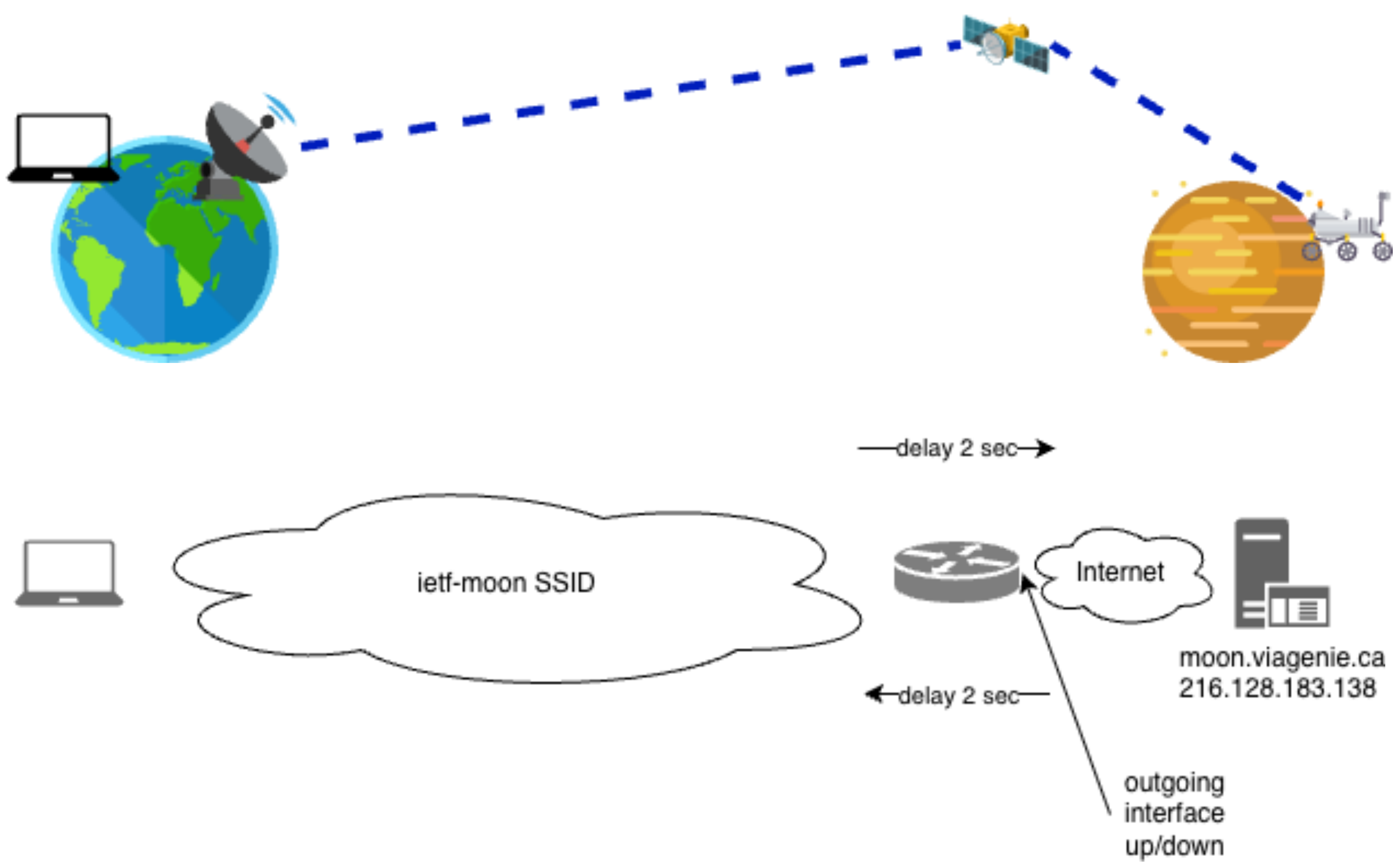
IETF124 Deep Space/TIPTOP Hackathon Experiment Report

Marc Blanchet, marc.blanchet@viagenie.ca, Montréal, 2025-11-06

Setup

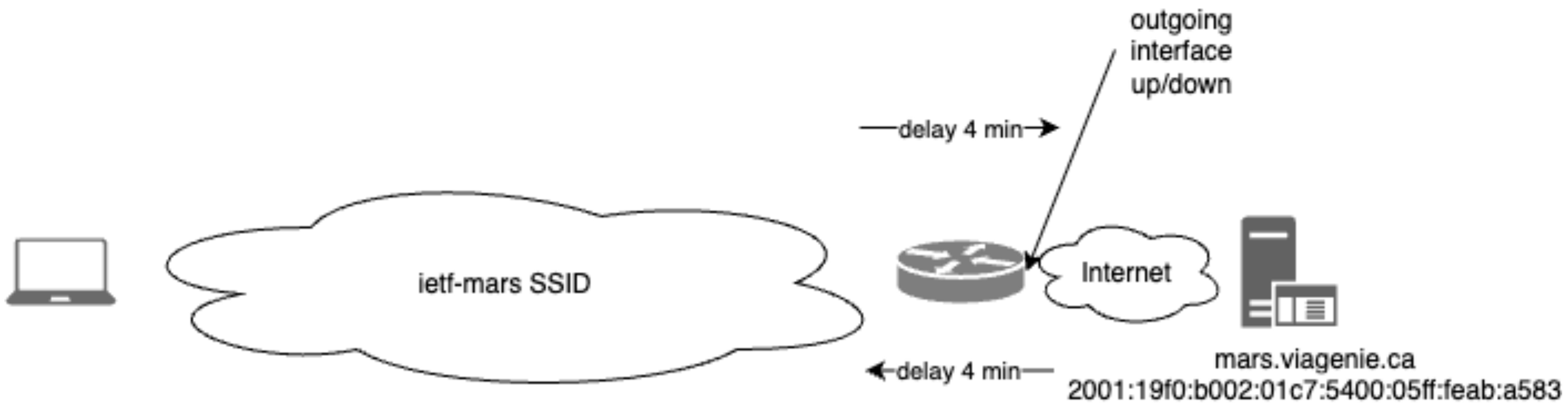
- Simulates Earth to {Moon,Mars} through an orbiter
- Client laptop on Earth
- Orbiter is a minipc running Ubuntu acting as a forwarder:
  - Earth side:
    - Wifi AP
    - Provision IP address space to clients
  - Celestial body side:
    - Upstream interface: Ethernet connected to the IETF network
    - IP forwarding
  - Delay injection (2 sec each way) using netem
  - Storing IP packets when interruptions
  - Interruptions using scheduled interface up and down
- Rover on Moon/Mars is cloud VM on Internet
- Clients tested profiled QUIC and out of the box QUIC: initial_rtt, max_idle_timeout and « simple » CC.

Setup - Moon



- IPv4-only, 2 seconds delay each way
- Laptop clients connect to specific wifi (ietf-moon)
- Forwarder is IPv4-only
    - DHCPv4
    - NAT44
- Moon rover is a cloud VM on Internet.

Setup - Mars



- IPv6-only, 4 minutes delay each way (nearest point between Earth and Mars)
- Laptop clients connect to specific wifi (ietf-mars)
- Forwarder is IPv6-only
  - DHCPv6-PD client on upstream side (IEEE network
  - RA on wifi
- Mars rover is a cloud VM on Internet.
- No DNS (for now): so typing « long » IPv6 addresses without typos…

IPv4 vs IPv6

- Initially planned for IPv4-only, to keep things simple
- But for Mars case, did not work well, because of multiple UDP timeouts in the path on the Internet.
- So during the first day, changed the Mars setup to be IPv6 only.
    - Linux IPv6 forwarder with the right tools (DHCPv6-PD, RA, …) configured correctly took some time.
- End state:
    - Moon: IPv4-only
    - Mars: IPv6-only
- Restricted to one IP version per network, just to avoid also testing happy eyeballs or dual-stack interactions. Again to keep things simple. Will test dual-stack in a next run.
- Good side effect: demonstrated both IP version running in the setup

Protocols/Applications Tested

- Moon and Mars:
  - Ping with timeout set (-W)
    - set value had different units on different OS…: ms or sec
  - http get over profiled QUIC: Quinn and Quiche stacks
  - SNMP with timeout set (-t): Net-SNMP

Protocols/Applications Tested: Moon

- Moon (apps: no profile, no change, as is):
  - Various messaging: Discord (the informal group channel for deepspace): no change, no impact seen (except the obvious delay), Apple Messages, Slack, … worked.
  - Various HTTP services, browser client (all seemed to use QUIC transport): worked
  - Youtube: worked as is (except the obvious delay, and some buffering happening at the beginning: so start of the stream on the screen was ~15 sec)
  - Mail (IMAPS): not fetching. TCP, as is, not that good
  - Ssh. Kinda work. Better if set in line mode (instead of real-time echo). More later

- Early (dumb) conclusion: Moon delay is not a big issue, many applications work as is with no changes.
- Caveats:
  - haven't investigated much on why one or the other worked or does not work out of the box. Just verified that http services that worked much better were all using QUIC.
  - These tests were not prepared really. Just found it worked. So no plan was defined for testing protocols, …

Hackathon is useful

- A controlled testbed (as we have with 10 VMs on virtual links) is not the same as real network with real users…
- Scaling
- Different setup
- New issues introduced…

- Many things that worked in your environment did not when put on another network…; but all those issues are unrelated to the actual test one want to do.

Not related to tiptop specs, but setup

- Netem delay uses socket memory space. Given trafic and temporary memory needed, had to significantly increase it.
  - Next time, might look into introducing delay in the tun user space interface implementation that also handles intermittence
- To handle intermittence, interface up/down was used, which is disruptive (upstream LAN disconnect, dhcp request, need to reinstall routes, …)
  - Still works
  - Interface up/down might not be what really will happen in space: interface (from the IP perspective) may remain up while there is intermittence
  - Next time, might look into other ways (ethernet driver, ipfilter, tc, …)
- Units… sometimes ms, sec, … even the same tool: ping -W on linux uses ms, while MacOSX uses sec… very easy to mix up in a combined environment.

Not related to tiptop specs, but setup

- Injecting 4 minutes delay on outgoing interfaces (both to Moon/Mars and to Earth), on all traffic means that dhcp requests and other low-level (arp, nd) are delayed too.
  - Not a problem on a normal setup (nor on our VM based testbed) since the delay is not the connecting LAN, but is an issue for this setup as the orbiter was also the AP and was giving IP addresses to the WIFI side.
  - Not a problem for real space deployment, as obviously the orbiter will not deliver IP address and be the next hop of Earth nodes…
  - Fix: in tc netem, do not apply delay for IPv4: ARP, DHCP; IPv6: ND, DHCPv6 . A little more complicated tc setup, more complicated to debug live…

Chasing UDP Timeouts

- A cloud VM was used as Moon and Mars rovers. Moon worked nice. Mars did not initially. UDP timeout somewhere (on the host of the VM or intermediate nodes on cloud provider network) made anything more than 20 seconds not working.
- Issues running QUIC IPv4 client for Mars on MacOSX (Tahoe) for long delays(Mars case): seems to have an internal UDP timeout (did not find a way to change): symptom: source port got unreachable in the middle of the connection. To be investigated. Might even not be in MacOSX. Will try to reproduce.
- In general: IPv4 deployment on Internet introduces state in the network in many places, often unknown to user/app/… A state typically mean to have a timeout, which will be too low, to avoid keeping too much state at high scale. Avoid network state: NAT, conntrack, …. .
  - See recent email of Lorenzo Coletti on DNSOP for this exact symptom and some data about UDP timeouts on Internet
  - Learning useful to add to the drafts?

Keep-Alives

- ssh: (over tcp): for Moon, not great, but kinda works. Mars: forget it.
  - Side note: there was an initial proposal for SSH3 (over QUIC) but work seems not progressing.
- However, need to disable sshd keepalives so that connection is not terminated too soon on the server side.
  - Should we add some generic (about application/protocols, not specific to ssh) text in the drafts about keepalives?

Next Steps

- Intend to repeat at next IETFs
- More protocol/application testing; more scenarios
- Thinking of having a complete autonomous network, to avoid various UDP timeouts of Internet intermediate nodes, for long delays testing
- Please provide input for what to do/test

- Thanks to all participants. Very much appreciated.

- Questions?
- marc.blanchet@viagenie.ca