NEW NETWORKING TECHNOLOGY IN ETSI ISG NIN

John S Grant
Nine Tiles, Cambridge CB25 9HT, UK

NOTE: Corresponding author: John Grant, j@ninetiles.com

Abstract *– ETSI ISG Non-IP Networking (NIN) was formed to standardize Flexilink, a networking technology which does not have the problems mobile operators had identified with current technology based on Internet Protocol (IP). This paper reviews the main features of Flexilink and the advantages of the services it provides for future network applications beyond 2030. Internet Protocol is connection-less, so every packet has to carry all the information needed to route it to its destination and this information has to be processed by each router the packet passes through. However, in most cases it is used to carry a "flow" of packets for a connection-oriented protocol such as TCP or RTP. Routing in SDN applies to flows rather than individual packets, but each packet header still contains the same routing information and must be processed to identify the flow to which the packet belongs. Flexilink replaces the per-flow information in the packet header with a pointer into the routing table, which eliminates the need for software processing of packet headers and means that packet formats are not dependent on the addressing scheme used. It also provides a separate service for live media such as audio, video, and tactile that easily achieves the lowest possible latency.*

Keywords – 6G, ETSI, Flexilink, latency, mobile, non-IP, routing

1. INTRODUCTION

ETSI ISG Next Generation Protocols (NGP) was formed to investigate a number of problems mobile operators had identified with technology based on Internet Protocol (IP). One issue was the amount of capacity on the air interface taken up by the multiple packet headers required to support mobility (via GTP) and security (via IPSec). Another was the difficulty of supporting new features promised for 5G: massive Machine-Type Communication (mMTC) and Ultra-Reliable Low Latency Communication (URLLC). It identified Flexilink as a technology with much lower overheads and which also natively supports the lowest possible latency, and ETSI ISG NIN was formed to standardize it. This paper reviews the main features of Flexilink and the suitability of the services it provides for future network applications.

A proof-of-technology implementation exists over the gigabit Ethernet physical layer, and ISG NIN is currently investigating implementation over wireless physical layers including 3GPP 5G NR (New Radio) [11], DECT-2020 NR [13], WiFi, and Bluetooth.

2. PACKETS AND FLOWS

In a typical scenario, two program instances exchange information over an IP network using a "sockets" API, and a "flow" is the set of packets that are transmitted from one socket to the other. Packets (or rather, their payloads) are identified within the programs by associating them with a "handle" value that identifies the socket. On the network, each packet's header includes all the information needed to route it to its destination and to identify it on arrival, including globally-significant destination and source addresses.

That was appropriate when IP was first specified, over 40 years ago, because it meant that a router did not need to keep any information about the various flows that passed through it, and there was no need for any formal process to reroute flows around faults. Communication links were slow enough that the packet headers could be processed by software. The amount of information routers needed to keep to be able to route to any address was manageable.

Now, communication links are about a million times as fast but processors are only about a thousand times as fast, so software processing of every packet header becomes more difficult. Also, there are now many more possible destinations, so the address tables that are needed to be able to route to any destination become more difficult to manage.

Software-Defined Networking (SDN) addresses these issues by dividing the network functions into a control plane, which is implemented in software and decides the route for each flow, and a forwarding plane, which transfers the packets from input to output in switches. A “flow table” (written by the control plane, read by the forwarding plane) has an entry for each flow, showing on which output it should be transmitted together with any other information needed for the forwarding process such as whether the packets should go into a higher-priority queue.

Thus the control plane only needs to process each flow, not every packet; and the forwarding plane only needs to identify and interpret the flow table entry for each packet, a task which should be simple enough to be done entirely by hardware. However, identifying the flow table entry for an IP packet is not trivial: it involves searching the table for a match on several fields including source and destination addresses and port numbers. Fig. 1 shows the header information that identifies the flow in blue.

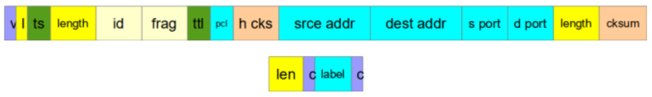

Fig.1- UDP/IP header (top) and Flexilink header

2.1 Addressing in Flexilink

Flexilink takes the transition begun by the introduction of SDN to its logical conclusion; the packet header contains only the length of the packet and a “label” which points to the entry in the flow table. The entry shows the label for the next hop as well as identifying the output. This greatly simplifies the hardware needed to implement the forwarding plane.

The simple table look-up requires only one or two clock cycles whereas the time for a table search will depend on how many entries need to be examined. Thus it is much easier to minimize the latency of packet forwarding with Flexilink.

Flows are set up by exchanging control plane messages. This allows the applications to agree details such as coding and higher-layer protocols, also security and authentication, before sending any data.

There are many other benefits to this method of setting up flows, which have perhaps been overshadowed by religious objections to anything that appears connection-oriented or by association with systems such as Asynchronous Transfer Mode (ATM) which have been unsuccessful for other reasons.

It allows many different forms of identification for the destination, eliminating the need to find an IP address using protocols such as DNS or SIP. This in turn removes the requirement for addressing to be the same across the whole network, which has caused adoption of IPv6 to be so slow, and made it so difficult to introduce new schemes such as Information-Centric Networking (ICN).

Network elements do not need to store address tables. When setting up a flow the route can be established by querying an external database; if the destination is a URL or the name of a service or an E.164 address (a telephone number), this takes no more effort than the DNS or SIP transaction that would otherwise be needed.

Although flow set-up still requires software processing, for most applications this is much less time-critical. For instance, performers receiving their own sound in in-ear monitors can detect latency of as little as 5 ms in the audio, but a delay of a few hundred milliseconds from requesting a connection to hearing the sound will be barely noticeable.

It is easy to identify loops when setting up a flow, removing the need for protocols such as Spanning Tree and for “time to live” fields in packet headers. Flows can easily be rerouted around faults or when handing a mobile device over to another base station.

2.2 Benefits for future networks

As file sizes get larger, with photos and videos having more pixels, software packages having more code, web pages having more content, and 3D video being used for telepresence, it will be increasingly important to reduce the amount of per-packet processing, moving as much as possible of the work to per-flow processing in the control plane. This also reduces the energy requirements of network equipment, another issue that will become increasingly important.

Line speeds, and the throughput of the hardware that implements the physical layer, are expected to continue to increase. This same increase in throughput would also apply to hardware implementing a forwarding plane. Processor (and hence software) speeds are unlikely to increase significantly, though.

Increasing numbers of IoT devices will be sending or receiving periodic data over wireless networks, for instance a thermometer sending a reading every few minutes. Each reading might only be 2 bytes, so adding even 48 bytes of header (for IPv6 + UDP) would be a 2400% overhead. Currently there are several different wireless IoT systems using more economical formats, but Flexilink would have a low enough overhead that its standard format could be used. Transmission over DECT-2020 NR is being standardized as ETSI GS NIN 004 [13].

3. SERVICE FOR LIVE MEDIA

Many future applications will have a requirement for sending continuous signals (such as digital audio, tactile feedback, or the position of robots in a factory) with sub-millisecond latency.

These kinds of flow transmit packets at regular intervals, and to control latency it is necessary to ensure that each packet can be transmitted with no more than a specified delay. With conventional packet networks this requires time-critical packets to be given priority over other packets, which in turn requires at least two separate queues for each output. It also requires "policing" the rate at which each source transmits, and "shaping" the flow to ensure that packets do not become bunched together. There are limits to how low the latency guarantee can be, for instance if a high-priority packet becomes available for transmission just as a maximum-length low-priority packet has begun, it must wait until the low-priority packet finishes. Also, if packets from several high-priority flows become available for transmission on the same link at the same time, some of them will need to wait.

Another contributor to latency is the time required to fill a packet. A typical uncompressed audio signal might have 4 bytes per sample and 44.1 samples per millisecond; if packets include GTP, IPv6, UDP, and RTP headers, and are sent once per millisecond, the headers will add about 60% to the size of each packet, and packetization will contribute 1 ms to the latency. Sending them less often to reduce the overhead will increase the latency; sending them more often to reduce the latency will increase the overhead.

3.1 Flexilink guaranteed service

The requirements for live continuous media are sufficiently different from those for traditional packet data that Flexilink provides a separate "guaranteed" service with characteristics that make it easy to provide latency that is stable and lower than can be provided by its "basic" service (described in Section 2.1 above) or by traditional networks. It is multiplexed with the basic service in a way that ensures guaranteed service packets are transmitted in their scheduled timeslots while allowing the basic service to use all the capacity that is not used for guaranteed service packets.

Guaranteed service packets consist of a 1-byte header and up to 63 bytes of data. The flow a packet belongs to is identified by the slot in which it arrives, so no label is needed and the header is not changed when the packet is forwarded. This contrasts with basic service packets, which carry up to 2000 bytes of data and include a label which is changed each time the packet is forwarded. It might be possible for future optical networks to forward guaranteed service packets in the optical domain.

Transmission slots are 64 bytes each; the first byte of each slot shows how many guaranteed service data bytes are present, and the remainder of the slot (following the guaranteed service data) contains basic service bytes. Thus the basic service is carried as a byte stream consisting of headers, data, and, when there are no packets to transmit, idle bytes; and this stream simply pauses when guaranteed service bytes, or framing bytes, are transmitted. There is no need to add any kind of fragmentation header.

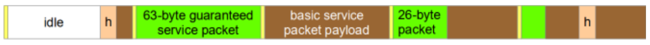
idle
h
63-byte guaranteed service packet
basic service packet payload
26-byte packet
h

Fig. 2 - Example of 5 consecutive slots; h = basic service header

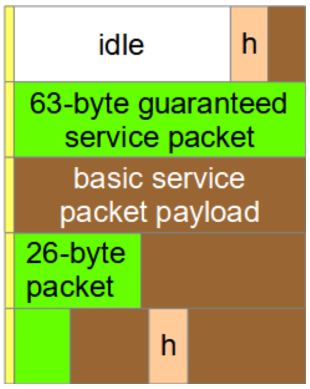
idle
h
63-byte guaranteed service packet
basic service packet payload
26-byte packet
h

Fig. 3 - Example from Fig. 2 rearranged as 5 rows of 64 bytes

The example in figures 2 and 3 shows five consecutive slots, with a maximum-length guaranteed service packet in the second slot and shorter ones in the fourth and fifth. A guaranteed service packet becomes available for transmission during the first slot, and a second becomes available before the first has finished; note that no inter-packet gap is required.

This format is easy for a MAC layer implemented in hardware to support, combining (on transmission) and separating (on reception) a continuous stream of guaranteed service slots and an intermittent stream of basic service bytes, as illustrated in Fig. 4. A similar scheme is used on wireless networks.

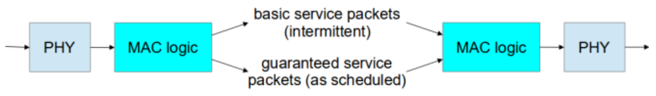
PHY
MAC logic
basic service packets (intermittent)
guaranteed service packets (as scheduled)
MAC logic
PHY

Fig. 4 - Multiplexing of services

Originally, it was proposed that slots would be of variable size, to support very short packets, for instance containing a single audio sample, and also packets containing several kilobytes of data, to support high data rate applications such as tele-presence. This was partly in order to be different from ATM's fixed-size cells.

To keep up with the total throughput required, the internal data paths in a switch need to be quite wide, and there is little benefit in having a slot size less than that width. Elements of a high data rate flow, such as uncompressed video frames, will in any case need to be divided into multiple packets, and if the packetization etc. is done in logic rather than by software there is no significant benefit in using larger packets; indeed, there may be some cost, e.g. larger registers for counting payload bytes.

Large slot sizes would also be a problem, because if a link is carrying several flows with frequent short slots it might be impossible to find a long enough gap to allocate a long slot for a new flow. Furthermore, signalling the size and position of slots adds significant complexity compared to a format with fixed-size slots starting at fixed positions within the frame.

Therefore, fixed size slots were chosen. Unlike ATM, unused space at the end of a slot is not wasted because it carries the basic service. The header format allows 6 bits for the data length, so an extra header byte would be needed for 128-byte slots, thus the overhead is 3.1% for 32-byte slots and 1.6% for 64 or 128. At 64 bytes an MPEG transport stream packet fits neatly into three slots.

3.2 Guaranteed service forwarding

Whereas basic service packets are forwarded in much the same way as in traditional packet networks, the process for guaranteed service packets is very different and in some ways more similar to the routers used for analogue (and point-to-point digital) audio and video. Fig. 5 shows the structure of a Flexilink switch, with separate data paths and configuration tables for the two services.

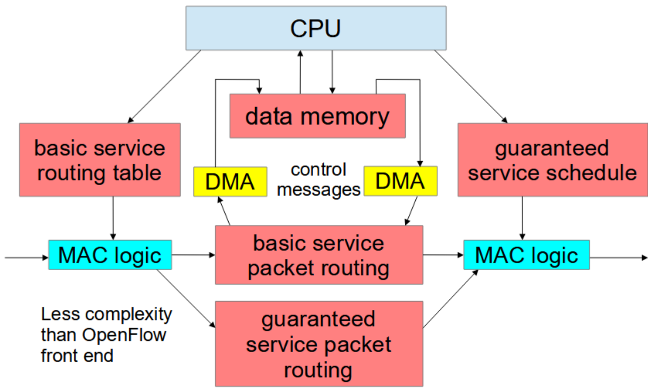
CPU
data memory
basic service routing table
DMA
control messages
DMA
guaranteed service schedule
MAC logic
basic service packet routing
MAC logic
Less complexity than OpenFlow front end
guaranteed service packet routing

Fig. 5 - Flexilink switch

The framing on all links is phase-locked, i.e. if slot 1 of an allocation period on link A is transmitted at the same time as slot *n* on link B then *n* will be the same for every allocation period. This has been found to be very easy to implement, and does not require any clock frequencies to be adjusted.

The content of all incoming slots is written to a circular buffer which therefore at any time contains all guaranteed service packets received in the last few microseconds (about 15 microseconds in the case of the gigabit Ethernet physical layer). A slot's address in the buffer is composed of an identification of the network port on which it arrived together with its position in the allocation period modulo the number of slots from that port in the buffer.

For each outgoing slot, the routing table shows the address to be read in the buffer. Thus the routing table is on the output side, whereas the basic service routing table is on the input side, and the size of the table for each network port depends only on the number of slots in an allocation period. A flow can be multicast by simply reading its slots for more than one output, so it is easy to multicast a live sports event (for example) to a very large number of destinations.

There is no need for traffic shaping or policing because each flow has its own set of slots in which its packets are transmitted and it cannot transmit more data than will fit into those slots.

3.3 Benefits for future networks

Many of the applications proposed for future networks and, indeed, for 5G, require latency to be within defined limits. Mixed reality, for instance, requires less than 15 ms including processing time and two transits across the network. The sound from a performer's microphone needs to be transported across the network, through a mixing desk, and back across the network to an in-ear monitor within 5 ms.

Flexilink's guaranteed service easily provides fixed latency, lower than is possible with current systems. Assigning slots to each flow ensures that each flow gets the service negotiated for it, independently of other traffic on the network, eliminating the need for careful network management that is a feature of current networks that carry live professional media. Using the guaranteed service for flows needing a defined service level means that measures such as multiple queues with different priority levels are not needed for the basic service.

The service's ability to natively provide multi-casting, noted above, will become increasingly important as viewing of live content such as sports events moves from over-the-air broadcasting to online.

Although primarily intended for continuous media, the service can also be used for transferring large files. Transport layer protocols such as TCP provide mitigation for loss of packets when buffers overflow and limit the data rate to a value the network and the destination can support. Setting up a guaranteed service flow, at a data rate negotiated with the network and the destination, eliminates the need for both these functions. Also, mass downloads such as software updates can be multicast to large numbers of destinations.

4. SIGNALLING

The formal specification of the control plane protocols is currently being written and will be published as ETSI GS NIN 005 [1]. Messages use a tag-length-value format and contain Information Elements (IEs) in a similar way to Q.2931 [2] and 3GPP S1AP [3].

A transaction can have up to four phases, as illustrated in Fig. 6; however, for most transaction types only one or two are required. FindRoute transactions are used to establish flows; in the case of a bidirectional basic service call, to make the connection quickly the forwarding plane flow towards the caller can be set up as the request message is processed, and in the opposite direction by the response message. Note that this means that the path on which to transmit (in either direction) has been set up by the time the message is received, and the other two phases are not required. In this case the client can be completely anonymous; it does not need to disclose its address to the server.

Alternatively, the first two phases can be used for authentication, key exchange, etc. and the flows set up in the forwarding plane by the other two phases, so the path is not set up in the forwarding plane until after the security checks have been completed.

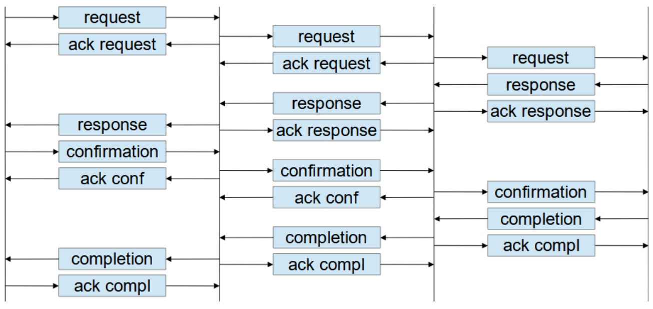

Fig. 6 - Signalling transaction phases

Each flow is identified in the control plane by a 128-bit globally unique identifier, and each control plane entity keeps a small amount of information about each flow that passes through it. This information can be kept in slower memory, such as dRAM, as it is not needed for the per-packet processes in the forwarding plane, but only by the software that manages flows. Multiple basic service flows can be tunnelled through a trunk network as a single flow, similar to ATM virtual paths and the "push" operation in MPLS; intermediate nodes in the trunk network only need a single table entry for the tunnel.

Control plane entities do not need to keep address tables; information on how to reach any particular kind of destination can be retrieved from a server in the same way that DNS is used in current networks, and the information for popular destinations can be cached. The information is only required when setting a flow up so fetching it from a remote location does not impact the speed with which data can be forwarded.

Flows can be bundled together to form a "call". One example would be a connection between studio and venue for an outside broadcast, carrying high quality programme audio and video to the studio and lower quality in the other direction, as well as intercom and camera control. Each flow has its own table entry in the forwarding plane, but only one FindRoute transaction is needed in the control plane. Flows can be added to a call via AddFlow transactions or removed via Terminate transactions. ChangeAllocation transactions can be used to negotiate changes in bandwidth for guaranteed service flows, for instance to react to changes in network loading.

IEs in FindRoute messages can include a wide variety of information for the network and for the remote application, including setting up security and authentication and negotiating call charges as well as specifying data formats and higher-layer protocols. Entries in forwarding plane routing tables can be set up immediately or only after negotiation, perhaps to establish the bandwidth available for compressed media (and hence the compression ratio), or to authenticate the caller.

A wide variety of addressing options are defined in addition to legacy address types such as IPv4 and IPv6, as noted in Section 2.1 above. A call to consume a particular piece of content can be routed to a locally cached copy if one is available, to a central resource otherwise.

Other message types include FrameSyncInfo which orchestrates the process of aligning slot allocation periods on different links, and TimeSyncInfo which configures the distribution of network time in the forwarding plane to an accuracy similar to PTP [4]. On wired links the current network time (modulo 4 seconds) is included in every frame, providing a more frequent update than PTP but with less overhead and no requirement for software intervention.

5. MIGRATION

Flexilink services can be carried over IP networks, and IP can be carried over Flexilink, so Flexilink “islands” can be introduced into an IP network and then expanded as the network grows or equipment needs to be replaced or a requirement arises to carry traffic that needs Flexilink features such as ultra-low latency or low per-packet overheads. An island can be anything from a single piece of equipment to a large network.

Flexilink-aware applications connected to the same island will use Flexilink throughout; if they are connected to different islands, signalling messages will be tunnelled between islands over the IP network in the same way as user packets, and the FindRoute messages will report the level of service to be expected for guaranteed service flows.

Applications running in devices that are part of a Flexilink island can continue to use IP addressing. If they access the network via an API such as Berkeley Sockets, the flow will be set up by the connect() call. The IP packets (v4 or v6) can be carried with their headers included, in the same way as over MPLS, or the headers can be added by the egress node from the island to the IP network. In the latter case the routing table has to hold the necessary information to construct the IP header, but an all-IP network would need to hold a similar amount of information if it would be doing address translation at that point, as in a mobile network’s packet gateway or the router that connects a local network to a broadband service.

Flexilink is particularly appropriate for private networks supporting new applications that would be difficult to support with current technology, for instance where guaranteed ultra-low latency is required in factory automation, augmented reality, or remote medicine; or for IoT applications that need to minimize each device’s transmissions, to save battery or radio spectrum.

6. IMPLEMENTATION

As a proof of the technology, a platform consisting only of a Xilinx Spartan 6 SLX45T FPGA, flash, dRAM, and physical interfaces has been used to implement a Flexilink switch with up to eight network ports, is illustrated in figures 7 and 8. The network ports support gigabit Ethernet. and there are also a number of digital audio ports to allow audio interfaces to the network to be prototyped. For four of the network ports the FPGA is connected to copper Ethernet PHYs via RGMII; for the others its serial transceivers are connected directly to SFP cages which can also be used for digital video at up to 2.97 Gb/s.

The guaranteed service forwarding plane is implemented entirely in the FPGA, while the basic service’s packet buffers are in the dRAM. The software that implements the control plane runs on a soft processor, also implemented in the FPGA. The MAC logic supports both Flexilink frames and Ethernet packets, auto-detecting which format is being received; in Ethernet mode the requirement to interpret MAC and IP headers adds considerably to the complexity of the logic, and some per-packet processing by software is still needed. If Ethernet and IP are not supported, there is enough room in the FPGA for video codecs implementing wavelet compression.

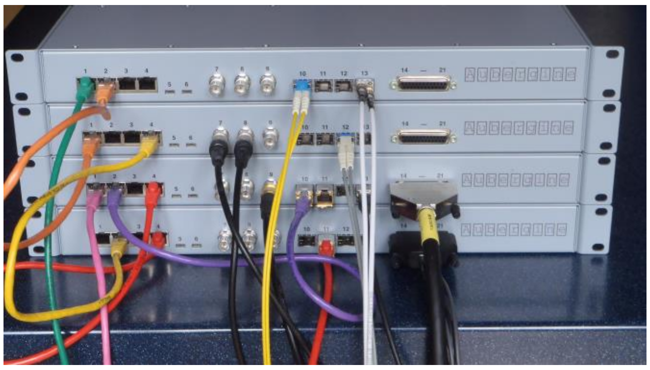

Fig. 7 - Photograph of implementation platform

The design could easily be adapted to implement a larger switch or an audio or video interface. The BoM for a Flexilink switch using an FPGA platform and built in small quantities has been estimated to be less than for a current-technology layer 3 switch with equivalent functionality built in large quantities on an SoC platform. Instead of supporting legacy formats on Flexilink equipment, it might be better to have a middlebox that interfaces to Flexilink on one side and IP on the other, and which can include the functionality of a firewall and Network Address Translation (NAT).

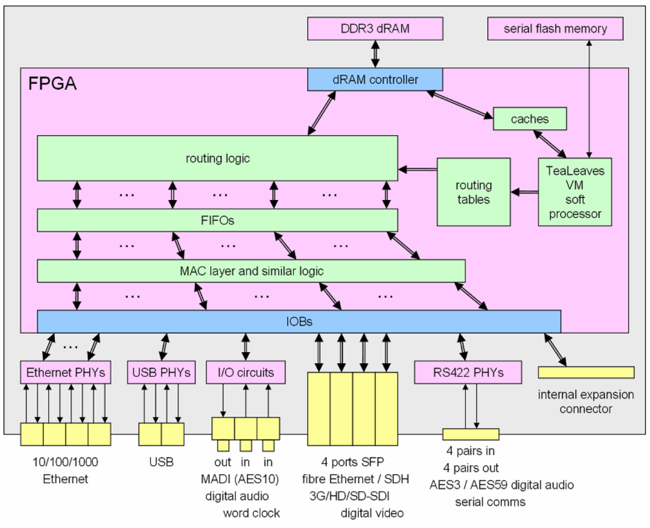

Fig. 8 - Implementation platform internal structure

Currently the user interface is a Windows application that runs on a computer plugged into one of the Ethernet interfaces. It crawls the network looking for switches and audio and video ports, and media flows are set up by clicking on the source and destination. This sends a command to the destination to send a signalling message requesting connection to the source. It would also be possible for media interfaces to include a user interface through which connections can be requested.

6.1 Power consumption

Power consumption of the units is very low; no fans are required unless several power-hungry SFP modules are used, and the FPGA does not require a heat sink.

Implementation of the forwarding plane entirely in logic contributes to the low power consumption; compared to a technology that requires packet headers to be processed by software, which involves fetching and executing CPU instructions, many fewer transistors will switch for each packet forwarded.

Power consumption is further reduced by carrying out security checks when a flow is set up, rather than needing to examine every packet as is the case with current firewalls.

6.2 Current projects

At the time of writing, a project to install a Flexilink network linking several buildings on the Birmingham City University central campus is at an early stage. It is planned that the system will be used for remote musical performance, and be extended to include other universities also.

Current development effort is focussed on implementation over wireless physical layers [11,13], using software defined radio for 3GPP NR and existing physical layer silicon for other standards.

6.3 Open challenges

The technology still has to be demonstrated for other applications that need ultra-low latency or large numbers of short messages, such as in industrial automation.

The power consumption advantage also needs to be quantified; there are some applications, such as data centre networks, for which it will be particularly important.

Once the technology is established in private networks, it can begin to be introduced into public networks. This is a much simpler process than the transition to IPv6 in the Internet (see Section 5 above), at least from a technical point of view, and would eliminate the delays and drop-outs that currently plague videoconferences.

7. STANDARDS

Flexilink has its origins in work on audio over ATM and ATM over Ethernet PHY in the early 21st century. The respective standards are AES47 [5] and AES51 [6].

For management of network elements, the multi-part IEC 62379 standard [7] was developed, originally for audio over ATM but then extended to cover video and other digital media as well as audio, and other networking technologies than ATM. Also, AES47 was republished as IEC 62365.

Current standardization is taking place in ETSI. From 2016 to 2019 ISG NGP studied problems mobile operators were experiencing with IP [8], including the amount of spectrum that was being used to carry IP headers and other overheads and also the difficulty of achieving the ultra-low latency that was promised for 5G. ISG NGP investigated both enhancements to IP and a "clean slate" approach, and in 2020 ISG NIN was formed to standardize Flexilink [9] as a clean slate solution, the name non-IP networking emphasising that it is not dependent on IP's mechanisms although it does support IP along with other protocol sets.

At the time of writing, ISG NIN has published three reports: reviews of the problems to be addressed [10] and of the options for implementation over 5G radio [11], and a high-level review of Flexilink [12]. Two standards are currently under development: the use of DECT-2020 NR as a physical layer [13] and the Flexilink control plane [1].

8. CONCLUSION

Flexilink provides a better and more efficient service for 21st century applications than the TCP/IP suite. The forwarding plane is simpler, reducing capex for new networks. The network is simpler to configure and to operate and power consumption is expected to be lower, reducing opex. Interworking with legacy technologies allows the new technology to be introduced gradually, with benefits available immediately rather than having to wait until the whole network has been converted.

REFERENCES

[1] ETSI GS NIN 005: Non-IP Networking (NIN); Flexilink signalling messages and protocols. ETSI, Sophia Antipolis, France. Available: https://www.etsi.org/deliver/etsi_gs/NIN/001_099/005/01.01.01_60/gs_NIN005v010101p.pdf.

[2] Q.2931: Digital Subscriber Signalling System No. 2 – User-Network Interface (UNI) layer 3 specification for basic call/connection control. ITU, Geneva, 1995. Available: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-Q.2931-199502-I!!PDF-E&type=items

[3] 3GPP TS 36.413: Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access Network (E-UTRAN); S1 Application Protocol (S1AP). 3rd Generation Partnership Project, 2022. Available: https://www.3gpp.org/ftp/Specs/archive/36_series/36.413/36413-g90.zip

[4] IEEE 1588: Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. IEEE, New York, 2019. Available from https://www.techstreet.com/standards/ieee-1588-2019?product_id=2075311

[5] AES47: AES standard for digital audio - Digital input-output interfacing - Transmission of digital audio over asynchronous transfer mode (ATM) networks. Audio Engineering Society, New York, 2006. Available from https://www.aes.org/publications/standards/

[6] AES51: AES standard for digital audio - Digital input-output interfacing - Transmission of ATM cells over Ethernet physical layer. Audio Engineering Society, New York, 2006. Available from https://www.aes.org/publications/standards/

[7] IEC 62379: Common control interface for networked digital audio and video products (6 parts). IEC, Geneva, 2007-2015.

[8] A. Sutton & R. Li (editors): Next Generation Protocols – Market Drivers and Key Scenarios; ETSI White Paper No. 17. ETSI, Sophia Antipolis, France, 2018. Available: https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp17_Next_Generation_Protocols_v01.pdf

[9] ETSI GR NGP 013: Next Generation Protocols (NGP); Flexilink: efficient deterministic packet forwarding in user plane for NGP; Packet formats and forwarding mechanisms. ETSI, Sophia Antipolis, France, 2018. Available: https://www.etsi.org/deliver/etsi_gs/NGP/001_099/013/01.01.01_60/gs_NGP013v010101p.pdf

[10] ETSI GR NIN 001: Non-IP Networking (NIN); Problem statement: networking with TCP/IP in the 2020s. ETSI, Sophia Antipolis, France, 2021. Available: https://www.etsi.org/deliver/etsi_gr/NIN/001_099/001/01.01.01_60/gr_NIN001v010101p.pdf

[11] ETSI GR NIN 002: Non-IP Networking (NIN); Implementing Non-IP networking over 3GPP cellular access. ETSI, Sophia Antipolis, France, 2021. Available: https://www.etsi.org/deliver/etsi_gr/NIN/001_099/002/01.01.01_60/gr_NIN002v010101p.pdf

[12] ETSI GR NIN 003: Non-IP Networking (NIN); Flexilink network model. ETSI, Sophia Antipolis, France, 2021. Available: https://www.etsi.org/deliver/etsi_gr/NIN/001_099/003/01.01.01_60/gr_NIN003v010101p.pdf

[13] ETSI GS NIN 004: Non-IP Networking (NIN); Carriage of Flexilink flows over DECT-2020 New Radio. ETSI, Sophia Antipolis, France. In preparation.

AUTHOR

John Grant MA (Cantab) FAES has been designing digital networks since 1981 when he created local area networking technology which was used in both industrial and commercial environments. Since then he has created products for carrying video and audio over digital networks, including network switching equipment; this has given him an insight into the requirements of audio and other live media, which are very different from those for data traffic. More recently he has been researching how packet networking can meet these requirements as well as avoiding the various problems that have been identified with IP. He is a Fellow of the Audio Engineering Society and chairs their standards subcommittee SC-02 on digital audio as well as the ETSI Industry Strategy Group on Non-IP Networking.