Multipath TCP over LEO Satellite Networks

Pengyuan Du, Xiao Li, You Lu, Mario Gerla
Department of Computer Science,
University of California, Los Angeles
Los Angeles, USA
{pengyuandu, pololee, youlu, gerla}@cs.ucla.edu

***Abstract*—Low earth orbit (LEO) satellite networks like Iridium have played a pivotal role in providing ubiquitous network access services to areas without terrestrial infrastructure because of their potential for global coverage and high bandwidth availability. With low orbit and short range as compared to geostationary satellites, LEO satellites are accessible by mobile devices with limited transmission power and small gain antennas. The drawback, however, is that LEO satellites move fast across the sky with average contact time in the order of 10 minutes, thus requiring frequent handover from one satellite to the next. To achieve smooth handover and efficiently utilize constellation capacity, we propose to use Multipath TCP (MPTCP) in LEO systems and maintain parallel, simultaneous connections between terrestrial handpoints via multiple satellites. In this paper, we discuss the feasibility of using MPTCP over LEO satellite networks and propose a framework of MPTCP-Routing design. Then the performance of this protocol is evaluated through simulation. We show that compared to traditional "single-path" TCP, MPTCP significantly improves throughput performance and prevents the interruption of transmission during handover. Furthermore, we show that our MPTCP-Routing interaction is essential for the end-to-end session to quickly recover from handover.**

Keywords*—*Multipath TCP, LEO Networks, On Demand Multipath Routing

I. INTRODUCTION

Satellite systems have been a prevailing candidate for ubiquitous communication because of its capability of providing global coverage and a consistent level of services. In recent years, the telecommunication community has drawn attention to the satellite systems deployed at altitudes between $500km$ and $1500km$, which are known as low earth orbit (LEO) satellite systems [19]. Compared to geostationary satellite networks, the propagation delay of LEO satellite systems is relatively low and the throughput is higher. Furthermore, LEO satellites have lower transmission power requirements. In many constellations, direct inter-satellite links (ISLs) between mutually visible LEO satellites are used to carry signaling and management traffic as well as data packets [24].

However, the topology of LEO satellite networks changes over time due to constant rotation of the LEO satellites. This will result in frequent satellite handover whenever the current gateway (or serving) satellite fades out and the ongoing connection has to be transferred to the next satellite visible to the ground terminal [6]. For example, In Iridium system, the visibility period of a satellite is usually 8-10 minutes. Without appropriate control and mitigation mechanisms, satellite handover can lead to route failures, drastic link quality change, call blocking and eventually result in severe performance degradation as bad as connection interruption.

We classify the transmission state as stable state and handover state. Stable state corresponds to the period when the gateway satellite is constantly visible to the ground terminal, while handover state corresponds to the satellite handover period. Even though the network condition is relatively steady during stable states, it has been well known that the performance of TCP is seriously impaired in LEO satellite systems, due to long Round Trip Times (RTTs) and possible presence of channel error based packet losses [5][1]. The RTT problem is even worse in LEO networks, since the end-to-end delay varies as satellites move along orbits.

Different high bandwidth delay product (BDP) TCP variants are proposed to deal with these issues [9][14][3]. Given their strengths in miscellaneous scenarios and applications, we notice that none of them are designed against satellite handover. To the best of our knowledge, satellite handover is mainly dealt with on link layer and network layer [18][6][23][22]. However most of the proposed approaches could only serve as a mitigation to the problem rather than a cure. A transport layer solution providing more robustness and responsiveness to the above-mentioned problems is yet to be discovered.

In this paper, we attempt to solve the problem using Multipath TCP (MPTCP). With the rise of highly connected mobile devices which possesses multiple wireless interfaces, MPTCP becomes an underway promising protocol matching the needs of today's multipath networks [21]. MPTCP is a new transport layer protocol which allows exploiting several efficient Internet paths between a pair of hosts, while presenting a single TCP connection to the upper layer. IETF has released a wealth of experimental standards for MPTCP implementation in the Internet [8][20][7]. It not only increases the potential transmission bandwidth, but also helps with load balancing by using a coupled congestion control algorithm [20]. Additionally, various experiments have been conducted to verify that MPTCP smoothly transfers traffic on a breaking sub-flow to other flows during vertical handovers [17].

These facts motivate us to investigate the advantage of deploying MPTCP in LEO satellite networks. Intuitively, MPTCP in LEO satellite networks would increase bandwidth and compensate the impacts of long RTT, while during handover, MPTCP would be able to "softly" transfer traffic from a breaking link to other on-going links. LEO satellite networks indeed can reasonably support MPTCP for two reasons. First, LEO satellite constellation is a mesh network. Thus during stable states, we could split traffic on different MPTCP subflows into satellite-to-satellite paths. On the other hand, during handover states, since the ground terminal is under coverage of multiple satellites, it has multiple GSL interfaces to set up connections with different gateway satellites, which imitates a multihomed host. However, a series of challenges need to be addressed in order for MPTCP to be constantly supported in LEO satellite networks. First, in stable states, only one gateway satellite is visible to the ground terminal. It means that all the MPTCP sub-flows would have to use the same first (ground-to-satellite) and last (satellite-to-ground) hop. Then the gateway satellite will choose the same path[1] for all sub-flows since they destine to the same ground endpoint. This implies that multipath routing scheme should be deployed in order to split MPTCP sub-flows. Secondly, assuming that multipath routing is supported in LEO system, it is still a question how to distribute MPTCP sub-flows to different paths instead of mixing traffic on a single path. Thirdly, when handover is about to happen, the ground terminal can access multiple satellites, which gives MPTCP multiple interfaces to allocate MPTCP sub-flows. Should we switch back to single-path routing or stick to the multipath routing at this point?

Our answer to the first two questions lies within the design of a multipath routing scheme. In particular, we propose an on-demand multipath source routing (OMSR) scheme. After MPTCP sub-flows are initiated, they branch out from the gateway satellite into multiple disjoint paths OMSR discovered. We define the necessary interactions between transport and routing layers to ensure that MPTCP sub-flows are split into different paths. We also demonstrate the use case of such MPTCP-OMSR joint protocol in both stable and handover states. To answer the third question, we compare the performance of MPTCP using our routing shceme and using single-path routing through simulation. Simulation results verify that the MPTCP-OMSR protocol improves throughput and delay performance and handles handover gracefully. The remainder of the paper is structured as follows. In Section II, we present the design of the OMSR protocol. Then, the MPTCP-OMSR framework is introduced and illustrated in Section III. Section IV is devoted to simulation setup and results presentation. Performance of the framework is evaluated in terms of throughput and packet delay. Finally, conclusion is drawn in Section V.

II. On Demand Multipath Source Routing Design

In this section, we present the design of an On-demand Multipath Source Routing (OMSR) scheme. OMSR is an on-demand routing algorithm that builds multiple disjoint paths between satellite pairs. Such a routing scheme is desired because by selecting multiple routes, we can prevent congestion on a certain link or node and exploit network bandwidth more efficiently. At the same time, MPTCP can allocate traffic on different sub-flows to these routes.

The design of OMSR protocol can be divided into three parts. First, it should be able to detect maximally disjoint paths in LEO satellites. Then it should return information of usable paths based on certain path selection rules. At last, packets should be forwarded along usable routes based on the information collected.

[1]The path is chosen based on specific routing implementation, e.g. the shortest path between gateway satellites.

According to [15], multiple disjoint paths can be of three types: link-disjoint, node-disjoint and zone disjoint. In OMSR we will select multiple link-disjoint paths because of the low implementation complexity. Additionally, due to the fact that LEO satellites are geographically far apart from each other, link interference will not be a problem. There are many existing link-disjoint multipath routing algorithms in literature. OMSR chooses the Split Multipath Routing (SMR) [13] as its underlying route discovery algorithm. The goal of SMR is to discover and select multiple maximally disjoint paths between source and destination. To achieve this, a source routing based approach is deployed in the route discovery phase, where the information of the intermediate nodes that consist the route is included in the RREQ packet. Initially, the source node will include its node ID in the RREQ packet. Later on, other intermediate nodes will append their node IDs to the packet. To facilitate the selection of maximally disjoint paths, [13] pointed out that duplicate RREQ packets at certain relay node should not be discarded, but rebroadcasted through outgoing links other than the incoming link of a previous received RREQ. Upon receiving multiple RREQs, it is the destination node's responsibility to select multiple maximally disjoint paths.

After the destination makes the routing decision, it will send RREPs back to the source, again with the complete route information contained in the headers. When the source receives these RREPs, it can construct a multipath routing table with entries destined to the destination node. Once such a routing table is constructed, OMSR will look up the route information and forward data to the destination. The traditional source routing scheme is utilized to forward the data packets, i.e. the route information will be included in the packet header as a sequence of node IDs and ISL links, e.g. $R_{s\to d} = [ID_s, ISL_{s\to 1}, ID_1, ..., ID_n, ISL_{n\to d}, ID_d]$. To reduce the header overhead, different encoding schemes can be used, e.g. in [12] the so-called index-based encoding scheme is used which defines the path ID as a short hash for the sequence of globally known link interface identifiers.

Based on the design of the OMSR routing scheme, we will present how MPTCP interacts with it to split traffic over multiple disjoint paths in LEO satellite networks.

III. MPTCP-Routing Interaction

In this section, we introduce a framework for MPTCP-Routing design in LEO satellite network. We will start with the basic routing architecture in LEO system. Then we show how MPTCP packets are forwarded through a two-level routing table. We also demonstrate the use case of MPTCP in both stable and handover states.

A. Routing Architecture in LEO Satellite Network

The routing architecture we assume in this paper follows the framework of Internet routing over LEO satellite networks proposed in [16]. According to this framework, externally, LEO satellite network should work transparently with terrestrial IP routing, and internaly use its proprietry routing protocol. For our case, we propose an IP-OMSR based routing protocol. After the ground terminal registers and obtains the IP address of the LEO satellite network, it sends packets to the entry satellite with source and destination IP addresses using the default exterior routing protocol. Next, the entry satellite extracts the destination IP address, looks up the exit satellite in the IP routing table, and forwards the packets using OMSR routing scheme. To separate the IP and OMSR routing layers and avoid unnecessary routing updates, IP packets are tunnelled at the entry satellite as the payload data of OMSR packets, and source routing options are injected in the packets' header to specify the entire route information. This process is illustrated in Figure 1. When OMSR packets reach the exit satellite, they are decapsulated and IP packets are reassembled. Then the next IP hop is determined based on the destination IP address. Eventually, all the IP packets are forwarded to this address, again, using external IP routing. For succinctness, we refer to this IP-OMSR routing protocol as OMSR protocol.

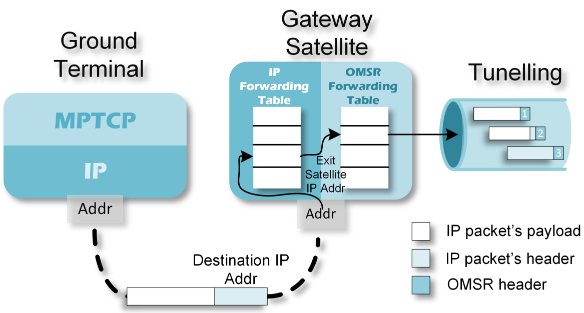


Fig. 1: IP-OMSR routing.

B. MPTCP-Routing Interaction

Now we show how the OMSR routing protocol supports MPTCP in LEO satellite networks. Several interactions between MPTCP and routing layer are defined to make this possible. Before we proceed, there are a few assumptions on the ground terminal and the system.

1) The ground terminals represent the users in LEO satellite system. They must be MPTCP-capable for the system to run MPTCP
2) The ground terminals should possess at least 2 IP addresses. Each address is configured to a unique network interface that can be used for a particular MPTCP sub-flow
3) The gateway satellites should support source address based routing in order to allocate MPTCP sub-flows to different disjoint paths.

The first assumption is straightforward. Regarding the second assumption, in fact, in Iridium system, Iridium phones have multiple interfaces which enable the users to access multiple channels simultaneously. This means that it is feasible for the users to possess multiple IP addresses. Besides, using multiple IP addresses would indeed facilitate the use of MPTCP. To simplify the discussion in this paper, we assume that each user possesses 2 interfaces, namely 2 IP addresses and MPTCP creates 2 sub-flows. For the third assumption, we note that it is desired to have MPTCP sub-flows go through different paths, preferably through disjoint paths. Therefore we suggest that source-address-based routing should be enabled at the gateway satellites. The motivation comes from [2].

1) Use Case in Stable State: Recall that by definition, ground terminals can see only one gateway, or entry satellite in the stable state. Based on assumption 2, even though a ground terminal has multiple IP addresses, all the connections will have to share one common entry satellite. This resembles the single next-hop router case in [10]. One possible solution proposed in [2] is to deploy source-address-based routing in the router. We use a similar approach to distribute MPTCP traffic to disjoint paths.

Suppose a pair of users A and B are communicating with each other using MPTCP, where A is the source and B the destination. A has two IP addresses A_1 and A_2, while B has B_1 and B_2. Assume that all the addresses are advertised, and A established 2 TCP sub-flow from A_1 to B_1 and A_2 to B_2. We denote them as $\langle A_1, B_1\rangle$ and $\langle A_2, B_2\rangle$. Then A starts to transmit data to the entry satellite S_1. S_1 will receive packets with IP address pair $\langle A_1, B_1\rangle$ and $\langle A_2, B_2\rangle$. As we introduced in the routing architecture, first S_1 will determine the exit satellite based the destination IP address. Suppose that B is currently registered with satellite S_2, also suppose OMSR discovers path P_1 and P_2. Before S_1 tunnels the packets and forwards them to S_2, we use source-address-based routing to split the traffic. This process is illustrated in Figure 2.

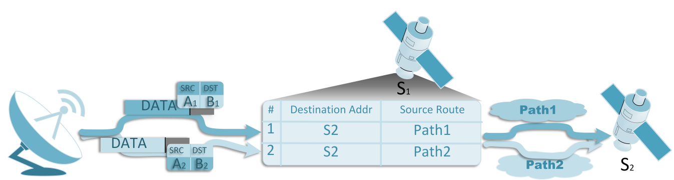


Fig. 2: Source-address-based routing in stable state.

When the packets reach S_2, the payload will be extracted and forwarded to address B_1 and B_2. In this way, we split two TCP sub-flows to two disjoint paths from entry satellite to exit satellite. The ACK packets from B to A are forwarded in a similar way along the reverse paths.

2) Use Case in Handover State: In handover states, the ground terminal can see two gateway satellites, thus can utilize them to distribute TCP sub-flows over different paths. Since the source and destination ground terminal enter handover state independently (from stable state), there are in total 3 combinations of source and destination states: 1) Source in handover state and destination in stable state; 2) Source in stable state and destination in handover state; 3) Both source and destination are in handover state. We will discuss the first case in more detail, and briefly summarize the other 2 cases.

Let us consider the same example where A is transmitting to B through S_1 and S_2 using P_1 and P_2. Suppose now S_1 is fading out and a new satellite S_1' comes up. At this point, A hears the advertisement from S_1' and will allocate sub-flow $\langle A_2, B_2\rangle$ to this new entry satellite based on its path delay measurement (assume P_1 is the shortest path so P_2 has a longer delay than P_1). At the new entry S_1', upon receiving the packets from $\langle A_2, B_2\rangle$, it will execute route discovery using OMSR and explore two disjoint paths from S_1' to S_2. Then $\langle A_2, B_2\rangle$ will be accommodated by the shortest route.

Now we briefly talk about the rest two cases. For the second case, where A stays in stable state, and B sees a new satellite S_2' coming up then connects to it with B_2. After this point,

TABLE I: NS-2 Iridium system parameters

Parameter	Value
Altitude	$780km$
Planes	6
Satellites per plane	11
Inclination	86.4°
Inter-plane separation[2]	31.6°
Seam separation [3]	22°
Elevation mask[4]	8.2°
ISLs per satellite	4
ISL bandwidth	$9Mb/s$
Up/downlink bandwidth	$2Mb/s$
Packet size	$1000Bytes$

the gateway satellites for $\langle A_2, B_2\rangle$ will be updated to S_1, S_2'. The two sub-flows will use two shortest paths between (S_1, S_2) and (S_1, S_2'). For the third case, no matter which ground terminal enters handover state first, the transition period will be the same as either case 1 or case 2. Later when the other ground terminal enters handover state, the two sub-flows will end up using two shortest paths between two different pair of gateway satellites (S_1, S_2) and (S_1', S_2').

For all the cases, eventually we will change back to stable state and all the sub-flows will end up using the same two gateway satellites. Again, source-address-based routing should be used to split the traffic.

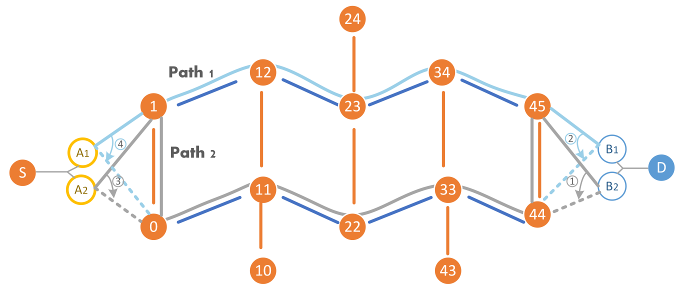
IV. Performance Evaluation

In this section, we present the simulation results of the MPTCP-OMSR framework and evaluate its performance. First we introduce the simulation set up. We choose Iridium constellation as a representative of LEO satellite networks and use Network Simulator 2 (NS-2) as the simulator. In our simulation, we adopt the default configuration for Iridium system. Important parameters are listed in Table I.

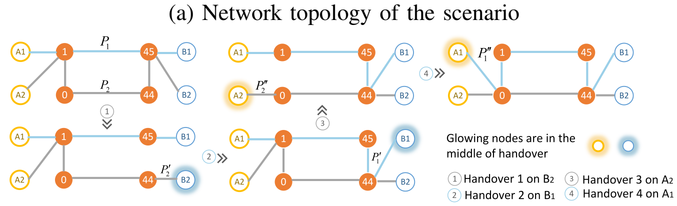
Next we will demonstrate the test scenario and present simulation results. Finally, we evaluate the performance of the MPTCP-OMSR protocol based on its comparison with two traditional "single-path" TCP variants and MPTCP with single shortest path routing.

A. Simulation Design and Results

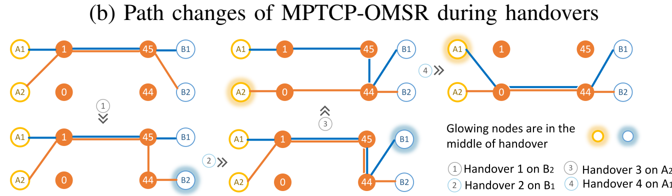
To test the MPTCP-OMSR protocol in NS-2, we design a test scenario with both source and destination side handovers. Figure 3a shows a graph illustrating the scenario. The source node S is located at position $(15.51^{\circ}N,\ 1^{\circ}W)$ and has two IP addresses A_1, A_2. The destination D is located at position $(17.4^{\circ}N,\ 127.55^{\circ}E)$ and has B_1, B_2. S and D establishes two sub-flows $\langle A_1, B_1\rangle$ and $\langle A_2, B_2\rangle$. Initially both the two sub-flows share the entry satellite S_1 and exit satellite S_{45}. However OMSR selects two disjoint paths P_1 and P_2 to split the sub-flows. 1 After certain time, D sees the upcoming satellite S_{44} and connects to it using B_2 (since $\langle A_2, B_2\rangle$ has longer propagation delay). P_2 changes to P_2' accordingly. 2 After a while, handover at D is triggered and P_1 changes to P_1'. 3 Next, when handover at S is about to happen, S firsts connects to S_0 with A_2, and P_2' changes to P_2''. 4 Finally after S handovers, P_1' changes to P_1''. This process is illustrated in Figure 3b. We note that the whole test scenario can be divided to three states: stable state before handover $\mathcal{SS}_{bf}$, handover state $\mathcal{HS}$, and stable state after handover $\mathcal{SS}_{af}$. Therefore the test scenario includes all the use cases discussed in Section III and can thus evaluate our protocol comprehensively.



(a) Network topology of the scenario



(b) Path changes of MPTCP-OMSR during handovers



(c) Path changes of MPTCP-SSP during handovers

Fig. 3: Test scenario.

For the sake of comparison, we also test two traditional "single-path" TCP variants, TCP Tahoe [11] and TCP Hybla [4], in the same scenario. TCP Tahoe is widely used in the current Internet which implements a standard congestion control algorithm introduced in [11]. Compared to TCP Tahoe, TCP Hybla is more suitable for high BDP scenarios such as LEO satellite constellation. As reported in [4], Hybla achieves better throughput performance in presence of long RTT and high packet loss rate than most TCP variants. Comparing MPTCP-OMSR with TCP Tahoe and Hybla will give us a clearer view of the performance gain MPTCP can bring.

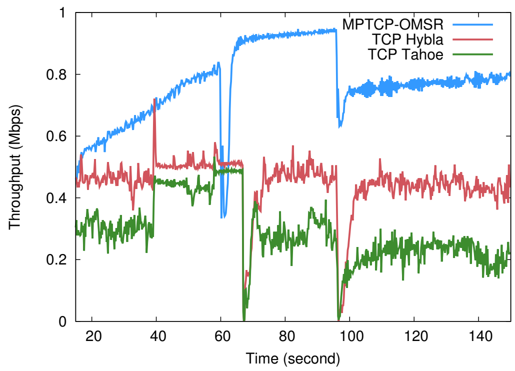
In the simulation, we randomly distribute 48 pairs of transmission within the area of $(5^{\circ}S \sim 35^{\circ}N, 10^{\circ}W \sim 130^{\circ}E)$ as background traffic. This simulates a regionally congested LEO satellite network where each TCP flow will have a share of $0.36Mbps$ bandwidth on average. We run 10 independent simulations with different random seeds and obtain the following throughput and packet delay results (with Confidence Intervals included).

As shown in Figure 4a, while TCP Tahoe approximately reaches the average bandwidth share ($0.36Mbps$) during $\mathcal{SS}_{bf}$,

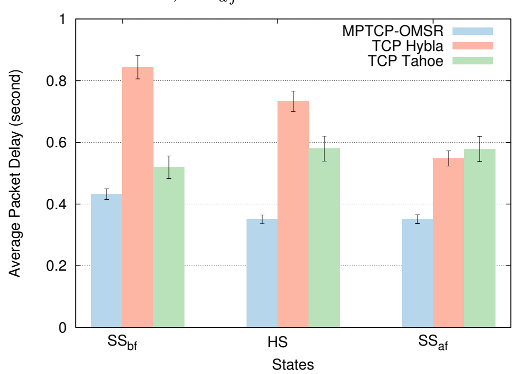
[2]The co-rotating planes are spaced 31.6° apart.

[3]The counter-rotating planes (orbit 1 and 6) are spaced 22° apart.

[4]The minimum elevation angle for an earth station is defined as elevation mask.

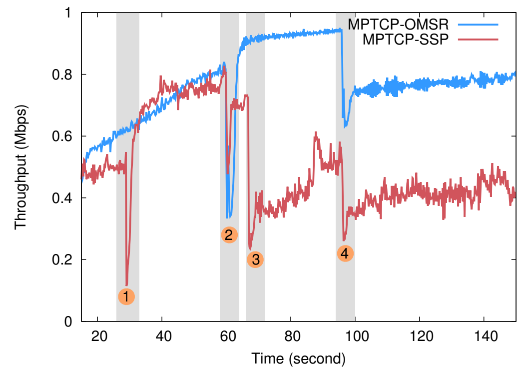


(a) Throughput performance comparison. MPTCP-OMSR: $\mathcal{SS}_{bf}$ 15-30 second, $\mathcal{HS}$ 30-96 second, $\mathcal{SS}_{af}$ 96-150 second. "Single-path" TCP: $\mathcal{SS}_{bf}$ 15-66 second, $\mathcal{HS}$ 66-96 second, $\mathcal{SS}_{af}$ 96-150 second.

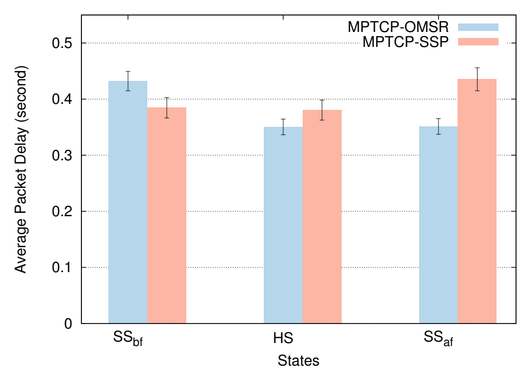


(b) Packet delay performance comparison.

Fig. 4: Performance comparison between MPTCP and "single-path" TCP



(a) Throughput performance comparison. $\mathcal{SS}_{bf}$: 15-30 second; $\mathcal{HS}$: 30-96 second; $\mathcal{SS}_{af}$: 96-150 second. 1: Handover on B_2; 2: Handover on B_1; 3: Handover on A_2; 4: Handover on A_1.



(b) Packet delay performance comparison.

Fig. 5: Performance comparison between MPTCP-OMSR and MPTCP-SSP.

its throughput drops to zero whenever handover happens. Additionally, during $\mathcal{HS}$[5], the average throughput of TCP Tahoe is lower than its normal level. In comparison, TCP Hybla achieves higher throughput than Tahoe during $\mathcal{SS}_{bf}$. But again, its throughput drops to zero when handover happens. Because of its aggressiveness, Hybla grabs more bandwidth even when the path quality fluctuates during $\mathcal{HS}$. Finally, we can see that MPTCP-OMSR obviously outperforms the other two variants in term of throughput. First, it has the highest throughput during $\mathcal{SS}_{bf}$. This is not surprising since MPTCP-OMSR uses two TCP flows going through disjoint paths. Second, when handover happens, the throughput of MPTCP-OMSR will not drop to zero. This verifies that MPTCP can indeed be utilized for "soft" handover. Third, the result shows that during $\mathcal{HS}$ and $\mathcal{SS}_{af}$, the throughput is even higher than during $\mathcal{SS}_{bf}$. This is due to the fact that OMSR guarantees that both TCP sub-flows are provided with shortest paths during $\mathcal{HS}$, while only one of the flows is using the shortest path during $\mathcal{SS}_{bf}$.

The packet delay result is shown in Figure 4b. We can observe that for all the three states, MPTCP-OMSR has shorter packet delay.

[5]We note that the handover state of MPTCP lasts longer than TCP Tahoe and Hybla, as shown in Figure 4a. This is because MPTCP experiences 2 more handovers when allocating one sub-flow to upcoming satellites.

B. MPTCP with and without OMSR

From the comparison with TCP Hybla and Tahoe, we see that MPTCP-OMSR outperforms these two "single-path" TCP in both throughput and delay. However an interesting question to ask is whether a multipath routing scheme such as OMSR is necessary for MPTCP to work in LEO satellite network. In this section, we present the comparison between MPTCP with OMSR routing and MPTCP with the single shortest path (SSP) routing. During $\mathcal{SS}_{bf}$ and $\mathcal{SS}_{af}$, MPTCP-SSP chooses the same shortest path for $\langle A1, B1\rangle$ and $\langle A2, B2\rangle$, while during $\mathcal{HS}$, shortest paths between different gateway satellites are computed independently. Results are shown in Figure 5a and 5b.

It can be seen that MPTCP-SSP has lower throughput than MPTCP-OMSR during $\mathcal{SS}_{bf}$. Such result makes sense since in this case, both of the two TCP sub-flows share the same shortest path, so MPTCP uses its coupled congestion control algorithm to restrict the data rate and be fair to other users. On the other hand, MPTCP-OMSR uses two disjoint paths during this state, so its throughput is slightly higher. This also implies that the other disjoint path will not bring too much throughput gain since it has a longer delay. When it comes to $\mathcal{HS}$ and $\mathcal{SS}_{af}$, MPTCP-OMSR has substantially higher throughput and slightly smaller delay than MPTCP-SSP. In addition, as we mark in Figure 5a, compared to

MPTCP-SSP, MPTCP-OMSR is not affected by the handovers of $\langle A_2, B_2 \rangle$. To see why MPTCP-OMSR performs so well during $\mathcal{HS}$ and can maintain the performance after handover, we demonstrate the path selection of OMSR and SSP during $\mathcal{HS}$ with Figure 3b and 3c. From the figures we see that, even though the first and last GSL of $\langle A_1, B_1 \rangle$ and $\langle A_2, B_2 \rangle$ keeps changing whenever link layer handover happens, OMSR always retains path segment $S_1 \rightarrow S_{45}$ for $\langle A_1, B_1 \rangle$ and $S_0 \rightarrow S_{44}$ for $\langle A_2, B_2 \rangle$. In contrast, SSP blindly selects shortest path after each handover and causes a larger scope of path changes, e.g. from $A_2 \rightarrow S_1 \rightarrow S_{45} \rightarrow S_{44} \rightarrow B_2$ to $A_2 \rightarrow S_0 \rightarrow S_{44} \rightarrow B_2$, and $A_1 \rightarrow S_1 \rightarrow S_{45} \rightarrow S_{44} \rightarrow B_1$ to $A_1 \rightarrow S_0 \rightarrow S_{44} \rightarrow B_2$. As a consequence, MPTCP-SSP is more vulnerable to frequent handovers and performs poorly in a scenario similar to our test case. MPTCP with OMSR scheme provides more robust path selection strategy against satellite handover and achieves much higher throughput and shorter packet delay.

V. Conclusion

In this work, we proposed and evaluated a Multipath-TCP and On-demand Multipath Source Routing (MPTCP-OMSR) protocol over LEO satellite networks. In order to support MPTCP, we designed the OMSR routing scheme which s-elects multiple link-disjoint paths to accommodate MPTCP sub-flows. Source-address-based routing is enabled to split TCP sub-flows. Inside the LEO constellation, multipath source routing is used between gateway satellites. We also showed that during the handover state, OMSR provides MPTCP with two shortest paths by leveraging the fact that ground terminal stays in the overlapped area of two satellites. Through simulation, we verified that our MPTCP-OMSR framework can increase throughput compared to “single-path” TCP variants. It also achieves the goal of “soft” handover and prevents transmission from being interrupted. Moreover, from the comparison with an MPTCP implementation using single shortest path routing, we showed how the sophisticated path selection strategy of OMSR helps MPTCP improve throughput and delay during handover state. It implies that multipath routing scheme is indeed necessary to better support MPTCP and exploit its best capabilities.

References

[1] M. Allman and J. Griner. Tcp behavior in networks with dynamic propagation delay. In *Global Telecommunications Conference, 2000. GLOBECOM ’00. IEEE*, volume 2, pages 1103–1108 vol.2, 2000.

[2] S. Barré, C. Paasch, O. Bonaventure, et al. Multipath tcp-guidelines for implementers. Technical report, UCL, 2011.

[3] C. Caini and R. Firrincieli. Tcp hybla: a tcp enhancement for heterogeneous networks. *International journal of satellite communications and networking*, 22(5):547–566, 2004.

[4] C. Caini, R. Firrincieli, and D. Lacamera. Comparative performance evaluation of tcp variants on satellite environments. In *Communications, 2009. ICC’09. IEEE International Conference on*, pages 1–5. IEEE, 2009.

[5] Y. Chotikapong and Z. Sun. Evaluation of application performance for tcp/ip via satellite links. In *Satellite Services and the Internet (Ref. No. 2000/017), IEE Seminar on*, pages 4/1–4/4, 2000.

[6] P. K. Chowdhury, M. Atiquzzaman, W. D. Ivancic, et al. Handover schemes in satellite networks: State-of-the-art and future research directions. *IEEE Communications Surveys and Tutorials*, 8(1-4):2–14, 2006.

[7] A. Ford, C. Raiciu, M. Handley, S. Barre, J. Iyengar, et al. Architectural guidelines for multipath tcp development. *IETF, Informational RFC*, 6182:2070–1721, 2011.

[8] A. Ford, C. Raiciu, M. Handley, O. Bonaventure, et al. Tcp extensions for multipath operation with multiple addresses. *IETF MPTCP proposal-http://tools. ietf. org/id/draft-ford-mptcp-multiaddressed-03. txt*, 2011.

[9] S. Ha, I. Rhee, and L. Xu. Cubic: a new tcp-friendly high-speed tcp variant. *ACM SIGOPS Operating Systems Review*, 42(5):64–74, 2008.

[10] M. Handley, C. Raiciu, and M. Bagnulo. Outgoing packet routing with mp-tcp. *draft-handley-mptcp-routing-00 (work in progress)*, 2009.

[11] V. Jacobson. Congestion avoidance and control. In *ACM SIGCOMM Computer Communication Review*, volume 18, pages 314–329. ACM, 1988.

[12] B. Jianjun, L. Xicheng, L. Zexin, and P. Wei. Compact explicit multi-path routing for leo satellite networks. In *High Performance Switching and Routing, 2005. HPSR. 2005 Workshop on*, pages 386–390. IEEE, 2005.

[13] S.-J. Lee and M. Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. In *Communications, 2001. ICC 2001. IEEE International Conference on*, volume 10, pages 3201–3205. IEEE, 2001.

[14] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang. Tcp westwood: Bandwidth estimation for enhanced transport over wireless links. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 287–297. ACM, 2001.

[15] N. Meghanathan. Performance comparison of link, node and zone disjoint multi-path routing strategies and minimum hop single path routing for mobile ad hoc networks. *arXiv preprint arXiv:1011.5021*, 2010.

[16] P. Narvaez, A. Clerget, W. Dabbous, et al. Internet routing over leo satellite constellations. In *Third ACM/IEEE International Workshop on Satellite-Based Information Services (WOSBIS98)*. Citeseer, 1998.

[17] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure. Exploring mobile/wifi handover with multipath tcp. In *Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design*, pages 31–36. ACM, 2012.

[18] C. Pan, H. Luo, L. Yang, and D. Wei. Leo satellite communication system handover technology and channel allocation strategy. *International Journal of Innovative Computing, Information and Control*, 9(11):4595–4602, 2013.

[19] E. Papapetrou, S. Karapantazis, and F.-N. Pavlidou. Distributed on-demand routing for leo satellite systems. *Computer Networks*, 51(15):4356–4376, 2007.

[20] C. Raiciu, M. Handley, and D. Wischik. Coupled congestion control for multipath transport protocols. *draft-ietf-mptcp-congestion-01 (work in progress)*, 2011.

[21] C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, M. Handley, et al. How hard can it be? designing and implementing a deployable multipath tcp. In *NSDI*, volume 12, pages 29–29, 2012.

[22] F. Shen, H. Yu, and X. Zhang. Hats: a handover optimized routing algorithm for the low earth orbit (leo) satellite network. In *Information, Communications and Signal Processing, 2009. ICICS 2009. 7th International Conference on*, pages 1–5. IEEE, 2009.

[23] H. Uzunalioğlu, I. F. Akyildiz, Y. Yesha, and W. Yen. Footprint handover rerouting protocol for low earth orbit satellite networks. *Wireless Networks*, 5(5):327–337, 1999.

[24] M. Werner, A. Jahn, E. Lutz, and A. Bottcher. Analysis of system parameters for leo/ico-satellite communication networks. *Selected Areas in Communications, IEEE Journal on*, 13(2):371–381, 1995.