End-to-end Reliability in Enterprise Ethernet Networks

Paul Willmann

ABSTRACT

Switched Ethernet technology has several benefits that make it attractive as a substrate for network enterprise deployment. Ethernet is virtually configuration-free and delivers economical high performance, and its ubiquity ensures compatibility with existing systems. However, switched Ethernet can suffer from poor reliability during topological changes, such as link failure. Such reliability issues could be catastrophic in large-scale enterprise settings, with hundreds or thousands of nodes affected. Prior research focused on layer-2 convergence properties after topological changes in switched Ethernet, but higher-layer communications can generate additional traffic during periods of unreliability that can interfere with convergence. This work establishes an infrastructure for studying cross-layer interactions during periods of Ethernet unreliability with the goal of identifying mechanisms for improving end-to-end convergence time to steady state behavior.

1 Introduction

Switched Ethernet technology has several benefits that make it attractive as a substrate for large-scale enterprise network deployment. Plug-and-play automatic configuration reduces administrative overhead, and modern Ethernet technology delivers economical high performance. Furthermore, Ethernet's ubiquity among deployed systems guarantees compatibility of existing hardware and software with new Ethernet-based networks.

One of the most common uses for Ethernet has been as a layer-2 technology for connecting workgroup or office local area networks (LANs), but Ethernet's numerous advantages have driven its adoption in new areas as well. Service providers are developing metropolitan-wide Ethernet networks that are much larger in scale than typical LANs. Data centers are leveraging commodity switched Ethernet and iSCSI to implement storage area networks (SANs), replacing costly special-purpose networks such as Fibre Channel.

However, switched Ethernet has undesirable reliability characteristics with respect to network topology changes. In emerging application domains, Ethernet reliability problems could have much farther reaching effects; large-scale Ethernet networks expose more users to service interruption from a single topology change, and any interruption in service to a SAN could have organization-wide effects. Hence, these periods of unreliability must be as limited in duration as possible. Though careful examination of reliability problems with switched Ethernet continues, the interaction among all network layers during periods of unreliability is not fully understood. This work seeks to identify properties of other layers that may exacerbate or even prevent recovery during periods of unreliability and to develop cross-layer mechanisms that will improve the end-to-end response to periods of temporary unreliability.

A primary source of unreliability in switched Ethernet is its mechanism for determining the network topology after a topology change. To prevent cycles in the network, each bridge participates in a distributed Spanning Tree Protocol (STP) that develops a logical spanning tree topology for the underlying physical network. When the topology changes (e.g., a link is added, a link goes down, or a bridge fails), bridges in the network rediscover the spanning tree topology. During this period of rediscovery, the entire forwarding topology is unstable, and packet delivery may be highly unreliable. Furthermore, forwarding loops may exist in the network, leading to severe congestion and further delaying topological convergence.

The Rapid Spanning Tree Protocol (RSTP) aims to improve STP's convergence properties by having each bridge keep local information about available alternate paths, thus enabling them to negotiate quick recovery from link failure rather than having to rediscover the entire topology. However, this information about alternate paths may be invalid after a link failure and may create a race condition with new, valid path information; this race condition can take tens of seconds to resolve [6].

However, recovery times in large-scale enterprise switched Ethernet networks can be several orders of magnitude longer than tens of seconds; anecdotally, networks at Rice University can take hours to recover after a power outage, even with all equipment operating correctly. With respect to recovery via RSTP, several cross-layer issues warrant examination. During the "count-to-infinity" race condition that delays RSTP convergence, forwarding loops exist within the network topology; if upper network layers persist in generating traffic during the "count-to-infinity" condition, severe congestion may result. It is not clear how busy networked workstations behave during network unreliability; TCP-based services will utilize a congestion avoidance mechanism and back off exponentially, but many services (e.g., ARP, DHCP, NIS, and NFS) across many different layers are not based on TCP. Some of these lower-priority services may not have robust support for congestion avoidance, which would further exacerbate RSTP convergence. Rather than requiring congestion avoidance in every layer, a simple feedback mechanism from the Ethernet layer available to all higher layers that indicates an unreliable condition may be sufficient to cease further network output and thus help improve recovery time.

In addition to end-to-end issues that may interfere with RSTP convergence, other network layers usually do not have any notion of instantaneous resumption of service. Window-based transport protocols (e.g., TCP) may take some time to rescale their windows to optimum size. More importantly, there may be a compound effect of each node on the network rescaling each of their connections simultaneously. Similar to layer-2 feedback for improving RSTP convergence, layer-2 feedback could indicate a "ready-to-resume" condition that other network layers could leverage to quickly reconverge to the prior steady state. Understanding how this reconvergence to the prior steady state occurs may lead to important insights for delivering improved end-to-end reliability for enterprise-class applications.

2 Related Work

Several previous works have addressed the reliability shortcomings of switched Ethernet's spanning tree protocols, though most are motivated by these protocols' lack of simultaneous utilization of alternate, redundant links. Rbridges replaces spanning trees with link-state routing in bridges; temporary loops are still allowed, albeit to a lesser extent than in RSTP's count-to-infinity scenario [8]. SmartBridges also eliminate spanning trees, but they require freezing the network during topological changes [9]. Elmeleegy et al. thoroughly examine RSTP's count-to-infinity problem and evaluate its convergence properties; however, this evaluation considers only bridge-to-bridge packet communication and does not consider the effects of data traffic among other network layers [6]. Regardless of the method employed, all schemes have a nonzero period of network unreliability during topology changes.

Some prior research has examined providing link congestion and topology information among network layers. Even in the most versatile of these implementations, link information feedback flows only to the TCP layer; as previously discussed, enterprise workstations may use network services (such as NFS version 3 and prior) that do not feature congestion control mechanisms similar to TCP's. Aweya et al. propose internetworking ATM networks and Ethernet networks, converting ATM's layer-2 available bit rate information to Ethernet's layer-2 802.3x flow control information, but this information does not flow upward to other network layers [2]. Kalampoukas et al. examine explicit manipulation of TCP window sizes by an intermediary ATM network to reflect its available bandwidth, but this congestion information flows only to TCP connections [7].

Chandran et al. developed a mechanism similar to the one proposed that provides explicit notification of link failure in ad-hoc wireless networks for improving TCP performance [4]. In that study, TCP connection state was frozen upon link failure notification and then unfrozen immediately after link re-establishment, thus avoiding TCP slow-start and performance degradation associated with misinterpreting transient packet loss as congestion. While that work is similar in technique to the proposed research, Chandran et al. consider a medium (e.g., wireless) in which retransmits are not harmful to other nodes or to the stability of the rest of the network. Therefore, the interplay between various nodes in the network and the eventual convergence of the network during a topology change is not exposed. Furthermore, Chandran et al. consider feedback only to the TCP layer.

3 Methodology

The goal of this research is to examine the cross-layer interactions in enterprise network workloads during periods of RSTP instability and convergence. Emulab is a network emulator and testbed for running experiments on user-configurable network topologies [11]. The Emulab facilities at the University of Utah provide the user with up to 328 PC nodes that can be connected in a user-defined topology; each node has four 100 Mb/s network interface cards (NICs) that may be connected to any other node, and a fifth NIC serves as a control-plane interface for remote access. The links themselves can be manipulated via configurable scripts during experiments.

However, Emulab does not provide bridge or router capabilities for interconnecting nodes. If the user wishes to connect multiple nodes into a single switched Ethernet network, then intermediate nodes must be configured as software Ethernet bridges. Unfortunately, no OS-level software bridges currently implement RSTP. Linux implements STP via its in-kernel Ethernet bridging facilities [1]. Rather than rewriting Linux bridging to support the intracacies of RSTP, the bridge interface can be redefined and plugged into `rstplib`, an existing open-source RSTP simulator [10]. The resultant switch would have higher latency than a commodity RSTP-enabled Ethernet switch, but the convergence trends should mimic those of networks using commodity hardware.

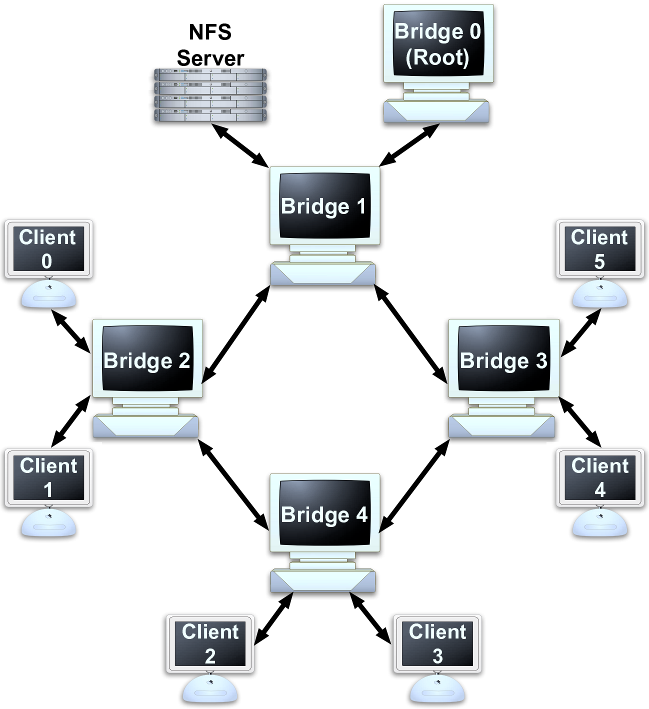

Figure 1: A Simple Ring Topology Realizable with Emulab.

Hence, an interesting experimental configuration might have edge nodes acting as clients or servers in an enterprise network in addition to nodes that act as 4-port switches. Emulab facilities could be used to mimic an enterprise switched Ethernet network with tens of nodes connected by 4-port switches and featuring redundant links; a simple ring topology and a scripted link failure to the root bridge can cause a count-to-infinity situation [6]. Figure 1 depicts a topology realizable with Emulab's facilities that would exhibit the count-to-infinity behavior if Bridge 0 failed.

A realistic workload might feature end nodes configured via DHCP, running NIS for login authentication, and each accessing a remote file server while running a filesystem or database access benchmark, such as Iozone or Bonnie++ [3, 5]. Such an experimental testbed would provide the facilities for gathering profiles of packets and measuring end-to-end data throughput during network topology instability, allowing further exploration of cross-layer interactions during such instability. Further, Emulab enables custom network layer profiling and modifications because its facilities permit arbitrary software to be run on the nodes.

4 Expected Contributions

This research should contribute a configurable infrastructure for examining end-to-end behavior in RSTP switched Ethernet networks during topological changes. This infrastructure should enable profiling of what is happening in various network layers before, during, and after the topological change. Using these profiles, I hope to identify cross-layer interactions that may exacerbate or prevent RSTP convergence, to develop strategies for preventing such interactions, and to quickly recover to performance levels achieved prior to the induced unreliability.

5 Acknowledgements

I wish to thank Khaled Elmeleegy for sharing his insights regarding the current status of both Linux Ethernet bridging and the rstplib project.

REFERENCES

[1] Linux Ethernet Bridging. http://bridge.sourceforge.net, 2005. Accessed September, 2005.

[2] J. Aweya, M. Ouellette, and D. Y. Montuno. Interworking of Switched Ethernet and ATM Flow Control Mechanisms. *International Journal of Network Management*, 12(6):357–366, 2002.

[3] D. Capps and W. Norcott. Iozone Filesystem Benchmark. http://www.iozone.org, 2005. Accessed September, 2005.

[4] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash. A Feedback Based Scheme for Improving TCP Performance in Ad-Hoc Wireless Networks. In *Proceedings of the The 18th International Conference on Distributed Computing Systems (ICDCS '98)*, page 472, Washington, DC, USA, 1998. IEEE Computer Society.

[5] R. Coker. Bonnie++ Benchmark Suite. http://www.coker.com.au/bonnie++, 2001. Accessed September, 2005.

[6] K. Elmeleegy, A. L. Cox, and T. S. E. Ng. On Count-to-Infinity Induced Forwarding Loops in Ethernet Networks. In Submission, Sept. 2005.

[7] L. Kalampoukas, A. Varma, and K. K. Ramakrishnan. Explicit Window Adaptation: A Method to Enhance TCP Performance. *IEEE/ACM Transactions on Networking (TON)*, 10(3):338–350, 2002.

[8] R. Perlman. Rbridges: Transparent Routing. In *Proceedings of IEEE Infocom 2004*, pages 1211–1218, Mar. 2004.

[9] T. L. Rodeheffer, C. A. Thekkath, and D. C. Anderson. SmartBridge: A Scalable Bridge Architecture. In *Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '00)*, pages 205–216, New York, NY, USA, 2000. ACM Press.

[10] A. Rozin. Rapid Spanning Tree (RSTP, 802.1w) Library and Simulator. http://sourceforge.net/projects/rstplib, 2002. Accessed September, 2005.

[11] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. An Integrated Experimental Environment for Distributed Systems and Networks. In *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation*, pages 255–270, Boston, MA, Dec. 2002.