ETSI GR NIN 003 V1.1.1 (2021-03)

Non-IP Networking (NIN);
Flexilink network model

Disclaimer

The present document has been produced and approved by the Non-IP Networking ETSI Industry Specification Group (ISG)
and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference

DGR/NIN-003

Keywords

core network, fixed networks, internet, layer 3, network, network performance, next generation protocol, non 3GPP access

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Copyright Notification

Contents

Intellectual Property Rights 4

Foreword 4

Modal verbs terminology 4

1 Scope 5

2 References 5
2.1 Normative references 5
2.2 Informative references 5

3 Definition of terms, symbols and abbreviations 5
3.1 Terms 5
3.2 Symbols 6
3.3 Abbreviations 6

4 Layers 6
4.1 OSI model 6
4.2 Current practice 7
4.3 Layering in Flexilink 7
4.4 Communicating entities 7

5 Flows 7
5.1 Background 7
5.2 Flow identification 8
5.3 Routing of flows 8
5.3.1 Flow set-up 8
5.3.2 Software Defined Networking 9
5.4 Identification and location 9
5.5 Control plane protocols 10

6 Distribution of "header" information 10
6.1 Flexilink headers 10
6.2 UDP over IPv4 11
6.3 TCP over IPv4 11
6.4 RINA 11
6.5 Live audio 12

7 QoS, resource allocation, and congestion control 12
7.1 Guaranteed service 12
7.2 Basic service 12

8 Security, privacy, and error protection 13

History 14

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Group Report (GR) has been produced by ETSI Industry Specification Group (ISG) Non-IP Networking (NIN).

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document illustrates how the technology of ETSI GS NGP 013 [i.1] can carry multiple services, using as examples RINA, TCP/IP, and digital audio and video; including example packet formats and requirements for the control plane protocols.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI GS NGP 013: "Next Generation Protocols (NGP); Flexilink: efficient deterministic packet forwarding in user plane for NGP; Packet formats and forwarding mechanisms".

[i.2] ISO/IEC 7498-1: "Information technology -- Open Systems Interconnection -- Basic Reference Model: The Basic Model -- Part 1".

[i.3] ETSI GR NGP 003: "NGP Next Generation Protocol; Packet Routing Technologies".

[i.4] ETSI GR NGP 009: "Next Generation Protocols (NGP); An example of a non-IP network protocol architecture based on RINA design principles".

[i.5] ISO/IEC 60958-1: "Digital audio interface -- Part 1: General".

[i.6] ISO/IEC 62379-5-2: "Common control interface for networked digital audio and video products -- Part 5-2: Transmission over networks -- Signalling".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

flow: set of packets all of which follow the same route and experience the same level of service

QUIC: UDP-based transport and session-control protocol with claimed performance improvements over TLS/TCP

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

API	Application Program Interface
ARQ	Automatic Repeat reQuest
BGP	Border Gateway Protocol
CRC	Cyclic Redundancy Check
DNS	Directory Name Service
FIB	Forwarding Information Base
GTP	GPRS Tunnelling Protocol
HARQ	Hybrid Automatic Repeat reQuest
IP	Internet Protocol
NAT	Network Address Translation
NGP	Next Generation Protocols
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
QoS	Quality of Service
RIB	Routing Information Base
RINA	Recursive InterNetwork Architecture
SDN	Software Defined Networking
SDU	Service Data Unit
SIP	Session Initiation Protocol
TCP	Transmission Control Protocol
TDM	Time Domain Multiplexing
UDP	Unacknowledged Datagram Protocol

4 Layers

4.1 OSI model

ISO/IEC 7498-1 [i.2] specifies a model in which there are seven layers, each performing a different function within the process of conveying information from one device to another.

Each layer is implemented by an entity in one device exchanging Protocol Data Units (PDUs) with its peer entity in the other device. Except for layer 1, the bottom (Physical) layer, the PDU is transmitted by handing it to the layer below where it becomes a Service Data Unit (SDU) which is typically a byte string. Thus a layer *n*+1 PDU is encoded as a byte string which is transmitted as a layer *n* SDU to the peer layer *n* entity and handed up to the receiver's layer *n*+1 entity.

In most cases layer *n* is supposed to treat the SDU purely as a string of bytes, without assuming any internal structure. (Layer 6, the presentation layer, is an exception.) Each layer should be a "black box", with its interaction with the adjacent layers defined by the service interface specification in the OSI standards, so that any part of the stack can be changed without affecting the rest.

In many cases a layer *n* PDU will consist of a "header" containing information for the peer layer *n* entity followed by a "payload" which is a layer *n* SDU and hence also a layer *n*+1 PDU, and there is a perception in some fora that any packet will consist of a header for each layer followed by the application data. However, this is not necessarily the case, for instance a layer (such as the presentation or session layer in IP networks) may have a recognizable function but no header fields.

4.2 Current practice

In most systems there is a "protocol stack" consisting of a number of layers as envisaged by ISO/IEC 7498-1 [i.2] but with the entities forming the stack not corresponding exactly to the seven sets of functionality that it specifies. For instance, a GTP tunnel in LTE has two instances of layers 3 and 4 (IP and TCP/UDP), one below the GTP layer with the address serving as a locator and another above with the address serving as an identifier.

TCP and UDP include information from the IP header in their checksum fields, and port numbers in the layer 4 header are used to identify flows in layer 3 switches. Thus in practice, layers 3 and 4 in an IP network are not independent.

Shallow Packet Inspection is used in layer 3 entities to affect the service a packet receives based on analysis of the contents of higher-layer SDUs.

4.3 Layering in Flexilink

As with the OSI model, an entity sends information to a peer entity by encoding it in an SDU which is conveyed to the peer entity by a service implemented by entities which are in some sense the next level down in a stack. Much of the information which would traditionally be encoded per packet in headers is instead conveyed per flow in control plane messages, so a layer will not in general correspond to an identifiable "header" in the packet.

The hierarchy is less rigid than in OSI, so that it supports tunnelling (for instance). However, layers are not expected to examine the SDUs they carry, using control plane negotiation instead of guesswork based on higher-layer headers to determine the service a flow receives. An important consequence of this is that payloads can be encrypted without affecting the efficiency of transmission through the network.

4.4 Communicating entities

A network is considered to consist of "nodes" and "links"; each link carries packets between two or more nodes. A node can have the role of "endsystem" or "switch" or "middlebox". A node is not necessarily a piece of physical equipment; it may, for instance, be a process running inside a computer.

Traffic is originated and terminated in endsystems; typical endsystems include application processes and audio-visual equipment. Switches forward packets from their source towards their destination along a "route" which is an ordered set of links, without inspecting or processing the payloads.

Middleboxes, like switches, form part of a route, but they also process the payload data. An example might be a function that converts video from one format to another, so a live event that is multicast in high definition can be forwarded at a lower resolution to a device with a lower-bandwidth connection. Note, however, that many functions that in IP networks use middleboxes, such as NAT and firewalls, will instead be implemented in the control plane.

Also note that the routing of flows within a computing system is decoupled from the identification of protocols, unlike IP where port numbers take both roles. The driver software for a network interface should behave as a switch, forwarding packets according to their flow labels and leaving higher-layer processing to the endsystem process instances. The control plane messages show which higher-layer protocols are to be used when a flow is set up.

5 Flows

5.1 Background

The task of delivering a packet to a remote system may be divided into two parts: deciding its route through the network, and transmitting its content. IP was designed for an age in which both parts were implemented by the same component of the system, for instance a computer called the Interface Message Processor in the case of the ARPANET. In today's networks the two tasks are usually separated, being respectively control plane and forwarding plane functions.

Most packets are not an isolated event but are part of a flow, for example a TCP, QUIC, etc., session or an audio or video signal. In early networks, routing decisions were made independently for each packet, in order to avoid taking up memory space remembering previous decisions, but as memory became more plentiful "route caching" was used to reduce the load on the control plane. Now, with technologies such as SDN and OpenFlow, there is a trend towards a further separation of control and forwarding planes, with routing decisions being applied to flows rather than to individual packets. The interface between the two planes is a "routing table" which holds a record for each flow containing the information the forwarding plane needs to forward packets for that flow.

Flows can be aggregated as described in clause 5.3.2.4 of ETSI GS NGP 013 [i.1], with a group of flows that all follow the same route sharing a single entry in the routing table.

5.2 Flow identification

Currently the flow to which a packet belongs is identified by searching for a match on (typically) five fields in its header. The searching process requires multiple accesses unless expensive "content-addressable" memory is used. The Flexilink "basic" service specified in clause 5 of ETSI GS NGP 013 [i.1] replaces the per-flow information in a packet header with a "flow label" which is an index into the routing table, thus greatly simplifying the forwarding plane front end and also significantly reducing the size of packet headers; the label is local to each link over which the packet passes. The synchronous "guaranteed" service specified in clause 6 of ETSI GS NGP 013 [i.1] allocates specific transmission slots to each flow, and the flow to which a packet belongs is therefore, in effect, identified by the slot in which it arrives.

In both cases a "flow set-up" process is required, to supply the per-flow information that would otherwise be in packet headers, and route the flow through the system. This process is implemented via control plane messages in which each flow has a globally-unique identifier. If required, flows can be set up that carry individual packets (each with its own addressing etc information in a header which is seen by switches as simply the first part of the payload) between routers (which are middleboxes as defined in clause 4.4); different flows could support different addressing schemes or packet formats. Alternatively, datagrams can be conveyed in control plane messages which are similar to flow set-up but do not set anything up in the forwarding plane.

The flow set-up process has the potential to carry much more information than can reasonably be included in packet headers, and can incorporate several functions that in IP networks need to use separate protocols, as described below. It is a good fit with APIs such as Berkeley Sockets, where it would be implemented by the `connect()` function and there is a 1:1 correspondence between flow labels and socket handles. For live media it replaces (and can interwork with) SIP.

5.3 Routing of flows

5.3.1 Flow set-up

The flow set-up process creates a path through the forwarding plane which is followed by packets that are transmitted on the flow. When a mobile device moves to another cell, its flows are switched from one cell to the other without affecting the rest of the path. Similarly, paths may be rerouted for load balancing purposes, or to take equipment down for maintenance; the process should ensure that packets are not lost or duplicated, but in the case of the basic service the first packets on the new route can overtake the last packets on the old route. Paths on fixed networks may also be rerouted in the event of failure of a link or switch, and this rerouting can occur locally as soon as the failure is detected.

The way in which the route is chosen will typically depend on the way in which the required destination is identified. In the case of IP addresses, this may involve existing protocols such as BGP or in some networks all IP-addressed packets may be routed to the nearest connection to the current Internet. A request to access a specific piece of content may be routed to the nearest cached copy. A request to access a service may be routed to the nearest available instance of that service.

5.3.2 Software Defined Networking

The pioneering idea behind SDN is that networks have distinct control and data planes, and the separation of these two planes should be codified in an open interface. In the most basic terms, the control plane determines the route packets should follow through the network (using a routing protocol such as BGP), while the interconnected set of switches (or nodes) in the network implements a data plane, making forwarding decisions at each switch on a packet-by-packet basis.

In practice, decoupling the control and data planes manifests in parallel but distinct data structures: the control plane maintains a routing table that includes any information needed to select the best route at a given point in time (e.g. alternative paths, cost of each, applied policies), while the data plane maintains a forwarding table that is optimized for fast packet processing. The routing table is often called the Routing Information Base (RIB) and the forwarding table is often called the Forwarding Information Base (FIB).

The original interface supporting disaggregation, is called OpenFlow (introduced back in 2008), and although it was instrumental in promoting the SDN concept, it proved to be only a small part of what defines SDN today. Equating SDN with OpenFlow significantly undervalues SDN, but it is an important milestone because it first introduced the 'Flow Rules' as a simple but very powerful way to specify the forwarding behaviour.

A 'flow rule' is a Match/Action pair: Any packet that Matches the first part of the rule should have the associated Action applied to it. A simple flow rule, for example, might specify that any packet with destination address 'D' be forwarded on output port 'O'.

The Actions originally included "forward packet to one or more ports", "drop packet" and "send packet up to the control plane", an escape hatch for any packet that requires further processing by the controller, but then the set of allowed Actions became more complex over time.

Building on the 'flow rule' abstraction, each switch then maintains a 'Flow Table' to store the set of flow rules the controller has passed to it. In effect, the flow table is the OpenFlow abstraction for the forwarding table.

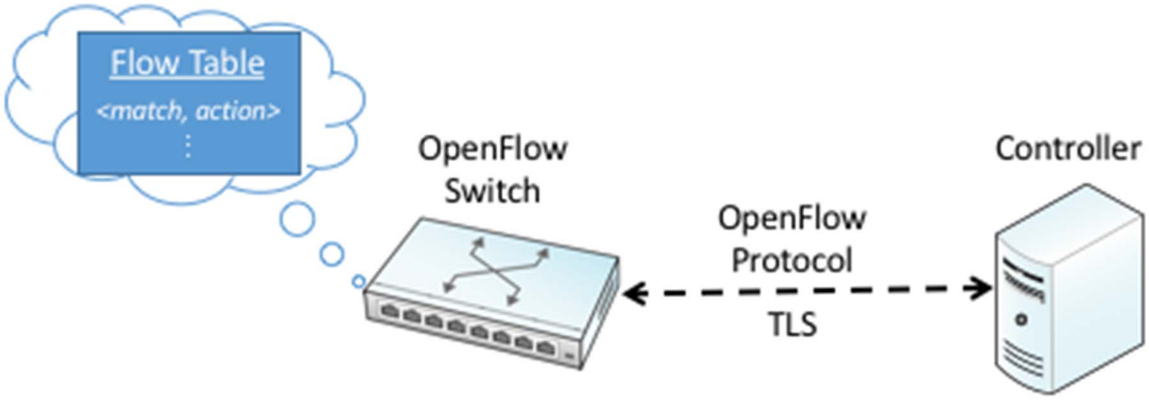

Figure 5.3.2-1: OpenFlow model

5.4 Identification and location

Nodes have both names, which identify them, and addresses, which specify their location, i.e. where to find them. This is analogous to the case with people, who when they move house have the same name but a different address. Also note that there may be several people (with different names) living at the same address, in the same way that there may be several devices connected to the same cell or broadband router, or several applications running on one physical device. Entities may also be identified by their function, in the same way that people can be identified by their job title.

A person's address is also an identifier for their house. In the same way, the identifier of a cell, gateway, etc, may serve as an address on a network. Locations are often hierarchical, with the scope at each level being the level above, for instance specifying a country, a town in that country, and a street in that town. E.164 addresses (telephone numbers) are another example.

Typically, the user (or the application) specifies an identifier for the remote entity, and the system finds the corresponding address and routes packets to it. For instance, in current systems, DNS is used to find an IP address for a domain name, and protocols such as BGP are used to find a route to that location.

Flexilink is agnostic as to the format and semantics of identifiers and addresses used to specify the destination of a flow. In a network there can, for instance, be some flows routed to IP addresses, others to RINA addresses, and others to content identifiers or service names; all are forwarded in the same way, so that the network service is to some extent virtualized. Flows need to have identifiers that are unique within any domain through which the flow might pass, for use in control plane messages, and clause 5.3.2.2 of ETSI GR NGP 003 [i.3] describes a possible format. These identifiers are not expected to be visible to applications (except for applications that are concerned with network management), and control plane messages should not require a rigid format for them; unlike IP addresses, they would not be used in the forwarding plane and therefore would not affect packet formats.

5.5 Control plane protocols

Control plane messages are used for processes such as setting flows up and tearing them down. More general network management, and particularly management of individual items of network equipment, may be considered as a separate piece of functionality referred to as the management plane.

The messages that set flows up include information on a number of different levels, including to:

- identify the flow;
- define the destination;
- specify or negotiate service parameters;
- describe the data format or protocol;
- authenticate the communicating parties; and
- agree charging information.

Some of these items of information are primarily for use by the network; others, such as the data format, are for the peer entity at the destination. In all cases the format should be as flexible and expandable as possible, for instance the destination may be defined by various kinds of address such as IPv4, IPv6 or RINA, or by a name (such as a domain name) or an identifier of a service or a piece of content.

Charges may be levied by any network across which the flow passes and/or by the destination service. There may need to be an exchange of messages before the forwarding plane flow is set up, for instance for authentication or to agree which of several possible data formats (perhaps requiring different amounts of bandwidth) is to be used.

6 Distribution of "header" information

6.1 Flexilink headers

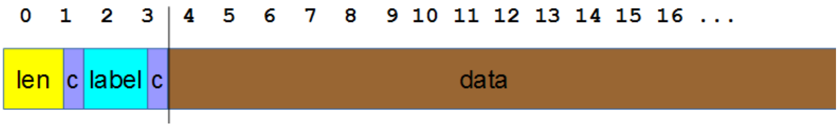

Figure 6.1-1: Flexilink basic service packet

Network packet headers will normally contain only a flow label and an indication of the packet's length, as illustrated in Figure 6.1-1 where "c" represents an integrity check (such as a checksum or CRC) on the preceding field.

Per-flow information can be conveyed in control plane messages.

Per-packet information for higher layers will be seen as part of the payload, and not processed by intermediate nodes. This applies not only to end-to-end information such as sequence numbers and checksums for layers 4 and above in the OSI model but also to headers for layers that come lower in the OSI model, for instance in the case of tunnelling or flow aggregation.

6.2 UDP over IPv4

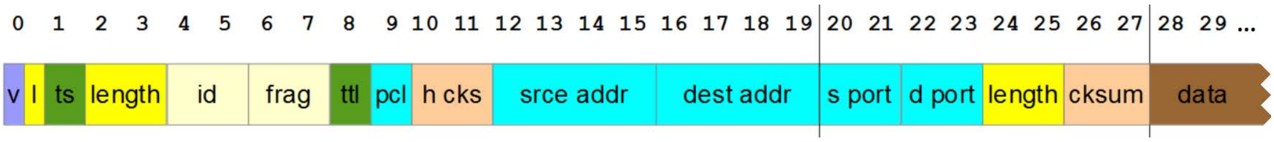

Figure 6.2-1: UDP/IPv4 packet

Figure 6.2-1 shows IPv4 and UDP header fields in the case where there are no extension fields in the IP header, so the first byte contains 0x45. The whole IP packet can be carried as the payload of a Flexilink basic service packet, perhaps on a flow that terminates at a gateway to the Internet. However, it will often be more efficient to create a Flexilink flow that corresponds to the IP flow; adding the IP header for onward transmission on an IP network would be similar to the Network Address Translation (NAT) process used in current networks.

Fields that define the IP flow are shown in blue; this information, along with QoS negotiation which replaces byte 1 (type of service), would be included in the flow set-up message. Byte 8 (time to live), which protects against circular routing, is not required.

If fragmentation is required it should be implemented as a separate layer, with bytes 4 to 7 (or equivalent information) forming part of the payload of the packets that carry the fragments.

The length is included in the Flexilink header, which, like the control plane messages, has its own data protection. The UDP checksum may be included in the payload of the Flexilink packet, or one of the protection formats specified in clause 5.3.2.3 of ETSI GS NGP 013 [i.1] can be used instead.

6.3 TCP over IPv4

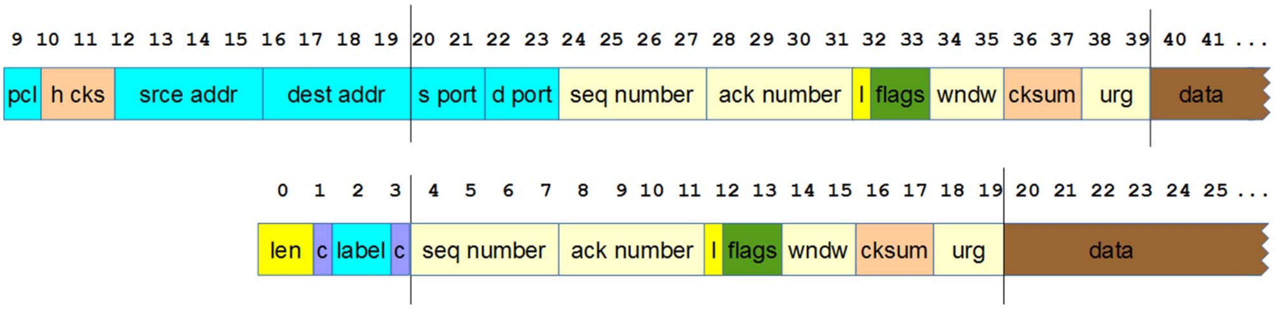

Figure 6.3-1: TCP over IP and over Flexilink

When not tunnelling the whole IP packet, the information from the IP header is transferred to control plane messages and the "length" field of the Flexilink header in the same way as for UDP. The remaining information is seen by the network as part of the payload.

6.4 RINA

Referring to clause 6.2.2 of ETSI GR NGP 009 [i.4], the information in all fields up to and including the PDU type would be conveyed in the control plane at the time of flow set-up. The length would be signalled by the "length" field of the Flexilink header.

The flags, sequence number, and user data would be carried as the payload.

6.5 Live audio

Digital audio would be carried in the guaranteed service. An example is 44,1 kHz stereo encoded according to ISO/IEC 60958-1 [i.5]. Clause 7.3 of ISO/IEC 62379-5-2 [i.6] provides detail of a possible format, including how it can be signalled in control plane messages; it requires 4 to 8 bytes per sample depending on which options are chosen.

The control plane messages implementing the flow set-up process (for video as well as audio) will include all necessary addressing information as well as the requested number of slots per second and a specification of the encoding used. They may include different options, e.g. for high quality if capacity is available, lower otherwise. They may also include other information, such as determining whether the caller is entitled to consume the content.

Audio samples at the source are assumed to be collected into a buffer, with a packet being transmitted in each of the slots that are allocated to the flow. Each packet contains the samples that have been accumulated since the previous packet. If the flow is set up with 8 000 slots per second, and the slots are evenly spaced, each packet will contain either 5 or 6 samples. At the destination, incoming samples are put in a buffer from which they are taken one by one for onward transmission or conversion to analogue. The flow set-up process will report an estimate of the end-to-end latency and the distribution of the incoming slots will define the amount of jitter in arrival times. Thus the amount of buffering required at the destination can be determined when the flow is set up.

The flow to which an incoming packet belongs is determined by the slot in which it arrives.

Except when rerouting around a fault or when handing over to a different radio cell, the latency (including packetization time) will be fixed to within one or two sample times, and at handover the change in latency can be signalled by a control plane message if required. For many applications, only the audio data will need to be transmitted; this, along with lightweight control plane protocols, makes the design of devices such as digital microphones very simple. Where sample-accurate timing is required the format cited above provides an economical mechanism for including the necessary information.

Similar considerations apply to video.

7 QoS, resource allocation, and congestion control

7.1 Guaranteed service

The guaranteed service specified in clause 6 of ETSI GS NGP 013 [i.1] requires each flow to be allocated transmission slots on each link over which it passes, as part of the control plane process of setting the flow up. This process can include negotiation between an application and the network, for instance in the case of compressed video to trade-off between image quality and network bandwidth. It is also possible to negotiate addition or removal of slots while the flow is connected, to react to changes in network load. However, except under fault conditions the negotiated bandwidth and latency will always be available to the flow.

The amount of forwarding plane information a node has to keep for each link that implements the synchronous service specified in clause 6.3 of ETSI GS NGP 013 [i.1] depends only on the data rate of the link and the length of an allocation period. The latter is defined by *m*, which is a design parameter; thus the size of the routing table is fixed and does not depend on the number of flows.

Unused space in slots (including the whole of unused slots) is available to the basic service, so (unlike, for instance, TDM systems) capacity is not wasted if packets are less than the maximum size; and a variable-bitrate flow can simply request an allocation for its peak rate.

7.2 Basic service

The basic service is defined as a best-effort service, with packets queuing for transmission on each link and being dropped if queues overflow. Various QoS mechanisms similar to those used in IP networks could be implemented, being signalled in control plane messages and with routing tables being expanded to include additional information, but it is likely that this will not be required as applications needing QoS will use the guaranteed service.

There is, however, one traffic class that needs special treatment: control plane messages need a separate, higher-priority, queue so that they can still be transmitted when there is an overload of data.

There is also a resource that needs to be allocated to basic service flows, namely flow table entries. Unlike the guaranteed service, the bandwidth of a link does not place a restriction on the number of flows it can carry; potentially an application could leave a flow connected for long periods without sending any packets at all. Switches should therefore have the ability to detect when a flow is not carrying any packets, and clear it down; the control plane protocols can include provision to mark flows as requiring not to be cleared down in such circumstances, or to adjust the timeout. Flow set-up messages should in any case include an estimate of the amount of data to be sent. Where appropriate, flow aggregation as described in clause 5.3.2.4 of ETSI GS NGP 013 [i.1] can be used to reduce the number of flow table entries required.

8 Security, privacy, and error protection

Forwarding plane routing in a node is controlled by the node's control plane entity. This can either implement protocols such as flow set-up directly or accept commands from a remote controller. In the former case, it communicates only with neighbouring nodes, either in the same network or border nodes in peer networks; in the latter case, appropriate procedures to prevent spoofing of the commands by remote attackers will be needed.

The flow set-up process allows a server to perform appropriate checks on the identity of a client before exchanging any data. At the other extreme, a client can be completely anonymous, whereas in current systems it needs to disclose its IP address. The security risks of fixed IP addresses and well-known port numbers are also avoided.

The flow set-up process can also include controls on the route to be followed, such as to avoid certain jurisdictions or to avoid networks that are in some sense untrustworthy. Border nodes can also implement additional checks on external requests, and can disconnect flows that are identified as, for example, being part of a denial-of-service attack.
A distributed denial of service attack might still be indistinguishable from a legitimate overload, but as each of the systems involved would need to set a flow up the overload may well be restricted to the control plane.

Guaranteed service flows are not affected by traffic on other flows because each flow has its allocation of slots reserved for it. This also means a guaranteed flow cannot use more capacity than has been allocated to it.

Packet headers include data protection. Protection of payloads is expected to be end-to-end; this provides protection against faults in switches as well as transmission errors on links. It also allows applications to use appropriate protection, for instance acknowledgement and retransmission, forward error correction, or using interpolation to replace lost data, depending on the application's requirements for timeliness. In a packet containing live uncompressed audio each sample should be protected separately so that a single-bit error will not result in the whole packet being discarded.

Errors in fixed equipment and wired links are expected to be rare. Wireless links are expected to mitigate the effects of interference etc, for instance via local ARQ and HARQ; an application can discover from flow set-up and other control plane messages the likely aggregate performance of the links over which its flows pass.

Flow identifiers include provision for multiple copies of a flow to be sent by different routes, to give added resilience. They also provide for multiple forwarding plane flows carrying different kinds of content to be bundled together and routed as a bundle, as required on connections between broadcasting studios for example. Back-up routes can also be established, carrying a second copy of the most important flows, with the remaining flows being added to it if the primary route fails.

History

Document history		
V1.1.1	March 2021	Publication