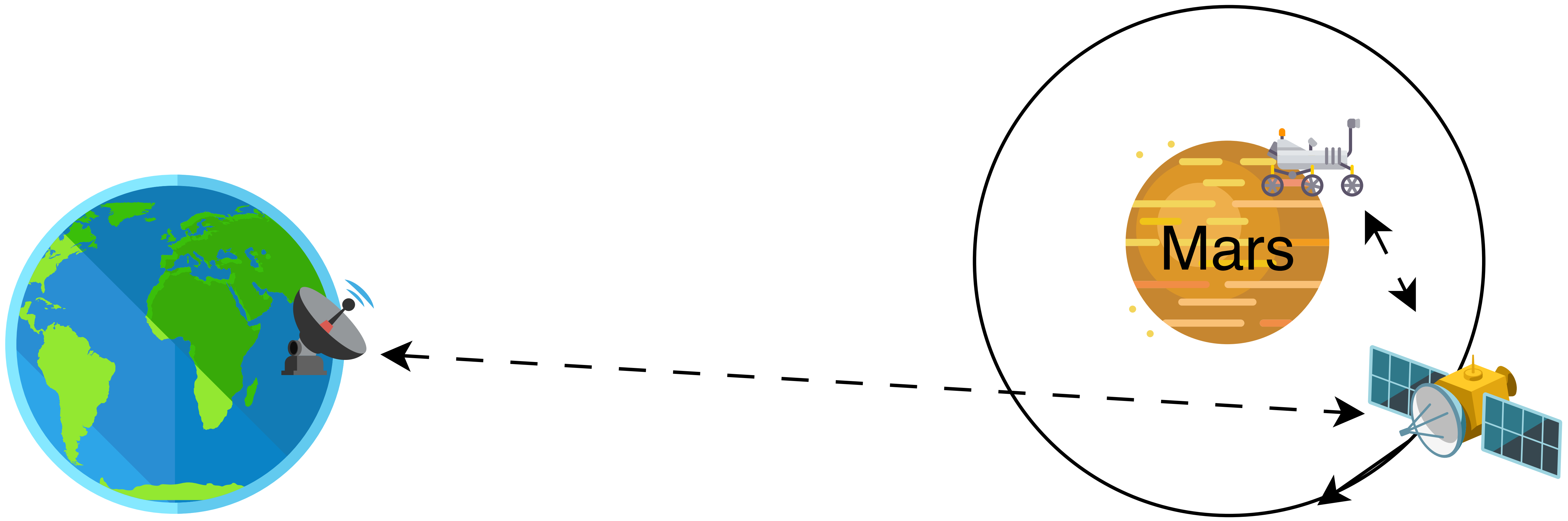


Using the Internet Protocol Suite for Deep Space Networking

About IETF TIPTOP working group

Industry Network Technology Council Seminar, June 19th 2025

Marc Blanchet, Viagenie, marc.blanchet@viagenie.ca

Me

- Internet engineer for 30+ years
- Developed protocols (wrote 17 RFC, wg chair of many IETF wg, IAB member)
- President of Viagenie, consulting for providers, large entreprises and manufacturers
- Space related:
    - Involved in space comm/networking since early 2000.
    - IETF delay tolerant networking(dtn) wg co-chair for ~10 years.
    - Member of Interplanetary Network SIG(IPNSIG) Architecture WG and Projects WG
    - Lead of the IOAG LunaNet networking governance working group
    - Designed, implemented and managed the Space Assigned Number Authority(SANA)
    - Instigated the Deep space IP initiative and proposed the IETF tiptop (Taking IP to Other Planets) working group, where I'm technical advisor and delegate

IETF TIPTOP WG

TIPTOP = Taking IP To Other Planets

TIPTOP Charter

- In a Nutshell:
  - Given the delays and disruptions involved in space communications, engineer IP-based internetworking and support IP applications end-to-end.
  - IP-based protocols may be profiled, but no new protocols are in scope.
  - Target scenarios: Moon and Mars, and deep space
  - Out of scope: Bundle Protocol(BP), LEO/MEO/GEO scenarios
  - Collaboration/coordination with space ecosystem and within IETF
- Work items:
  - key characteristics, use cases, and requirements
  - differences that apply to the IP architecture
  - recommendations for existing transport protocols (+ their security protocols), starting with QUIC.
  - considerations for DNS

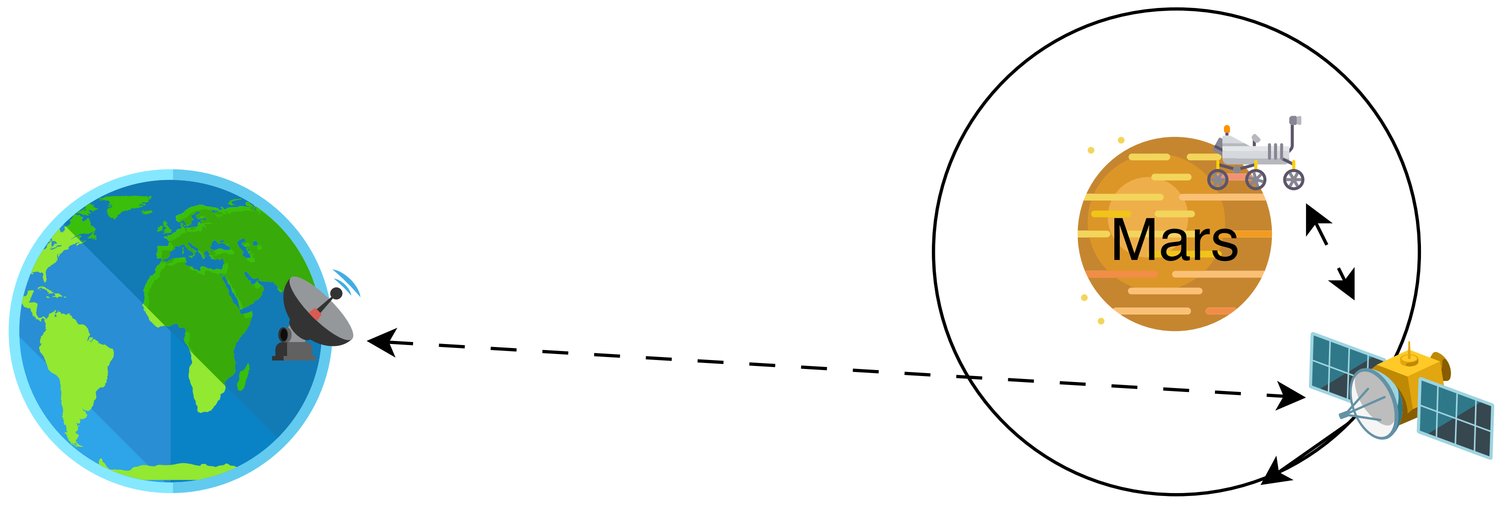
But why are we doing this?

Networking in Space?

- First order question: do we need a network in deep space?
    - Multiple providers, multiple users, various assets, shared links, full link utilization
        - Compared to the last ~50 years: links usage is planned and dedicated to single mission per comm window
    - we need to build a Layer 3 network, end to end!
- Choices for the network layer (above CCSDS link layer):
    - Bundle Protocol(BP)
    - Internet Protocol(IP)
        - this presentation
- Notes:
    - terminology: Deep space includes Moon in this presentation
    - This content is my personal contribution and opinion. I do not talk on behalf of IETF or TIPTOP wg

Main Challenges for Networking in Deep Space

- Long delays (one-way delay: Moon: ~2s; Mars: ~4-22min)
  - "Simpler to fix": expect to take longer... adjust timers.
  - Cannot expect immediate reaction to events



- Intermittent communications
  - "More complicated": from the end to end point of view, the round-trip time (RTT) is large, but more importantly very very variable, with jumps due to orbiters going off line of sight
    - A mechanism assuming a relative stable RTT will just fail.
      - RTT is not stable on Internet: congestion happens, then recovery kicks in. But immediate/fast reaction is possible. Not in space
- DTN: Delay-Tolerant Networking. Then became Delay and Disruption Tolerant Networking.
- DTN using BP or IP. Latter is the subject of this presentation

Lunar and Mars Comm/Network Architecture

- From IOAG* architecture for Moon and Mars (and NASA LunaNet, ESA Moonlight,...):
  - on celestial body surface and around: 5-6G (3GPP) network and Wifi, creating an IP network
  - Orbiters around celestial bodies, carrying IP and BP
  - CCSDS Deep space links, carrying IP and BP
- LunaNet Interoperability Specification(LNIS): IP and BP as network layers
- Therefore we have the following "regions":
  - Earth (running IP)
  - Deep space links using CCSDS link layers carry IP or BP
  - Moon/Mars surface network running IP over 5-6G/Wifi
  - Orbiters (IP and BP forwarding)
  - On-Board Spacecraft: internal IP network

* The Future Lunar Communications Architecture, Report of the Interagency Operations Advisory Group", January 2022, <https://www.ioag.org/Public%20Documents/Lunar%20communications%20architecture%20study%20report%20FINAL%20v1.3.pdf>.

The Future Mars Communications Architecture, Report of the Interagency Operations Advisory Group", February 2022, <https://www.ioag.org/Public%20Documents/MBC%20architecture%20report%20final%20version%20PDF.pdf>.

IP then everywhere?

- IP as the single network layer makes everything simpler and way more cost effective
  - No complex gateways between protocols: much less fragile and brittle
  - For an application, a single network layer, not two (vastly different) network layers to support
  - Ability to view, manage the whole network from any vantage point
  - Vast knowledge available for all layers: network, applications, security, management, ...
  - Vast number of implementations, open-source or vendor-based, for all layers
  - Used by IoT: low bandwidth, small memory, slow cpu, energy efficient
  - Security available at all levels, reviewed and scrutinized
  - Very fast forwarding. Optical links!
  - With QUIC transport:
    - end to end reliability
    - end to end security
    - Mobility

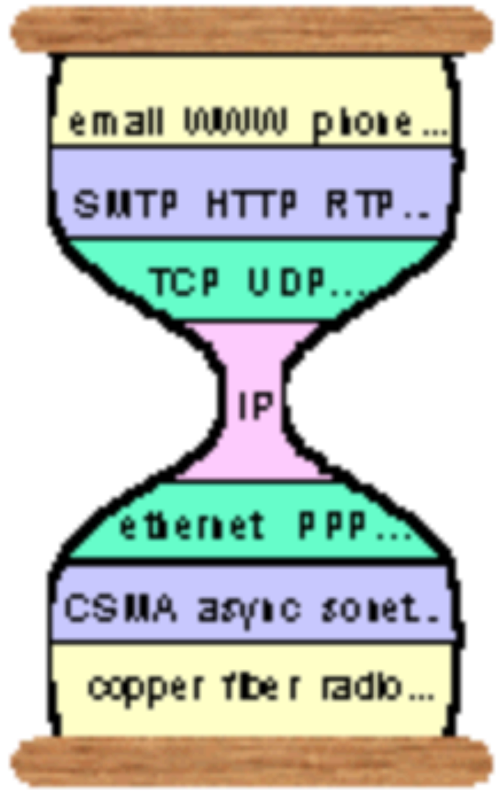
So Why (not) BP?

- Study of JPL+Vint Cerf early 2000 (RFC4838). At that time:
    - TCP and UDP were the dominant transports
    - Chatty protocols
    - "Low" usage of IoT
    - Concluded IP suite not usable in space. Need something different. Therefore design new networking stack: BP.
        - Many issues still (my own opinion), one of which is it does not have any transport semantics. ...
- Since then:
    - A modern, agile, configurable, efficient, user-space, mobile, all-encrypted-e2e transport: QUIC
        - unlimited requests/responses/streams/applications within a possibly long-lasting connection
    - IoT: low bandwidth, small memory, slow cpu, energy efficiency: pretty much same requirements as space
- Did a reassessment of IP in 2022
- First question I'm getting: How do they(IP vs BP) compare? (Caveat: Need a whole presentation by itself)

IP Hour Glass vs BP

Why the Hourglass Architecture?

- Why an internet layer?
  - make a bigger network
  - global addressing
  - virtualize network to isolate end-to-end protocols from network details/changes
- Why a *single* internet protocol?
  - maximize interoperability
  - minimize number of service interfaces
- Why a *narrow* internet protocol?
  - assumes least common network functionality to maximize number of usable networks



* me adding:
  * Layered architecture: each layer does its jobs (well)
  * intermediate nodes can forward fast by having limited functionality to process/implement

* Steve Deering, IETF 51 Plenary, Aug 2001

- BP suite does not have:
  - Transport
    - therefore: no congestion control, no flow control, no reordering and duplication recovery. No e2e loss recovery (only hop-by-hop)
    - Nor transport security (aka like TLS)
- BP suite has very few application protocols/applications
- Developing a BP application essentially requires to (re-)engineer transport (and application protocol) within the application, not standardized.
  - No standardized API (like socket API) to call BP: each BP implementation has its own proprietary API
- BP engineering work has been:
  - To make the BP networking layer "fatter" by adding new features into it.
    - One recent example: bundle sequence numbers in extension blocks (headers in IP parlance)
  - (For some) To try to retrofit IP application protocols (RTP, HTTP,...) into BP payloads.
    - Still need to adapt them, if needed, for long delays and interruptions
    - Most IP application protocols assume/require reliable transport. BP does not provide any.
  - Providing security at the network layer (aka like IPsec), therefore not known to/guaranteed for the application layer
- BP nodes have permanent ids, wherever they are.
  - No possible way to aggregate routes
  - Mobility means advertising to the whole network or at least all your current peers, wherever they are.

What needs to be done on IP suite for Deep Space?

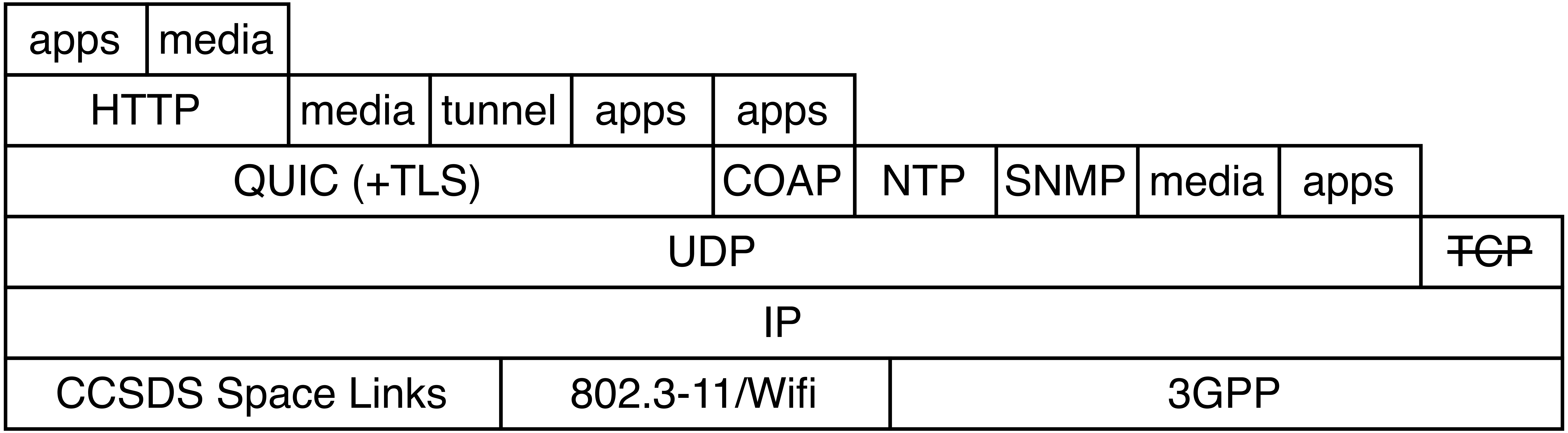
- IP and UDP (and HTTP) have no notion of time. Nothing to do.

A. For forwarding devices (like orbiters or space edge) facing intermittent links:
  - Store packets (instead of dropping them) when no route to destination
    - Not difficult to do: our implementation: ~500 lines of C (or Rust) code.
    - Not needed for:
      - surface or 5-6G/Wifi forwarders/routers
      - Layer 2 orbiters/gateways (if they don't know about IP, just forward based on CCSDS link layers, like Mars orbiters currently)
      - Non-forwarding end nodes

B. To deliver end to end reliability, configure transport (QUIC) based on a deep space profile
  - Right set of values for timers
  - Intermittence is not directly seen by transport: it is just long and variable delays
  - Do not rely on typical RTT for internal calculations

C. Applications/Tools/...: asynchronous design, adjust timers appropriately

Deep Space IP protocol Stack

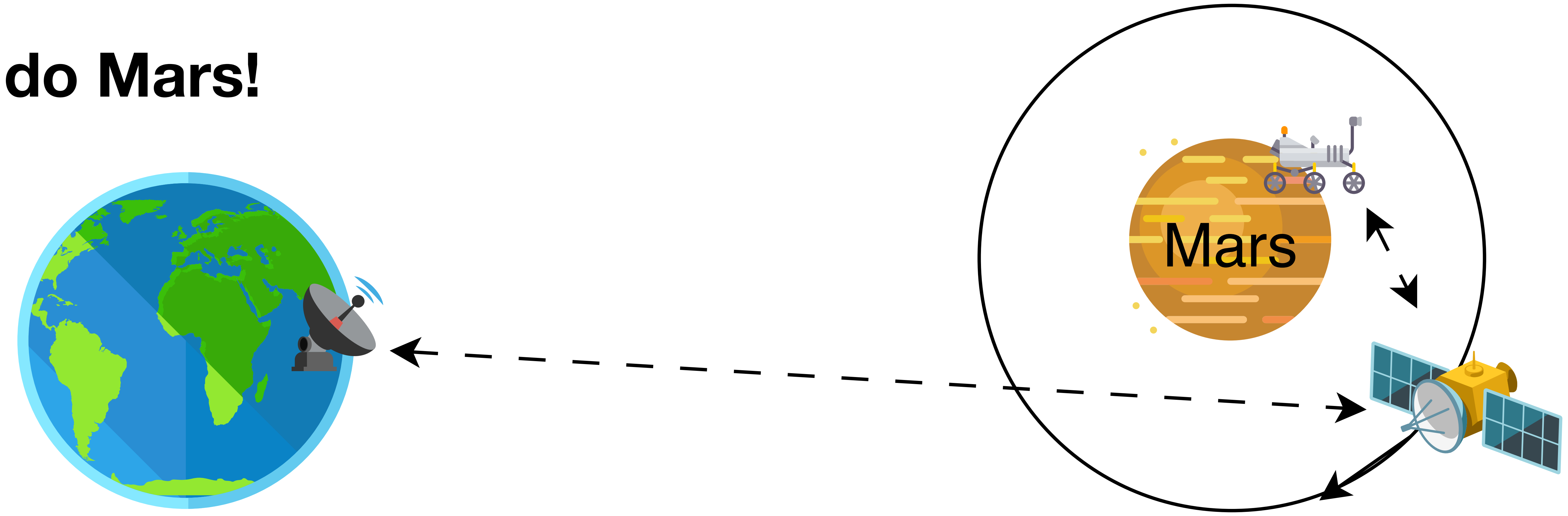


- CoAP: optimized HTTP-like semantics for IoT

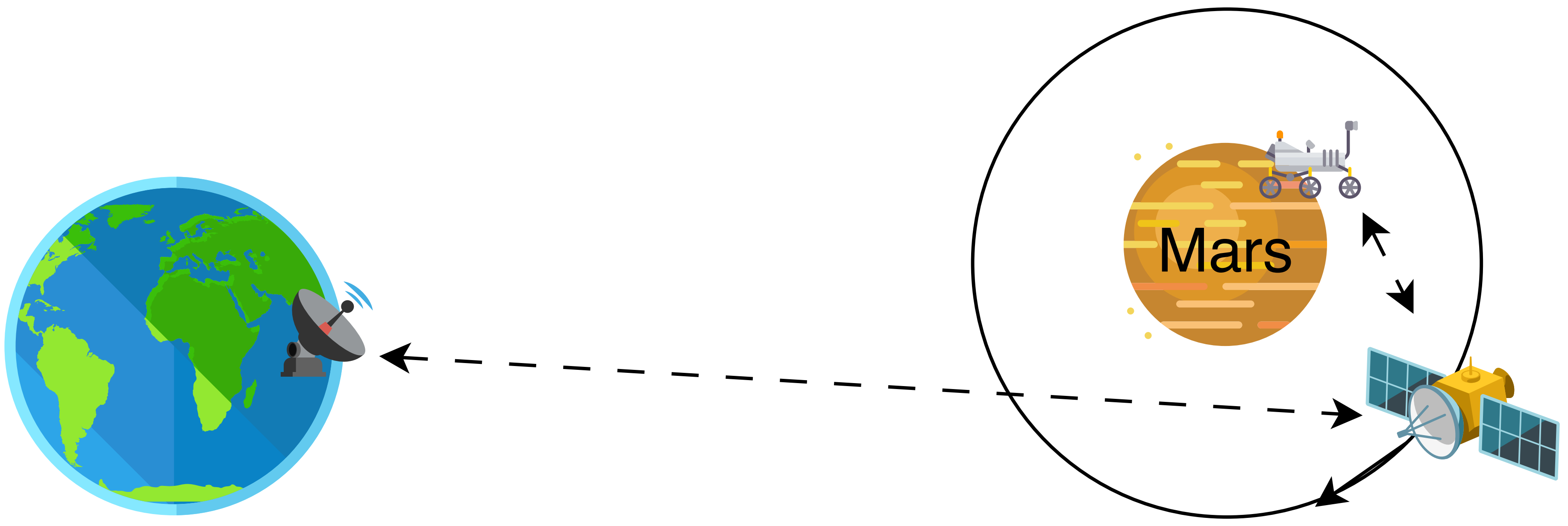
Does IP work in Deep Space?

Let's put it to test!

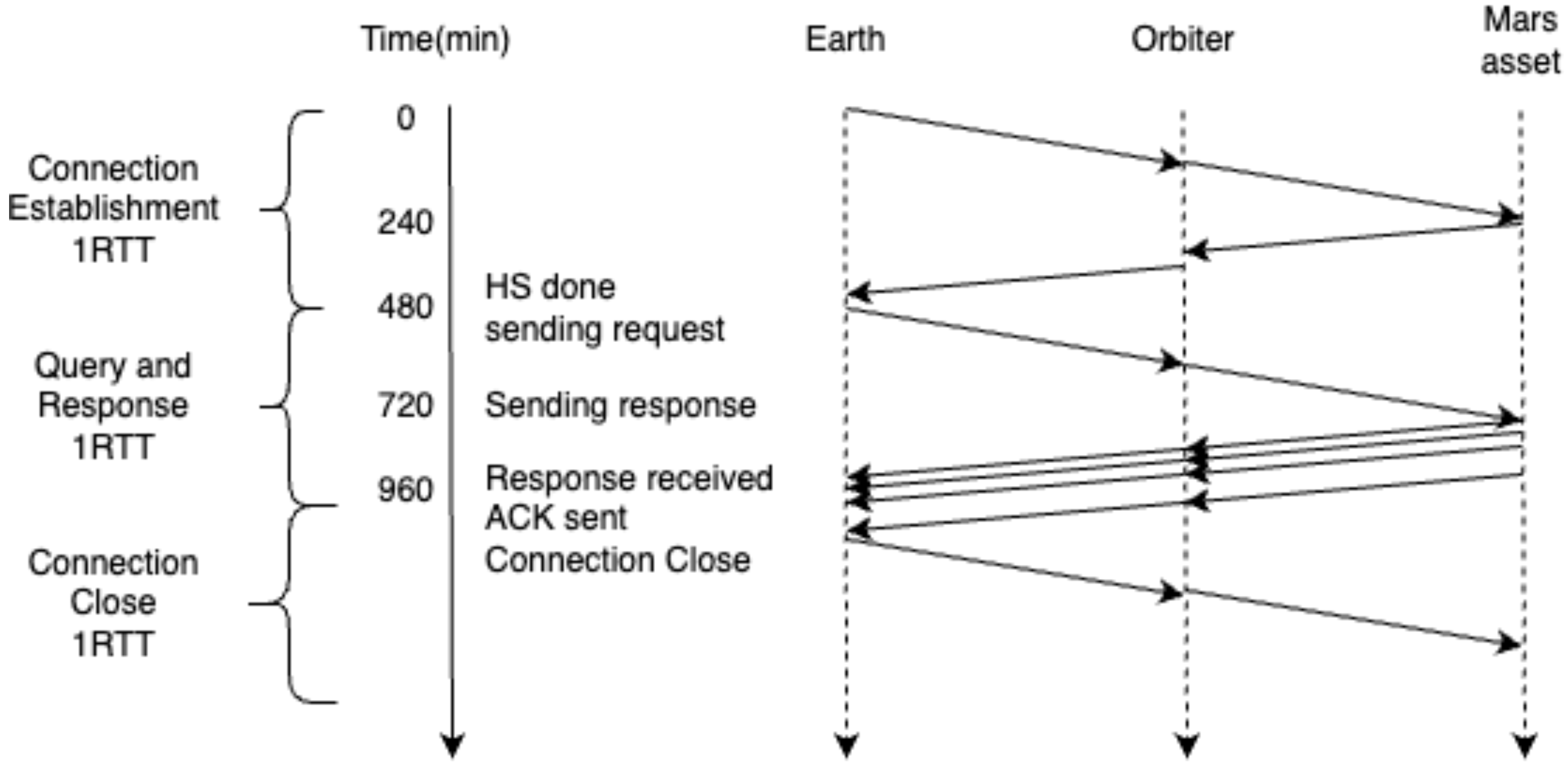
Moon: 1.5 seconds. Too easy ;-). Let's do Mars!



Earth to Mars via Orbiter



- Simulation: HTTP/QUIC request and response
- 4 min (240s) one-way delay (Mars and Earth nearest)
    - Side note: <270s max for tc netem delay before 2024-02 fix
- Direct Earth node - Mars orbiter - Mars asset: no intermittence
- HS = 1RTT Handshake
- Connection close: not needed, can keep connection opened "forever" for additional requests
- Two different QUIC implementations used



Client Wireshark

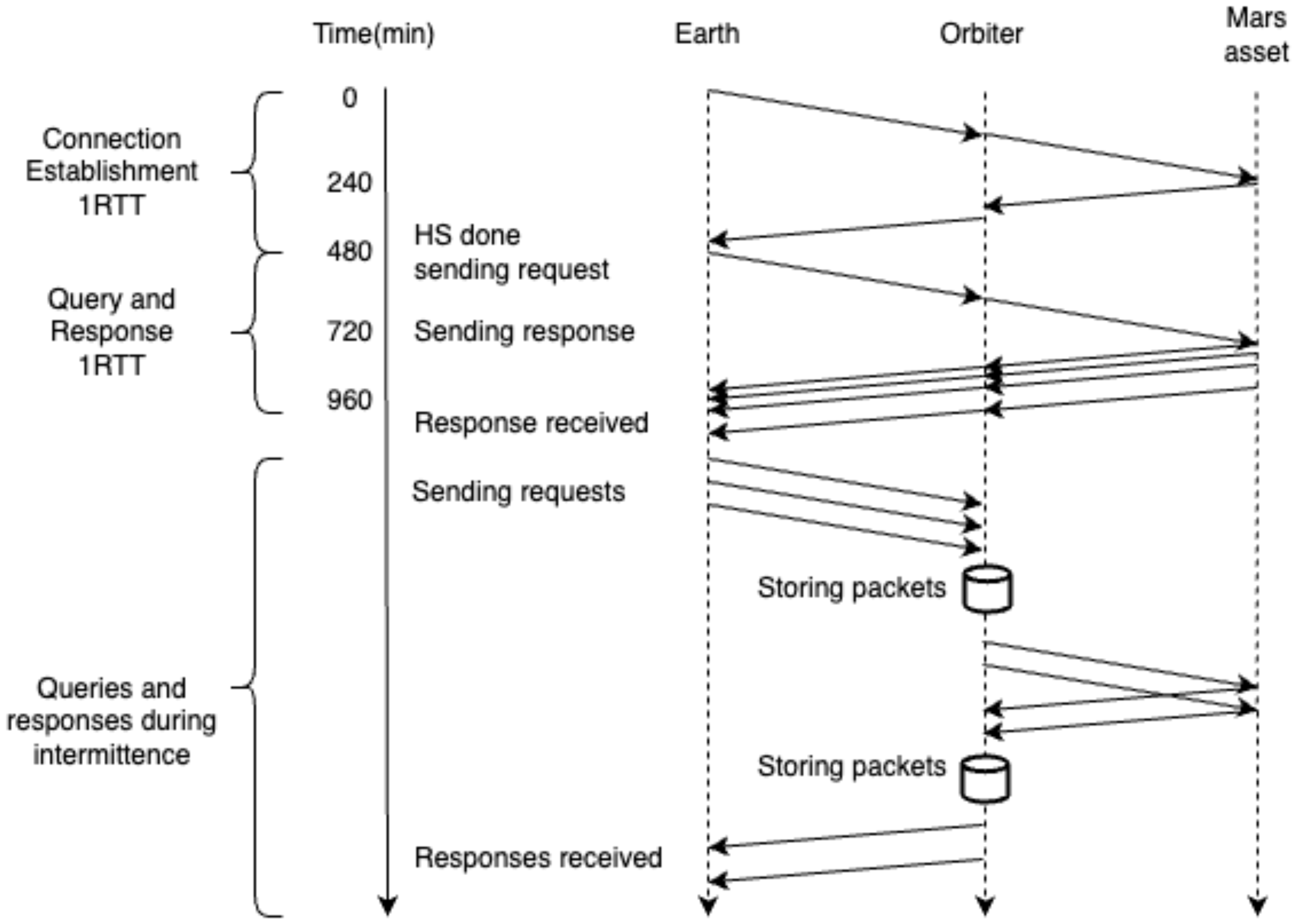
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.40.1	192.168.42.1	QUIC	1242	Initial, DCID=ba7bb2be15d544e9aa76900070e41a9bacaa826e, SCID=dbd14607fed99229, PKN: 0, CRYPTO, PADDING
2	240.763219	192.168.42.1	192.168.40.1	QUIC	1686	Handshake, DCID=dbd14607fed99229, SCID=bc54d768409abe435a4c5c4904abe9788b088cc9, PKN: 2, CRYPTO
3	480.801468	192.168.40.1	192.168.42.1	QUIC	1242	Handshake, DCID=bc54d768409abe435a4c5c4904abe9788b088cc9, SCID=dbd14607fed99229, PKN: 0, ACK, CRYPTO, PADDING
4	480.801600	192.168.40.1	192.168.42.1	QUIC	276	Protected Payload (KP0), DCID=bc54d768409abe435a4c5c4904abe9788b088cc9, PKN: 0, NCI, NCI, NCI, NCI, NCI, NCI, NCI
5	480.801602	192.168.40.1	192.168.42.1	QUIC	100	Protected Payload (KP0), DCID=bc54d768409abe435a4c5c4904abe9788b088cc9, PKN: 1, STREAM(0)
6	721.486731	192.168.42.1	192.168.40.1	QUIC	803	Protected Payload (KP0), DCID=dbd14607fed99229, PKN: 3, ACK, NCI, NCI, NCI, NCI, DONE, CRYPTO, STREAM(0)
7	961.609775	192.168.40.1	192.168.42.1	QUIC	86	Protected Payload (KP0), DCID=bc54d768409abe435a4c5c4904abe9788b088cc9, PKN: 2, ACK
8	961.609810	192.168.40.1	192.168.42.1	QUIC	93	Protected Payload (KP0), DCID=bc54d768409abe435a4c5c4904abe9788b088cc9, PKN: 3, ACK, CC

What about intermittence?

Such as orbiter with blackout periods

Earth to Mars with Intermittence

- IP packets stored during intermittence
- Intermittence: 1h, 2 times
- 4 min. one-way delay
- Send 1 request every 15 minutes
  - 20 times: aka 20 requests, 20 responses



Earth to Mars with Intermittence

Requests
every 15 min.

Orbiter-Asset
link down.
Packet storage.

1	0.000000	192.168.40.1	192.168.42.1	QUIC	1242	Initial, DCID=bfac32299c63c485a3db2566ad9b8a36dd5faa27, SCID=896064ae61744008, PKN: 0, CRYPTO, PADDING
2	240.705935	192.168.42.1	192.168.40.1	QUIC	1380	Protected Payload (KP0), DCID=896064ae61744008
3	480.733715	192.168.40.1	192.168.42.1	QUIC	1242	Initial, DCID=cc5a86d7b7951123, SCID=896064ae61744008, PKN: 1, ACK_ECN
4	480.733733	192.168.40.1	192.168.42.1	QUIC	180	Protected Payload (KP0), DCID=cc5a86d7b7951123
5	480.733735	192.168.40.1	192.168.42.1	QUIC	88	Protected Payload (KP0), DCID=cc5a86d7b7951123
6	480.733736	192.168.40.1	192.168.42.1	QUIC	88	Protected Payload (KP0), DCID=cc5a86d7b7951123
7	721.601456	192.168.42.1	192.168.40.1	QUIC	416	Protected Payload (KP0)
8	721.601456	192.168.42.1	192.168.40.1	QUIC	203	Protected Payload (KP0)
9	961.659318	192.168.40.1	192.168.42.1	QUIC	76	Protected Payload (KP0), DCID=cc5a86d7b7951123
10	961.659340	192.168.40.1	192.168.42.1	QUIC	96	Protected Payload (KP0), DCID=cc5a86d7b7951123
11	1380.733785	192.168.40.1	192.168.42.1	QUIC	88	Protected Payload (KP0), DCID=cc5a86d7b7951123
12	1621.433896	192.168.42.1	192.168.40.1	QUIC	76	Protected Payload (KP0)
13	1621.433897	192.168.42.1	192.168.40.1	QUIC	203	Protected Payload (KP0)
14	2280.842765	192.168.40.1	192.168.42.1	QUIC	88	Protected Payload (KP0), DCID=cc5a86d7b7951123
15	2342.318865	192.168.40.1	192.168.42.1	QUIC	79	Protected Payload (KP0), DCID=cc5a86d7b7951123
16	2521.647591	192.168.42.1	192.168.40.1	QUIC	203	Protected Payload (KP0)
17	3002.520627	192.168.42.1	192.168.40.1	QUIC	79	Protected Payload (KP0)
18	3180.961165	192.168.40.1	192.168.42.1	QUIC	88	Protected Payload (KP0), DCID=cc5a86d7b7951123
19	3242.549809	192.168.40.1	192.168.42.1	QUIC	79	Protected Payload (KP0), DCID=cc5a86d7b7951123
20	3421.709413	192.168.42.1	192.168.40.1	QUIC	203	Protected Payload (KP0)
21	4081.022304	192.168.40.1	192.168.42.1	QUIC	88	Protected Payload (KP0), DCID=cc5a86d7b7951123
22	4142.610331	192.168.40.1	192.168.42.1	QUIC	79	Protected Payload (KP0), DCID=cc5a86d7b7951123
23	4981.097722	192.168.40.1	192.168.42.1	QUIC	88	Protected Payload (KP0), DCID=cc5a86d7b7951123
24	5881.122075	192.168.40.1	192.168.42.1	QUIC	88	Protected Payload (KP0), DCID=cc5a86d7b7951123
25	6781.207992	192.168.40.1	192.168.42.1	QUIC	88	Protected Payload (KP0), DCID=cc5a86d7b7951123
26	7681.127528	192.168.40.1	192.168.42.1	QUIC	88	Protected Payload (KP0), DCID=cc5a86d7b7951123
27	8581.162835	192.168.40.1	192.168.42.1	QUIC	88	Protected Payload (KP0), DCID=cc5a86d7b7951123
28	8614.130449	192.168.42.1	192.168.40.1	QUIC	102	Protected Payload (KP0)
29	8854.144409	192.168.40.1	192.168.42.1	QUIC	78	Protected Payload (KP0), DCID=cc5a86d7b7951123
30	9481.129946	192.168.40.1	192.168.42.1	QUIC	88	Protected Payload (KP0), DCID=cc5a86d7b7951123
31	10381.362734	192.168.40.1	192.168.42.1	QUIC	88	Protected Payload (KP0), DCID=cc5a86d7b7951123
32	11281.370662	192.168.40.1	192.168.42.1	QUIC	88	Protected Payload (KP0), DCID=cc5a86d7b7951123
33	12181.474612	192.168.40.1	192.168.42.1	QUIC	88	Protected Payload (KP0), DCID=cc5a86d7b7951123
34	13081.634629	192.168.40.1	192.168.42.1	QUIC	88	Protected Payload (KP0), DCID=cc5a86d7b7951123
35	13981.768204	192.168.40.1	192.168.42.1	QUIC	89	Protected Payload (KP0), DCID=cc5a86d7b7951123
36	14398.274138	192.168.42.1	192.168.40.1	QUIC	78	Protected Payload (KP0)
37	14398.274139	192.168.42.1	192.168.40.1	QUIC	365	Protected Payload (KP0)
38	14398.274139	192.168.42.1	192.168.40.1	QUIC	364	Protected Payload (KP0)
39	14638.302876	192.168.40.1	192.168.42.1	QUIC	79	Protected Payload (KP0), DCID=cc5a86d7b7951123
40	14881.742836	192.168.40.1	192.168.42.1	QUIC	89	Protected Payload (KP0), DCID=cc5a86d7b7951123
41	15122.532603	192.168.42.1	192.168.40.1	QUIC	204	Protected Payload (KP0)
42	15603.527251	192.168.42.1	192.168.40.1	QUIC	79	Protected Payload (KP0)
43	15781.746285	192.168.40.1	192.168.42.1	QUIC	89	Protected Payload (KP0), DCID=cc5a86d7b7951123
44	15843.428086	192.168.40.1	192.168.42.1	QUIC	81	Protected Payload (KP0), DCID=cc5a86d7b7951123
45	16022.426337	192.168.42.1	192.168.40.1	QUIC	204	Protected Payload (KP0)
46	16503.365773	192.168.42.1	192.168.40.1	QUIC	79	Protected Payload (KP0)
47	16681.808234	192.168.40.1	192.168.42.1	QUIC	89	Protected Payload (KP0), DCID=cc5a86d7b7951123
48	16743.511417	192.168.40.1	192.168.42.1	QUIC	79	Protected Payload (KP0), DCID=cc5a86d7b7951123
49	16922.628931	192.168.42.1	192.168.40.1	QUIC	204	Protected Payload (KP0)

Longer Delays. Possible?

An HTTP Request to Voyager!

(In simulation)

- 18 hours (64800s) one-way delay
- Direct link, Earth and Voyager nodes
- HTTP over configured QUIC
- Full QUIC flow: connection establishment (1,2), request and response (4,5), connection close(7,8). Additional features (3,6)

	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.65.33	192.168.65.25	QUIC	1242	Initial, DCID=d61b8e047f
2	64800.438656	192.168.65.25	192.168.65.33	QUIC	1380	Handshake, DCID=2f26ef8a
3	129600.8077…	192.168.65.33	192.168.65.25	QUIC	1242	Handshake, DCID=bf92a7a2
4	129600.8086…	192.168.65.33	192.168.65.25	QUIC	200	Protected Payload (KP0),
5	194401.1215…	192.168.65.25	192.168.65.33	QUIC	691	Protected Payload (KP0)
6	259201.4231…	192.168.65.33	192.168.65.25	QUIC	79	Protected Payload (KP0),
7	259201.4236…	192.168.65.33	192.168.65.25	QUIC	96	Protected Payload (KP0),
8	259201.4245…	192.168.65.33	192.168.65.25	QUIC	86	Protected Payload (KP0),

What about packet loss?

Let's try 5% packet loss over very long delay

Delay of 24 hours and 5% packet loss

- One way 24 hours delay(86400s), packet loss 5%, 10 times repeat HTTP request and response in the same connection
- Total time: 1987200s
  - same as without packet loss, since loss was recovered using the next packets
- Client data packets sent: 20, 3087 bytes
- Server data packets sent: 22, 12313 bytes
  - Server packets dropped: 2
  - (by the network simulation)
- Conclusion: QUIC recovered successfully and all data were properly 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

✅

QUIC Connections and Requests

- A QUIC connection may last minutes, hours, days, weeks, months.
- Within a QUIC connection:
  - Multiple requests and responses are carried out
  - Additional protocols/data can also be carried out.
    - Example: the same connection used for requests/responses can also carry video/audio streaming, network management queries, etc
- Therefore the connection establishment time (1RTT) may only be needed at start of the mission. If for whatever reason, the connection dies, then the marginal cost is 1RTT.
- If no need for reliability, then use "pure" UDP.

What about Network Management? QoS? Streaming?

Network Services

- Network Management: use SNMP/UDP (IETF deprecated) or NETCONF-RESTCONF/QUIC
- QoS: use the whole IP QoS/TE toolkit; can use: source/destination addresses, diffusers marking, port/service, flow label, ...
- Naming: use DNS locally (on celestial body network)
- Emergency messaging: may use terrestrial framework (ECRIT)
- Time distribution: may use NTP
- Media/Streaming: many choices: RTP, HTTP, MoQ, ...

Possible Deployment Scenarios

- End to end: Earth/Asset applications talk directly
- Proxy based architecture:
    - Space edge proxy:
        - On surface side, handles specifics of local celestial body networks
            - Earth well connected, high-speed networking
            - Celestial body surface and around well connected, high-speed networking
        - On deepspace side, handles specifics of deepspace
            - Handles intermittence, long delays
            - Apply network policies (QoS, routes, security) to avoid unwanted trafic
            - May have more knowledge of intermittence (only those highly managed have that knowledge)

IETF TIPTOP WG

TIPTOP Charter (repeat)

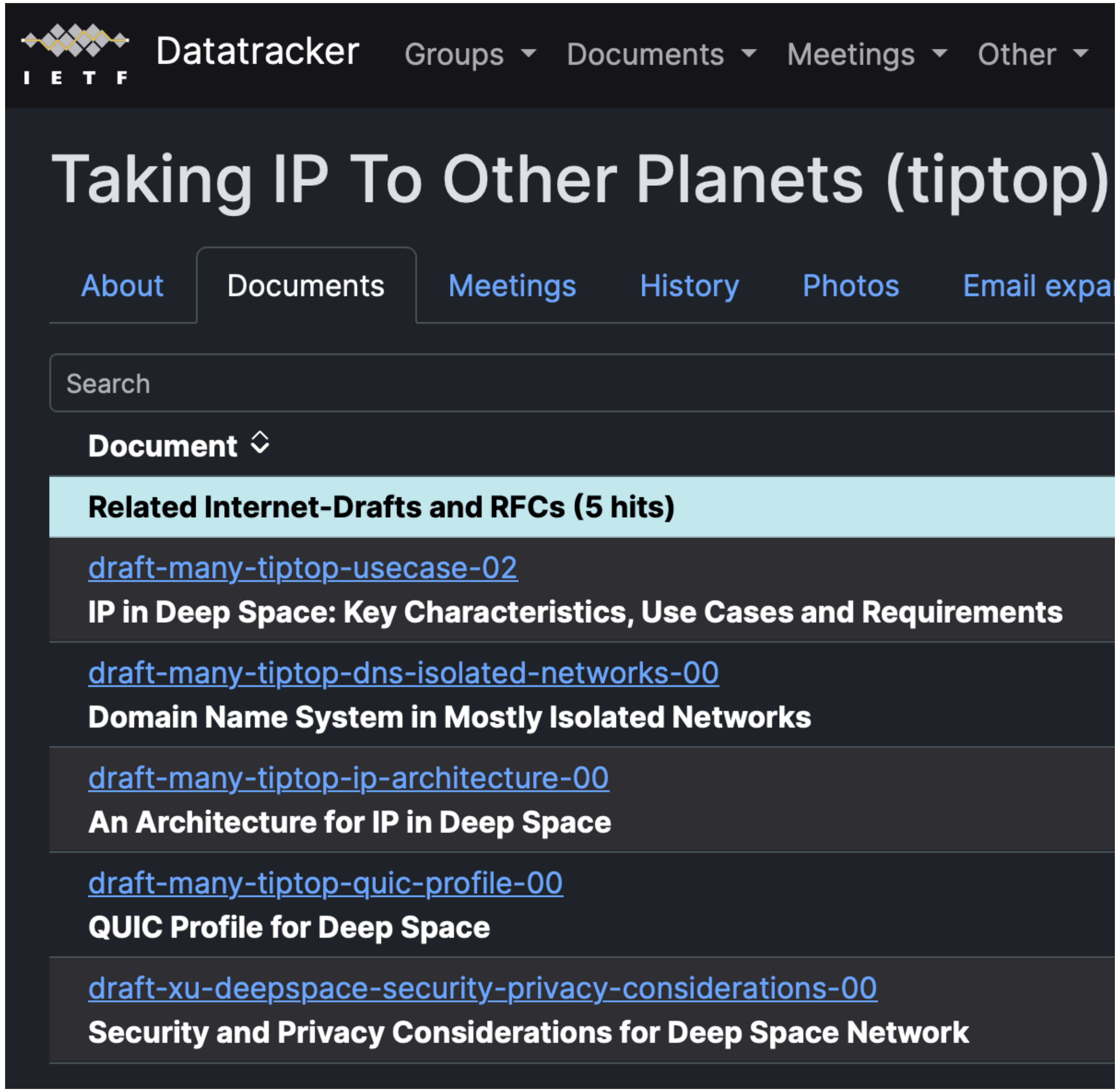
- In a Nutshell:
  - Given the delays and disruptions involved in space communications, engineer IP-based internetworking and support IP applications end-to-end.
  - IP-based protocols may be profiled, but no new protocols are in scope.
  - Target scenarios: Moon and Mars, and deep space
  - Out of scope: Bundle Protocol(BP), LEO/MEO/GEO scenarios
  - Collaboration/coordination with space ecosystem and within IETF
- Work items:
  - key characteristics, use cases, and requirements
  - differences that apply to the IP architecture
  - recommendations for existing transport protocols (+ their security protocols), starting with QUIC.
  - considerations for DNS

Taking IP to Other Planets (TIPTOP)

- Timeline:
  - 2022-2023: initial re-assessment work on IP in deepspace. (draft-many-deepspace-ip-assessment)
  - Summer 2023: IETF-hosted deepspace informal mailing list.
  - 2023-2024: informal deepspace side meetings at IETFs: problem, solutions, simulations results, CoAP, SCHC, SRv6, quic profile implementations, study on Mars orbiters-rovers intermittence. All this available at: https://deepspaceip.github.io
  - Fall 2024, Dublin IETF: Working group forming BOF. Presentations from/about: China Space Agency, KDDI, NASA/LNIS/LunaNet, proposed architecture and quic profile
  - Feb 2025: Working group formed.
    - Deepspace name was not agreed. TIPTOP name suggestion from Wesley Eddy
      - Been told (by Sylvia Hagen) that tiptop means "perfect" in German! What a great coincidence... or too difficult expectation!
    - Chaired by Padma Pillay-Esnault. Technical Advisor/Delegate: Marc Blanchet. Area Director: Eric Vyncke (INT)
  - Spring 2025, Bangkok IETF: First meeting. Co-chair added: Zaheduzzaman Sarker. Agenda: use case, architecture, quic profile. Other topics pushed because lack of time.
  - Summer 2025, Madrid IETF: Second meeting planned

TIPTOP List of (Related) Documents

- None are wg docs yet
- draft-many-tiptop-{usecase, ip-architecture, quic-profile} have been presented during TIPTOP first meeting (last IETF)
- Drafts that have been presented during the deepspace informal meetings (prior to TIPTOP formation), or have direct relevance:
  - draft-many-deepspace-ip-assessment
  - draft-gomez-core-coap-space
  - draft-huitema-quic-in-space
  - draft-li-tiptop-address-space



Additional Topics

- While IP and BP are different, they share some generic difficult topics:
  - Routing on a large scale: given delays and disruptions, how to advertise(?) changes "in-time"
  - How to properly use link schedules in any protocol/element, where the schedule may change (and need to be updated everywhere it needs and in time) or unscheduled events happen, while not being fragile
  - How to handle congestion, where typical signalling/finding may happen "too late"
  - How to handle deep buffers/storage (for store and forward)
  - How to properly pre-fetch/cache"data" on celestial body networks in advance of its needed usage? (Ex: key validation chains, cert revocation, ...)
  - Other transports for IP? A "transport" for BP?
- There has been a proposal for a IRTF research group for space...

Acknowledgments

- Deepspace/TIPTOP BOF: chairs: Alvaro Retena, Lars Eggert; presenters: Xiongwen He, Atsushi Tagami, Wesley Eddy
- TIPTOP wg: Padma Pillay-Esnault, Zaheduzzaman Sarker, Eric Vyncke
- Quinn workbench (QUIC simulation engine for deepspace): Adolfo Ochagavia
- IP storage (store and forward): Jean-Philippe Dionne
- IP assessment: Christian Huitema, Dean Bogdanovic
- draft-many-tiptop*: Wesley Eddy, Tony Li, Marshall Eubanks
- SCHC: Laurent Toutain
- CoAP: Carles Gomez Montenegro
- DNS: Warren Kumary, Mark Andrews
- QUIC discussions: François Michel, Maxime Piraud, Christian Huitema, Émile Stéphan, Martin Duke, Martin Thompson, Lars Eggert, Ian Swett,
- Architecture discussions: Vint Cerf, Felix Flentge, Scott Burleigh, Keith Scott, James Schier, Juan Fraire, ... too many to list
- Documents reviews: too many too list

- NB: not everybody agrees, but they all spent significant time to discuss and argue.
- NB2: I'm sure I forgot some people. Sorry.

Conclusion and More Information

- The Internet Protocol (IP) suite in deep space is discussed in the IETF TIPTOP working group.
- To make it "work" in deep space, in a nutshell:
    - Store IP packets in forwarders facing intermittence
    - Configure transport such as QUIC with a space profile
    - For applications, modify timeouts appropriately and apply asynchronous design
- For more information:
    - IETF tiptop working group: https://datatracker.ietf.org/group/tiptop/about/
        - Join the discussion on the mailing list!
    - Some IETF documents (not yet wg docs):
        - https://datatracker.ietf.org/doc/draft-many-tiptop-usecase/
        - https://datatracker.ietf.org/doc/draft-many-tiptop-ip-architecture/
        - https://datatracker.ietf.org/doc/draft-many-tiptop-quic-profile/
- Additional Information
    - Deep Space IP initiative: https://deepspaceip.github.io
    - QUIC simulation engine: https://github.com/aochagavia/quinn-workbench
- Contact information:
    - Marc Blanchet, Viagenie, marc.blanchet@viagenie.ca

Annexes

IP-BP Comparison in One (Imperfect and Incomplete) Table

	Key Benefit/Detriment	Internet Protocol(IP) Suite	Bundle Protocol (BP)
Packet/Bundle loss	E2E reliability	Transport level. End-to-end. QUIC. Nothing to do by app. Resending from the source.	Hop-by-hop, not end-to-end. Application has to create state and connection.
Packet/Bundle reordering	E2E reliability	Transport level. End-to-end. QUIC. Nothing to do by app	Not available. Must be done by each application
Packet/Bundle duplication	E2E reliability	Transport level. End-to-end. QUIC. Nothing to do by app	Not available. Must be done by each application
Flow control	Network/endpoint not overflown	Transport level. End-to-end. QUIC. Nothing to do by app	Not available. Must be done by each application
Route aggregation	Scalability	Base architecture	Not, by design. NodeID are permanent to the node.
Mobility	Scalability	Done at transport level between endpoints, no impact on app or network.	Given id permanent, (very) difficult to scale
Network-wide QoS	Emergency	Marking and policy routing/forwarding available/implemented.	Being discussed. Currently only hop-by-hop.
Fast forwarding	Scalability, performance	Can forward at 100+Gbps. Simple header. Fixed length fields. Fixed mapped header fields.	Headers are variable lengths and encoded. Extensive use of extension headers. Not fast
What needs to be done	Easiness to deploy	Simple deep buffers in forwarders. Use profiled QUIC. Set timers in apps	Deploying BP is complicated and gives base functionality
Application-level gateways	Complexity, brittleness	Not needed. Choice of deployment	Required to bring IP apps transported over BP
Network Management	Know and control	SNMP, Netconf available and working.	Being discussed. One partial implementation known
Application Frameworks	Faster to develop	Many available, open-source	Only one known (AMS)
Network wide Protocol	Simpler to deploy, way less brittle	Single network protocol everywhere: Earth, deep space links, celestial body networks, on-board spacecraft	Need to mix both IP and BP. Independent management, routing, security, ...
Security	Confidentiality, Integrity, Authentication	Large set of tools, protocols, standards, reviewed/scutinized/certified (HAIPE) implementations, at all levels of the stack	Only at the BP layer. Not seen by the app. Single implementation (not fully open-source)
Fragmentation	Key for variable frame length	Implemented	Being discussed
Availability of Open-source software, OS, ...	Cost effective, faster	Very large. Socket API. Many many languages	A few, only one has a majority of features
Availability of Knowledge	Cost effective, faster	Very large	Very very few