Floodless in SEATTLE: A Scalable Ethernet Architecture for Large Enterprises

Changhoon Kim, Matthew Caesar, and Jennifer Rexford

Ethernet in Enterprise Nets?

• Ethernet has substantial benefits
  – Simplifies network management, greatly reducing operational expense
  – Naturally supports host mobility
  – Enhances network flexibility

• Why do we still use IP routing inside a single network?
Ethernet Doesn’t Scale!

• Reasons for poor scalability
  – Network-wide flooding
  – Frequent broadcasting
  – Unbalanced link utilization, low availability and throughput due to tree-based forwarding

• Limitations quickly growing with network size

• Scalability requirement is growing very fast
  – 50K ~ 1M hosts

Current Practice

A hybrid architecture comprised of several small Ethernet-based IP subnets interconnected by routers

Sacrifices Ethernet’s simplicity and IP’s efficiency only for scalability

• Loss of self-configuring capability
• Complexity in implementing policies
• Limited mobility support
• Inflexible route selection
Key Question and Contribution

- Can we maintain the same properties as Ethernet, yet *scales* to large networks?
- SEATTLE: The best of IP and Ethernet
  - Two orders of magnitude more scalable than Ethernet
  - Broadcast domains in *any size*
  - Vastly simpler network management, with host mobility and network flexibility
  - Shortest path forwarding

Objectives and Solutions

<table>
<thead>
<tr>
<th>Objective</th>
<th>Approach</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Avoiding flooding</td>
<td>Never broadcast unicast traffic</td>
<td>Network-layer one-hop DHT</td>
</tr>
<tr>
<td>2. Restraining broadcasting</td>
<td>Bootstrap hosts via unicast</td>
<td></td>
</tr>
<tr>
<td>3. Reducing routing state</td>
<td>Populate host info only when and where it is needed</td>
<td>Traffic-driven resolution with caching</td>
</tr>
<tr>
<td>4. Shortest-path forwarding</td>
<td>Allow switches to learn topology</td>
<td>L2 link-state routing maintaining only switch-level topology</td>
</tr>
</tbody>
</table>

* Meanwhile, avoid modifying end hosts*
Network-layer One-hop DHT

- Switches maintain <key, value> pairs by commonly using a hash function $F$
  - $F$: Consistent hash mapping a key to a switch
  - $F$ is defined over the live set of switches
  - LS routing ensures each switch knows about all the other live switches, enabling one-hop DHT operations

- Benefits
  - Fast and efficient reaction to changes
  - Reliability and capacity naturally growing with network size

Location Resolution

<key, val> = <MAC addr, location>
Address Resolution

\[<\text{key, val}> = \langle \text{IP addr, MAC addr}\rangle\]

Traffic following ARP takes a shortest path without separate location resolution

Handling Network Dynamics

- Events **not modifying the set of live switches**
  - E.g., most link failure/recovery
  - LS routing simply finds new shortest paths
- Events **modifying the live set of switches**
  - E.g., switch failure/recovery
  - \( F \) works differently after a change
  - Two simple operations ensure correctness
    - \( |F_{\text{new}}(k)| \neq |F_{\text{old}}(k)| \), owner re-publishes to \( F_{\text{new}}(k) \)
    - Remove any \( <k, v> \) published by non-existing owners
Handling Host Dynamics

- Host location, MAC-addr, or IP-addr can change

Dealing with host mobility

MAC- or IP-address change can be handled similarly

Further Enhancements

- **Goal:** Dealing with switch-level heterogeneity
  - **Solution:** Virtual switches

- **Goal:** Attaining very high availability of resolution
  - **Solution:** Replication via multiple hash functions

- **Goal:** Dividing administrative control to sub-units
  - **Solution:** Multi-level one-hop DHT
Performance Evaluation

• Large-scale packet-level simulation
  – Event-driven simulator optimized for control-plane evaluation
  – Synthetic traffic based on real traces from LBNL
    • Inflated the trace while preserving original properties
  – Real topologies from campus, data centers, and ISPs

• Emulation with prototype switches
  – Click/XORP implementation

Amount of Routing State

SEATTLE reduces the amount of routing state by more than an order of magnitude
Conclusion and Future Work

- **SEATTLE** is a **plug-and-playable** network architecture ensuring both **scalability** and **efficiency**

- Enabling design decisions
  - One-hop DHT tightly coupled with LS routing
  - Reactive location resolution and caching
  - Shortest-path forwarding

- **Future work**
  - Using SEATTLE to improve network security
  - Utilizing indirect delivery for load balancing
  - Optimizations when end hosts can be changed