Using SCTP Multihoming for Fault Tolerance and Load Balancing*

Armando L. Caro Jr. and Janardhan R. Iyengar
Paul D. Amer, Gerard J. Heinz, Randall R. Stewart†

http://pel.cis.udel.edu

Protocol Engineering Lab
CIS Department, University of Delaware
{acaro, iyengar, amer, heinz}@cis.udel.edu

† Cisco Systems Inc.
rrs@cisco.com

1 SCTP Overview

Mission critical systems rely on redundancy at multiple levels to provide uninterrupted service during resource failures. Such systems when connected to IP networks often deliver network redundancy by multihoming their hosts. A host is multihomed if it can be addressed by multiple IP addresses. An endpoint's IP address can become inaccessible, possibly due to an interface failure, severe congestion, or due to BGP's slow route convergence around path outages. Redundancy at the network layer allows a host to be accessible even if one of its IP addresses becomes unreachable; packets can be rerouted to one of its alternate IP addresses.

TCP does not support multihoming. Any time either endpoint's IP address becomes unreachable, TCP's connection will timeout and abort, thus forcing the upper layer to recover. The recovery delay can be unacceptable for mission critical applications such as IP telephony, IP storage, and military battlefield communications. To address TCP's shortcoming, the Stream Control Transmission Protocol (SCTP) has been designed with fault tolerance in mind. SCTP supports multihoming at the transport layer to allow SCTP associations to remain alive even when an endpoint's IP address becomes unreachable.

2 Adaptive Failover Mechanism

SCTP has a built-in failure detection and recovery system, known as failover, which allows associations to dynamically send traffic to an alternate peer IP address when needed. SCTP's failover mechanism is static and does not adapt to application requirements or network conditions.

Network dynamics however, vary greatly among different networks and hence, the heuristics used for determining failover should be adjusted accordingly. Applications also have different requirements that the failover mechanism should cater to. For example, SS7 signalling applications require that failovers take no longer than 800 ms. On the other hand, a file transfer may be more concerned with the total transfer time. Therefore, we argue that network fault tolerance should cope with dynamic network conditions and varying application needs.

We have developed a two-level (alpha-beta) threshold mechanism for SCTP which provides added control over failover actions. We have formally specified and modeled our failover mechanism, and are currently investigating the relationships between failover thresholds and network parameters, such as round trip times, packet loss rates, etc. From these relationships, we will develop an adaptive failover mechanism for SCTP.

3 End-to-End Load Balancing

SCTP provides for application-initiated changeovers so that the sending application can change the sender's primary destination address, thus moving the outgoing traffic to a potentially different path. Although the motivations for providing a changeover mechanism are different, it is not difficult to envision application developers using this feature for load balancing at the application layer.

With the provisioning for multihoming in SCTP, we believe that end-to-end load balancing can be performed at the transport layer. Being better informed than the application layer about the end-to-end paths, the transport layer can perform fine-grain load balancing. We foresee issues during load balancing in areas such as congestion control and loss detection and recovery. These issues also suggest that the transport layer be involved.

We are currently looking at issues due to changeover. We have uncovered a problem that results in cwnd overgrowth during changeover. Analysis shows that this problem may not be a corner case but may occur under various network and changeover conditions. We propose the Rhein algorithm and the Changeover Aware Congestion Control (CACC) Algorithms as solutions to the problem. In the future, we will investigate (1) end-to-end techniques for shared bottleneck detection to aid in performing correct congestion control during load balancing, and (2) algorithms for scheduling traffic on the multiple paths.

* Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.