DIMECC

ONE SEA
Autonomous maritime ecosystem

Sea4Value

Future Fairway Navigation

Benchmarking of Emulated Wireless Edge Cloud Connectivity for Maritime Environments

19.03.2021

awake.ai BRIGHTHOUSE INTELLIGENCE MEYER TURKU SHIPYARD 1737 Haltian ERICSSON FINNPILOT ESL Shipping NESTE Suomen Varustamot Rederierna i Finland Finnish Shipowners' Association FINNISH METEOROLOGICAL INSTITUTE TRAFICOM VÄYLÄ PORT OF HELSINKI RAUMAN SATAMA PORT OF RAUMA PORT OF TURKU FINLAND Aalto University NOVIA UNIVERSITY OF APPLIED SCIENCES UNIVERSITY OF TURKU Tampere University JYVÄSKYLÄN YLIOPISTO UNIVERSITY OF JYVÄSKYLÄ BUSINESS FINLAND

Team

- Tampere university
  - Antti Kolehmainen
  - Billhanan Silverajan
- Ericsson
  - Miika Komu
  - Jimmy Kjällman
  - Tero Kauppinen
  - Fayezeh Ghavimi
  - Sepehr Javid

Participation

WP3 Technical transformation

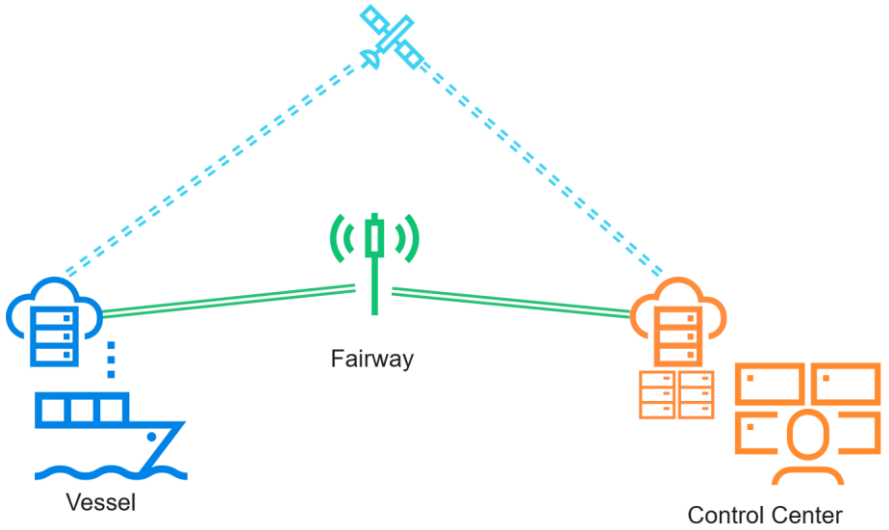
- WP3.3 Fog and Edge Computing

Experiments

I External traffic shapers

II Internal traffic shapers

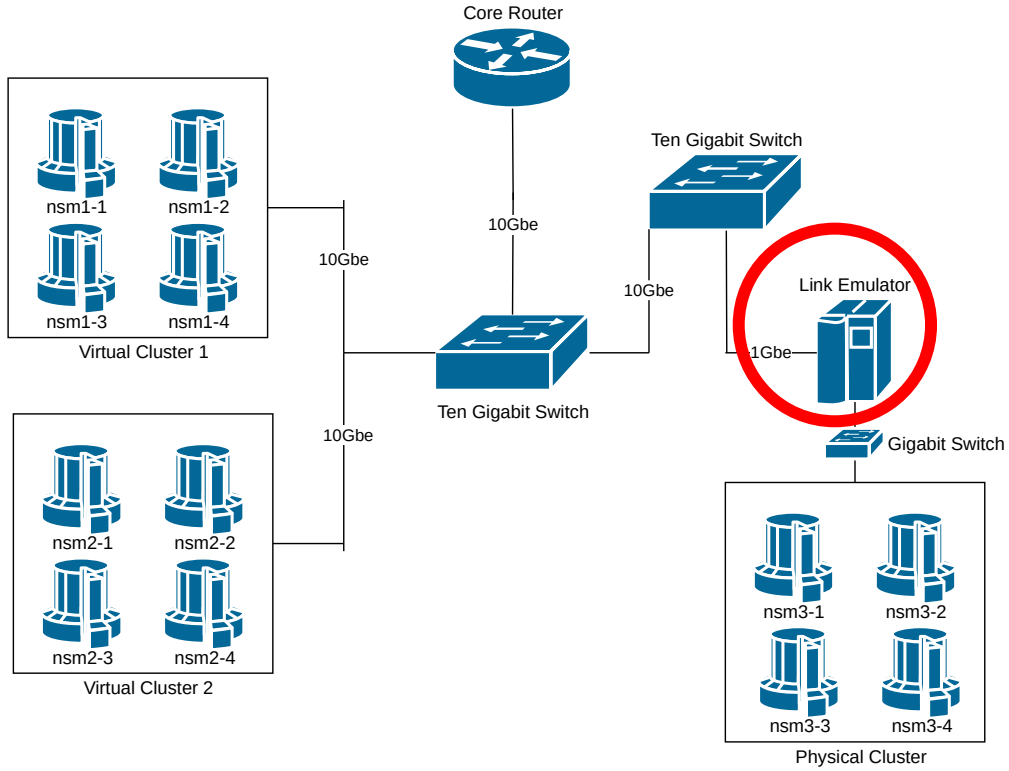
Introduction



- Multi-cloud edge computing in ships and control centers
  - Edge clouds operated as a single unit..
  - ..but remain autonomous upon network partitioning (Federation)
- Ships can utilize both cellular and satellite connectivity
  - Different latency and bandwidth characteristics
  - Moving ships + wireless network = varying traffic loss
- Wireless connectivity + multi-cloud edge computing = ???
  - Emulated testbed for wireless multi-cloud networking (Kubernetes)
  - Wireless network emulated using a traffic shaper in a wired network

Testbed

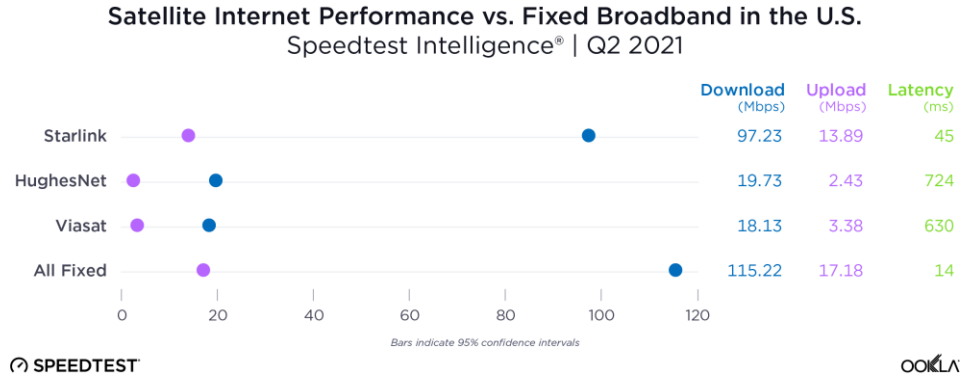
- Federated Kubernetes
  - Bare metal + virtual machines
- Network Service Mesh
- External link emulator
  - Mimics 5G and satellite network
  - FreeBSD + dummynet
  - Controls packet loss rate, bandwidth and latency
- Traffic generator: iperf3 (TCP)



Traffic Generation and Emulation Parameters

Service	Data Rate
GMDSS	10 Kbps
Radar/AIS	100 Kbps
Infotainment	1500 Kbps
LIDAR	2000 Kbps

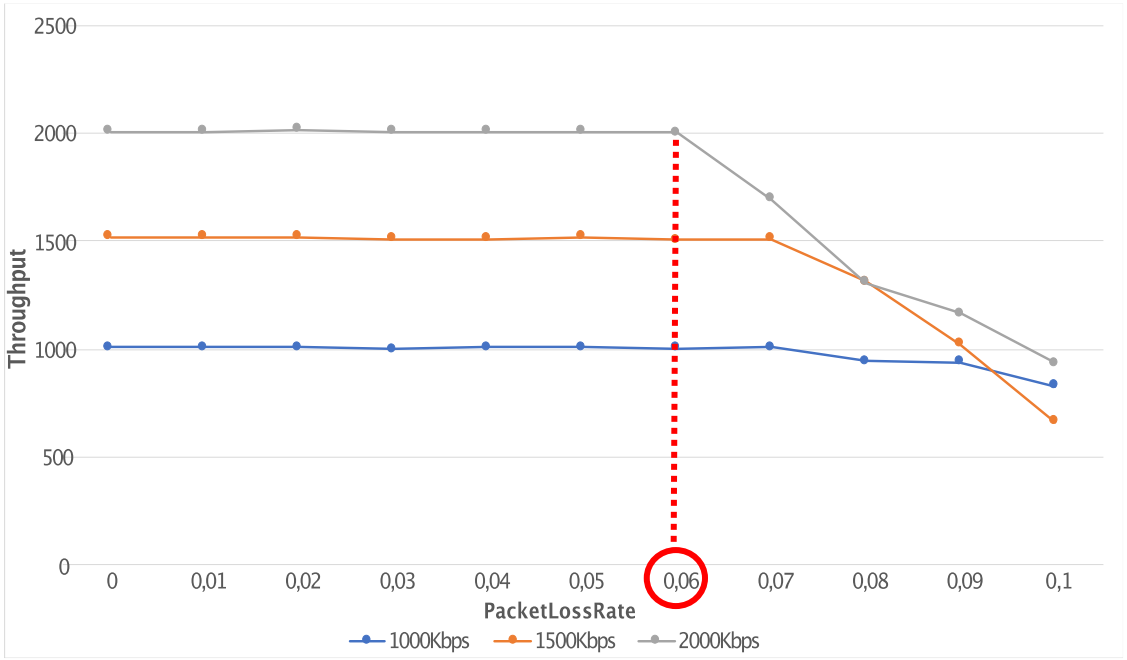
Table III
BANDWIDTH USED BY DATA SERVICE



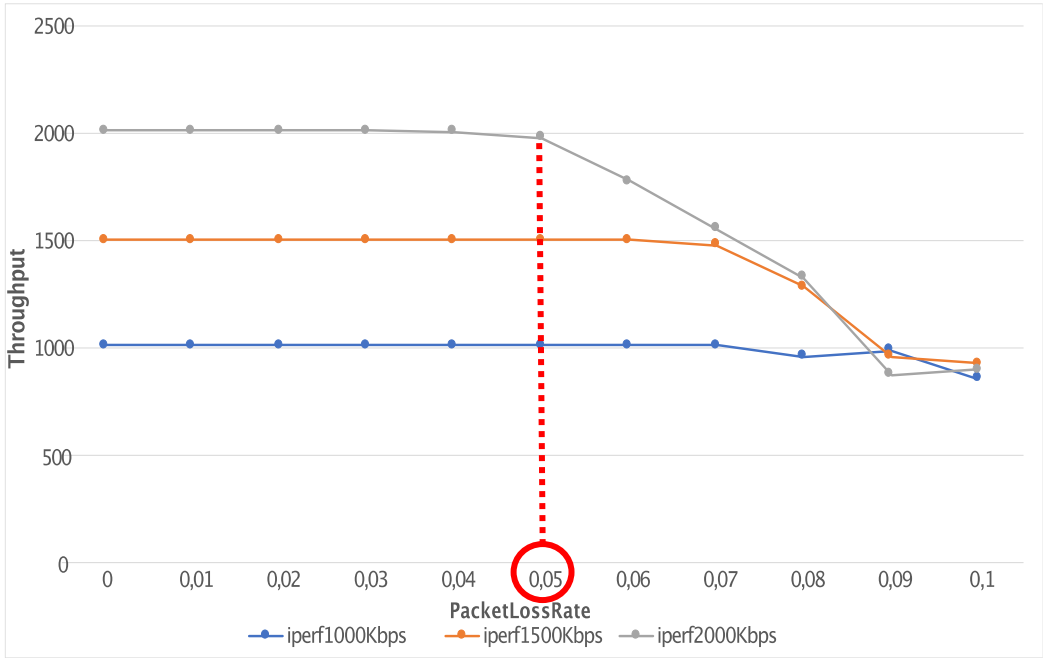
Service	Uplink (Mbps)	Downlink (Mbps)	Latency up/down (ms)
Satellite #1	5	30	125/125
Satellite #2	3	18	315/315
5G #1	88	535	6/6
5G #2	29	592	13/13

Table IV
DUMMYNET LINK PARAMETERS

Benchmark results 1/4

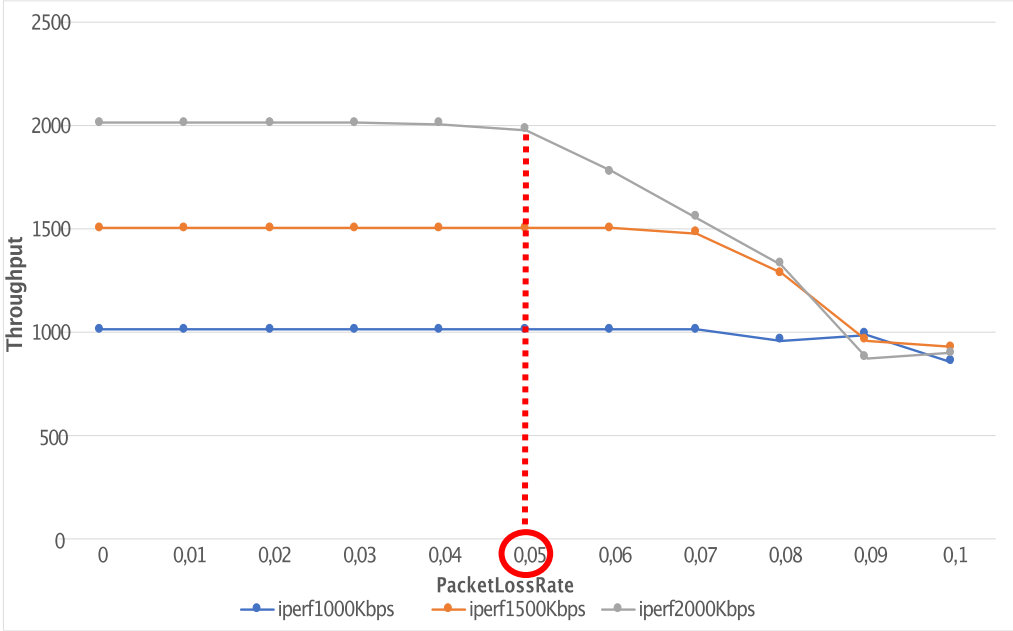


Host networking with 5G #1 (non-Kubernetes)

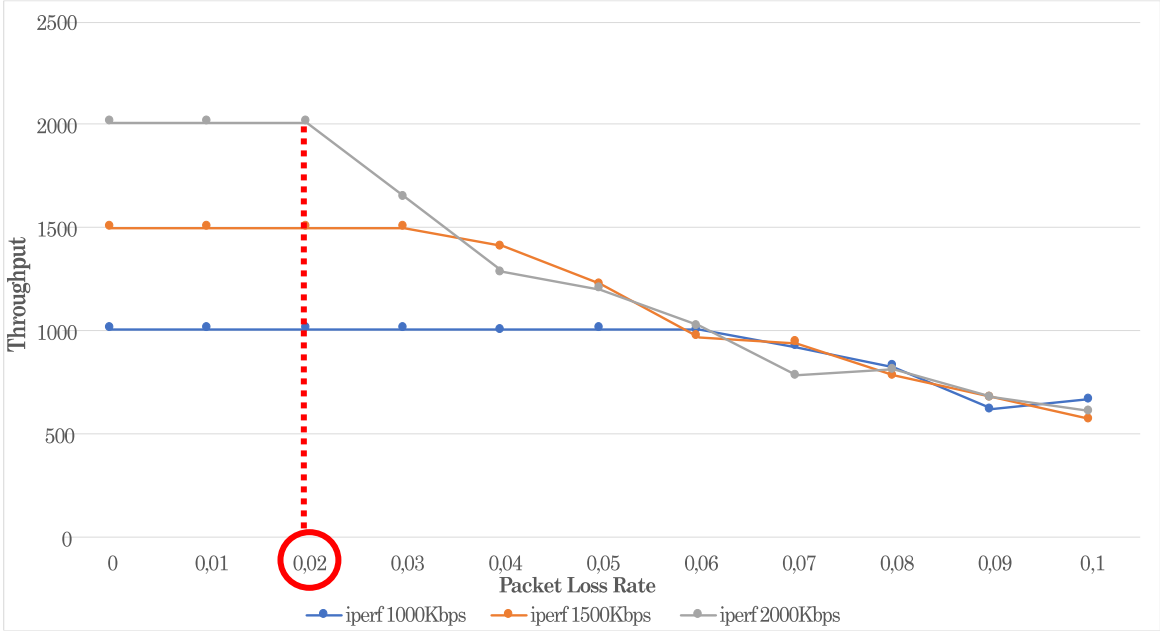


Kubernetes with 5G #1

Benchmark results 2 / 4

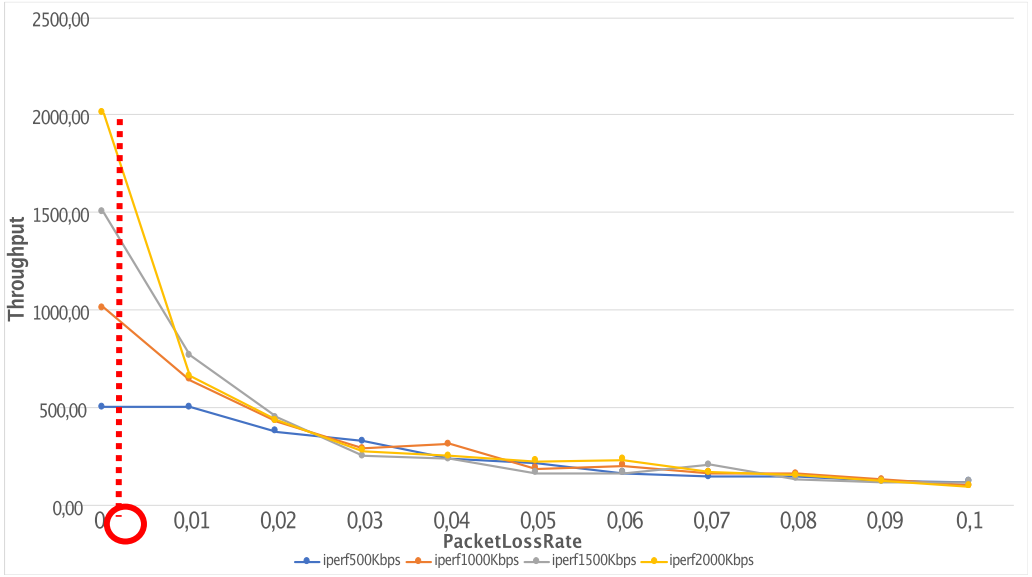


Kubernetes with 5G #1 (6 ms latency)

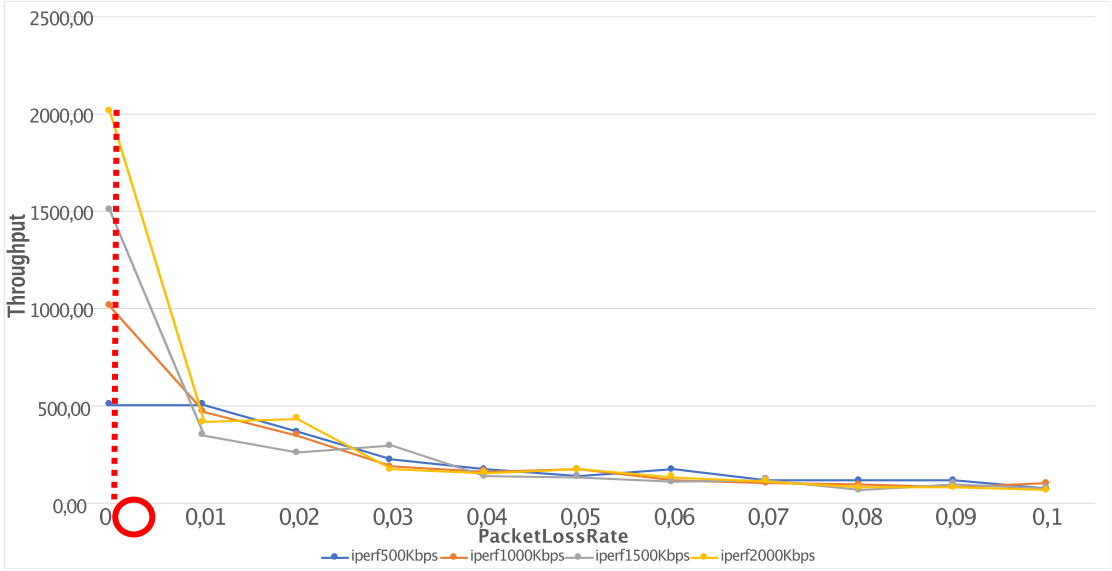


Kubernetes with 5G #2 (13 ms latency)

Benchmark results 3 / 4

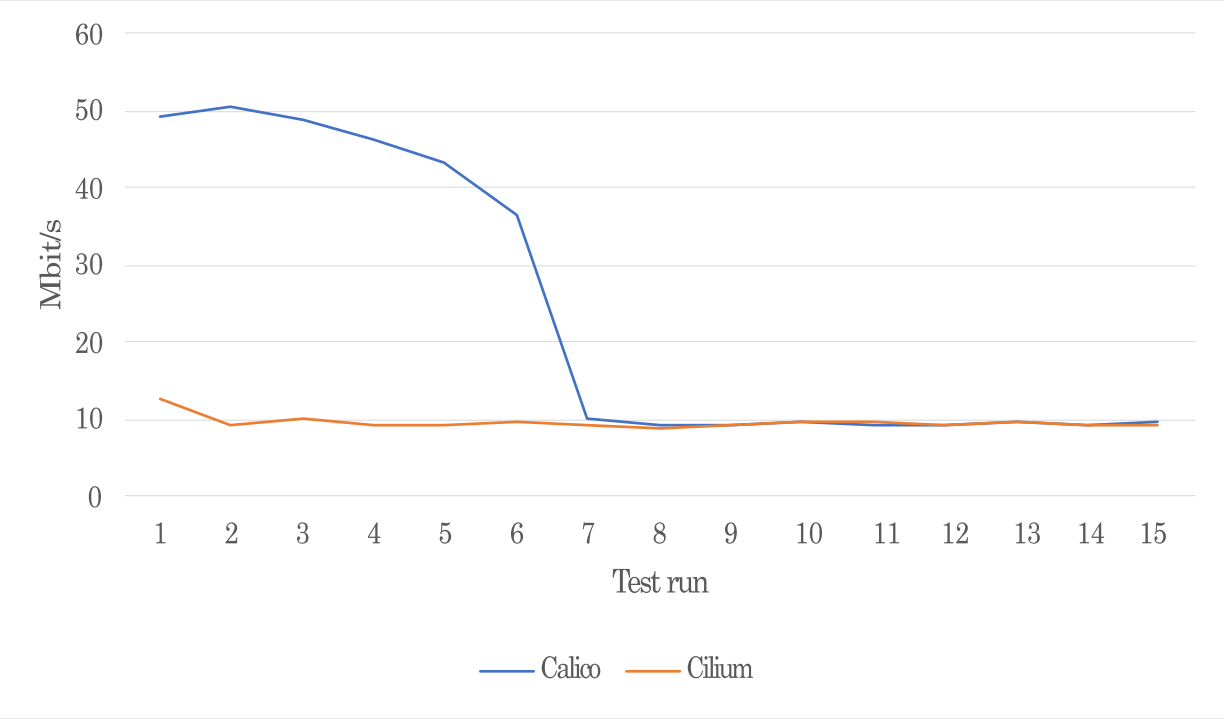


Kubernetes with Satellite #1 (latency 125 ms)



Kubernetes with Satellite #2 (latency 315 ms)

Benchmark results 4 / 4



- Benchmarked also Kubernetes internal traffic shapers
  - Calico and Cilium
  - Single cloud because only Cilium supports multi-cloud
- Bandwidth capped to 10 Mbit/s
  - Cilium works ok but caps only outbound direction
  - Calico lets some traffic through initially and did not work reliably with smaller b/w cap

Conclusions & Future Work

- Emulated multi-cloud networking in a wireless environment
  - Benchmarked cloud internal and external traffic shapers in Kubernetes
  - Paper accepted to IEEE World Forum for Internet of Things
- Main findings
  - Kubernetes networking does not add much overhead
  - Increase in latency impacts bandwidth negatively with TCP
  - Kubernetes internal traffic shapers can further be improved
- Future work
  - Benchmarking of CPU usage
  - Parallel data connections & automated traffic prioritization and shaping

Thank you!

Any questions?