# Traffic Optimization in Software Defined Naval Network for Satellite Communications

Pengyuan Du*, Fan Pang*, Torsten Braun[†]
Mario Gerla*, Ceilidh Hoffmann [‡], Jae H. Kim [‡]
*Department of Computer Science,University of California, Los Angeles, USA
[†]Institute of Computer Science, University of Bern, Switzerland
[‡]Boeing Research & Technology, Seattle, USA
{pengyuandu, fpangmomo, gerla}@cs.ucla.edu, braun@inf.unibe.ch,
{ceilidh.hoffmann2, jae.h.kim}@boeing.com

***Abstract*—Naval surface fleets of the United States and its allies rely on multiple satellite communication systems (SATCOM) for onboard communication with other entities such as ships, shore nodes and hosts from external networks. Current practice is for an onboard ship router to select a particular SATCOM link for each outgoing traffic flow based on a mission-specific routing policy. In this paper we propose an alternative solution by viewing the multi-SATCOM link utilization task as a traffic engineering and load balancing problem—in particular, as a Multi-Commodity Flow (MCF) optimization problem. We propose using the Flow Deviation Method (FDM) as a network-wide optimal load-balancing solution that maximizes total throughput and minimizes traffic flow delay and jitter. Our approach is equally valid for both UDP and TCP flows. Network-wide global optimization is carried out via a central controller in a Software Defined Networking (SDN) framework. For TCP flows we propose a novel solution that combines the best attributes of Multi-Path TCP, SDN and FDM. Compared to single-path TCP or MPTCP-SDN without FDM-based traffic optimizer, our proposed combined scheme is more efficient in bandwidth utilization, delay/jitter minimization and also robust against jamming and intermittent link failure. Network performance results are validated via Mininet emulation tests.**

***Index Terms*—Naval Satellite Network, Multipath TCP, Software-defined Networking, Multi-Commodity Flow**

## I. Introduction

In naval battlefield networks, communications among surface vessels, aircrafts and shore nodes is enabled by governmental and commercial satellite communication systems (SATCOM). Each navy ship is equipped with one or more SATCOM terminals. A common router arbitrates traffic from shipboard local area networks (LAN) to these terminals, which is then forwarded to the SATCOM network through a dedicated uplink. In the conventional policy-based routing architecture [7], each LAN traffic flow is served by a specific SATCOM based on its service or mission thread type. Therefore, a SATCOM system can become congested due to mission overload while the capacity of other SATCOM systems are under-utilized.

In our previous paper [17] we described architectural details of Software Defined Networking (SDN) implementation for the naval fleet network with multi-SATCOM connectivity, which we termed Software-Defined Naval Network using SATCOM services (SDN-SAT). In SDN-SAT, onboard LAN hosts use either UDP or Multipath TCP (MPTCP) transport protocols for their LAN traffic flows. MPTCP splits a packet stream into multiple TCP subflows and the global SDN controller distribute the subflows to multiple SATCOMs. The SDN solution also breaks the tight integration of control and data plane functions that is commonly seen in the shipboard hardware routers, enabling the adoption and dynamic reconfiguration of innovative networking protocols.

The problem not yet addressed in [17] is network-wide traffic optimization and bandwidth allocation among SDN-SAT nodes. We assume that every SATCOM system has limited bandwidth that is assigned to SDN-SAT and every ship has a desired transmission rate (traffic demand) that can be estimated based on metered LAN traffic statistics. We also assume that the assigned SATCOM capacities are known to the global SDN controller and ships faithfully report their traffic demand to the global controller. Without a global bandwidth broker, navy ships rely on default scheduling mechanisms (policy-based routing, MPTCP, greedy load balancing algorithms etc.) to share multiple SATCOM resources. In the MPTCP case the local scheduler is greedy and unaware of demands from other flows, resulting in suboptimal load balancing outcomes.

In this paper, we propose an optimal load balancing solution by viewing the multi-SATCOM link utilization task as a Multi-Commodity Flow (MCF) optimization problem. Our formulation is closely related to the “Routing Assignment” problem in [10], whose solution is given by the Flow Deviation Algorithm (FDM). We propose using FDM as a network-wide optimal load-balancing solution that maximizes total throughput and minimizes traffic flow delay and jitter. Load balancing operation is carried out within the SDN framework via the SDN controller serving as the centralized traffic broker for SDN-SAT. The broker accepts traffic demands from navy ships and has knowledge of SATCOM capacities. In turn it instructs each ship in selecting a set of SATCOM channels and in configuring bandwidth demands on these channels. To validate network performance results and the feasibility of SDN implementation in current naval SATCOM systems, our proposed optimization algorithms and SDN protocols are tested in Mininet-based SDN emulation platform.

The rest of the paper is structured as follows. Section II introduces related work. Section III reviews the SDN-SAT system of [17]. Then, we construct the MCF formulation of

SDN-SAT traffic load balancing problem and provide details of the proposed optimization solution in Section IV. Mininet emulation testbed details are discussed in Section V, followed by performance analysis in Section VI. Finally, conclusions are drawn in Section VII.

## II. Related Work

### A. Multipath TCP

Multipath TCP (MPTCP) [8] is an enhancement to conventional single-path TCP protocol by exploiting the availability of multiple communication paths between a pair of hosts for the benefit of increased reliability in end-to-end data transport. Applications in Layer 5 and above "see" a single logical “master” TCP connection even though multiple sub-flows may be running underneath—each flow as a conventional TCP connection. MPTCP runs between the applications and TCP sub-flows. It reconstitutes out-of-order packets from different sub-flows and performs *coupled* congestion control among them. Further details of MPTCP implementation and draft standards for the current Internet are provided in [9].

In terrestrial wireless and mobile networks, MPTCP can be implemented in multihomed devices (e.g. smart-phones) equipped with multiple wireless communication links such as cellular and WiFi. Such devices can establish sub-connections through multiple paths and thereby enhance end-to-end data delivery performance. For SATCOM systems, [5], [6] describes MPTCP benefits in improving throughput and overcoming handover-based intermittence issue in Low Earth Orbiting satellite networks. An integration of MPTCP and network coding is shown to be beneficial for satellite systems with lossy channels [2], [3].

### B. Software Defined Networking (SDN)

SDN is an emerging concept that decouples the network control plane from data forwarding plane. Commodity hardware routers and switches are treated as simple packet forwarding devices, thereby promoting logical centralization of network control [14]. The intelligence associated with routing, flow control and other traffic engineering processes are executed at a remote location (known as the controller) and instruction sets are relayed to forwarding devices using an SDN protocol such as OpenFlow [16]. This paradigm provides a unified and global view of the network and simplifies policy enforcement, network (re)configuration and scalability. An important SDN topic that has received growing attention and the focus of this paper is traffic engineering [1], [19].

### C. Multi-Commodity Flow (MCF)

Multi-Commodity Flow (MCF) problem is in the class of network flow problems in which distinct commodities exist between source-destination pairs. It has been widely applied to the routing design problem in systems where multiple traversing paths exist—e,g. computer networks [20] and vehicular navigation maps [13]. MCF can also model and solve bandwidth allocation problems [22], [21]. In recent years, MCF models and algorithms have been adapted from a traffic engineering perspective. See [15] where MCF models are used to optimize load balancing, routing cost and average delay. Traffic engineering in SDN framework is treated as an MCF problem in [19].

## III. Review of SDN-SAT System

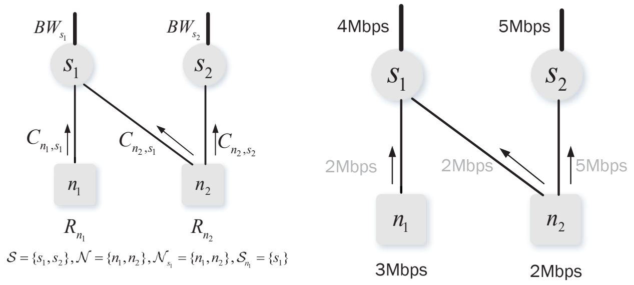
In our previous paper [17], we described the system architecture of SDN-based naval multi-SATCOM system. Here we briefly review its main features and components. SDN-SAT consists of ships (as nodes), multiple SATCOM systems (as data service providers), onboard LAN users (as SATCOM customers), a common router (serving as an SDN switch) per ship for load balancing of SATCOM resources and an SDN controller serving as the SATCOM bandwidth broker. The SDN controller monitors available supply (i.e. SATCOM capacities) and matches them against demand (i.e. ships' LAN traffic load). The controller itself has multi-SATCOM connectivity and thus can be co-located on a ship, a shore node or the ground hub of a SATCOM system.

From the perspective of a LAN user, SDN-SAT enhances its data delivery efficiency (due to global load balancing operation of all LAN flows) and provides redundancy as well as robustness against SATCOM link failure (via multi-route delivery options). In SDN-SAT system, each ship node has a desired transmission rate (i.e. aggregate demand of LAN users) and each SATCOM has a fixed capacity that is shared by all ship nodes that have access to it. The topic that has not been addressed in [17]—and the focus of this paper—is the controller's load balancing optimization method—both algorithmic details and implementation in SDN framework.

## IV. Bandwidth Allocation as a Multi-Commodity Flow Problem

In SDN-SAT the set of navy ships (nodes) is denoted by $\mathcal{N}$ and the set of SATCOM systems as $\mathcal{S}$. Network topology is characterized by $\mathcal{N}_s$: the set of ships attached to any SATCOM system $s \in \mathcal{S}$, and $\mathcal{S}_n$: the set of SATCOM systems accessible by any ship $n \in \mathcal{N}$, respectively. Certain ships have full connectivity to all SATCOMs, i.e. $\mathcal{S}_n = \mathcal{S}$ while others only have partial connectivity, i.e. $\mathcal{S}_n \subset \mathcal{S}$. Each ship-satellite connection (i.e. SATCOM channel) is composed of an uplink and a downlink.

To avoid ambiguity in terminology and usage, we define the following: Each SATCOM system $s$ allocates total data rate $BW_s$ (in bits per second)—which we call SATCOM "capacity"—that is shared among SDN-SAT nodes. Each SATCOM channel can carry per-ship data rate (i.e. SATCOM "bandwidth"), denoted as $C_{n,s}$ for SATCOM $s$ and ship $n$. The LAN user set of ship $n$ is $\mathcal{L}_n$. Each LAN user's traffic demand is called user "bandwidth" and denoted as $r_{l,n}$, $l \in \mathcal{L}_n$. Each ship $n$ has total bandwidth demand $R_n = \sum_{\forall l} r_{l,n}$ from its LAN users to destinations on another ship, shore nodes or external hosts switched via a SATCOM gateway. In this model we assume a rate-stationary system where both demands and supplies do not change over time. Our algorithms can be readily adapted to dynamic systems by running periodic updates. An example of a 2-node, 2-SATCOM SDN-SAT system is given in Fig. 1a. Finally, for notational convenience we assume each LAN user traffic applies to a single UDP or TCP flow such that there is an entry per flow in the flow table in a SDN switch for selective treatment such as SATCOM link selection, rate metering, etc.



(a) Access network

(b) Naive bandwidth allocation

Fig. 1: Bandwidth allocation in SDN-SAT

### A. Problem Formulation

The task of the SDN controller as a traffic broker is to provide instruction sets to SDN switches for their respective bandwidth allocations $C_{n,s}$ for $n \in \mathcal{N}_s, \forall s \in \mathcal{S}$, given $\mathcal{S}, \mathcal{N}, \mathcal{S}_n, \mathcal{N}_s, R_n, BW_s$. A naive approach is to distribute $BW_s$ equally among the connected ships in $\mathcal{N}_s$. In Fig. 1b, 2 Mbps is assigned to each ship on SATCOM $s_1$ while the entire capacity 5 Mbps of $s_2$ is available to $n_2$ since $n_1$ cannot access $s_2$. In this scenario if both $n_1$ and $n_2$ were to utilize their assigned SATCOM bandwidths, $s_1$ would be saturated, resulting in large packet delay and jitter. Our optimization objective is to balance traffic demands versus assigned SATCOM bandwidths across all nodes and SATCOM systems such both SATCOM capacity and per-ship bandwidth saturation cases are avoided—provided that SDN-SAT network is under-saturated; i.e. total SATCOM capacity supply is greater than total node traffic demand.

In an under-saturated service-demand traffic model, packet delay $D$ can be expressed as the inverse of residual bandwidth:

$$D \propto \frac{L}{BW_s - \sum\limits_{n \in \mathcal{N}_s} C_{n,s}}$$

where $L$ is the packet size. As traffic demand approaches SATCOM bandwidth, packet delay becomes excessive ($D \to \infty$). We formulate the bandwidth allocation problem in SDN-SAT in the following MCF form, which is an adaptation to the "Routing Assignment" problem in [10]:

$$\min_{C_{n,s}} T = \frac{1}{\gamma} \sum_{s \in \mathcal{S}} \frac{L \cdot f_s}{BW_s - f_s} \tag{1}$$

$$s.t.: f_s = \sum_{n \in \mathcal{N}_s} C_{n,s} \leq BW_s, \forall s \in \mathcal{S}$$

$$\sum_{s \in \mathcal{S}_n} C_{n,s} = R_n, \forall n \in \mathcal{N}, \, C_{n,s} \geq 0$$

where $\gamma = \sum\limits_{n \in \mathcal{N}} R_n$. This formulation minimizes total packet delay $T$ while satisfying both capacity and bandwidth demand constraints.

### B. The Flow Deviation Method

The Flow Deviation Method (FDM) [10] is a gradient-based iterative algorithm that converges to the solution of the MCF problem 1. Details of the algorthm as pseudo-code is provided in Algorithm 1. For successful execution, a complete network topology must be specified. In general end-to-end data delivery over a SATCOM system is either 2 or 4 communication hops. If the destination node is external (non SDN-SAT), then its trafic is forwarded via the ground hub gateway to external networks—thus two hops with uplink access only. For SDN-SAT node-to-node data delivery, four hops are required:

$$\text{node} \rightarrow \text{satellite} \rightarrow \text{ground hub} \rightarrow \text{satellite} \rightarrow \text{node}$$

In this paper we consider uplink bandwidth optimization only;[1] Based on this model all source nodes have a common destination: the SATCOM ground hub as shown in Fig. 2. Our model also assumes that the feeder link between the satellite and its ground hub has enough capacity to accommodate total uplink data rate $BW_s$. We denote the resulting network topology as graph $\mathcal{G}$. The FDM algorithm splits ships' total bandwidth demand into flows and assigns them to multiple paths (SATCOM channels from different SATCOM systems), aiming to balance the load and minimize packet delay. Once converged, FDM returns the flow assignment on each path from every source-destination pair. For the example in Fig. 1b, FDM-based allocations are $C_{n_1,s_1}$ = 3 Mbps, $C_{n_2,s_1}$ = 0 Mbps and $C_{n_2,s_2}$ = 2 Mbps. Further details of FDM and its proof of correctness are provided in [10][12].

## V. System Implementation

In this section we discuss design and implementation details of the SDN-SAT traffic optimization algorithms and associated SDN protocols on the SDN network emulator Mininet [4]. In summary, our SDN emulation testbed uses Mininet 2.1.0, Floodlight 1.2 (for the controller), OpenFlow 1.3 Software Switch and MPTCP Linux Kernel Implementation v0.90. The MPTCP congestion control is set as Linked Increase Algorithm `lia`, the path manager is `fullmesh`. Experiments are run on a Linux Ubuntu 16.04 machine. Performance analysis is limited to TCP flows, highlighting unique features of MPTCP and SDN; however, our optimization methods and emulation procedures are equally valid for UDP flows.

### A. Mininet-based SDN Environment

Mininet provides a virtual network environment that consists of hosts, SDN switches, and SDN controller. Each host is a virtual representation of the physical machine. The hosts enable Multipath TCP (MPTCP) protocol to access available bandwidth from multiple SATCOMs. For SDN switches and the controller, we use OpenFlow 1.3 Software Switch and Floodlight 1.2 controller, respectively. An element in OpenFlow 1.3 Software Switch that is essential to our implementation is the *Meter*. A Meter is used to measure and control the rate of flow. It triggers a meter band if the

[1] Joint uplink-downlink optimization research is currently ongoing.

packet rate passing through exceeds a predefined threshold—hence, also referred to as the rate limiter. Floodlight controls the network by communicating with OpenFlow switches via the OpenFlow protocol.

### *B. Floodlight Implementation*

We implement the FDM-based traffic optimization in the Floodlight controller. Three main functionalities we add to the Floodlight controller are *MPTCP-aware forwarding*, *the FDM algorithm* and *bandwidth metering*. Our Floodlight implementation is available at https://github.com/momokipf/floodlight_mptcp/tree/add_feature.

*1) MPTCP-aware Forwarding:* To support MPTCP we rely on two components in the Floodlight controller: the Topology Manager (TM) and the Forwarding Module (FM). The TM builds the global topology and enables other modules to get topology information such as the set of paths between hosts. The FM is one customer of these services. When a new MPTCP session joins the network, the FM assigns each subflow a different path from the set of paths between source and destination. This requires the FM to look into the MPTCP option and match each subflow to an existing MPTCP session. We reuse the matching algorithm proposed in our previous paper [17].

*2) FDM Algorithm: FDMCalculator* implements the FDM algorithm and populates the bandwidth limiter on target links. The input to this module is the up-to-date global topology as well as the traffic demands and SATCOM capacities. Similar to the FM, the FDM module queries TM to retrieve links and nodes in current topology. The current active paths (flows) will be stored in the FDM module in order to instruct FDM computation. As for the demand and bandwidth resource input, we implement REST API, which accepts user-specified configuration at runtime.

```
Input: Graph $\mathcal{G}$ resulted from the SDN-SAT network.
Output: The optimal routing assignment.
Define: $f_l^k$ the traffic on link $l$ in iteration $k$; $F_k$ the traffic
  profile in iteration $k$ whose elements are $f_l^k$; $iBW_l^k$ the
  inflated capacity of link $l$ in iteration $k$; $d_l^k$ the length (cost)
  of link $l$ in iteration $k$.
Initialize: Link length $d_l^0 = \frac{\partial T}{\partial f_l^0} = \frac{L}{\gamma}\frac{BW_l}{(BW_l - f_l^0)^2}$, $f_l^0 = 0$, and
  $BW_l$ is the link capacity.
while $F_k$ does not converge do
    for Link $l$ do
        $f_l^{k+1} \leftarrow 0$;
    end
    for $n \in \mathcal{N}$ do
        Find the shortest path $\pi_n$ from $n$ to destination; for
          Link $l$ along $\pi_n$ do
            $f_l^{k+1} \leftarrow R_n$;
        end
    end
    Binary search $x^* \in [0,1]$ such that
      $F_{k+1} \leftarrow x^* \cdot F_{k+1} + (1 - x^*) \cdot F_k$
    yields the optimal $F_{k+1}$ that minimizes $T$;
    $iBW_l^{k+1} \leftarrow \alpha \cdot BW_l$, where $\alpha = \max(1, \max_l \frac{f_l^{k+1}}{BW_l})$;
    for Link $l$ do
        $d_l^{k+1} \leftarrow \frac{L}{\gamma}\frac{iBW_l^{k+1}}{(iBW_l^{k+1} - f_l^{k+1})^2}$;
    end
end
```

**Algorithm 1:** Pseudocode of FDM algorithm

The *FDMCalculator* module has several components. The *FDMTopology* class is a topology builder that translates the actual network topology (built from Mininet script) into a logical topology for the ease of FDM computation. At the same time, the topology builder reads the demand and capacity passed by the REST API. These will be the input of the FDM algorithm. The second component is the *FlowDeviationMethod* class, which implements the main FDM algorithm. Finally, the *FDMCalculator* and *IFDMCalculatorService* classes wrap up the module and publish it as a Floodlight service.

*3) Meter:* As noted earlier, we use OpenFlow switch meter to limit bandwidth demand of the links. We wrap our meter related functions into a class *DropMeter*. It implements the creation of meters on switches and flow-meter binding functions. The first time a flow arrives at a particular switch, the switch either consults the controller or relies on its local look-up table for an instruction list regarding this flow. Besides forwarding instructions, Floodlight controller also creates a meter (with rate obtained from FDM service) and binds the flow with the meter on this switch.

## VI. Performance Evaluation

In this section, we present emulation results of traffic flow optimization in SDN-SAT. First we introduce the emulation setup, followed by description of test scenarios and emulation results. Performance of the proposed scheme is evaluated based on comparison among three different data delivery options: 1) single path TCP (without SDN), 2) MPTCP-SDN (without FDM optimization) and 3) MPTCP-SDN-FDM (with FDM optimization).

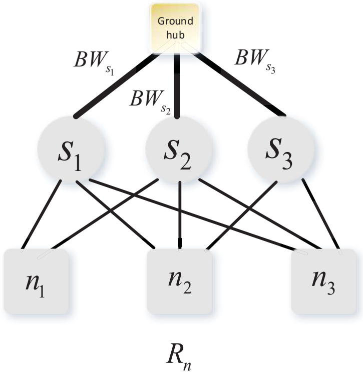


Fig. 2: Emulation topology for SDN-SAT system

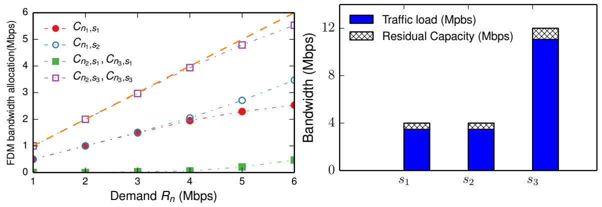
### *A. Emulation Setup*

The emulated SDN-SAT network topology is shown in Fig. 2. It consists of 3 ships and 3 SATCOM systems. This is the smallest network topology we can construct (without cluttering the graph $\mathcal{G}$ ) and still differentiate the outcomes of three data delivery options using pen-and-paper analysis.

The following parameters are carefully chosen to elucidate asymmetric features among nodes and SATCOMs: SATCOM capacities $BW_{s_1}$ = 4, $BW_{s_2}$ = 4, $BW_{s_3}$ = 12 (in Mbps); bandwidth demand of the ships $R_n$ are equal, stepped up from 1 to 6 (Mbps) to generate over-saturated scenarios per link, per node or per SATCOM system. Ship 1 ($n_1$) has no access to SATCOM 3 ($s_3$) while the other two have full SATCOM access. Traffic that arrives at a particular satelite on uplink channels are forwarded to the common ground hub before reaching its final destination.

If all ships use single path TCP (e.g. TCP Cubic [11]), each ship relies on a default SATCOM system for data delivery. In the best case they each choose a different SATCOM that may or may not support their respective demands. In the worst case, each ship is unaware of demand from other ships—and without the coordination of SDN controller—some ships may select the same SATCOM system, resulting in poor throughput and delay statistics. MPTCP allows each ship to use multiple subflows but it alone cannot resolve the bandwidth congestion problem since the underlying routing protocol uses single-path forwarding. With SDN and MPTCP-aware forwarding, every ship can obtain bandwidth from multiple sources (i.e. different SATCOM systems). This additional agility of MPTCP-SDN allows each TCP connection to traverse through multiple SATCOM links, thus improving its individual throughput and delay jitter as well as the overall system bandwidth utilization via global load balancing offered by the FDM algorithm.

### B. Emulation Results and Analysis



(a) $C_{2,2} = C_{3,2} = 0$

(b) Traffic load of SATCOMs

Fig. 3: FDM bandwidth allocation

When all ships transmit using single path TCP, $n_1$ gets at most 4. Ships $n_2$ and $n_3$ either share $s_3$ or one of them chooses $s_2$. To maximize bandwidth utilization, we assume that $n_2$ and $n_3$ share $s_3$. Note that this is already the best configuration we can set up under single path TCP. In contrast, MPTCP uses as many SATCOMs as possible ($n_1$ uses two, while $n_2$ and $n_3$ use all three). Since the total capacity is greater than the demand, ideally all the ships should be satisfied if they use MPTCP. Moreover, FDM balances the load at the SATCOMs by adjusting assigned bandwidths between SATCOMs and ships. The allocation given by FDM is plotted in Fig. 3a when $R_n$ increases from 1 to 6. Due to the symmetry of $n_2$ and $n_3$, their bandwidth allocations are the same. Moreover, $s_2$ and $s_3$ are also symmetric. Therefore their total traffic loads are equal, i.e. $C_{n_1,s_1} + C_{n_2,s_1} + C_{n_3,s_1} = C_{n_1,s_2}$. Fig. 3b shows the total traffic load of the three SATCOMs when $R_n$ = 6. Indeed, the load is well balanced by FDM as every SATCOM as has residual bandwidth.

The average goodput (and its standard deviation) of three different schemes are shown in Fig. 4. In Fig. 4a, the goodput of $n_2$ and $n_3$ increases linearly with their demand as they share $s_3$ which has abundant capacity. The goodput of $n_1$ suffers since its transmission rate is limited by the capacity of $s_1$. In Fig. 4b and 4c, all ships have similar goodput performance as they employ MPTCP to share the SATCOMs. The difference is that, as the demand exceeds the saturation point of 4, goodput in Fig. 4b becomes unstable. Without FDM intervention, the three MPTCP sessions try to optimize their packet scheduling independently. This results in greedy local traffic optimization which eventually saturates $s_1$ and $s_2$. The consequence is that subflows passing through $s_1$ and $s_2$ slow down the entire transmission. This is known as the head-of-line blocking issue of MPTCP [18]. On the other hand, FDM globally optimizes bandwidth allocation and restricts the rate of subflows that access $s_1$ and $s_2$.

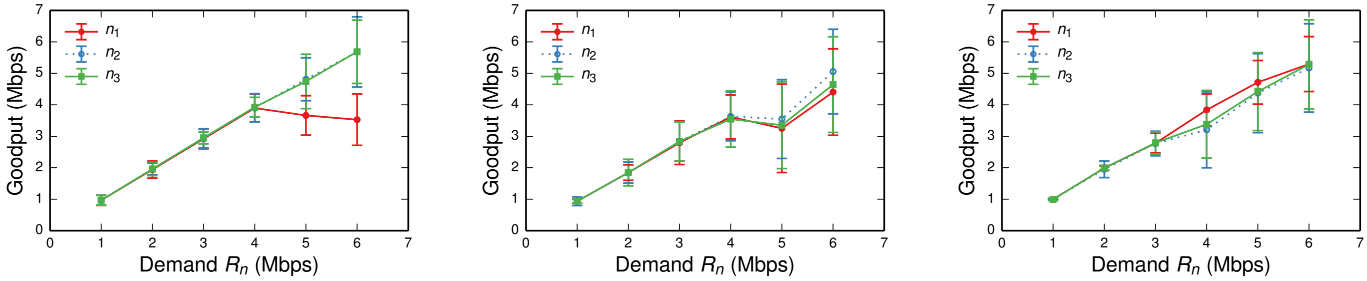
We also present the jitter of Round Trip Time (RTT) per ship under different schemes in Fig. 5. We only show the results of case $R_n$ = 6. Using single path TCP, all the SATCOMs are saturated and delay jitter is large. Using MPTCP without FDM, the delay jitter is almost equally large as the MPTCP scheduler greedily injects packets whenever the buffer of a subflow is not full. Therefore, $s_1$ and $s_2$ are saturated and packets that access these SATCOMs have large RTT jitter. Particularly for $n_2$ and $n_3$, the slow subflows that access $s_1$ and $s_2$ also drag down the third subflow that accesses $s_3$ as a result of the head-of-line blocking issue. Lastly, with the help of FDM traffic optimization, the RTT of MPTCP is small and smooth, as traffic load is well balanced and none of the SATCOMs are saturated.

As a final note, in our previous paper we have shown that another advantage of MPTCP compared to single path TCP is that it improves system robustness. In the scenario of Fig. 2, suppose $n_2$ and $n_3$ are sharing $s_3$ to maximize their throughputs. If $s_3$ is out due to jamming or intermittence, both ships will lose connection and have to be rerouted to other channels. In contrast, MPTCP maintains multiple concurrent connection to the SATCOMs and is able to smoothly shift traffic to other channels when $s_3$ fails.

## VII. Conclusion and Future Work

The paper is our follow-on to the application of SDN framework to multi-SATCOM network architecture such as that of a naval surface fleet network. Unlike the conventional policy-based routing procedure, we treat the multi-SATCOM link utilization problem as a Multi-Commodity Flow optimization problem and propose a partciular derivative solution known as the Flow Deviation Method. Additionally, we propose using Multipath TCP in combination with SDN and FDM since it offers unique benefits (higher throughput, lower delay jitter and robustness against link failure) compared to its single-path counterpart.

Two related topics absent in this paper are convergence time of FDM and control overhead (in terms of SATCOM bandwidth) exchanged between SDN switches and their controller. We are currently developing large-node SDN emula-

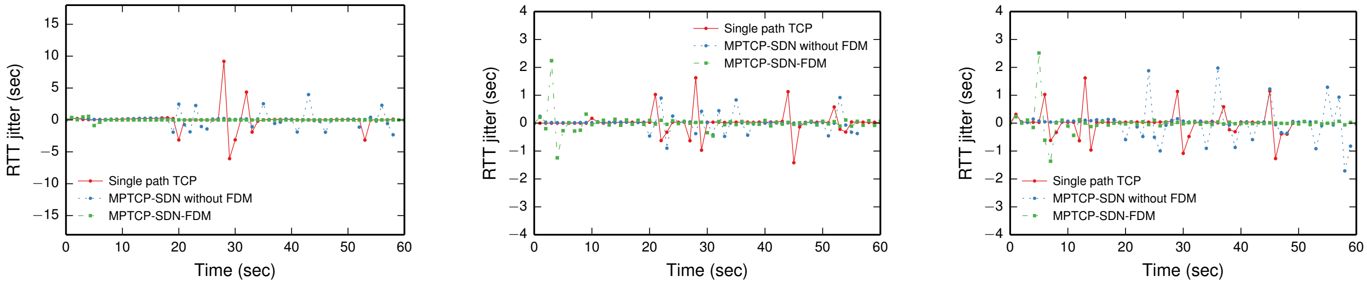


(a) Non-SDN single path TCP

(b) MPTCP-SDN without FDM

(c) MPTCP-SDN with FDM

Fig. 4: Goodput performance of three different TCP connection options



(a) $n_1$

(b) $n_2$

(c) $n_3$

Fig. 5: Round Trip Time (RTT) performance of TCP connections

tion testbed and quantitative measures of these parameters will be addressed in our future work.

## Acknowledgments

This work was supported by the Office of Naval Research (ONR) Contract N00014-15-C-5038 Software Defined Naval Battlefield Network (SD-NBN). The authors would like to thank Dr. Santanu Das (ONR Program Manager) for his support and guidance, and Zheng Yang (UCLA Summer Research Intern) for helping with the implementation.

## References

[1] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou. Research challenges for traffic engineering in software defined networks. *IEEE Network*, 30(3):52–58, 2016.

[2] T. Braun, P. Du, and M. Gerla. Integration of multipath tcp and network coding for satellite networks. 2017.

[3] J. Cloud, F. du Pin Calmon, W. Zeng, G. Pau, L. M. Zeger, and M. Medard. Multi-path tcp with network coding for mobile devices in heterogeneous networks. In *Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th*, pages 1–5. IEEE, 2013.

[4] R. L. S. de Oliveira, A. A. Shinoda, C. M. Schweitzer, and L. Rodrigues Prete. Using mininet for emulation and prototyping software-defined networks. In *Communications and Computing (COLCOM), 2014 IEEE Colombian Conference on*, pages 1–6. IEEE, 2014.

[5] P. Du, X. Li, Y. Lu, and M. Gerla. Multipath tcp over leo satellite networks. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2015 International*, pages 1–6. IEEE, 2015.

[6] P. Du, S. Nazari, J. Mena, R. Fan, M. Gerla, and R. Gupta. Multipath tcp in sdn-enabled leo satellite networks. In *Military Communications Conference, MILCOM 2016-2016 IEEE*, pages 354–359. IEEE, 2016.

[7] K. Fiscko. Shipboard networks. *PEO C4I, US Navy, SPAWAR*, 2008.

[8] 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.

[9] 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.

[10] L. Fratta, M. Gerla, and L. Kleinrock. The flow deviation method: An approach to store-and-forward communication network design. *Networks*, 3(2):97–133, 1973.

[11] 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.

[12] A. Kershenbaum. *Telecommunications network design algorithms*. McGraw-Hill, Inc., 1993.

[13] W. Kim and M. Gerla. Navopt: Navigator assisted vehicular route optimizer. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on*, pages 450–455. IEEE, 2011.

[14] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76, 2015.

[15] X. Liu, S. Mohanraj, M. Pióro, and D. Medhi. Multipath routing from a traffic engineering perspective: How beneficial is it? In *Network Protocols (ICNP), 2014 IEEE 22nd International Conference on*, pages 143–154. IEEE, 2014.

[16] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.

[17] S. Nazari, P. Du, M. Gerla, C. Hoffmann, J. H. Kim, and A. Capone. Software defined naval network for satellite communications (sdn-sat). In *Military Communications Conference, MILCOM 2016-2016 IEEE*, pages 360–366. IEEE, 2016.

[18] C. Paasch, S. Ferlin, O. Alay, and O. Bonaventure. Experimental evaluation of multipath tcp schedulers. In *Proceedings of the 2014 ACM SIGCOMM workshop on Capacity sharing workshop*, pages 27–32. ACM, 2014.

[19] S. Paris, A. Destounis, L. Maggi, G. S. Paschos, and J. Leguay. Controlling flow reconfigurations in sdn. In *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, pages 1–9. IEEE, 2016.

[20] M. Pióro and D. Medhi. *Routing, flow, and capacity design in communication and computer networks*. Elsevier, 2004.

[21] W. Szeto, Y. Iraqi, and R. Boutaba. A multi-commodity flow based approach to virtual network resource allocation. In *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE*, volume 6, pages 3004–3008. IEEE, 2003.

[22] L. Xiao, M. Johansson, and S. P. Boyd. Simultaneous routing and resource allocation via dual decomposition. *IEEE Transactions on Communications*, 52(7):1136–1144, 2004.