The Auckland Satellite TCP Simulator Facility

Ulrich Speidel

u.speidel@auckland.ac.nz

Wayne Reiher
PhD candidate

Background

Internet connectivity in many Pacific Island states relies on satellite – mostly GEO and MEO still, but also some Starlink LEO now

Background

More and more islands get cable – widening the gap for satellite-connected islands

Background

MEO and GEO: Low and expensive bandwidth, long RTT – a challenge for transport protocols like TCP

Background

Satellite Internet is still poorly understood – why?

So far: Studying real geostationary (GEO) and medium earth orbit (MEO) links

- Options:
  - Use dedicated experimental link
    - Pro: controlled environment
    - Cons: Link capacity is expensive (US$200+/Mb/s/month)
  - Use a production link
    - Pro: cheaper
    - Cons: uncontrolled environment, users of (usually narrowband) production links don't like active measurement traffic competing with theirs
- Neither option works well. Also:
  - Some aspects can be provider- or hardware-specific
  - Can't always get access to both ends of a link

Future work: Studying real low earth orbit (LEO) links

- Only one provider (Starlink)
    - Pros:
        - Comparatively cheap
        - Higher minimum bandwidth
        - Easy setup
    - Cons:
        - Opaque system
        - Geographical location matters
        - System today differs from tomorrow
        - Uncontrolled and hard-to-observe environment
        - Bandwidth and latency vary enormously
        - Access to one end of the link only
        - Cannot pick link parameters

Simulation in software

- Large number of simulators available
    - ns-2, ns-3, GNS3, Opnet, Mininet, …
- Slow - not real-time
- Low degree of parallelism
- Simulating large networks is difficult
- Representing the complexity of hardware components with offloads etc. is difficult
- Configuration of realistic traffic profiles (mix of different flows) is challenging

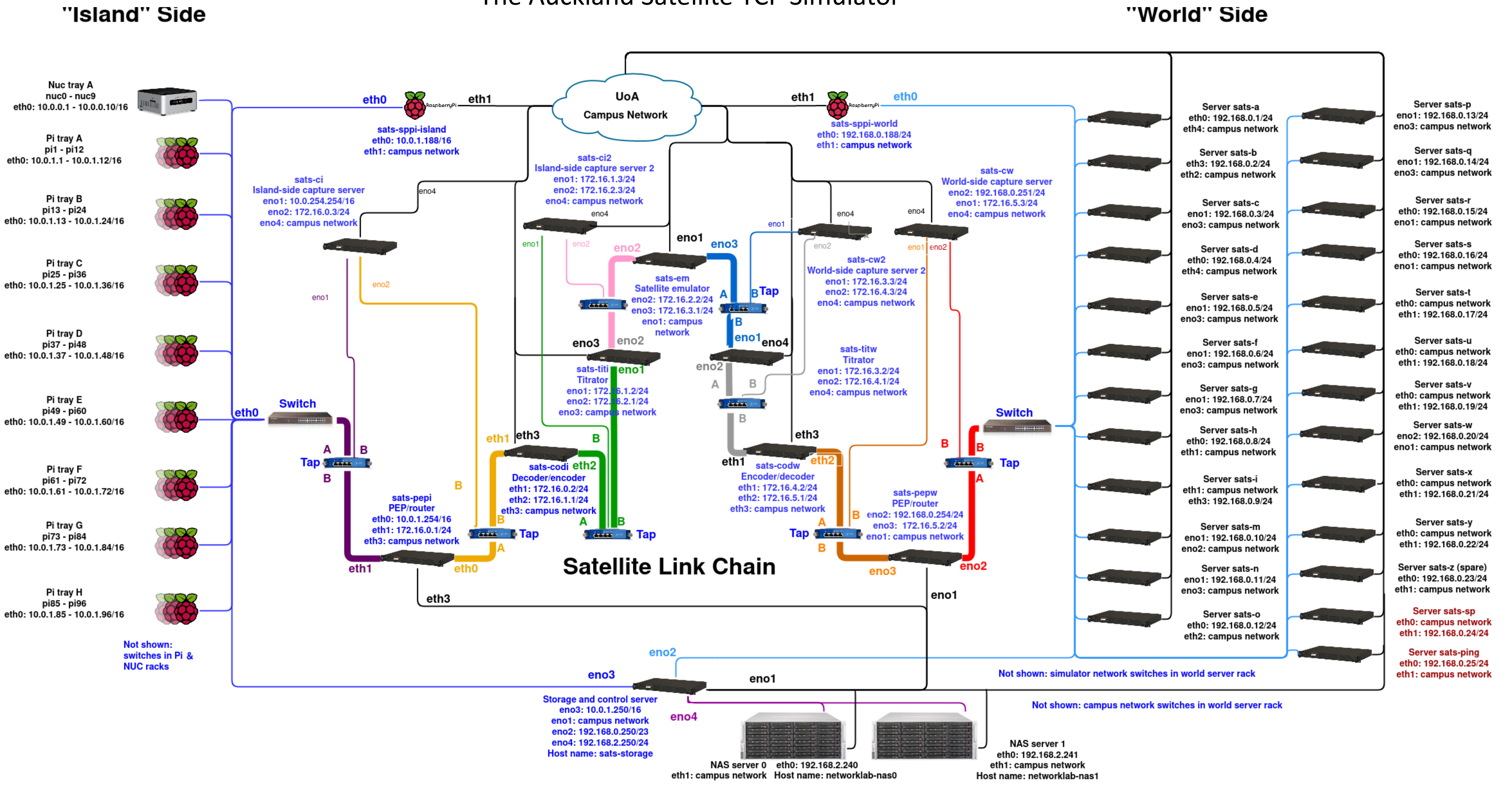
Simulation in hardware

- Uses real packets on real hardware in real time.
- Can build almost any topology.
- Doesn't have a lot of topological flexibility
    - But neither have satellite networks to remote communities – in GEO and MEO, they are dumbbell topologies!
- Cost is medium

The Auckland Satellite TCP Simulator

- Uses real packets on real hardware in real time.
- Can build almost any topology.
- Doesn’t have a lot of topological flexibility
    - But neither have satellite networks to remote communities – in GEO and MEO, they are dumbbell topologies!
- Cost is medium

The Auckland Satellite TCP Simulator



The simulator’s core: the satellite link chain

“World” gateway + PEP
(router and optional connection breaking PEP)

“World”-side network coder
(router and optional network coder/decoder)

“World”-side titrator
(works with world-side network coder to feed link at constant max rate)

Satellite emulator
(latency, bandwidth constraint, input queue)

“Island”-side titrator
(works with island-side network coder to feed link at constant max rate)

“Island”-side network coder
(optional peer to world-side network coder)

“Island”-side gateway + PEP
(router and optional connection breaking PEP)

NTAPs between all stages of the chain

The “world” side:

- 22 SuperMicro 1U servers as TCP background traffic sources
- 1 SuperMicro 1U server as “special purpose” TCP transfer source
- 1 SuperMicro 1U as a “pinger” for RTT measurements
- 1 spare 1U SuperMicro
- 1 Raspberry Pi for signalling
- All have 2 network interfaces (1 GbE or 10 GbE) for:
  - Simulated traffic
  - Simulator control traffic
- Networks tied together by switches at the back
  - Tree topology with world gateway of satellite link chain at the root

The “island” side

10 x Intel NUCs (1 GbE interface)

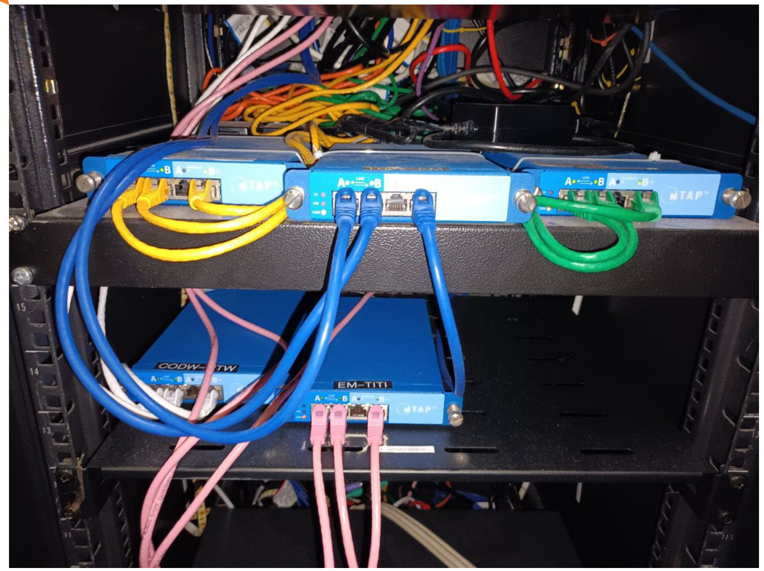
96 Raspberry Pis (100 Mb/s interface)

1 Raspberry Pi for signalling / external access

2 UPS for NUCs and Raspberry Pis

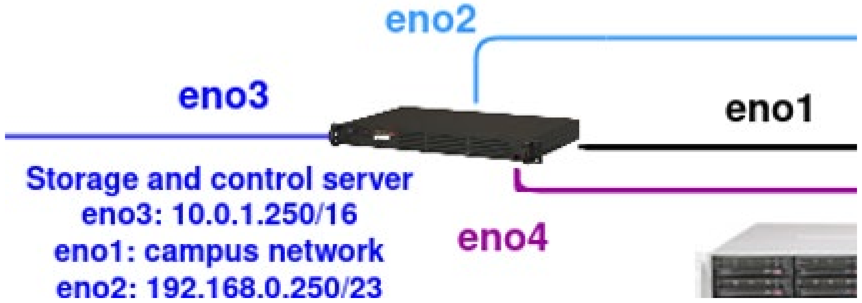
Capture and storage:

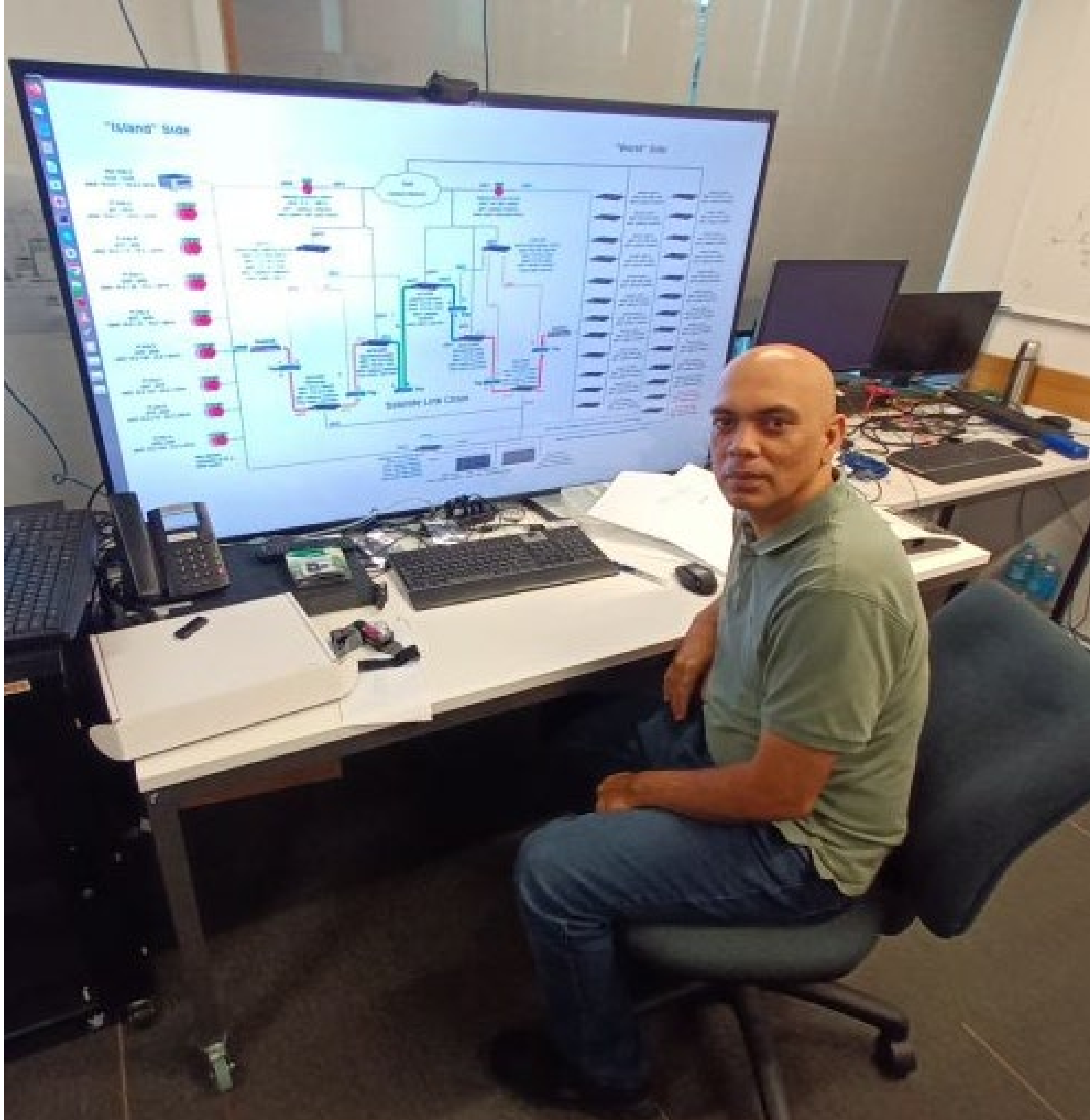
- 4 SuperMicro 1U capture servers
  - Capturing from 8 NTAPs by dedicated cable
- 2 x 96 TB RAIDed (14 x 8 TB each) storage servers
  - On control network
  - On dedicated storage-only network



Command and control

- 1 SuperMicro 1U server with a huge monitor
- Interfaces to all networks: control, island side, world side, storage





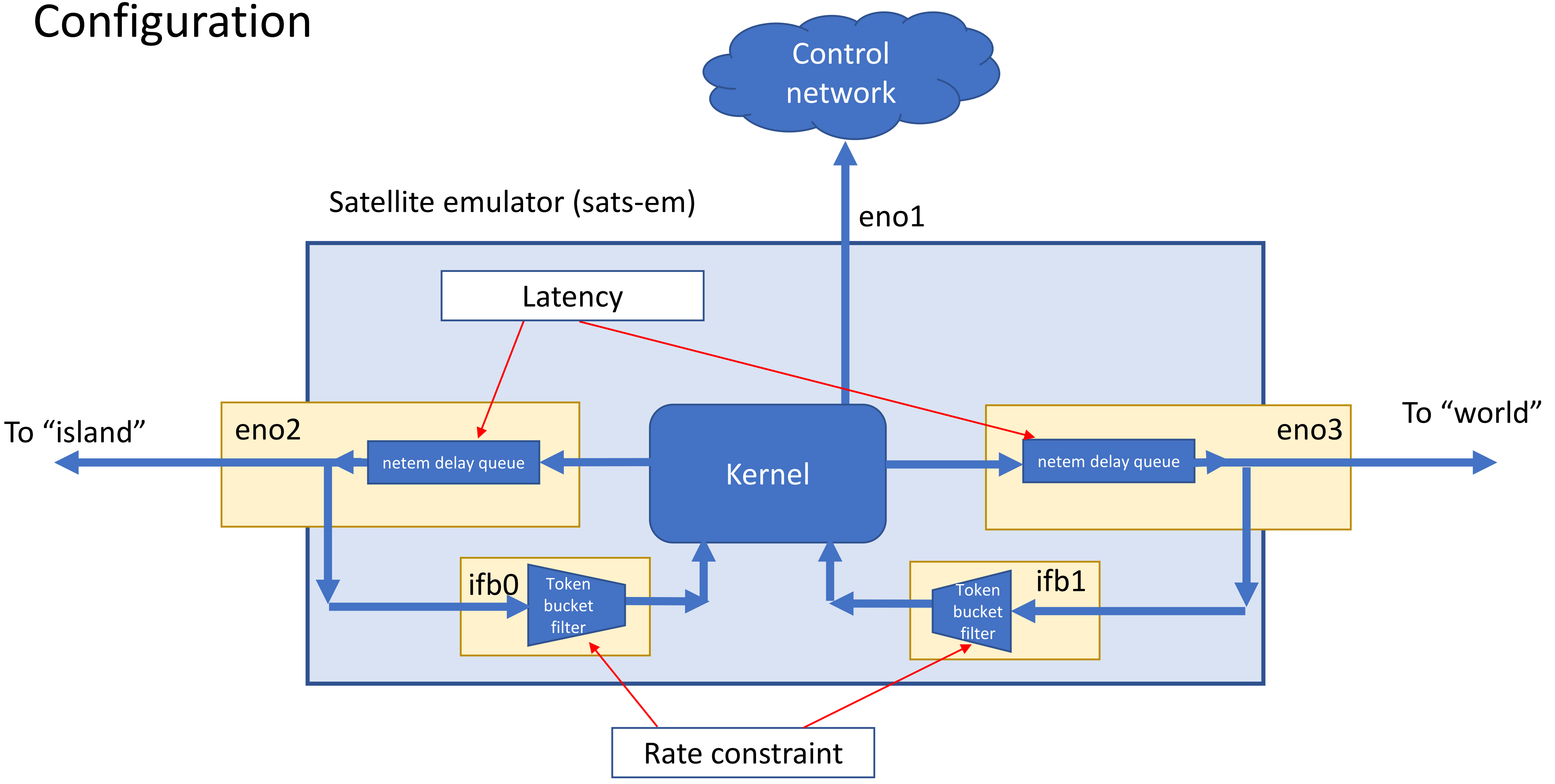
Satellite emulator

- Job: Provide satellite latency and bandwidth constraints
  - E.g., model of a GEO or MEO link is a bandwidth bottleneck with an input queue (e.g., byte FIFO classically), followed by a constant delay queue.
  - Soon to commence: LEO experiments with time-varying latency and bandwidth – allows for satellite movement and handover effects

- Uses both standard Linux and custom qdiscs

- Use of intermediate function block devices avoids accidental buffer insertion

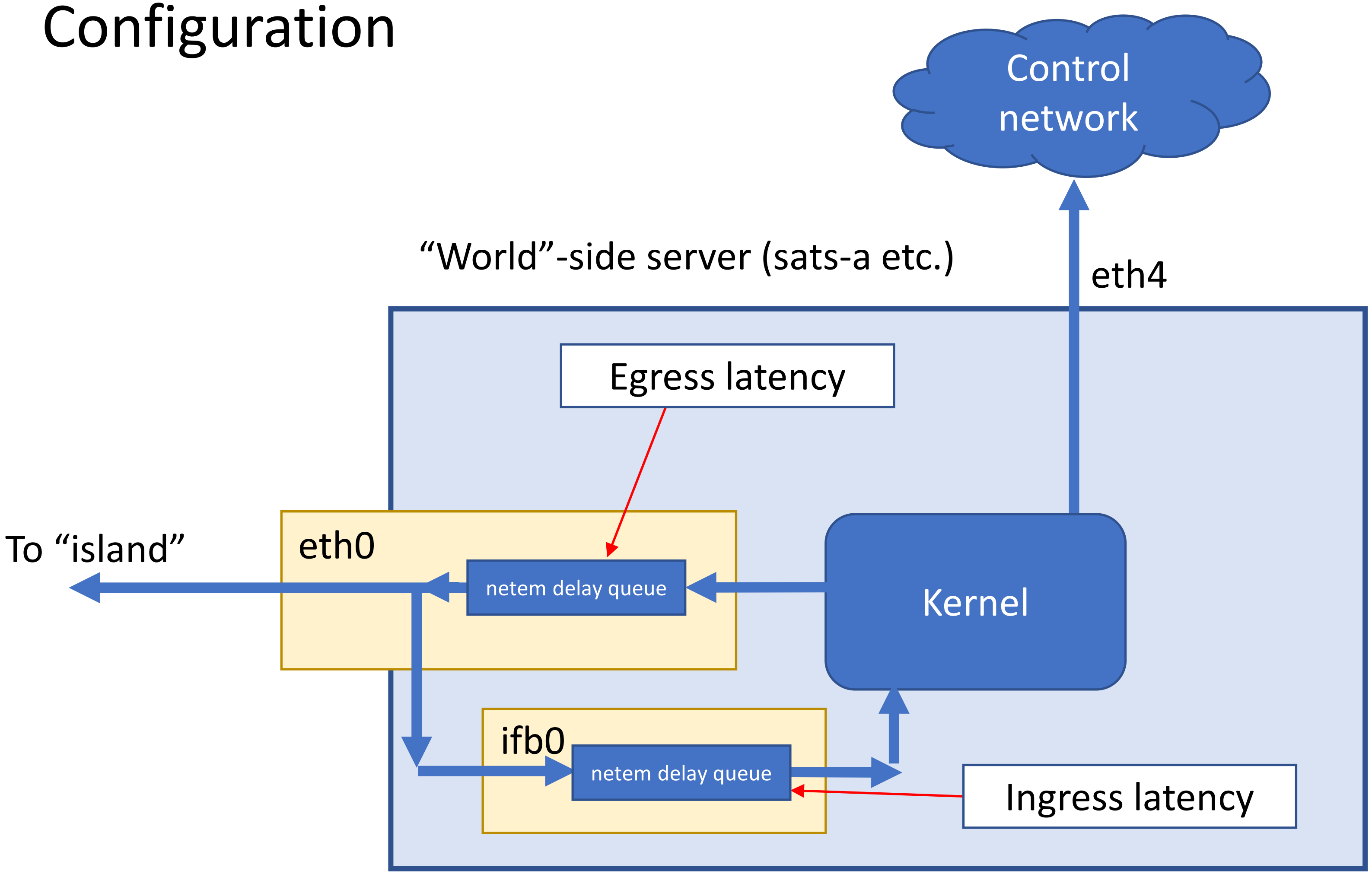
Configuration



“World”-side server configuration

- Job: Provide background TCP traffic volume
- Different “terrestrial” latency between satellite link chain and each server
  - Split into ingress and egress latency via intermediate function block device
- Server functionality:
  - Accept incoming TCP connection requests from “island” clients
  - Randomly select flow size from configurable flow size distribution and deliver the respective data quantity to client.
  - Disconnect client
- Uses purpose-built server software
- Flow size distribution captured on the island of Rarotonga (Cook Islands)

Configuration



Note: Latencies on small islands are negligible

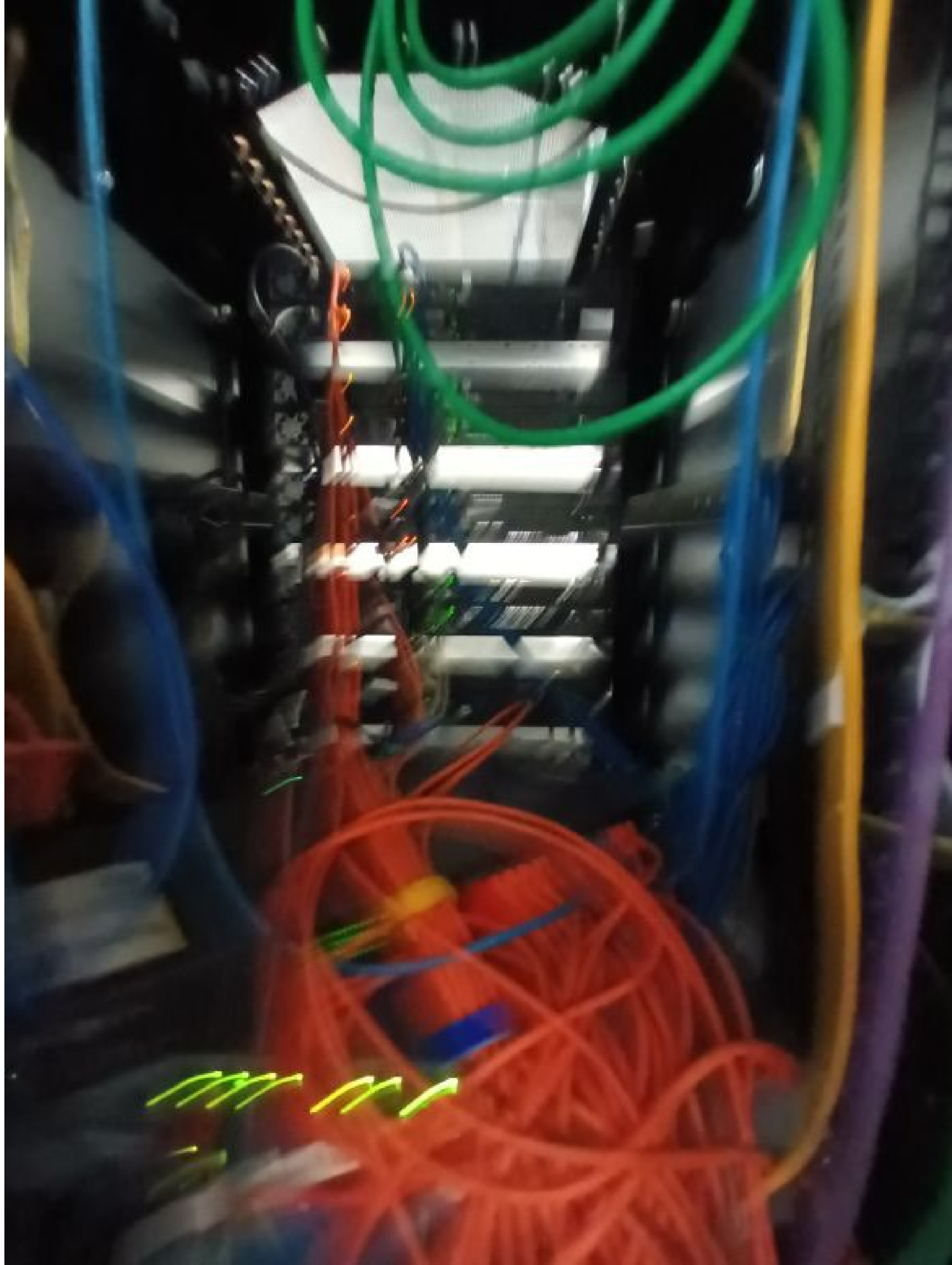
This arrangement is needed for terrestrial latency on "world" side of link only

"Island" client configuration

- Job: Produce demand for background TCP traffic
- No "terrestrial" latency between satellite link chain and clients – islands are small!
- Client functionality (one "channel"):
  1. Randomly select a "world" server
  2. Connect to that server
  3. Receive data from server and count the number of bytes received
  4. Repeat when disconnected
- Run N channels in parallel distributed over as many client machines as required
  - Number of channels = demand level
- Uses purpose-built client software on the "island" Raspberry Pis and Intel NUCs

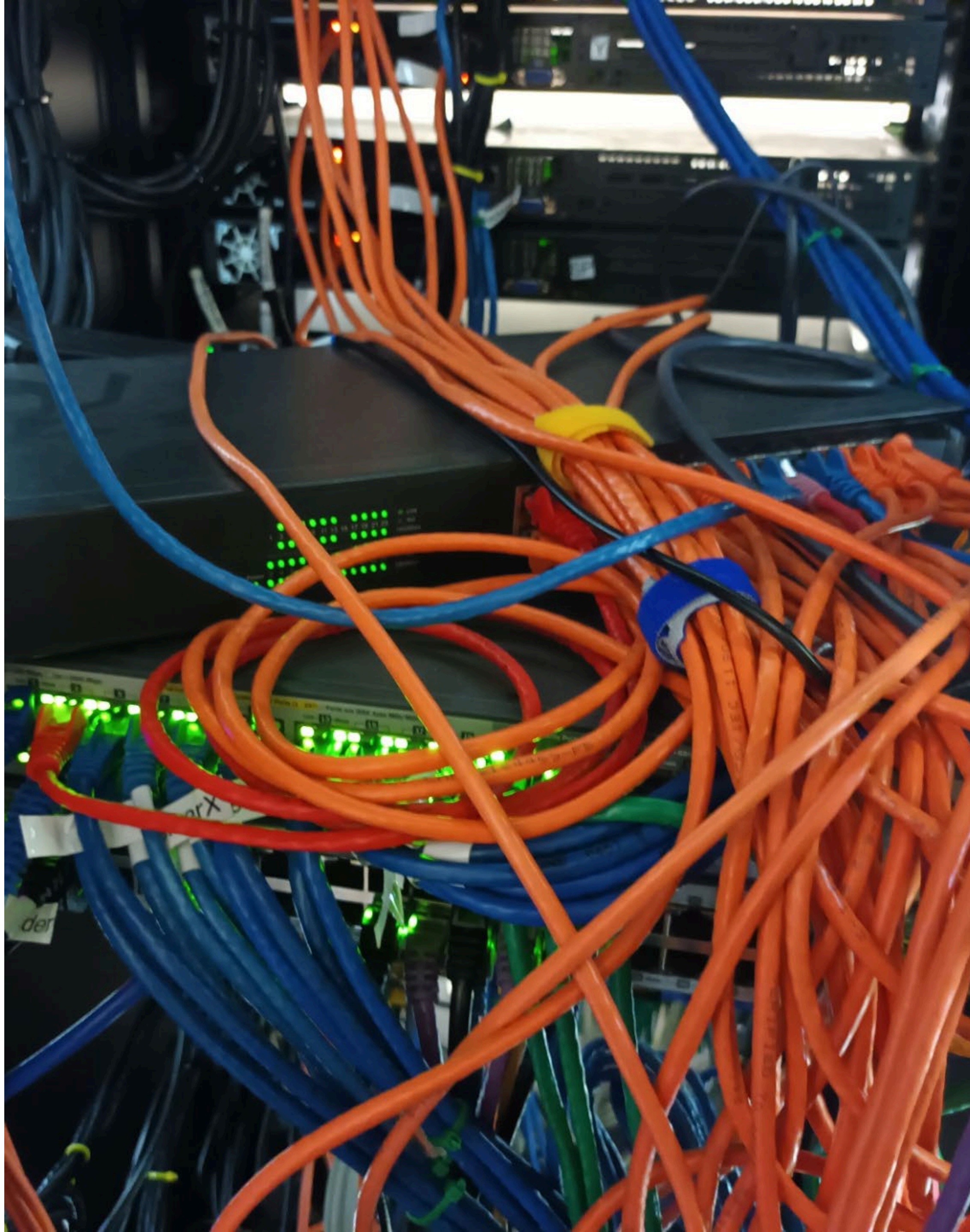
Typical experiment sequence

1. Configure satellite emulator, “world” server latencies and, if used, network coders, titrators and PEPs
2. Start “world” servers
3. Start capture of “island”-bound traffic on NTAPs progressively from “island” side to “world” side
4. Start “island” clients
5. Signal “measurement start” via unusual UDP packet sequence that becomes a marker in all traces
6. Start a rapid-fire ping series from the “world” to the “island” side to measure queue occupancy at the satellite emulator
7. Start a (time- or volume-bound) iperf3 transfer from “world” special purpose server to one of the NUCs
8. Keep running for a while (~90 s to 600 s depending on link type)
9. Shut down “world” servers (“island” clients time out automatically)
10. Shut down packet capture on NTAPs progressively from “world” side to “island” side (reverse order of capture start)
11. Retrieve captured traces, ping log, goodput data, and iperf3 log
12. Analyse data



Experiment observables

- Goodput (as seen by client application layer or as TCP payload data passing into the island network – these are not the same!)
- Satellite link input queue occupancy over time
- Either:
  - Time required to transfer a given amount of TCP data with iperf3, or
  - Amount of data transferred by iperf3 in a given amount of time
- Packet loss
- Payload data loss
- …

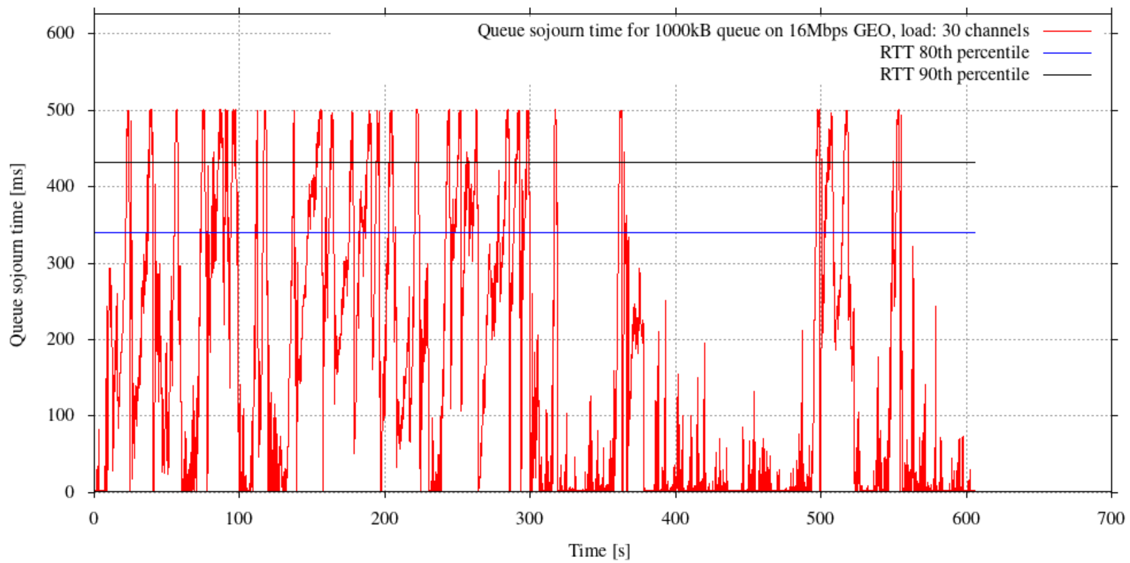


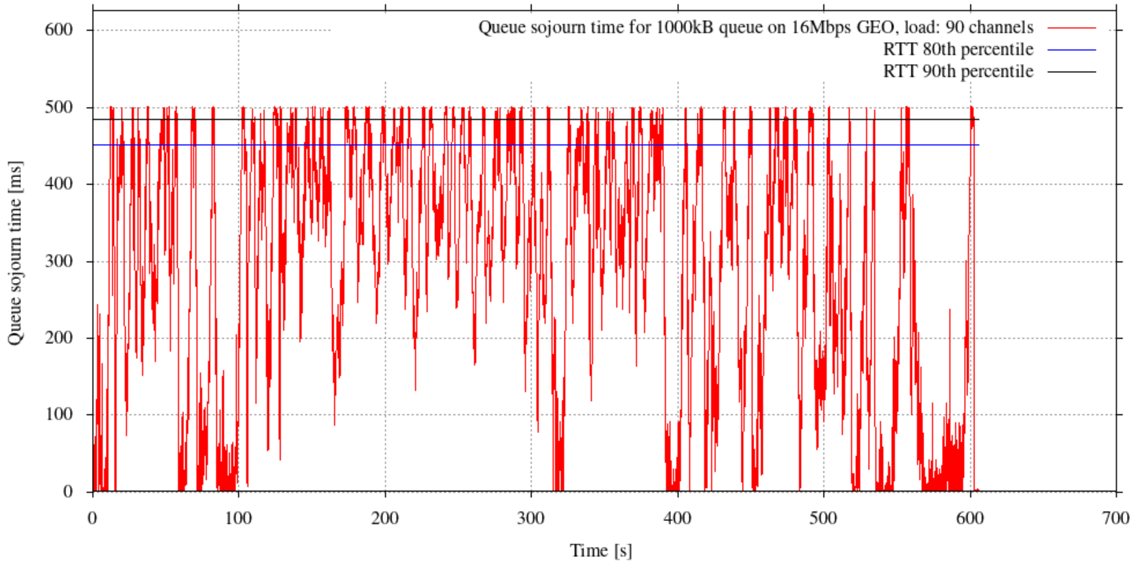
Experiment batches

- One experiment doesn't tell us much. Variation due to:
    - Random choices on "island" clients and "world" servers
    - Natural differences in processing due to independent machines being used
- But: a single experiment can produce up to almost 1 GB in trace and other data
- Need to repeat each experiment in each configuration multiple times => run batches
    - Practitioner's Pitfall Alert:
      ARP and Path MTU cache state
      can impact on next experiment!
- Need large file storage for results

Sample results

Queue sojourn time on a 16 Mb/s GEO link, 1000 kB buffer, 30 and 90 channels, 600 s





300 s iperf3 transfer: 348 MB
Total goodput seen by clients: 824 MB
Link utilisation: 11.7 Mb/s

300 s iperf3 transfer: 79.4 MB
Total goodput seen by clients: 1083 MB
Link utilisation: 15.3 Mb/s

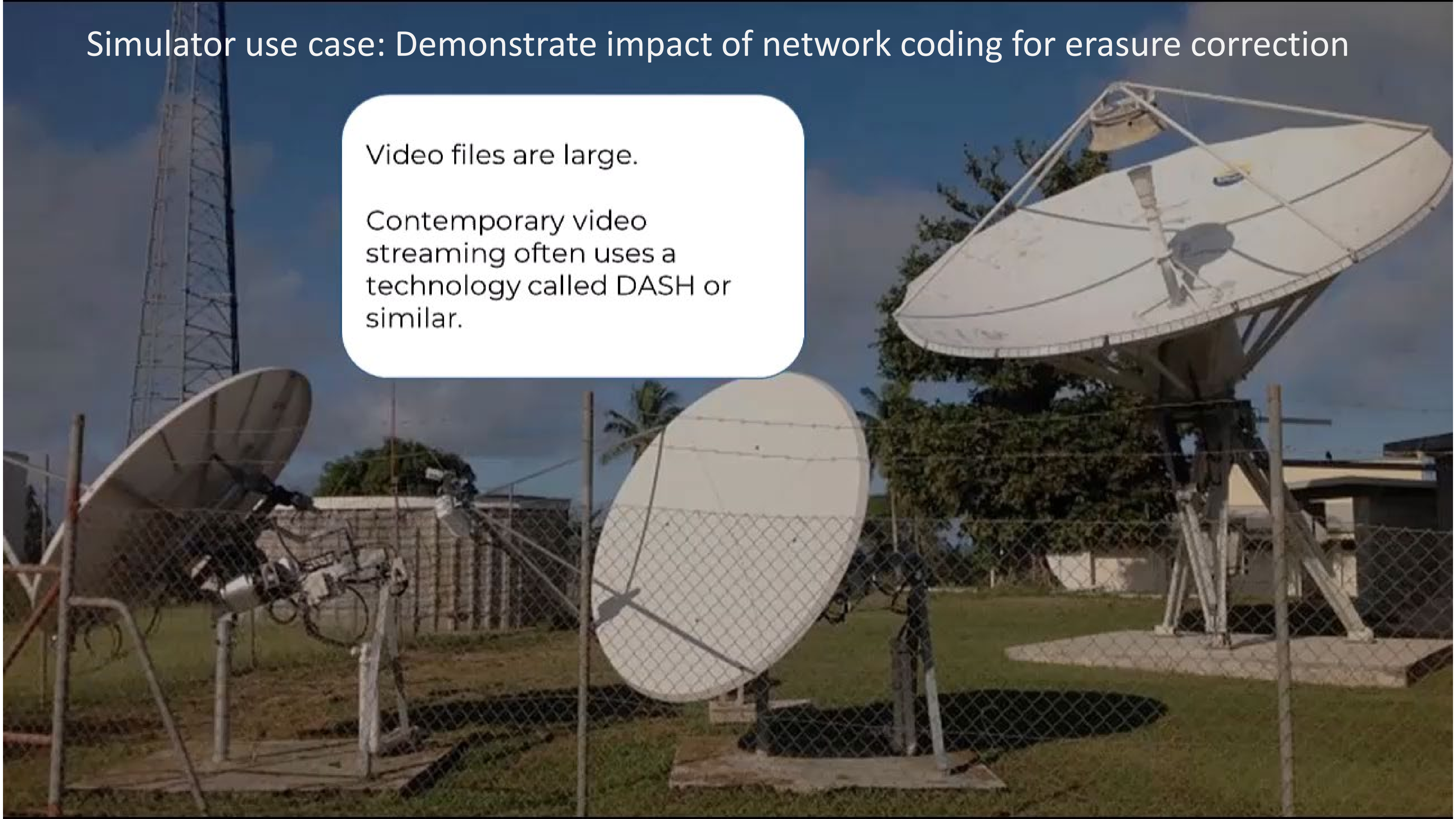
Observations:

- Long TCP transfers (iperf3) suffer disproportionately when load increases
- Low TCP link utilisation, long maximum RTT

Simulator use case: Demonstrate impact of network coding for erasure correction

Video files are large.

Contemporary video streaming often uses a technology called DASH or similar.



From GEO and MEO to studying LEO

- "Dumbell" topology is unusual on LEO
  - "Direct to site" rules! Do we need as many clients?
- Latency:
  - Continuous change from satellite movement
  - Jump change from handovers
    - Ground-to-sat handovers
    - Inter-satellite link handovers
- Bandwidth / capacity:
  - Jump changes in capacity during handover to satellites with more / fewer users
  - Complete change to new queues in a handover
    - Do we need those extra clients after all?
- Location:
  - Satellite density varies with constellation design and location
  - User density impacts on available capacity (slots)
  - Geostationary arc protection in some areas impacts on availability
- Content Delivery Network caches
  - Carry about 70% of current Internet traffic
  - No obvious location for caches in LEO topology
    - Do we need to use a different traffic profile?

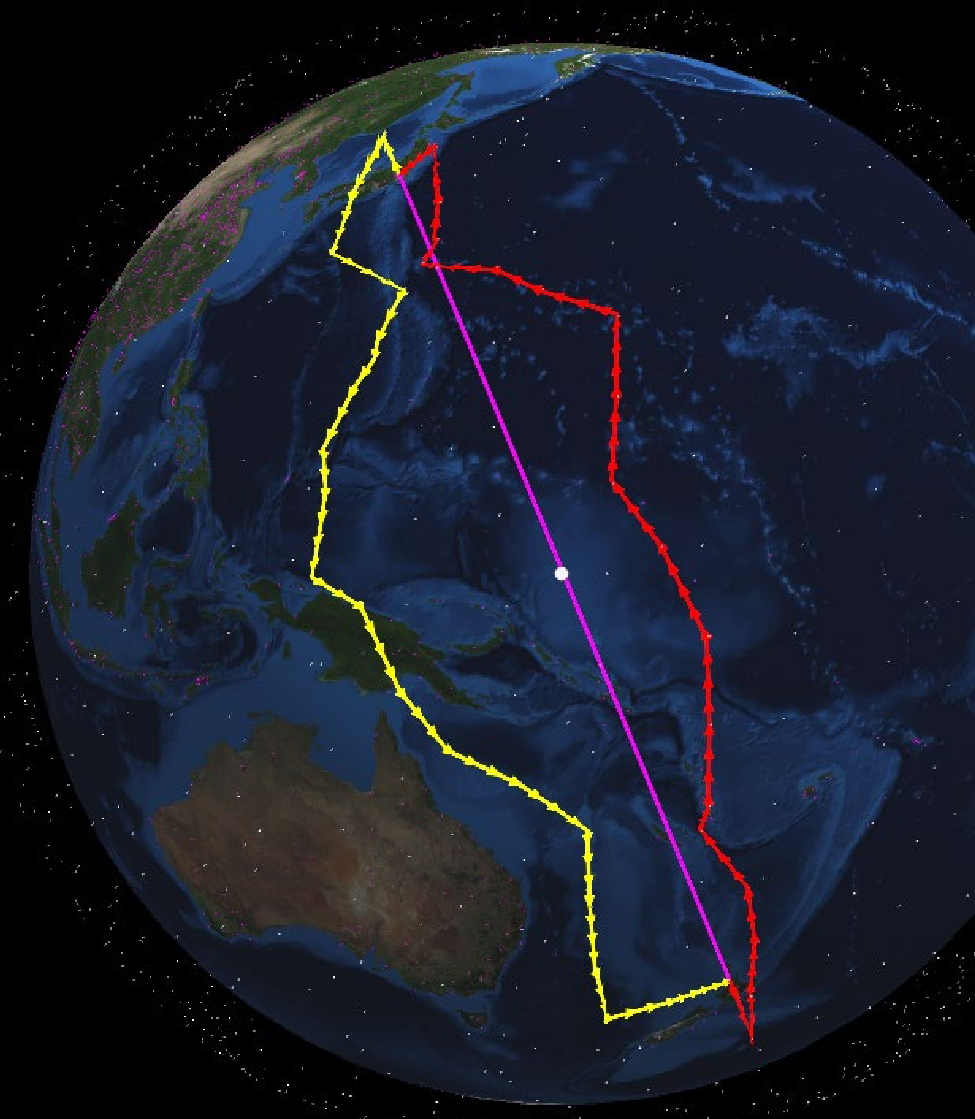
LEO bandwidth and latency simulation

- Netem queue for latency
  - Can change latency on the fly
  - Data source: Toby Tomkinson's constellation and ISL simulator

- Lucas Betts' custom token bucket filter qdisc enforces maximum instantaneous bandwidth
  - Can change latency on the fly
  - Data source for bandwidths? May need to make something up here. Starlink observations?

- Use scripts to apply these changes

- But can we simulate competing user groups on satellites during handovers?
  - BTW: What happens to packets already enroute during handovers? Are they lost / delayed / redirected?



Toby Tomkinson's simulator simulating a return path Tokyo-Auckland

Conclusion

- Can run realistic simulations in real time
- Complexity! The devil is in the detail…
- Orchestration of over 140 devices
- If one part falls over, batches stop and need to be restarted
- Going into the future: How do we simulate LEO networks?
  - Direct-to-site, but also to small networks now
  - Many more variables – even higher complexity:
    - Constellations – many different options
    - Routing – we know it happens, but how?
    - Load changes – during handovers
    - Business model – how do we account for (no) CDNs?
    - Generally opaque system operations


Thank you!
Kia Orana
Welcome
Aitutaki
ANDWIDTH PLACE
Speed Test. Any Device. Anywhere.
Start
Download
0.78
Upload
1.43
Ping
855
My Results
Status
Ready
CS-NRG-CTCP1.TEL
esa
Ulrich Speidel - ulrich@cs.auckland.ac.nz

Acknowledgements

APNIC Foundation/ISIF Asia

Internet NZ

Nevil Brownlee

‘Etuate Cocker

Telecom Cook Islands (Bluesky)

Brian Carpenter

Internet Niue

Emina Soljanin

Tuvalu Telecom

Lei Qian

PICISOC

Steinwurf ApS

Muriel Médard

MICT Kiribati

Wayne Rei

Frank Kschischang

…and many more!