A NEW END-TO-END FAILOVER MECHANISM WITH TRANSPORT LAYER MULTIHOMING

JIEMIN LIU, SHAOHUI CHENG, YUHUI ZHAO AND YUAN GAO

Abstract Using the application of bulk data transfer, we investigate current end-to-end temporary failover mechanism, permanent failover mechanism and relative threshold’s design for transport protocols that support multihoming (e.g., SCTP) again. We find that they have a common defect, both of them neglect idle-multipath’s initiative utilization when they are interested in fault tolerance and failover. Both mechanisms increases the possibility of “global failover synchronization”. Based on this, we first present a new approach to end-to-end failover mechanism (i.e., Circular Failover mechanism based on three-level threshold). Such a mechanism further improved failover’s performance by a new threshold design, realized idle-multipath’s initiative utilization and “logical traffic balancing” by the other new threshold design. We demonstrate that such a mechanism can further improve failover mechanism’s comprehensive performance than existing well-known failover mechanism.

Key Words, Transport Layer Multihoming, Circular Failover mechanism, Path.Max.Tomeout (PMT), Path.Max.Slow-start (PMS), Logical Traffic Balancing

1 Introduction

A host is multihomed if it can be addressed by multiple IP address. Multihoming can be expected to be the rule rather than the exception in the 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. So it is very important to research and adopt reasonable and high performance end-to-end failover mechanism.

The transport layer is in the best position to detect failure (i.e., loss of connectivity) and make failover decisions, and it also is the lowest layer responsible for end-to-end QoS.

TCP does not support multihoming. Stream Control Transport Protocol (SCTP) [1], [2], [3] and the Datagram Congestion Control Protocol (DCCP) support multihoming at the transport layer. However, SCTP currently uses Multihoming for fault tolerance purpose only, and not for concurrent multipath transfer [4]. The motivation for Multihoming in DCCP is mobility, while SCTP is driven by a broader and more generic application base – fault tolerance.

Our research focuses on transport layer Multihoming techniques. Our goal is to find a new end-to-end failover mechanism to further improve end-to-end fault tolerance (i.e., failover performance) and idle-multipath’s initiative utilization (i.e., logical traffic balancing). Logical traffic balancing is a new concept first presented in this paper, it means gain traffic balancing by idle-multipath’s initiative circular-utilization in time’s order, and eliminate the possibility of “global failover synchronization”.

We use SCTP in our experiment because of its relative maturity and our focus on a
fault tolerance and idle-multipath's initiative circular-utilization (i.e., logical traffic balancing), but the results and conclusions presented in this paper apply in general to reliable SACK-based transport protocols that support multihoming.

Recently, SCTP use temporary failover mechanism and a tunable failover threshold that RFC2960 recommends should be set to a conservative value of six consecutive timeouts that translates to failure detection time of at least 63 seconds-unacceptable for many applications. The defect of such mechanism is long detection time and slowdown sending rate. Permanent failover mechanism [5] can improve performance (e.g., shorten detection time and avoid sending rate slowdown), but it ignored the idle-multipath’s initiative circular-utilization and “global failover synchronization” caused by synchronized permanent failover. We present a circular failover mechanism to further improve performance, such mechanism can pre-failover (change transmission path) quickly, avoid sending rate slowdown, change transmission mode of multipath’s passive circular-utilization (i.e., change transmission path only when failure occurs) and eliminate the possibility of global failover synchronization.

Section 2 describes SCTP’s current failover mechanism. Section 3 introduced permanent failover mechanism. Section 4 presents and evaluates circular failover mechanism and it’s performance. We conclude the paper in section 5.

2 SCTP’s Current Failover Mechanism

SCTP’s Current Failover Mechanism (i.e., temporary Failover) is based on a one-level threshold, called Path.Max.Retrans (PMR). Figure 1 specifies SCTP’s current failover mechanism for n destinations. The association begins in phase I, where destination D_i is the primary destination, D_j is in the active state, and all new data are sent to D_i. When D_i fails, “failover” occurs and the association moves into Phase II.

In Phase II, D_i remains the primary destination, but in a failed state; all new data are redirected to an alternate destination, D_j. If more than one alternate destination address exists, RFC2960 leaves the alternate destination selection method unspecified. If D_j’s error count should exceed PMR, a failover occurs to yet another alternate destination and the association stays in Phase II.

While in Phase II, the sender explicitly probes primary destination, D_i, with periodic heartbeats. If D_i ever responds (i.e., recovers), failover is cancelled and the association returns to Phase I.

Failure detection time depends on three tunable parameters, which RFC2960 recommends to be set as: (1) minimum RTO = 1s, (2) maximum RTO = 60s, and PMR = 5. Using these defaults, the first timeout towards failure detection takes 1s in the best case. Then, the exponential back-off procedure doubles the RTO on each subsequent timeout towards failure detection. With RFC2960’s current recommended PMR = 5, six consecutive timeouts are needed to detect failure, taking at least $1 + 2 + 4 + 8 + 16 + 32 = 63s$. In the worst case, the first timeout takes the maximum of 60s, and failure detection requires $6*60 = 360s$! This is unacceptable for many applications.

Reducing PMR can decreases failure detection time, but increase the possibility of spurious failover, where a sender mistakenly concludes a failure has occurred, But spurious failover do not degrade performance, and often actually improve goodput regardless of the path’s RTT and loss rate. PMR = 3 is robust enough for the Internet. This setting translates to a 15 second failover time, and is robust for all “lossless” paths and the average “lossy” path [6].
3 Permanent Failover Mechanisms

When failover are temporary, traffic migrates back to the primary path when it recovers. This migration throttles the sending rate, because the sender returns to slow start’s cwnd of one MTU. To avoid this slowdown, permanent failover mechanism for SCTP has been proposed in [6], [7]. It is based on a two-level threshold (i.e., PMR, CPT - Change Primary Threshold). Once failover occurs, the sender can make the failover permanent (i.e., change the primary destination) when more than CPT heartbeat probes sent to the primary destination timeout.

The specification for permanent failover, shown in Figure 2, adds some new transition and status to finite state machine in Figure 1. While the association is in Phase II or III, if the primary destination’s CPT threshold is exceeded, the primary destination is changed to the alternate destination currently in use. In Phase II, the association returns to Phase I with the new primary destination. In Phase III, however, the association remains in Phase III when a new primary destination is set; that is, changing the primary destination does not change the status of any destination, and thus the association remains in dormant state.

![Fig. 2. FSM for permanent failovers](image_url)

4 Circular Failover Mechanism

We design and present a new end to end failover mechanism, called circular failover mechanism, it is a three-level threshold failover mechanism based on Path.Max.Timeout.
Our goal is design such a mechanism that can further improve failover’s performance (controlled by PMT threshold), especially multipath’s initiative circular-utilization and eliminate the possibility of global failover synchronization (controlled by PMS threshold).

4.1 The Main Design Ideas

We find that sent data to a destination permanently increased the possibility of Congestion that is the main cause of destination failure for all association build between many multihoming user. Initiative abandon to a permanent useful destination by some policy (i.e., set up a threshold in this paper) is beneficial to eliminate congestion. Initiative change destination does not led to slowdown. During sent data to a destination, a slow-start and congestion avoidance will be exit and a next cycle of slow-start and congestion-avoidance will be begin when a timeout occurs. A new cycle of slow-start and congestion-avoidance processes in which destination does dot affect data transmission. After some cycles of slow-start and congestion-avoidance controlled by a threshold, sender initiative send data to another destination. We should set up a threshold to control the cycle of slow-start and congestion-avoidance. The best value of threshold relay on simulation and balance another factor.

Both temporary failover and permanent failover are send all new data to alternate path passively when failure occurs only, such two failover mechanisms lacks idle-multipath’s initiative circular-utilization. All new data always send to the destination (i.e., the path always in use) if both PMT and PMS are not exceeded, another path (or destination) always in idle. Such mechanism can’t be beneficial to the logical traffic balancing. This also is important potential factor that led to global failover synchronization. AQM techniques does not thoroughly eliminate global synchronization [6] includes global failover synchronization. Based on this, in order to improve performance of reasonable multipath usage and avoid dreadful failover synchronization phenomenon, we proposed a new end-to-end failover mechanism. We present a major potential change to SCTP – the concept of “circular permanent failovers” using two new threshold of PMT and PMS. PMT is the first new threshold that led to probe-over, which is pre-failover (i.e., PMT is the pre-PMR). It also is a count of consecutive timeout. All new data sent to alternate destination when a destination’s PMT is exceeded (i.e., Even if PMR is not exceeded, all new data sent to alternate destination in advance when PMT is exceeded until this destination responds). Primary destination is probed with heartbeats simultaneously, when Di’s probes cause PMR to be exceeded, the primary destination is failed. PMT should be set a value that is less than PMR. We evaluate different PMT setting find that PMT = 0 (i.e., a single timeout) is the best choice. Here, we keep the second threshold PMR. PMS is the third threshold that should increments a count when the destination keeps reachability (i.e., the path is in use) and sender returns to slow start’ cwnd of one MTU [8], so PMS is a count of non-consecutive timeout in essentially. Even if the destination or path is not failure (i.e., the destination’s or path’s PMT or PMR is not exceeded), all new data sent to another destination (or another path) when PMS is exceeded, PMS is more than PMR. multipath’s initiative circular-utilization realized the logical traffic balancing and eliminate the possibility of global synchronization

The best value of PMT ‘s setting and PMS’s setting is explained later.

4.2 FSM for Circular Failover Mechanism
Figure 3 specifies our proposed failover mechanism for \( n \) destinations. The association begins in Phase I, with \( D_i \) as the primary destination, \( D_j \) in the active state, and all new data sent to \( D_j \). The association transitions to Phase II (i.e., it is a middle state between association’s beginning state and failure state) when PMT exceed, where \( D_i \) remains the primary destination, \( D_j \) is probed with heartbeats, and new data are sent to an alternate destination (\( D_i \)). If \( D_i \)’s probes cause PMR to be exceeded, the association transitions to Phase III, where \( D_i \) is marked failed. While in Phase II or Phase III, each exceeded PMT or PMS redirects new data to different destination (skipping failed destinations). Any time \( D_i \) responds, the association returns to Phase I.

![Fig. 3. FSM for circular failover mechanism](image)

**4.3 Performance Evaluation**

**4.3.1 Simulation Setup**

We evaluate different PMT setting and PMS setting using the University of Delaware’s SCTP module\[11\] for the ns-2 network simulator\[12\]. Figure 4 illustrates the network topology. The multihomed sender, A has two paths (labeled Primary and Alternate) to the multihomed receiver, B. The primary path’s core link and alternate path’s core link has a 10Mbps bandwidth, a 30ms one-way delay and a 0-10% loss rate. Each router, R, uses drop-tail queuing and is attached to a dual-homed node (A or B) via an edge link with 100Mbps bandwidth and 10ms one-way delay. The end-to-end RTT are 100ms, which sample reasonable delays on the Internet today.

Note that we do not simulate different one-way delay and bandwidths in primary path’s core link and alternate path’s core link. But it does not affect our results and conclusions. We introduce uniform loss on these paths (0–10% each way) at the core links. We realize that using cross-traffic to cause congestion would more realistically simulate loss, but we found the simulation time for such a technique became impractical. On the other hand uniform loss is a simple, yet sufficient model to provide accurately detecting failure. To evaluate if Figure 4’s loss model was reasonable, we compared representative simulations using a cross-traffic model to produce self–similar, burst traffic. Although the absolute results differed for those examples compared, relative relationships remained consistent – leading to the same conclusions. We therefore proceed with the simpler uniform loss model.

In our simulation, the sender uses a hybrid retransmission policy\[10\] than specified in RFC2960. This policy has been shown to perform better, and has been proposed to the
IETF as a change to SCTP. In our simulation, we used Multiple Fast Retransmit Algorithm [12] to reduce the number of timeouts.

To observe long term average, we simulate 100MB file transfer with PMT = {0, 1, 2, 3, 4, 5} and PMS = {0, 1, 2, 3, 4, 5}. In this study, no link or interface failures are introduced; hence, all failovers that do occur are spurious. Each simulation has four parameters;
1) primary path’s loss rate
2) alternate path’s loss rate
3) PMT setting
4) PMS setting

![Simulation network topology](image)

**Fig. 4. Simulation network topology**

### 4.3.2 Different PMT Setting’s Result and Discussions

We collected results for 0-10% loss on the primary and alternate paths, but due to space constraints in this paper. We do not include all results. The optimal transfer time (i.e., the primary path loss rate is 0%) of a 100MB file is 153.7 seconds. Figure 5 plots the average 100MB file transfer time for {3, 5, 8, 10}% primary path loss, a 100ms primary path RTT, and a 100 alternate path RTT. Each graph has a fixed primary path loss rate, and varies the alternate path loss rate on the x-axis from 0-10%. Note that the scale of the y-axis is different for each primary path loss rate to allow reader to observe a performance difference between the different threshold settings at each primary path loss rate.

Counter to our intuition, we observe that the PMT setting has little effect on the goodput for primary path loss rates less than 8%. Above 8%, the results show that lower (!) PMT settings begin to improve performance, with PMT = 0 (i.e., a single timeout) providing the most improvement. That is, surprisingly, being more aggressive with probe-over often provides improved performance, even when the alternate path loss rate is higher than that of the primary path. For example, reducing the PMT from 5 to 0 improve the performance by 5% when the primary and alternate path loss rate are 8% and 10%, respectively.

Above, we evaluate different PMT setting, Figure 5 plots PMT = 0 is the best choice, especially, the primary path in higher loss rate and RTT (e.g., for {8, 10}% primary path loss rate).
4.3.3 Different PMS Setting’s Result and Discussions

Figure 6 plots the average 100MB file transfer time for \{0, 1, 2, 3, 5\} % primary path
loss rate, a 100ms primary path RTT, and a 100ms alternate path RTT. When the alternate path loss rate is lower than primary path loss rate, more aggressive circular-failover (i.e., lower PMS setting) dramatically improve performance. On the flip side, the performance is degraded relatively little when the alternate path loss rate is higher than that of the primary path. For example, when the primary path loss rate is 5%, reducing PMS from 5 to 0 improve performance by as much as 88% and degrade performance by at most 9%.

Associations with lower PMS setting tend to (i.e., be forced to) spend less time on the higher loss rate path or lower loss rate path and quickly redirect traffic to another destination. The intuition is as follows. If a sender change path to another path with a higher loss rate, the performance may degrade, but only temporarily. Eventually, PMT or PMR will be exceeded again. The loss rate and RTT of path are two dynamic parameter, so the path’s characteristic is changeful.

Counter to our intuition, lower PMS setting (i.e., high frequency of change destination) reduced transfer time, because the side-effects of path congestion or bottleneck are avoided effectively when a flow moves to a new path circularly.

Figure 6 plots PMS = 0 is the best value, it means initiative change sending destination after one cycle of slow-start and congestion-avoidance. Change destination in high frequency (e.g., PMS = 0) make the route oscillation became possibility. Take into account PMT setting, the best PMT setting is 0 (i.e., a single timeout with one MTU), so the best PMS setting is 2 (i.e., three cycles of slow-start and congestion-avoidance).

Circular Failover Mechanism improved transfer performance without sacrificing failure detection accuracy. Its initiative traffic migration design can reduce the transfer time, improve idle-multipath’s utilization in one association and avoid the possibility of bottleneck path occurrence caused by uses one path permanently or synchronized traffic migration.

5 Conclusions and Future Work

We presented and explored the concept of circular failover mechanism based on three-level threshold (i.e., PMT, PMR and PMS) to further improve end-to-end performance of failover and multipath’s utilization. Above results has show that such a failover mechanism can significantly improve performance no matter what failure occurs.

Although our testing and simulation limited specific network topology and parameter and absolute results differed for those examples compared, relative relationships remained consistent – leading to the same conclusions.

It is necessary to do some optimize for Circular Failover Mechanism, especially the design of threshold and the further explanation of many possible scenarios about PMT and PMR threshold, in order to further improve the transfer performance in transport layer multihoming environment (e.g., SCTP).

Acknowledgments

The authors acknowledge Zhigang Liu, Yuhui Zhao, Laiquan Han, Bing Zhou, and the anonymous reviewers for their valuable comments and suggestion.

This work is supported by the National Science Foundation of China under grant No.60273078 and the Doctor Fund of Hebei under grant No.05547010D-3.
Fig. 6. The performance comparison of different PMS setting

References
NEW END-TO-END FAILOVER MECHANISM WITH TRANSPORT LAYER MULTIHOMING


Jiemin Liu, Protocol Engineer Laboratory, Northeastern University at QinHuangDao, Taishan Rode 143, P.R.China. Email: liuji@mail.neuq.edu.cn. Jiemin Liu is a vice-professor at Northeastern University at QinHuangDao. He received his BSc degree in Computer science & Engineering from Yanshan University in 1988, MSc degree in Computer science& Engineering from Yanshan University in 1997. His research interests are in the areas of Computer Network Protocol and Database Technology. He has published several papers in these areas.

Shaohui Cheng, Protocol Engineer Laboratory, Northeastern University at QinHuangDao, Taishan Rode 143, P.R.China. Email: cheng@mail.neuq.edu.cn. Shaohui Cheng is a vice-professor at Northeastern University at QinHuangDao. He received his BSc degree in Computer science & Engineering from Dalian University of Technology in 1992, MSc degree in Computer science& Engineering from Yanshan University in 2002. He has published several papers in Computer Network Protocol area.