ResTP – A Transport Protocol for FI Resilience

Truc Anh N. Nguyen*, Justin P. Rohrer§, and James P.G. Sterbenz*†‡
*Information and Telecommunication Technology Center
Department of Electrical Engineering and Computer Science
The University of Kansas, Lawrence, KS 66045, USA
†School of Computing and Communications (SCC) and InfoLab21
Lancaster University, LA1 4WA, UK
‡Department of Computing
The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
§Department of Computer Science
Naval Postgraduate School, Monterey, CA
{annguyen|rohrej|jpgs}@ittc.ku.edu, jprohrer@nps.edu
www.ittc.ku.edu/resilinets

ABSTRACT

To support emerging application classes and network use paradigms for Future Internet resilience, we are designing a new transport protocol: ResTP. ResTP overcomes the limitations of TCP and UDP that evolved in the context of the fixed, wired, connected, relatively reliable, and low-to-moderate delay Internet. ResTP is developed to efficiently carry traffic from various application types across a wide variety of network types. By supporting cross-layering, ResTP allows service tuning by the upper application layer while promptly reacting to network condition changes by using the feedback from the lower network layer. ResTP supports a set of transport-layer services, and each service is comprised of many mechanisms and algorithms that can be combined based on the specific mission requirement, application type, and underlying network characteristics. In addition, ResTP can exploit multiple available paths for its data transmission to increase redundancy while better utilizing network resources. With the design based on our ResiliNets framework, we believe that ResTP is the first transport-layer protocol that considers all disciplines related to resilience.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms

Architecture Design

Keywords

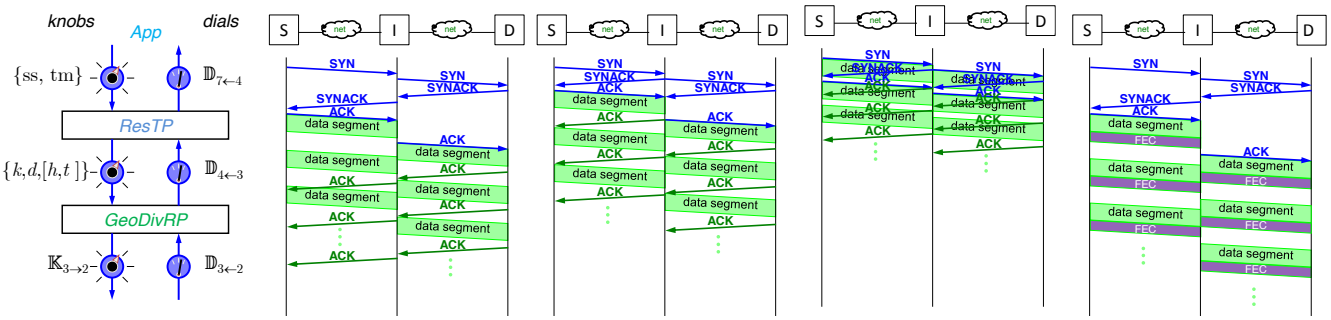
resilient survivable network, multipath transport protocol, DTN, TCP, error control, flow management

1. INTRODUCTION AND MOTIVATION

TCP and UDP evolved as the dominant transport protocols in the Global Internet, which is one of the critical infrastructures on which almost every aspect of lives depend. However, the introduction of new application classes and use paradigms such as wireless access, mobility, mobile ad-hoc networks (MANETs), disruption- and delay-tolerant networks (DTNs), wireless mesh networks (WMNs), and wireless sensor networks (WSNs) challenge TCP capabilities and mechanisms that are more suitable for fixed wired networks. This includes the rigid intertwined error / flow / congestion control and unipath operation. The limitations of TCP are primarily due to the assumptions behind its design and evolution. While wireless environments exhibit a very high bit-error rate (BER), TCP is unable to distinguish between a corruption-based and a congestion-based loss. Every packet loss encountered by TCP is assumed to be a signal of congestion, which causes TCP to invoke its congestion control algorithm to reduce its congestion window. This reduction in the sending rate degrades TCP performance when facing data corruption. The high latency in challenged networks prevents TCP from reacting promptly to the changes in network conditions due to its ACK clocking mechanism, the overhead in its 3-way handshake, and taking several RTTs to learn about the network's available bandwidth before reaching steady state. Moreover, TCP assumption of a stable end-to-end (E2E) path between a pair of communicating hosts is violated by frequently partitioned networks such as the deep space environment due to node's high mobility and limited power or signal fading. TCP also fails to take advantage of multiple physical paths for data transmission within a TCP session, essential for resilience and survivability.

The drawbacks of TCP have motivated the development of numerous algorithms to fix its operation and new transport protocols to replace TCP. We believe that because TCP operation was defined based on the inherent set of assumptions that lie behind its design, any modification or extension on top of TCP is only a temporary solution to address a single problem for a specific mission requirement. Moreover, because TCP is normally extended through the use of options, its 40-byte limitation on the total number of option octets places a constraint on expanding its functionality beyond its default. This is particularly evident in the limits imposed on TCP SACK with only 3-available SACK blocks. Therefore, we are developing a new transport protocol *ResTP* that can efficiently accommodate the new types of application and environments, that is flexible, composable, and with partic-

CFI '15 June 8–10, 2015, Seoul, Korea



(a) ResTP in the stack (b) Fully reliable (c) Nearly reliable (3-way hanshake) (d) Nearly reliable (opportunistic) (e) Quasi reliable

Figure 1: ResTP cross-layering model and transfer mode flow diagrams

ularly emphasis on resilience and survivability.

ResTP (originally proposed in [15]) is designed based on the ResiliNets framework in which the broad definition of *resilience* covers multiple disciplines, including survivability, disruption tolerance, traffic tolerance, dependability, security, performability, and robustness [20]. ResTP is designed as a resilient general-purpose transport layer protocol that is flexible in efficiently supporting various application classes operating across different network environments by employing a set of mechanisms that are composable and tunable by the upper layer. ResTP is capable of exploiting multiple physical paths for data transmission and of selecting diverse paths to avoid correlated failures including from large-scale disasters. ResTP can be deployed either at the gateways where it performs seamless splicing with the conventional TCP, UDP, RTP, or SCPS-TP [9, 3], or at the end-systems as a fully-functional, stand-alone transport protocol.

The rest of the paper is organized as follows: Section 2 gives a short survey on some other developments related to our ResTP. Section 3 discusses some key features of ResTP and presents a brief analysis of AeroTP, a subset of ResTP. Finally, Section 4 concludes our paper and gives direction for future work towards the completion of our ResTP design.

2. RELATED WORK

Challenged networks or intermittently connected networks (ICNs) that are characterized by high latency, limited bandwidth, high error probability, limited node longevity, and lacking of stable E2E path, have been under extensive study. The most prominent approach to provide interoperability among such networks is the Delay Tolerant Networking (DTN) Architecture that was originally designed for the Interplanetary Internet [18, 22]. The key objectives of the DTN Architecture are achieved through the use of the Bundle Protocol (BP), which sits at the application layer of the Internet architecture to construct a store-and-forward overlay network mainly addressing the issue of intermittent connectivity and providing custody-based retransmission [19]. BP's operation is assisted by a convergence layer that adds missing features to the underlying transport protocol such as the TCP Convergence Layer (TCPCL) [8] or the Licklider Transmission Protocol (LTP) [2].

Before the development of the DTN Architecture and its related work, many attempts were made to address the issues in space communications, including the Space Communications Protocol Standards Transport Protocol (SCPS-TP) [9, 3]. SCPS-TP extends TCP by incorporating multiple mechanisms to deal with different types of loss. To cope with asymmetric and limited-capacity channels, SCPS-TP employs a new loss-tolerant header compression scheme and a hybrid SNACK acknowledgement option, which is supported by our ResTP. Some of the other transport protocol designs that influence our development of ResTP include TP++ [10] composability, Stream Control Transmission Protocol (SCTP) [21], and Multipath TCP (MPTCP) [1].

3. RESTP

In this section, we present an overview of the main transport-layer services ResTP provides to the application layer, including its flow management, error control, reliability modes, and multipath spreading modes. We also briefly discuss the cross-layering framework supported by ResTP and its header. Finally, we briefly present AeroTP, a domain-specific subset of ResTP developed to address the issues posed by the highly-dynamic, airborne telemetry network environment.

3.1 Cross-layering Framework

ResTP supports cross-layering to vertically interconnect with the upper application and the lower network layers as illustrated in Figure 1a. In the model, the downward *knobs* $\mathbb{K}$ allow service tuning from the application layer while the upward *dials* $\mathbb{D}$ allow feedback updating from the network layer, in which we employ our GeoDiverse Routing Protocol (GeoDivRP) [6, 4, 7, 5] implementation.

3.2 ResTP header

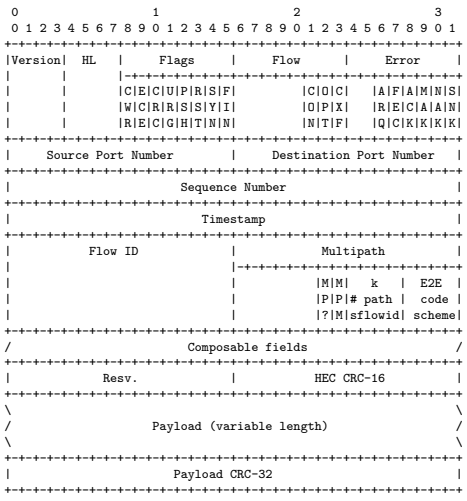


Figure 2: ResTP data segment format

Figure 2 displays the main fields in a ResTP header that are common for all ResTP data segments (ResTPDU). The 4-bit version field specifies the protocol version while the 4-bit HL field contains the header length in words. The flag field contains 2 ECN bits (`CWR` and `ECE`) used for TCP splicing when ResTP is deployed at the gateways and 6 other flags in which some of them are TCP's. The TCP's `RST` bit is used for flow modification for dynamic adaptation in addition to its original use of resetting a connection. The flow field in which 3 out of 8 bits are currently in use contains the flags for flow management mechanisms, including `CON` (connection-oriented), `OPT` (opportunistic), and `CXF` (custody transfer). 6 out of 8 bits in the error field specify the flags for setting up a specific error control mode, including `ARQ`, `FEC`, `ACK`, `MAK` (MACK), `NAK` (NACK), and `SNK` (SNACK). The combinations of bits in the flow and error fields define the various reliability modes in ResTP as discussed in the next section. The source and destination port numbers identify the sending and the receiving applications. The sequence number field contains a 32-bit unsigned integer assigned to the ResTPDU. Data segments transmitted for the first time are numbered using only even sequences. Odd numbers are used when a previously transmitted ResTPDU needs to be retransmitted using a smaller payload size. In this case, the payload is split into 2 ResTPDUs with the first carrying the even sequence number of the original segment while the second carrying one sequence number above the original. The 32-bit timestamp field contains the time when the ResTPDU is transmitted. The 16-bit flow ID uniquely identifies a flow while the 16-bit multipath field specifies 10 flags required for setting up a multiple-path data transfer, which includes the `MP?` (whether or not multiple paths will be used), `MPM` (multipath mode: E2E coding across paths or alternate path as hot-standby), 4-bit sub-field `k` that can be used to specify the number of paths k required or a subflow ID for a data segment, and another 4-bit sub-field specifying the coding scheme. The HEC CRC-16 field contains the integrity check for the header while the 32-bit CRC-32 field contains the integrity check for the entire segment.

3.3 Flow Management

ResTP supports both the connection-oriented and connectionless flow management schemes. With the connection-oriented mode, a flow is established as with TCP's 3-way handshake technique with the exchange of SYN, SYNACK, and ACK control segments between communicating hosts before the transmission of application data. The 3-way handshake is modified to overlap data with control segments resulting in opportunistic flow management. This opportunistic flow establishment reduces the setup overhead, permitting ResTP to be appropriate in a bandwidth-constrained, high-delay, error-prone, dynamic-topology environment. For time-sensitive applications without strict requirements in reliability such as Internet telephony/video conferencing, the flexible ResTP provides the connectionless flow management mode in which individual datagrams are transmitted whenever data is available.

3.4 Error Control

ResTP supports multiple error correction mechanisms, including Automatic Repeat reQuest with acknowledgments (ARQ), E2E Forward Error Correction (FEC), and HARQ (hybrid ARQ over FEC). While ARQ is used for reliable data transfer, FEC is well-suited for quasi-reliable data transfer, and HARQ provides reliable data transfer with E2E FEC providing statistical reliability. ResTP also allows the use of alternative acknowledgment techniques, including the traditional positive ACK, aggregated Multiple ACK (MACK), Negative ACK (NACK), and Selective Negative ACK (SNACK). These acknowledgment techniques are combinable based on the application type, specific mission requirements, and the underlying physical path's characteristics. For highly loss-tolerant applications, ResTP can also operate in the no error-control mode.

3.5 Reliability Modes

ResTP defines multiple reliability modes from coupling flow management and error control techniques to satisfy the service requirements of various applications:

Fully-reliable connection mode: This fully reliable mode ensures correct data delivery by preserving the E2E ACK semantics from source to destination (Figure 1b).

Nearly-reliable connection mode: This mode provides reliability, but does not guarantee correct data delivery since the gateway uses custody transfer and immediately returns TCP ACKs to the source with the assumption that ResTP will successfully deliver the data to the destination by using its ARQ system (Figure 1c and Figure 1d).

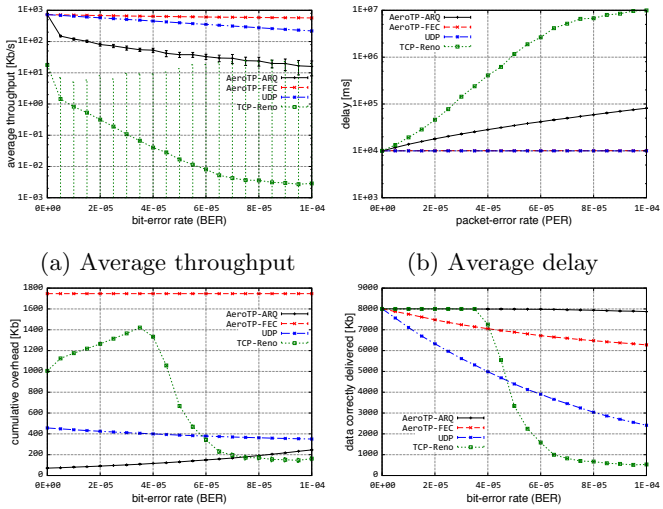
Quasi-reliable connection mode: Instead of using ACKs and ARQ, this mode provides some level of statistical reliability by relying on open-loop error recovery mechanisms such as FEC and erasure coding. The coding strength can be tuned according to the network conditions and traffic type (Figure 1e).

Unreliable connection mode: In this mode, ResTP invokes the connection-oriented flow management but provides no error correction. The only means to preserve data integrity is the FEC at the link layer. This mode has a similar flow diagram to the quasi-reliable mode, but without the FEC.

Unreliable datagram mode: This mode provides no assurance of data delivery with no flow setup. This mode has a similar flow diagram to the quasi-reliable mode, but without the 3-way handshake and the FEC.

3.6 Multipath Spreading

With the existence of multiple physical paths through an overlay network, multi-tunnels, or transport layer multihoming, ResTP can exploit these paths for data transfer to increase redundancy and better utilize the network resources. One of ResTP's key features is the ability to select diverse paths by using the cross-layering framework to reduce the likelihood of suffering from correlated failures. ResTP manages these selected paths using 2 modes: actively spreading data over all paths to survive a single path failure (for example using a 2-of-3 erasure code), or transmitting data on one path and using another as a hot-standby for rapid failover. In the former case, E2E communication will survive the failure of individual paths with no need for retransmission, appropriate for real-time applicaitons. In the latter case, with cross layering, after discovering the disruption of the active path from the lower layer, ResTP will promptly switch to an alternate path. The selection between these two modes is made based on the path attributes, application type, and mission requirements. In addition, ResTP allows additional paths to be added on-demand. These features distinguish ResTP from MPTCP, which simply splits user data into



(a) Average throughput (b) Average delay

(c) Cumulative overhead (d) Cumulative goodput

Figure 3: AeroTP simulation performance results

one or more subflows and distributes them over available paths. When multiple flows are in service, ResTP's congestion control algorithm preserves fairness among the ResTP flows and friendliness with other transport flows.

3.7 AeroTP: A subset of ResTP

During the process of developing our resilient transport protocol, we have designed, implemented, and analysed our Aeronautical Transport Protocol (AeroTP), which is a domain-specific subset of ResTP designed to handle the unique characteristics of a highly-dynamic airborne telemetry network environment such as its asymmetric, error-prone channels, limited bandwidth, and intermittent connectivity with frequent topology changes [17, 14]. AeroTP employs the opportunistic flow management and the 5 reliability modes discussed above. Through several analyses [13, 11, 16, 12], we have shown that AeroTP outperforms the conventional TCP in most scenarios as depicted in Figures 3a, 3b, 3c, and 3d.

4. CONCLUSIONS

In this paper, we present the main features of our ResTP architecture. Although we are still in the design and implementation process, based on the performance of its subset AeroTP, we are optimistic that ResTP will be well-suited the resilient Future Internet. We are in the process of completing the detailed design of ResTP. For future work, we will implement it in ns-3 (based on the current AeroTP model) and analyse it in comparison with other transport protocols such as TCP, SCPS-TP, and MPTCP. This will be followed by a prototype implementation and analysis.

5. REFERENCES

[1] A. Ford et al. TCP Extensions for Multipath Operation with Multiple Addresses. RFC 6824 (Experimental), Jan. 2013.

[2] S. Burleigh, M. Ramadas, and S. Farrell. Licklider Transmission Protocol - Motivation. RFC 5325 (Informational), Sept. 2008.

[3] CCSDS. Space Communications Protocol Specification (SCPS)-Transport Protocol (SCPS-TP). http://public.ccsds.org/publications/archive/714x0b2.pdf, October 2006.

[4] Y. Cheng, M. T. Gardner, J. Li, R. May, D. Medhi, and J. P. Sterbenz. Optimised heuristics for a geodiverse routing protocol. In *Proceedings of the IEEE Design of Reliable Communication Networks (DRCN)*, Ghent, Belgium, April 2014.

[5] Y. Cheng, M. T. Gardner, J. Li, R. May, D. Medhi, and J. P. Sterbenz. Analysing geopath diversity and improving routing performance in optical networks. *Elsevier Computer Networks Journal*, 2015. (accepted in January 2015).

[6] Y. Cheng, J. Li, and J. P. G. Sterbenz. Path Geo-diversification: Design and Analysis. In *IEEE/IFIP Reliable Networks Design and Modeling (RNDM)*, Almaty, September 2013.

[7] Y. Cheng and J. P. Sterbenz. Geodiverse routing with path jitter requirement under regional challenges. In *Reliable Networks Design and Modeling (RNDM)*, pages 179–186, Nov 2014.

[8] M. Demmer, J. Ott, and S. Perreault. Delay-Tolerant Networking TCP Convergence-Layer Protocol. RFC 7242 (Experimental), 2014.

[9] R. C. Durst, G. J. Miller, and E. J. Travis. TCP extensions for space communications. In *ACM MobiCom '96*, pages 15–26, New York, NY, USA, November 1996. ACM Press.

[10] D. Feldmeier. An overview of the TP++ transport protocol project. In A. N. Tantawy, editor, *High Performance Networks: Frontiers and Experience*, chapter 8. Kluwer, Boston, MA, USA, 1993.

[11] K. S. Pathapati, T. A. N. Nguyen, J. P. Rohrer, and J. P. Sterbenz. Performance Analysis of the AeroTP Transport Protocol for Highly-Dynamic Airborne Telemetry Networks. In *International Telemetering Conference (ITC)*, Las Vegas, NV, October 2011.

[12] K. S. Pathapati, J. P. Rohrer, and J. P. Sterbenz. Comparison of adaptive transport layer error-control mechanisms for highly-dynamic airborne telemetry networks. In *International Telemetering Conference (ITC)*, San Diego, CA, October 2012.

[13] K. S. Pathapati, J. P. Rohrer, and J. P. G. Sterbenz. Edge-to-edge ARQ: Transport-layer reliability for airborne telemetry networks. In *International Telemetering Conference (ITC)*, San Diego, CA, Oct 2010.

[14] J. P. Rohrer, A. Jabbar, E. K. Çetinkaya, E. Perrins, and J. P. Sterbenz. Highly-dynamic cross-layered aeronautical network architecture. *IEEE Trans. Aerospace and Electronic Systems*, 47(4):2742–2765, October 2011.

[15] J. P. Rohrer, R. Naidu, and J. P. G. Sterbenz. Multipath at the transport layer: An end-to-end resilience mechanism. In *IEEE/IFIP Reliable Networks Design and Modeling (RNDM)*, pages 1–7, St. Petersburg, Russia, October 2009.

[16] J. P. Rohrer, K. S. Pathapati, T. A. N. Nguyen, and J. P. G. Sterbenz. Opportunistic transport for disrupted airborne networks. In *IEEE MILCOM*, pages 737–745, Orland, FL, November 2012.

[17] J. P. Rohrer, E. Perrins, and J. P. G. Sterbenz. End-to-end disruption-tolerant transport protocol issues and design for airborne telemetry networks. In *International Telemetering Conference (ITC)*, San Diego, CA, October 2008.

[18] S. Burleigh et al. Delay-Tolerant Networking: An Approach to Interplanetary Internet. *IEEE Communications*, 41(6):128–136, June 2003.

[19] K. Scott and S. Burleigh. Bundle Protocol Specification. RFC 5050 (Experimental), Nov. 2007.

[20] J. P. G. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith. Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. *Computer Networks*, 54(8):1245–1265, 2010.

[21] R. Stewart. Stream Control Transmission Protocol. RFC 4960 (Standard), Sept. 2007.

[22] V. Cerf et al. Delay-Tolerant Networking Architecture. RFC 4838 (Informational), Apr. 2007.