

Storage in Space

Jörg Ott · Jussi Kangashargu · Nitinder Mohan

2 July 2026

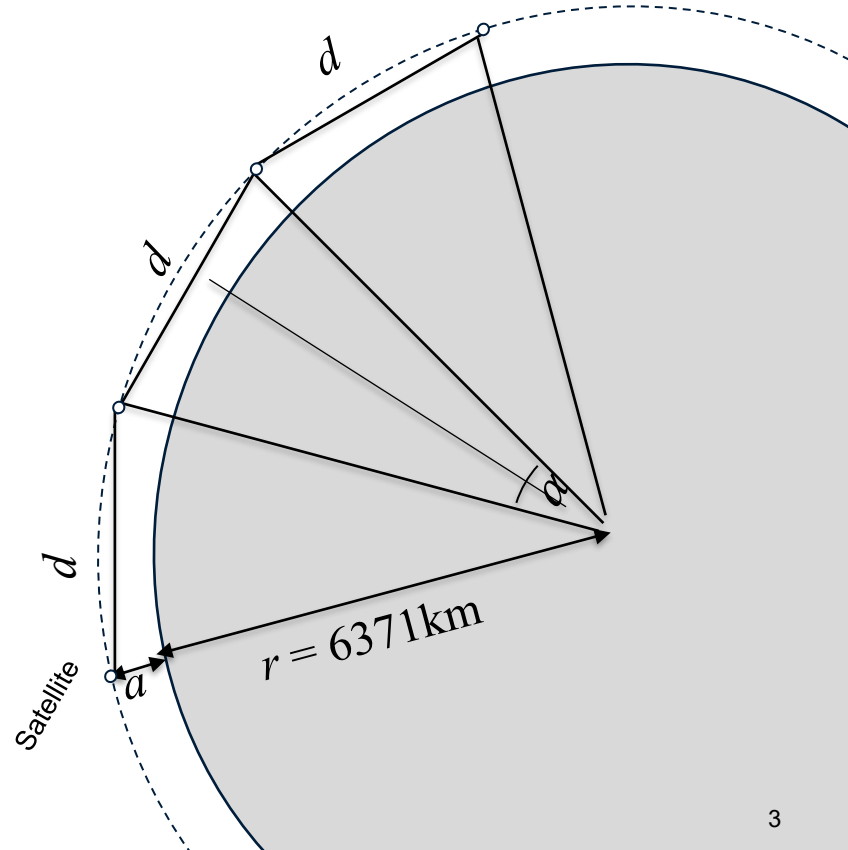
Coseners 2026

Two simple old observations

- Bandwidth x Delay Product $\approx$ short-term “storage” of data in flight
 - $O(\text{ms} - \text{s})$
 - Long fat pipes
 - Link via a geostationary satellite @ 300 Mbps: ~ 23 MB
- Router queues $\approx$ short-term storage in devices
 - $O(\mu\text{s} - \text{s})$
 - Identifiable content could be accessed while queued: tagged packets
 - Implicit caching in Delay-tolerant Networking (DTN) nodes
 - Explicit caching in Information-centric Networks (e.g., CCN/NDN content store)

Let's look at Starlink

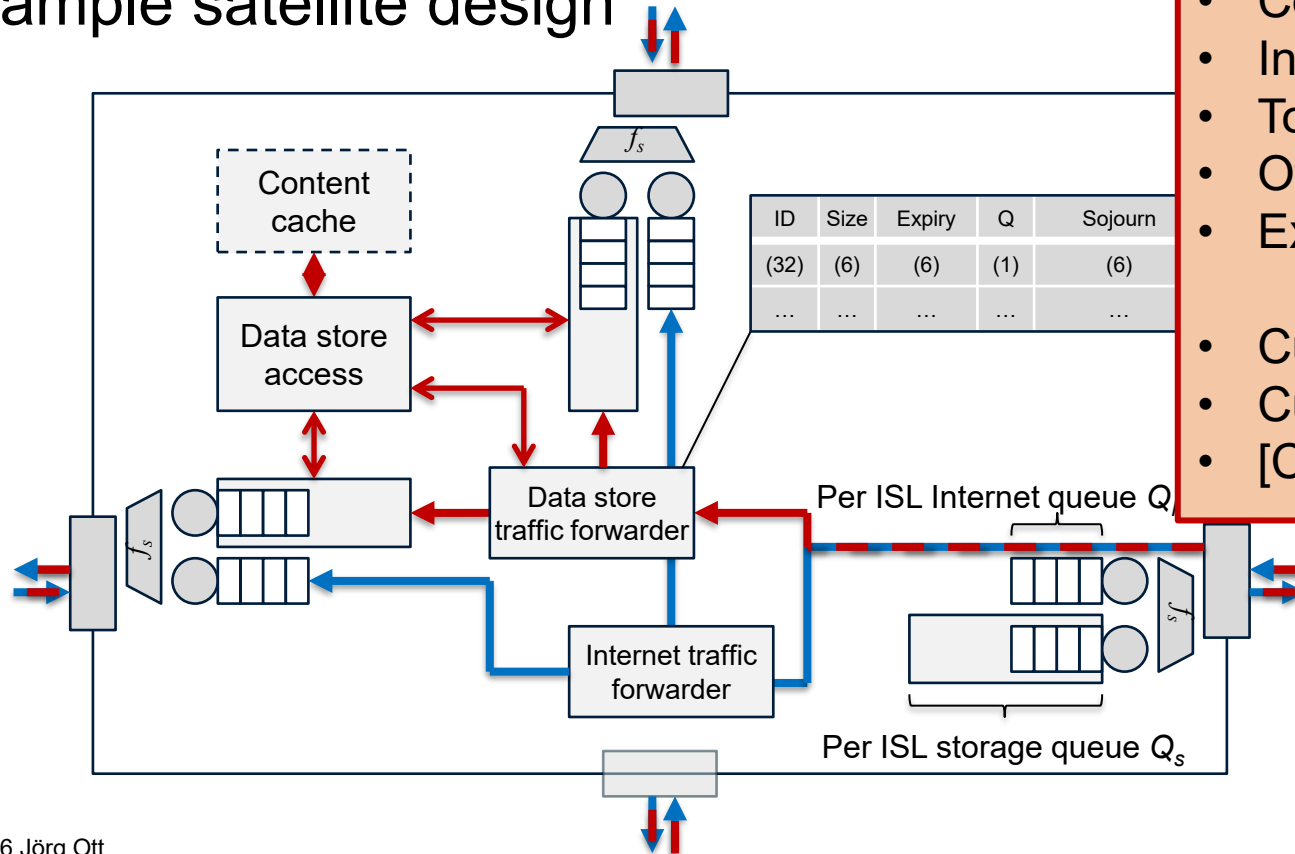
- $a = 550 \text{ km}$
- $N_o = 72$ orbital planes
- $N_p = 22$ satellites per orbits
 - $\alpha = 16.36^\circ$
 - $d = 1970 \text{ km}$
- Max inter-orbit distance
 - $d_p = 604 \text{ km}$
- Max inter-sat distance of adjacent orbits
 - $d_s = 2060 \text{ km}$
- Total distance through all 1584 satellites
 - 3.27M km
 - BDP: 136 GB @ 100 Gbps ISLs



Storing data in space – literally...

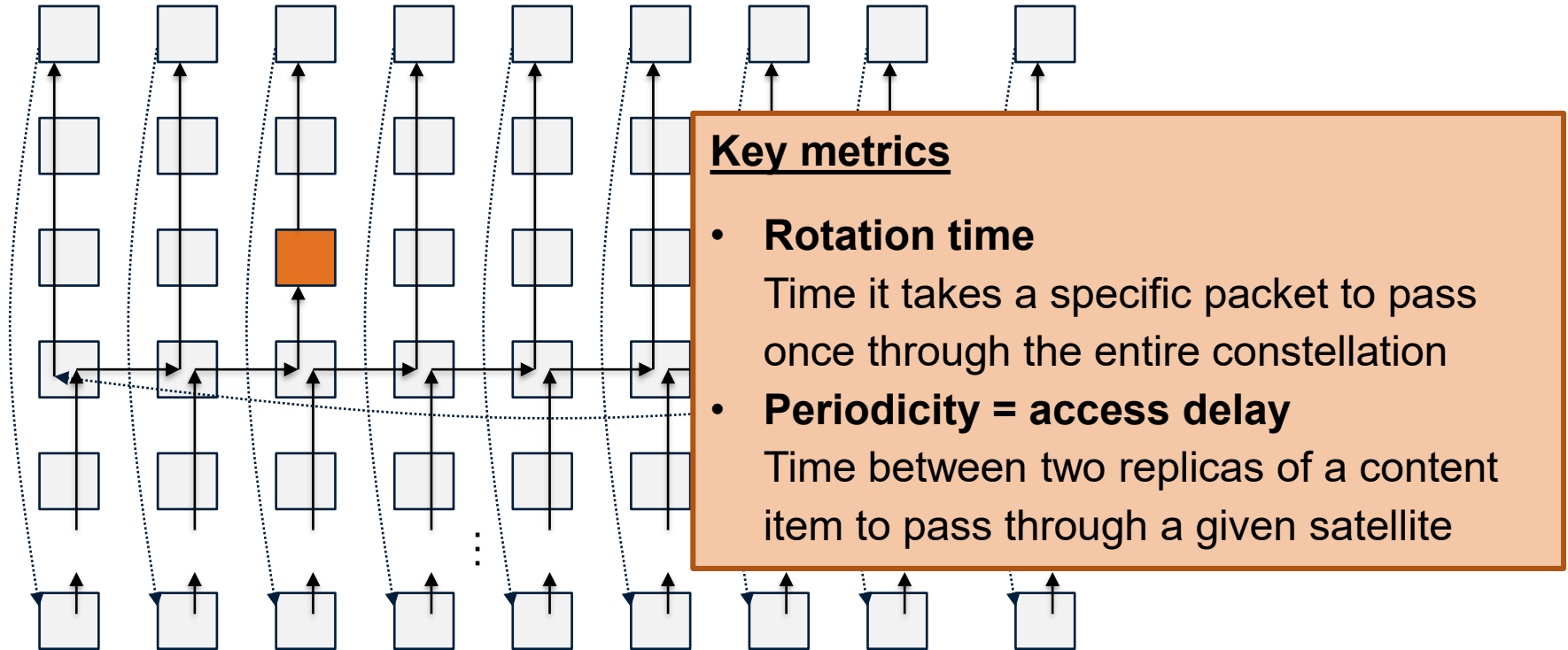
- Keep data constantly in flight along the orbits through all satellites
 - Receive – [queue] – forward
 - Using a suitable forwarding regime
- With brief pauses while queued in each satellite
 - Accessible at a satellite for this brief time window
 - Also allows for recovery / repair
- Practical side effect
 - No need to index data for retrieval – will pass by anyway
- Don't use ISLs exclusively: assume 80% share for now

Sample satellite design



- Content identifier
- Instance count
- Total content size
- Offset
- Expiration time
- Current satellite
- Current orbit
- [Current orbital plane]

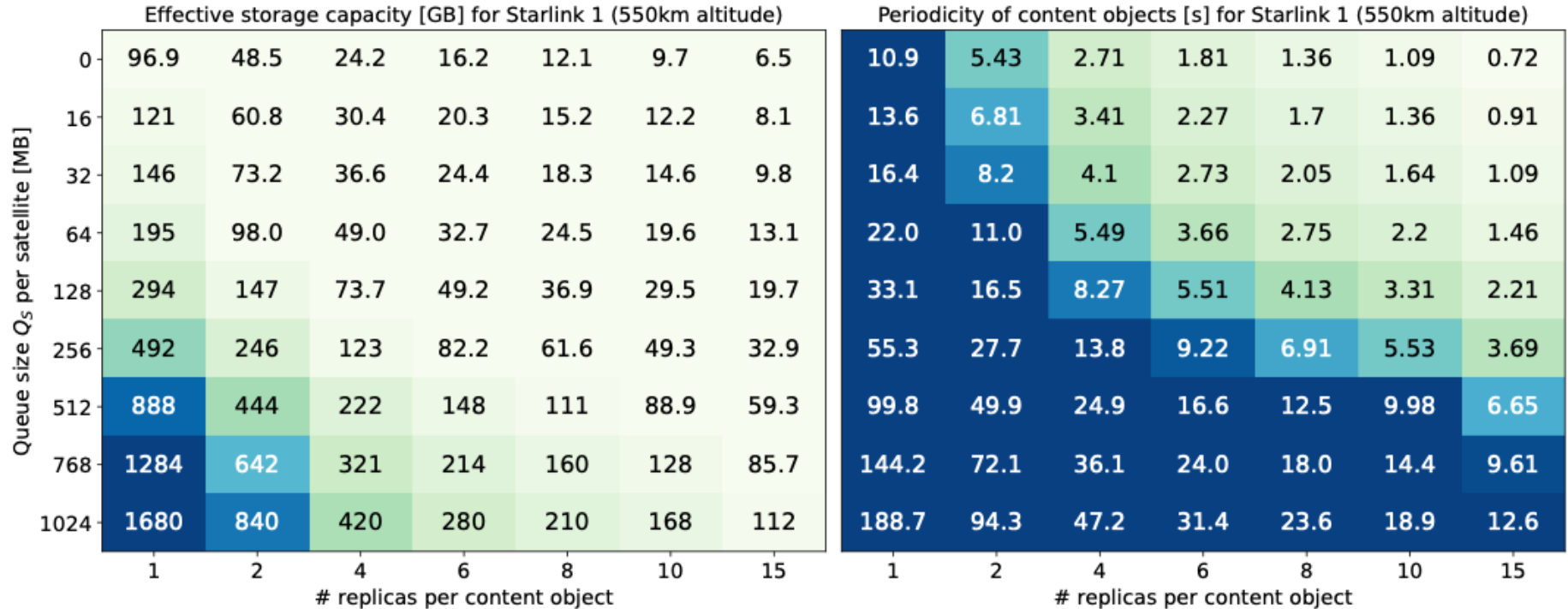
Simple packet routing



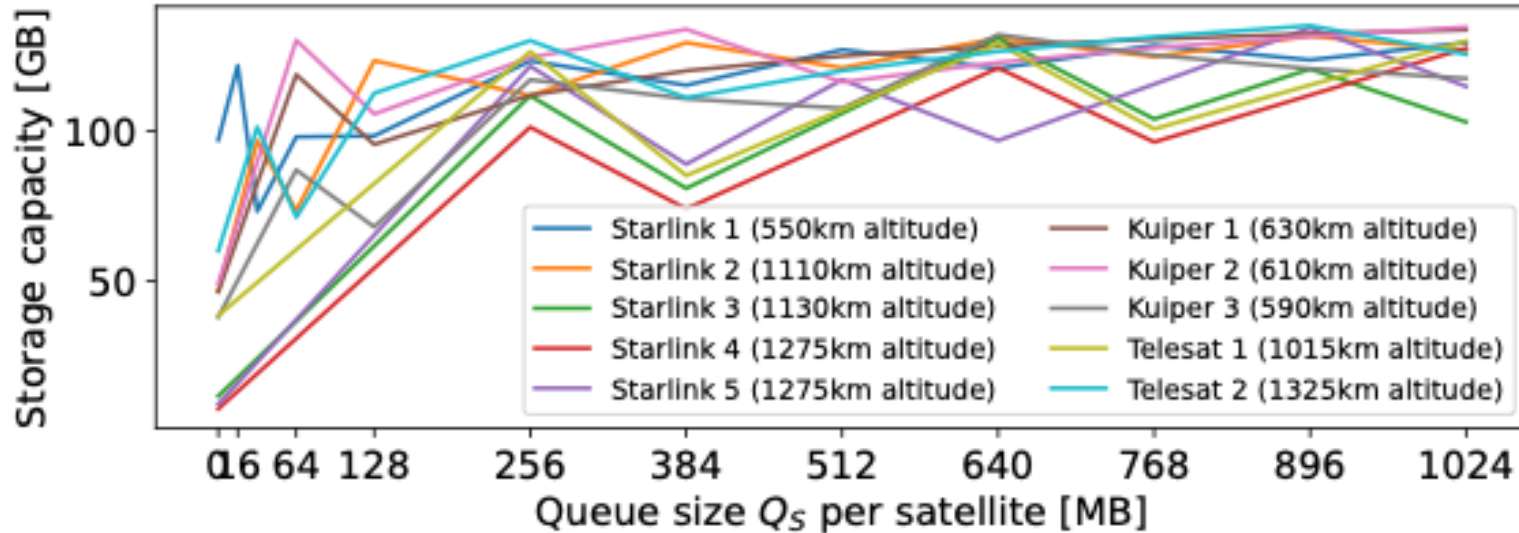
Periodicity, capacity = f (constellation, queue size)

Const.	Periodicity (s)							Storage capacity (GB)						
Q_s (MB)	–	16	32	64	128	192	256	–	16	32	64	128	192	256
Starlink 1	10.9	13.6	16.4	22.0	33.1	44.2	55.3	97	122	146	196	295	394	493
Starlink 2	5.3	8.1	10.8	16.3	27.2	38.2	49.2	47	72	97	147	247	347	447
Starlink 3	1.4	2.1	2.8	4.2	6.9	9.7	12.4	12	18	24	37	62	87	112
Starlink 4	1.0	1.6	2.3	3.5	6.1	8.6	11.2	7	13	19	31	54	78	101
Starlink 5	1.2	1.9	2.7	4.2	7.3	10.3	13.4	9	16	23	37	65	93	122
Kuiper 1	5.3	7.3	9.3	13.3	21.2	29.2	37.2	46	64	83	119	191	263	335
Kuiper 2	5.6	7.8	10.0	14.5	23.5	32.4	41.3	49	69	90	130	211	292	373
Kuiper 3	4.3	5.7	7.0	9.8	15.2	20.7	26.1	38	50	62	87	136	185	234
Telesat 1	4.4	5.0	5.7	6.9	9.5	12.0	14.5	39	44	50	60	82	104	126
Telesat 2	6.8	9.0	11.3	15.9	25.0	34.1	43.3	60	81	101	142	225	307	390
Iridium	1.1	1.2	1.3	1.5	2.0	2.5	3.0	8	9	10	12	17	21	25

Storage capacity: f (queue size, # replicas)



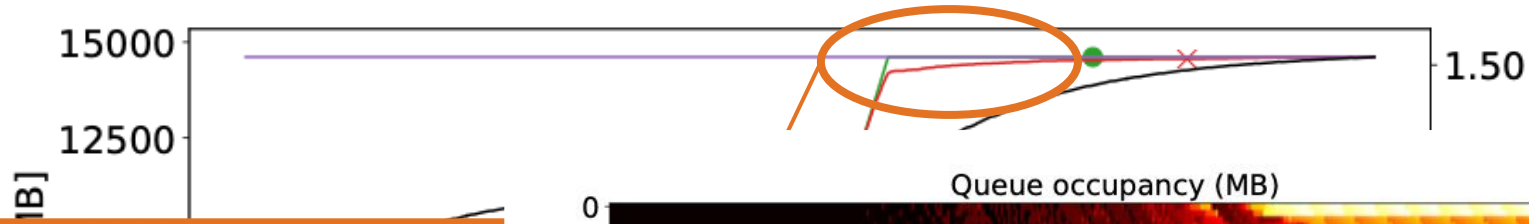
Storage capacity: f (queue size, constellation, periodicity)



$$t_a \leq 15s$$

	<i>shell</i>	<i>h (km)</i>	<i>Orbits</i>	<i>Sats/orbit</i>	<i>i</i>						
Starlink	S1	550	72	22	53°	Kuiper	K1	630	34	34	51.9°
	S2	1,110	32	50	53.8°		K2	610	36	36	42°
	S3	1,130	8	50	74°		K3	590	28	28	33°
	S4	1,275	5	75	81°	Telesat	T1	1,015	27	13	98.98°
	S5	1,325	6	75	70°		T2	1,325	40	33	50.88°

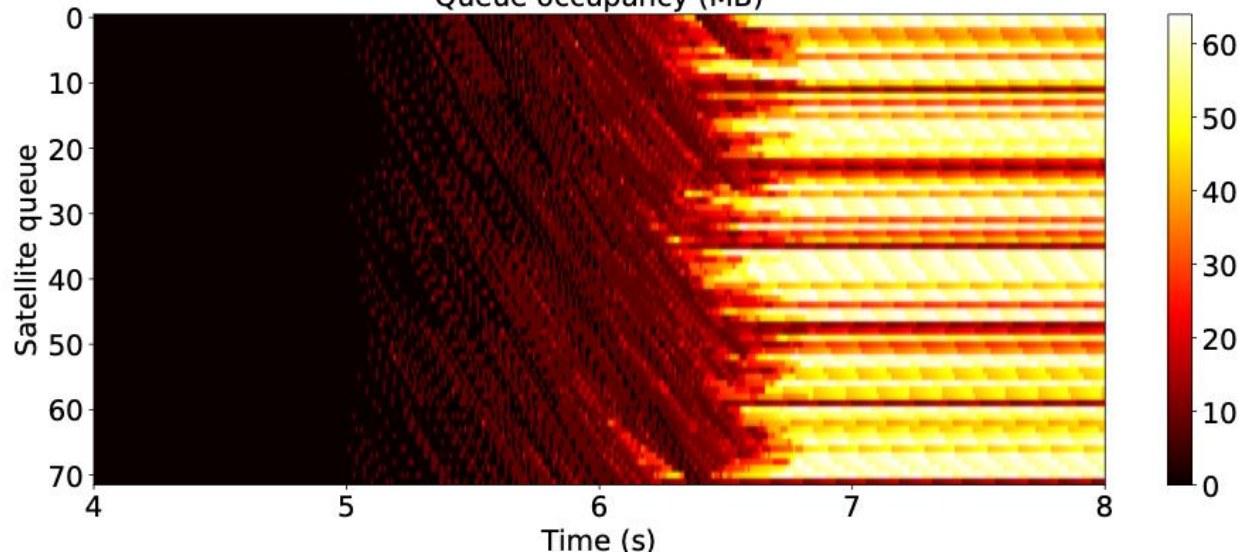
Simplified simulations [Iridium, $Q_S = 64$ MB, $S = 8$ MB]



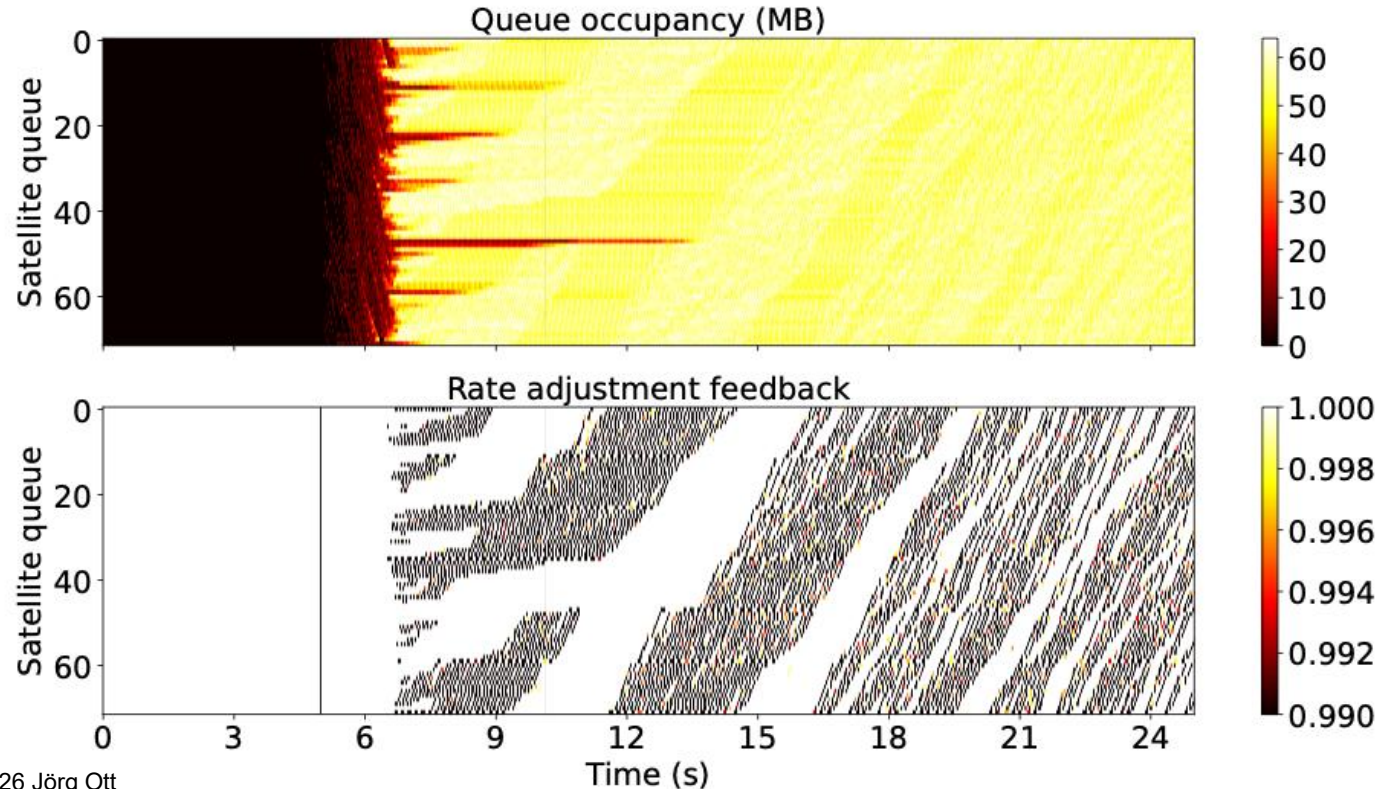
Growing load leads to standing

Uneven content injection distribution
may cause full queues in some
while others could take more

Need to rebalance using
careful feedback

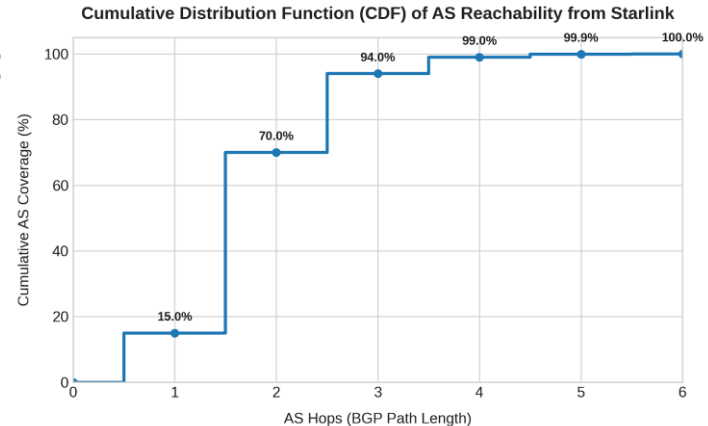


Simplified simulations [Iridium, $Q_S = 64$ MB, $S = 8$ MB] (2)



So what?

- Fun gedankenexperiment
- But 8–100 GB @ 1–11s access delay – well...
- How about multicast – possibly with “memory” for repair?
- A flavor of the traveling salesman problem – with many salesperson replicas
- Starlink peers with 600+ ASes and at 100+ IXPs:
not too bad a direct reach
- Could be used to e.g. spread transactions
for distributed consensus systems



Wrap-up

- “Storing” content in flight in-between satellites in an orbital plane
- Trade-offs
- Under construction
 - Packet-level simulations – including more satellite and ISL “realism”
 - Exploring suitable forwarding and replication strategies (also for multicasting)
- Up next
 - Adaptive algorithms
 - Operating using scavenged capacity only (think LEDBAT)
- Other thoughts? Worth it?