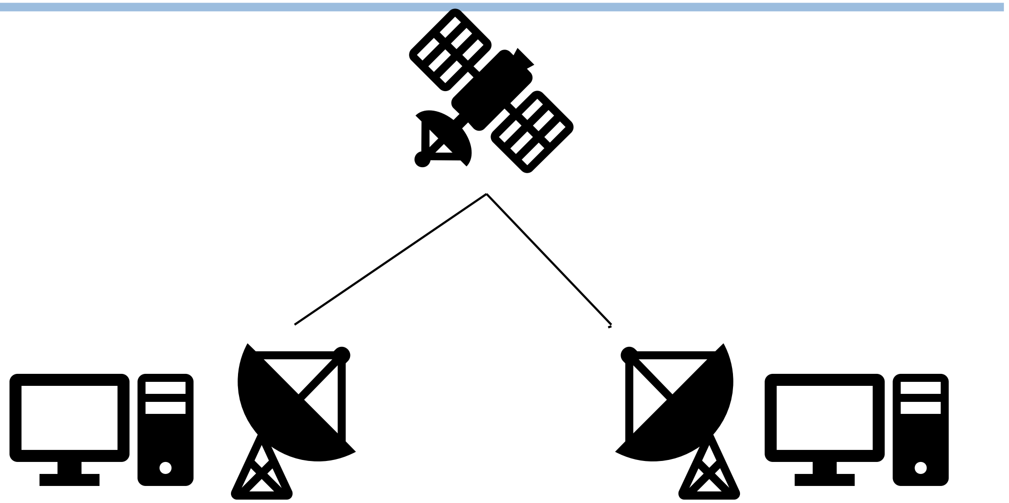
CDS Seminar
Measurements of MP-TCP protocols in Satellite Communications

Dr. Eryk Schiller

Universität Bern

25.09.2017

Introduction to Satellite Communications



- 1. Different Orbits:
  - International Space Station (ISS) 408 km
  - Iridium Satellite Constellation 781 km
  - Geostationary satellites 35786 km
- Satellite transmission is between 1-40 GHz
  - Satellite communication systems are in the 30/20 GHz spectrum range
  - Broadband communication systems
- Geostationary communications is suspected to long round-trip-delay
  - The speed of light is ~300’000 km/s
  - The round-trip delay is of around 0.4 sec.

Impact of Geostationary Satellite Communications on the transport protocols.

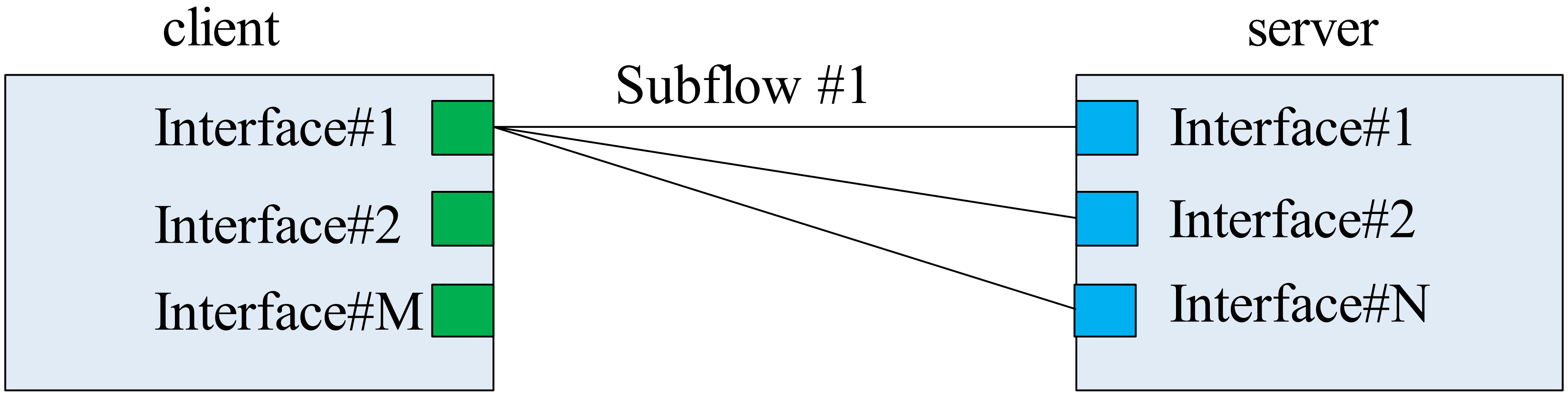
- Transmission Control Protocol (TCP) operates with the so called transmission window
  - the number of sent, but unacknowledged segments.
- Satellite communications is characterized by
  - High bandwidth (B) [bps]
  - Long Round-Trip-Delay (RTT) [s]
- The theoretical window size in satellite communications is high
  - WINDOW [MSS] = B * RTT //Maximum Segment Size
- There is a large number of outgoing not acknowledged packets.

The problem of large window in satellite commnications

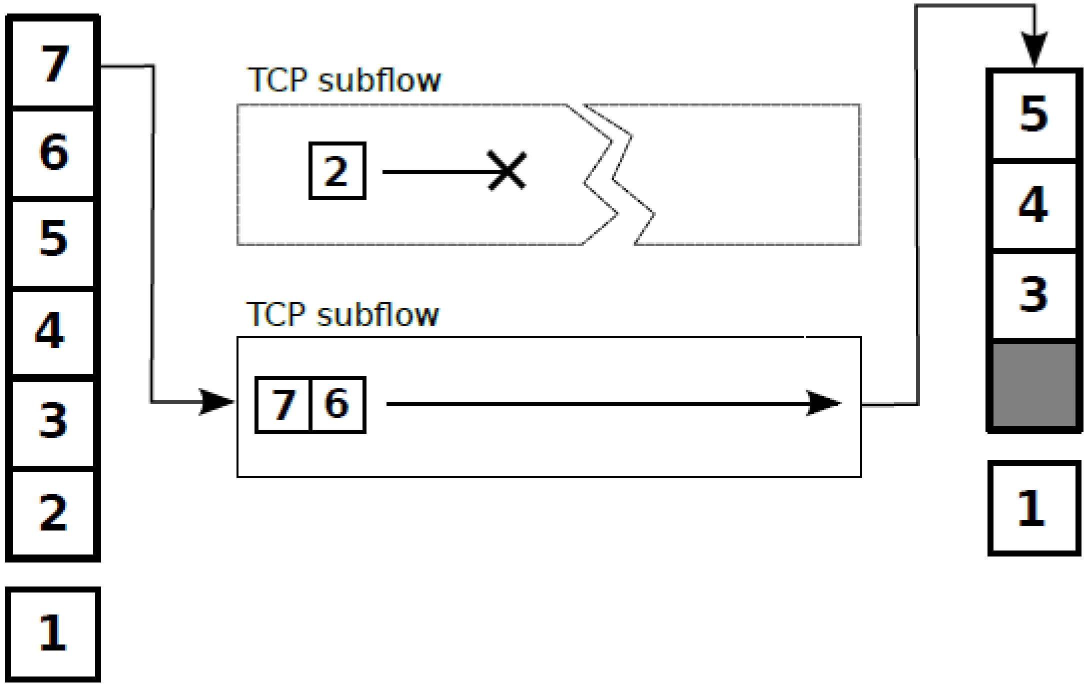
> Many different TCP variants provided in the past
  - New Reno [1999]
  - Vegas [2000]
  - Westwood [2003]
  - BIC
  - CUBIC [2007]
  - H-TCP [2004]
  - Hybla [2004]
  - TCP/NC [2011] introducing network coding to TCP connections

Multi-Path TCP [2013]

- TCP could benefit from many interfaces possessed by the host.
  - the client has M interfaces
  - the server has N interfaces,
- We establish M x N simultaneous sub-flows between server & client.
- TCP can saturate a few links at the same time receiving better overall throughput.



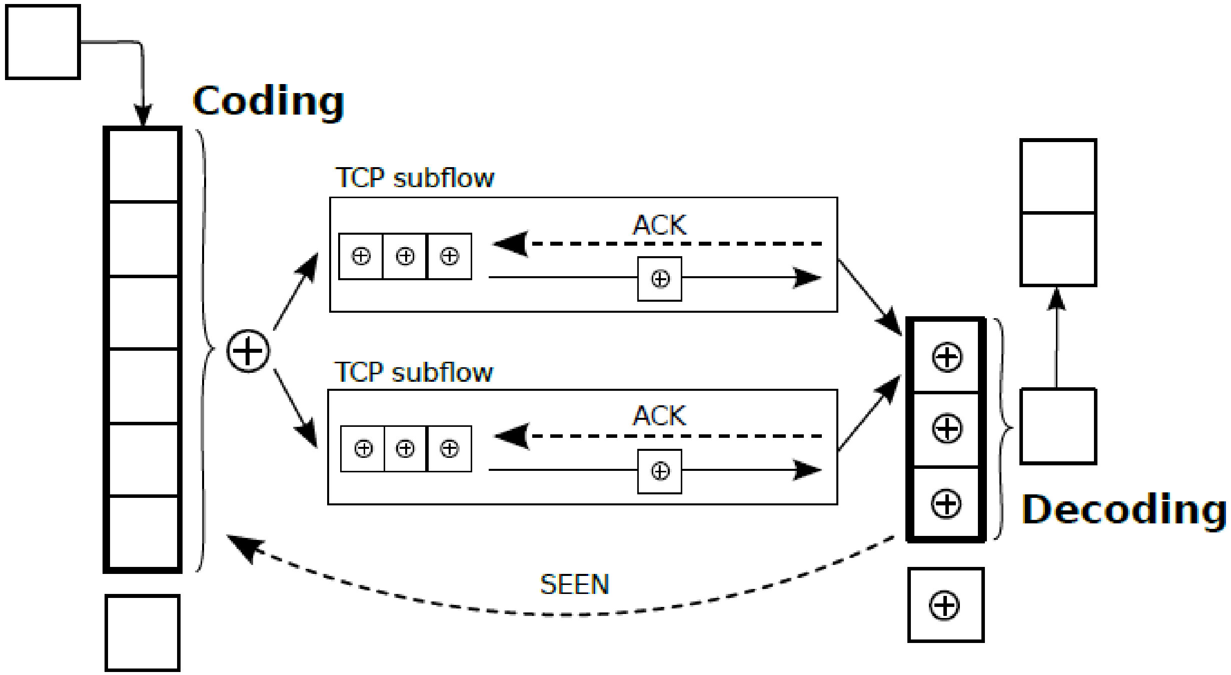
Blocking in the case of link losses/congestion



- TCP losses are blocking the global window.
- A link loss in one subflow, can freeze the communication in the entire MP-TCP flow, while the missing packet is not received by the receiver.

Figure from L. Ageneau, N. Boukhatem and M. Gerla, "Practical random linear coding for MultiPath TCP: MPC-TCP," *2017 24th International Conference on Telecommunications (ICT)*, Limassol, 2017, pp. 1-6. doi: 10.1109/ICT.2017.7998267

Use of network coding to decipher packets at the receiver: MPC-TCP (MultiPath Coded TCP)

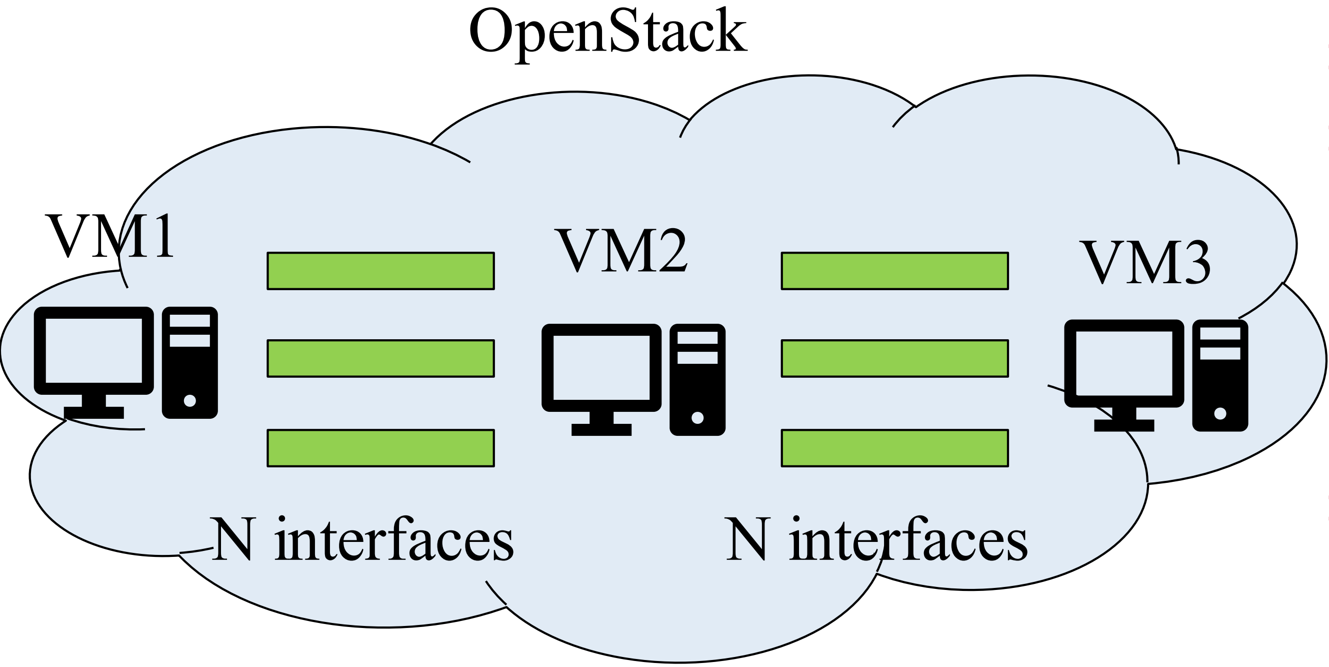


Use of the TCP/NC on every sub-flow.

Every packet on the sub-flow carries out new information and allows us to decode all sub-streams even if there is an error on a sub-flow.

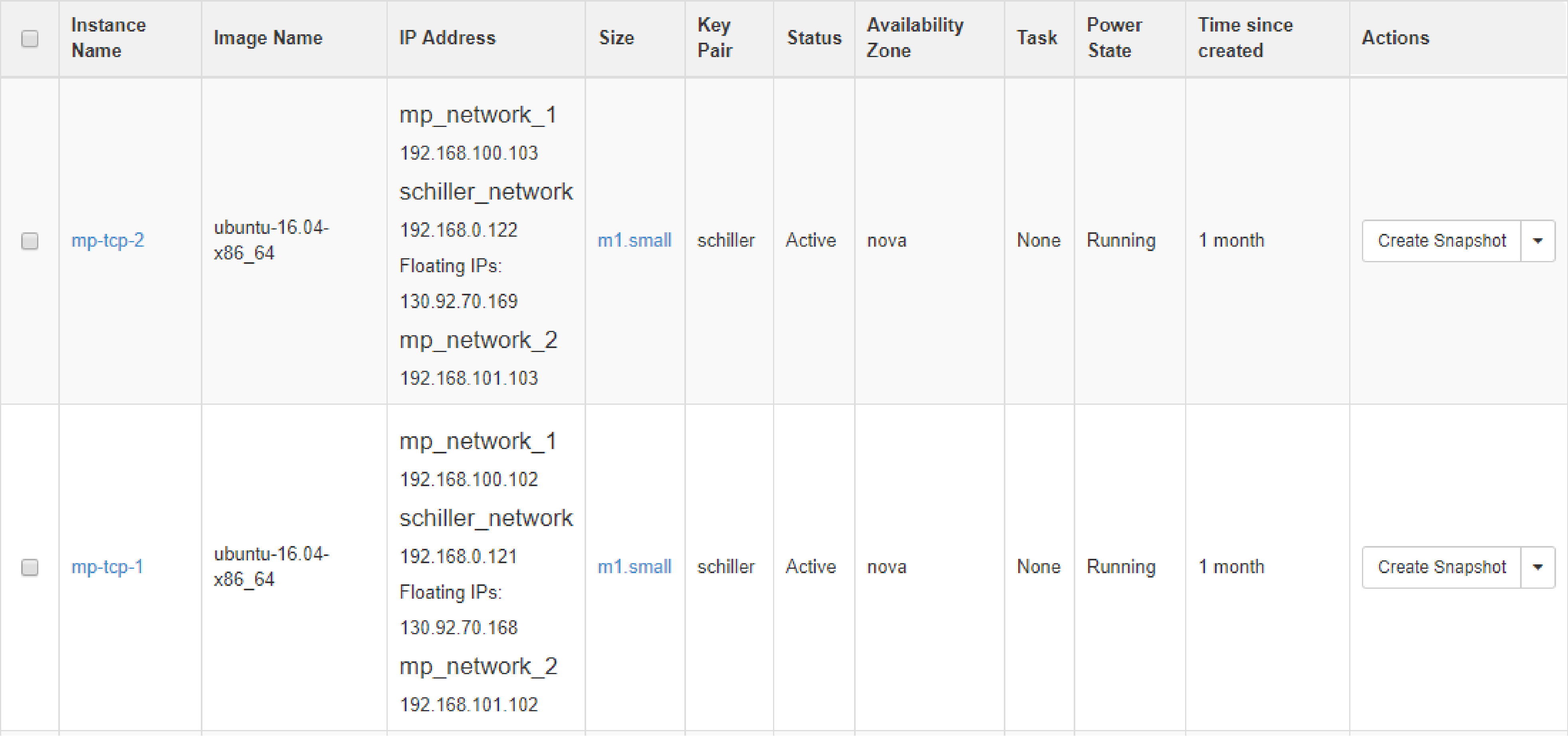
Figure from L. Ageneau, N. Boukhatem and M. Gerla, "Practical random linear coding for MultiPath TCP: MPC-TCP," *2017 24th International Conference on Telecommunications (ICT)*, Limassol, 2017, pp. 1-6. doi: 10.1109/ICT.2017.7998267

Architecture of the solution



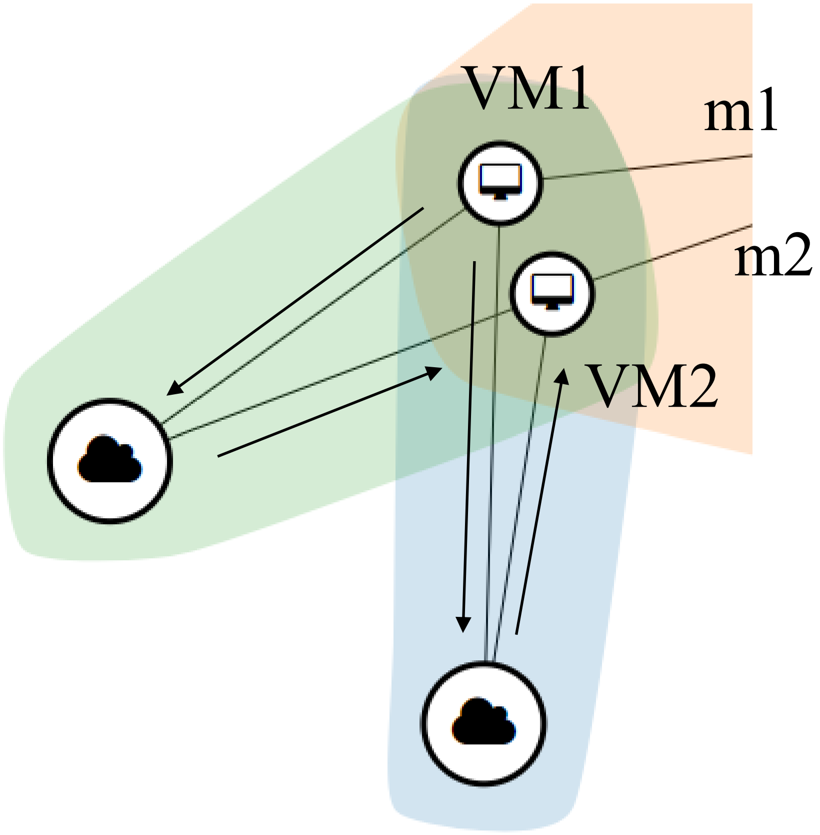
- VM1 – sender
- VM2 – emulator of the satellite connection
  - Configurable delay
  - Configurable link throughput
  - Configurable error model ON/OFF
- VM3 - receiver

The VM Configuration



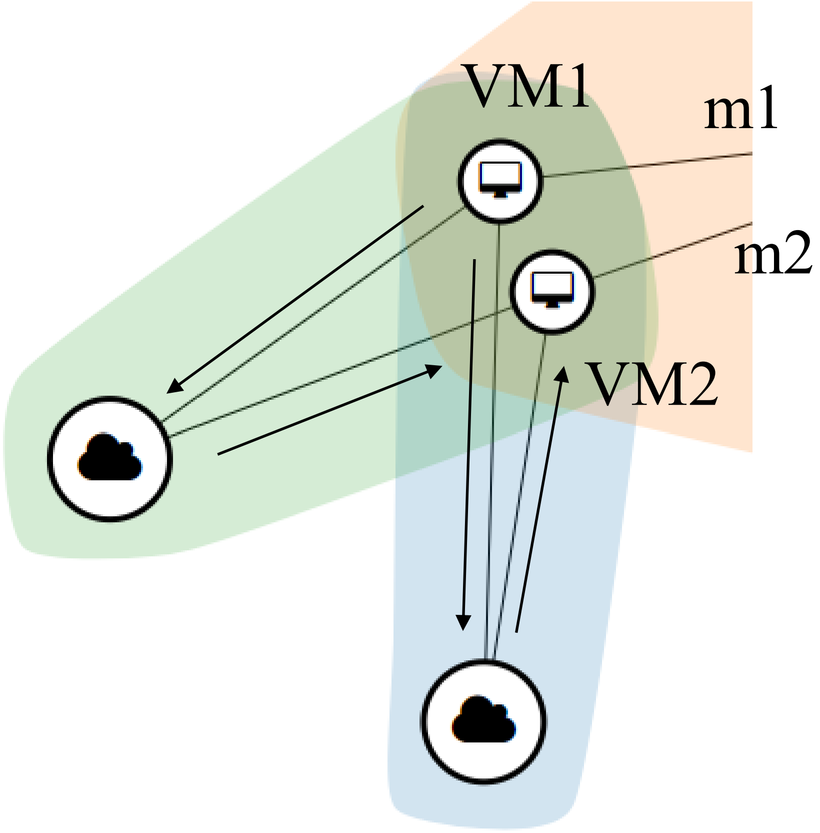
□	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
□	mp-tcp-2	ubuntu-16.04-x86_64	mp_network_1 192.168.100.103 schiller_network 192.168.0.122 Floating IPs: 130.92.70.169 mp_network_2 192.168.101.103	m1.small	schiller	Active	nova	None	Running	1 month	Create Snapshot ▾
□	mp-tcp-1	ubuntu-16.04-x86_64	mp_network_1 192.168.100.102 schiller_network 192.168.0.121 Floating IPs: 130.92.70.168 mp_network_2 192.168.101.102	m1.small	schiller	Active	nova	None	Running	1 month	Create Snapshot ▾

Current Software Implementation



- VM1, VM2 – typical Ubuntu based VMs
- Applied kernel patches for MP-TCP
  - http://multipath-tcp.org/patches/mptcp-v4.4-da46ec3d7b6e.patch
  - Used Linux 4.4
- To start MP-TCP connection, I used mesh mode.
- The links do not overlap, therefore only two simultaneous sub-flows are possible

Modifications



- VM1, VM2 – typical Ubuntu based VMs
- I am currently integrating the
  - https://github.com/paullouisageneau/mptcp/tree/mptcp_rlc MPTCP-RLC implementation
  - It is provided as a patch to Linux Kernel 3.14.16+.
  - Currently, I resolve the problems with starting Ubuntu Xenial with this kernel version.

Start measurements

- mp-tcp-1# iperf -s
- mp-tcp-2# iperf -c 192.168.100.102
- tcpdump -i ens4

  ```
  13:27:08.904263 IP 192.168.100.103.38366 > 192.168.100.102.5001: Flags [.],
  seq 45110721:45112149, ack 0, win 457, options [nop,nop,TS val 703781 ecr
  702039,mptcp dss ack 2181998009 seq 3117938866 subseq 394384929 len 1428
  csum 0xe597], length 1428
  ```

-

  tcpdump -i ens5

  ```
  13:28:05.335999 IP 192.168.101.103.60586 > 192.168.101.102.5001: Flags [.],
  seq 29754884:29756312, ack 1, win 457, options [nop,nop,TS val 717889 ecr
  716148,mptcp dss ack 1546215783 seq 1584746838 subseq 158552413 len 1428
  csum 0xde8b], length 1428
  ```

Future Works

- Finish with the MPTCP-RLC kernel module
  - Probably having two images:
    - Ubuntu 14.04 for kernel 3.14 (old MPTCP-RLC implementation)
    - Ubuntu 16.04 for kernel 4.4 (newer implementation of the MP-TCP).
- Implement the VM2 to
  - Delay communication by a configurable parameter
  - Limit the throughput on the link, currently the link speed is limited by the OVS switch of OpenStack
  - Introduce the ON/OFF error model.
- Deliver images & configuration scripts for images.
- Provide the measurements of MP-TCP with and without network coding as a function of delay and error model.

Questions

- Q/A