

MPUDP: Multipath Multimedia Transport Protocol over Overlay Network

Shaowei Liu^a, Weimin Lei^b, Wei Zhang, Hao Li

School of Computer Science and Engineering, Northeastern University, Shenyang, 110819, China

^aemail: shaoweiliu@stumail.neu.edu.cn, ^bemail: leiweimin@mail.neu.edu.cn

Keywords: Multipath Transmission, Multimedia, Transport protocol, Bandwidth, Reliability

Abstract. Multipath transport in communication networks is gaining momentum due to its attractive features of increased reliability, aggregated bandwidth, fault tolerance, and load balancing. Multipath transport framework based on application-level relay (MPTF-AR) that uses overly network technique is regarded as a simple way to establish multipath transport scenario without any change on underlying network. However, existing multipath transport protocols are not suitable for MPTF-AR because they either work in multihomed scenario or cannot support transmission in overlay network. This paper proposes a transport protocol named multipath UDP (MPUDP) to support both reliable and unreliable transport in MPTF-AR. With the help from application and transport layer, MPUDP support precise performance detection and dynamic data distribution. End host uses two-level buffer to reorder packets. In addition, end hosts no longer keeps a congestion window and always send packets over optimal paths. Finally, extensive OMNeT++ simulations are conducted and show that MPUDP exploits the benefit of multipath transport.

Introduction

With the rapid development of information technology, numerous applications appear in various areas, e.g., video conferencing, artificial intelligence, high definition video, and virtual reality. They have stringent quality of service (QoS) requirements. Currently, the majority of data transmissions go through a single path. In the most cases, a single path is not capable of completely supporting these emerging applications. In addition, in the case with high loss or delay, the performance of single path transmission degrades significantly.

To provide high quality delivery service for these emerging applications, transmission technology is vital. The increasing availability of multiple paths between end devices provides an opportunity to address the above requirements. Concurrent multipath transmission (CMT) is regarded as a promising solution since it can aggregate bandwidth of multiple paths and protect transmission from failure of a path.

In recent years, multipath transmission has attracted extensive research interests. State-of-art smart devices have multiple network interfaces. Multiple IP addresses to transport layer is regarded as one of the most popular approaches to construct multiple end-to-end paths. However, it requires at least one of the end devices has multiple network interfaces, which is hard to be implemented within the scope of the whole network. Overlay network technique is another approach to establish multiple paths between end devices by application layer routing without any changes on the underlying network.

How to make full use of available paths to meet application requirement is a challenging topic. Traditional transport protocols, e.g., TCP and UDP, cannot support the using of multiple paths concurrently. To address this problem, many researchers have been devoted to the simultaneous harmonization of multiple paths to improve network throughput and enhance quality of service. For instance, the Internet engineering task force (IETF) has standardized multipath TCP (MPTCP) [1] and stream control transmission protocol (SCTP) [2]. Both MPTCP and SCTP reuse partial mechanisms of TCP and provide reliable communication. Multipath RTP (MRTP) that supports to retransmit the lost packet defines how to take advantage of the multiple paths between end devices to deliver media [3]. These protocols only apply to a specific kind of applications and require existing network devices to support multipath routing.

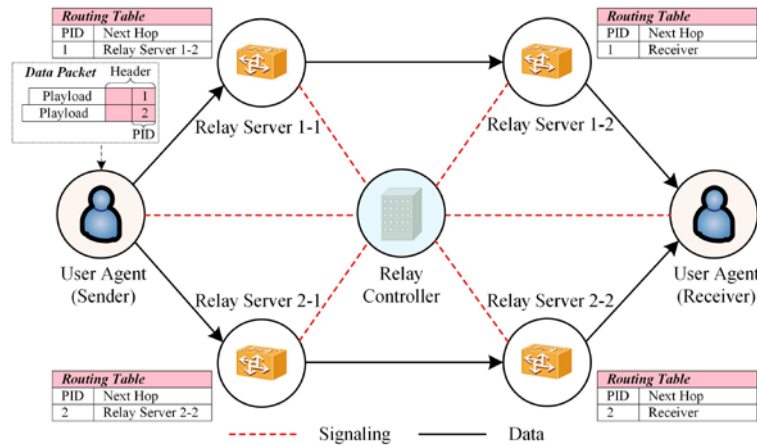


Fig. 1. An overview of multipath framework that provides multiple paths between the sender and receiver.

In this research, multiple paths between sender and receiver are provided by a multipath transport framework based on application-level relay (MPTF-AR) that uses overlay network technology. The multipath framework consists of three logical entities, i.e., user agent, relay controller, and relay server [4], [5]. These logical entities form an overlay network. Relay controller maintains a good knowledge of the overlay network topology. Relay server forwards data packet to next hop logical entity based on local routing table. Before data transmission, user agent (sender) requests relay controller to allocate multiple paths. The path is composed of a set of continuous next hop of relay servers between two user agents (sender and receiver), and it is identified by a unique path identity (PID). Relay controller informs user agent and related relay server of allocated path information. Each item of local routing table includes a PID and the next hop address of relay server or user agent. During transmission, each packet carries a PID which helps relay server to look for the address of next hop logical entity. Fig. 1 describes a scenario that relay controller allocates two paths for the transmission from the sender to the receiver.

In this paper, we present a multipath multimedia transport protocol which exploits path diversity over overlay network MPUDP. The goal of MPUDP is to provide a high quality reliable/unreliable delivery service for applications. Our main contributions are summarized as below:

- Accurate path quality evaluation: Two-level sequence space and two-level buffer support end host to detect packet loss accurately.
- Dynamic data distribution: Path quality play a key role in data distribution which optimize path utilization.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 discusses the details of MPMTP. Section 4 conducts extensive performance study of MPMTP. Section 5 concludes the paper.

Related Work

Recently CMT has attracted extensive academic research interests. Dreibholz et al. introduced the SCTP and gave an overview of activities and challenges in the areas of CMT [6]. In the primary version of SCTP, multiple paths are only considered as backups of the primary path. In recent extension, SCTP adds the feature of CMT [7]. Paasch et al. gave the design principles of MPTCP [8]. The design and implementation of MPTCP was specified by [1].

Huang et al. proposed a fast retransmission model to deal with packet loss in vehicular networks by relay gateway (RG-CMT) [9]. When detecting packets loss due to handoff loss or error loss, RG-CMT retransmit them from the nearest relay gateway to destination which saves transfer time and bandwidth. Wu et al. proposed distortion aware concurrent multipath transfer (CMT-DA) solution [10]. CMT-DA analyzes data distribution among multiple paths to minimize the video distortion based on the utility maximization theory. Wu et al. proposed a scheduling approach that uses frame splitting based on weibull distribution and graph theory to minimize the end-to-end

delivery delay and reduce out-of-order reception [11]. Xu et al. proposed a quality-aware adaptive concurrent multipath transfer solution (CMT-QA) [12]. CMT-QA takes advantages of reordering delay reduction and unnecessary fast retransmission to alleviate out-of-order data reception.

Apostolopoulos et al. presented the overview of media streaming systems that use path diversity was proposed [13]. Chebrolu et al. uses earliest delivery path first (EDPF) algorithm to estimate delivery time and distributes each packet to the path that delivers it the earliest [14]. Mao et al. proposed multiflow real time transport protocol (MRTP) [15] that enables real time application to partition and transmit data using multiple flows. Data partition improves queuing performance by reducing the short-range dependence of media data. Singh et al. presented multipath RTP [3] that defines how to take advantages of the multiple paths between end devices to deliver media. Yilmaz et al. introduced non-renegable selective acknowledgments (NR-SACKs) to decrease buffer size and end-to-end delay by releasing buffer space earlier and reuse it for new data [16]. These solutions are able to improve transmission performance. However, they do not consider data distribution mechanism at the sender and the data arrival at the receiver.

Cui et al. proposed a fast selective ACK (SACK) scheme for SCTP to enhance transmission throughput in multihomed scenario. In the networks with different path characteristics, a multihomed receiver sends SACK chunks to the sender over the fastest reverse path, which facilitates to inflate the congestion window and to retransmit the lost data packets as quickly as possible. This solution cannot enhance the overall transmission efficiency, since it only considers the transmission of control data.

Multipath UDP

MPUDP is a session-based protocol and provides a convenient vehicle for upper application to distribute and transmit data over multiple flows. From the perspective of protocol stack architecture, MPUDP acts as an interlayer between application layer and transport layer. Application interacts with MPUDP through the regular socket API, and MPUDP manages multiple TCP/UDP flows.

During transmission, sender assigns a flow to each active path. The original application data is distributed into multiple flows. On arriving at the destination, these packets combine to reconstruct original message. Now, we are in a position to formally start our discussion on multipath-related issues in MPUDP.

Session establishment: How to establish session with multiple paths? MPUDP uses similar scheme as that in TCP to establish session between end-points. In reliable transmission, end host also negotiate the initial sequence.

Flow control: Flow control can be performed on a per session or per path. MPUDP selects session level flow control and maintains a shared buffer for all paths while keeps a separate buffer per path.

Sequence number splitting: Conventional TCP uses a single sequence space to reconstruct message and detect lost packets at the receiver side. In multipath transport, single sequence space may be insufficient to support packet loss detection and message reconstruction. MPUDP adopts double sequence space, i.e., transmission sequence number (TSN) per session and flow sequence number (FSN) per path.

Acknowledgement (ACK): With multiple paths, are ACKs required at per session or per path or both? When multiple paths have heterogeneous characteristics, packets may arrive at the receiver with different rate. So, MPUDP selects session level SACK to acknowledge correct reception.

Feedback: In addition to ACK, end host also sends report to inform the peer about the delivery status, which including sender report (SR) and receiver report (RR).

Data distribution: In single path, data distribution is not required. With multiple paths, there is a need to schedule data across multiple paths due to heterogeneous path characteristics. Path characteristics plays an important role in data distribution.

Reconstruction: With multiple paths, packets may experience different jitter divided into two types, i.e., the jitter within each path, and the jitter across the paths. Two kinds of jitter result in queuing of received data packet in the reorder buffer. Just like sequence number, MPUDP uses two

level buffer to reconstruct original packet.

A quick glance at the above-mentioned seven points reveals that the first five questions are related to protocol specifications in general while the last two related to multiple paths. And we focus on path quality evaluation and data distribution in the next subsections.

Path quality evaluation

Basing on historical evaluation results and real-time QoS feedback, MPUDP evaluates path performance timely and distribute data to multiple paths accordingly.

MPUDP session uses N paths, i.e., path 0, path 1, ..., path $N-1$. When sender receives k -th RR from the receiver, it updates delivery status. The number of sent and correctly received data packet in bytes on path i since the last RR are s_{ki} and a_{ki} , respectively. The total number of sent and acknowledged packet in byte are S_k and A_k , respectively.

$$\begin{cases} A_k = \sum_{i=0}^{N-1} a_{ki} \\ S_k = \sum_{i=0}^{N-1} s_{ki} \end{cases} \quad (1)$$

Then delivery reliability from the perspective of path i and the session are R_k and r_{ki} , respectively. They are calculated as follows:

$$\begin{cases} R_k = A_k / S_k \\ r_{ki} = a_{ki} / s_{ki} \end{cases} \quad (2)$$

The difference of the number of acknowledged packet from the view of path and session is Δa_{ki} which can be calculated as follows:

$$\Delta a_{ki} = a_{ki} - R_k s_{ki} \quad (3)$$

Then the difference of the number of sent packet over path i from the view of theory and practice is Δs_{ki} .

$$\Delta s_{ki} = \Delta a_{ki} / R_k \quad (4)$$

We assume the load weight of path i assigned by last evaluation period is w_{ki} and set $w_{1i} = 1/N$. The load weight of path i inferred from the current RR is

$$w'_{ki} = w_{ki} (1 + \Delta s_{ki} / s_{ki}) = w_{ki} r_{ki} / R_k \quad (5)$$

Finally, we update the load weight of path i by

$$w_{(k+1)i} = \beta w_{ki} + (1 - \beta) w'_{ki} \quad (6)$$

β is an impact factor. Lots of experiments not shown in the paper recommend that $\beta = 0.6$. Load weight indicates the path quality which directs data distribution in the next subsection.

Data distribution

The goal of data distribution is to select appropriate path for packet. Data distribution policy need to increase the load on the path with better performance and decrease the load on the path with worse performance. Based on load weight calculated in previous subsection, sender tries to keep load balancing across multiple paths at any time when it assigns packets to path.

Data distribution is triggered to distribute packets to multiple paths. It selects the path with minimum utilization of load weight to transmit packet. Finally, each packet is assigned a PID, FSN, and TSN.

Performance Study

To evaluate MPUDP performance, it is implemented on OMNeT++ platform which is a discrete event simulator. Note that the most important feature of MPUDP is reliable/unreliable multipath transport capability. Consequently, our studies focus on how MPUDP achieves multipath transport over MPTF-AR and improves transport performance.

The studies have stated that a relay-path with one relay server provides close performance to that

with multiple relays [15], [16]. In the following simulations, each relay-path has only one relay server, and link parameters are configurable.

Reliable Transport Service

Comparing to unreliable transport service, reliable transport service is much more complex. Hence, we conduct a reliable transmission to demonstrate basic functions of MPUDP. At the beginning, there is a default-path that is selected by network layer routing protocol between sender and receiver; then the sender requests controller to allocate relay-path that go across one relay server; finally, sender and receiver are connected by a default-path and a relay-path, as shown in Fig.2. After obtaining relay-path, sender adds relay-path into the session immediately. Default-path and relay-path are denoted as flow 1 and flow 2, respectively. We set up the flow 1 with 100 ms delay and 1% loss rate, flow 2 with 100 ms delay and 5% loss rate. At 3 second, flow 1 and flow 2

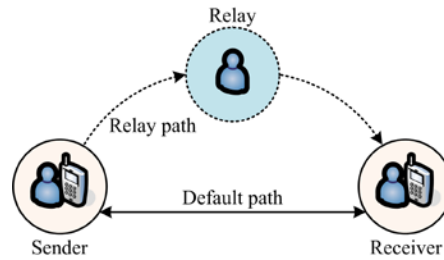


Fig. 2. Simulation topology. There are two paths between sender and receiver.

exchanges link parameter.

Fig. 3 displays the TSN and FSN. From it, we can see that sender obtains relay-path and adds it into the session quickly within 0.4 second. After 3 second, sender detects performance changes of the two flows, and adjusts data distribution ratio. Flow with higher reliability delivery more data packet than that on the other flow. In addition, all lost data packets caused by error and congestion are retransmitted

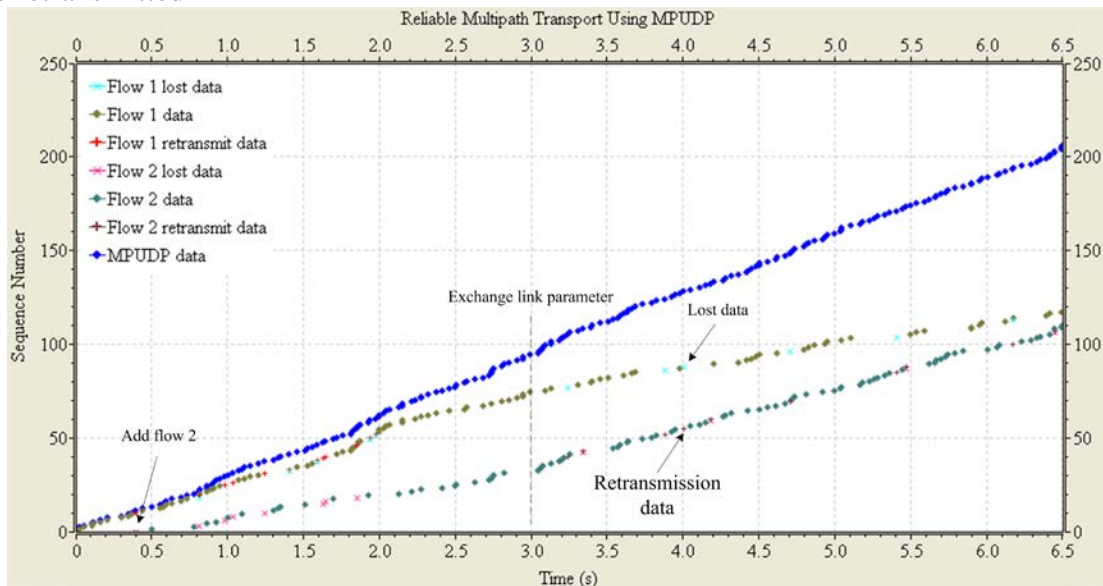


Fig. 3. Data reliable multipath transport. (Sender obtains relay-path from controller and adds it into the MPUDP session immediately. Default-path exchanges link parameter with relay-path at 3 second.)

The result demonstrates that path quality evaluation and data distribution support MPUDP to adapt to underlying networking change. We could infer that the MPUDP will also perform well in unreliable transport service. Due to delay of establishing multiple flows, MPUDP is not suitable for service with low delivery duration.

Video Transport Service

A video transport is conducted to study the performance of unreliable transport, and compare it with that of regular single-path UDP, denoted as RTP/MPUDP and RTP/UDP, respectively. The simulation topology is described as Fig. 2. RTP/UDP uses a path that has the same link parameter

with that of default-path of MPUDP. We set up relay-path with 100 ms delay and 2% loss rate and default-path with 2% loss rate. The mean delay (ms) of default-path obeys normal distribution $N(\text{MeanDelay}, 20)$, where *MeanDelay* is a variable. At the beginning of the transmission, the sender adds the relay-path into the session. The video is encoded using quarter common intermediate format (QCIF) (176×144) at 25 fps. Fig. 4 presents average PSNR traces from RTP/MPUDP and RTP/UDP. It is obvious that RTP/MPUDP outperforms RTP/UDP in all test cases.

In Fig. 4(a), the *MeanDelay* is set to 100 ms, while the playout buffer size varies from 250 ms to 3000 ms. When playout buffer size increases from 250 ms to 3000 ms, RTP/MPUDP and RTP/UDP increase 1.35 dB and 0.56 dB in average PSNR, respectively. This is significant in terms of user's QoE. However, the improvement becomes smaller for further increase in playout buffer size. As most of the remaining lost packets are dropped by path failure rather than congestion of queue.

In Fig. 4(b), the playout buffer size is set to 750ms, while the *MeanDelay* varies from 100 ms to 300 ms. We can observe that RTP/MPUDP and RTP/UDP respectively experience 0.98 dB and 0.5 dB average PSNR reductions, as the *MeanDelay* vary from 100 ms to 300 ms. In addition to path failure, part of packets is not able to reach the receiver within the predetermined time since *MeanDelay* increases. While the difference of average PSNR between RTP/MPUDP and RTP/UDP also decreases, as low-quality path limits the total performance of MPUDP.

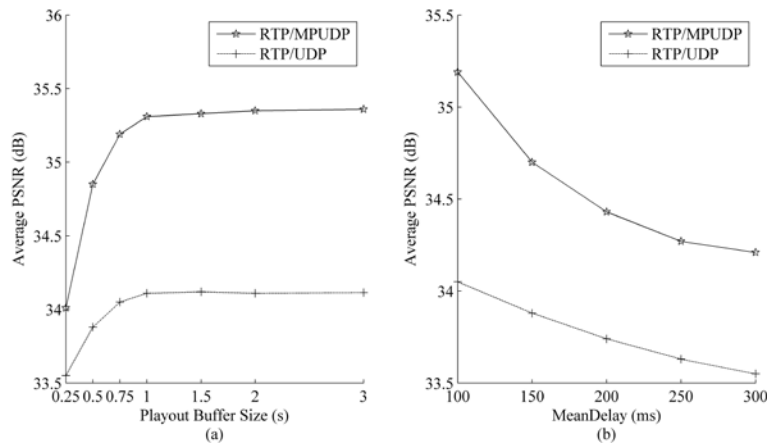


Fig. 4. Average PSNR under various playout buffer size and mean delay, respectively. (The parameters of the relay-path remain the same (100 ms delay, 2% loss rate). In (a), default-path with 100ms *MeanDelay*, and 2% loss rate. In (b), playout buffer size is set to 750 ms, while default-path with 2% loss rate.)

Conclusion

This work presented a multipath multimedia transport protocol, i.e., MPUDP. Our proposal was motivated by the potential benefits of multipath transport. An appropriate transport protocol is the critical element to exploit the benefits of multipath transport. Considering the architecture of MPTF-AR and transport requirement of application, MPUDP deals with general issue of transport protocol, establishes session before data transmission to associate multiple paths, distributes packets to appropriate path based on path performance, uses two sequence space to support packet loss detection and message reconstruction, adopts two level buffer to absorb jitter. With the help of accurate path quality evaluation, sender selects path with minimum utilization of load weight to transmit packet. Simulation results show that MPUDP can exploit the benefits of multipath transport.

Acknowledgement

In this paper, the work was supported by the National Natural Science Foundation of China (No. 61401081), the Fundamental Research Funds for the Central Universities (No. N150404005), and Ministry of Education and China Mobile Scientific Research Fund (No. MCM20150103).

References

- [1] J. Iyengar, C. Raiciu, S. Barre, M. J. Handley, and A. Ford. Architectural Guidelines for Multipath TCP Development. RFC 6182, Mar. 2011
- [2] R. R. Stewart. Stream Control Transmission Protocol. RFC 4960, Sep. 2007.
- [3] S. Ahsan, L. Eggert, V. Singh, T. Karkkainen, and J. Ott. Multipath RTP (MPRTP). Internet-Draft draft-ietf-avtcore-mprtp-07, Jan. 2017
- [4] S. Liu, W. Lei, and W. Zhang. A Framework of Multipath Transport System Based on Application-Level Relay (MPTS-AR). Internet-Draft draft-leiwmtsvwg-mpts-ar-06, Jul. 2016
- [5] W. Zhang, W. Lei, S. Liu, and G. Li. A general framework of multipath transport system based on application-level relay [J]. *Computer Communications*, 2015 51 70–80.
- [6] T. Dreibholz, E. P. Rathgeb, I. Rungeler, R. Seggelmann, M. Tuxen, and R. R. Stewart. Stream control transmission protocol: Past, current, and future standardization activities [J]. *IEEE Communications Magazine*, 2011 49(4) 82–88.
- [7] J. R. Iyengar, P. D. Amer, and R. Stewart. Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths [J]. *IEEE/ACM Transactions on Networking*, 2006 14(5) 951–964.
- [8] C. Paasch and O. Bonaventure. Multipath TCP: Decoupled from IP, TCP is at last able to support multihomed hosts [J]. *ACM Queue*, 2014 12(2) 1-12.
- [9] C. M. Huang and M. S. Lin. Fast retransmission for concurrent multipath transfer (CMT) over vehicular networks [J]. *IEEE Communications Letters*, 2011 15(4) 386–388.
- [10] J. Wu, B. Cheng, C. Yuen, Y. Shang, and J. Chen. Distortion aware concurrent multipath transfer for mobile video streaming in heterogeneous wireless networks [J]. *IEEE Transactions on Mobile Computing*, 2015 14(4) 688–701.
- [11] J. Wu, B. Cheng, Y. Shang, J. Huang, and J. Chen. A novel scheduling approach to concurrent multipath transmission of high definition video in overlay networks [J]. *Journal of Network and Computer Applications*, 2014 44 17–29.
- [12] C. Xu, T. Liu, J. Guan, H. Zhang, and G.-M. Muntean. CMT-QA: Quality-aware adaptive concurrent multipath data transfer in heterogeneous wireless networks [J]. *IEEE Transactions on Mobile Computing*, 2013 12(11) 2193-2205.
- [13] J. G. Apostolopoulos and M. D. Trott. Path diversity for enhanced media streaming [J]. *IEEE Communications Magazine*, 2004 42(8) 80–87.
- [14] K. Chebrolu and R. R. Rao. Bandwidth aggregation for real-time applications in heterogeneous wireless networks [J]. *IEEE Transactions on Mobile Computing*, 2006 5(4) 388–403.
- [15] S. Mao, D. Bushmitch, S. Narayanan, and S. S. Panwar. MRTP: a multiflow real-time transport protocol for ad hoc networks [J]. *IEEE Transactions on Multimedia*, 2006 8(2) 356–369.
- [16] E. Yilmaz, N. Ekiz, P. Natarajan, P. Amer, J. Leighton, F. Baker, and R. Stewart. Throughput analysis of non-renegable selective acknowledgments (NR-SACKs) for SCTP [J]. *Computer Communications*, 2010 33 1982–1991.