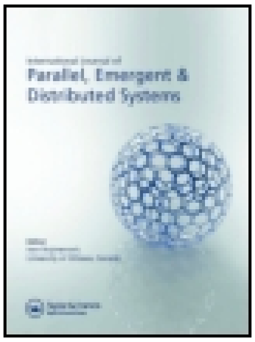


International Journal of Parallel, Emergent and Distributed Systems

ISSN: 1744-5760 (Print) 1744-5779 (Online) Journal homepage: https://www.tandfonline.com/loi/gpaa20

Evaluation of video payload over low latency networks: Flexilink

Rongxuan Ma, Yonghao Wang, Wei Hu & Mahir Payyanil Karalakath

INTERNATIONAL JOURNAL OF PARALLEL, EMERGENT AND DISTRIBUTED SYSTEMS
https://doi.org/10.1080/17445760.2018.1487065

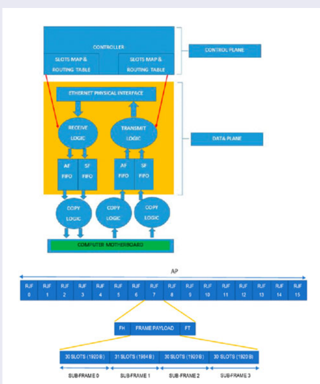
Evaluation of video payload over low latency networks: Flexilink

Rongxuan Ma[a,b,c], Yonghao Wang [a,b], Wei Hu[a,c] and Mahir Payyanil Karalakath[b]

[a]College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, People's Republic of China; [b]DMT Lab, Birmingham City University, Birmingham, UK; [c]Hubei Province Key Laboratory of Intelligent Information Processing and Real-Time Industrial System, Wuhan, People's Republic of China

ABSTRACT

Broadcast organisations need to handle both time deterministic traffic and best effort traffic. The adoption of packet networks allows a single converged network for handling both real-time (RT) media traffic as well as data traffic. Due to the historic pervasive growth of IP network, they are widely used to realise this converged network. However, they do have several drawbacks like big header size, software switching delays and inability to support multiple addressing schemes. To overcome these issues, solutions like traffic engineering, quality of service and over-provisioning are implemented in IP networks but has resulted in higher cost, power and system requirements. Flexilink network has been tested for audio payloads using simulation model and outperforms any existing RT network by providing an exceptional average end-to-end delay of microsecond range and a jitter below the audible threshold for live digital audio streaming. However, it is not tested for video payloads. A review of various networks used RT media distribution is included in this paper. The Flexilink architecture design modification to support the most widely used video formats such as MPEG-TS, MPEG-PS and SDI streams is explained. The average delay for multiple video streams over Flexilink is provided using simulation model. The recommendation for testing Flexilink using hardware is also explained in this paper.



The Flexilink Mac design uses dual-buffer system to handle AF (asynchronous flow) and SF (synchronous flow) separately, with the Frame structure that supports current Ethernet frames.

ARTICLE HISTORY
Received 23 November 2017
Accepted 6 June 2018

KEYWORDS
Low latency; Flexilink; real-time audio video; SDI; MPEG-PS and MPEG-TS

CONTACT Yonghao Wang yonghao.wang@bcu.ac.uk College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430081, People's Republic of China

1. Introduction

Real-time (RT) audio video (AV) transmission is crucial in applications like video conferencing, Robotic Surgery, Traffic monitoring, Drone piloting, online gaming, remote security, driver assistance camera systems and broadcasting. It can also enhance the interactive media features such as fast forwarding, changing channels and Random indexing. The main requirements for RT multimedia transmission are minimal packet loss rate, low delay, low jitter and consistently available bandwidth [1].

Broadcast organisations have been making a continuous effort to deliver RT AV with low latency, least packet loss rate as well as minimal jitter. A steady bandwidth availability is also necessary for RT AV transmission. Currently, they heavily rely on IP networks for RT media delivery due to the low cost of installation, ease of maintenance and the possibility of system control using existing infrastructure. The use of network technologies like MPLS (multi-protocol label switching) and VPNs (virtual private networks) allow IP networks to provide quality of service (QoS).

However, the underlying fact about IP networks is that they were designed to carry bursts of data at random time with primary concern for high accuracy delivery and the delivery time is of second priority, resulting in jitters. In case of video payloads, the speed of delivery is of great importance and any latency can affect the viewing experience.

Media transfer over an IP network suffers end-to-end (E2E) latency due to buffering at the internet nodes as well as at the intranet nodes [2]. The packet header size, as well as the limitation in restricting the latency due to buffering, makes it unfit for applications like live sound, virtual reality and tactile internet. This drawback is mainly due to the connectionless store and forward technology used in the IP networks [3]. In IP networks, several measures like Traffic engineering and QoS mechanisms are used to overcome this but still not capable of providing a complete solution. The existing QoS makes the system complex, thereby increasing the power requirements as well as the cost. Video transmission over the internet is further affected by instability in throughput, delay, and packet loss because of the congestion in network and diverse infrastructure [4]. To overcome these effects, Media streaming systems make use of a large buffer at the receiver end. But this can lead to slow start-up high latency, which is undesirable.

Broadcast production facilities use separate networks for AV transmission. This is because standard Ethernet cannot handle both data and AV traffic efficiently. Hence for professional AV applications with low latency requirement, closed networks with AV-specific layer technology are commonly used. As a result, there is a limited convergence between the two networks resulting in increased costs. So, the usage of a single network that can handle RT AV payloads with low latency along with supporting the non-time deterministic traffic can reduce the investment, cost of maintenance and a lot of human work hours.

There are three ways in traditional RT network transmission:

- Increasing the transport protocol for RT stream base on TCP/IP protocol, like the optimisation of RTP.
- Building different transmission models through the network QoS model.
- In industrial Ethernet, different protocols for different RT streams. Such as high precision clock synchronisation, network topology ordering and resource reservation can enhance the RT performance of the transmission.

J Wu et al. have proposed a GALTON (Goodput-Aware Load distribuTiON) model and a bandwidth aggregation framework that integrates energy-minimised rate adaptation, delay-constrained unequal protection and quality-aware packet distribution (ELBA) to get a low E2E delay for the spectrum limitation of single [5,6].

Flexilink added priority judgement, specific transmission frames and transmission bandwidth planning based on the traditional network framework to achieve high RT transmission.

Flexilink is an RT network solution based on the modified Ethernet concept. It can provide superior quality for professional live multimedia transmission along with a solution for best effort data as well enabling network convergence. The Flexilink network overcomes the problems in IP network with the help of efficient packet scheduling and traffic awareness. To provide QoS to audio and video streams, Flexilink reserves network resources prior to the transmission from the sender to receiver as in circuit-switched networks like asynchronous transfer mode (ATM) and ISDN. This project is to investigate the potential of the Flexilink network in delivering RT video over converged networks.

2. Network classification

The existing networks can be broadly classified into:

2.1. ISDN network

ISDN provides a relatively low bandwidth of up to two Mbps data rate. As a result, ISDN could support only the services having data rate requirement ranging from voice to a medium data rate. Hence, video transmission over ISDN is not popular. To overcome this, CCITT proposed Broadband ISDN (B-ISDN). But the technical complications in B-ISDN concept remained unrectified because a different concept namely ATM was proposed by CCITT as the answer for B-ISDN [7].

2.2. ATM network

Regardless of the traffic inside the network, propagation delay and cell segmentation delay over transmission lines are predetermined and unalterable. Both the cell loss as well as the wrong delivery in transmission are because of the header field error and are regardless of the traffic inside the network. As a result, the key reason for the drop in the performance of ATM network is cell loss and the output queue to output link delay in the ATM nodes. ATM networks need new hardware and software for its implementation and the availability of ATM equipment is limited as well [2].

2.3. Standard Ethernet

Standard Ethernet was originally designed for the best effort traffic transmission [8]. Video over IP networks faces many challenges when compared to that of non-time-critical IP traffic. Professional broadcast environments demand a very stringent QoS requirements such as: Minimum network delay variations, low Packet loss rate, Timing Synchronisation, and adequate Bandwidth

However, the packet header size and the difficulty in restricting the latency within acceptable limits [9] make standard Ethernet unreliable for RT audio/video applications. Hence for professional Video/audio applications with low latency requirement, closed networks with video/audio specific layer technology are commonly used. This will be mentioned as RT Ethernet (RTE) which will be explained in the next section.

2.4. RT Ethernet

According to [10] existing RTEs can be classified into three categories (Figure 1) – on top of TCP/UDP (User Datagram Protocol)/IP, on top of Ethernet, and with modified Ethernet.

2.5. Ethernet audio video bridging

The Ethernet audio video bridging (AVB) project was recognised under the IEEE802.1 [12]. It was defined by the AVnu Alliance [13]. AVB includes a set of specifications to support Audio/Video streams over Ethernet, which is low-latency time-sensitive traffic. By using various mechanisms AVB standard

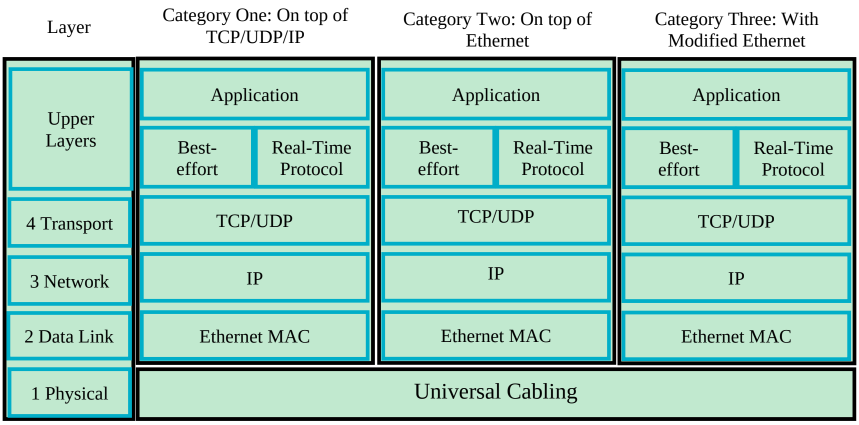


Figure 1. RT Ethernet protocol classification [11].

provides low latency time synchronised streaming services. To provide these services, AVB makes use of 802 networks specified by the AVB task group [3]. AVB promotes prioritisation as well as over-provisioning [14].

2.6. Dante networks

Dante was created by Audinate. It is a multi-channel digital media networking technology for uncompressed audio with near-zero latency and synchronisation [15]. Dante is designed to make use of gigabit Ethernet. It can transmit 512 channels in each direction in a single 1 Gb link. The latency can be set as low as .25 millisecond. A link in Dante network is capable of concurrently carrying audio having non-identical bit depths as well as sample rates. Dante systems are scalable from a single link between a computer and audio console to large networks with several hundred audio channels. This is made possible by using logical routes rather than physical point-to-point connections.

3. Fexilink implementation

3.1. Simulation model

The Flexilink simulation model is implemented using SimEvents platform in the Simulink. The SimEvents platform is created for the discrete event simulation. Simulation model consists of a node A and node B connected by a cable of length 100 m. The Simulation is generated by modifying a simulation model which was originally designed for frame-based uncompressed audio payload. The SF source generates MPEG-TS packets of fixed 188 bytes size with a packet generation frequency ranging between 3325 and 5320 packets per second. While an SF source generates MPEG-PS packets of fixed 2000 bytes size with a packet generation frequency of 500 packets per second. Since the model explained here is based on the previous Flexilink design, the consideration of 8 K APs is adopted. The average E2E obtained from the model in case of multiple streams in MPEG-PS and that in MPEG-TS are given in the Evaluation section.

Any evaluation of Flexilink in future using simulation model should be based on the new Flexilink design (discussed in Section 4.1) which is having an allocation period of 999.68 microseconds at the frame layer instead of 124.96 microseconds in the previous design.

3.2. Physical link

The proposed physical link comprises of Black Magic HDMI (high-definition multi-media interface) to SDI converter, Aubergine switches, media interfaces, Black Magic SDI to HDMI converter, Laptop acting as a video source. At the transmitter end, a separate laptop or the same laptop acting as the video source is used to control the Aubergine switches with the help of a controller software. A laptop running applications like LOLA or VLC (video LAN client) is being used to monitor and record video transmitted from video camera to monitors.

4. Evaluation

The evaluation of the Flexilink network is carried out using a simulation model as well as using a physical link. By using both the simulation model and physical link the goal is to find the end to end delay for video payloads in a single hop Flexilink network. This is because, in case of RT video, the most challenging problem is the E2E delay of the video frame [16]. The E2E delay has a great effect on viewers' broadcast latency [17]. So, QoS needs to be defined for synchronous flow like audio and video streams. QoS corresponds to the way of sending data across a network so that the data reaches its destination reliably [1].

4.1. Testing with the simevents Flexilink model

Node A represents the sender and node B represents the receiver. These two nodes are connected by a cable block of 100 m length. A one gigabit Ethernet link is used to connect node A and node B. The transmission delay, propagation delay and the E2E delay is given below

- Transmission delay = sample size/link bandwidth. For MPEG-TS, Transmission delay = 188 byte/1 Gbps = $1.504*10^{-6}$.
- Propagation delay: depends on the cable length and material of cable. It is calculated by using the formula: (cable length)/two-thirds of the speed of light. The propagation delay is same for both MPEG-TS and MPEG-PS.

$$\text{Propagation delay} = \frac{100\text{m}}{2 \times 10^8 \text{m/s}} = 5 \times 10^{-7}.$$

o E2E delay = buffering delay at the sender + buffering delay at the receiver + transmission delay + propagation delay [11].

The calculations performed to find the number of slots, AP and packet frequency is explained in the calculation section of Appendices on page 82. The E2E delay is calculated for MPEG-PS and MPEG-TS for multiple streams (Tables 1 and 2). Unlike Priority-based Ethernet or standard Ethernet, there is a trace amount of increase in the latency with the increase in the number of streams. Since most production facility networks need to handle AF traffic as well, a background AF traffic consuming 60% of the total bandwidth is utilised in the testing to replicate the real-life network conditions.

Table 1. Average E2E in case of multiple MPEG-TS streams.

No. of TS streams	Average E2E delay (millisecond)
1	.74
2	.74
10	.75
20	.765
40	.795
60	.825
80	.855
100	.885

Table 2. Average E2E in case of multiple MPEG-PS streams.

NO. of PS streams	Average E2E delay
1	.847 microsecond
2	.101 millisecond
10	.228 millisecond
20	.388 millisecond
40	.708 millisecond
60	.0010 second
80	.0013 second
100	.0017 second

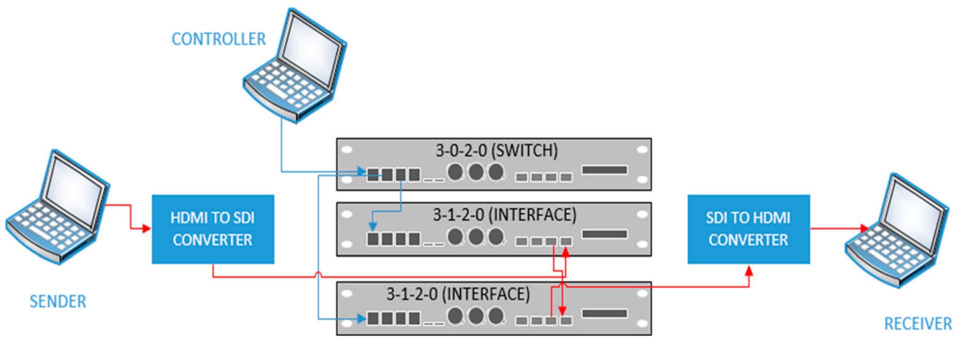


Figure 2. Proposed test network.

4.2. *Testing with Flexilink physical link*

The software running on the controller laptop detects the connected Flexilink switches using information which it gets from the Aubergines. The top box acts as the network and the other two act as interfaces which plug into the network (Figure 2). This is so because the Flexilink logic cannot handle both decoding of the IP headers and the video coding at once and hence the task of handling the network is entrusted on one Flexilink switch and a separate switch is used for AV input–output purposes.

This setup gives a user PC access to the Internet in the [PC ↔ Flexilink network ↔ IP network] scenario. When a link comes up, the switch tries to identify what device had made a connection to it by sending out AES51 and dynamic host configuration protocol (DHCP) requests

If switch gets a reply to the AES51 request, it switches to the RJF format and enters the Active state when it has synced up to the neighbour).

- If it gets a reply to the DHCP request, it acquires an IP address and enters Outwards state, showing it is connected to an IP network.
- Otherwise, if the switch finds all the Ethernet packets it receives, the switch assumes that it is connected to a device like a PC and enters Inwards state.
- Otherwise, it enters Passive state.

When the switch enters the Inwards state, it requests an IT flow connection (termed as a tunnel) to a port in Outwards state. Once the request is successful, everything arriving at the Inwards port with a MAC address different to that of the Inwards port as the (Ethernet) destination is sent down the tunnel; and everything arriving at the Outwards port with the PC's MAC address (or a multicast address) as its destination is sent down the tunnel in the other direction. The Flexilink switches do not care about the contents of the Ethernet packets or they don't run ARP or anything. All they care about is whether a

DHCP server is present in the network at the other end of the tunnel, on the assumption that the PC will need one.

4.2.1. Flexilink controller

A Flexilink controller is the PC application that communicates with Aubergines. It is a prime part of the system. Its main function is to set up connections between the sending and receiving nodes. Apart from that, what Controller does is system monitoring. Each switch looks after the routing themselves and hence the Flexilink controller is not an equivalent of an SDN (software-defined network) controller. The decision regarding the routes and slots allocations are made by the switches themselves.

5. Recommendations for Flexilink design in case of video payload

5.1. Modified frame layer (for compressed SDI video payload)

The standard Aubergines video interface only support SDI video streams. It does not currently support MPEG-TS or PS. SDI is having 3 Gbps data rate in the 1080p formats and 1.5 Gbps in the 1080i formats. The Aubergines' network ports are 1 Gbps; so compression is needed and Aubergine uses Wavelet compression. The reason for choosing wavelet compression over other compression techniques (like DCT in MPEG) is because the main architect of Flexilink had a software implementation of the wavelet transform that he could easily re-implement in logic and which would fit in the Spartan6 FPGA.

The frame layer structure of Flexilink implementation over a gigabit Ethernet link for compressed SDI video stream is given (Figure 3).

Unlike the previous Flexilink design which had an AP of 124.96 microsecond at the frame layer, the new design has an AP of 999.68 microseconds and consist of 1936 contiguous slots. The advantage of using longer slots is that we can have a longer AP without increasing the size of the tables in which the allocations are stored. Instead of an AP with 2 RJF in the previous design, the new design has an AP with 16 RJF. Each RJF contains four sub-frames with subframe 1 having 31 slots (1984 bytes) and the rest of the sub-frames have 30 slots (1920 bytes). A total of 64 sub-frames (16 RJF* 4 subframe) are stored in the routing table. These sub-frames are the parameter of primary interest as it is the amount that fits in the forwarding buffer. The buffer is 64-bit wide in Aubergine and hence read/write 64 bits at a time.

5.2. Modified frame layer (for uncompressed SDI video payload)

To transmit an uncompressed SDI video stream over Flexilink, 10 Gbps ports are needed. This 10 Gbps port can also be used to transmit multiple compressed video streams as well. A single AP will

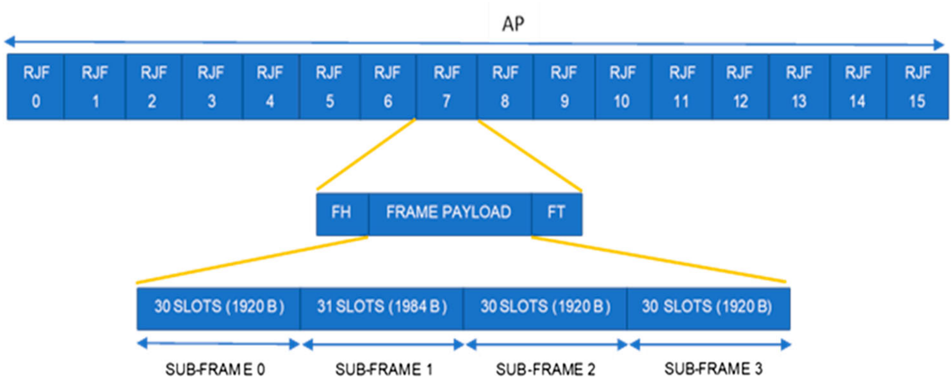


Figure 3. Frame layer structure of Flexilink [18].

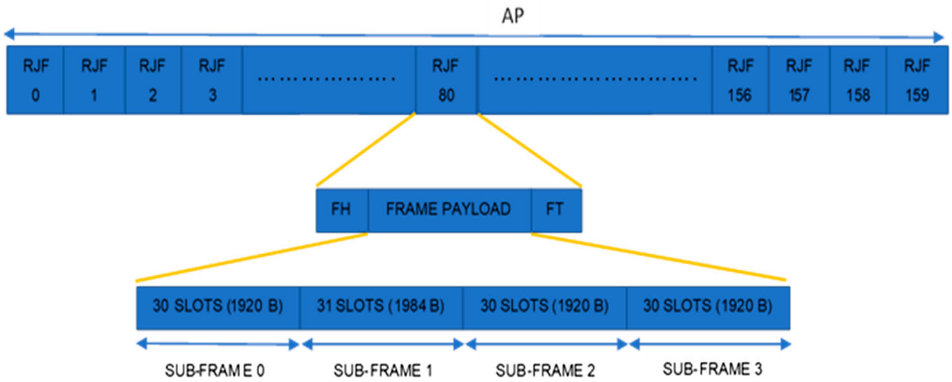


Figure 4. Frame layer structure of Flexilink implementation over a 10-gigabit Ethernet link.

have 160 frames instead of 16 which enables sending SDI uncompressed. However, those video streams in the 4 K format are not supported by Flexilink since they have a data rate exceeding 12 Gbps. The frame layer structure of Flexilink implementation over a 10-gigabit Ethernet link for uncompressed/compressed SDI video stream is shown in Figure 4.

5.3. Modified frame format

The total payload size is 7744 bytes instead of the 7785 bytes in the previous design (Figure 5). These 7744 Bytes payloads consist of 121 slots and 40 trailing bytes. The IFG is set between 12 and 16 bytes, nominally 14 bytes so that the total frame length is 7810 bytes but adjusted by a byte or two to keep frames in sync on links that don't have the same clock.

5.4. Modified AV packet format

The main design concept of Flexilink was to support AV packets with flexible sizes ranging in steps of 2 bytes up to 4 k bytes. The advantage of this concept was that it facilitated the effective forwarding of smaller packets mainly the audio samples as well as avoiding the problem of fragmentation associated with bigger packets. However, the AV packet size should be restricted because of the following reasons:

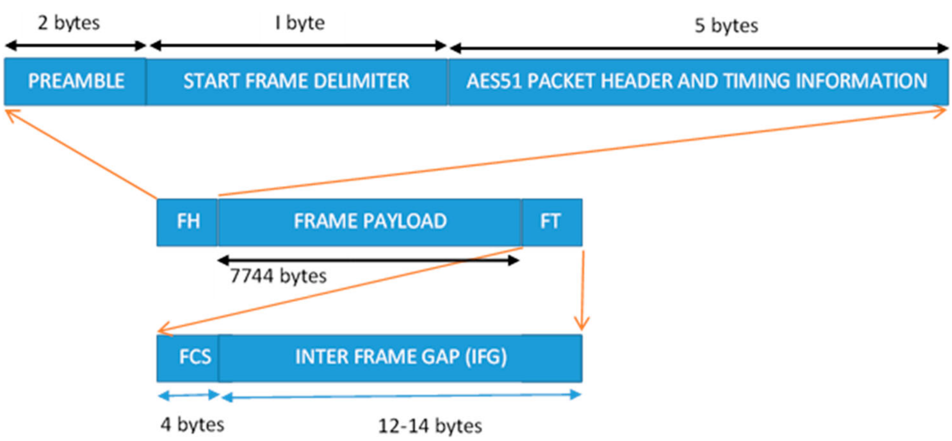


Figure 5. Frame structure on an Ethernet link having 1 Gbps bandwidth [18].

(1) Small packet size prevents the hindering of major portions in an allocation period, thereby reducing the latency faced by best effort traffic as well as avoiding the need of larger buffers.
(2) Small packet size helps in the even spreading of slot allocation for flows [18].
(3) The size of the internal data paths within a switch should be taken into consideration while restricting the packet/slot size such that the packet/slot size should be no less than the size of the switch data paths. The data paths in the switch should have enough size to support the throughput requirements of present-day networks. Based on the experiments performed using a prototype model, the slot size is fixed to be 64 bytes (1-byte header and 63-byte payload) (Figure 6).

The reason for not choosing a bigger slot size, say 128-byte slot is that:

- It would require 7 bits to represent a number in the range of 0–127. The header will need two more bits to accommodate the parity and flag field resulting in 9 bits. So, a 2-byte header is needed in case of a 128-byte slot resulting in a 126-byte payload size which is like using two 64-byte slots.
- It will have fewer slots per subframe resulting in less flexibility in routing.
- It also increases latency since more data need to be inserted into a packet.

Even though this 64-byte slot size is much like the ATM cell size of 53 bytes, the major difference between the two is in the utilisation of unused slot portion. In Flexilink, the unused portion within a slot is utilised to accommodate best effort traffic packets while in case of ATM cells they remain vacant.

Unlike in the previous version, the SF header length is not variable and is of a single byte size (Figure 7).

Unlike in the previous Flexilink design, the new design no longer uses a 3-bit CRC. Instead, a 1 bit odd parity is used for error checking and the 2 bits saved by replacing CRC with odd parity is added to the 4 bits used to encode the length information (*n* field) thus making it a 6-bit field. This expansion to the *n*-field is crucial because the current payload size is up to 63 bytes long and 6 bits are needed to represent a number in the range 0–63. The replacement of CRC with odd parity won't be having much effect since the frame header consists of frame check sequence (FCS), a 4-byte CRC capable of error detection and will warn if the link is unreliable. The error checking is not included in the frame header because when transmitting the frame header, the information regarding the payload is still unknown (unless a whole frame is buffered, which adds about 60 microseconds of latency).

The parameter of importance to the network is n and not f. The one and only thing n tells is: where the background (IT) data restarts. The AV routing forwards whole slots, though on the output the bytes that are not part of an AV packet are replaced by background data.

Unallocated slots or vacant slots are coded with length $= 0$ and $f = 1$. Any slot having a nonzero value in the header length field is occupied. The f bit can be used only by the end-systems and is not used for routing purposes. $f = 0$ for the last fragment of the higher-layer data unit and $f = 1$ if the higher-layer data unit (message) continues in the flow's next slot (Table 3). As a result, the higher layer packet/data unit/message is unaffected by the insertion/addition and deletion/dropping of empty

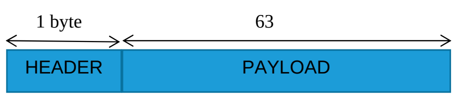


Figure 6. Revised slot/packet structure [18].

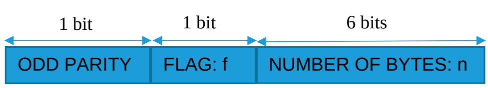


Figure 7. Modified header format of SF packet [18].

Table 3. length information and flag values in an AV packet header [18].

Length information, *n*	Flag, *f*	Higher layer data unit status
0	1	Null packet/null slot
Not zero	1	Continues in the next slot
Not zero	0	Ends in that very slot
0	0	Ends in that very slot

slots to flow. At present, it is not possible to insert/or drop AV packets. However, inserting or dropping an AP is possible once the sending end and receiving end agree on precise time of change. The above told SF packet header format is used across the entire Flexilink network and does not undergo any change as result of packet forwarding [18].

5.5. Modified IT/BE traffic packet format

The IT packet header consists of the following information

- Length information (*n*): represent the number of payload bytes.
- Flow label: used to identify the flow. It is the address to be read in the routing table.
- Three-bit CRC is used to cover each length information as well as flow label.

The flow label is local to the respective packet transmission link and is changed with each packet forwarding while the payload length remains unchanged, unless that if the packet is forwarded between links having different header formats then the entire header may need to be replaced. In case of a frame or packet-based network like the internet, a maximum transmission unit (MTU) is used by the TCP in order decide the maximum packet size or frame size that can be sent across the network [19]. The Flexilink prototype implementation had an MTU of 1788 bytes (Figure 8).

IT packets are encapsulated in Ethernet or UDP headers to be carried over virtual links (Figure 9). The AV packets are carried over a virtual link by encapsulating them in an IT packet (Figure 10). The AV packet flow is recognised by the label in the IT packet header.

In the previous Flexilink design to transport audio payload, the idea was that slots would be smaller so that there is minimal vacant space if a packet just carried a single stereo (or even mono) audio sample. For instance, in case of Audio, the Aubergine software needs 9 bytes per sample, so to make more efficient use of the slots we put five or six samples in a packet. For larger packets, a group of adjacent slots could be allocated. However, this poses the following problems while implementing the AV packet forwarding:

(1) The AV forwarding buffer in a switch needs to be quite wide (total bits per second for all the ports divided by the clock rate of the memory interface). In the Aubergine implementation, the buffer is 64 bits wide; so it writes 64 bits (8 bytes, one-eighth of a slot) at a time. The design allows the buffer memory to be up to 512 bits wide, enabling writing one slot at a time.
(2) For a large packet, many adjacent slots must be allocated, and the start must be within a 'window' of about 1800 bytes and the packet must not cross a frame boundary. It is unable to guarantee

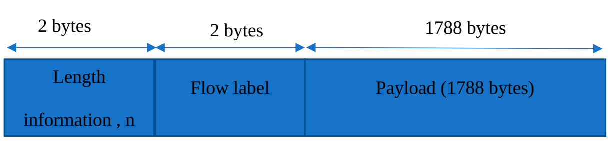


Figure 8. IT packet [18].

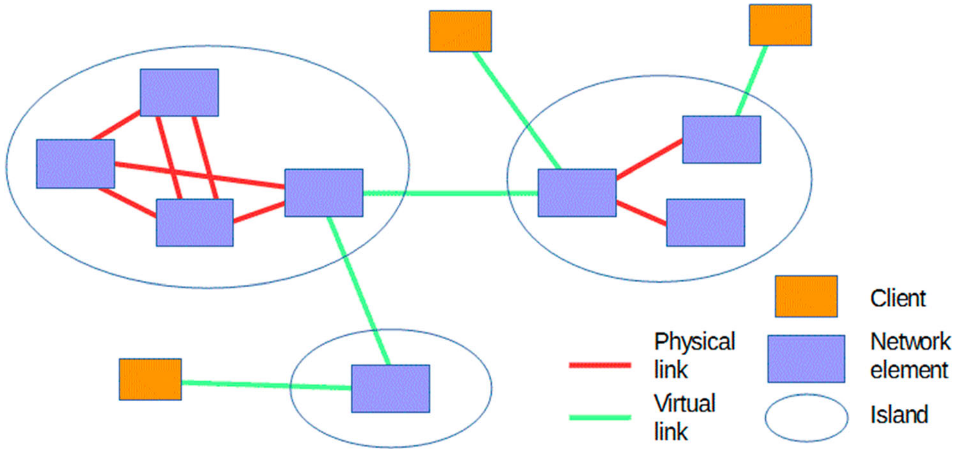


Figure 9. Physical and virtual links in Flexilink network [18].

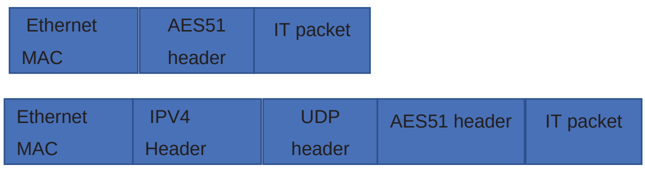


Figure 10. IT packet encapsulation in Ethernet (top) and IP network (bottom).

routing a 2K-byte packet at all, and if a few flows with frequent small packets are routed then the space will be too fragmented to be able to route a flow that carries large packets.

(3) In the original design, the sending end will inform the receiving end which all slots contained the AV packet headers. In case of any change of adding or deleting an allocation, both ends had to agree on when exactly the changes would take place. By using fixed-size slots all of which contain an AV packet header, there is no more the need of agreement on the allocation changes between the sender and receiver ends.

The above factors caused the original Flexilink design to evolve from a flexible slot size to a fixed 64-byte slots containing a 1-byte AV packet/slot header.

5.6. *For MPEG-TS/PS*

MPEG-PS/TS output from a MPEG encoder can be fetched to the Flexilink switch via the network port alone since the Media port support SDI video stream alone. So, the MPEG video stream is fed to any of the network ports ranging from 1 to 4 or 10 or 11. But the problem with feeding AV stream to any network port is that Aubergine does not know it is a AV stream and will treat it as a best effort traffic and hence no Flexilink QoS features will be provided. To overcome this, we can either make use of:

5.6.1. *IP tunnelling for AV packets*

In IP networks, MPEG data will be carried in RTP (real-time protocol) packets (because live streams are sent on IP networks using RTP). To send a stream (MPEG or any other), the address and port number should be specified in the packet headers. This information regarding the address and port numbers are carried in SIP (Session Initiation Protocol) and other control plane messages. It is not carried with the media in the RTP packets. One way to find that information is by using SIP. The main purpose of control protocols like RTSP (real-time streaming protocol), H.245 and SIP with SDP is to:

- Provide information regarding media stream availability
- Create physical or virtual connections
- Exchange the transmitter and receiver capabilities
- Govern an ongoing session [18].

On IP networks, SIP packets use port number 5060 and the MPEG data in RTP packets use port 5004. By intercepting the SIP message, it is possible to find out what data rate the stream needs and hence we can get set up an AV flow as a tunnel to carry the flow they describe and route the media packets (RTP) for that flow down it.

The Ethernet header consists of 14 bytes plus 4 bytes for each virtual local area network (VLAN) tag (now we consider a single VLAN tag) and another 6 bytes for subnetwork access protocol (SNAP) header (sender node adds the SNAP header to the packets so that the destination nodes can forward the incoming frames to the specific device driver capable of recognising the given protocols). Then in total, Ethernet header comprises 24 bytes.

The original IP payload will consist of UDP and RTP headers in addition to the media data. This is because for media applications, we generally run RTP on top of the UDP. So, the actual size of the original IP payload will be 216 bytes, which is the sum of UDP header (8 bytes), RTP header (20 bytes) and MPEG-TS packet (188 bytes). The incoming MPEG-TS stream over IP will have a total size of 260 bytes (considering the IP header as 20 bytes) (Figure 11). In case of MPEG-PS consisting of 2000 bytes packets, the original IP payload will be of size 2028 bytes (Figure 12).

There are three options to handle an MPEG stream over an IP tunnel. They are:

- Option 1 (transmit the IP packets in a Flexilink IT flow)

The IP packets containing the MPEG stream is transmitted in a Flexilink IT flow. For this, an IT packet header is added on the front of each packet. The main drawback of this option is that it does not give any QoS guarantees. The MPEG stream transmitted over IP is treated as best effort traffic. So, this option is no better than the IP network. In option 1, we do not have to slice the IP packet into fragments of 63 bytes each since the MTU is 1788 bytes.

- Option 2 (transmit the IP packets in a Flexilink AV flow)

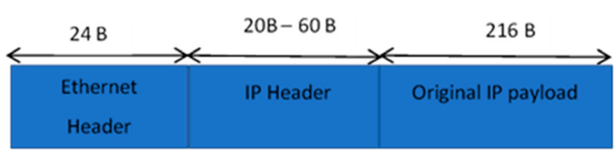


Figure 11. Incoming MPEG-TS stream in IP payload.

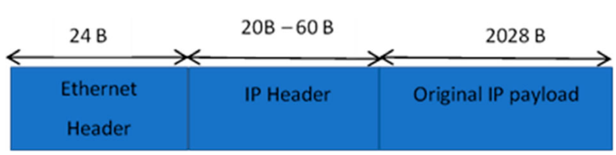


Figure 12. Incoming MPEG-PS stream in IP payload.

Figure 13. Packet format in case of using virtual media card.

This option can be chosen if the destination also uses IP. It helps avoid having to do any processing on the packets since the destination can process IP packets. Since an AV packet can only have up to 63 bytes of payload, the IP packet must be chopped into fragments of up to 63 bytes each, with $f = 0$ in the last one and $f = 1$ in all the others. In that case, the Ethernet header, the IP header and a portion of payload (19-byte size) will be in the first AV packet and the rest of the payload in the adjacent AV packets.

The original incoming MPEG stream is fed to the network port of an interface and the MPEG stream passed over an IP tunnel is extracted through the network port of the interface. As a result, it is possible to provide QoS features to the MPEG stream fed via the network port of Aubergine. In this case, the number of slots required to transmit the AV packets needs to be found out and this can be done by reading the SIP/SDP (Session description protocol) information. An AV packet header needs to be inserted at the front of each 63byte AV payload. All the AV header will have length $n = 63$ and $f = 1$ (representing information is continued in next packet) except for the last one, which has a length equivalent to the number of bytes left and $f = 0$. There will be more AV packets generated because the IP packet must be chopped into fragments of up to 63 bytes each, with $f = 0$ in the last one and $f = 1$ in all the others.

- Option 3 (ripping the MPEG stream over IP and fetching via media port)

The original incoming MPEG stream is fed to the network port of an interface and the MPEG stream passed over an IP tunnel is extracted through the media port of the interface. This option can be chosen if we need to specifically take the AV output via media port. In this method, QoS features are Guaranteed. Unlike in option 2, all the headers except the Flexilink AV header will be ripped out so that we will probably need fewer slots. It consists of unpacking the media data from the RTP packets and send it on a Flexilink AV flow in the same format we use in the Aubergines that are configured as interfaces. Same as in option 2, the number of slots required to transmit the AV packets is found out by reading the SIP/SDP information. An AV packet header is inserted at the front of each 63 bytes AV payload. All the AV payload will have length $n = 63$ and $f = 1$ except for the last one, which has a length equivalent to the number of bytes left and $f = 0$.

Irrespective of whichever option we use to transmit the MPEG stream over Flexilink, we will have to extract the audio/video/etc from the RTP packets and it should be done on the way in so that carrying all the Ethernet/IP/UDP/RTP headers across the Flexilink network can be avoided.

5.6.2. Implementing virtual media card

If a laptop is being used as the sender and receiver nodes, then a better solution for transmitting MPEG stream over Flexilink will be to implement an Ethernet driver supporting Flexilink protocols and avoids the requirement of Aubergines to implement all the RTP/UDP/IP protocols. It will be shown as secondary digital media interfaces such as a virtual sound card or virtual video card. The packet format on the connection between the PC and the Flexilink network is shown (Figure 13).

6. Conclusion and future work

Based on a text-based research exercise, a new Flexilink architecture design with fixed slot size is proposed to provide low latency for video payloads. The video formats like MPEG-PS, SDI and MPEG-TS were taken into consideration, which is commonly used video formats for media distribution within a broadcast facility and media transmission to end-users, respectively. A brief description regarding the latest file format namely MXF (which is primarily used in live broadcast environments) is also included in the report.

The testing of Flexilink using a simulation model based on the previous Fexilink model was conducted and the attempt to evaluate the Flexilink performance across the existing fibre link in BCU was unsuccessful due to the time barrier in getting the equipment on time. However, Flexilink performance is evaluated using the simulation model based on the previous Flexilink design. The latency so measured for a single MPEG-PS AND MPEG-TS are 0.847 microseconds and 0.74 milliseconds, respectively. Future work should focus on testing video over Flexilink using a simulation model based on the new Flexilink design and performing latency tests for video over Flexilink using the proposed hardware test setup. And a combination with cloud computing can also save energy. K Gai proposed a low-cost model for cloud computing called Energy-Aware Heterogeneous Cloud Management (EA-HCM) model [20]. The main algorithm Heterogeneous Task Assignment Algorithm (HTA2) can produce optimal task assignments at a high success rate.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by National Natural Science Foundation of China (61472293) and Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, China (znxx2018MS03 and znxx2018QN07).

ORCID

Yonghao Wang http://orcid.org/0000-0002-4924-2508

References

[1] Bouillot N, Cohen E, Cooperstock JR, et al. AES white paper: best practices in network audio. J Audio Eng Soc. 2009;57(9):729–741.
[2] Wang Y, Grant J, Foss J. Flexilink: a unified low latency network architecture for multichannel live audio. San Francisco (CA): Audio Engineering Society; 2012.
[3] Nine Tiles. Nine Tiles Flexilink. 2007. [cited 2017 Apr 16]. Available from: http://www.ninetiles.com/FlexilinkIntro.html
[4] Liang Y, Girod B. Network-adaptive low-latency video communication over best-effort networks. IEEE Trans Circuits Syst Video Technol. 2006;16(1):72–81.
[5] Wu J, Yuen C, Cheng B, et al. Goodput-aware load distribution for real-time traffic over multipath networks. IEEE Trans Parallel Distr Syst. 2015;26(8):2286–2299.
[6] Wu J, Cheng B, Wang M, et al. Energy-efficient bandwidth aggregation for delay-constrained video over heterogeneous wireless networks. IEEE J Sel Areas Commun. 2017;35(1):30–49.
[7] Derbyshire A, Rajh K. Video transmission over broadband networks. 1996. [cited 2017 Jun]. Available from: https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/arad/report.html
[8] Brunner T. Extending video networks with IP and ethernet. SMPTE 143rd technical conference and exhibition; 2001.
[9] Liang Y-J, Farber N, Girod B. Adaptive playout scheduling and loss concealment for voice communication over IP networks. IEEE Trans Multimedia. 2003;5(4):532–543.
[10] Schumacher M, Jasperneite J, Weber K. A new approach for increasing the performance of the industrial ethernet system PROFINET. Dresden: Proc. 7th IEEE Int. Workshop Factory Commun. Syst; 2008; WFCS 2008.
[11] Song Y, Wang Y, Bull P, et al. Performance evaluation of a new flexible time division multiplexing protocol on mixed traffic types. 31st IEEE international conference on advanced information networking and application (AINA); 2017; Taipei.
[12] Kojima T, John S, Chen J-R, et al. A practical approach to IP live production. SMPTE Motion Imaging J. 2015;124(2):29–40.
[13] Sommer MO. ATM & MPEG-2: integrating digital video into broadband networks. 1st ed. Upper Saddle River (NJ): Prentice Hall PTR; 1998.
[14] Steinbach T, Lim H-T, Korf F, et al. Tomorrows in-car interconnectA competitive evaluation of IEEE 802.1 AVB and time-triggered ethernet. Proceedings of the 76th IEEE Vehicular Technology Conference (VTC2012-Fall); 2012; Berlin.
[15] Audinate. Three pillars of audio networks. Audinate; 2015. [cited 2017 Jun]. Available from: http://go.audinate.com/

hubfs/campaign/3pillars/three-pillars-audio-networking-whitepaper-audinate.pdf?hsLang = es&t = 1529040206419.

[16] Ma X, Zhang C, Liu J. Live broadcast with community interactions: bottlenecks and optimizations. IEEE Trans Multimedia. 2017;19(6):1184–1194.

[17] Wu J, Yang J, Wu X, et al. A low latency scheduling approach for high definition video streaming over heterogenous wireless networks. Atlanta (GA): IEEE Conference Publications; 2013.

[18] Ninetiles. Flexilink. 2016. [cited 2017 Jun]. Available from: http://www.ninetiles.com/Flexilink_details.html

[19] Wenger S. H.264/AVC over IP. IEEE Trans Circuits Syst Video Technol. 2003;13(7):645–656.

[20] Gai K, Qiu M, Zhao H. Energy-aware task assignment for mobile cyber-enabled applications in heterogeneous cloud computing. J Parallel Distr Comput. 2017;111.