Flexilink scenarios for project REASON

Introduction

Flexilink is the layers 2 and 3 technology being standardised by ETSI ISG NIN, to meet a number of requirements that were identified as being problematic with current-generation technology.

Headline features include: a separate, ultra-low latency, service for live media; very low per-packet overheads for all traffic; and much lower complexity than systems based on the TCP/IP protocol suite, which in turn means lower power and better security. For more details see https://www.etsi.org/technologies/non-ip-networking.

The network architecture is described in ETSI GR NIN 003. There is a clean separation between the forwarding plane, specified in ETSI GS NGP 013 and expected to be implemented in logic (FPGA or ASIC), and the control plane, specified in ETSI GS NIN 005 and expected to be implemented in software. Work to specify implementation over wireless (to be published as ETSI GS NIN 004) is ongoing. Annex A to ETSI GS NGP 013 is an assessment against the KPIs specified in ETSI GS NGP 012.

An important feature is the provision of two separate services, each with its own path through switching elements. The “guaranteed” service is appropriate for live media such as audio and video, and provides very low, and stable, latency, while the “basic” service provides conventional best-effort routing. On wired links, each guaranteed service flow is allocated specific timeslots, and the remaining capacity (including any space within a slot that is allocated but not used) is used for the basic service. A similar regime is used over wireless.

An implementation over gigabit Ethernet PHY has been used for experiments at Birmingham City University and elsewhere; the platform is described here. Porting to other platforms, including switches with more (and faster) network ports, or simple audio or video interfaces, would be straightforward.

The basic technology includes many functions that in IP-based systems are implemented by separate protocols, for instance it distributes time in the same format as PTP and with a similar accuracy; and domain names can be used directly as destination addresses instead of needing to be translated into IP addresses. Thus for new designs it will often be easier to implement Flexilink than to provide all the protocols needed to interface to an IP-based system. Such an implementation would still support applications written for an IP-based environment, and be able to access IP networks via gateways.

This document suggests some areas in which it is likely to require less development effort than achieving equivalent performance with current-generation protocols.

Low latency, high-capacity resilient x-haul

Data rates continue to increase while CPU speeds have plateaued. Flexilink’s forwarding plane is designed to be implemented entirely in hardware, and can therefore be expected to match the physical layer hardware in supporting increasing data rates, which would be difficult for a system (such as IP) that is designed for software processing of packet headers.

Flexilink's guaranteed service natively provides latency of less than 15 μs through a switch with 1 Gb/s links, and proportionately less with faster links, so the benefits of lower-delay links (such as hollow core fibres) will not be lost in buffering delays in switches.

The design for a core network Flexilink switch can be ported from the existing prototype. The FPGA in its platform is a Xilinx Spartan 6 SLX45T; a switch with more or faster ports (but without the audio and video interfaces) could use a larger Xilinx part or for higher performance one from Achronix.

"Guaranteed service" packets are routed based on their time of arrival and switching equipment does not need to read any part of the packet. They can therefore potentially be switched in the optical domain.

End-to-end signalling

A major part of the control plane protocols specified in ETSI GS NIN 005 is concerned with setting flows up, including not only routing within the network but also negotiation between the endpoints. Some information elements in the messages will only be used by the endpoints, for instance those specifying the protocol and data format to be used.

Other information elements are used by all entities through which the flow passes. For instance, the requested data rate can be reduced if necessary to a value that both the network and the endpoints can support. There are also protocols for negotiating a change in an existing flow, for instance to adjust compression parameters to match changes in capacity on the network. Note that this is a formal process, not the empirical reaction to data loss or buffering delays that occurs in current generation networks.

Network edge

Flexilink's flow set-up process allows a client to connect to a server using an identifier for the service, without needing to convert it to any kind of network address or locator. If there are servers available in different locations, the network can choose which to route it to based on whatever metrics are appropriate, for instance to the closest one, or the most lightly loaded, or one that doesn't require the flow to be routed through a congested part of the network. Thus the results of monitoring edge devices can feed directly, and cleanly, into routing decisions.

A client can also request to consume an identified piece of content. The network can connect it to a local or regional server if a copy is cached there, and otherwise to the server that holds the original. The network does not need to know what each cache holds; it can offer the request simultaneously to a number of servers and select the first that accepts it, or wait until several have replied and then choose between them.

In the case of live content, the client can be added to a flow that is already carrying it, without needing to route the flow all the way from the source. The source can specify that it needs to approve any request to join the flow; this approval process is implemented in the control plane, before anything is set up in the forwarding plane.

Security and trustworthiness

The control plane processes support much more authentication and verification than is possible in the packet headers of protocols such as IP, and the flow is not set up in the forwarding plane until the checks have been completed. Processes such as exchanging cryptographic keys and providing the client's log-in details can be included in the flow set-up, reducing the number of end-to-end exchanges needed.

It is also possible for a client to be completely anonymous; the flow is set up as bidirectional, and the server simply sends the response back along it, without needing to be told the client's address.

Packets can only be routed along flows that have been set up in the forwarding plane's routing tables, and a switching element's routing table can only be written by its associated control plane entity. (In the current implementation, that is in the same box, but it could also be a separate unit that controls a subnetwork, as in SDN.)

Control plane entities communicate directly with their neighbours in the physical network, but not with any other entities. Authentication information can be exchanged when the link comes up. Network ports can be configured to show whether they are connected to a trusted entity, and flows can be marked to show whether all the entities they pass through are trusted.

Energy efficiency

The reduction in complexity and elimination of most per-packet processing are expected to result in lower power consumption, though no work has been done yet to quantify this effect.

Network interfaces

Where new equipment that is being developed, for instance to support new kinds of physical layer, needs to include network layer functionality, implementing Flexilink is likely to require less effort than implementing the TCP/IP suite.

Flexilink-aware applications in PCs etc can tunnel Flexilink over UDP or directly over Ethernet. Also, it may be possible to implement the native Flexilink framing specified in ETSI GS NGP 012 clause 8.2 on cards such as the NetFPGA where the MAC layer is implemented in programmable logic.

API in endpoints

Flexilink is a good fit with the Sockets paradigm, with the socket handle identifying the flow. It can be used with IP addresses, though in some cases there will be significant benefit in defining a new address family that implements the address format specified in ETSI GS NIN 005 clause 4.4 and eliminating calls of gethostbyname().

Simple IoT devices

Where large numbers of sensors are transmitting periodic readings, packet payloads will typically be much smaller than an IPv6 header so the overheads will add significantly to the amount of spectrum required if IP is used. They also add to the amount of power drawn from the battery. Current systems therefore use different protocols on the wireless network, with a gateway or agent to connect to wider networks.

With Flexilink, the overheads are so much smaller that the end devices can use the same protocol that is used in the wider area.

Simple devices do not need to implement the whole stack, for instance a camera or microphone can transmit an audio or video stream in a fixed format. (Logic for encapsulation etc can be ported from the existing implementation.) Annex C of ETSI GS NIN 005 specifies a protocol that allows equipment to which it is connected (e.g. switch or radio base station) to discover the data rate, encoding, etc, of the stream, and respond appropriately to requests to receive from it.

Where large numbers of robots occupy the same physical space, for example in a warehouse, power is less of an issue but efficient use of spectrum is still likely to be important, and low latency will be an additional requirement.

Remote surgery

For the majority of robotic surgery, the robot is largely autonomous, and ultra-low latency is not required.

However, where the surgeon is directly controlling an instrument such as a scalpel there needs to be tactile feedback, which needs a round-trip latency of about 1 ms. When converting such a system from local operation (where it is likely that most of the latency budget has already been used up) to remote, the delay across the network must be an order of magnitude less than 1 ms. Flexilink easily achieves that provided physical distances are not too great.

Replacing over-the-air broadcasting

Terrestrial TV broadcasting currently occupies spectrum that could be used to enhance mobile coverage. While much viewing is now on catch-up services, there is still demand for linear television both for live events and for “water-cooler” drama series, so digital networks will need to be able to distribute live content to large numbers of viewers simultaneously.

Current technology requires each viewer to be sent their own copy, and to acknowledge receipt of the data, with retransmission in the case of loss due to buffer overflow. The Flexilink guaranteed service does not suffer from buffer overflow, and can multicast to an unlimited number of viewers.

Distributing software updates

The main function of protocols such as TCP is to limit the rate of transmission to a level the network and the receiving entity can support, and to retransmit packets that are lost due to buffer overflow or transmission error.

When using the Flexilink guaranteed service, it is only transmission errors that need to be considered, and they are likely to be comparatively rare. A large file can be downloaded by setting up a flow from a server that hosts it, transferring the data, and then signalling (via the control plane) whether any part needs to be retransmitted. Usually the whole file will have been received and the client can simply clear the flow down.

Where a large file is to be sent to a very large number of clients, for instance when distributing a software update, the file can be transmitted several times as a multicast, and a client only needs to request retransmission of a packet if all the copies of it were lost.