Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Computer Communications 29 (2006) 1798–1810
computer communications
www.elsevier.com/locate/comcom

Retransmission policies for multihomed transport protocols☆

Armando L. Caro Jr. [1,a], Paul D. Amer [b,*], Randall R. Stewart [c]

[a] Internetwork Research Department, BBN Technologies, 10 Moulton St, Cambridge, MA 02138, USA
[b] Protocol Engineering Lab, CIS Department, University of Delaware, Newark, DE 19716-2586, USA
[c] Internet Technologies Division, Cisco Systems, Suite 200, 4875 Forest Drive, Columbia, SC 29206

Received 8 March 2005; received in revised form 6 October 2005; accepted 7 October 2005
Available online 28 November 2005

Abstract

We evaluate three retransmission policies for transport protocols that support multihoming (e.g. SCTP). The policies dictate whether retransmissions are sent to the same peer IP address as the original transmission, or sent to an alternate peer IP address. Each policy presents tradeoffs based on the paths' bandwidth, delay, loss rate, and IP destination reachability. We find that sending all retransmissions to an alternate peer IP address is useful when the primary IP address becomes unreachable, but often degrades performance in non-failure scenarios. On the other hand, sending all retransmissions to the same peer IP address as the original transmission reverses the tradeoffs. We balance the tradeoffs by proposing a hybrid policy that sends fast retransmissions to the same peer IP address as the original transmission, and sends timeout retransmissions to an alternate peer IP address. We show that even with extensions which we proposed to improve the policies' performance, the hybrid policy is the best performing policy in failure and non-failure scenarios.

Keywords: SCTP; Transport protocols; Multihoming; Failover; Failure detection; Retransmission policy

1. Introduction

A host is multihomed if it can be addressed by multiple IP addresses, as is the case when the host has multiple network interfaces. Multihoming can be expected to be the rule rather than the exception in the near future as cheaper network interfaces and Internet access motivate content providers to have simultaneous connectivity through multiple ISPs, and more home users install wired and wireless connections for added flexibility and fault tolerance. Furthermore, wireless devices may be simultaneously connected through multiple access technologies, such as wireless LANs (e.g. 802.11) and cellular networks (e.g. GPRS, CDMA).

The current transport protocol workhorses, TCP and UDP, do not support multihoming; TCP allows binding to only one network address at each end of a connection. When TCP was designed, network interfaces were expensive components, and hence multihoming was beyond the ken of research.

Two recent transport layer protocols, the Stream Control Transmission Protocol (SCTP) [23,9] and the Datagram Congestion Control Protocol (DCCP) [18] support multihoming at the transport layer. The motivation for multihoming in DCCP is mobility [17], while SCTP is driven by a broader and more generic application base—fault tolerance. We use SCTP in our experiments primarily because of its relative maturity and our focus on fault tolerance, but we believe the results and conclusions presented in this paper apply in general to reliable SACK-based transport protocols that support multihoming.

SCTP allows binding of one transport layer *association* (SCTP's term for a connection) to multiple IP addresses at each end of the association. SCTP's n to m binding allows a multihomed sender with n interfaces to send to any of a multihomed receiver's m destination addresses. For example, an SCTP multihomed association between hosts A and B in

☆ *Disclaimer*. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the US Government.

* Corresponding author. Tel.: +1 302 831 1944; fax: +1 302 831 8458.
E-mail addresses: acaro@bbn.com (A.L. Caro), amer@cis.udel.edu (P.D. Amer), rrs@cisco.com (R.R. Stewart).

[1] This research results from the author's PhD dissertation, while with the Protocol Engineering Lab, CIS Department, University of Delaware. Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the US Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. Supported in part by the University Research Program of Cisco Systems, Inc.

0140-3664/$ - see front matter
doi:10.1016/j.comcom.2005.10.017

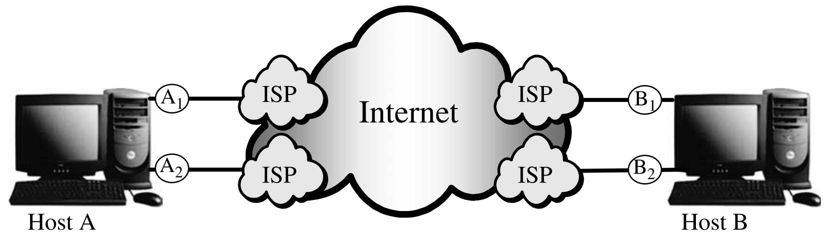

Fig. 1. Example multihoming topology.

Fig. 1 could be bound to both IP addresses at each host: ({A_1,A_2}, {B_1,B_2}). Such an association allows data transmission from host A to host B to be sent to either B_1 or B_2.

Currently, SCTP uses multihoming for fault tolerance purposes only, and not for concurrent multipath transfer [14]. Each endpoint chooses a single peer IP address as the primary destination address to transmit new data during normal transmission. If the primary destination address becomes unreachable, the SCTP sender detects the failure, and *fails over* to using an alternate destination address without requiring action by the user or application layer.

When data is lost, a sender uses an alternate destination address for retransmissions. SCTP's current retransmission policy [23] states that "when its peer is multihomed, an endpoint SHOULD try to retransmit [data] to an active destination transport address that is different from the last destination address to which the [data] was sent." This policy, which we refer to as *AllRtxAlt (All Retransmissions to Alternate)*, attempts to improve the chance of success by sending all retransmissions to an alternate destination address [22]. The underlying assumption is that loss indicates either that the network path to the primary destination is congested, or the primary destination is unreachable. Thus, retransmitting to an alternate destination might avoid yet another loss of the same data.

We show that this policy actually degrades performance in many circumstances. We explore two alternative retransmission policies and find that the best policy, for both failure and non-failure scenarios, is to send (a) fast retransmissions to the primary destination, and (b) timeout retransmissions to an alternate destination. We show that this hybrid policy performs best when combined with two enhancements: our Multiple Fast Retransmit algorithm, and either timestamps or our Heartbeat After RTO mechanism. The Multiple Fast Retransmit algorithm reduces the number of timeouts. Timestamps and the Heartbeat After RTO mechanism both improve performance when timeouts are common by providing extra RTT measurements and maintaining low RTO values.

This paper combines and extends results published by the authors in three incremental conference publications [5–7], thereby documenting the complete development of this research. Section 2 demonstrates the problem with SCTPs current retransmission policy (AllRtxAlt) by comparing it to an alternative policy, *AllRtxSame (All Retransmissions to Same)*. Section 3 introduces and evaluates a third hybrid policy, *FrSameRtoAlt (Fast Retransmissions to Same, Timeouts to Alternate)*, which attempts to balance the tradeoffs between AllRtxAlt and AllRtxSame. Section 4 introduces and evaluates three extensions to further improve the performance of the three policies. Section 5 compares the policies' performance with their best extensions in non-failure scenarios, and Section 6 compares them in failure scenarios. Section 7 concludes the paper.

2. AllRtxAlt's problem

AllRtxAlt is the retransmission policy currently specified for SCTP in RFC2960. This policy attempts to bypass transient network congestion and path failures by sending all retransmissions to an alternate destination. Intuitively, we would expect that sending retransmissions to an alternate path would be beneficial, particularly when the alternate path's quality is better (i.e. higher bandwidth, lower delay, and/or lower loss). Similarly, when the alternate path's quality is worse, we expect sending retransmissions to the same destination as their original transmission should provide better performance. To test these hypotheses, we evaluate the performance of AllRtxAlt and the AllRtxSame policy—send all retransmissions to the same destination as their original transmission [6].

2.1. Analysis methodology

We evaluate the retransmission policies using University of Delaware's SCTP module [8] for the ns-2 network simulator [3]. Fig. 2 illustrates the network topology simulated: a dual-dumbbell topology whose core links have a bandwidth of 10 Mbps and a one-way propagation delay of 25 ms. Each router, R, is attached to five edge nodes. One of these five nodes is a dual-homed node for an SCTP endpoint, while the other four are single-homed and introduce cross-traffic that creates loss for the SCTP traffic.

The links to the dual-homed nodes have a bandwidth of 100 Mbps and a one-way propagation delay of 10 ms. The single-homed nodes also have 100 Mbps links, but their propagation delays are randomly chosen from a uniform distribution between 5 and 20 ms. The end-to-end one-way propagation delays range between 35 and 65 ms. These delays roughly approximate reasonable Internet delays for distances such as coast-to-coast of the continental US, and eastern US to/from western Europe. Also, each link (both edge and core) has a buffer size twice the link's bandwidth-delay product.

Our configuration has two SCTP endpoints (sender A, receiver B) on either side of the network, which are attached to the dual-homed edge nodes. A has two paths, labeled primary