19

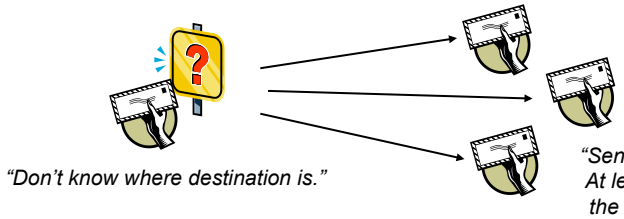
Overview: Objectives

- Objectives
  - Avoiding flooding
  - Restraining broadcasting
  - Keeping forwarding tables small
  - Ensuring path efficiency
- SEATTLE architecture
- Evaluation
- Applications and Benefits
- Conclusions

20

Avoiding Flooding

- Bridging uses flooding as a routing scheme
  - Unicast frames to unknown destinations are flooded



  - Does not scale to a large network

- Objective #1: Unicast unicast traffic
  - Need a control-plane mechanism to discover and disseminate hosts' location information

21

Restraining Broadcasting

- Liberal use of broadcasting for bootstrapping (DHCP and ARP)
  - Broadcasting is a vestige of shared-medium Ethernet
  - Very serious overhead in switched networks

- Objective #2: Support unicast-based bootstrapping
  - Need a directory service

- Sub-objective #2.1: Yet, support general broadcast
  - Nonetheless, handling broadcast should be more scalable

22

Keeping Forwarding Tables Small

- Flooding and self-learning lead to unnecessarily large forwarding tables
  - Large tables are not only inefficient, but also dangerous
- Objective #3: Install hosts' location information only when and where it is needed
  - Need a reactive resolution scheme
  - Enterprise traffic patterns are better-suited to reactive resolution

23

Ensuring Optimal Forwarding Paths

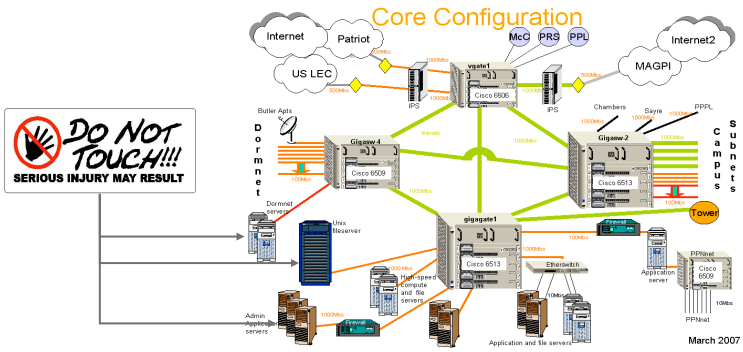
- Spanning tree avoids broadcast storms. But, forwarding along a single tree is inefficient.
  - Poor load balancing and longer paths
  - Multiple spanning trees are insufficient and expensive

- Objective #4: Utilize shortest paths
  - Need a routing protocol
- Sub-objective #4.1: Prevent broadcast storms
  - Need an alternative measure to prevent broadcast storms

24

Backwards Compatibility

- Objective #5: Do not modify end-hosts
  - From end-hosts' view, network must work the same way



  - End hosts should
    - Use the same protocol stacks and applications
    - Not be forced to run an additional protocol

25

Overview: Architecture

- Objectives
- SEATTLE architecture
  - Hash-based location management
  - Shortest-path forwarding
  - Responding to network dynamics
- Evaluation
- Applications and Benefits
- Conclusions

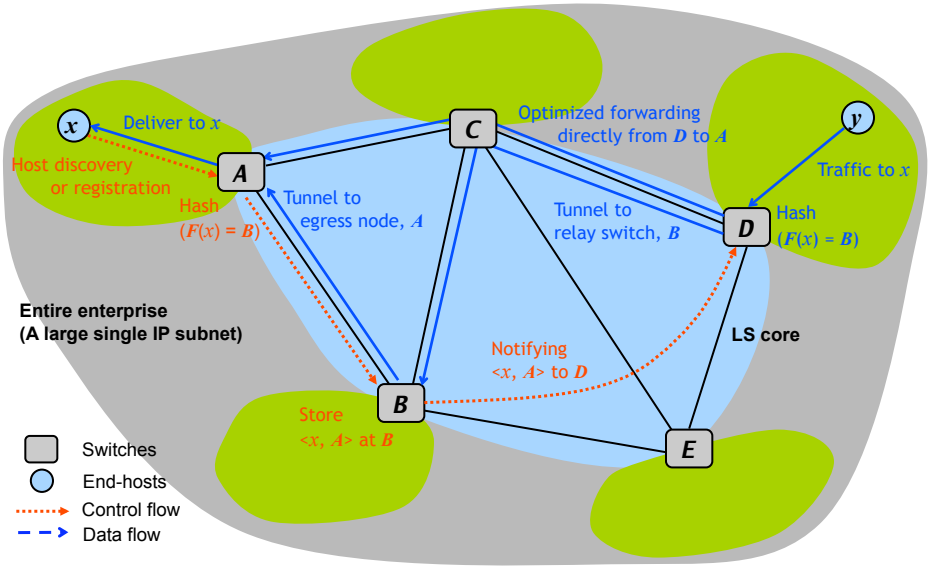
26

SEATTLE in a Slide

- Flat addressing of end-hosts
  - Switches use hosts' MAC addresses for routing
  - Ensures zero-configuration and backwards-compatibility (Obj # 5)
- Automated host discovery at the edge
  - Switches detect the arrival/departure of hosts
  - Obviates flooding and ensures scalability (Obj #1, 5)
- Hash-based on-demand resolution
  - Hash deterministically maps a host to a switch
  - Switches resolve end-hosts' location and address via hashing
  - Ensures scalability (Obj #1, 2, 3)
- Shortest-path forwarding between switches
  - Switches run link-state routing to maintain only switch-level topology (i.e., do **not** disseminate end-host information)
  - Ensures data-plane efficiency (Obj #4)

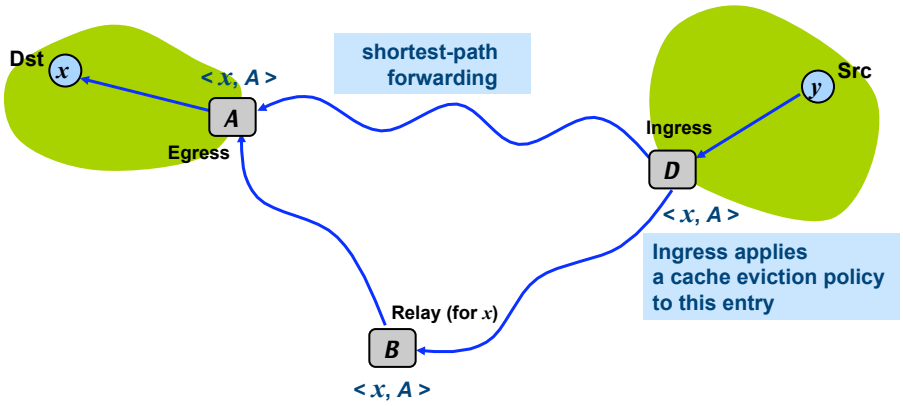
27

How does it work?



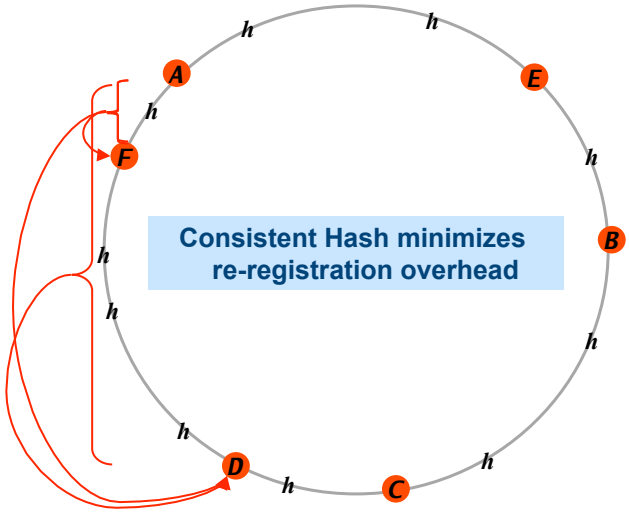
28

Terminology

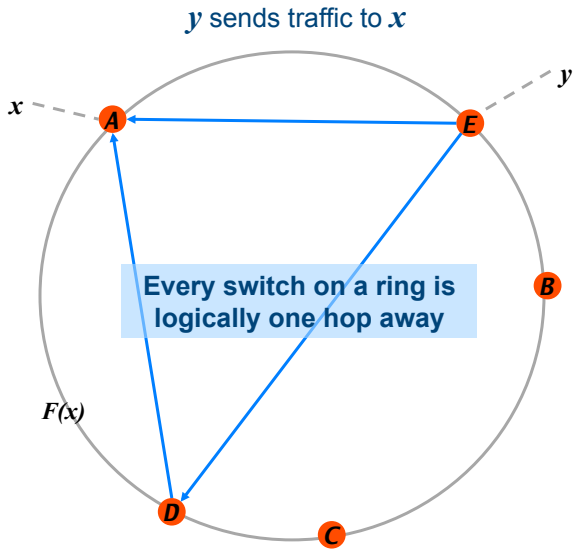


Responding to Topology Changes

- The quality of hashing matters!

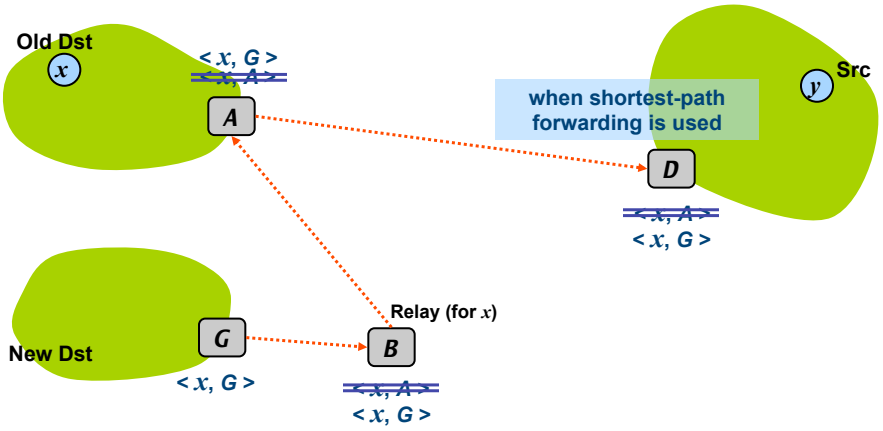


Single Hop Look-up



31

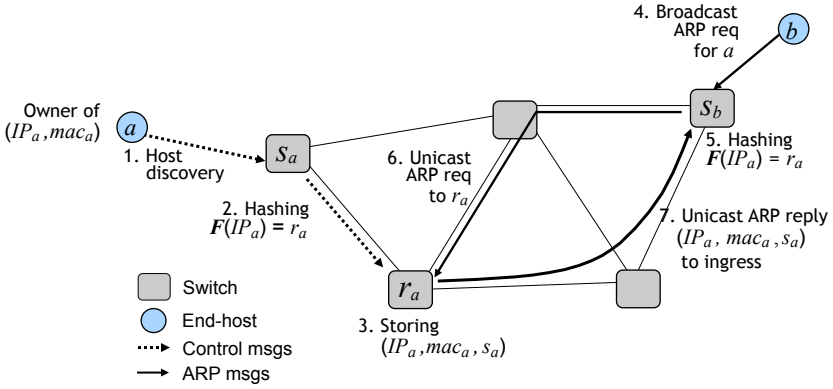
Responding to Host Mobility



32

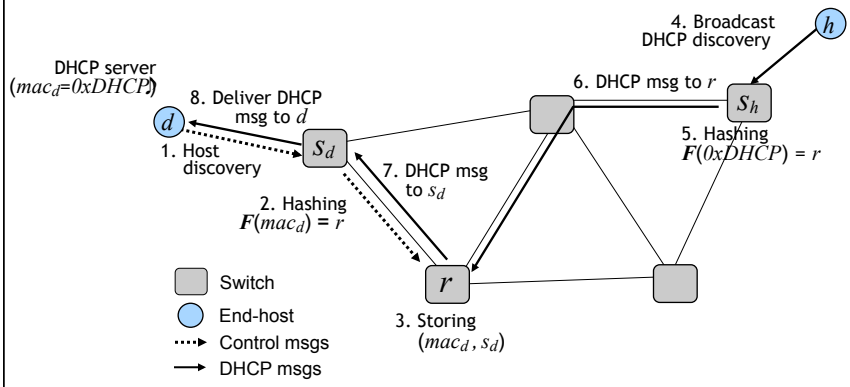
Unicast-based Bootstrapping: ARP

- ARP
  - Ethernet: Broadcast requests
  - SEATTLE: Hash-based on-demand address resolution



Unicast-based Bootstrapping: DHCP

- DHCP
  - Ethernet: Broadcast requests and replies
  - SEATTLE: Utilize DHCP relay agent (RFC 2131)
    - Proxy resolution by ingress switches via unicasting



Overview: Evaluation

- Objectives
- SEATTLE architecture
- Evaluation
  - Scalability and efficiency
  - Simple and flexible network management
- Applications and Benefits
- Conclusions

Control-Plane Scalability When Using Relays

- Minimal overhead for disseminating host-location information
  - Each host's location is advertised to only two switches
- Small forwarding tables
  - The number of host information entries over all switches leads to $O(H)$, not $O(SH)$
- Simple and robust mobility support
  - When a host moves, updating only its relay suffices
  - No forwarding loop created since update is atomic

Data-Plane Efficiency w/o Compromise

- Price for path optimization
  - Additional control messages for on-demand resolution
  - Larger forwarding tables
  - Control overhead for updating stale info of mobile hosts
- The gain is much bigger than the cost
  - Because most hosts maintain a small, static communities of interest (COIs) [*Aiello et al., PAM'05*]
  - Classical analogy: COI ↔ Working Set (WS); Caching is effective when a WS is small and static

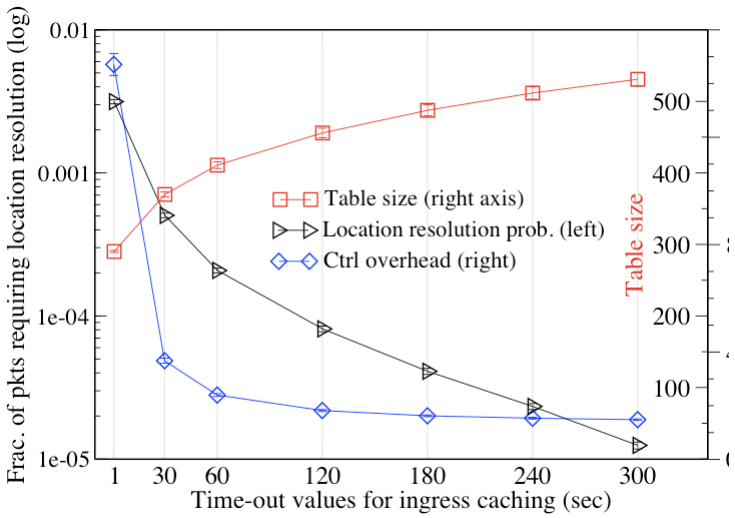
37

Large-scale Packet-level Simulation

- In-house packet level simulator
  - Event driven (similar to NS-2)
  - Optimized for intensive control-plane simulation; models for data-plane simulation is limited (e.g., does not model queueing)
- Test network topology
  - Small enterprise (synthetic), campus (a large state univ.), and large Internet service providers (AS1239)
  - Varying number of end hosts (10 ~ 50K) with up to 500 switches
- Test traffic
  - Synthetic traffic based on a large national research lab's internal packet traces
    - 17.8M packets from 5,128 hosts across 22 subnets
    - Inflate the trace while preserving original destination popularity distribution

38

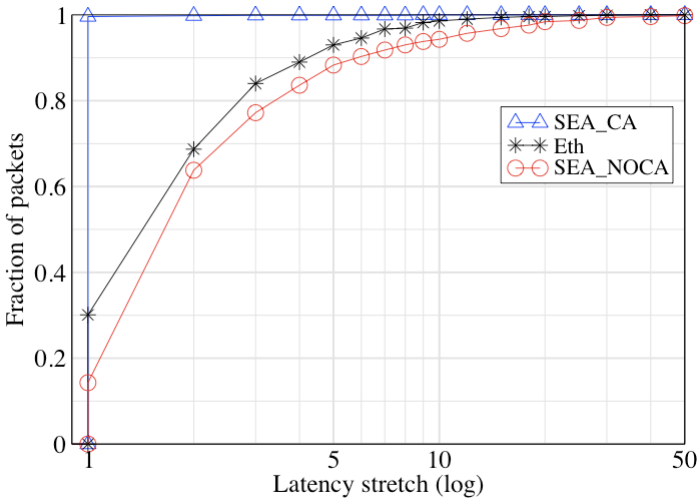
Tuning the System



39

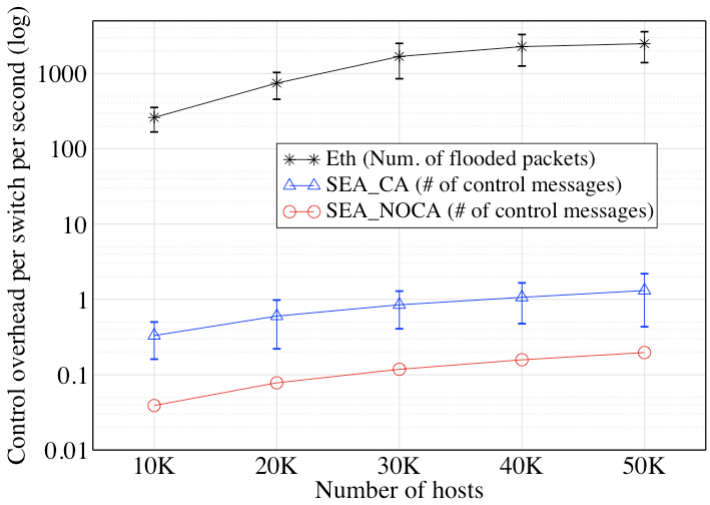
Stretch: Path Optimality

Stretch = Actual path length / Shortest path length



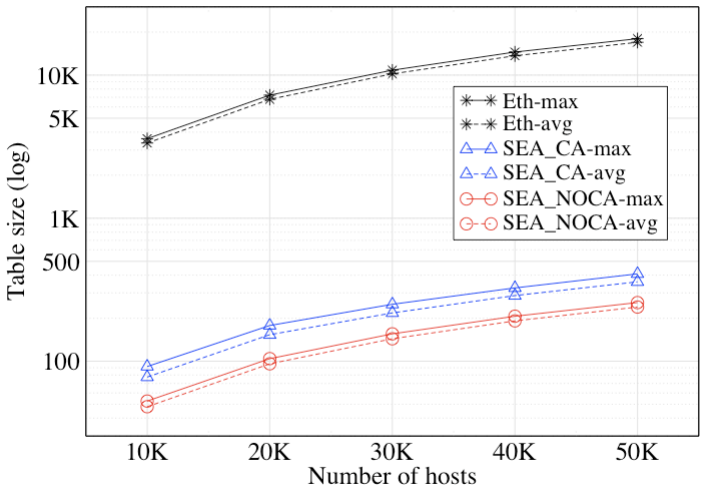
40

Control Overhead: Noisiness of Protocol



41

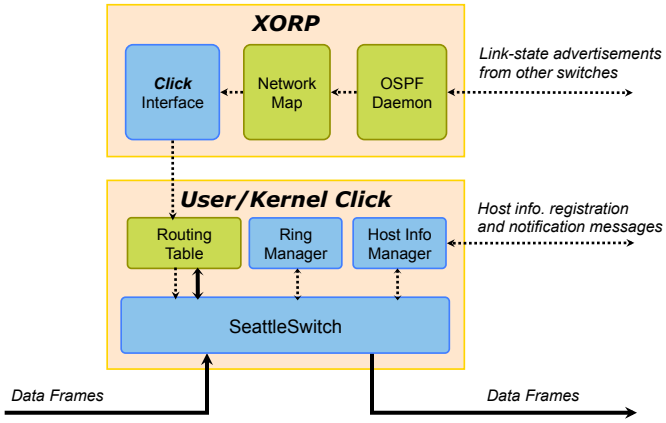
Amount of State: Conciseness of Protocol



42

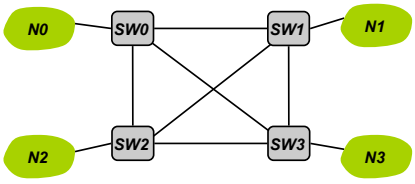
Prototype Implementation

- Link-state routing: eXtensible Open Router Platform
- Host information management and traffic forwarding: The Click modular router



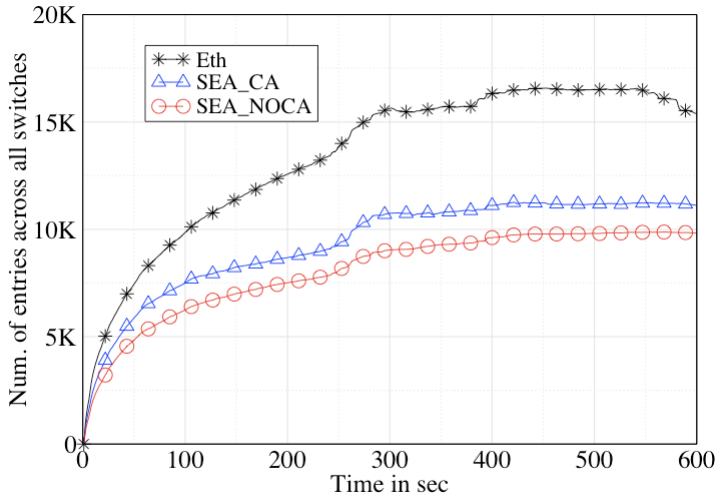
Emulation Using the Prototype

- Emulab experimentation
  - Emulab is a large set of time-shared PCs and networks interconnecting them
- Test Network Configuration
  - 10 PC-3000 FreeBSD nodes
  - Realistic latency on each link



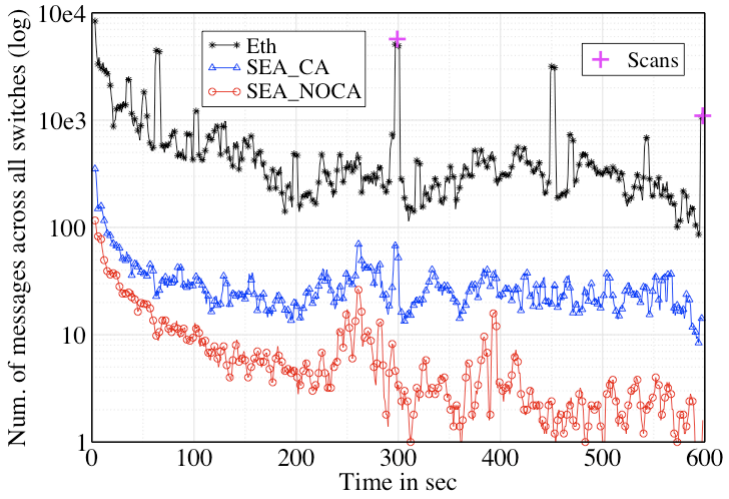
- Test Traffic
  - Replayed LBNL internal packet traces in real time
- Models tested
  - Ethernet, SEATTLE w/o path opt., and SEATTLE w/ path opt.
  - Inactive timeout-based eviction: 5 min *ltout*, 60 sec *rtout*

Table Size



45

Control Overhead



46

Overview: Applications and Benefits

- Objectives
- SEATTLE architecture
- Evaluation
- Applications and Benefits
- Conclusions

Ideal Application: Data Center Network

- Data centers
  - Backend of the Internet
  - Mid- (most enterprises) to mega-scale (Google, Yahoo, MS, etc.)
    - E.g., A regional DC of a major on-line service provider consists of 25K servers + 1K switches/routers
- To ensure business continuity, and to lower operational cost, DCs must
  - Adapt to varying workload → **Breathing**
  - Avoid/Minimize service disruption (when maintenance, or failure) → **Agility**
  - Maximize aggregate throughput → **Load balancing**

DC Mechanisms to Ensure HA and Low Cost

- Agility and flexibility mechanisms
  - Server virtualization and virtual machine migration to mask failure
  - Could virtualize even networking devices as well
- IP routing is scalable and efficient, however
  - Can't ensure service continuity across VM migration
  - Must reconfigure network and hosts to handle topology changes (e.g., maintenance, breathing)
- Ethernet allows for business continuity and lowers operational cost, however
  - Can't put 25K hosts and 1K switches in a single broadcast domain
  - Tree-based forwarding simply doesn't work
- **SEATTLE meets all these requirements neatly**

Conclusions

- SEATTLE is a plug-and-playable enterprise architecture ensuring both scalability and efficiency
- Enabling design choices
  - Hash-based location management
  - Reactive location resolution and caching
  - Shortest-path forwarding
- Lessons
  - Trading a little data-plane efficiency for huge control-plane scalability makes a qualitatively different system
  - Traffic patterns are our friends

More Lessons

- You can create a new solution by combining existing techniques/ideas from different layers
  - E.g., DHT-based routing
    - First used for P2P, CDN, and overlay
    - Then extended to L3 routing (id-based routing)
    - Then again extended to L2 (SEATTLE)
  - Deflecting through intermediaries
  - Link-state routing
  - Caching
  - Mobility support through fixed registration points
- Innovation is still underway

51

Thank you.

Full paper is available at
http://www.cs.princeton.edu/~chkim/Research/SEATTLE/seattle.pdf

52

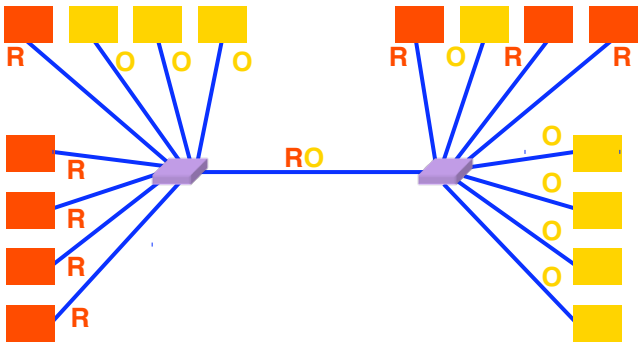
Backup Slides

Solution: Sub-dividing Broadcast Domains

- A large broadcast domain
  → Several small domains
  - Group hosts by a certain rule (e.g., physical location, organizational structure, etc.)
  - Then, wire hosts in the same group to a certain set of switches dedicated to the host group
- People (and hosts) move, structures change …
  - Re-wiring whenever such event occurs is a major pain
- Solution: VLAN (Virtual LAN)
  - Define a broadcast domain logically, rather than physically

54

Example: Two Virtual LANs



Red VLAN and Orange VLAN
Switches forward traffic as needed

Neither VLAN is Satisfactory

- VLAN reduces the amount of broadcast and flooding, and enhances mobility to some extent
  - Can retain IP addresses when moving inside a VLAN
- Unfortunately, most problems remain, and yet new problems arise
  - A switch must handle frames carried in every VLAN the switch is participating in; increasing mobility forces switches to join many, sometimes all, VLANs
  - Forwarding path (i.e., a tree) in each VLAN is still inefficient
  - STP converges slow
  - Trunk configuration overhead increase significantly

More Unique Benefits

- Optimal load balancing via relayed delivery
  - Flows sharing the same ingress and egress switches are spread over multiple indirect paths
  - For any valid traffic matrix, this practice guarantees 100% throughput with minimal link usage
    [*Zhang-Shen et al., HotNets'04/IWQoS'05*]
- Simple and robust access control
  - Enforcing access-control policies at relays makes policy management simple and robust
  - Why? Because routing changes and host mobility do not change policy enforcement points