June 30, 2022

ATSSS_ph3 Conf call on KI#2 reordering

Apostolis Salkintzis

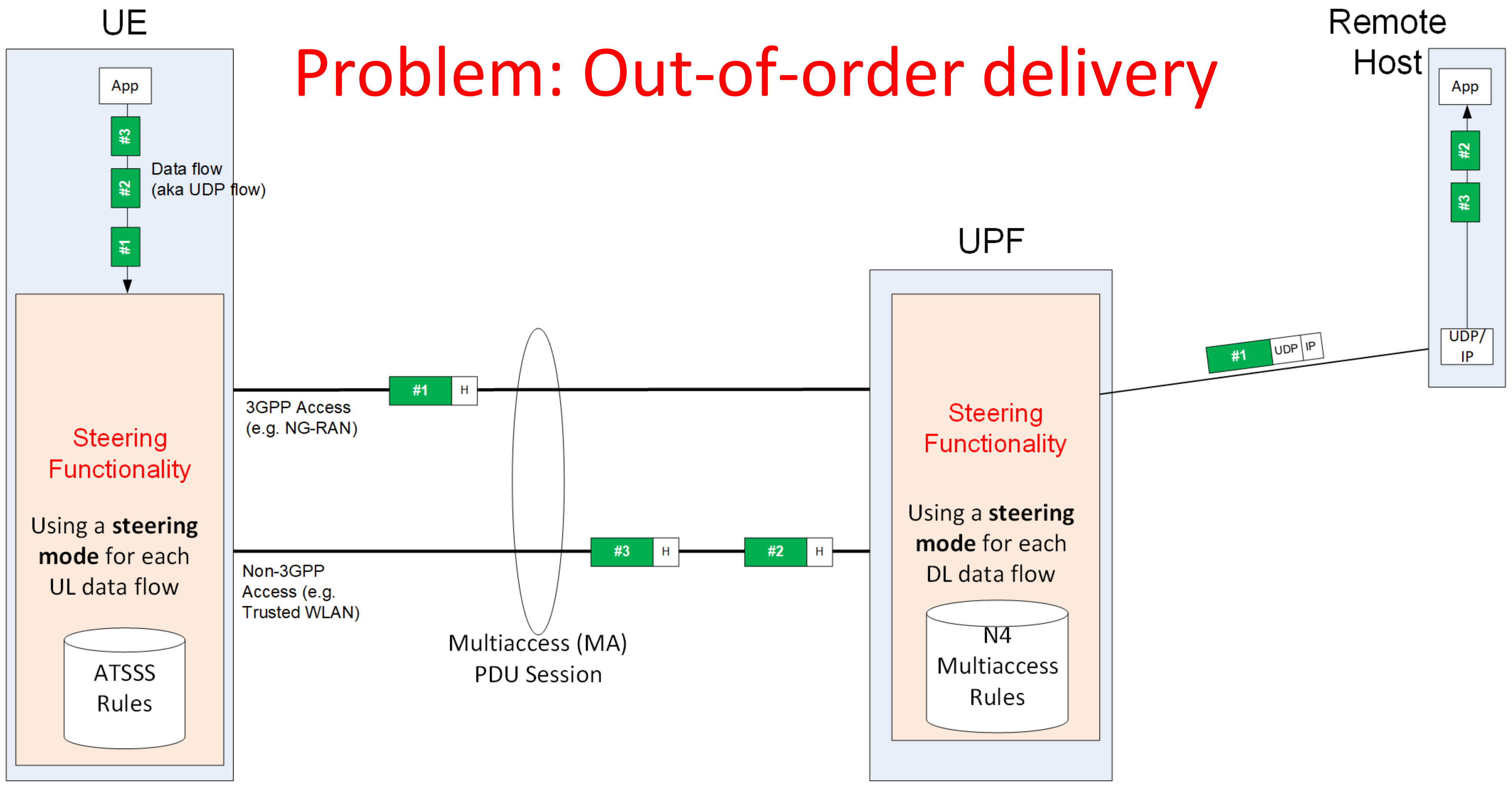
Lenovo

Introduction

- Reminder: KI#2 is about defining a new steering functionality based on MP-QUIC or MP-DCCP. Three solutions have been proposed in TR 23.700-53: #2.1 (MP-DCCP), #2.2 (MP-QUIC), #2.3 (MP-QUIC).
- Today's discussion NOT about comparing or evaluating the three solutions for KI#2. It is about the packet re-ordering issue that is applicable to all solutions.
- Agenda:
  1. Identify the problem in the context of ATSSS (slides 3)
  2. Discuss the slides shared by DT on "Multipath Re-ordering findings"
  3. Discuss how the MP-QUIC based solution addresses the problem (slides 4-8)
  4. Conclusions and proposed actions (slide 9)

Problem: Out-of-order delivery

- When a data flow is steered using the Load-Balancing steering mode, the packets of the data flow are split across 3GPP and the non-3GPP accesses.
- When the two accesses have significantly different delays (tens of msec), the packets of the data flow can be received out of order.
- In many cases, re-ordering of the packets can be done by the app itself. For example, when the app uses the QUIC protocol or the RTP protocol.
- In other cases, re-ordering should be done in the steering functionality as the app may not perform packet re-ordering (does not expect excessive out-of-order delivery).

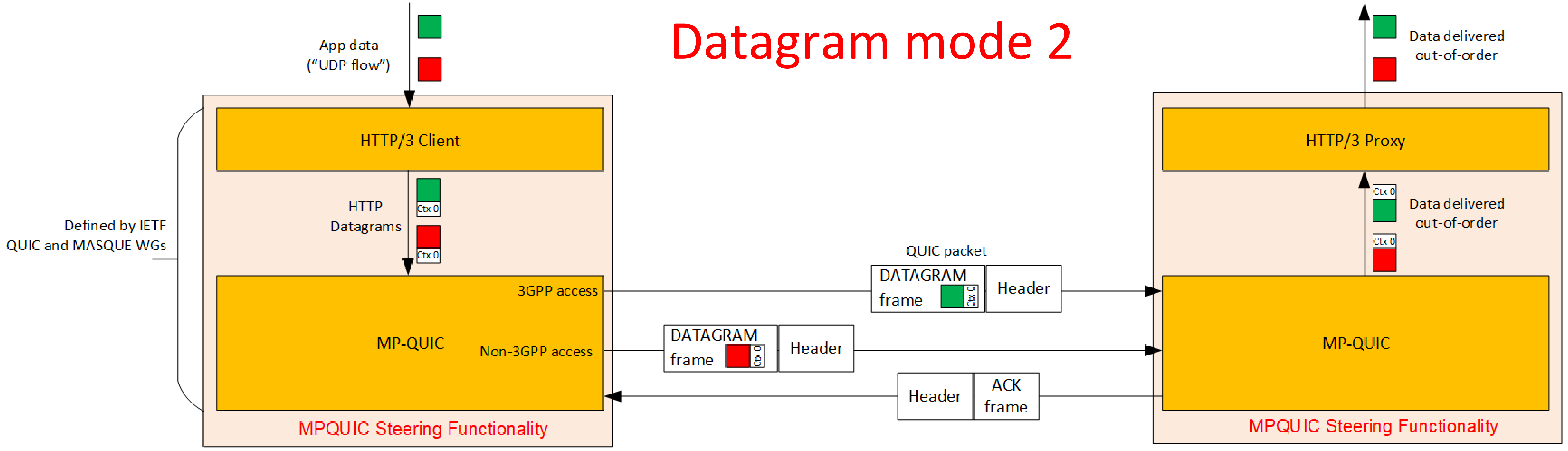

UE
App
#3
#2
#1
Data flow (aka UDP flow)
Steering Functionality
Using a steering mode for each UL data flow
ATSSS Rules
#1 H
3GPP Access (e.g. NG-RAN)
#3 H
#2 H
Non-3GPP Access (e.g. Trusted WLAN)
Multiaccess (MA) PDU Session
UPF
Steering Functionality
Using a steering mode for each DL data flow
N4 Multiaccess Rules
#1 UDP IP
Remote Host
App
#2
#3
UDP/ IP

- **Requirement**: The steering functionality should be able to perform packet re-ordering for data flows that (a) use the Load-Balancing steering mode and (b) when packet re-ordering is not supported by the app. When packet re-ordering is supported by the app, the steering functionality should not perform re-ordering.
- The text we have for KI#2 does not specify strict requirements for handling out-of-order delivery. It only indicates we are committed to studying the issue: "*How to treat out-of-order delivery caused by per packet-splitting.*"

How the MP-QUIC based solution addresses the problem

- 3 transport modes considered for the MP-QUIC based solution:
  - Datagram mode 2 (unreliable QUIC transport / Datagrams without re-ordering)
  - Datagram mode 1 (unreliable QUIC transport / Datagrams with re-ordering)
  - Stream Mode (reliable QUIC transport / Streams with re-ordering)
- Also, the ATSSS rules are enhanced to indicate the transport mode, e.g.:
- - Traffic descriptor
- o Application identity: com.example.app1
- o Protocol: UDP
- - Access Selection descriptor
- o Steering functionality: MP-QUIC
- o Steering mode: Load-balancing
- o Steering Mode Information: 50% over 3GPP and 50% over non-3GPP
- o Threshold Values: Max Round-Trip-Time (RTT) = 20ms
- o Transport mode: Datagram mode 2

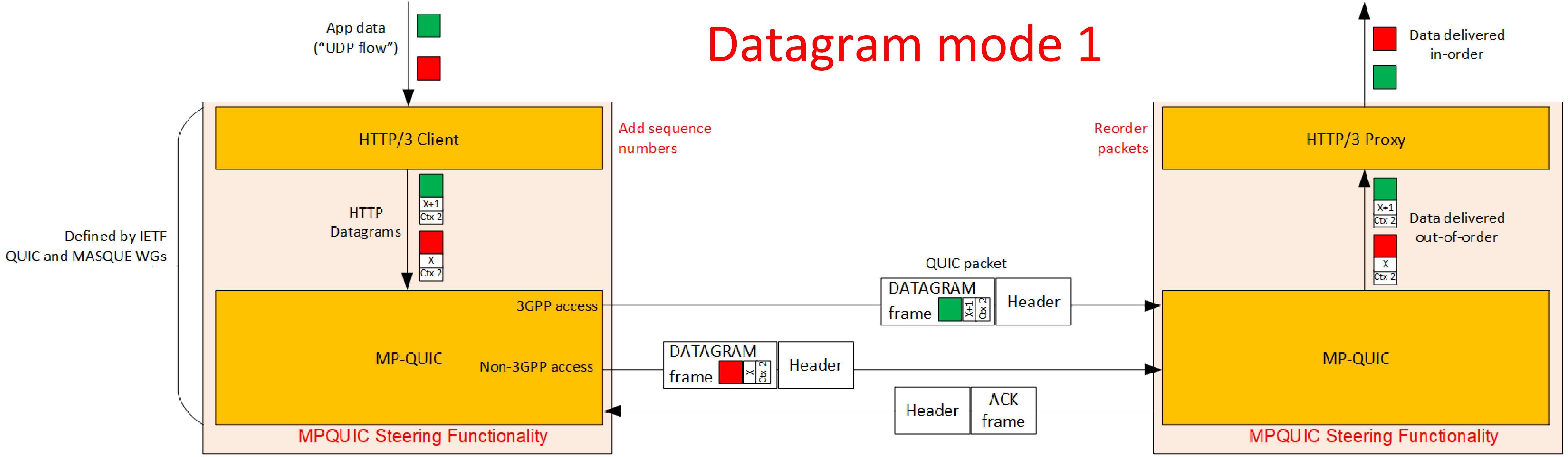
Datagram mode 2



- Data from app is encapsulated into HTTP Datagrams and then into QUIC Datagram frames, as specified in *draft-ietf-masque-connect-udp* and in *RFC 9221* (An Unreliable Datagram Extension to QUIC). No packet re-ordering is supported. The Context ID value of zero is defined in *draft-ietf-masque-connect-udp*.
- Datagram frames are subject to congestion control. They trigger ACKs to facilitate loss detection but retransmissions are not used (unreliable transport).
- Useful for (a) steering modes that do not create out-of-order delivery and for (b) apps that support re-ordering.
- Example ATSSS rule:
  - Traffic descriptor
    - Application identity: com.example.app1 ----> can support re-ordering (e.g., implements QUIC)
    - Protocol: UDP
  - Access Selection descriptor
    - Steering functionality: MP-QUIC
    - Steering mode: Load-balancing
    - Steering Mode Information: 50% over 3GPP and 50% over non-3GPP
    - Threshold Values: Max Round-Trip-Time (RTT) = 20ms
    - Transport mode: Datagram mode 2

```
HTTP/3 Datagram {
   Quarter Stream ID,
   HTTP Datagram Payload (Context ID=0, App-Data)
 }
```

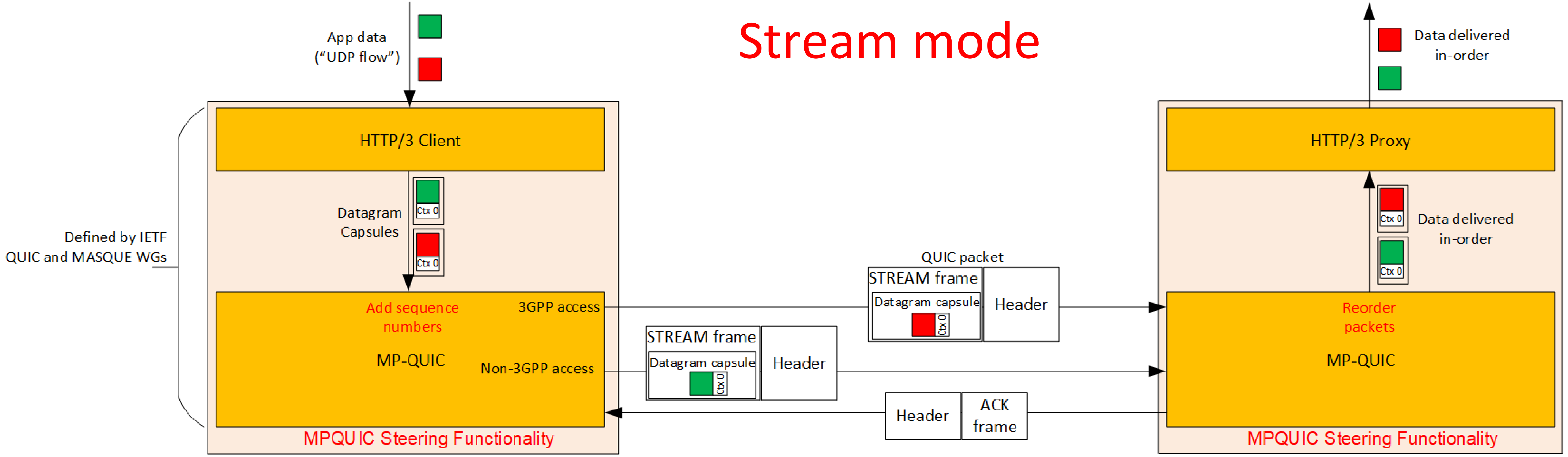
Datagram mode 1



- Data from app is encapsulated into HTTP Datagrams and then into QUIC Datagram frames, as specified in draft-ietf-masque-connect-udp and in RFC 9221.
- Packet re-ordering is supported in the HTTP/3 layer using a new Context ID (e.g., with value 2), which should be defined in 3GPP and its value should be registered with IANA.
- The re-ordering algorithm may be left to the implementation or may be defined by an SDO (3GPP or IETF?).
- Useful for (a) the Load-Balancing steering mode (that can create out-of-order delivery) and for (b) apps that do not support re-ordering.
- Example ATSSS rule:
  - Traffic descriptor
    - Application identity: com.example.app2 ----> <u>does not support re-ordering</u>
    - Protocol: UDP
  - Access Selection descriptor
    - Steering functionality: MP-QUIC
    - Steering mode: Load-balancing
    - Steering Mode Information: 50% over 3GPP and 50% over non-3GPP
    - Threshold Values: Max Round-Trip-Time (RTT) = 20ms
    - Transport mode: Datagram mode 1

```
HTTP/3 Datagram {
   Quarter Stream ID,
   HTTP Datagram Payload (Context ID=2, Seq. Num, App-Data)
 }
```

Stream mode



- Data from app is encapsulated into HTTP Datagrams and then the HTTP Datagrams are sent on a QUIC stream using the Capsule Protocol specified in draft-ietf-masque-h3-datagram.
- Packet re-ordering is supported in the QUIC layer using the existing mechanisms for stream transport. Data packets in a stream are numbered and retransmitted to recover from losses.
- Useful for (a) the Load-Balancing steering mode (that can create out-of-order delivery) and for (b) apps that do not support re-ordering.
- The stream transport mode is not used for apps that support QUIC (because re-ordering can be done by the app), so the meltdown effects are avoided.
- Example ATSSS rule:
  - Traffic descriptor
    - Application identity: com.example.app3 ----> does not support re-ordering
    - Protocol: UDP
  - Access Selection descriptor
    - Steering functionality: MP-QUIC
    - Steering mode: Load-balancing
    - Steering Mode Information: 50% over 3GPP and 50% over non-3GPP
    - Threshold Values: Max Round-Trip-Time (RTT) = 20ms
    - Transport mode: Stream mode

```
Datagram Capsule {
    Type (i) = 0x00,
    Length,
    HTTP Datagram Payload (Quarter Stream ID, Context ID=0, App-Data),
}
```

Summary of re-ordering support with MP-QUIC

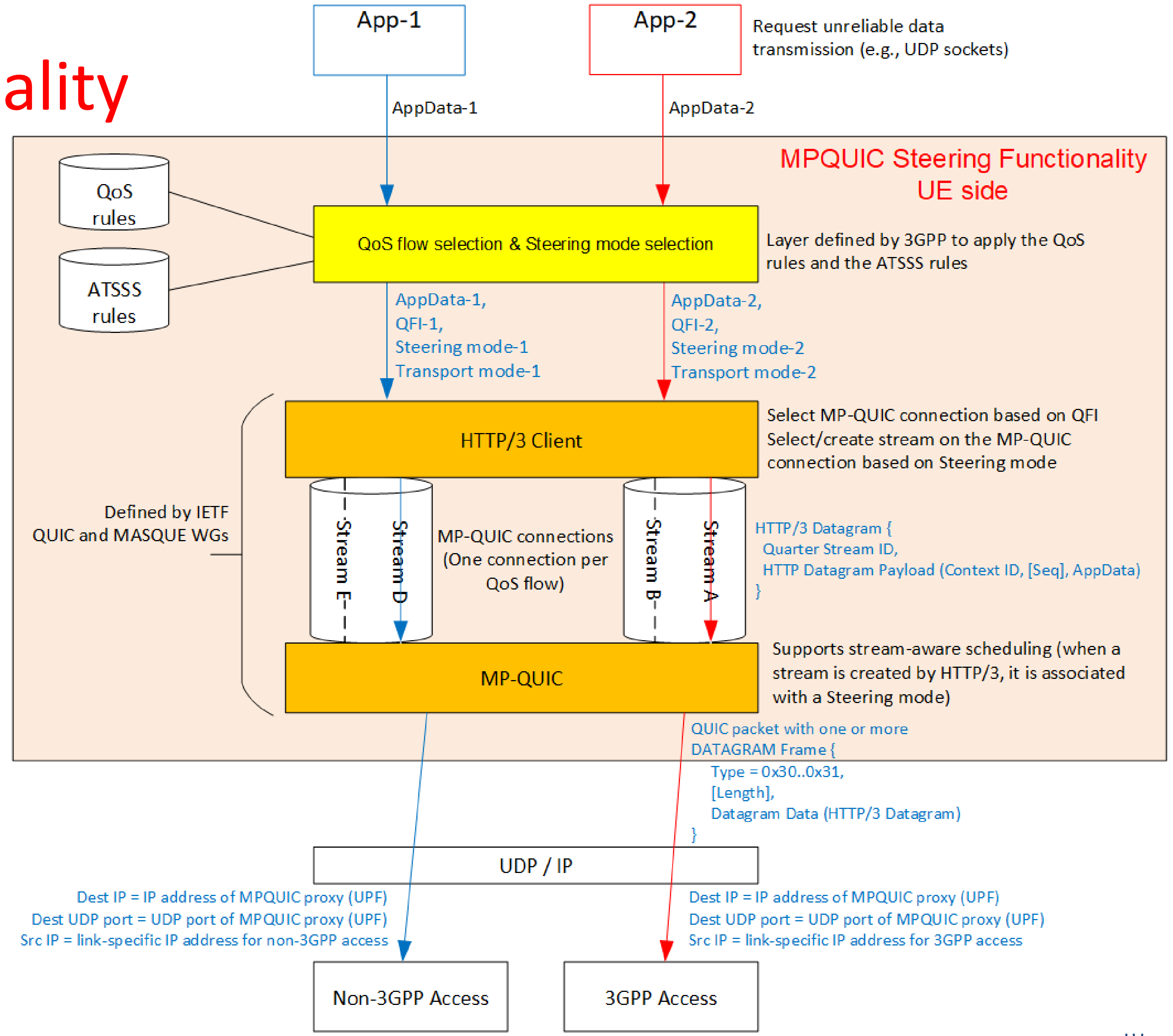
- An ATSSS rule indicates which transport mode to apply for the traffic matching this rule. The PCF selects the transport mode, e.g., based on whether re-ordering can be supported by the app or not.
- 3 transport modes considered:
  - **Datagram mode 2**: Supports unreliable QUIC transport without re-ordering.
    - Can be used for the traffic of apps that can perform re-ordering, such as apps using the QUIC protocol. Based on existing IETF specs and does not require additional work in 3GPP. Superior to ATSSS-LL as it supports congestion control, PLR and RTT measurements.
  - **Datagram mode 1**: Supports unreliable QUIC transport with re-ordering.
    - Can be used for the traffic of apps that cannot perform re-ordering. Based on existing IETF specs but requires additional work in 3GPP to define a new context ID and procedures for packet numbering and re-ordering (unless these procedures are left to the implementation).
  - **Stream Mode**: Supports reliable QUIC transport with re-ordering using Streams.
    - Can be used for the traffic of apps that cannot perform re-ordering. Based on existing IETF specs and does not require additional work in 3GPP.
- For apps that cannot support re-ordering, we should select either **Datagram mode 1** or **Stream mode**.
  - The **Datagram mode 1** may seem better for UDP flows but its performance compared to **Stream mode** is questionable, given that the re-ordering mechanism can introduce a lot of delay and jitter. The **Datagram mode 1** needs also a lot more work in 3GPP to be specified and will require additional support from the HTTP/3 layer, which can complicate implementations with no obvious advantage.
  - We tend to believe that the **Stream mode** is sufficient for Rel-18 and the **Datagram mode 1** can be avoided. However, more work is needed to validate this.

Conclusions & Proposed Actions

- The steering functionality defined in Rel-18 shall be able to support packet re-ordering.
- Whether packet re-ordering is needed for a data flow, is decided by PCF and is communicated to UE and UPF via the corresponding steering rules.
- For the MP-QUIC based solution:
  - It should be decided whether the **Stream mode** or the **Datagram mode 1** will be selected for supporting packet re-ordering.
  - If the **Datagram mode 1** is selected, it should be decided whether the re-ordering mechanism will be left to the implementation or whether it will be specified (and by which SDO: IETF or 3GPP?).
- The MP-DCCP based solution:
  - Supports "path sequencing, connection sequencing and has inherent latency information, i.e. all properties that are necessary for the packet receiving side, i.e. UE or UPF, to be able to do proper re-ordering".
  - What is missing is to determine whether the re-ordering mechanism will be left to the implementation or whether it will be specified (and by which SDO: IETF or 3GPP?). The proponents of MP-DCCP prefer to standardize the re-ordering mechanism.

Backup:
Overview of MPQUIC Steering functionality

- For more details, see solution #2.2 in TR 23.700-53



App-1
App-2
Request unreliable data transmission (e.g., UDP sockets)
AppData-1
AppData-2
MPQUIC Steering Functionality UE side
QoS rules
ATSSS rules
QoS flow selection & Steering mode selection
Layer defined by 3GPP to apply the QoS rules and the ATSSS rules
AppData-1, QFI-1, Steering mode-1 Transport mode-1
AppData-2, QFI-2, Steering mode-2 Transport mode-2
HTTP/3 Client
Select MP-QUIC connection based on QFI Select/create stream on the MP-QUIC connection based on Steering mode
Defined by IETF QUIC and MASQUE WGs
Stream E
Stream D
MP-QUIC connections (One connection per QoS flow)
Stream B
Stream A
HTTP/3 Datagram { Quarter Stream ID, HTTP Datagram Payload (Context ID, [Seq], AppData) }
MP-QUIC
Supports stream-aware scheduling (when a stream is created by HTTP/3, it is associated with a Steering mode)
QUIC packet with one or more DATAGRAM Frame { Type = 0x30..0x31, [Length], Datagram Data (HTTP/3 Datagram) }
UDP / IP
Dest IP = IP address of MPQUIC proxy (UPF)
Dest UDP port = UDP port of MPQUIC proxy (UPF)
Src IP = link-specific IP address for non-3GPP access
Dest IP = IP address of MPQUIC proxy (UPF)
Dest UDP port = UDP port of MPQUIC proxy (UPF)
Src IP = link-specific IP address for 3GPP access
Non-3GPP Access
3GPP Access