Tilli, Juha Matti; Kantola, Raimo

Data plane protocols and fragmentation for 5G

Published in:
2017 IEEE Conference on Standards for Communications and Networking, CSCN 2017

DOI:
10.1109/CSCN.2017.8088623

Published: 27/10/2017

Document Version
Peer reviewed version

Please cite the original version:
Tilli, J. M., & Kantola, R. (2017). Data plane protocols and fragmentation for 5G. In 2017 IEEE Conference on Standards for Communications and Networking, CSCN 2017 (pp. 207-213). [8088623] Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/CSCN.2017.8088623

Data Plane Protocols and Fragmentation for 5G

Juha-Matti Tilli
Bell Labs
Nokia
Espoo, Finland
juha-matti.tilli@nokia-bell-labs.com

Raimo Kantola
Department of Communications and Networking
Aalto University
Espoo, Finland
raimo.kantola@aalto.fi

Abstract— **Mobile networks up to this point have used general packet radio service (GPRS) tunneling protocol (GTP) in the core network. The fifth generation (5G) will be a major revolution in mobile networks and there have been discussions onto adopting a different protocol, with the proposals being generic routing encapsulation (GRE), virtual extensible local area network (VXLAN) and generic network virtualization encapsulation (Geneve). In this paper, these protocols are compared. GTP has a 32-bit tunnel endpoint identifier (TEID) and runs on top of user datagram protocol (UDP), thus allowing many applications in the same IP address in different ports. However, GTP unnecessarily stores packet length many times in the Internet protocol (IP), UDP and GTP levels, and the fragmentation strategy defined in third generation partnership project (3GPP) specifications is suboptimal. GRE has well-defined fragmentation strategy but runs directly on top of IP, so multiplexing based on ports is not possible. VXLAN and Geneve have only 24 bits for network identifier and encapsulate media access control (MAC) frames, which is not really what is needed here, and the fragmentation strategy of VXLAN is suboptimal albeit in a different way than in GTP. In this paper, we propose a common fragmentation strategy that may be applicable to all protocols.**

Keywords—Generic Routing Encapsulation; General Packet Radio Service Tunneling Protocol; Virtual Extensible Local Area Network; Generic Network Virtualization Encapsulation

I. Introduction

The mobile networks in the future will use fifth generation (5G) technology. However, existing second generation (2G), third generation (3G) and fourth generation (4G) networks are still in active use. The mobile packet data revolution started from 2G general packet radio service (GPRS) [1] which had data rate order of magnitude of 100 kbit/s. Enhanced data rates for global system for mobile telecommunications (GSM) evolution (EDGE) [2] improved data rates manyfold while still continuing to use the 200 kHz carrier bandwidth of GSM and therefore also GPRS. GPRS introduced the GPRS tunneling protocol (GTP) which has been used in all existing packet data generations. The 3G universal mobile telecommunications system (UMTS) [3] was a radio network update that still used the GPRS core network. Data rates were poor for the 5 MHz carrier bandwidth, but fortunately high speed packet access (HSPA) [4] improved the rates. The 4G long term evolution (LTE) [5] evolved the whole system including the core network, but the tunneling protocol on which the system is based continues to be GTP.

There have been proposals to adopt a different user plane tunneling protocol for 5G. The proposals have included generic routing encapsulation (GRE), virtual extensible local area network (VXLAN) and generic network virtualization encapsulation (Geneve). However, continuing to use GTP is possible too. To understand which protocol is the best, information about their differences is needed. There are publicly available specifications for all protocols, but a clear summary of their differences is lacking and thus currently the information needs to be obtained by carefully reading the specifications and observing the differences. Furthermore, the fragmentation strategies of some protocols are suboptimal and require improvement. Fragmentation is something that perhaps could be handled in a protocol independent manner. A common approach to fragmentation would ideally make it possible to use different tunneling protocols in different network segments of the end-to-end path and mangle the packets with e.g. OpenFlow switches on the segment boundaries.

The intention of this paper is to provide a comparison of the various user plane protocols. Furthermore, the intention is to provide a common fragmentation strategy that is applicable to all protocols. Note that control plane protocols are beyond the scope of this article. This paper is organized in the following way. Firstly, in Section 2, the tunneling protocols GTP, GRE, VXLAN and Geneve are introduced, and in Section 3 discussed in detail and compared. Section 4 contains header byte overhead comparison of the protocols. Section 5 introduces a radical proposal, direct use of IP over Ethernet without tunneling. Section 6 discusses multicore aspects of the protocols. Section 7 discusses fragmentation issues and proposes a common fragmentation strategy that is applicable to all protocols. Finally, Section 8 concludes this paper.

The summary of our proposals is the following:

- The fragmentation strategy of GRE should be adopted to replace GTP's poor strategy
- GRE should be considered as GTP replacement

Note that third generation partnership project (3GPP) has already made a decision to continue using GTP in 5G, so the second proposal would require a major turn of direction. The rationale for continuing to use GTP is that it is under 3GPP control, so extensions are easy to implement.

II. Tunneling protocols proposed for 5G

There are two options how to build the underlying backhaul/core network. First option is that the underlying backhaul/core network is IP based and some routers are used to connect base stations to the edge of the Internet. The second option is that the backhaul network would be a switched packet network controlled by an SDN connection controller e.g. using OpenFlow (OF) between the controller and the OF-switches. Depending on which packet formats would be used in the network, it could also contain either Ethernet or MPLS switches provided that these could also be controlled by the same centralized controller. OF-switches could be used at some switched path endpoints but simple switching operations could be done using existing hardware switches (802.1 or MPLS). It is also possible to terminate switched paths using such switches but the connection controller must be able to control the operations – i.e. populate the connection state in the switches as needed.

In a mobile network, there is a need for a fine-grained control over the service that is provided to the user. This can be done by introducing a switched connection called a *tunnel* over a routed underlying network. In case of using an OF-controlled switched packet network, the task would be to provide a similar fine grained control over the service using a combination of the SDN controller and some signaling similar to the GTP control protocol.

We assume that the choice of the approach for the network may depend on existing investments into the operator's network among other things. This paper will mainly focus on the case of a routed underlying network but will also discuss the case of switched packet network briefly.

There are several proposals for the user plane tunneling protocols to be used in 5G. One proposal is to continue the use of GTP, perhaps improving it in some manner by introducing a new protocol version. This paper discusses only the existing versions of GTP because the possible proposal to improve GTP can in practice mean anything, even making GTP very similar to GRE or VXLAN. Thus, it is not meaningful to discuss a hypothetical improved version of GTP without knowing what the improvements are. GTP tunnels layer 3 (L3) packets, runs on top of UDP and supports a 32-bit tunnel endpoint identifier (TEID) which is likely large enough even for 5G networks given the potential Internet of things (IoT) and massive machine to machine (M2M) use cases, because TEID does not need to be globally unique.

Another proposal is to use GRE. A valid observation here is that GTP is in practice used only in mobile packet data networks, whereas the rest of the industry almost unanimously uses GRE to tunnel L3 packets. Why should the mobile packet data networks continue to do things differently? GRE, however, does not have anything comparable to TEID as mandatory. Fortunately, GRE has a 32-bit optional key field which can be used in a similar manner than TEID. GRE differs from GTP in that it runs directly on top of IP instead of running on top of UDP. This has benefits as well as drawbacks.

Yet another proposal is to use VXLAN instead of GTP or GRE. VXLAN is quite different from GTP and GRE in two ways. Firstly, VXLAN tunnels layer 2 (L2) frames instead of L3 packets. Secondly, VXLAN has only 24 bits available for the key field which is now called VXLAN network identifier (VNI). Like GTP, VXLAN runs on top of UDP.

Also, Geneve has been proposed as an alternative to the other protocols. Geneve is similar to VXLAN in tunneling L2 frames, running on top of UDP and also having only 24 bits for the network identifier field, but is more versatile.

VXLAN and Geneve are somewhat different than GTP and GRE because they tunnel L2 frames instead of tunneling L3 packets, but as they are data plane tunneling protocols as well, they have been included in this paper for completeness.

In the next sections, we discuss GTP, GRE, VXLAN and Geneve in detail. This is followed by a quick comparison of the protocols and our proposed approaches to fragmentation issues in all protocols, after which the paper is concluded.

III. Tunneling protocols

A. *General packet radio service (GPRS) tuneling protocol (GTP)*

GTP as its name suggests was originally designed for packet tunneling within GPRS. There are actually several variants: GTP-C for control plane, GTP-U for user plane and GTP' for transferring charging data. In this paper, we only discuss GTP-U because the subject of this paper is user plane tunneling. There are also several versions of GTP including GTPv1 and GTPv2, but user plane tunneling is not supported in GTPv2. Thus, this paper discusses GTPv1-U only. GTPv1 is defined in 3GPP technical specification (TS) 29.060 [6].

The message format of GTP is shown in Fig. 1. It can be observed that there are 32 optional bits which are present if any of the three fields in the optional bits is present. There is also a message type, a total length field and TEID.

GTP is in practice used almost always on top of UDP, although GTP supports in theory operation over transmission control protocol (TCP) too.

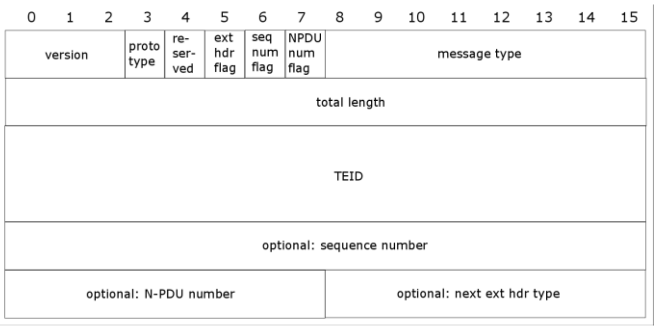

Fig. 1. Header format of GTP.

One weakness of GTP is that as it operates on top of UDP, there is an unnecessary checksum at the UDP level. The reason this is unnecessary is that the inner IP packet likely contains TCP or UDP data, which is already checksummed, so there is no need to checksum the outer packet. Fortunately, UDP over IPv4 allows setting the checksum field always to zero instead of calculating a proper checksum. In IPv6, the checksum field may not be zero. However, RFC 6935 [7] relaxes the checksum requirement for tunneled packets in IPv6 packets by allowing the application to tell to the UDP stack that the use case is tunneling and thus the checksum may be omitted.

Another weakness of GTP is that the total length field is needlessly stored to the GTP header. This should be unnecessary, as there is a belt and suspenders approach for the total length already in place in the IP and UDP levels: both IP and UDP store the total length in their own headers. Thus, in the case of GTP, the total length is stored three times. Clearly, this is unnecessary. A protocol operating on top of UDP could avoid storing the total length in its own header, but cannot avoid the repeated length storing in the IP and UDP levels. Thus, if it is desired that the total length is not needlessly stored over and over, the only solution is to operate the protocol directly on top of IP bypassing the UDP processing completely.

Yet another weakness of GTP is that it provides no mechanism for detecting the protocol of the inner layer. GTP is designed to tunnel L3 packets, so in practice the inner layer is IPv4 or IPv6. Fortunately, IPv4 and IPv6 can be distinguished by inspecting the version field. However, for other protocols than IP, GTP is not a good choice, as the packet inspection may not provide a clear identification for the protocol.

However, GTP has strengths too. As it operates on top of UDP, the inherent port based multiplexing provided by UDP is available. Ordinarily, GTPv1-U operates on top of UDP port 2152, but in theory it is possible to run multiple network elements on a single computer as separate user space processes, utilizing the same IP address but different UDP ports. The 32 bits of TEID is likely enough for the foreseeable future as the TEID does not need to be globally unique like IP addresses, and it is a strength that the TEID is always present and is not an afterthought in the form of an optional extension header. Furthermore, operation on top of UDP means that it operates well with firewalls, although GTP is typically used in operator's internal networks that are not directly connected to Internet, thus not needing firewalls. However, roaming may have additional requirements related to firewalls.

GTP uses IP fragmentation if the packet is too large to fit within the maximum transmission unit (MTU). The way that GTP standards describe the use of fragmentation is not compliant with many RFCs. It is recommended that the outer IP packet should be fragmented regardless of whether the inner IP packet has the don't fragment (DF) bit set. This directly contradicts RFC 2003 [8], which says that if the DF bit is set in the inner IP header, it MUST be set in the outer IP header too and if not set in the inner header, it MAY be set in the outer IP header. Furthermore, given that GTP produces non-atomic packets that are fragmentable, this in practice violates RFC 6864 [9] which says that "sources of non-atomic IPv4 datagrams MUST rate-limit their output to comply with the ID uniqueness requirements". The GTP specification has no requirement for such rate limiting, and such rate limiting would make GTP too slow.

B. Generic routing encapsulation (GRE)

GRE is the industry standard protocol for tunneling layer 3 frames that is specified in RFC 2784 [10]. For this reason, the use of GRE has been proposed for mobile packet data networks too. GRE operates directly on top of IP having the protocol number 47. Thus, the UDP layer and the trivial services it provides are omitted. The message format of GRE is illustrated in Fig. 2.

GRE overcomes most of the weakness in GTP. Operating directly on top of IP, the repeated length field storage on IP and UDP headers is omitted and unlike GTP header, GRE header does not contain an additional length field. Furthermore, the UDP header checksum is omitted. However, GRE has support for checksum within the GRE level. Fortunately, this checksum is optional so it does not need to be used. GRE also always contains a 16-bit protocol type that uses the same namespace as the Ethernet protocol type field. So, it is possible to carry routed protocols other than IPv4 and IPv6 and still provide capability for detecting the protocol.

The GRE header much like the GTP header has a variable size, and there is no header length field, so the header length needs to be calculated. In the case of GRE, there are three bits and the count of set bits tells how many 32-bit additional fields there are in the header, so length calculation should be relatively easy on a computer that has a population count instruction. GTP has three optional field bits and if any of these bits is set, the whole 32 bits of optional data is included. However, GTP has support for extension headers, so if these are present, calculating the total header length requires iteration. So, GRE header length calculation is easier than GTP. On the other hand, this means that GRE is not as easily extensible as GTP is.

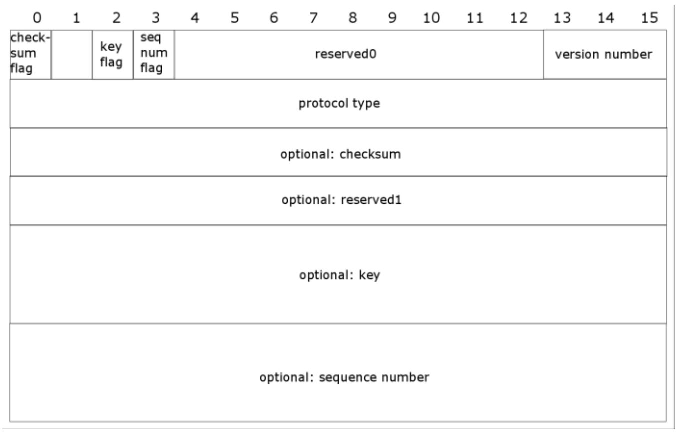

Fig. 2. The header format of GRE.

However, GRE has drawbacks too. As it operates directly on top of IP, port multiplexing is not available. Fortunately, these days it is easy to give multiple IP addresses to a single computer, thus making IP address based multiplexing available. Operating on top of IP instead of UDP, the compatibility of GRE with some firewalls may be an issue in some use cases, but fortunately, GRE being an industry standard somewhat helps in this case. Furthermore, the key field in GRE is an afterthought [11] and it may be a problem that the key is not in all use cases within the first 8 bytes of the IP packet payload, complicating application detection in the case of Internet control message protocol (ICMP) error messages that in the case of IP version 4 (IPv4) contain only 8 bytes of the payload. Fortunately, the key contains 32 bits like it does in the case of GTP.

A clear strength of GRE is that there is a widely deployed solution to the fragmentation problem that is well-defined in an RFC [12]. In the default configuration, the outer packet is never fragmented. The inner IP packet, however, in the case of IPv4 may be fragmented if the DF bit is not set, thus avoiding the need for GRE tunnel endpoints to perform fragment reassembly. If the DF bit is set or the packet is IPv6 and the packet does not fit within the tunnel MTU, an ICMP error message is sent to inform the packet sender that the packet is too large. The DF bit in the outer IP header is always set to 1, thus making the packet atomic and thus not needing to rate-limit according to RFC 6864 [9]. We propose in Section 7 that this fragmentation solution could be applied to VXLAN and GTP as well.

C. Virtual extensible local area network (VXLAN)

VXLAN solves a somewhat different problem than GTP and GRE. VXLAN provides virtualized L2 networks on top of a routed L3 network infrastructure. Thus, VXLAN provides bridged, not routed connectivity. This may require major system re-designs if VXLAN is taken into use in 5G, because the existing systems carry L3 packets and VXLAN requires L2 frames. In theory, it would be possible to just invent fake media access control (MAC) addresses and use VXLAN like a L3 on top of L3 tunneling solution, but this would be contrary to the purpose of VXLAN. Also it is not clear why one would want to use such hacks if the clean solution of simply using GRE is available. The message format of VXLAN is illustrated in Fig. 3. Note that the encapsulated L2 frame does not contain a frame check sequence (FCS).

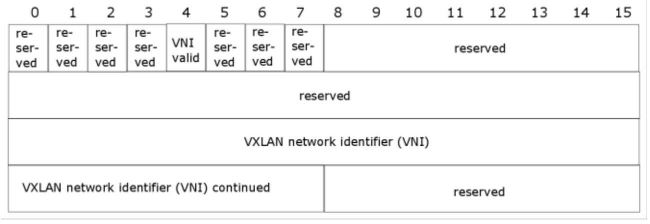

Fig. 3. The message format of VXLAN.

VXLAN operates on top of UDP like GTP. Fortunately, VXLAN has not made the mistake of providing yet another length field, so in the case of VXLAN, only the length fields in the IP header and in the UDP header are used. VXLAN also does not have a separate checksum field, but the UDP checksum may be used. However, it is proposed that the UDP checksum SHOULD be transmitted as zero and that if a non-zero checksum is received, it MAY be verified. VXLAN also has a well-defined port like GTP, but the specification says that the port SHOULD be configurable.

VXLAN has only 24 bits for network identifier. This is less than the 32 bits that are available in GTP and GRE, and may be a problem if VXLAN would be used in 5G. For example, one user plane gateway (UGW) may serve dozens of base stations each having a million connected sensors that transmit data only rarely. The 24 bit network identifier does not have a large enough namespace for this application.

VXLAN has a particularly poor fragmentation strategy. RFC 7348 [13] says that tunnel endpoints must not fragment packets. However, it says that intermediate routers may fragment packets. Presumably, this means that in the case of IPv4, the don't fragment (DF) bit must not be set to allow fragmentation. This means that VXLAN packets are non-atomic, thus directly violating the rate limiting requirement of RFC 6864 [9]. If the DF bit is not set in the outer packet but is set in the inner packet, this also violates RFC 2003 [8]. In the case of IPv6, the claim that intermediate routers may fragment packets seems absurd, and forbidding the tunnel endpoint from fragmenting packets prevents the use of IPv6 fragmentation completely. Nevertheless, if an intermediate router fragments packets, it may be fatal, as RFC 7348 further says that the destination tunnel endpoint may discard such fragments. This may result in hard to diagnose connectivity problems. However, the recommendation of VXLAN is to ensure that the network that carries tunneled packets has a big enough MTU, and if the requirement is followed, fragmentation is not needed at all and the solution works perfectly.

D. Geneve

Geneve [14] is in many ways similar to VXLAN. It runs on top of UDP and tunnels L2 frames. However, Geneve has an additional protocol type field, so it can tunnel other L2 protocols in addition to Ethernet. Also, Geneve has a variable-length options field, so the protocol is more versatile than VXLAN.

Geneve's recommended fragmentation strategy is somewhat better than VXLAN's. The Geneve Internet draft recommends using the DF bit and path MTU discovery. The message format is shown in Fig. 4.

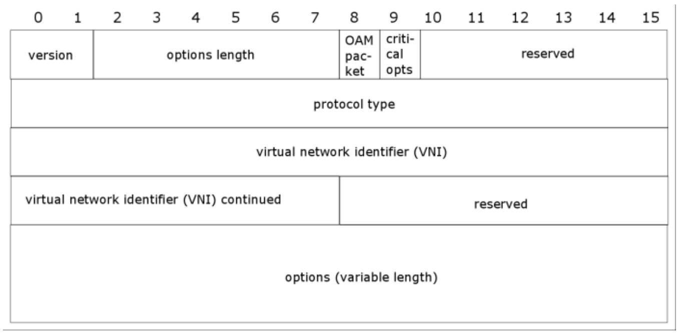
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
version
options length
OAM pac-ket
criti-cal opts
reserved
protocol type
virtual network identifier (VNI)
virtual network identifier (VNI) continued
reserved
options (variable length)

Fig. 4. The message format of Geneve.

E. A quick comparison

A quick comparison can be made between GTP, GRE, VXLAN and Geneve, which is shown in Table 1.

Aspect	**GTP**	**GRE**	**VXLAN**	**Geneve**
Underlying protocol	UDP	IP	UDP	UDP
Encapsulates	L3 packets	L3 packets	L2 frames	L2 frames
Protocol type field	no	yes	yes	yes
Extension headers	yes	no	no	yes
Specified in	3GPP TS	RFC	RFC	Internet draft
Key space	32 bits	32 bits (optional)	24 bits	24 bits
Header length	variable	variable	constant	variable
Length field count	3	1	2	2
Checksum	UDP (can be 0)	optional	UDP (can be 0)	UDP (can be 0)
Overhead	medium	small	large	large
Fragmentation	poor solution	good solution	poor solution	good solution

Table 1. The comparison between GTP, GRE and VXLAN.

IV. Total system overhead

Let us consider the total system overhead of the various protocols for small IPv6 voice over IP (VoIP) packets and for large IPv4 TCP packets. Of these, IPv4 TCP is probably today the most common but with voice moving from circuit-switched networks to packet-switched networks that will eventually have to adopt IPv6, the IPv6 VoIP case may become common as well. All of the tunneling protocols run in an IPv4 core network implemented using Ethernet with slightly raised MTU to prevent fragmentation.

The IPv6 VoIP packets have an IPv6 header of 40 bytes and a UDP header of 8 bytes and an RTP header of 12 bytes. A sample period of 20 ms is common. For adaptive multi rate (AMR) codec at 12.2 kbps, this means on average 30.5 bytes of payload. Let's make it 31 bytes to make it an integer.

The IPv4 TCP packets have an IPv4 header of 20 bytes and a TCP header of 20 bytes. There is 1460 bytes of payload, making the packet full-sized (1500 bytes).

An Ethernet packet has a preamble of 7 bytes, start frame delimiter of 1 byte, Ethernet header of 14 bytes, payload (which in our case may be slightly bigger than the usual maximum 1500 bytes), frame check sequence of 4 bytes and interpacket gap of 12 bytes.

A GTP packet without extension headers transmitted in an Ethernet network has 20 bytes of IPv4 header, 8 bytes of UDP header and 8 bytes of GTP header.

A GRE packet with key but without sequence number or checksum has 20 bytes of IPv4 header and 8 bytes of GRE header.

A VXLAN packet has 20 bytes of IPv4 header, 8 bytes of UDP header, 8 bytes of VXLAN header and also needs to have a separate 14 byte inner Ethernet header. The same is true for Geneve as well, assuming there are no options in the packet.

Use case	Overhead	Payload	Payload % of data
GTP IPv6 VoIP	134 bytes	31 bytes	18.8%
GTP IPv4 TCP	114 bytes	1460 bytes	92.8%
GRE IPv6 VoIP	126 bytes	31 bytes	19.7%
GRE IPv4 TCP	106 bytes	1460 bytes	93.2%
VXLAN IPv6 VoIP	148 bytes	31 bytes	17.3%
VXLAN IPv4 TCP	128 bytes	1460 bytes	91.9%
Geneve IPv6 VoIP	148 bytes	31 bytes	17.3%
Geneve IPv4 TCP	128 bytes	1460 bytes	91.9%

Table 2: The overheads of various use cases.

The overheads are shown in Table 2.

In summary, GRE has the smallest header byte overhead and VXLAN and Geneve the largest. Considering that Markets and Markets has found mobile backhaul market to increase from 17.85 billion dollars to 33.15 billion dollars annually[1], TCP difference for GRE and GTP is 0.4%, meaning 0.4% of about 20 billion dollars or 80 million dollars worldwide could be saved annually by using GRE instead of GTP.

Note that this very rough savings estimate assumes that bitrate is the limitation and not the packet rate. In software based packet processing, packet rate may be more limiting. We tested processing merely the headers of 1500-byte GTP packets using our tunneling software modified to process only headers. Processing 100 million packets took 4.132 seconds, a performance of 24.2 million packets per second using a single thread. Then we tested the full packet processing infrastructure with memory allocation but without accessing real network interfaces. Performance was now 5.0 million packets per second. With accessing network interfaces using netmap, performance went down to 2.0 million packets per second. This order of magnitude difference (2.0 million packets per second vs 24.2 million packets per second) means that the main overheads are elsewhere than in the tunneling protocols. We did not test GRE, but we expect that GRE would be slightly faster due to not having UDP headers.

V. Eliminating overheads with IP over Ethernet

As a summary, all tunneling protocols have overheads requiring larger MTU than usual. Also, the extra header fields need additional packet processing overheads. All of these

[1] http://www.marketsandmarkets.com/PressReleases/mobile-wireless-backhaul.asp

overheads could be eliminated by directly transmitting IP frames over Ethernet, eliminating all tunneling [15].

In this proposal, the backhaul/core network would be one big SDN controlled Ethernet network or a set of such networks. Any device connected to the core network can reach any other device connected to the backhaul/core network at the Ethernet level using MAC addresses, and the different tags in the 802.1ad/ah header. A tunnel as supported by GTP in current networks would be replaced by a switched tunnel controlled by SDN. Also, base stations and user plane gateways would have at least one dedicated MAC address used for user plane packets, or perhaps two: one for uplink and one for downlink. Packets to these MAC addresses are always treated as user plane packets. Control plane signaling, if running on top of IP, must use a different MAC address. For user plane, address resolution protocol (ARP) / neighbor discovery protocol (NDP) would not be used, as base stations and user plane gateways can reach each other through using MAC addresses directly.

The benefit of this proposal is that the 1500 byte packet size is supported without needing to raise the MTU, and the overheads (packet processing and header byte overheads) are the smallest possible for any technology. The drawback is that an existing IP core network cannot be used, because the entire core network must be SDN controlled.

VI. Multicore aspects

If there is a small number of highly loaded base stations, it makes sense to make sure that data from one base station is evenly distributed to all processor cores in a user plane gateway. Packet hashing is generally based on IP addresses and maybe ports. In the case of VXLAN, GTP and Geneve, selecting the source port to be different for different tunnels or flows is enough.

However, GRE has no ports, and thus packet hashing can only use IP addresses. It is possible to assign multiple IP addresses to one tunnel endpoint, and then select the IP address based on the tunnel or the flow by using e.g. a hash function of the tunnel/flow identifier.

VII. A common fragmentation strategy for all protocols

One might wonder why fragmentation is necessary at all, if the operator can set the MTU in its network to a high enough value, eliminating the need for fragmentation. The answer to this question is roaming: there will be a tunnel from the visited network to the home network, and this tunnel most likely runs over an infrastructure for which the MTU is 1500 bytes. The need for common fragmentation strategy can be justified by noting that different operators may have different protocols in their core networks, leading to various problems in hybrid paths (i.e. paths having many protocols) from home core network to visited core network. Also, for highest possible performance, the core network should be implemented with application specific integrated circuits (ASICs), and the fragmentation strategy needs to be hardcoded to such an ASIC, which means that a common fragmentation strategy is useful.

The fragmentation approach employed should be compatible with RFC 2003 and RFC 6864. In particular, RFC 2003 requires tunnel endpoints to maintain soft state of the tunnel including MTU of the tunnel. RFC 2003 also permits the sender to set the outer IP packet DF field always to one. RFC 6864 requires senders of non-atomic datagrams to rate-limit their output. This means that in practice the best thing to do is to always set the outer IP packet DF field to one. Thus, the outer packet may not be fragmented. This eliminates one from the two potential fragmentation solutions: fragment the outer IP packet or fragment the inner IP packet. This also eliminates the need for tunnel endpoints to perform IP fragment reassembly.

The only solution left thus is to fragment the inner IP packet. This may not be always possible, because the inner IP packet may have the DF bit set. Furthermore, in the case of IPv6, inner IP packet fragmentation is not allowed except by the original sender. Typically, most TCP packets have the DF bit set due to the operation of path MTU discovery, and if the requirements of RFC 6864 are followed, of the rate limiting option and atomic packet option it is better to select the atomic packet option.

The solution we propose is the following, the same as the solution in RFC 7588:

- Maintain a tunnel MTU value following the recommendations of RFC 2003
- Set the DF bit always in the IP header of the enclosing packet, following the permission of RFC 2003
- Fragment the inner IP packet if it is fragmentable and fragmentation is needed
- Send the original sender an ICMP packet too big message if fragmentation would be needed but is not permitted (e.g. due to the use of IPv6 or due to the DF bit being set in an IPv4 header)
- When receiving a tunneling packet fragment, silently discard the fragment without attempting to reassemble it

This solution is compatible with RFC 6864 regardless of the packet rate because it produces only atomic packets. The solution is also compatible with RFC 2003, the specification for IP tunneling within IP.

In theory, the solution could be applicable to VXLAN and Geneve as well. However, as they encapsulate L2 frames and not L3 packets and because it is not possible to fragment L2 frames, the solution would require breaking the L2 encapsulation, fragmenting the L3 frame and building again the L2 encapsulation for each fragment. Therefore, using this solution in VXLAN and Geneve could be considered a hack. Furthermore, since VXLAN and Geneve are bridged tunnels instead of being routed tunnels, it is unclear what source IP address to use when sending the ICMP packet too big message. Nevertheless, the solution even in the case of VXLAN may have in some applications better characteristics than the poorly designed existing fragmentation solution for VXLAN.

There is nothing that prevents the use of this solution to GTP. It is better than the existing solution in GTP which is

RFC-incompliant due to generating non-atomic packets at a great rate and always setting the DF bit to zero. It is proposed that if GTP is adopted for 5G as well, the fragmentation solution is updated to follow our recommendations.

VIII. CONCLUSIONS

Mobile networks of the future will use 5G technology. Existing generations have used GTP as the tunneling protocol, but there have been proposals to change the protocol in 5G. However, more information is needed about the differences of the protocols, and therefore, this paper aims to fill that gap.

One proposal is to continue using GTP, perhaps a new version. Another proposal is to use GRE which is used almost unanimously in the industry to tunnel L3 packets. Yet another proposal is to use VXLAN or Geneve.

GTP runs on top of UDP and has a variable-length header containing a 32-bit TEID. Weaknesses include unnecessary UDP checksum, repeated length information and lack of protocol identifier. Advantages are that it uses the UDP port multiplexing, firewall friendliness of UDP and that it has 32-bit TEID. The fragmentation solution has room for improvement.

GRE is the industry standard for tunneling L3 frames, operating directly on top of IP. Advantages of GRE are that the checksum field is optional and that repeated length information is omitted. GRE also has 16 bits for protocol identification. GRE, like GTP, also has a variable length header. Weaknesses of GRE are that it cannot use UDP's port multiplexing service, firewall compatibility and the fact that the 32-bit key field is an afterthought. GRE has a well-defined fragmentation solution.

VXLAN and Geneve solve somewhat different problem than GTP and GRE, providing virtualized L2 networks on top of a routed L3 network. VXLAN and Geneve do not have a length field in the tunneling layer, but it is stored repeatedly in the IP and UDP layers. Only 24 bits are available for network identifier. The fragmentation strategy of VXLAN is particularly poor.

It is possible to design a common fragmentation strategy applicable to all protocols, as explained in this paper. However, it may not be the most appropriate fragmentation strategy for VXLAN and Geneve although in theory it may work.

In 5G instead of carrying on with the routed backhaul/core network for packet transport, it is possible to move to a switched backhaul/core network that most likely would be managed by a centralized SDN controller. In such a network, many different packet formats are possible under a unified flow abstraction created by for example the Open Flow protocol. The formats include variants of 802.1ad/ah and MPLS. This would require the availability of high speed and economical Open Flow and other packet switches that submit to a centralized controller and scale well to the number of links needed in a large backhaul/core network. It is also likely that operators would like to make use of their past investments even in the case that they adopt this radical approach. Data plane nodes must be able to mangle packets from one forwarding format to another. For example, OF 1.3 supports many options of such mangling under unified control.

ACKNOWLEDGMENT

The first author thanks Nokia Bell Labs for permitting part of this work to be carried out during working time and for paying the salary. Also, Finnish education fund (koulutusrahasto) is acknowledged for adult education allowance (aikuiskoulutustuki) and Emil Aaltonen foundation for a grant.

REFERENCES

[1] C. Bettstetter, H.-J. Vogel and J. Eberspacher, "GSM phase 2+ general packet radio service GPRS: Architecture, protocols, and air interface," Communications Surveys & Tutorials, IEEE, vol. 2, no. 3, pp. 2-14, 1999. http://dx.doi.org/10.1109/COMST.1999.5340709

[2] A. Furuskar, S. Mazur, F. Muller and H. Olofsson, "EDGE: enhanced data rates for GSM and TDMA/136 evolution," Personal Communications, IEEE, vol. 6, no. 3, pp. 56-66, 1999. http://dx.doi.org/10.1109/98.772978

[3] E. Dahlman, B. Gudmundson, M. Nilsson and A. Skold, "UMTS/IMT-2000 based on wideband CDMA," Communications Magazine, IEEE, vol. 36, no. 9, pp. 70-80, 1998. http://dx.doi.org/10.1109/35.714620

[4] T. E. Kolding, F. Frederiksen and P. E. Mogensen, "Performance aspects of WCDMA systems with high speed downlink packet access (HSDPA)," in Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th, 2002. http://dx.doi.org/10.1109/VETECF.2002.1040389

[5] D. Astely, E. Dahlman, A. Furuskar, Y. Jading, M. Lindstrom and S. Parkvall, "LTE: the evolution of mobile broadband," Communications Magazine, IEEE, vol. 47, no. 4, pp. 44-51, 2009. http://dx.doi.org/10.1109/MCOM.2009.4907406

[6] Third generation partnership project (3GPP). 3GPP technical specification (TS) 29.060: General Packet Radio Service (GPRS); GPRS Tunnelling Protocol (GTP) across the Gn and Gp interface, version 13.2.0, 2015.

[7] M. Eubanks, P. Chimento, M. Westerlund. RFC 6935: IPv6 and UDP Checksums for Tunneled Packets, 2013.

[8] C. Perkins. RFC 2003: IP Encapsulation within IP, 1996.

[9] J. Touch. RFC 6864: Updated Specification of the IPv4 ID Field, 2013.

[10] D. Farinacci, T. Li, S. Hanks, D. Meyer, P. Traina. RFC 2784: Generic Routing Encapsulation (GRE), 2000.

[11] G. Dommety. RFC 2890: Key and Sequence Number Extensions to GRE, 2000.

[12] R. Bonica, C. Pignataro, J. Touch. RFC 7588: A Widely Deployed Solution to the Generic Routing Encapsulation (GRE) Fragmentation Problem, 2015.

[13] M. Mahalingam, D. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, C. Wright. RFC 7348: Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks

[14] J. Gross, I. Ganga, T. Sridhar. Internet Draft draft-ietf-nvo3-geneve-03: Geneve: Generic Network Virtualization Encapsulation

[15] J. Costa-Requena, R. Kantola, J. Llorente, V. Ferrer, J. Manner, A. Y. Ding, Y. Liu, S. Tarkoma, “Software defined 5G mobile backhaul,” in 5G for Ubiquitous Connectivity (5GU), 2014 1st International Conference on, 2014. https://doi.org/10.4108/icst.5gu.2014.258054