Performance Evaluation of a New Flexible Time Division Multiplexing Protocol on Mixed Traffic Types

Yangyang Song, Yonghao Wang and Peter Bull
School of Computing and Digital Technology,
Birmingham City University, UK
Email: {yangyang.song, yonghao.wang, peter.bull}@bcu.ac.uk

Joshua D. Reiss
Centre for Digital Music,
Queen Mary University of London, UK
Email: joshua.reiss@qmul.ac.uk

***Abstract*—The broadcasting industry has recently begun to adopt statistical multiplexing based network platform in their workflow to support professional live audio/video (AV) transmission instead of the Time Division Multiplexing (TDM) based system. These audio-over-packet switched systems require a carefully designed and managed network to ensure key quality measures of the real-time (RT) media, such as low jitter and low latency. Often the best effort traffic or different types of media are still physically or logically segregated from these dedicated systems, or require large redundant links. The proposed Flexilink architecture is an alternative that combines both circuit switched and best effort features. However, there is no research evaluation that shows the actual performance of this proposed architecture. In this paper, we give a simulation based study and critical evaluation of the performance of the Flexilink network. The simulation results show that Flexilink has a better and more stable RT performance when compared with both Ethernet and priority queueing networks, especially when given a burst of traffic and/or multiple RT traffic sources. In addition, Unlike other networking protocols, jitter in Flexilink is below the audible threshold.**

I. Introduction

The broadcasting industry has begun to adopt statistical multiplexing based network platforms in their workflow to support professional live AV transmission instead of a TDM based system [1], such as the 4-wire and the BBC ViLoR (Virtualised Local Radio) project [2]. Still, these audio over packet switched systems require a carefully designed and managed network [3] to ensure key quality measures of the RT media such as low jitter, low latency, and clock synchronisation.

Often the best effort traffic network and the different types of media network, such as audio and video are physically or logically segregated with dedicated network systems. This is because in the professional AV production environment, the business critical data traffic is the RT periodic data [4] such as audio and video with extremely rigorous requirements of jitter, clocking, and in many cases, the low latency. Therefore, the common experiences of quality issues of the Voice-over Internet Protocol (VoIP) over an open uncontrolled network [5] is not acceptable in such a professional environment. For the same reason, within a professional AV production and broadcasting industry, the network convergence, which means transmitting different types of traffic in the same physical media, rarely happens.

Among multimedia systems, RT AV transmission is particularly demanded in applications such as video conferencing, on-line gaming, and broadcasting. The main requirements for RT multimedia transmission are minimal packet loss rate, low delay, low jitter and consistent available bandwidth [6].

There are many efforts trying to support RT multimedia traffic on top of Ethernet. However, most standard Ethernet-TCP/IP based solutions are not able to fulfil the deterministic or even isochronous timing requirements. Some RT protocols may achieve a better QoS, however, most of them have their own limitations [7]. For instance, usually the TDM based solution has a low bandwidth utilisation when the link is light loaded, the master-slave principle experiences the problem of single point of failure, and packets with the same priority will lead to a queueing latency when using priority based schemes.

The challenge is that the network can achieve the extremely low jitter as audio playback systems, which is in the nanosecond scale [8]. Most RT systems cannot achieve that without re-clocking and de-jitter buffers, hence increases the system complexity and overall latency. In this paper, the Flexilink protocol, an alternative professional media infrastructure reported by the European Broadcasting Union (EBU) et al. [9], will be evaluated. The Flexilink architecture provides a mechanism to achieve not only the TDM grade quality for professional live multimedia transmission, but also a potential true converged solution for different traffic types including the best effort data [10].

The remainder of this paper is organised as follows. Section II and III give a brief introduction to existing RT protocols and the structure of Flexilink, while Section IV describes the Flexilink MAC layer design and network simulation models. Section V expresses the parameters and scenarios in the simulation, and analyses the performance based on the simulation results. Finally, Section VI concludes the paper and discusses the future work.

1550-445X/17 $31.00 © 2017 IEEE
DOI 10.1109/AINA.2017.80

Layer	Category One: On top of TCP/UDP/IP		Category Two: On top of Ethernet		Category Three: With Modified Ethernet	
Upper Layers	Application		Application		Application	
	Best-effort	Real-Time Protocol	Best-effort	Real-Time Protocol	Best-effort	Real-Time Protocol
4 Transport	TCP/UDP		TCP/UDP		TCP/UDP	
3 Network	IP		IP		IP	Modified Ethernet
2 Data Link	Ethernet MAC		Ethernet MAC		Ether. MAC	
1 Physical	Universal Cabling					

Fig. 1: RT Ethernet protocol classification

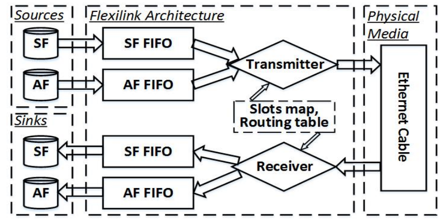


Fig. 2: The MAC layer design of Flexilink

II. Existing RT protocols review

Standard Ethernet was originally designed for the best effort IP transmission. To use it for the RT transmission providing acceptable QoS, additional mechanisms are needed. There are many protocols have been proposed to achieve the RT performance, which typically can be classified into three categories [11], [12], [7] as presented in Figure 1, on top of TCP/UDP/IP, on top of Ethernet, and with modified Ethernet.

1) Solutions on top of TCP/UDP/IP: Examples in RT multimedia systems are Real-Time Transport Protocol (RTP) [13], Real-Time Control Protocol (RTCP), and Integrated Services (IntServ) and Differentiated Services (DiffServ) based solutions. The RTP suite can achieve soft RT behaviour with a delay of millisecond level [14], which can be used in scenarios such as simple multicast audio conference, audio and video conference, mixers and translators, and layered encodings [13]. Many current network music researches focus on improved latency management and predication at millisecond scale, which may affects the music ensembles over the network [15], [16], [17].

Typically, IntServ and DiffServ are deployed as fine-grained and coarse-grained systems, respectively. The former uses the per-flow reservation with Resource ReserVation Protocol (RSVP), while the latter is based on traffic classification and marking. Per-flow reservation requires procedures like call set up, maintain and termination and thus many status information to store in the routers through a path, which makes pure IntServ complex and not scalable [18]. In contrast, DiffServ needs no setup time and offers scalability. However, no end-to-end guarantees are provided in DiffServ due to the lack of bandwidth reservation.

2) Solutions on top of Ethernet: This approach is based on software configuration. Realisations like Time-Critical Control network (TCnet) and PROFINET CBA (Component-Based Automation) fall into this category [11]. These solutions can obtain hard RT requirements with a cycle time of 1-10ms. However, some professional multimedia applications would require a low jitter within the range of nanosecond [8].

3) Solutions with modified Ethernet: This process focuses on modifying the Ethernet MAC layer to realise the isochronous RT requirements. In theory, this approach could get a much better RT performance without the limitations of best effort based Ethernet IP, in spite that they may increase the complexity when customise the lower layers or even use dedicated firmwares.

For instance, the SERCOS, EtherCAT, TTEtherent and Time-Sensitive Networking (TSN) Protocols are based on this mechanism [11]. These systems are capable of isochronous transmission which would satisfy most current time-critical applications. Nonetheless, the first two protocols both are based on the master-slave principle with its limitations [7], TTEtherent adopts a TDM based time cycle which does not utilise the full bandwidth, while TSN implements a priority based control scheme with tree types of services, which has not prevent the queueing problem among different flows with the same priority.

III. The Flexilink Architecture

A. Introduction to Flexilink

The Flexilink protocol was proposed for low latency multi-channel interactive AV streams. It is a layer-2 protocol based on modified and flexible TDM techniques with improved QoS, less overhead, lower complexity and more flexible bandwidth allocation in theory [19], which supports both the time deterministic and best effort traffic with a almost full bandwidth utilisation. Flexilink employs many mechanisms to overcome the problems of existing RT Ethernet solutions. It is a Category Three protocol and therefore should have a better RT behaviour, especially when transmitting time-critical data. Flexilink uses preallocated space in fixed time slots to guarantee deterministic transmission, and to alleviate queueing delays. Flexilink is loosely coupled, which would alleviate the single point of failure problem encountered in master-slave based systems. Also, Flexilink uses the reduced jumbo frame (RJF) format to reduce overhead. In addition, when the synchronous flow (SF) packet size is variable, the extra space left can be used for the best effort data transmission, which therefore increases bandwidth utilisation. However, the current Ethernet standard would need to be extended to support the theoretical MAC layer design of Flexilink. And to fully utilise its SF features, the end-to-end reservation needs to be established before the real-time traffic session starts, which will add some complications in multihop situations.

Flexilink uses a new flow based architecture which combines the advantages of both packet and circuit switching. In other words, it employs a specialised packet switching method based on the circuit switching scheme for transmission. Flexilink reserves bandwidth for RT traffic to guarantee its transmission with a low latency and also provides space for the asynchronous flow (AF) packets. Flexilink provides two

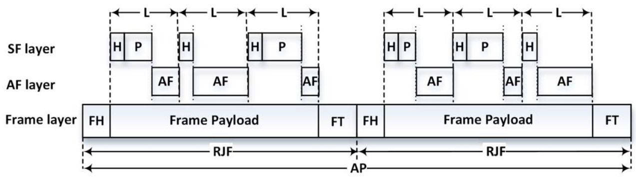


Fig. 3: The layered structure of Flexilink

TABLE I: The length of a SF packet's header

Header Length	Data Length
1 byte	$0 \sim 15$ bytes
2 bytes	$16 \sim 255$ bytes
3 bytes	$256 \sim 4096$ bytes

basic services, synchronous service and asynchronous service. The SF packets are put into predetermined locations on a link while the AF packets are fitted into the space between two SF packets. Overlarge AF packets are segmented to fit the space. The link containing the deterministic traffic is transmitted based on the circuit switching architecture while the best effort traffic is based on IP routing. An overview of the basic MAC layer design of the Flexilink architecture is presented in Figure 2. Flexilink adopts a dual-buffer mode to deal with the real-time and best-effort traffic separately.

B. The Flexilink structure

Flexilink employs a three layered traffic architecture, the SF, AF and frame layers, as depicted in Figure 3. The length of a fixed time slot, called frame size, is decided by the frame layer along with some other control messages. Each frame's payload (P) has a fixed size, in which fixed slots can contain a synchronous packet. The length of a synchronous packet is variable, therefore, the space left for the asynchronous packet is also variable. The SF packet has a simple header (H) containing only the length information which is used by the controller. To minimise the header size, the header length is variable depending on the length of the SF packet's payload. The detailed length information of the SF packet header is presented in Table I. A zero length SF packet, or empty slot, also needs a one-byte header to inform the controller so that it can use the space for AF packets.

Both SF and AF packets are allocated in the frame's payload. In the example of a Gigabit Ethernet implementation of Flexilink, a frame has a fixed allocation period (AP) of 124.96μs at the frame layer, which could allocate two jumbo frames of 7810 bytes. Flexilink uses a RJF structure without some unnecessary areas such as the source and destination addresses when compared to the standard Ethernet jumbo frame, because the packets are identified by their fixed positions which are predetermined by the controller. A link is formatted into a sequence of APs, each contains two consecutive RJFs. The structure of the RJF is expressed in Figure 3.

A RJF has a frame header (FH) and a frame tailer (FT) containing formatting messages. The FH begins with a two-byte preamble and start frame delimiter followed by a five-byte AES51 packet header and timing information. The FT is comprised of a four-byte frame check sequence (FCS) and a fourteen-byte inter frame gap (IFG). This design gives each RJF a maximal payload of 7785 bytes, which provides a bandwidth utilisation of 99.68 percent in theory.

IV. Simulation Model

The simulation structure model is illustrated in Figure 4 based on the MAC layer design of Flexilink demonstrated in Figure 2. The main Flexilink modules are simulated in node A, such as two traffic sources, the dual-buffer model, and the control logic. The transmitter and receiver work together to schedule the slots allocation and transmission. Flexilink can use the existing physical media. In this section, a simulation model is going to be built for Flexilink to verify the performance of this architecture. The model is realised on the SimEvents [20] platform within Simulink, which is developed for the discrete event simulation. The main parameters associated with each block are listed in Figure 5.

Assuming node A is the transmitter and node B the receiver, connected by the *cable* block. The propagation delay is calculated as dividing the cable's length by $2{\times}10^8\mathrm{m/s}$, 2/3 of the speed of light. It is the same for every packet as it depends on only the materials of the physical media. The SF and AF sources are controlled separately given a packet generation rate and packet sizes (a specified distribution). The *set SF packet header* block will first check the length of the SF packet's payload and compare it with the parameter settings expressed in Table I, then a calculated header will be generated and attached to the SF packet. Following the two traffic sources, there are two buffers to store the packets before they can be forwarded to the transmitter.

The *SF allocation* block works with the transmission clock and the transmission controller. It determines the SF packets' allocation and transmission time. When there is an empty slot, it will accept a 1-byte header from the *generate SF header* block. The *transmission control* block schedules the transmission of both SF and AF as well as skipping the frame control messages. The frame control messages containing the FH and FT are used to encapsulate the SF and AF packets into a RJF. The size and position of a RJF are fixed on a link as presented in Figure 3. The fix-sized frame control messages are transmitted at a constant frequency, which is controlled by the *frame control* and *synchronous clock* blocks.

All packets will stay in the single server located in the transmitter for some time depending on the value of transmission delay calculated as dividing the packet length by the link bandwidth. The transmitter has a feedback scheme which gives the transmission information back to the *transmission controller* block to further schedule the transmission. The transmission controller decides when the packets are able to be transmitted, depends on the packet size and available bandwidth. The link slots are allocated based on SF transmission rate characteristics to minimise the SF delay. An AF packet may need more than one slot to be transmitted.

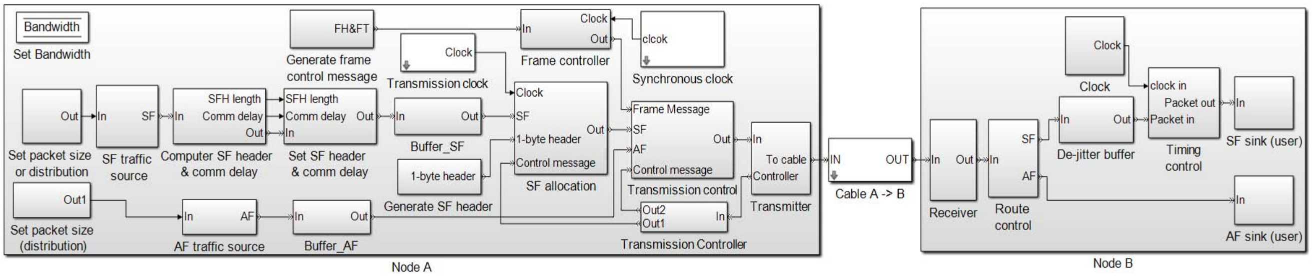


Fig. 4: The Flexilink network architecture

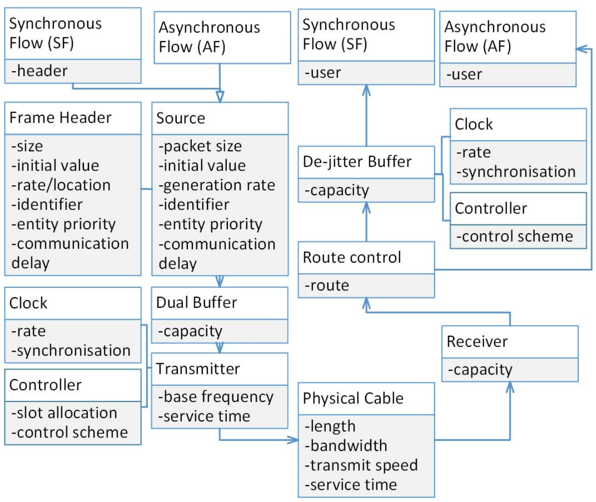


Fig. 5: The parameter list diagram

Given a 44.1K audio transmitted on a one Gigabit Ethernet link, 5.5 slots are needed in one AP. In practice, 6 slots will be allocated. Therefore, there will be some empty slots, one empty slot in every two APs in this case. An empty slot needs a one-byte header to inform the controller so that it can transmit AF packets to make a better utilisation of the available bandwidth. The one-byte header is generated whenever there is no SF packet waiting to be transmitted in a slot.

In node B, packets are first received and stored by the *receiver* block. The *route control* block will route the packets to different sinks (users) identified by their locations on the link. A de-jitter buffer is applied to SF packets before they are sent to the sink, which is controlled by the synchronous clock given a fixed frequency the same as the SF traffic source. Some SF packets will have a little bit of jitter caused by the transmission (empty slots) and synchronisation.

V. Simulation Evaluation

A. *Simulation Overview*

The Flexilink protocol is designed to provide a guaranteed deterministic traffic transmission with an acceptable low latency. It also supports the best effort traffic without affecting the RT traffic. In the following, several scenarios are employed

TABLE II: The global parameters

Parameter	Value	Parameter	Value
SF packet frequency	44.1 KHz	FH size	7 bytes
SF allocation slot	48 KHz	FT size	18 bytes
Frame frequency	16 KHz	Cable length	100 meters

to evaluate the performance of Flexilink, including increasing the amount of AF data, adding a burst of traffic and having multiple sources. The amount of AF traffic is increased through all the scenarios to verify whether the AF traffic would have any influence on the transmission of SF traffic.

In this model, assume the SF traffic source is a flow comprised of 128 audio channels. Each channel has a 44.1 KHz stereo 16 bits flow. The SF packet will have a 1-byte meta data, therefore, the size of each SF packet is 5 bytes and the total rate of the SF source is 640 bytes per sample (packet) in average. The size of each SF packet is variable which obeys a uniform distribution between 390 and 890 bytes. According to Table I, each SF packet will have a 3-byte header. The AF traffic source is simulated using a uniform distribution given a minimal and a maximal packet size, 64 and 1518 bytes, respectively. Flexilink will put AF packets into the gaps between two successive SF packets, therefore, it is the amount of AF data rather than the size of each AF packet that matters in this simulation.

Using a one-Gigabit Ethernet link, Flexilink guarantees 8 K APs per second. Thus 48 K allocation slots are needed when transmitting a 44.1 KHz audio. The basic parameters that will be used all through these several scenarios are listed in Table II.

For comparison, two Ethernet network models are also built. The main differences are listed as follows.

1) Basic Ethernet Network: This simulation model has only the *transmitter* and *receiver* blocks. There is no QoS guaranteed transmission in this model.

2) Priority based Ethernet: It has a higher priority for the SF port than the AF port, as well as a priority based transmission buffer, which gives the SF packet a higher priority to be transmitted.

In both models, the two traffic sources are kept the same, but without the Flexilink reservation, allocation and control schemes. All the parameters and settings used for the sources, the cable, the sinks and so on will also be kept the same as

TABLE III: Cases introduction

Test Case	Brief Description
Linear stress test	Increase the AF load of the link.
Burst test	Give a burst of AF traffic to each scenario.
Multiple sources	Extend each scenario to three SFs and three AFs.
Multiple-port mixed sources	Given two source nodes with both SFs and AFs.

the Flexilink model.

B. Simulation Scenarios

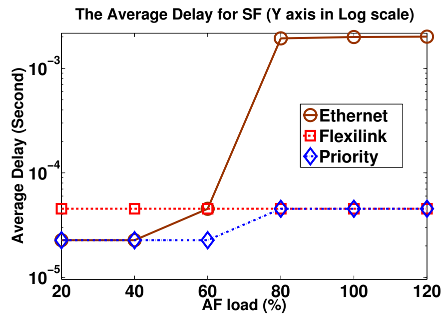
Given the global parameters described in the last section, we can calculate how much bandwidth is left for the AF traffic. This model simulates variable SF packet sizes. The average SF bandwidth utilisation is 22.688% of the total bandwidth including SF packet headers of the empty slots. This amount of SF data reflects the AV traffic in the real world scenario according to paper [21]. The frame's control messages take 0.32% of the total bandwidth. Therefore, there is about 77% of the total bandwidth available for the AF traffic. In the following, several simulation cases are used to compare their performances. Table III gives a brief introduction to each case.

1) Case One: Linear stress test: In this simulation, the amount of AF traffic is increased to increase the overall network load in each scenario to see whether it will affect the transmission of the SF flow. The AF flow will take 20%, 40%, 60%, 80%, 100% and 120% of the total bandwidth, respectively.

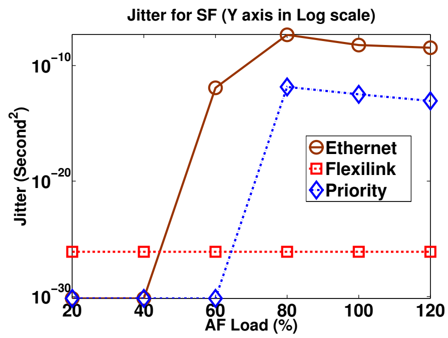
The simulation results are presented in Figure 6a, the average End-to-End (E2E) delay, and Figure 6b, the jitter, for each model against the increasing AF load. The jitter is calculated as the variance of the E2E delay in each scenario.

The basic Ethernet network without QoS implementation has a poor behaviour unless there is only a little traffic on the link. Generally speaking, Flexilink maintains a very stable performance no matter how much AF traffic we have pushed to the link. However, Flexilink performs slightly worse than the Priority Ethernet (PE) when it has a low network load. This is because Flexilink has an allocation mechanism to guarantee SF's transmission, which requires SF packets to wait in the buffer for some periods for the right slot to be transmitted. It gives one sample's delay in this situation. While in the PE model, the SF packet can be transmitted immediately unless there is a packet being transmitted, which will give a maximal delay of 12.144 µs. The Flexilink SF can be configured to be close to the source data rate in order to reduce the latency, although additional mechanisms may be needed for automatic configuration. However, the PE network will get worse when having a burst of traffic and/or multiple SFs, which will be discussed in the next several cases. PE also becomes precarious and gets more delay when given massive AF traffic.

The simulation models all have a de-jitter buffer implemented, which alleviates the time fluctuation problem. It is designed in the Flexilink architecture, while in reality, most network switches do not have this mechanism.



(a) Average Delay in Case One



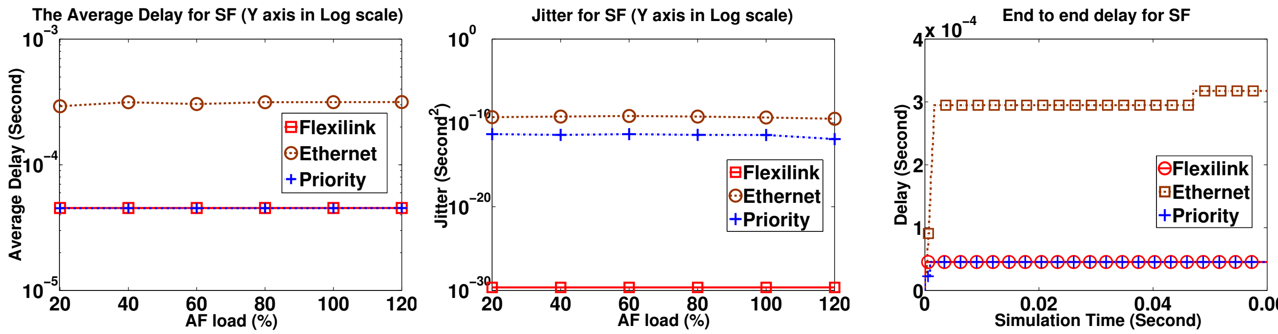
(b) Jitter in Case One

Fig. 6: Simulation results in Case One

In addition, AF sources will not always obey a uniform distribution in practice. Sometimes people may occasionally need to transmit or download a big file, then the burst of traffic may appear during some periods. Therefore, we are going to add a burst of traffic source with massive data which will take up all the bandwidth for some time.

2) Case Two: Burst test: In this case, a burst of traffic source is added to every model. It will generate plenty of 1518-byte packets during a small period with a frequency of 82345 packets per second, which will be able to take up all the available bandwidth on the link. The E2E delay and jitter for the SF are illustrated in Figure 7a and Figure 7b, respectively. It can be seen that the basic Ethernet network model has a significant delay and dramatic fluctuation. The E2E delays are similar for the Flexilink and PE networks. However, the PE model has a much larger jitter, than that of Flexilink. If we represent the jitter as standard deviation, it is around 1.88µs for the PE, which exceeds the 20ns limitation to be audible [8]. For Flexilink, it is about 1.54×10^{-6}ns, is much lower than that.

This change in performance can be explained by Figure 7c, which shows the detailed E2E delay for every SF packet in scenario three which has an AF load of 60%. When there is a burst of traffic, it will take up all the available bandwidth, which leads to an overflow in the buffer. The following packets will take more buffering time and may need to wait for another sample's period in the de-jitter buffer at the receiver. Thus,



(a) Average Delay

(b) Jitter

(c) The E2E delay for each packet

Fig. 7: Simulation results in Case Two: Burst test

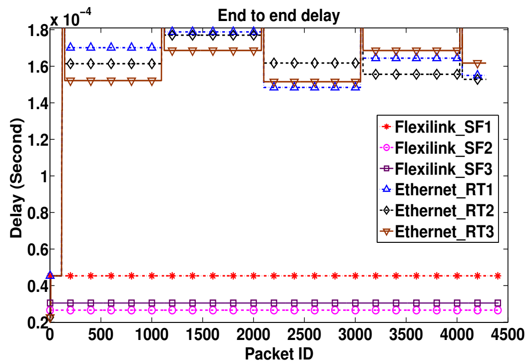


Fig. 8: Detailed E2E delay for SF in Case Three

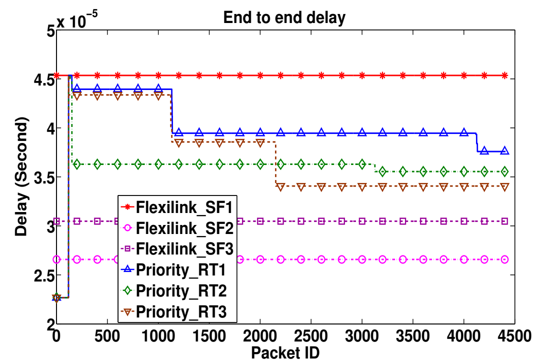


Fig. 9: Detailed E2E delay for SF in Case Three

there is a jump in the PE's performance, which gives a higher delay as well as a higher jitter. We can also find that Flexilink is not affected by the burst of traffic.

3) Case Three: Multiple sources: The model is extended to several SF and AF sources, which is a more general situation in practice. In this simulation, we will use three SFs and three AFs. The amount of traffic in each flow will be decreased to keep the total traffic the same as the last two cases. In the PE model, SF packets have higher priorities than AF packets, but there are no priorities among the three SFs.

Here we choose the third scenario with an AF load of 60% as an example to analyse their performance. The detailed E2E delays for Flexilink compared with the Ethernet and PE models are illustrated in Figure 8 and Figure 9, separately.

It can be seen from Figure 8 that the E2E delays for all SFs preserve at a relatively high level and experience fluctuations during some periods in the basic Ethernet model. It also started to drop SF packets with a rate of 2.8%, approximately. From Figure 9 we can see that SFs all have experienced a jump and several steps of decrease in the PE model. Different SFs do not have the same delay at the same time, and there are no patterns during a period. This relatively random delays will also lead to a higher jitter. Each SF in the Flexilink model holds the same E2E delay, which leads to a very small jitter. Different SFs have different delays. This is because a world clock was used in the model which will cause a different but stable delay within a sample period, depending on the synchronisation between the transmission and receiving ends.

4) Case Four: Multiple-port mixed sources: Considering the following scenario as described in . Nodes A and B are both transmitting several SFs and AFs to node C, simultaneously. There is only one node can be set to a higher priority, for example, node A. Based on Case Three, we assume node A has one SF and two AFs, while node B has two SFs and one AF. Within each node, the SF has a higher priority than the AF. All other parameters are kept the same as Case Three. Figure 11 shows the detailed E2E delay results.

A similar conclusion can be given to the basic Ethernet and Flexilink network models as Case Three. However, the PE model experiences a big change where there are some spikes whenever there is a burst of traffic, which means some SF packets have significant large delays. The third SF also has a higher average E2E delay than any other SFs in the PE model or the Flexilink model. In addition, the second and third SFs began to drop SF packets with an average rate of 1.77% and 4.88%, respectively.

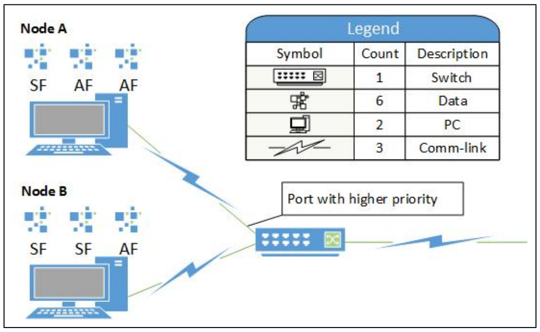


Fig. 10: Case Four settings description

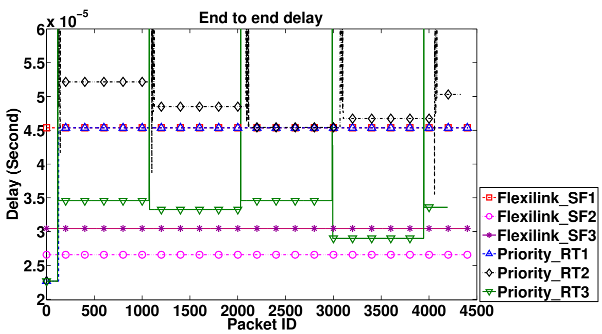


Fig. 11: Detailed E2E delay for SF in Case Four

TABLE IV: Factors that cause delay

Factor	Delay (Second)
Buffer	1.038×10^{-5}
Transmission delay	5.140×10^{-6}
Propagation delay	1.500×10^{-7}
De-jitter buffer	2.933×10^{-5}
In total	4.500×10^{-5}

C. Simulation Results Analysis

In the Flexilink network model, the average E2E delay of each SF packet for the first SF in all the scenarios is approximately 45.35µs. The delay is mostly caused by the transmission buffer, the transmission delay (transmitter), the propagation delay (cable) and the de-jitter buffer [22]. The values of these parameters are listed in Table IV. The total delay, a summation of the delays mentioned above, is shown in the end of the table. The calculated total delay is close to the value obtained from the simulation, 45.35µs.

Note that the average delay caused by the buffer is slightly less than 10.417µs which is half a sample period of the transmission link which has a base frequency of 48 KHz in this case. Similarly, the average delay caused by the de-jitter buffer is about 1.3 audio samples. If we change the synchronous clock at the receiver end, the delay may be slightly different but within an audio sample's period.

$$D_{difference} = D_{Flexilink} - D_{PE} \tag{1}$$

$$J_{ratio} = J_{Flexilink}/J_{PE} \tag{2}$$

TABLE V: Average E2E delay and jitter improvement for Flexilink compared to Priority based Ethernet

Case	Delay Difference (µs)	Jitter Ratio
Case 1 [a]	+ 11.35	2.48×10^{-14}
Case 2 [a]	+ 0.16	6.71×10^{-19}
Case 3 [b]	- 3.41	1.934×10^{-19}
Case 4 [b]	- 8.07	8.06×10^{-21}

[a] Delay and jitter are calculated as the mean of six scenarios (AF loads).
[b] Delay and jitter are calculated as the mean of three real-time flows.

A summary of the average E2E delay and jitter performance improvement for Flexilink compared to Priority based Ethernet is presented in Table V, which are calculated using Equation 1 and 2, respectively. In Equation 1, $D_{difference}$ denotes the different difference, while $D_{Flexilink}$ and D_{PE} denote the average delay of the Flexilink and PE networks, respectively. Similarly, J_{ratio} denote the jitter ratio, while $J_{Flexilink}$ and J_{PE} denote the mean jitter of the Flexilink and PE networks, respectively.

Through all scenarios, we can see that the E2E delay over the Flexilink network will not increase as the amount of AF data grows, and the jitter for all scenarios is close to zero. For comparison, both the E2E delay and jitter increase rapidly in the Ethernet model. The PE has a similar low latency and jitter when there are a small amount of AF data, but it gets worse when having massive AF data and/or multiple SF sources. The burst of traffic also has a significant influence on the performances of the basic Ethernet and PE models. Flexilink keeps performing well when more and more AF data are pushed to the Ethernet link, even with the burst of traffic and/or multiple sources.

VI. Conclusion

This work demonstrates the advantages of the Flexilink architecture and the protocol design in dealing with the fixed rate real-time multimedia traffic, in contention with the best effort data, using simulation, although it may introduce minor latency due to slot scheduling. These advantages are surrounding some of the key multimedia quality measures such as the end-to-end delay, jitter and packet loss rate. In particular, Flexilink is the only architecture examined whose jitter is below the audible threshold in the case of streaming live digital audio signals.

The uniqueness of Flexilink is that it combines the merits of both TDM and best effort network features. Hence it is a promising approach as an efficient model of the converged network technology in the professional media industry, without complex QoS management. The main challenge for Flexilink is the multi-hop end-to-end bandwidth reservation. Efficient scheduling and routing mechanism are key requirements to guarantee professional performance in the wide area networks. Therefore, future work should investigate how the slot reservation based mechanism can be developed and optimised over multiple hops and routes.

REFERENCES

[1] A. Faust, "Low latency replacement of isdn and 4-wire for remote broadcasts," in *135th Audio Engineering Society Convention*, 2013.

[2] BBC Academy, Technology, "Virtualising local radio," [Online]. Available: http://www.bbc.co.uk/academy/technology/article/art20130802172702565.

[3] D. C. Verma, H. Zhang, and D. Ferrari, "Delay jitter control for real-time communication in a packet switching network," in *Proceedings of TRICOMM'91*, 1991, pp. 35–43.

[4] J.-D. Decotignie, "Ethernet-based real-time and industrial communications," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1102–1117, 2005.

[5] V. Fineberg, "A practical architecture for implementing end-to-end qos in an ip network," *IEEE Commun. Mag.*, vol.40, no.1, pp. 122-130, 2002.

[6] T. Szigeti, R. Barton, C. Hattingh, and K. Briley, *End-To-End QoS Network Design: Quality of Service for Rich-Media and Cloud Networks*. Cisco Press, 2013.

[7] J.-D. Decotignie, "The many faces of industrial ethernet [past and present]," *IEEE Ind. Electron. Mag.*, vol. 3, no. 1, pp. 8–19, 2009.

[8] E. Benjamin and B. Gannon, "Theoretical and audible effects of jitter on digital audio quality," in *105th Audio Engineering Society Convention*, 1998.

[9] Joint Task Force on Networked Media (JT-NM), "Gap Analysis Report," European Broadcasting Union (EBU), Society of Motion Picture and Television Engineers (SMPTE) and Video Services Forum (VSF), 2013.

[10] Y. Wang, J. Grant, and J. Foss, "Flexilink: A unified low latency network architecture for multichannel live audio," in *133th Audio Engineering Society Convention*, 2012.

[11] M. Felser, "Real-time ethernet–industry prospective," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1118–1129, 2005.

[12] J. Jasperneite, J. Imtiaz, M. Schumacher, and K. Weber, "A proposal for a generic real-time ethernet system," *IEEE Trans. Ind. Informat.*, vol. 5, no. 2, pp. 75–85, 2009.

[13] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," *IETF RFC3550*, 2003.

[14] P. Neumann, "Communication in industrial automation what is going on ?" *Control Engineering Practice*, vol. 15, no. 11, pp. 1332–1347, 2007.

[15] C. Chafe and M. Gurevich, "Network time delay and ensemble accuracy: Effects of latency, asymmetry," in *117th Audio Engineering Society Convention*, 2004.

[16] B. Vera and E. Chew, "Towards seamless network music performance: Predicting an ensemble's expressive decisions for distributed performance." in *ISMIR*, 2014, pp. 489–494.

[17] C. Rottondi, M. Buccoli, M. Zanoni, D. Garao, G. Verticale, and A. Sarti, "Feature-based analysis of the effects of packet delay on networked musical interactions," *Journal of the Audio Engineering Society*, vol. 63, no. 11, pp. 864–875, 2015.

[18] M. A. El-Gendy, A. Bose, and K. G. Shin, "Evolution of the internet qos and support for soft real-time applications," *Proceedings of the IEEE*, vol. 91, no. 7, pp. 1086–1104, 2003.

[19] J. Grant, "Method and apparatus for transceiving data," Jul. 22 2010, wO Patent App. PCT/GB2010/050,029. [Online]. Available: http://google.com/patents/WO2010082042A1?cl=ru

[20] MathWorks, *SimEvents User's Guide*, The MathWorks, Inc., 2014.

[21] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet inter-domain traffic," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 75–86, 2011.

[22] D. D. Clark, S. Shenker, and L. Zhang, *Supporting real-time applications in an integrated services packet network: Architecture and mechanism*. ACM, 1992, vol. 22, no. 4.