Software Defined Multi-Path TCP Solution for Mobile Wireless Tactical Networks

Qi Zhao, Pengyuan Du, Mario Gerla, Adam Brown, Jae Kim
Department of Computer Science, UCLA
Boeing Research & Technology, Seattle

10/31/2018
Outline

- Introduction
- Background
- Solution Design
- Evaluation
- Conclusion
Introduction

- Naval Battlefield Network (NBN)
  - Shipboard satellite communication
  - Multi-path TCP & Software Defined Networking
  - Bandwidth sharing and load balancing
Introduction

- Modern NBN network
  - Naval entity: Ship, Soldier, Aircraft...
  - Communication media: Satellite, UAV
  - Static --> Dynamic
Introduction

▪ Does the old solution still work?
▪ Answer: No, because of:
  ▪ Node mobility
    ▪ Dynamic link connection
    ▪ Dynamic traffic flow allocation
  ▪ SATCOM / UAV links
    ▪ Link capacity: large / small
    ▪ Link latency: high / low
    ▪ Signal range: wide / narrow
Outline

- Introduction
- Background
- Solution Design
- Evaluation
- Conclusion
Background

- Multipath TCP (MPTCP)[1]
  - Presenting a **single** TCP connection to the application
  - Utilize different interfaces underneath
  - Work over today’s networks

Background

- **Software Defined Networking**
  - SDN controller manages sub-flows **globally**
Outline

- Introduction
- Background
- Solution Design
- Evaluation
- Conclusion
Problem analysis

- Mobile naval network scenarios
  - Ship to Ship
  - Ship to Shore

- Data transmission must not be interrupted: Smooth network handover and reliable communication
- Traffic flow allocation must be able to reconfigure: Real-time traffic engineering and network configuration
Proposed solution

- Multi-path TCP
  - Smoother reaction to network changes
    - Immediate utilization of available links
    - Low overhead and no interruption to existing sessions
- Software defined networking
  - Controller defined by our own
    - Real-time traffic flow calculation and configuration
    - Avoid congestion due to MPTCP’s greedy scheduler
System Architecture

SDN-Controller ➔ FDM ➔ Calculating flow allocation ➔ Stats Collecting ➔ Alloc deploying ➔ User movement
SDN Controller with FDM module

- Traffic engineering in SDN can be formulated as an **Multi-Commodity Flow problem**[1]
- Solve with the solution to the “Routing Assignment” problem in the **Flow Deviation Method**[2]
- **Objective**: minimize total packet delay while satisfying both capacity and bandwidth demand constraints.

---


Outline

- Introduction
- Background
- Solution Design
- Evaluation
- Conclusion
Mininet-WiFi-based Emulation Testbed

- **Process-based nodes**
  - Linux kernel implementation
  - MPTCP on sender & receiver
- **Traffic control link**
  - Enable link capacity and delay configuration
- **Node mobility is supported**
- **Self-implemented SDN controller and FDM module**
- **Flow table:**
  - Decides routing
  - OVS queues to restrict bandwidth
- **Traffic generator:**
  - iPerf3 (custom rate)
- **Capture packets with Wireshark**
More Details

- Testbed platform:
  - Linux Ubuntu 14.04 with 8GB RAM
  - MPTCP v0.92 and Open vSwitch installed
- Experiment with 3 different protocols for every scenario to evaluate the performance of our proposed solution
  - Single-path TCP (SPTCP) – baseline
  - Multi-path TCP without FDM (MPTCP)
  - Multi-path TCP with FDM (FDM)
Evaluation Scenario I

- Direct move experiment
  - 2 Mobile users & 1 host
    - 3Mbps sending rate
  - 1 OVS switch – SATCOM
    - 250ms delay
    - 50Mbps bandwidth
  - 1 OVS AP – UAV
    - 10ms delay
    - 1Mbps bandwidth
  - 100s total execution time
  - 2 users enters UAV’s range at 60s
Experiment Results I

- ~4 seconds communication interruption caused by network handover in SPTCP case
- Average throughput
  - 0.4875Mbps – SPTCP
  - 0.3728Mbps – MPTCP
  - 0.4188Mbps – FDM
Results Summary I

- **MPTCP vs SPTCP**
  - Reliable continuous communication is guaranteed by MPTCP protocol
  - SPTCP’s overall throughput is slightly higher due to infrequent network handover

- **MPTCP only vs MPTCP with FDM**
  - FDM’s overall throughput and throughput variation is better
  - FDM’s optimizer allocates bandwidth more efficiently than greedy heuristics of MPTCP’s default scheduler
Evaluation Scenario II

- Random walk experiment
  - 2 Mobile users & 1 host
    - 3Mbps sending rate
  - 1 OVS switch – SATCOM
    - 250ms delay
    - 50Mbps bandwidth
  - 1 OVS AP – UAV
    - 10ms delay
    - 1Mbps bandwidth
  - 100s total execution time
  - 2 users randomly move
Experiment Results II

- Multiple communication interruptions caused by network handover in SPTCP case
- Average throughput
  - 0.3121Mbps – SPTCP
  - 0.4738Mbps – MPTCP
  - 0.4602Mbps – FDM
Results Summary II

- **MPTCP vs SPTCP**
  - As expected, SPTCP’s overall throughput is degraded comparing to scenario I and MPTCP case

- **MPTCP only vs MPTCP with FDM**
  - FDM’s overall throughput is slightly worse presumably due to the frequency of the network handover
  - FDM’s throughput variation is much better because of the fairly allocated bandwidth of FDM
Outline

- Introduction
- Background
- Solution Design
- Evaluation
- Conclusion
Conclusion

- Supporting dynamic bandwidth allocation in real time
- Handling the mobility management of heterogeneous naval networks for both sparse and dense network handover cases
- In terms of overall throughput, dense network handover outperforms sparse network handover
- In terms of bandwidth fairness, FDM outperforms all non-FDM cases
Contributions

- A dynamic SDN controller to allocate traffic flows in mobile wireless tactical networks
  - FDM-based flow allocation module
  - Support dynamic flow adjustment
  - Support multi-scenario, e.g., sparse and dense handover

- A complete MPTCP-enabled Mininet-WiFi-based emulation testbed integrated with our dynamic SDN controller
Thank you!

Q&A