Reconciling Zero-conf with Efficiency in Enterprises

Chang Kim and Jennifer Rexford
Princeton University
{chkim, jrex}@cs.princeton.edu

ABSTRACT

A conventional enterprise or campus network comprises Ethernet-based IP subnets interconnected by routers. Although each subnet runs with minimal (or zero) configuration by virtue of Ethernet's flat-addressing and self-learning capability, interconnecting subnets at the IP-level introduces significant amount of configuration overhead on both end-hosts and routers. The configuration problem becomes more serious as an enterprise network grows by merging multiple remote sites and by supporting more number of portable end-hosts. Deploying enterprise-wide Ethernet, however, cannot solve this problem because Ethernet bridging does not scale. As an alternative, we propose a scalable and efficient zero-conf architecture (SEIZE) for enterprise networks. SEIZE provides "plug-and-play" capability via flat addressing and allows for scalability and efficiency through a combination of enhanced information dissemination schemes, such as link-state protocols and consistent hashing. SEIZE also supports backward compatibility and partial deployment.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Network**]: Network Architecture and Design – *Distributed Networks*

General Terms

Management, Design

Keywords

Enterprise Networking, Ethernet, IP, Zero-configuration

1. INTRODUCTION

Zero-configuration networking is especially beneficial to enterprise administrators because their resources (human, time, money, skill, etc.) are more limited than in commercial service providers. Considering the increase of volatility by incorporating portable hosts, and the increase of scale by natural growth or by combining multiple remote sites via VPNs, the importance of reducing configuration overhead and enhancing scalability becomes more prominent.

At a first glance, deploying enterprise-wide Ethernet seems promising because of the benefits of self-learning [1] and flat addressing. Ethernet, however, introduces non-negligible side effects from a scalability perspective. Ethernet bridging prohibits a network from employing back-up paths (a.k.a., loops) or, when back-up paths are needed for survivability, requires a spanning tree protocol [2]. Forwarding along a spanning tree does not scale because the entire traffic must share a single forwarding path, regardless of source-destination pairs [3]. Spanning tree based flooding also requires a conservative approach to failover because flooded traffic trapped in a transient loop would cause packet proliferation [4]. This forces a network to endure slow convergence [5].

On the other hand, IP is more efficient in utilizing redundant resources and provides rapid failover, but its hierarchical addressing and routing requires subnet configurations[1]. Using administrative resources for configuring basic networking functions (i.e., reachability provisioning) should be avoided; scarce resources should rather be used for value-creating tasks, such as network design, performance management, etc. Hierarchical addressing is also inefficient because address blocks are not optimally utilized. Poor support for mobility is yet another matter.

SEIZE is an Ethernet-based subnet interconnection architecture which minimizes the dependence on configuration, yet is efficient and scalable. In this paper, we motivate and describe the SEIZE architecture especially in comparison with related work. A short analysis of the architecture follows the description.

2. Related Work

There are a number of solutions proposed recently. Perlman introduced rbridges [2] which can extend Ethernet bridging to interconnect multiple subnets without being confined to a spanning tree. Since this extension employs an additional Ethernet header with a Time-To-Live (TTL) field, a transient loop does not swamp involved rbridges, making fast convergence attainable. An rbridge network delivers traffic along optimal paths because rbridges share a complete network topology using a link-state protocol. An end-host's location and address are discovered by its immediate rbridge via its data traffic. For global synchronization, host information is then disseminated through link-state advertisements. Although pair-wise shortest paths employed by an rbridge network enables more efficient network resource utilization than conventional Ethernet bridging does, disseminating end-hosts' information via link-state protocol introduces significant control overhead when the network grows. End-hosts' volatility aggravates this problem.

Myers et al. proposed a high-level framework for Ethernet to support a million end-hosts [5]. In order to ensure scalability, the architecture forbids flooding (both for unknown unicast destinations and known broadcast/multicast destinations) and requires end hosts to actively register themselves to immediate switches. Switches then disseminate hosts' information via link-state advertisements. Since hosts' information is ubiquitously disseminated, data traffic can be forwarded on a hop-by-hop basis. This scheme, however, requires each end-host's information to be flooded across the network whenever it is discovered. Like an rbridge network, this can introduce unbearably high control overhead and huge forwarding tables. Meanwhile, the architecture requires a modification of the current Ethernet protocol implementations and service models (e.g., ARP and DHCP) as well. Table 1 summarizes the related work and our architecture.

3. SEIZE Architecture and Analysis

We summarize key design features of the SEIZE architecture.

- Ethernet addressing and frame format

  Ethernet addresses (IEEE 802 MAC-48 addresses) are used as unique identifiers of interfaces across an entire network. Since Ethernet addresses are flat and unique, no subnet configurations are required on network nodes [2]. Intra-enterprise mobility also does not require host reconfiguration. IP addresses are given to end-hosts only for external reachability and application-layer compatibility. Conceptually, an entire enterprise appears as a large single IP subnet carrying data end-to-end in the Ethernet format. This guarantees backward compatibility to end-hosts because they can use the same Ethernet interfaces and protocol implementations, whereas rbridges requires a new Ethernet header format.

- Link-state protocol for distributing topology information

  A link-state protocol allows network nodes to unanimously share a complete view of the connectivity among themselves, except in transient periods. Using this topology information, each network node maintains

[1] Although DHCP can automate host configurations, operators still have to configure subnets on interfaces and routing instances. Moreover, configuring DHCP servers also carries configuration overhead.

[2] We intentionally use a generic term, "network node", because the packet delivery entity in our architecture is different both from the conventional bridges and routers.

Table 1. Ethernet Extension Architectures

	Packet format and Addressing	Information Dissemination		Host Discovery	Forwarding
		Topology - core connectivity	End-hosts - location & address		
Rbridge	Ethernet with a new shim hdr, MAC-48	Link-state protocol	List-state advertisement	Network discovers	Tunneling
Myer's et al.	Ethernet, MAC-48	Link-state protocol	Link-state advertisement	Hosts registers	Hop-by-hop
SEIZE	Ethernet, MAC-48	Link-state protocol	Consistent Hash	Network discovers, or hosts registers	Tunneling, or address swapping

shortest paths to all other nodes, making optimal use of topological richness. Unlike both rbridge and Myer's architecture, the link-state protocol does NOT disseminate end-hosts' information for the sake of scalability.

- Hash-based end-host location and address dissemination

  This is the key idea that ensures scalability and efficiency for SEIZE. To maintain (i.e., to register, deregister, and look-up) end-hosts' locations and addresses (MAC and layer-3 addresses), network nodes use *consistent hash* [6] with the end-host's address as the key. That is, each network node maintains only a small portion of the entire end-hosts' information, and the mapping is dictated by the consistent hash. This dissemination scheme ensures that, when an end-host's information needs updating, only a small, constant number (usually two) of network nodes are involved in the process. Additionally, the overhead of dealing with unstable network nodes is also minimized. In order to guarantee availability, an end-host's information should be mapped to more than one node (e.g., the first and the second successor node on a consistent hash ring). ARP requests are also substituted for hash-based look-up operations, since ARP requests result in significant flooding overhead. That is, a network node plays a role of an ARP proxy for the end-hosts residing in its own segment. In order to support this, SEIZE maintains another consistent hash ring for <IP addr, hash value> pairs. Figure 1 illustrates an example host information registration procedure.

- End-host discovery: Compatibility mode vs. Scalability mode

  For backward compatibility, SEIZE can support the conventional "discovery-from-data" mechanism used by Ethernet. This mechanism intrinsically requires flooding to deliver data to an unknown destination, which is unscalable and dangerous[3] with a large number of hosts. As a supplement, upon discovering a new host, SEIZE stores the end-host's information using the hash-based dissemination scheme. This efficiently reduces redundant flooding attempts for unknown or forgotten hosts. On the other hand, when scalability and safety is more demanding a concern than backward compatibility, SEIZE can make use of DHCP as an active host registration scheme. By collocating DHCP servers at edge routers and periodically polling via ARP, SEIZE can effectively trace end-hosts' up and down events. This obviates the need for flooding and significantly enhances the entire network's efficiency and scalability.

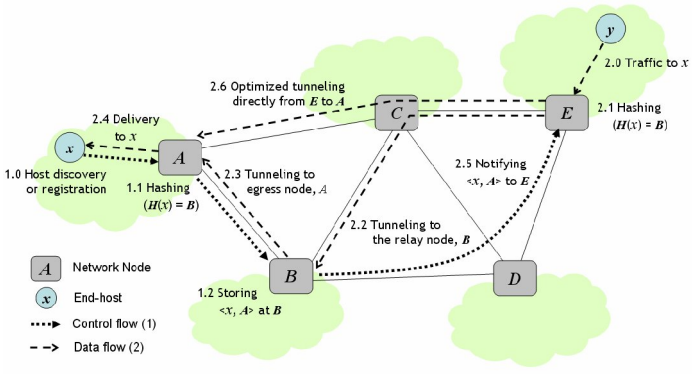

Figure 1. Host information management and traffic delivery

Procedures with sequence number 1 denote the registration process of the end-host *x*'s location information. Procedures with sequence number 2 show the data delivery process from host *y* to *x*. Network nodes (*A* through *E*) run a link-state routing protocol to maintain core connectivity amongst themselves.

- Data traffic delivery: Scalability mode vs. Optimality mode

  Since each network node possesses partial knowledge about end-hosts, when SEIZE needs to deliver a packet to an unknown end-host, it transmits the packet to a "relay" network node that is in charge of maintaining the destination host's location as per the consistent hash. This relay can be accomplished by tunneling or address swapping[4]. For performance's sake, SEIZE can optimize this detour path by letting the initiating network node keep the destination host's location in its cache and use a direct tunnel to the destination. This exercise of trading scalability for optimality can be dynamically adjusted by controlling the number of cacheable paths at each node according to administrative goals. Figure 1 also shows an example case where data is delivered via a relay path or a direct path.

- Securing flooding

  For the sake of service-level compatibility, SEIZE supports broadcast/multicast as well. Flooding, which is a conventional method to support broadcast, proliferates packets when it coincides with a transient loop. As the Ethernet frame format does not carry a TTL, packet proliferation can easily bog down involved network nodes. As a substitute of the conventional physical flooding, SEIZE makes use of pseudo-flooding which systematically replicates a unicast packet along a pre-determined cycle-free graph. Because pseudo-flooding is built on top of unicast, a loop does not proliferate packets. Further, with an intelligent loop detection scheme that does not resort on TTL, SEIZE can aggressively remove packets trapped in a transient loop, providing better loop-evasion performance than IP does.

4. Conclusion

Conventional enterprise architectures require unnecessary configuration overhead, yet are unscalable and sub-optimal. We described and analyzed the SEIZE architecture, which works effectively with minimal configurations, and efficiently with unstable network nodes and a large number of volatile hosts.

5. REFERENCES

[1] IEEE Std 802.1D – 2004, IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges, IEEE Computer Society and ANSI.

[2] R. Perlman, "Rbridges: Transparent Routing," *in Proceedings of Infocom 2004*, Hong Kong, March 2004.

[3] T. Rodeheffer, C. Thekkath, and D. Anderson, "SmartBridge: A Scalable Bridge Architecture," *in Proceedings of ACM SIGCOMM*, pp. 205-216, August 2000.

[4] R. Perlman, *Interconnections: Bridges, Routers, Switches, and Internetworking Protocols*, Addison-Wesley Professional Computing Series, 1999.

[5] A. Myers, T. S. Eugene Ng, and H. Zhang, "Rethinking the Service Model: Scaling Ethernet to a Million Nodes," *in Proceedings of HotNets III*, November, 2004.

[6] D. Karger, E. Lehman, T. Leighton, M. Levine, D. Lewin, and R. Panigrahy, "Consistent Hashing and Random Trees: Tools for Relieving Hot Spots on the World Wide Web," *in Proceedings of ACM Symposium on Theory of Computing*, pp. 654-663, 1997.

[3] Some malicious attacks, such as MAC spoofing and ARP flooding, become more devastating as more end-hosts get involved.

[4] For example in figure 1, when E handles a frame from Y, E can put A's address in the frame's destination field, saving x's address in the source field. When the frame arrives at A, x's address is restored into the destination field.