The Internet Protocol Suite for Deep Space Networking

draft-many-deepspace-ip-assessment

Marc Blanchet, marc.blanchet@viagenie.ca

To infinity and beyond!

This Talk is

- Not about Internet users/applications
  - Specialized space applications, not typical Internet applications
- Not about Near Earth (LEO, MEO, GEO)

- It is about space applications using the Internet Protocol suite.

- This project is benchmarking with Mars, but will be implemented first on Moon (« simpler » problem)

Space Ecosystem Context

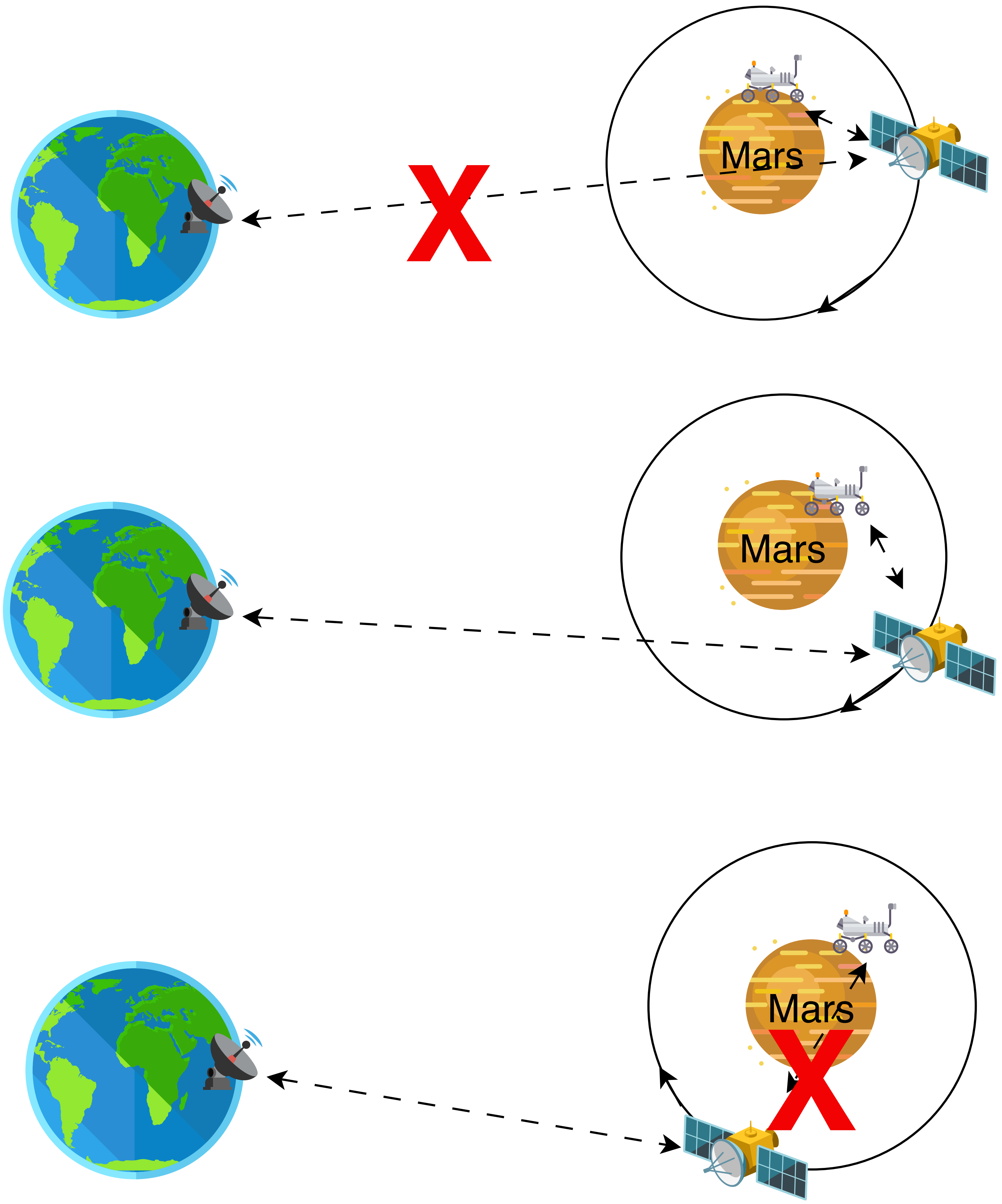
- Space agencies are moving into buying services, starting with Moon:
  - « rent a rover for 1 year »
  - Comms and networking providers
  - Private sector offering various services to non space agencies
- Moving from specialized equipment to RAD tested COTS or automotive grade
- Interoperability and governance becomes strategic and paramount
- Moon networking plan (IOAG) is to use 802.* + 3GPP on surface and orbit (therefore IP)

Deep Space Communications

- Delays:
  - Earth-Mars:
    - one-way delay: 5-20 minutes
    - Round trip time(RTT): 10-45 minutes
- Links:
  - Interrupted

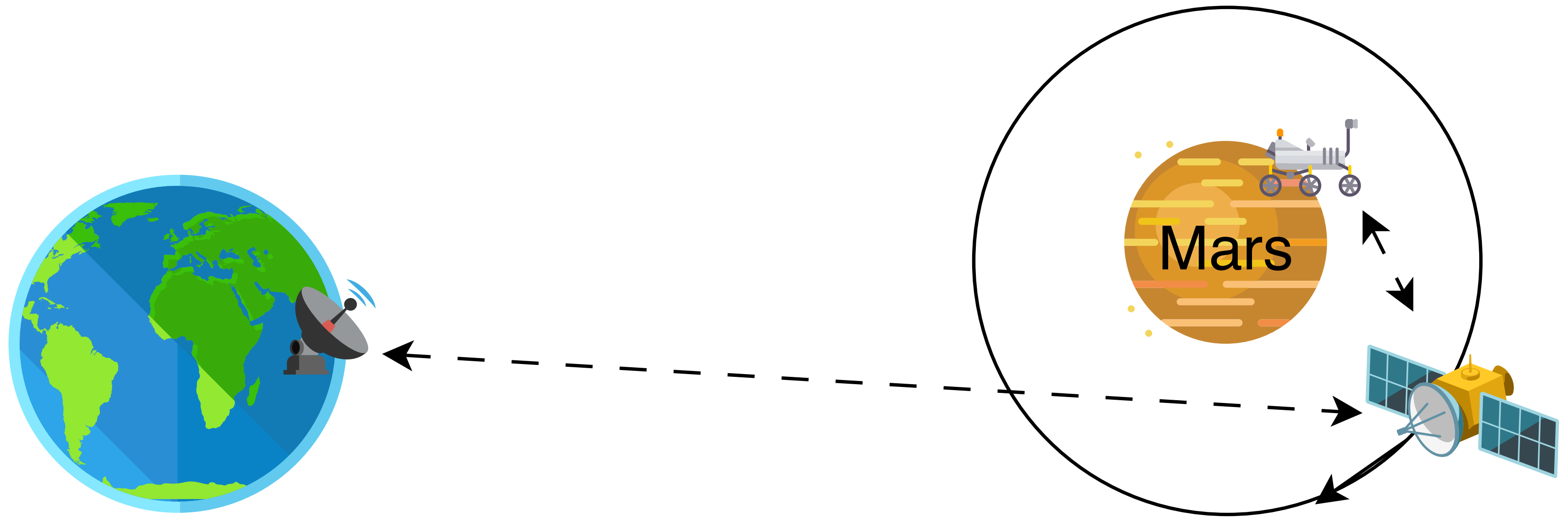
Deep Space Communications Are Interrupted

- Planets, Moons are orbiting
- Orbiters are orbiting ;-)
- Consequence:
    - No continuous communications: planned windows of communication between peers for each link
        - Earth-Mars relay
        - Mars relay - Mars rover/habitat/…
    - Relays/forwarders/routers need to store frames/packets until next hop becomes reachable.
        - Can be done at L2 (example: MRO for Mars) or L3 or ..



Calculating Deep Space RTT to Mars

- Earth-Mars round-trip time: 10-45 minutes
- Relay pass over: Mars Reconnaissance Orbiter (MRO) example: ~15 minutes of comms every 6 hours
- Smallest RTT: Line of sight and planets are near.
  - ~5 minutes one-way
  - Total: 10 minutes RTT
- Pretty large RTT scenario:
  - 22 minutes one-way delay forward
  - + 2 over passes (when the rover cannot respond within the current window, unlikely but…) = 12 hours
    - Assuming orbiter temporarily stores frames/packets
  - 22 minutes one-way delay return
  - Total ~13 hours RTT
- Worst case: solar mars conjunction: ~2 weeks no comms every ~2 years. In this case, RTT ~ 2 weeks
- A constellation of orbiters (a la Starlink) would cover the whole planet all the time, so very small RTT for comms around the planet.



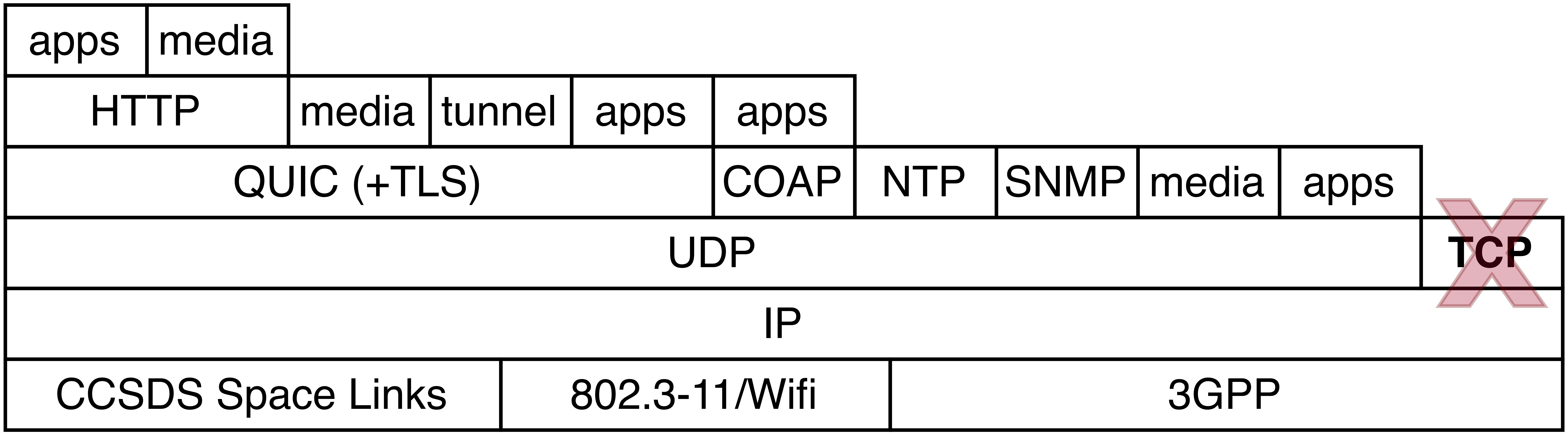
Earth-Moon

- Few seconds delay
  - Not very long, but long enough that typical Internet applications and transports will not be usable as is.
- Until a full constellation of orbiters (a la Starlink), then interruptions of communication will happen.

Can We Use our Internet Protocol Suite for Deep Space?

- RFC4838 (published in 2007) concludes "no". Very short summary:
  - Transport (TCP), protocols and applications are too chatty for delays and disruptions in space.
  - Consequence was the invention of a new Networking Stack: Bundle Protocol (RFC5050, RFC9171), based on a store and forward design, with its own new: transport, routing, naming, security, neighbor discovery, application API, network management, …
  - DTN: Delay (and Disruption)-tolerant Networking. Typically, DTN means Bundle Protocol.
    - This presentation is about DTN using IP.
- Reassessment of the use of the Internet Protocol Suite for Deep space. Short answer: "yes"!
  - https://deepspaceip.github.io
  - Documented in draft-many-deepspace-ip-assessment
  - Subject of this presentation

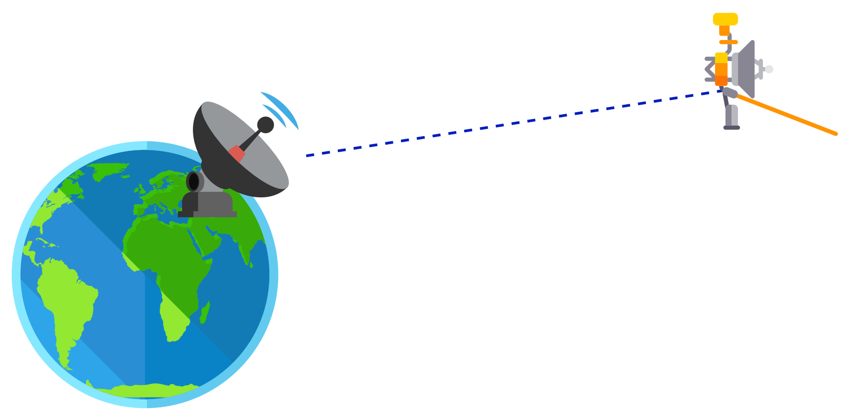
Deep Space IP Stack

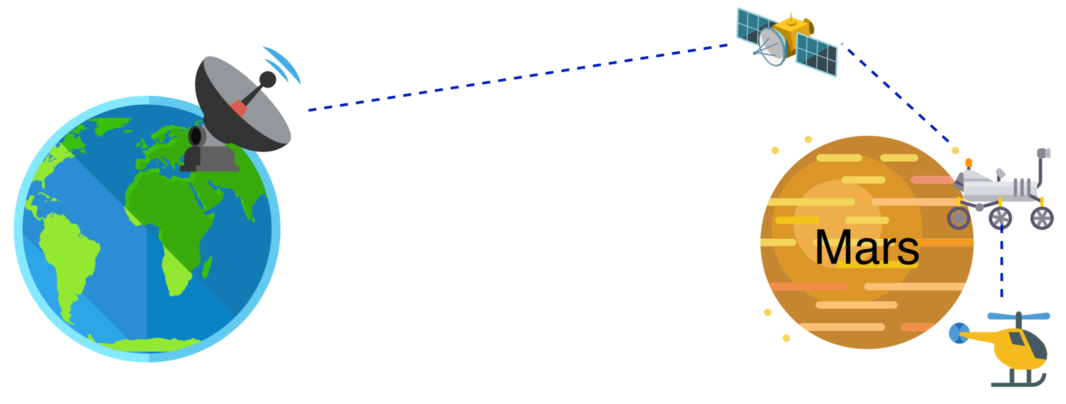


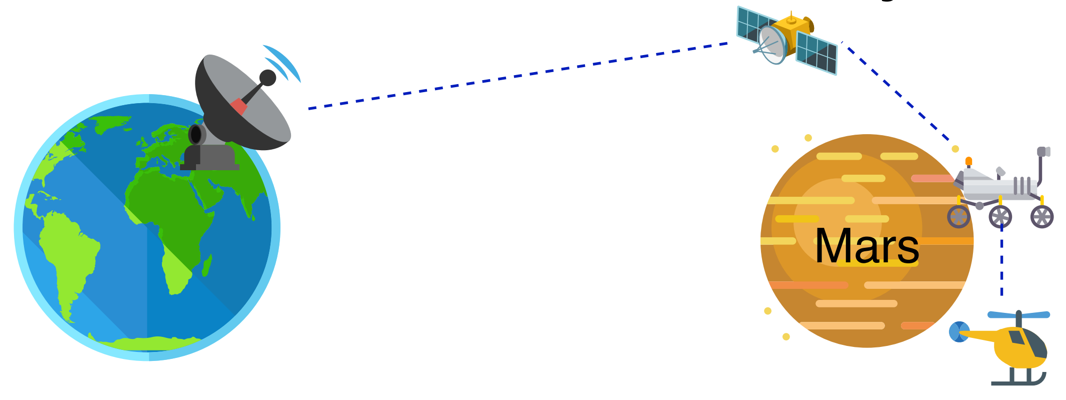
* The Interagency Operations Advisory Group(IOAG) identifies that 3GPP and IEEE 802.11 Link layers will be used on and around celestial body surface and orbits, while CCSDS Space Links will be used in deep space (and to/from surface).

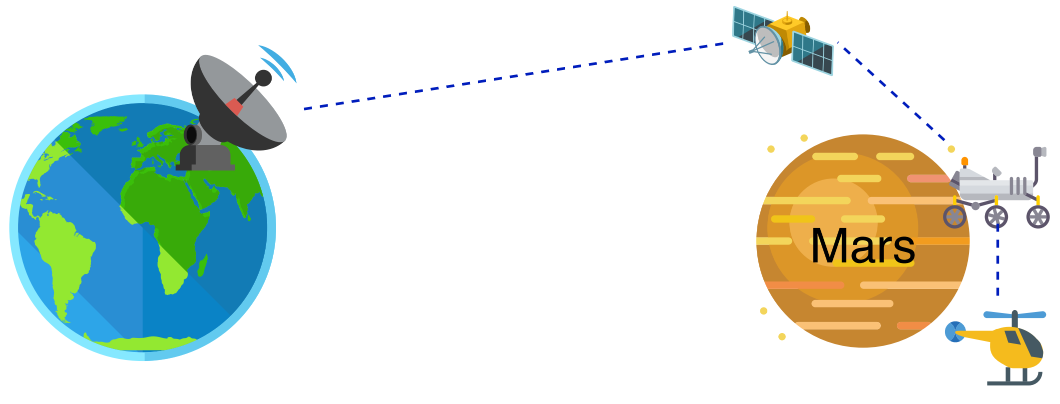
Deep Space Network

- High-level scenarios:
  - Earth - deep space - cruising spacecraft
  - Earth - deep space - celestial body network - spacecraft
- Deep Space may be designed as:
  - L2 network, where relays bridge CCSDS link frames, as today
    - IP is unknown to relays. IP is just an encapsulated payload.
    - Way simpler. Can be used today. Our assumption
  - L3 network, where relays are IP routers.
    - More scalable and reliable in the future.









IP and TCP-UDP in Deep Space

- IP ✅
  - has no notion of time, no reliability
  - can be encapsulated into CCSDS space links
  - since IP over space links are point to point, header compression can be used. (Makes a whole IP-UDP headers into a single byte)
  - Forwarders with interrupted links have to temporarily store packets. Our Linux prototype is 200 lines of C code.
- TCP: does not work in space because chattiness and timers TCP
- UDP has no notion of time, no reliability ✅

QUIC in one slide

- Over UDP, Reliable transport, e2e, single handshake, 0RTT, user-space
- Mandatory security (TLS)
- Single connection is a pipe of streams, bidirectional, may be long-lived
- Carry HTTP and other application protocols, including tunnelling
- Useful characteristics for deep space ✅

QUIC for Deep Space

- But QUIC by default assumes short delays, interactive communications and does congestion control
- **By setting various QUIC parameters to appropriate values (typically large), such as initial_rtt, max_idle_timeout, window_size, and by simplifying the congestion control, QUIC works just fine in space.** ✅
- A QUIC connection to/from a spacecraft can be:
    - Long-lived (hours, days, weeks, months)
    - Pre-established while spacecraft has not left Earth
- QUIC may use proxies, which could be used at space edges, where access policies, buffering, QoS, … is applied.
- With appropriate setting of the QUIC stack by the application, and no modification of protocol or the QUIC stack itself, we were able to …

Sent An HTTP Request to Voyager!

- 24 hours (84600 seconds) delay each way; 48 hours RTT
- Using configured QUIC as transport

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	1.1.1.1	88.88.88.88	QUIC	1228	Initial, DCID=a68c4660e2db0b33521d8ae1d3e3cfb5c17abbb8, PKN: 0, CRYPTO, PADDING
2	86400.000000	88.88.88.88	1.1.1.1	QUIC	1228	Handshake, PKN: 0, CRYPTO
3	172800.000000	1.1.1.1	88.88.88.88	QUIC	1228	Handshake, PKN: 0, ACK_ECN, CRYPTO
4	172800.000000	1.1.1.1	88.88.88.88	QUIC	64	Protected Payload (KP0), PKN: 1, STREAM(0)
5	259200.000000	88.88.88.88	1.1.1.1	QUIC	427	Protected Payload (KP0), PKN: 1, DONE, AF, CRYPTO
6	259200.000000	88.88.88.88	1.1.1.1	QUIC	1074	Protected Payload (KP0), PKN: 2, STREAM(0)
7	345600.000000	1.1.1.1	88.88.88.88	QUIC	54	Protected Payload (KP0), PKN: 2, ACK_ECN
8	345600.000000	1.1.1.1	88.88.88.88	QUIC	57	Protected Payload (KP0), PKN: 3, ACK_ECN, CC
9	432000.000000	88.88.88.88	1.1.1.1	QUIC	58	Protected Payload (KP0), PKN: 3, ACK_ECN, CC

- 1-2: client-server initial connection handshake. Crypto set.
- 3-4. GET HTTP REQUEST
- 5-6. HTTP RESPONSE
- 7-8-9. Client connection close

- HTTP over QUIC in deep space: ✅

10 days RTT!!!

- 5 days delay one way (432000s), 10 days RTT (864000s)
- If connection is already established and is not closed, then only one RTT for request and response

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.132.239.239	10.132.239.121	QUIC	1242	Initial, DCID=31e20cceeacc6ddb3903704db5c9be687df54441, SCID=05e7e0f8a63866b8, PK…
2	432000.0…	10.132.239.121	10.132.239.239	QUIC	1380	Handshake, DCID=05e7e0f8a63866b8, SCID=c2d1484de334b668, PKN: 0, CRYPTO
3	864000.0…	10.132.239.239	10.132.239.121	QUIC	1242	Handshake, DCID=c2d1484de334b668, SCID=05e7e0f8a63866b8, PKN: 0, ACK_ECN, CRYPTO
4	864000.0…	10.132.239.239	10.132.239.121	QUIC	200	Protected Payload (KP0), DCID=c2d1484de334b668, PKN: 1, NCI, NCI, NCI, NCI, STREA…
5	1296000.…	10.132.239.121	10.132.239.239	QUIC	689	Protected Payload (KP0), DCID=9e463c45aa076440
6	1728000.…	10.132.239.239	10.132.239.121	QUIC	77	Protected Payload (KP0), DCID=c2d1484de334b668, PKN: 2, ACK_ECN
7	1728000.…	10.132.239.239	10.132.239.121	QUIC	84	Protected Payload (KP0), DCID=c2d1484de334b668, PKN: 3, ACK_ECN, CC
8	2160000.…	10.132.239.121	10.132.239.239	QUIC	81	Protected Payload (KP0), DCID=9e463c45aa076440, PKN: 4, ACK_ECN, CC

HTTP and Applications for Deep Space

- HTTP does not have any notion of time, by itself.
  - If some time-related HTTP headers are used, such as Cache-Control and Expires, then proper value must be set.
  - The server and client typically have timeouts. Set value properly
    - Examples: curl -m; nginx *_timeout config
- Typical web browsers with GUI should work, but may not be the best tool for looong delays
  - Disable timeout or set timeout value to large value
  - Make sure the browser app and window stays there for the whole time.
- HTTPS when used over QUIC: TLS is done at the QUIC layer
- One can design a space application using HTTP, REST API, Javascript, …
    - Have the right design:
      - Asynchronous
      - Local references with local (pre-)caching/preloading of assets
      - Timers set properly
    - Consider various Web optimizations:
      - WASM
      - HPACK

QUIC Testing: some results

Network conditions simulations: delay, reorder, duplicates

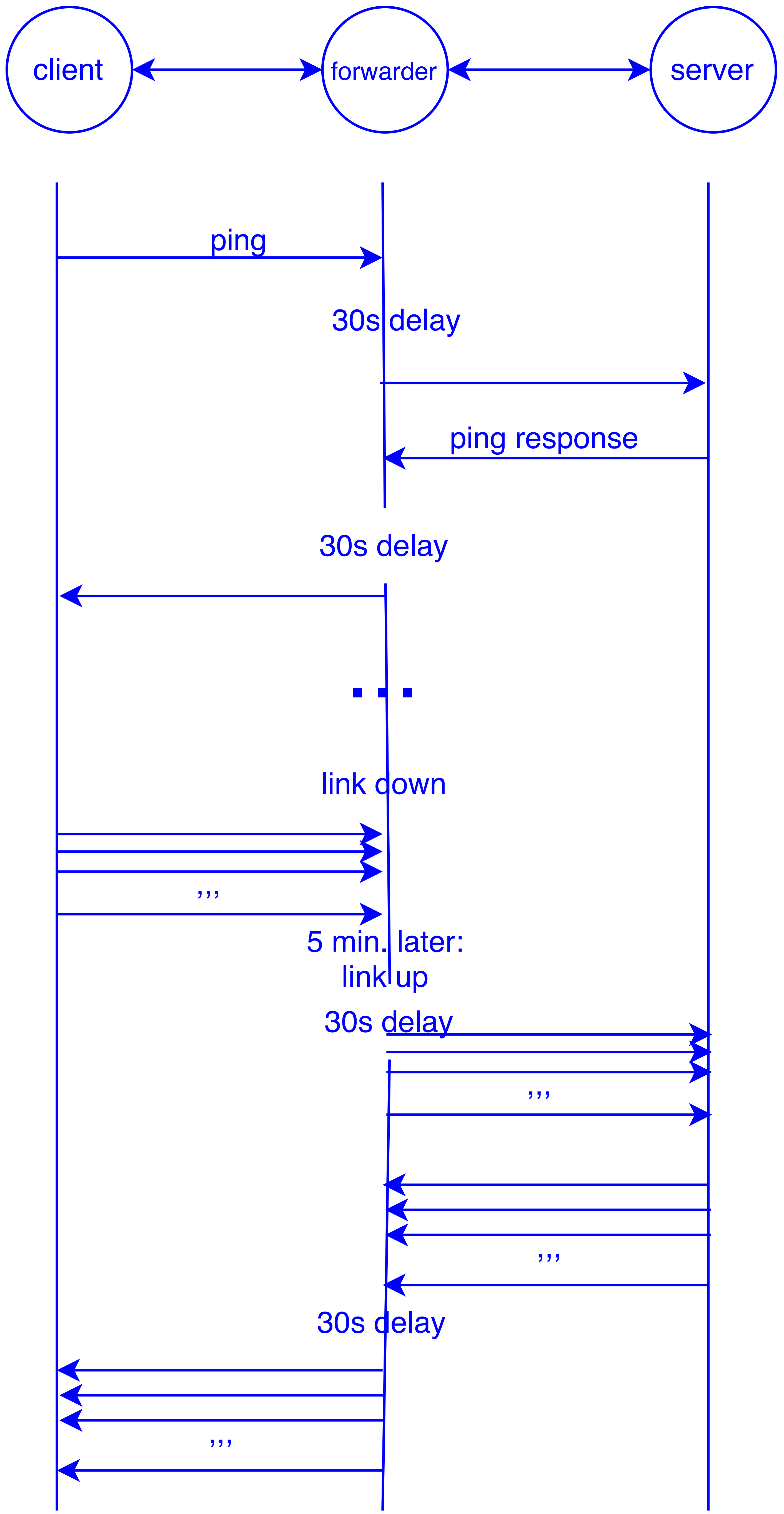
Delay 1 day, loss 5%, repeat 10, QUIC-large values

- One way 1 day delay, packet loss 5%, 10 times repeat HTTP request and response, QUIC stack set with large values (space compatible)
- Total time: is the same as without packet loss, since loss was recovered using the next packets)
- Client packets sent: 22 (3087 bytes)
- Server packets sent: 20 (12313 bytes)
  - Server packets dropped: 2
  - (by the network simulation)
- Conclusion: QUIC recovered successfully and all data were pro ✅ sent reliably

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	1.1.1.1	88.88.88.88	QUIC	1228	Initial, DCID=22a3467b8c1180a3eeba67d7dfc1fe9b8e9111ff, PKN: 0, CRYPTO, PADDING
2	86400.000000	88.88.88.88	1.1.1.1	QUIC	1228	Handshake, PKN: 0, CRYPTO
3	172800.000000	1.1.1.1	88.88.88.88	QUIC	1228	Handshake, PKN: 0, ACK_ECN, CRYPTO
4	172800.000000	1.1.1.1	88.88.88.88	QUIC	64	Protected Payload (KP0), PKN: 1, STREAM(0)
5	259200.000000	88.88.88.88	1.1.1.1	QUIC	427	Protected Payload (KP0), PKN: 1, DONE, AF, CRYPTO
6	259200.000000	88.88.88.88	1.1.1.1	QUIC	1074	Protected Payload (KP0), PKN: 2, STREAM(0)
7	345600.000000	1.1.1.1	88.88.88.88	QUIC	54	Protected Payload (KP0), PKN: 2, ACK_ECN
8	345600.000000	1.1.1.1	88.88.88.88	QUIC	64	Protected Payload (KP0), PKN: 3, STREAM(4)
9	432000.000000	88.88.88.88	1.1.1.1	QUIC	54	Protected Payload (KP0), PKN: 3, ACK_ECN
10	432000.000000	88.88.88.88	1.1.1.1	QUIC	1074	Protected Payload (KP0), PKN: 4, STREAM(4)
11	518400.000000	1.1.1.1	88.88.88.88	QUIC	64	Protected Payload (KP0), PKN: 4, STREAM(8)
12	604800.000000	88.88.88.88	1.1.1.1	QUIC	1074	Protected Payload (KP0), PKN: 5, STREAM(8)
13	691200.000000	1.1.1.1	88.88.88.88	QUIC	61	Protected Payload (KP0), PKN: 5, ACK_ECN
14	691200.000000	1.1.1.1	88.88.88.88	QUIC	64	Protected Payload (KP0), PKN: 6, STREAM(12)
…						
33	1555200.000000	1.1.1.1	88.88.88.88	QUIC	61	Protected Payload (KP0), PKN: 15, ACK_ECN
34	1555200.000000	1.1.1.1	88.88.88.88	QUIC	64	Protected Payload (KP0), PKN: 16, STREAM(32)
35	1641600.000000	88.88.88.88	1.1.1.1	QUIC	61	Protected Payload (KP0), PKN: 16, ACK_ECN
36	1641600.000000	88.88.88.88	1.1.1.1	QUIC	1074	Protected Payload (KP0), PKN: 17, STREAM(32)
37	1728000.000000	1.1.1.1	88.88.88.88	QUIC	61	Protected Payload (KP0), PKN: 17, ACK_ECN
38	1728000.000000	1.1.1.1	88.88.88.88	QUIC	64	Protected Payload (KP0), PKN: 18, STREAM(36)
39	1814400.000000	88.88.88.88	1.1.1.1	QUIC	61	Protected Payload (KP0), PKN: 18, ACK_ECN
40	1814400.000000	88.88.88.88	1.1.1.1	QUIC	1074	Protected Payload (KP0), PKN: 19, STREAM(36)
41	1900800.000000	1.1.1.1	88.88.88.88	QUIC	61	Protected Payload (KP0), PKN: 19, ACK_ECN
42	1900800.000000	1.1.1.1	88.88.88.88	QUIC	57	Protected Payload (KP0), PKN: 20, ACK_ECN, CC
43	1987200.000000	88.88.88.88	1.1.1.1	QUIC	61	Protected Payload (KP0), PKN: 20, ACK_ECN
44	1987200.000000	88.88.88.88	1.1.1.1	QUIC	58	Protected Payload (KP0), PKN: 21, ACK_ECN, CC

Link interruptions

- Three nodes network: Client - link - Forwarder - link - Server
- Ping
- 30 seconds delay on each outgoing interface of the forwarder => 60s RTT
- Link down for 5 minutes
  - During link down, forwarding node stores packets
- when link up, data is de-stored and forwarded (still with 60s RTT)



Link Interruptions

- ping -v -c 400 -W 100000 fc00:1::3
- PING fc00:1::3 (fc00:1::3) 56 data bytes
- 64 bytes from fc00:1::3: icmp_seq=1 ident=12259 ttl=62 time=60015 ms
- 64 bytes from fc00:1::3: icmp_seq=2 ident=12259 ttl=62 time=60004 ms
- 64 bytes from fc00:1::3: icmp_seq=3 ident=12259 ttl=62 time=60004 ms
- ...
- 64 bytes from fc00:1::3: icmp_seq=35 ident=12259 ttl=62 time=60004 ms
- 64 bytes from fc00:1::3: icmp_seq=36 ident=12259 ttl=62 time=60004 ms
- **64 bytes from fc00:1::3: icmp_seq=37 ident=12259 ttl=62 time=360184 ms**
- 64 bytes from fc00:1::3: icmp_seq=38 ident=12259 ttl=62 time=359160 ms
- 64 bytes from fc00:1::3: icmp_seq=39 ident=12259 ttl=62 time=358137 ms
- 64 bytes from fc00:1::3: icmp_seq=40 ident=12259 ttl=62 time=357113 ms
- ...
- 64 bytes from fc00:1::3: icmp_seq=317 ident=12259 ttl=62 time=74110 ms
- 64 bytes from fc00:1::3: icmp_seq=318 ident=12259 ttl=62 time=73087 ms
- 64 bytes from fc00:1::3: icmp_seq=319 ident=12259 ttl=62 time=72065 ms
- ...
- 64 bytes from fc00:1::3: icmp_seq=398 ident=12259 ttl=62 time=60004 ms
- 64 bytes from fc00:1::3: icmp_seq=399 ident=12259 ttl=62 time=60004 ms
- 64 bytes from fc00:1::3: icmp_seq=400 ident=12259 ttl=62 time=60004 ms
- --- fc00:1::3 ping statistics ---
- 400 packets transmitted, 400 received, **0% packet loss**, time 407896ms
- rtt min/avg/max/mdev = 60003.980/170727.305/360183.665/99770.604 ms, pipe 353

- 60 seconds RTT
- 5 minutes interruption
- link back
- Implemented forwarding policy is FIFO, so the « oldest» is forwarded first.
- After all stored packets forwarded, then the packets are forwarded directly

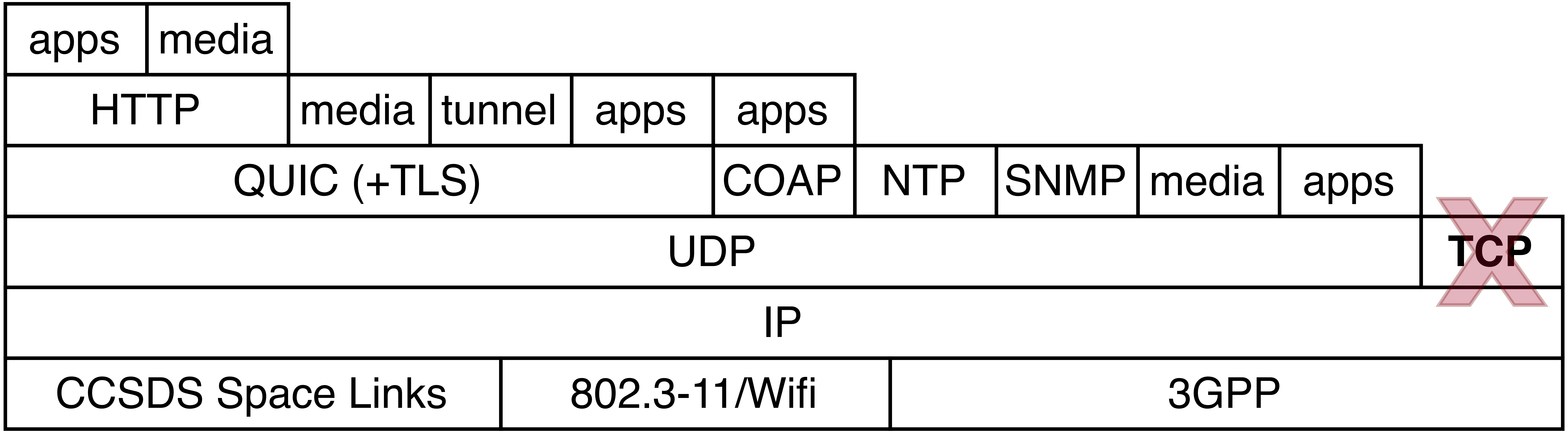
UDP-Based Applications

- With properly set timeouts, and if protocols are « compatible » (aka not too chatty), one can also use UDP-based applications:
    - Streaming (RTP/RTSP) or over QUIC or over HTTP-QUIC
    - SIP/RTP. (E.T. Phone Home!)
    - Over QUIC (MOQ)
- CoAP

Network Services

- Network Management:
  - SNMP (over UDP) with MIBs
  - NETCONF (over QUIC-UDP, or RESTCONF) with YANG
  - SSH (does not seem to work in space)
- Time distribution with NTP: seems to work!
- DNS:
  - Resolving a name through a deep space link such as Mars to Earth does not make a lot of sense.
  - Appropriate pre-caching and proper deployment techniques are necessary so that name resolution and security(aka DNSSEC) is local to the planetary body network
  - No need for protocol changes.
  - DNSSEC key and RR lifetimes need to be carefully set for enough long values
  - Naming hierarchy has an impact on how to deploy but any level is possible.
  - See draft-many-dnsop-isolated-networks

Deep Space IP Stack



* The Interagency Operations Advisory Group(IOAG) identifies that 3GPP and IEEE 802.11 Link layers will be used on and around celestial body surface and orbits, while CCSDS Space Links will be used in deep space (and to/from surface).

Summary

- By:
  - Temporarily storing IP packets in forwarding nodes until links are back up
    - Only for IP forwarders involved in deep space that are subject to link up/down
  - Setting appropriate QUIC parameters
    - Or using UDP-based protocols and applications
  - Properly design applications to be asynchronous and have large timers
- The Internet Protocol Suite can be used end to end in deep space.
  - Significantly decrease costs, decrease risks, increase security, increase access to knowledge and expertise, hardware available today for very high speed processing
- Results from simulations in a testbed have confirmed that deep space IP is working.
- Obviously more work needed

Acknowledgements and Further Info

- Acknowledgements:
  - IP forwarding storage implementation: Jean-Philippe Dionne
  - QUIC: Christian Huitema(picoquic, IETF), Martin Thompson(neqo, IETF), Benjamin Saunders(quinn), François Michel(UCL), Maxime Piraud(UCL), Martin Duke(Google)
  - QUIC workbench: Adolfo Ochagavia
  - Routing: Dean Bogdanovic, Tony Li
  - COAP: Carlos GomezMontenegro
  - TAPS: Émile Stephan
  - DNS: Warren Kumary, Mark Andrews,
  - Discussions and argumentation: Vint Cerf, WIDE project, KDDI, James Schier(NASA), Felix Flentge(ESA), Juan Fraire(INRIA), …
  - Specifications reviews: too many to list.

- Further info:
  - IETF deepspace mailing list (mailto:deepspace@ietf.org)
  - https://deepspaceip.github.io
  - draft-many-deepspace-ip-assessment
  - Marc Blanchet, marc.blanchet@viagenie.ca

Deepspace IP side meeting this week:

Wednesday 8h00-9h30 Tennyson