Nine Tiles

Flexilink implementation for BCU

Introduction

The original prototype implementations of Flexilink on the Aubergine platform and in the associated Windows programs were developed for use within a “walled garden” where security would not be an issue. For the trials in a “real world” environment at BCU we need to comply with Cyber Essentials, which includes protecting Aubergine management requests with some kind of authentication which is not sent in cleartext nor susceptible to replay attacks.

The management protocol (between the “Controller” Windows program and the Aubergines) is based on IEC 62379 which defines four “privilege” levels (“capability” would probably have been a better term) for controlling what functions are available to different users of a media network. Numbered 1 to 4, they are respectively: **listener**, able to set up connections from audio and video sources to their own equipment but not do much else; **operator**, also able to connect media calls within a studio; **supervisor**, also able to connect media calls between studios and set parameters such as the names of media ports; and **maintenance**, able to do anything, including upload new software. Each instance of the Controller program operates at a defined level, set as described later in this document.

The following regime has been implemented in the most recent release of the software.

The Management Information Base (MIB) has a “privilege” marking on each object (separately for Get, Set, and non-volatile Set), such that a management request with a lower privilege can’t access it. Each Aubergine unit has a password associated with each privilege level except listener, and each management request can carry a hash value derived from the password as described under “security issues” below.

Audio and video ports on the Aubergines can also have privilege levels associated with them, and an output cannot be connected or disconnected by a management call from a Controller with a lower privilege level.

Listeners are not able to do anything via management commands that would change any part of the state of the unit. Thus for example they are not able to request any change to the flows connected to an Aubergine’s audio or video ports, nor to properties of a port such as its name. They can, of course, potentially affect the unit’s state via signalling messages, including by setting up or tearing down their management connection. A “virtual sound card” application would be able to connect to an audio or video source, or offer a signal for destinations to connect to, using signalling messages; note that a source can be as picky as it likes with regard to who is allowed to receive from it, including requiring a destination to be authenticated before setting up the forwarding plane connection that will deliver the content.

Network topology

There are two kinds of link between Flexilink network elements. A **physical link** is a cable or fibre linking two ports and carrying a continuous stream of Flexilink frames; the Aubergines implement this over 1Gb/s Ethernet PHY as specified in clause 8.2.2.2 of ETSI GS NGP 013 and Annex A to ETSI GS NIN 005. Physical links are "tightly synchronised" as specified in clause 7 of ETSI GS NIN 005; this is necessary to keep the latency of guaranteed service (e.g. live media) flows low and constant.

A **virtual link** is a connection between two network elements that is carried over another network technology such as the Ethernet MAC layer or UDP. Virtual links are "loosely synchronised": the latency in the underlying network is assumed to be variable, so there is some buffering of guaranteed service flows but the average packet rate over time is the same for both network elements.

An **island** is a set of network elements connected together by physical links, and a **cloud** is a set of islands connected together by virtual links.

Controller program

The Controller provides a "user" interface for the units that make up a Flexilink network, communicating with the Flexilink cloud over IP, and can update the units' software and control and monitor various functionality within the units. It is intended to provide an "engineering" interface to the system, and to show developers of control systems how a back-end interface to the network would function. It is not really intended as an interface for non-technical users, although in a small network it provides an easy way to set up and tear down connections between audio and video equipment.

It communicates with the Flexilink cloud via an external virtual link (see "Network port details" below) over UDP. It was developed in Visual Studio 2019 on Windows 10, and uses an SNMP-like protocol to read and display information from the units' MIBs.

There isn't any formal "installation" process, just double-click on the .exe file in Windows Explorer to run it. However: (1) you will probably have to tell Windows Firewall to allow it to access the network; (2) it should be in a separate directory (or folder) along with the relevant product files, software images, symbol tables, and "debug dump format" files, though these other files aren't necessary unless you want to use the facilities they implement, such as automatic software updates and decoding of crashdumps; and (3) for any privilege level other than "operator" you need to create a shortcut as described under "Privilege and passwords" below.

There are two ways you can connect a PC to the Flexilink network for levels other than maintenance:

(a) if the PC is on an IP network that has a DHCP server, just connect a port on one of the Aubergines into the IP network; or

(b) connect the PC's Ethernet port to an Aubergine; except in the case of the Aubergine's port 3, if there is a connection from anywhere in the Flexilink network into an IP network that has a DHCP server, this will also create a layer 2 tunnel between the PC and the IP network.

Except for “maintenance” level, when the program is run it broadcasts a request on the IP network to which the Aubergine responds.

For “maintenance” level, the PC’s Ethernet port must be connected to port 3 on an Aubergine; this is referred to as being “directly” connected. See also “Security issues” below. The Controller’s request is unicast to the “link local” address 169.254.9.9.

In either case, if it fails to connect, it retries every 7 to 8 sec.

As soon as the virtual link is connected, the Controller sets up a management connection to the unit at the other end of the link and downloads various objects from its MIB. It doesn’t need a password at this stage because all the information is readable at “listener” level. If any of the information shows it has a link to another unit, the Controller also sets up a management connection to that unit; this process continues until it has management connections to all the units it has discovered. If there is, or recently was, another Controller connected to the Flexilink cloud, there may be a delay of up to 20 seconds before the information arrives; this is because the Controller uses information from periodic Status Broadcast messages, and if the unit is already sending out status broadcasts it won't restart the cycle when the new Controller joins it.

If there is a product file in the Controller's directory that matches a unit's product code but the software the unit is running does not match the version numbers in the product file, this indicates the unit’s software is out-of-date. If the Controller is at “maintenance” level, it puts up a dialogue box asking whether to upload new software.

Two windows are present in all cases; for “maintenance” level there is also a third, “console” window. (It is still present for other levels, but minimised.) The “sources” window lists all the media inputs the program has found, and “destinations” lists the media outputs. (Note that the software running in the Aubergine will not tell the Controller about any physical interfaces it doesn't support.) See “Media flows” below for more information.

Except as noted under “Privilege and passwords” below, there is also a “heading” line for each unit; in the destinations window it reports the sources of the audio sync (see 'v' console command below), network time (which is similar to PTP; first number is sync source type, e.g. 02 = local, 8C = from word clock; second is source unit, third is distance from source measured as estimated uncertainty in the time in ns), and the alignment of Flexilink frames (numbers similar to network time; “*m*+*n*” as distance means *m* virtual links + physical links with a combined uncertainty of *n* ns). Currently the “uncertainty” is fixed (in the Aubergine software) at 80 ns for a physical link; I think in practice the performance is better than that, and there are some things that could be done to improve the information, such as exchanging PTP packets before switching to the Flexilink format. In the sources window it simply reports the product name.

The colours for the header lines are: magenta = trying to connect, green = connected and is running the software versions in the product file, blue = connected and is running different software (or product file not found), red = some kind of failure (but usually it switches to magenta almost immediately, as it tries to reconnect).

Colours for media outputs in the “destinations” window depend on their “importance” value (see “Configuring units” below); if connected to a network flow they are: green for 1-63, blue for 64-127, brown for 128-191, red for 192-255. If not connected, they are black for 1-127, magenta for 128-255. If the flow connected to a destination with an importance of 128 or more fails (e.g. a cable is unplugged or the flow is disconnected at the source) the Aubergine immediately attempts

to reconnect; this often results in the flow being rerouted around a fault with no noticeable interruption to the audio.

Media inputs in the “sources” window are displayed in black if not connected, magenta if the flow to which they are connected cannot be identified. Otherwise they show the importance of the flow, using the same colours as for outputs.

Each flow has a 16-byte globally-unique identifier, which is shown against the port that is transmitting it in the “sources” window. Destinations that are connected show the transmitting source if it is in the MIB, the flow identifier otherwise.

If a management connection is lost, the Controller tries to re-establish it, repeating the attempt every 10 seconds if unsuccessful. The information in the “sources” and “destinations” windows for that unit is frozen at the state it had when the connection was lost, and the header line changes to red or magenta. This happens immediately if a connection within a Flexilink island is lost, but there is a delay of several seconds if the lost connection is over an Ethernet or IP network. Right click on the header line (in either the “sources” or the “destinations” window) to tell the Controller to give up trying to connect to the unit and forget the information it was holding about it.

For “maintenance” level, left click on a unit's header line to select it in the console window. The initial display shows the unit's status, including all the MIB objects the program reads and a dump of the messages (green for incoming, black for outgoing, red for errors, yellow background for signalling messages relating to the connection between the unit and the controller program). Typing anything other than ‘@’ switches to the “console” display and sends the character to the unit. The Esc (Escape) key switches between the two displays.

Console commands are only available at “maintenance” level, and only for the unit (if any) to which the Controller is directly connected. They are all single characters; any that aren't recognised (including '?') output a list of commands. There may be other commands that don't appear in the list. The ones most likely to be interesting are:

d - outputs the table of Flexilink flows

D - lists IP addresses etc

i - outputs a table showing internal routing

N – lists “non-volatile” object values that are set at power-on

p - shows what has been detected at each input, including the state of each network port; for ports using the Ethernet MAC it also shows the state of each virtual link connected through the port and a dump of the filtering engine's code

v - shows the state of the process of adjusting the audio clock; “adj” shows how often it is monitored (in milliseconds), larger values indicating it is more stable; the first number after the colon shows what it syncs to as: -1 = nothing, 0 = word clock, 3 = network time, 4-7 = AES3 input, >500 = incoming audio flow on the network

Some of the others can stop various parts of the system working, e.g. stop an input decoding the incoming signal by changing parameters.

Debug interface

Each Aubergine unit includes a debug server using the Simple Control Protocol (SCP, specified in Annex C of ETSI GS NIN 005) on its port 4. If port 4 is connected to another Aubergine unit, and a Controller at "maintenance" level is directly connected to that second unit, a "debug server" line appears in its "sources" window. A left click on that line opens the debug facility in the console window. As with the console interface, '?' produces a list of available commands and Esc switches back to the previous display. Unlike the console interface, commands can be multiple characters and are terminated with the Enter key.

Commands that list stack frames (l, g, and m) need the symbols file (.9t3sym) for the software that crashed to be in the directory from which the Controller was run. The d command needs vm.9t3dbf which specifies how to interpret the debug dump.

Typing '@' in the console window switches to the debug server, or to the next debug server if more than one is open.

Privilege and passwords

The privilege level can be set by including "-privilege *n*" or "-priv *n*" on the command line, where *n* is a single digit in the range 1 to 4 inclusive. Alternatively, "-listener" etc can be used. If not set on the command line, the privilege level defaults to operator.

To set the level, create a shortcut to the Controller.exe file. Right click on it and select "Properties". In the dialogue that appears, under the "Shortcut" tab in the box labelled "Target:", should be a file path ending "Controller.exe". Add the required parameter at the end of the line, separated from it by a space, e.g. "`C:\j\Flexilink\Controller.exe -supervisor`". The right-click menu also has options to pin the shortcut to the task bar or start menu.

For maintenance level, the unit's name is always displayed in the "sources" and "destinations" windows. A left click on either copy selects the unit in the console window. A right click brings up a dialogue box in which the unit's product code can be changed (the version that supports passwords is 3-0-4-0) and any remembered "update all units" state can be removed. Also in the dialogue box is the password to be used to calculate the hash value to be included in management requests; passwords consist of UTF-8 characters, truncated or NUL-padded to 32 octets. However, if the box labelled "Include password in request messages to unit" is unticked the messages will be sent without a password; this is required for units running earlier software and can also be used if the Controller is directly connected to the unit.

Management calls at the "maintenance" level cannot be routed over virtual links. Combined with the requirement for the Controller to be directly connected, this ensures that maintenance level commands cannot pass over an IP network, where they might be intercepted.

For supervisor level, unit names appear in both windows in the same way as for maintenance level. For operator level, the name of any unit for which the user has not supplied a password appears in the "sources" window. For listener level, there are no unit names at all, just the names of audio and video ports.

For the levels other than maintenance, left click on a unit name has no effect. For operator and supervisor levels, a right click on a unit name brings up a dialogue box for the user to type the unit's password for the Controller's privilege level. This is the same as in the maintenance level

dialogue described above, though note that it is only maintenance level Controllers that can be directly connected.

The password dialogue also appears when requesting an action that needs a password if no password has been supplied.

The Controller's record of the password is deleted if the unit clears the call down because of an incorrect hash value. The Controller will then re-establish the call and give the user another opportunity to input the correct password. Note that clearing the management call down will not affect any other aspects of the unit's state.

If no password has been set in the unit for the Controller's level, any password supplied in the request message is ignored and the requested action carried out at listener level. Thus initial setting of a unit's maintenance level password must be done using a Controller that is directly connected to it. Thereafter, passwords can be set or changed by maintenance level Controllers connected to any port within the same island.

Media flows

As with previous versions of the Controller, an audio or video source is selected by left clicking on it in the "sources" window, and left clicking on a destination in the "destinations" window connects it to the selected source. Media flows are multicast, so if several destinations are receiving from the same source the source is only sending one copy. If you select an output that is already receiving from another source, it will be disconnected from the other source; this may cause a click because we simply switch from one digital stream to the other, without doing any kind of fade-across. If there is a signal on an input, an indication such as "44100" for 44.1 kHz audio or "1080p60" for full HD video is shown next to the name; if there is no signal it might still be possible to set up a flow but it might have inappropriate parameters. Similarly, changing the source does not renegotiate the network flow, e.g. if a flow is connected to an input that is receiving 48 kHz audio and the audio source is then changed to 96 kHz nearly half the audio samples from the new source will be lost.

Right clicking on a destination disconnects it from the flow; right clicking on a source clears the flow down, thus disconnecting all the destinations that were receiving it. Any destinations that had an importance level of 128 or above will then immediately reconnect.

The management command to make the connection is sent to the destination, which then uses signalling messages to make the connection. Therefore, the password for the unit containing the destination is required, but not for the unit containing the source. When clearing a flow down by right clicking on its source, the password for the source is required but not for any of the destinations.

A Controller making a connection or clearing one down must have a high enough privilege level, see "Capabilities" below. If not, either the Controller will not send the request, in which case it leaves the selected destination highlighted, or the unit will return an error which will be displayed below the list of ports.

Configuring units

Each destination (audio or video output port) has an "importance" value in the range 1 to 255 associated with it. This was originally provided for use in broadcast studios, where outputs

connected to transmitters have the highest importance values, those that connect to studio outputs the next highest, and so on.

For maintenance and supervisor levels, double clicking on a source or destination port brings up a dialogue that allows the port's name and importance to be changed. This is done using the non-volatile Set request, so the new values are saved in the unit's flash memory, and persist across resets. Note that changing the configuration of a port won't affect any flows that are connected to it at the time the change is made, but will affect new flows that are connected subsequently.

Double clicking on a unit's header line in the sources or destinations window brings up a similar dialogue that allows properties of the unit to be reconfigured. For maintenance level these include the passwords, and for both levels they include the unit's name.

Transparent tunnels

The December 2023 release supports configuring layer 2 "transparent tunnels" between pairs of ports on the network. This is intended to be used where a fibre link has been installed between two IP switches, to allow the link to be used in a Flexilink network. An Aubergine unit is connected to each end of the fibre link, and the IP switch at each end is connected into an Aubergine port instead of directly to the fibre. A transparent tunnel is configured between the two ports that connect to the IP switches; this carries Ethernet packets between the IP switches over the fibre in the same way as if they had been connected directly.

When a port is configured to be one end of a transparent tunnel, all incoming packets are sent down the tunnel without being examined by the Aubergine. If the tunnel is not connected, they are discarded. All packets received from the tunnel, but no other packets, are output on the port.

Configuration of transparent tunnels requires maintenance level. An extra option has been added to the dialogue box that is invoked by right clicking on the unit's name. A tunnel between port P of unit A and port Q of unit B can be configured by: right clicking on the name of unit A; ticking the "Configure a transparent tunnel" box; filling in B as the remote unit number, Q as the remote port number, and P as the local port number; and clicking on "OK". Port P of unit A will then be in "transparent" mode and as soon as its link comes up (or immediately if its link is already up) will attempt to set the tunnel up. If the link to port Q of unit B is up and using the Ethernet MAC, it will enter "transparent" mode. Otherwise (if the link is down or it is using the Flexilink MAC) the call is rejected and unit A will re-try it every 7.5 seconds.

To prevent port Q of unit B beginning the process of joining the IP network during the time (up to 7.5 seconds) between its link coming up and the tunnel being connected, it can also be configured as transparent (right click on unit B, set A as the remote unit number, P as the remote port, Q as the local port). Then the only incoming calls the port will accept are transparent tunnels from unit A for its port P.

Configuring both ends is thus safer and will result in the tunnel being connected immediately the second link comes up.

The configuration is "sticky", i.e. will persist across resets. To undo it (so the port operates normally), use the same dialogue, specifying the remote unit number as zero; in this case the remote port number is ignored. If both ends have been configured, this needs to be repeated for the other end.

Ports that are configured as transparent are listed in the “destinations” window showing the “partner address” that is used in the connection request, e.g. 00 09 05 00 90 A8 99 00 00 00 [B] 09 [Q] (where [B] and [Q] are the unit and port numbers in hex), in a brownish-yellow if the link is down, else green if the tunnel is connected or red if it is not. Ports that are not configured as transparent but have accepted an incoming transparent tunnel connection are also listed but without the partner address.

For LED indications see below; in all cases the lefthand LED is off.

Security issues

Capabilities

As stated in the Introduction, each MIB object has a minimum privilege level for each kind of access. For example, listeners are allowed to read objects that show the state of each flow but not write them, so they can see what flows exist but not connect or disconnect them.

Additionally, each media flow inherits the importance of its destination, or the highest of them if it serves several destinations with different importances., and a Controller with privilege level n can only set up or tear down flows with an importance of less than $64n$. Thus for example operators cannot connect or disconnect flows to destinations with importance 128 and above. If any of a flow’s destinations has an importance of 128 or more, operators cannot disconnect it at source, though they can add and remove destinations with a lower importance.

Passwords

The two main vulnerabilities are the SCP debug interface and the text console messages; both give low-level access to memory and other components, and neither of them has an easy way to provide password protection, so we restrict them to maintenance-level Controllers that are directly-connected.

All other facilities are implemented by management commands, which can carry a hash value along with a 32-bit “hash count”. The hash calculation includes a password, and the unit clears the management call down if the hash value in the request doesn’t match the locally-calculated hash value. We use a separate password for each level so that (for instance) operators are not able to access supervisor-level facilities, even if they reconfigure their Controller program for the higher level. The privilege level is signalled in the FindRoute request that sets the call up.

Each unit has its own set of passwords, and the Controller asks the user for the password separately for each unit. Thus the passwords can be different for each unit if required. Passwords are write-only and can only be written by a maintenance-level controller.

If a password is set but no hash value is included in a management request, it is processed at “listener” level except in the special case of a directly-connected maintenance-level controller – both the SCP interface and the console allow the password to be discovered, so for convenience (and to cover the “forgotten password” case) we also allow a maintenance level Controller that is directly connected to a unit to access the unit’s MIB without quoting the password.

The hash value is calculated according to SHA-3 from the password concatenated with a 32-octet “random string”, and then xor’d with the remainder of the message. Including the content of the message defends against piggy-in-the-middle attacks changing it.

The random string consists of: 12 octets chosen by the Controller and included as a Password IE in the FindRoute request that sets the call up; 16 octets chosen by the Aubergine unit and included as a Password IE in the FindRoute response; and the hash count.

The part chosen by the Aubergine unit includes its local time, which has a resolution of 8ns. Thus if the unit has access to a time reference such as UTC it will never use the same string for two different calls. Even if the clock simply counts time from the unit's last reset, the chance of an attacker setting up a call with the same value as in a previous call, and thus being able to replay requests from that call, is infinitesimal.

The hash count is the number of hash values that have been sent on the call, including the current one. The Aubergine unit checks that it is greater than in the previous hash, and clears the call down if not. This defends against replay of earlier messages from the same call.

Aubergine ports 3 and 4

There might be a possibility that a Controller connected via an Ethernet network to port 3 could appear to be directly connected, so port 3 should only be used for (a) point-to-point wired connection to a local PC running a Controller instance or (b) a physical link to another Aubergine unit (which will use the Flexilink MAC and Flexilink routing).

Also note that when a PC is plugged into any network port on an Aubergine unit except port 3, the unit will attempt to set up a tunnel to an IP network.

The SCP engine is connected to the Flexilink MAC layer logic on port 4. Thus connecting port 4 to an IP network or a device such as a laptop will always be safe because they use the Ethernet MAC layer. If port 4 is used for a physical link to another unit, the debug service is available to a maintenance level Controller directly connected to the latter unit, but not otherwise.

Network ports

The state of each network port is reported in the MIB. For ports 1 to 4 it is also indicated by the LEDs.

The main states that a network port can be in are as follows:

Reset (or "down"): no network connection

Passive: physical layer shows "link up" but no information about what it is connected to

Local: connected to an Ethernet network but no IP layer available

Outwards: connected to an IP network from which it has acquired IP addresses via DHCP

Inwards: connected to a "client" (see below)

Active: Flexilink physical link

The Active state uses the Flexilink MAC layer; all the others listed above use the Ethernet MAC layer. Other states occur transiently, during the process of configuring a Flexilink physical link; this normally completes within a second or two of the link coming up.

If no reply is received to the request to switch to the Flexilink MAC layer, nor to the DHCP request, and Ethernet packets are only seen from one source, the port enters Inwards state, on the assumption that a PC or similar device, which we refer to as the "**client**", has been plugged into it. Except on port 3, if there is a port in the same island that is in Outwards state a basic service flow is connected to it (or to the nearest of them, if there are several) and is identified as a tunnel carrying Ethernet packets. The port uses the same IP address as the Outwards port at the other end of the tunnel; its only use of this address is for communication with the client.

The Aubergine's Ethernet MAC includes a "**filtering engine**" which examines the headers of incoming packets and assigns them to a basic service flow. This is similar to the way SDN works, though it doesn't currently include all the facilities of OpenFlow, and even with this reduced functionality it takes up considerably more FPGA resource than the Flexilink MAC, so it is not available in the version that supports video (which needs significant FPGA resource for the video codec). For an Inwards port, packets addressed to the port's own MAC address, and also multicast packets, are routed to the unit's processor; all others are routed into the tunnel. In the case of multicast packets, they are processed locally and also copied to the tunnel; this means they are liable to be overtaken by unicast packets, but that doesn't seem to cause any problems in practice. Similarly, the filtering engine at the Outwards port routes unicast packets for the client's address into the tunnel, and multicast packets to its processor, which copies them to all tunnels connected to the port. Thus the Flexilink island looks to the rest of the system like a layer 2 switch.

Virtual links

A virtual link carries Flexilink packets over an IT network. There are two kinds:

internal: a connection between two Flexilink islands

external: connecting an application such as the Controller into a Flexilink cloud

Internal links carry both guaranteed service and basic service flows, as well as packets that convey information about the alignment of frames between the islands.

External links are asymmetrical, with the external device being referred to as the "client" and the other as the "switch". They carry basic service flows, and may also carry guaranteed service flows towards the client if the client supports it.

With layer 2 (raw Ethernet) encapsulation there can only ever be one link between a pair of ports, though there may be multiple links between different pairs of ports of the same two units. With UDP there can be several separate links between a pair of ports. External links will normally use UDP, so that each client process running in a computer can have its own link; internal links should use Ethernet encapsulation where possible, to reduce overheads.

See Annex B to ETSI GS NIN 005 for details of the encapsulation. A Wireshark plug-in that decodes management messages encapsulated in UDP is available.

LED indications

The LEDs on ports 1 to 4 indicate their state; the lefthand LED shows the major state and how the righthand LED should be interpreted, as detailed below. Note that if nothing is connected to any

of those ports (also if connected devices are powered off) none of the LEDs will be lit and the only indication that the unit is powered on will be a green LED on the pcb inside the box.

The colours are chosen such that both LEDs are red if the port carries a flow with the highest importance (a signal that is on air in the case of broadcasters' networks); more generally, a red light indicates a cable that should not be unplugged. There are also some error conditions that are indicated by the righthand LED flashing.

There are some indications that appear on all four ports during start-up and if the software has crashed. All the LEDs flash red and green together if the logic is in reset, red and yellow if the soft processor is halted, in which case the SCP interface can be used for diagnosis. When switching on a unit that doesn't have any of ports 1 to 4 connected the LEDs will all flash briefly and then turn off.

States of the lefthand LED are:

Off: standard Ethernet, no other features, no IP address

If the port is in "transparent" state (see "Transparent tunnels" above), the righthand LED is coded as: flashing green = link is down, flashing red = link is up but no tunnel connected, red = tunnel is connected.

Otherwise it is coded as: off = link is down, yellow = connected at 1Gb/s, green = connected at a slower speed.

Green: standard Ethernet, Inwards or Outwards state, no virtual links

The righthand LED is yellow if the port is in Outwards state (see "Network port details" above). If it is in Inwards state, the righthand LED is: green if incoming packets are tunnelled to an Outwards port (in the same unit or elsewhere on the network), in which case it uses the Outwards port's IP address; else flashing yellow if there is no tunnel but it has an IP address; else flashing green. In standard Ethernet states other than Inwards and Outwards, the lefthand LED is off.

Flashing yellow: transitioning to Flexilink physical link

The colour of the righthand LED shows different stages of the process.

Yellow: Flexilink, no guaranteed service flows

The righthand LED is red for a physical link; otherwise the port is in standard Ethernet mode and carries one or more virtual links, none of which carries any guaranteed service flows, and the righthand LED shows the port's state as yellow = Outwards, green = Inwards, off = other.

Red: carrying at least one Flexilink guaranteed service flow

This may apply to either a physical link or an Ethernet port carrying at least one virtual link. The righthand LED shows the highest importance of a guaranteed service flow that traverses the link as: off = 1 to 63, green = 64 to 127, yellow = 128 to 191, red = 192 to 255.