PDU SET IDENTIFICATION

TECHNICAL FIELD

The present disclosure relates to wireless communications, and more specifically to supporting PDU set identification when an end-to-end XR session is fully encrypted.

BACKGROUND

A wireless communications system may include one or multiple network communication devices, such as base stations, which may support wireless communications for one or multiple user communication devices, which may be otherwise known as user equipment (UE), or other suitable terminology. The wireless communications system may support wireless communications with one or multiple user communication devices by utilizing resources of the wireless communication system (e.g., time resources (e.g., symbols, slots, subframes, frames, or the like) or frequency resources (e.g., subcarriers, carriers, or the like)). Additionally, the wireless communications system may support wireless communications across various radio access technologies including third generation (3G) radio access technology, fourth generation (4G) radio access technology, fifth generation (5G) radio access technology, among other suitable radio access technologies beyond 5G (e.g., sixth generation (6G)).

The transport of multimedia application data (e.g., AR/VR/XR, interactive and streaming services, web applications) is often conveyed over Real Time Protocol (RTP). For supporting XR media over 3GPP, 3GPP defines extensions to the RTP headers allowing an XR media application server to provide additional information to assist the 3GPP network to identify a group of packets (PDU set identification) and deliver such packets with specific QoS requirements (e.g. PDU Set Delay Budget) to a UE. Such an approach requires the RTP headers to be sent unencrypted to allow a User Plane Function (UPF) within the 5G network to identify PDU set information. However, the industry is now focusing on fully encrypting the media packet including the protocol headers either by using solutions relying on QUIC (IETF RFC 9000) or by enhancing the RTP protocol with fully encrypted headers as described in RFC 9335.

Consequently there is a need for a solution to provide support for PDU-set identification when the end-to-end connection between the UE and the Application server is fully encrypted.

SUMMARY

An article “a” before an element is unrestricted and understood to refer to “at least one” of those elements or “one or more” of those elements. The terms “a,” “at least one,” “one or more,” and “at least one of one or more” may be interchangeable. As used herein, including in the claims, “or” as used in a list of items (e.g., a list of items prefaced by a phrase such as “at least one of” or “one or more of” or “one or both of”) indicates an inclusive list such that, for example, a list of at least one of A, B, or C means A or B or C or AB or AC or BC or ABC (i.e., A and B and C). Also, as used herein, the phrase “based on” shall not be construed as a reference to a closed set of conditions. For example, an example step that is described as “based on condition A” may be based on both a condition A and a condition B without departing from the scope of the present disclosure. In other words, as used herein, the phrase “based on” shall be construed in the same manner as the phrase “based at least in part on. Further, as used herein, including in the claims, a “set” may include one or more elements.

Some implementations of the method and apparatuses described herein may further include receiving a packet detection rule, PDR, including an indication to establish an encapsulation protocol with a server address when the network node detects that a packet is sent towards the server address, and configuration information to enable PDU set inspection and extract PDU set information from information contained within the encapsulation protocol when the network node receives a packet from the server address. The method may further include receiving a first uplink packet, inspecting the first uplink packet and identifying the first uplink packet as matching the PDR, and sending a session request to establish a connection using the encapsulation protocol. The method may further include receiving a first downlink packet, inspecting the first downlink packet and identifying the first downlink packet as matching the PDR.

The method may further include using the configuration information to perform PDU set inspection of the first downlink packet and to extract PDU set information and a second

downlink packet encapsulated within the first downlink packet, wherein the PDU set information is associated to the second downlink packet.

Alternatively or in addition, the method may comprise extracting a second downlink packet from the first downlink packet where the second downlink packet and protocol description of the first downlink packet match the PDR.

The method may further include constructing a third downlink packet including the second downlink packet as a payload; and sending the third downlink packet to a GPRS Tunneling Protocol, GTP, session, wherein the third downlink packet includes within a GTP-U header the extracted PDU set information or PDU set information derived from the extracted PDU set information.

Some implementations of the method and apparatuses described herein may further include, sending, to a second network node, a packet detection rule, PDR, including an indication to establish an encapsulation protocol with a server address when the second network node detects that a packet is sent towards the server address, and configuration information to enable PDU set inspection and extract PDU set information from information contained within the encapsulation protocol when the network node receives a packet from the server address.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a Core Network Extended Reality Media architecture.

Figure 2 illustrates a 1 byte Real Time Protocol header extension for Packet Data Unit Set marking.

Figure 3 illustrates a 2 byte Real Time Protocol header extension for Packet Data Unit Set marking.

Figure 4 is a flow diagram illustrating handling of Packet Data Unit Set to Quality of Service flow to data radio bearer mappings.

Figure 5 illustrates a Real Time Protocol and Real Time Control protocol stack over Internet Protocol networks.

Figure 6 illustrates Real Time Protocol and Secure Real Time Protocol packet formats and header information.

Figure 7 illustrates a Web Real Time Communication protocol stack over Internet Protocol networks.

Figure 8 illustrates Real Time Protocol and Secure Real Time Protocol header extension formats and syntax.

Figure 9 illustrates Transport Layer Security 1.3 based encryption and authentication of a Quick User Datagram Protocol Internet Connection (QUIC) Packet Data Unit (PDU).

Figure 10 illustrates a HTTP/3 datagram as a payload to a QUIC datagram frame in a User Datagram Protocol (UDP) proxying configuration running HTTP/3 extensions CONNECT proxying over QUIC and UDP.

Figure 11 illustrates an example of a wireless communications system in accordance with aspects of the present disclosure.

Figure 12 illustrates a general framework for supporting PDU set identification for fully encrypted media packets.

Figure 13 is a flow diagram illustrating a general procedure for supporting PDU set identification for fully encrypted media packets.

Figure 14 illustrates a framework for using Connect-UDP or HTTP/3 proxy for proxying fully encrypted media packets via N6.

Figure 15 is a flow diagram illustrating using HTTP/3 proxy for supporting PDU set identification for fully encrypted media packets.

Figure 16 illustrates an example of a network equipment (NE) 200 in accordance with aspects of the present disclosure.

Figure 17 illustrates a flowchart of a method performed by a UE in accordance with aspects of the present disclosure.

Figure 18 illustrates a flowchart of a further method performed by a UE in accordance with aspects of the present disclosure.

DETAILED DESCRIPTION

Many applications and services where multimedia flows are multiplexed under a single network application session (i.e., a 5-tuple containing a source IP address, a destination IP address, a source network port, a destination network port, and a protocol number as an identifier) relate to the domain of eXtended Reality (XR). As an example, an XR application based on WebRTC (Web Real-Time Communication), or alternatively, on Real-time Protocol (RTP)/ Secure Real-time Protocol (SRTP) protocol stack, may

contain one or more video streams and audio streams multiplexed with control and feedback metadata and application metadata (e.g., such as user pose information, user input actions etc.) over a single application data network session.

The XR Media (XRM) feature in 3GPP Release 18 at the core network (CN) level introduced the concept of a Packet Data Unit (PDU) Set to handle Quality of Service (QoS) requirements of XRM applications and streams with a better granularity beyond 5G Rel-17 QoS flow possibilities. As such, a PDU set is composed of one or more PDUs carrying the payload of one unit of information generated at the application level (e.g. a frame or video slice for XRM Services). I n some implementations all PDUs in a PDU Set are needed by the application layer to use the corresponding unit of information. In other implementations, the application layer can still recover parts or all of the information unit, when some PDUs are missing.

In addition, the PDU set is associated with QoS requirements in terms of delay budget and error rate:

- a PDU Set Delay Budget (PSDB) which defines an upper bound for the time that a PDU-Set may be delayed between a User Equipment (UE) and the N6 termination point at the User Plane Function (UPF). PSDB applies to the Downlink (DL) PDU-Set received by the UPF over the N6 interface, and to the Uplink (UL) PDU-Set sent by the UE, and respectively,
- a PDU Set Error Rate (PSER) which defines an upper bound for the rate of PDU-Sets (e.g. set of IP packets constituting a PDU-Set) that have been processed by the sender of a link layer protocol (e.g. Radio Link Control (RLC) in a Radio Access Network (RAN) of a 3GPP access) but where all of the PDUs in the PDU-Set are not successfully delivered by the corresponding receiver to the upper layer (e.g. PDCP in RAN of a 3GPP access), whereas the PSER is used to determine an upper bound for a rate of non-congestion-related packet losses.

An overview of the CN XRM architecture handling of PDU sets is depicted in Figure 1, and the general processing steps for downlink traffic in such an architecture are described with reference to Figure 1.

In step 1001, an Application Function (AF) provides QoS requirements for packets of a PDU set to a Policy Control Function (PCF) (for example, a PSDB and PSER) and information to identify the application (i.e. 5-tuple or application id). The AF may also include an importance parameter for a PDU set and information for the core network to identify packets belonging to a PDU set.

In step 1002, the PCF derives QoS rules for the XR application (for example, using a 5G QoS identifier (5QI) for XR media traffic) and specific QoS requirements for the PDU set and configures a Session Management Function (SMF). The PCF may include Policy and Charging Control (PCC) rules per importance of a PDU set (according to information received from the AF or based on operator configuration).

In step 1003, the SMF establishes a QoS flow according to the QoS rules established by the PCF and configures the UPF to route packets of the XR application to a QoS flow, and, in addition, to enable PDU set handling. The SMF also provides the QoS profile containing PDU set QoS requirements to the RAN via an Access and Mobility Function (AMF).

In step 1004, the UPF inspects the packets and determines packets belonging to a PDU set (e.g., based on a UPF implementation given, for instance by inspecting the RTP packet headers, or based on Application Server (AS)-marked PDU set information transmitted over RTP PDU Set header extensions). When the UPF detects packets of a PDU set the UPF marks the packets belonging to a PDU set within a General Packet Radio Service (GPRS) tunneling protocol user plane (GTP-U) header. The GTP-U header information includes a PDU set sequence number and the size of the PDU set. The UPF may also determine the importance of the PDU set either based on UPF implementation means, information provided by the AF or information provided as metadata from the application server. Based on the importance of the PDU set the UPF may route the traffic to a corresponding QoS flow (according to the rules received from the SMF) or include the importance of the PDU set within a GTP-U header.

In step 1005, a RAN identifies packets belonging to a PDU set (based on the GTP-U marking) and handles the packets of the PDU set according to the QoS requirements of the PDU set provided by the SMF. In one implementation the RAN node may use a

different radio bearer with a higher QoS requirement (according to the PDU set PSDB/PSER) to guarantee delivery of the packets of the PDU set, while using a different radio bearer according to the 5QI of the QoS flow for the non-PDU set packets. However, in XRM Release 18, [3GPP Technical Specification TS 23.501 (v18.2.2 – Jun 2023). System architecture for the 5G System (5GS)], once PDU set QoS integrated handling is enabled, the PSA UPF identifies PDUs that belong to PDU sets and determines for each PDU set, PDU set information sent over to a next generation radio access network (NG-RAN) in the GTP-U header. The PDU Set Information comprises:

- PDU Set Sequence Number.
- Indication of an End PDU of the PDU Set.
- PDU Sequence Number within a PDU Set.
- PDU Set Size in bytes.
- PDU Set Importance, which identifies the relative importance of a PDU Set compared to other PDU Sets within a QoS Flow.

The PDU Set information is then used by the NG-RAN for PDU Set based QoS handling. The NG-RAN may use Priority Levels as of TS 23.501, Clause 5.7.3.3, across QoS Flows and PDU Set Importance within a QoS Flow for PDU Set level packet discarding in presence of congestion.

It is also specified in TS 23.501 that the PSA UPF identifies PDUs that belong to PDU Sets and if the UPF receives a PDU that does not belong to a PDU Set based on Protocol Description for PDU Set identification (e.g., has not been marked with PDU Set information by the AS), then the UPF still maps the PDU to a PDU Set and determines the PDU Set Information as described above. This ensures that, for a QoS flow with PDU Set enabled, all the PDUs belong to a PDU Set. To this end, if a PDU Session Anchor (PSA) UPF receives a PDU that does not belong to a PDU Set, it is assumed that the UPF determines the PDU Set Importance value based in some embodiments on pre-configuration and in other embodiments on an Application Server/Application Function (AS/AF) signaled default importance.

The AS PDU Set information listed above is provided in some embodiments via a one-byte RTP Header Extension for the marking of PDU Sets and End of Bursts as defined by TS 26.522 illustrated in Figure 2. In other embodiments the AS may use a two-byte

RTP Header Extension for the marking of PDU Sets and End of Bursts as defined by TS 26.522 illustrated in Figure 3.

The semantics of the fields denoted in Figure 2 and Figure 3 of the RTP Header Extension for the marking of PDU Set and End of Bursts are:

- End PDU of the PDU Set [E] (1 bit field) is a flag set to 1 for the last PDU of the PDU Set and set to 0 for all other PDUs of the PDU Set.
- End of Data Burst [EDB] (up to 3 bits field) indicates the end of a Data Burst being set to a non-zero value when the end of data burst is present and 0 otherwise.
- PDU Set Importance [PSI] (4 bits field) indicates the importance of a PDU Set compared to other PDU Sets within the same QoS flow. Lower values indicate a higher importance PDU Set with the highest importance PDU Set of 0 and the lowest importance PDU Set of 15.
- PDU Set Sequence Number [PSSN] (10 bits field) encodes the sequence number of the PDU Set to which the current PDU belongs acting as a 10-bit numerical identifier for the PDU Set and wraps around at 1023.
- PDU Sequence Number within a PDU Set [PSN] (6 bits field) indicates the sequence number of the current PDU within the PDU Set. The PSN is set to 0 for the first PDU in the PDU Set and incremented monotonically for every PDU in the PDU set in order of transmission from the sender. PSN wraps around 63.
- PDU Set Size [PSSize] (24 bits field) indicates the total size of all PDUs of the PDU Set to which this PDU belongs. This field is optional and subject to an SDP signaling offer/answer negotiation, where the AS may indicate whether it will be able to provide the size of the PDU Set for that RTP stream. If not enabled, the field is not present. If enabled, but the AS is not able to determine the PDU Set Size for a particular PDU Set, it should set the value to 0 in all PDUs of that PDU Set. The PSSize indicates the size of a PDU Set including RTP/UDP/IP header encapsulation overhead of its corresponding PDUs. The PSSize is expressed in bytes.

Reciprocal processing is applicable to uplink communication, wherein the role of UPF packet inspection is taken by the UE which is expected to inspect packets, determine packets belonging to a PDU set, and signal accordingly the PDU set to the RAN for

scheduling and resource allocation corresponding to an associated Data Radio Bearer (DRB) capable of fulfilling the PDU set QoS requirements (i.e., PSDB and PSER). The low-level signaling mechanism associated with the UL UE-to-RAN information passing is dependent on the specification and implementations of RAN signaling procedures.

Depending on the QoS flow mappings and RAN procedures, several alternative PDU set to QoS flow to DRB mappings are possible given two distinct PDU sets with different PDU set attributes, such as PDU set importance. Figure 4 illustrates this, where 2 PDU sets of different importance and characteristics are mapped to QoS flows and respectively to DRBs. Consider in this example PDU set 1 to be of high importance with strict QoS requirements (i.e., PSDB, PSER etc.) and PDU set 2 to be of low importance with potentially lower QoS requirements (i.e., PSDB, PSER etc.) than PDU set 1. As illustrated in Figure 4, the PDU set to QoS to DRB flow can take the following instantiations depending on QoS flow policies and Layer 2 RAN procedures:

a) 1-to-1-to-1: wherein the separation of QoS flows and DRBs is complete between high and low importance PDU sets optimizing finely the radio and network resources on a per PDU set basis;
b) M-to-M-to-1: whereby the separation between high and low importance PDU sets is performed only at QoS flow level, whereas the same DRB is used for the over-the-air transmission of both PDU sets, which may lead to overprovisioning of radio resources for low importance PDU sets yet require a lower overhead of RAN complexity and management;
c) M-to-1-to-1: whereby there is no separation between the QoS flows and DRBs of different importance PDU sets and the higher importance PDU set QoS requirements are prioritised in handling the QoS management across both the CN and RAN; this may lead to overprovisioning of resources for low importance PDU sets in both the CN and RAN implementations but requires lower overhead and control within the 5GS QoS framework;
d) M-to-1-to-M: whereby there is no separation across the QoS flows between PDU set importance levels, yet distinct DRBs are used to cater for the individual requirements of the distinct importance levels; this compromises the QoS flow management complexity and uses PDU set information to filter the PDU sets on different DRBs in order to better match the QoS requirements at RAN level and optimize resource allocation according to individual PDU set needs.

The transport of multiplexed multimedia flows is preponderantly being addressed by WebRTC, or alternatively, RTP/SRTP, and QUIC. In Release 17, SA4 analyzed for instance the XR traffic model, cf. [3GPP Technical Report TR 26.928 (v17.0.0 – Apr 2022). Extended Reality (XR) in 5G], and concluded the QoS requirements in terms of delay budget, data rate and error rate necessary for a satisfactory experience at the application level. These led to 4 additional 5QIs for the 5GS XR QoS flows, as part of [TS 23.501, Table 5.7.4-1] as delay-critical GBR 5QIs valued 87-90. The latter are applicable to XR video streams and control metadata necessary to provide the immersive and interactive XR experiences.

XR video traffic is mainly composed of multiple DL/UL video streams of high resolution (e.g., at least 1080p dual-eye buffer), frames-per-second (e.g., 60+ fps) and high bandwidth (e.g., usually at least 20-30 Mbps) which need to be transmitted across a network with minimal delay (typically upper bounded by 15-20 ms) to maintain a reduced end-to-end application round-trip interaction delay. The latter requirements are of critical importance given the XR application dependency on cloud/edge processing (e.g., content downloading, viewport generation and configuration, viewport update, viewport rendering, media encoding/transcoding etc.). Furthermore, XR traffic also includes one or more audio streams, as well as user interaction metadata (e.g., user actions, user inputs, user pose), or alternatively, feedback and control metadata (actions, XR space metadata, as well as transport control and feedback messages over RTCP for instance). The traffic of immersive and interactive XR applications such as those described above often operate on real-time suited transport architectures and protocols. As part of the latter, the state of art is represented by the Real-time Transport Protocol (RTP) [RFC 3550 - RTP: A Transport Protocol for Real-Time Applications (ietf.org)], its securely provisioned Secure Real-time Transport Protocol (SRTP) [RFC 3711 - The Secure Real-time Transport Protocol (SRTP) (ietf.org)], and its web-targeted stack Web Real-Time Communications WebRTC [WebRTC 1.0: Real-Time Communication Between Browsers (w3.org)], respectively.

RTP is a media codec agnostic network protocol with application-layer framing used to deliver multimedia (e.g., audio, video etc.) data in real-time over IP networks, cf. [RFC 3550 - RTP]. It is used in conjunction with a sister protocol for control, i.e., Real-time

Transport Control Protocol (RTCP), to provide end-to-end features such as jitter compensation, packet loss and out-of-order delivery detection, synchronization, and source streams multiplexing. An overview of the RTP and RTCP stack is offered in Figure 5.

SRTP is a secured version of RTP, cf. [RFC 3711], providing encryption (mainly by means of payload confidentiality), message authentication and integrity protection (by means of PDU, i.e., headers and payload, signing), as well as replay attack protection. Similarly to RTP, the SRTP sister protocol is SRTCP. This provides the same functions to its RTCP counterpart. The RTP and SRTP header information share the same format as provided in Figure 6.

In “vanilla” SRTP versions, the RTP header information is still accessible but non-modifiable, whereas the payload is encrypted. These security provisions are illustrated in part in the right-hand side of Figure 6. Furthermore, the key exchange and additional security parameters necessary to use SRTP are based upon the Datagram Transport Layer Security (DTLS) key exchange procedure. SRTP is used for these reasons as the transport protocol for media in the WebRTC stack which ensures secure RTC multimedia communications over web browser interfaces.

An overview of the WebRTC stack is provided in Figure 7. As illustrated, an IP layer carries signaling from the data plane and the control plane. The data plane stack comprises functions for User Datagram Protocol (UDP), Interactive Connectivity Establishment (ICE), Datagram Transport Layer Security (DTLS), SRTP, SRTCP, media codecs, Quality Control and Stream Control Transmission Protocol (SCTP). ICE may use the Session Traversal Utilities for Network Address Translation (NAT) (STUN) protocol and Traversal Using Relays around NAT (TURN) to address real-time media content delivery across heterogeneous networks and NAT rules and firewalls. The SCTP data plane is mainly dedicated as an application data channel and may be non-time critical, whereas the SRTP based stack including elements of control, i.e., SRTCP, encoding, i.e., media codecs, and Quality of Service (QoS), i.e., Quality Control, is dedicated to time-critical transport.

The data plane is usually established over a single application session (i.e., a 5-tuple) multiplexing the transport of media (i.e., video, audio, haptic media codec payloads), of user plane quality control and feedback based on RTCP messages and of WebRTC data channels (i.e., application data such as user chat messages, notifications, user inputs, user actions, user pose etc.) over SCTP/DTLS.

The RTP/SRTP fixed header information and complete header information (including header extensions) is as follows as per [RFC 3550], [RFC 3711]:

Fixed Header Info

- V – 2 bits indicating the protocol version used.
- P – 1 bit field indicating that one or more zero-padded octets at the end of the payload are present, whereby, among others, the padding may be used for fixed-sized encrypted blocks or for carrying multiple RTP/SRTP packets over lower layer protocols.
- X – 1 bit indicating that the standard fixed RTP/SRTP header will be followed by an RTP header extension usually associated with a particular data/profile that will carry more information about the data (e.g., as per [https://www.rfc-editor.org/rfc/rfc8285.html] generic RTP header extensions framework for RTP/SRTP)
- CC – 4 bits indicating number of contributing media sources (CSRC) that follow the fixed header
- M – 1 bit intended to mark an information frame boundary in the packet stream, whose behavior is exactly specified by RTP profiles (e.g., H.264, H.265, H.266, AV1 etc.)
- PT – 7 bits indicating the payload type, which in case of video profiles is dynamic and negotiated by means of SDP (e.g., 96 for H.264, 97 for H.265, 98 for AV1 etc.)
- Sequence number – 16 bits indicating the sequence number which increments by one with each RTP data packet sent over a session
- Timestamp – 32 bits indicating timestamp in ticks of the payload type clock reflecting the sampling instant of the first octet of the RTP data packet (associated for video stream with a video frame), whereas the first timestamp of the first RTP packet is selected at random

- Synchronization Source (SSRC) identifier – 32 bits field indicating a random identifier for the source of a stream of RTP packets forming a part of the same timing and sequence number space, such that a receiver may group packets based on synchronization source for playback.
- Contributing Source (CSRC) identifier – list of up to 16 CSRC items of 32 bits each given the amount of CSRC mixed by RTP mixers within the current payload as signaled by the CC bits; the list identifies the contributing sources for the payload contained in this packet given the SSRC identifiers of the contributing sources.

Complete Header Information (incl. header extensions)

- RTP header extension – a variable length field present if the X bit is marked; the header extension is appended to the RTP fixed header information after the CSRC list if present; the RTP header extension is 32-bit aligned and formed of the following fields:
- A 16-bit extension identifier defined by a profile and usually negotiated and determined via the Session Description Protocol (SDP) signaling mechanism
- A 16-bit length field describing the extension header length in 32-bits multiples excluding the first 32 bits corresponding to the 16 bits extension identifier and the 16 bits length fields itself
- A 32-bit aligned header extension raw data field formatted according to some RTP header extension identifier specified format.

The RTP header extension format and syntax are similar to those of SRTP, commonly illustrated in Figure 8. In addition, in both RTP and SRTP, only one RTP extension header may be appended to the fixed header information, cf. [RFC 3550]. However, for both RTP and SRTP, extensions to the base protocols exist to allow for multiple RTP header extensions of predetermined types to be appended to the fixed header information of the protocols, as per [RFC 8285].

In some embodiments, RTP header extensions produced at the source may be ignored by the destination endpoints that do not have the knowledge to interpret and process the RTP header extensions transmitted by the source endpoint.

QUIC (RFC 9000, [RFC 9000 - QUIC: A UDP-Based Multiplexed and Secure Transport (ietf.org)]), is a new transport protocol meant to replace TCP. Originally, QUIC was designed to benefit HTTP/3 applications by avoiding application-layer and transport-layer head-of-line blocking directed at improving the performance of HTTP/2 over TCP.

QUIC is a multiplexed and framed protocol implemented in the user space of Operating Systems running on top of UDP transport layer. QUIC is designed to be a connection-based stream-oriented transport with reliable transmission based on stream-aware acknowledgements. A list of basic QUIC frame types registered with IANA is provided by the RFC 9000, Clause 12.4, and replicated for convenience below in Table 1.

Value (Frame Type ID)	**Frame Type Name**	**Specification**
0x00	PADDING	[RFC9000, Section 19.1]
0x01	PING	[RFC9000, Section 19.2]
0x02-0x03	ACK	[RFC9000, Section 19.3]
0x04	RESET_STREAM	[RFC9000, Section 19.4]
0x05	STOP_SENDING	[RFC9000, Section 19.5]
0x06	CRYPTO	[RFC9000, Section 19.6]
0x07	NEW_TOKEN	[RFC9000, Section 19.7]
0x08-0x0f	STREAM	[RFC9000, Section 19.8]
0x10	MAX_DATA	[RFC9000, Section 19.9]
0x11	MAX_STREAM_DATA	[RFC9000, Section 19.10]
0x12-0x13	MAX_STREAMS	[RFC9000, Section 19.11]
0x14	DATA_BLOCKED	[RFC9000, Section 19.12]
0x15	STREAM_DATA_BLOCKED	[RFC9000, Section 19.13]
0x16-0x17	STREAMS_BLOCKED	[RFC9000, Section 19.14]
0x18	NEW_CONNECTION_ID	[RFC9000, Section 19.15]
0x19	RETIRE_CONNECTION_ID	[RFC9000, Section 19.16]
0x1a	PATH_CHALLENGE	[RFC9000, Section 19.17]
0x1b	PATH_RESPONSE	[RFC9000, Section 19.18]
0x1c-0x1d	CONNECTION_CLOSE	[RFC9000, Section 19.19]
0x1e	HANDSHAKE_DONE	[RFC9000, Section 19.20]
0x30-0x31	DATAGRAM	[RFC9221]

Table 1: QUIC frame types according to RFC 9000 and RFC 9221.

Furthermore, QUIC has connection and stream-aware flow control, and congestion control transport procedures for paced Internet transport capable of establishing and exploiting the available bandwidth over a connection. QUIC also relies on TLS 1.3 as per RFC 9001, [RFC 9001 - Using TLS to Secure QUIC (ietf.org)], to encrypt all non-essential protocol elements, including large portions of QUIC headers and QUIC frames (including QUIC frames' headers), and respectively, to authenticate through a QUIC authentication message (QUIC mac) the critical QUIC headers (i.e., the header form, fixed bit, spin bit, and the QUIC destination connection ID header fields) alongside the encrypted content. The encrypted and authenticated content of a QUIC packet is shown in Figure 9, whereby the QUIC payload is embodied to carry an ACK frame, a flow control frame, and a stream frame corresponding to HTTP data.

Besides the reliable transport mode based on stream frames, QUIC also offers a datagram frame extension, i.e., 0x30-0x31 defined in RFC 9221. The QUIC datagrams may be IDed and mapped to different logic flows but this is up to the application layers rather than the QUIC protocol. In essence, datagrams are unreliable data containers with minimum overhead, i.e., a header indicating just the frame type (0x30, 0x31) and optionally the frame length (just for datagrams of type 0x31) to allow for datagrams multiplexing with other QUIC frames. Furthermore, datagrams do not contribute to flow-control as they are inherently unreliable, and a receiver may drop them in case it cannot accommodate a sender's rate. Nonetheless, datagrams are acknowledged, yet their acknowledgements (ACKs) do not require retransmission by a sender in case of lost QUIC packets and as such multiplexing datagrams and stream frames over the same QUIC packet is generally not a good sender strategy. However, datagrams are subject to congestion control by the sender and as such they may be impacted by a sender reacting to congestion events which may impact delaying datagram sending up to congestion controller decision or to an optional maximum sending time window upon which the expiration of a datagram may happen as configured by an application exercising paced transmission.

QUIC was developed to benefit HTTP application-layer protocol, but it has evolved into a fully-fledged general transport protocol that can serve different applications and associated protocols besides the web. For instance, QUIC has been regarded as an efficient, multiplexed transport for multimedia applications and proxying. To this end, QUIC is studied and evolved further in the Internet Engineering Task Force (IETF) in the working groups of Media over QUIC (MoQ) and Multiplexed Application Substrate over QUIC Encryption (MASQUE). QUIC based HTTP/3 is seen as a replacement of HTTP/2 for DASH and adaptive low-latency streaming, but also as an end-to-end secure transport mechanism for RTP in RTP over QUIC (RoQ), as per [draft-ietf-avtcore-rtp-over-quic-05].

QUIC's multiplexing-aware transport and its datagram frames extension make it desirable to deploy in proxying scenarios whereby QUIC is used to multiplex heterogeneous flows over a single connection (i.e., corresponding to a single 5-tuple) in an encrypted manner end-to-end. This allows endpoints to cater for traffic multiplexing and potential prioritization, QoS flow control without any intermediary middleboxes interventions given the end-to-end integrity and confidentiality protection offered out-of-the-box by QUIC.

QUIC-based proxying relies on the well-established CONNECT HTTP logic existent for TCP sockets. In MASQUE IETF, recently HTTP/3 extensions to CONNECT over QUIC and UDP has been proposed to allow for UDP proxying as per RFC 9298, [RFC 9298: Proxying UDP in HTTP (rfc-editor.org)] and for IP proxying as per [draft-ietf-masque-connect-ip-13 - Proxying IP in HTTP]. To this end the HTTP/3 capsule framing has been defined to match the concept of datagrams in QUIC as per [RFC 9297: HTTP Datagrams and the Capsule Protocol (rfc-editor.org)]. Defining a HTTP/3 datagram capsule framing allows HTTP/3 datagrams to be encapsulated natively within both QUIC datagrams for high-performance unreliable proxying but also over QUIC streams (in case QUIC datagrams are not available or desirable to a proxying implementation) for reliable proxying, respectively. Coupled with the CONNECT HTTP extensions, this enables end-to-end encrypted and multiplexed HTTP/3 CONNECT proxying over QUIC and UDP of TCP, UDP and IP based protocols between any proxy server and a client.

The HTTP/3 datagram used as payload to QUIC datagrams are illustrated in Figure 10. A HTTP/3 datagram has a low overhead consisting of a Quarter Stream ID. The concept of a Stream ID from QUIC streams is reused to be able to differentiate and multiplex among different types of QUIC datagrams and expose multiplexing control to the QUIC endpoints's applications. As such the Quarter Stream ID is a variable-length integer that contains the value of the client-initiated bidirectional flow that this datagram is associated with divided by four. The division by four, stems from the fact that HTTP requests are sent on client-initiated bidirectional streams, which would have in an embodiment Stream IDs that are divisible by four. In this embodiment, the largest permitted QUIC Stream ID value is 2^62-1, so the largest permitted value of the Quarter Stream ID field is 2^60-1 as per RFC 9297.

In some embodiments, as QUIC datagrams may be unavailable as a transport protocol, RFC 9297, defines a capsule protocol that can be used. The Capsule Protocol is a sequence of type-length-value tuples out of which HTTP upgrade tokens can choose to use. It allows endpoints to reliably communicate request-related information end-to-end on HTTP request streams, even in the presence of HTTP intermediaries. As such, in some embodiments the Capsule Protocol can be used to exchange HTTP/3 datagrams (i.e., the payload carried in Figure 10), which is necessary when HTTP is running over a transport that does not support the QUIC datagram frame. In some embodiments, this can run atop of QUIC stream frames to provide a reliable proxying as HTTP/3 extensions CONNECT proxying over QUIC streams, as per RFC 9297.

Up to end of Release 18 Stage 2 normative work for XRM in TS 23.501, the 5GS specify minimal support for multimodal flows in terms of providing a common multimodal service ID for the PCF to correlate QoS policy and rules for multiple flows over a common PDU Session as per [TS 23.501, Clause 5.37.2]. However, no architectural features in the CN (e.g., at the UPF) or in the RAN for dedicated support of multimodal flows and differentiated traffic support are available yet. Furthermore, from the perspective of 5G multimedia, interactive and immersive architecture, the interface between the AS and the AF, i.e., M3 in the context of 5G Media Streaming TS 26.501, and respectively, RTC-3 in the context of 5G Real-time Communications TS 26.506, is up to the implementation and out of scope of 5GS within the timeline of current Release 18. This is set to change given the extension of AS-driven provisioning of 5GS service support regarding dynamic

policies, service adaptation, as well as requests for optimized QoS handling of multimedia related applications.

Multimodal flow support within the 5GS falls also in the latter category whereby AS-assistance information towards the 5GS, such as protocol description of multimodal services and/or applications, is further needed. Currently, minimal specification support is offered over N5 interface at the PCF via the Npcf_PolicyAuthorization service (cf. TS 29.514, [3GPP Technical Specification TS 29.514 (v18.3.0 – Sep 2023). 5G System; Policy Authorization Service; Stage 3.]), or alternatively, over N33 interface at the NEF via the Nnef_AFSessionWithQoS service (cf. TS 29.122, [3GPP Technical Specification TS 29.112 (v18.3.0 – Sep 2023). T8 reference point for Northbound APIs.]) to provide a multimodal service ID, a protocol description limited only to PDU Set marked flows and a list of QoS requirements for multiple IP data flows associated with the same multimodal service ID.

Gaps are therefore present in supporting multimodal flows served over the same flow whereby at least of the multimodal protocol data flows are encrypted and/or not marked with PDU Set information by the AS.

Aspects of the present disclosure are described in the context of a wireless communications system.

Figure 11 illustrates an example of a wireless communications system 100 in accordance with aspects of the present disclosure. The wireless communications system 100 may include one or more NE 102, one or more UE 104, and a core network (CN) 106. The wireless communications system 100 may support various radio access technologies. In some implementations, the wireless communications system 100 may be a 4G network, such as an LTE network or an LTE-Advanced (LTE-A) network. In some other implementations, the wireless communications system 100 may be a NR network, such as a 5G network, a 5G-Advanced (5G-A) network, or a 5G ultrawideband (5G-UWB) network. In other implementations, the wireless communications system 100 may be a combination of a 4G network and a 5G network, or other suitable radio access technology including Institute of Electrical and Electronics Engineers (IEEE) 802.11 (Wi-Fi), IEEE 802.16 (WiMAX), IEEE 802.20. The wireless communications system 100 may

support radio access technologies beyond 5G, for example, 6G. Additionally, the wireless communications system 100 may support technologies, such as time division multiple access (TDMA), frequency division multiple access (FDMA), or code division multiple access (CDMA), etc.

The one or more NE 102 may be dispersed throughout a geographic region to form the wireless communications system 100. One or more of the NE 102 described herein may be or include or may be referred to as a network node, a base station, a network element, a network function, a network entity, a radio access network (RAN), a NodeB, an eNodeB (eNB), a next-generation NodeB (gNB), or other suitable terminology. An NE 102 and a UE 104 may communicate via a communication link, which may be a wireless or wired connection. For example, an NE 102 and a UE 104 may perform wireless communication (e.g., receive signaling, transmit signaling) over a Uu interface.

An NE 102 may provide a geographic coverage area for which the NE 102 may support services for one or more UEs 104 within the geographic coverage area. For example, an NE 102 and a UE 104 may support wireless communication of signals related to services (e.g., voice, video, packet data, messaging, broadcast, etc.) according to one or multiple radio access technologies. In some implementations, an NE 102 may be moveable, for example, a satellite associated with a non-terrestrial network (NTN). In some implementations, different geographic coverage areas 112 associated with the same or different radio access technologies may overlap, but the different geographic coverage areas may be associated with different NE 102.

The one or more UE 104 may be dispersed throughout a geographic region of the wireless communications system 100. A UE 104 may include or may be referred to as a remote unit, a mobile device, a wireless device, a remote device, a subscriber device, a transmitter device, a receiver device, or some other suitable terminology. In some implementations, the UE 104 may be referred to as a unit, a station, a terminal, or a client, among other examples. Additionally, or alternatively, the UE 104 may be referred to as an Internet-of-Things (IoT) device, an Internet-of-Everything (IoE) device, or machine-type communication (MTC) device, among other examples.

A UE 104 may be able to support wireless communication directly with other UEs 104 over a communication link. For example, a UE 104 may support wireless communication directly with another UE 104 over a device-to-device (D2D) communication link. In some implementations, such as vehicle-to-vehicle (V2V) deployments, vehicle-to-everything (V2X) deployments, or cellular-V2X deployments, the communication link 114 may be referred to as a sidelink. For example, a UE 104 may support wireless communication directly with another UE 104 over a PC5 interface.

An NE 102 may support communications with the CN 106, or with another NE 102, or both. For example, an NE 102 may interface with other NE 102 or the CN 106 through one or more backhaul links (e.g., S1, N2, N3, or network interface). In some implementations, the NE 102 may communicate with each other directly. In some other implementations, the NE 102 may communicate with each other indirectly (e.g., via the CN 106). In some implementations, one or more NE 102 may include subcomponents, such as an access network entity, which may be an example of an access node controller (ANC). An ANC may communicate with the one or more UEs 104 through one or more other access network transmission entities, which may be referred to as a radio heads, smart radio heads, or transmission-reception points (TRPs).

The CN 106 may support user authentication, access authorization, tracking, connectivity, and other access, routing, or mobility functions. The CN 106 may be an evolved packet core (EPC), or a 5G core (5GC), which may include a control plane entity that manages access and mobility (e.g., a mobility management entity (MME), an access and mobility management functions (AMF)) and a user plane entity that routes packets or interconnects to external networks (e.g., a serving gateway (S-GW), a Packet Data Network (PDN) gateway (P-GW), or a user plane function (UPF)). In some implementations, the control plane entity may manage non-access stratum (NAS) functions, such as mobility, authentication, and bearer management (e.g., data bearers, signal bearers, etc.) for the one or more UEs 104 served by the one or more NE 102 associated with the CN 106.

The CN 106 may communicate with a packet data network over one or more backhaul links (e.g., via an S1, N2, N3, or another network interface). The packet data network may include an application server. In some implementations, one or more UEs 104 may

communicate with the application server. A UE 104 may establish a session (e.g., a protocol data unit (PDU) session, or the like) with the CN 106 via an NE 102. The CN 106 may route traffic (e.g., control information, data, and the like) between the UE 104 and the application server using the established session (e.g., the established PDU session). The PDU session may be an example of a logical connection between the UE 104 and the CN 106 (e.g., one or more network functions of the CN 106).

In the wireless communications system 100, the NEs 102 and the UEs 104 may use resources of the wireless communications system 100 (e.g., time resources (e.g., symbols, slots, subframes, frames, or the like) or frequency resources (e.g., subcarriers, carriers)) to perform various operations (e.g., wireless communications). In some implementations, the NEs 102 and the UEs 104 may support different resource structures. For example, the NEs 102 and the UEs 104 may support different frame structures. In some implementations, such as in 4G, the NEs 102 and the UEs 104 may support a single frame structure. In some other implementations, such as in 5G and among other suitable radio access technologies, the NEs 102 and the UEs 104 may support various frame structures (i.e., multiple frame structures). The NEs 102 and the UEs 104 may support various frame structures based on one or more numerologies.

One or more numerologies may be supported in the wireless communications system 100, and a numerology may include a subcarrier spacing and a cyclic prefix. A first numerology (e.g., μ=0) may be associated with a first subcarrier spacing (e.g., 15 kHz) and a normal cyclic prefix. In some implementations, the first numerology (e.g., μ=0) associated with the first subcarrier spacing (e.g., 15 kHz) may utilize one slot per subframe. A second numerology (e.g., μ=1) may be associated with a second subcarrier spacing (e.g., 30 kHz) and a normal cyclic prefix. A third numerology (e.g., μ=2) may be associated with a third subcarrier spacing (e.g., 60 kHz) and a normal cyclic prefix or an extended cyclic prefix. A fourth numerology (e.g., μ=3) may be associated with a fourth subcarrier spacing (e.g., 120 kHz) and a normal cyclic prefix. A fifth numerology (e.g., μ=4) may be associated with a fifth subcarrier spacing (e.g., 240 kHz) and a normal cyclic prefix.

A time interval of a resource (e.g., a communication resource) may be organized according to frames (also referred to as radio frames). Each frame may have a duration,

for example, a 10 millisecond (ms) duration. In some implementations, each frame may include multiple subframes. For example, each frame may include 10 subframes, and each subframe may have a duration, for example, a 1 ms duration. In some implementations, each frame may have the same duration. In some implementations, each subframe of a frame may have the same duration.

Additionally or alternatively, a time interval of a resource (e.g., a communication resource) may be organized according to slots. For example, a subframe may include a number (e.g., quantity) of slots. The number of slots in each subframe may also depend on the one or more numerologies supported in the wireless communications system 100. For instance, the first, second, third, fourth, and fifth numerologies (i.e., μ=0, μ=1, μ=2, μ=3, μ=4) associated with respective subcarrier spacings of 15 kHz, 30 kHz, 60 kHz, 120 kHz, and 240 kHz may utilize a single slot per subframe, two slots per subframe, four slots per subframe, eight slots per subframe, and 16 slots per subframe, respectively. Each slot may include a number (e.g., quantity) of symbols (e.g., OFDM symbols). In some implementations, the number (e.g., quantity) of slots for a subframe may depend on a numerology. For a normal cyclic prefix, a slot may include 14 symbols. For an extended cyclic prefix (e.g., applicable for 60 kHz subcarrier spacing), a slot may include 12 symbols. The relationship between the number of symbols per slot, the number of slots per subframe, and the number of slots per frame for a normal cyclic prefix and an extended cyclic prefix may depend on a numerology. It should be understood that reference to a first numerology (e.g., μ =0) associated with a first subcarrier spacing (e.g., 15 kHz) may be used interchangeably between subframes and slots.

In the wireless communications system 100, an electromagnetic (EM) spectrum may be split, based on frequency or wavelength, into various classes, frequency bands, frequency channels, etc. By way of example, the wireless communications system 100 may support one or multiple operating frequency bands, such as frequency range designations FR1 (410 MHz – 7.125 GHz), FR2 (24.25 GHz – 52.6 GHz), FR3 (7.125 GHz – 24.25 GHz), FR4 (52.6 GHz – 114.25 GHz), FR4a or FR4-1 (52.6 GHz – 71 GHz), and FR5 (114.25 GHz – 300 GHz). In some implementations, the NEs 102 and the UEs 104 may perform wireless communications over one or more of the operating frequency bands. In some implementations, FR1 may be used by the NEs 102 and the UEs 104,

among other equipment or devices for cellular communications traffic (e.g., control information, data). In some implementations, FR2 may be used by the NEs 102 and the UEs 104, among other equipment or devices for short-range, high data rate capabilities. FR1 may be associated with one or multiple numerologies (e.g., at least three numerologies). For example, FR1 may be associated with a first numerology (e.g., μ=0), which includes 15 kHz subcarrier spacing; a second numerology (e.g., μ=1), which includes 30 kHz subcarrier spacing; and a third numerology (e.g., μ=2), which includes 60 kHz subcarrier spacing. FR2 may be associated with one or multiple numerologies (e.g., at least 2 numerologies). For example, FR2 may be associated with a third numerology (e.g., μ =2), which includes 60 kHz subcarrier spacing; and a fourth numerology (e.g., μ=3), which includes 120 kHz subcarrier spacing.

Figure 12 illustrates an example of a network architecture in accordance with aspects of the present disclosure.

The solution proposes to have a framework where fully encrypted packets from an application server (or UE) are tunneled via an encapsulation protocol between the UPF and the Application Server (or UE) over an N6 reference point. The main steps of the solution are described with reference to Figure 13.

In step 1301 an AF requests establishment of an AF session with a QoS flow by invoking an Nnef_AFSessionWithQoS Create service operation as described in clause 4.15.6.6 of 3GPP TS 23.502. The request includes PDU Set QoS parameters for the service provided, for example, an XR service. The AF additionally includes information to enable an encapsulation protocol connection between the UPF and the AS and the address of the AS where the UPF can establish the encapsulation protocol session. In one embodiment the protocol description includes additional information about the encapsulation protocol and the server address. For example, the AF includes the address of the XR Video application server where the UPF will need to establish a connection using the encapsulation protocol. It is assumed that the 3rd party provider of the AF and the 3GPP network have an service level agreement on the type of encapsulation protocol to use between the UPF and AS over the N6 reference point. In another embodiment the server address is provided via a different parameter. The AF also includes additional information indicating to the 3GPP network that the UPF can

identify PDU-set information by inspecting the headers or the payload of an encapsulation protocol. Although not shown in Figure 13, the NEF authorizes the request and forwards the request to the PCF by invoking an Npcf_PolicyAuthorization_Create request including the information provided by the AF.

In step 1302, a PCF in the 3GPP network creates PCC rules taking into account the PDU Set QoS parameters as described in clause 6.1.3.22 of 3GPP TS 23.503. The PCC rules also include the encapsulation protocol details and server address. The PCC rules may include all of the information provided by the AF in step 1301.

In step 1303, the SMF creates Packet Detection Rules (N4 rules) for the UPF based on the PCC rules, including uplink and downlink Packet Detection Rules. The Uplink Packet Detection Rule includes an indication to the UPF to establish an encapsulation protocol session with the server address (provided by the AF) when the UPF detects that a packet is sent towards the server address, i.e. for uplink packets sent from a specific UE (or any UE) to a specific application server, the UPF is to establish a connection using an encapsulation protocol and route the packet via the encapsulation protocol to the Application Server address. The downlink Packet Detection Rule (PDR) includes configuration information to enable PDU set inspection and extract PDU-set information from information contained within the encapsulation protocol. Specifically, for downlink packets received by the UPF over the N6 configuration, the downlink PDR includes information indicating that the UPF is to: (i) extract an encapsulated UDP packet from the received IP packet; (ii) enable PDU set identification and retrieve PDU set information from information contained within the encapsulation protocol; (iii) route the de-encapsulated UDP packet over a QoS flow with PSDB requirements within GTP signalling towards the RAN.

In step 1304, the N4 rules are sent to the UPF.

In step 1305, an application in the UE is triggered to connect to the AS, and in step 1306, the application in the UE sends an application packet via uplink over the 3GPP network.

In step 1307, the UPF inspects the packet and determines that there is a matching PDR rule indicating that an encapsulation protocol session is to be established with a server address and that the uplink packet is to be routed via the encapsulation protocol.

In step 1308, the UPF sends a session request to establish a connection with the AS using the encapsulation protocol procedure.

In step 1309, the AS acknowledges the request to establish a connection.

In step 1310, the UPF encapsulates the uplink packet within the encapsulation protocol header or payload, and in step 1311, the session packet is sent to the AS via the N6 connection.

In step 1312, further downlink and uplink packets may be routed to fully encrypt the connection between the UE and AS.

In step 1313, the AS identifies PDUs belonging to a PDU set, and adds PDU set information within the encapsulation protocol.

In step 1314, the AS send a UDP packet via the encapsulation protocol to the UPF. In sending downlink packets via the encapsulation protocol, the AS includes additional PDU set information provided in-band within the encapsulation protocol to assist the UPF in identifying PDU set information. Embodiments describing encapsulation protocols are provided in section 2.

In step 1315, the UPF extracts the UDP packet and determines the PDU set information using the PDU set information provided within the encapsulation protocol. The UPF then routes the packet via a QoS flow with PSDB requirements based on the N4 rules and includes within the encapsulation protocol header or payload PDU set information for each PDU.

In one embodiment, the “encapsulation protocol” between the UPF and AS may be realised using HTTP/3 proxy by leveraging the Connect-UDP protocol which proxies UDP packets in HTTP as per RFC 9298. The architecture for such an embodiment is

shown in Figure 14 and supports two alternatives. In alternative 1, the content server is collocated with the HTTP/3 proxy server, whilst in alternative 2 the content server is separate to the Proxy server. With a Connect-UDP procedure the proxy can forward packets to the content server based on the information included within the HTTP/3 session. In alternative 2, a new application programming interface (API) is needed between the content server and the proxy server to include information about PDU set information .

Detailed steps of the procedure are shown in Figure 15. In step 1501, an AF requests establishment of an AF session with QoS by invoking the Nnef_AFSessionWithQoS Create service operation as described in clause 4.15.6.6 of 3GPP TS 23.502 including PDU Set QoS parameters for the XR service. The AF additionally includes information to enable HTTP/3 proxy connect-UDP session between the UPF and AS and the address (IP address and port) of the HTTP/3 server where the UPF can establish the HTTP/3 session. In one embodiment the protocol description includes additional information about the HTTP/3 proxy and HTTP/3 proxy address. Additional information indicating to the 3GPP network that the UPF needs to establish a QUIC/Connect-UDP session with a HTTP/3 proxy and indicating that the UPF can identify PDU-set information by inspecting PDU-set information contained within QUIC datagrams may be included. The AF also includes the address of a HTTP/3 proxy server where the UPF will need to establish a QUIC session. In another embodiment the server address is provided via a different parameter.

Although not shown in Figure 15, the NEF authorizes the request and forwards the request to the PCF by invoking an Npcf_PolicyAuthorization_Create request including the information provided by the AF. It is assumed that the 3rd party provider of the AF and the 3GPP network have SLA agreement on the type of encapsulation protocol to use between the UPF and AS over N6 reference point.

In step 1502, the PCF creates PCC rules taking into account the PDU Set QoS parameters as described in clause 6.1.3.22 of 3GPP TS 23.503. The PCC rule also includes the HTTP/3 proxy connect-UDP protocol information and server address. The PCC rules may include all of the information provided by the AF in step 1501.

In step 1503, the SMF creates Packet Detection Rules (N4 rules) for the UPF. The Uplink Packet Detection Rule includes an indication to the UPF that the UPF is to establish an HTTP/3 connect-UDP protocol session with an HTTP/3 proxy server address (provided by the AF) when the UPF detects that a packet is sent towards a content server address and the origin address is the address of the UE (the origin address of the UE may be omitted). The downlink Packet Detection rule includes configuration information to enable PDU set inspection and extract PDU-set information from information from the received QUIC Datagram frame. Specifically , the N4 rules are based on the PCC rules and include uplink and downlink Packet Detection Rules. The uplink PDR indicates to the UPF that for uplink packets sent from a specific UE (or any UE) to a specific application server, the UPF is to establish a QUIC/Connect UDP session and route the packet within a QUIC Datagram to the Application Server address. The downlink PDRs received from the QUIC connection by the UPF over N6 include configuration information indicating that the UPF is to: (i) extract the UDP packet from the received QUIC Datagram;l (ii) enable PDU set identification and retrieve PDU set information from the QUIC Datagram. The QUIC Datagram includes a new Context ID that contains PDU-set information for the extracted UDP packet; route the extracted UDP packet over a QoS flow with PSDB requirements towards the RAN.

In step 1504, the N4 rules are sent to the UPF.

In step 1505, an application in the UE is triggered to connect to the AS, and in step 1506, the application in the UE sends an application (uplink) packet towards a content server address via uplink over the 3GPP network.

In step 1507, the UPF inspects the uplink packet and determines that there is a matching PDR rule (i.e. the uplink packet matches the uplink PDR provided by the SMF) indicating that an HTTP/3 session is to be established with a specific HTTP/3 proxy server address and that the uplink packet is to be routed via the HTTP/3 session.

In step 1508, the UPF (HTTP/3 client in the UPF) sends a CONNECT request using CONNECT-UDP protocol to the proxy with the target address (IP address/port) of the content server.

In step 1509, the server acknowledges the request by sending a STATUS-200 message.

In step 1510, the UPF includes the UDP packet received from the UE within a QUIC Datagram and, in step 1511, the QUIC Datagram is sent to the HTTP/3 proxy via N6. The proxy extracts the UDP packets and forwards the UDP packets to the content server address according to the target address included in step 1508 within the Connect-UDP protocol. The content server address may be collocated with the HTTP/3 proxy

In step 1512, further downlink and uplink packets may be routed to fully encrypt the connection between the UE and content server

In step 1513, the content server/HTTP/3 proxy determines PDUs belonging to PDU set and determines PDU set information. The PDU set information is added within QUIC datagram headers. A new Context ID is defined that indicates additional PDU set information is added within the Datagram payload. If the content server is not collocated with the proxy address then an additional API is provided for the content server to provide the PDU set information to the HTTP/3 proxy as the HTTP/3 proxy cannot read encrypted packets and determine PDU set information. The AS/HTTP/3 Proxy includes a downlink packet within a QUIC Datagram using a transport mode which is an extension of the mode defined in RFC 9298. The AS/QUIC includes the encrypted UDP packets within QUIC Datagram frames and provides unreliable transport and includes PDU set information of the encrypted packet within the QUIC Datagram. A new Datagram mode needs to be defined by using the Context ID (defined in RFC 9298) and include within the Datagram payload PDU set size, PDU set sequence number, PDU sequence number and PDU set importance. The UPF supports the same functionality in the uplink.

In step 1514, the QUIC datagram is sent to the UPF (HTTP/3 client) via N6. A new Transport Mode is defined for the QUIC Datagram that includes a Context ID denoting additional PDU set information included within the QUIC Datagram.

In step 1515, the UPF extracts the UDP packet from the QUIC datagram and determines to enable PDU set inspection based on configuration rules provided by the SMF. The UPF determines PDU set information using the PDU set information provided within the QUIC datagram, i.e. the UPF identifies PDU set information of the extracted PDU by

checking the payload of the QUIC Datagram. The UPF then routes the packets, via a QoS flow with PSDB requirements based on configuration rules from the SMF, towards the RAN within a GTP packet, the GTP packet including the identified PDU set information within a GTP-U header.

In an alternative embodiment, the encapsulation protocol between the UPF and AS is a GPRS Tunnelling Protocol that is used between the RAN and UPF over an N3 reference point. The UPF proxies GTP packets received from the RAN for a UE to GTP packets between the UPF and the Application Server and GTP packets received from the Application Server to the UPF to GTP packets between the UPF and RAN.

In the downlink, the AS for every UDP packet sent in the downlink will encapsulate the downlink UDP packet within a GTP packet and include PDU set information within its GTP-U header. The GTP-U header will be encoded as currently supported by the UPF as described in 3GPP TS 23.501. The advantage of this approach is that the UPF, by receiving the GTP packet from the AS, will simply have to de-encapsulate the UDP packet from the GTP session and encapsulate the UDP packet in a GTP packet in the GTP session between the UPF and RAN for a UE. The UPF includes within the GTP-U header of the UPF-RAN GTP session the same information included in the GTP-U header of the GTP packet of the AS-UPF GTP session.

The solution focuses on the UPF being the client and AS being the server of the tunnelling protocol. However, it is also possible for the tunnelling protocol client to be located on the AS and the server at the UPF. The AS will trigger a tunnelling protocol session to the UPF for an XR session. The AS will be aware of the UPF address based on configuration information. In such an embodiment, the procedure to encode PDU set information within the tunnelling protocol will be the same as described above.

Figure 16 illustrates an example of a NE 200 in accordance with aspects of the present disclosure. The NE 200 may include a processor 202, a memory 204, a controller 206, and a transceiver 208. The processor 202, the memory 204, the controller 206, or the transceiver 208, or various combinations thereof or various components thereof may be examples of means for performing various aspects of the present disclosure as

described herein. These components may be coupled (e.g., operatively, communicatively, functionally, electronically, electrically) via one or more interfaces.

The processor 202, the memory 204, the controller 206, or the transceiver 208, or various combinations or components thereof may be implemented in hardware (e.g., circuitry). The hardware may include a processor, a digital signal processor (DSP), an application-specific integrated circuit (ASIC), or other programmable logic device, or any combination thereof configured as or otherwise supporting a means for performing the functions described in the present disclosure.

The processor 202 may include an intelligent hardware device (e.g., a general-purpose processor, a DSP, a CPU, an ASIC, an FPGA, or any combination thereof). In some implementations, the processor 202 may be configured to operate the memory 204. In some other implementations, the memory 204 may be integrated into the processor 202. The processor 202 may be configured to execute computer-readable instructions stored in the memory 204 to cause the NE 200 to perform various functions of the present disclosure.

The memory 204 may include volatile or non-volatile memory. The memory 204 may store computer-readable, computer-executable code including instructions when executed by the processor 202 cause the NE 200 to perform various functions described herein. The code may be stored in a non-transitory computer-readable medium such the memory 204 or another type of memory. Computer-readable media includes both non-transitory computer storage media and communication media including any medium that facilitates transfer of a computer program from one place to another. A non-transitory storage medium may be any available medium that may be accessed by a general-purpose or special-purpose computer.

In some implementations, the processor 202 and the memory 204 coupled with the processor 202 may be configured to cause the NE 200 to perform one or more of the functions described herein (e.g., executing, by the processor 202, instructions stored in the memory 204). For example, the processor 202 may support wireless communication at the NE 200 in accordance with examples as disclosed herein.

The NE 200 may be configured to support a means for receiving a packet detection rule, PDR, including an indication to establish an encapsulation protocol with a server address when the network node detects that a packet is sent towards the server address, and configuration information to enable PDU set inspection and extract PDU set information from information contained within the encapsulation protocol when the network node receives a packet from the server address. The NE may be further configured to support a means for receiving a first uplink packet, inspecting the first uplink packet and identify the first uplink packet as matching the PDR, and sending a session request to establish a connection using the encapsulation protocol. The NE may be further configured to support a means for receiving a first downlink packet, inspecting the first downlink packet and identifying the first downlink packet as matching the PDR. The method may further include using the configuration information to perform PDU set inspection of the first downlink packet and to extract PDU set information and a second downlink packet encapsulated within the first downlink packet, wherein the PDU set information is associated to the second downlink packet.

The NE may be configured to extract a second downlink packet from the first downlink packet where the second downlink packet and protocol description of the first downlink packet matching the PDR.

The NE may be further configured to support a means for constructing a third downlink packet including the second downlink packet as a payload; and sending the third downlink packet to a GPRS Tunneling Protocol, GTP, session, wherein the third downlink packet includes within a GTP-U header the extracted PDU set information or PDU set information derived from the extracted PDU set information.

The PDR may indicate that PDU set inspection is to be enabled. The PDR may indicate that PDU set inspection is to be enabled based on the encapsulation protocol between the UPF and the server address. The configuration information may include a protocol description within a QoS Enforcement Rule within the PDR indicating a method of retrieving PDU set information for the second downlink packet from the first downlink packet. The protocol description within configuration information may indicate HTTP/3 and/or CONNECT-UDP. The protocol description within configuration information may indicate GTP. The connection may be a QUIC connection such that the first downlink

packet is a QUIC datagram containing a Context ID denoting PDU set information within the QUIC Datagram payload. The encapsulation protocol may be a CONNECT-UDP protocol. The connection may be a GTP session using GTP protocol, the first downlink packet is a GTP packet, and the PDU set information is included within a GTP-U header of the first downlink packet. The encapsulation protocol may be a GTP protocol. The network node may be a User Plane Function node.

Alternatively, the NE 200 may be configured to support a means for sending, to a second network node, a packet detection rule, PDR, including an indication to establish an encapsulation protocol with a server address when the second network node detects that a packet is sent towards the server address, and configuration information to enable PDU set inspection and extract PDU set information from information contained within the encapsulation protocol when the network node receives a packet from the server address. The network node may be a Session Management Function SMF.

The controller 206 may manage input and output signals for the NE 200. The controller 206 may also manage peripherals not integrated into the NE 200. In some implementations, the controller 206 may utilize an operating system such as iOS®, ANDROID®, WINDOWS®, or other operating systems. In some implementations, the controller 206 may be implemented as part of the processor 202.

In some implementations, the NE 200 may include at least one transceiver 208. In some other implementations, the NE 200 may have more than one transceiver 208. The transceiver 208 may represent a wireless transceiver. The transceiver 208 may include one or more receiver chains 210, one or more transmitter chains 212, or a combination thereof.

A receiver chain 210 may be configured to receive signals (e.g., control information, data, packets) over a wireless medium. For example, the receiver chain 210 may include one or more antennas for receive the signal over the air or wireless medium. The receiver chain 210 may include at least one amplifier (e.g., a low-noise amplifier (LNA)) configured to amplify the received signal. The receiver chain 210 may include at least one demodulator configured to demodulate the receive signal and obtain the transmitted data by reversing the modulation technique applied during transmission of the signal.

The receiver chain 210 may include at least one decoder for decoding and/or processing the demodulated signal to receive the transmitted data.

A transmitter chain 212 may be configured to generate and transmit signals (e.g., control information, data, packets). The transmitter chain 212 may include at least one modulator for modulating data onto a carrier signal, preparing the signal for transmission over a wireless medium. The at least one modulator may be configured to support one or more techniques such as amplitude modulation (AM), frequency modulation (FM), or digital modulation schemes like phase-shift keying (PSK) or quadrature amplitude modulation (QAM). The transmitter chain 212 may also include at least one power amplifier configured to amplify the modulated signal to an appropriate power level suitable for transmission over the wireless medium. The transmitter chain 212 may also include one or more antennas for transmitting the amplified signal into the air or wireless medium.

Figure 17 illustrates a flowchart of a method in accordance with aspects of the present disclosure. The operations of the method may be implemented by a NE as described herein. In some implementations, the NE may execute a set of instructions to control the functional elements of the NE to perform the described functions.

At step 1702, the method may include receiving a packet detection rule, PDR, including an indication to establish an encapsulation protocol with a server address when the network node detects that a packet is sent towards the server address, and configuration information to enable PDU set inspection and extract PDU set information from information contained within the encapsulation protocol when the network node receives a packet from the server address. The operations of step 1702 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of 1702 may be performed by a NE as described with reference to Figure 16.

At step 1704, the method may include receiving a first uplink packet, inspecting the first uplink packet and identifying the first uplink packet as matching the PDR, and sending a session request to establish a connection using the encapsulation protocol. The operations of step 1704 may be performed in accordance with examples as described

herein. In some implementations, aspects of the operations of step 1704 may be performed by a NE as described with reference to Figure 16.

At step 1706, the method may include receiving a first downlink packet, inspecting the first downlink packet and identifying the first downlink packet as matching the PDR. Alternatively, extracting a second downlink packet from the first downlink packet where the second downlink packet and protocol description of the first downlink packet matching the PDR. The operations of step 1706 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of step 1706 may be performed by a NE as described with reference to Figure 16.

At step 1708, the method may include using the configuration information to perform PDU set inspection of the first downlink packet and to extract PDU set information and a second downlink packet encapsulated within the first downlink packet, wherein the PDU set information is associated to the second downlink packet. The operations of step 1708 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of step 1708 may be performed by a NE as described with reference to Figure 16.

At step 1710, the method may include constructing a third downlink packet including the second downlink packet as a payload. The operations of step 1710 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of step 1710 may be performed by a NE as described with reference to Figure 16.

At step 1712, the method may include sending the third downlink packet to a GPRS Tunneling Protocol, GTP, session, wherein the third downlink packet includes within a GTP-U header the extracted PDU set information or PDU set information derived from the extracted PDU set information. The operations of step 1712 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of step 1712 may be performed by a NE as described with reference to Figure 16.

Figure 18 illustrates a flowchart of a further method in accordance with aspects of the present disclosure. The operations of the method may be implemented by a NE as described herein. In some implementations, the NE may execute a set of instructions to control the functional elements of the NE to perform the described functions.

At step 1802, the method may include sending, to a second network node, a packet detection rule, PDR, including an indication to establish an encapsulation protocol with a server address when the second network node detects that a packet is sent towards the server address, and configuration information to enable PDU set inspection and extract PDU set information from information contained within the encapsulation protocol when the network node receives a packet from the server address. The operations of step 1802 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of step 1802 may be performed by a NE as described with reference to Figure 16.

It should be noted that the methods described herein describe possible implementations, and that the operations and the steps may be rearranged or otherwise modified and that other implementations are possible.

Although some of the embodiments and discussions hereafter use extended reality (XR) as a reference use case, one skilled in the art should realize that the proposed solutions are generally applicable and may be embodied by different types of transport and network protocol stacks (e.g., Quick UDP Internet Connections (QUIC), WebRTC, WebTransport or alike). Furthermore, XR is referred to hereafter as an umbrella term for different types of realities:

- Virtual Reality (VR) is a rendered version of a delivered visual and audio scene. The rendering is in this case designed to mimic the visual and audio sensory stimuli of the real world as naturally as possible to an observer or user as they move within the limits defined by the application. Virtual reality usually, but not necessarily, requires a user to wear a head mounted display (HMD), to completely replace the user's field of view with a simulated visual component, and to wear headphones, to provide the user with the accompanying audio. Some form of head and motion tracking of the user in VR is usually also necessary to allow the simulated visual and audio components to be updated to

ensure that, from the user's perspective, items and sound sources remain consistent with the user's movements. In some implementations additional means to interact with the virtual reality simulation may be provided but are not strictly necessary.

- Augmented reality (AR) is when a user is provided with additional information or artificially generated items, or content overlaid upon their current environment. Such additional information or content will usually be visual and/or audible and their observation of their current environment may be direct, with no intermediate sensing, processing, and rendering, or indirect, where their perception of their environment is relayed via sensors and may be enhanced or processed.
- Mixed reality (MR) is an advanced form of AR where some virtual elements are inserted into the physical scene with the intent to provide the illusion that these elements are part of the real scene.

XR refers to all real-and-virtual combined environments and human-machine interactions generated by computer technology and wearables. It includes representative forms such as AR, MR and VR and the areas interpolated among them. The levels of virtuality range from partially sensory inputs to fully immersive VR. A key aspect of XR is the extension of human experiences especially relating to the senses of existence (represented by VR) and the acquisition of cognition (represented by AR).

The description herein is provided to enable a person having ordinary skill in the art to make or use the disclosure. Various modifications to the disclosure will be apparent to a person having ordinary skill in the art, and the generic principles defined herein may be applied to other variations without departing from the scope of the disclosure. Thus, the disclosure is not limited to the examples and designs described herein but is to be accorded the broadest scope consistent with the principles and novel features disclosed herein.