2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering

Evaluation of Flexilink as Deterministic Unified Real-Time Protocol for Industrial Networks

Tianao Ma*†‡, Wei Hu*†,Yonghao Wang‡, Dalia El-Banna‡, John Grant§, Hongjun Dai¶
*College of Computer Science and Technology, Wuhan University of Science and Technology, China
†Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, China
‡DMT Lab, Birmingham City University, UK
§Nine Tiles Networks Ltd, Cambridge CB25 9HT, UK
¶School of Software Engineering, Shandong University, Jinan, P.R. China
martiniel.ace@outlook.com, huwei@wust.edu.cn, yonghao.wang@bcu.ac.uk, Dalia.Elbanna@mail.bcu.ac.uk,
j@ninetiles.com, dahogn@sdu.edu.cn

***Abstract*—The QoS of real-time network based on the traditional TCP/IP architecture is always perceived as a dilemma. The current real-time network design is trade-off between different network performances such as delay, packet loss, and transmission speed. We believe the traditional network architecture becomes a barrier of deterministic network performance. Time deterministic networking is demanded by some applications such as industrial control network, aviation, and interactive multimedia. In addition, it is important to guarantee the data integrity and timing accuracy even when the network is congested. This paper proposed a scheduling model for Flexilink which is a newly proposed dynamic TDM network protocol and architecture that is approved to be secure and stable. We compared proposed scheduling algorithm for Flexilink in comparison with classic best effort network and priority based network using simulation. The experiments results show that the proposed scheduling algorithm performs very stable even when the network is heavily loaded.**

***Index Terms*—real-time transmission, QoS, protocols, bandwidth allocation.**

I. INTRODUCTION

There is a huge increasing of audio video capturing and delivery using computer network in recent years, not only in the traditional entertainment media streaming applications but also in high reliable mission critical environment such as in autonomous vehicle and live surveillance. However, to ensure large amount of multimedia data to be transmitted over shared network media with close to full bandwidth utilisation is an challenge when low latency and realisable packets delivery are required. Some of these applications require carefully designed and managed network to ensure the real-time traffic to meet time constraints, meanwhile to support different types of non-real time traffic such as emails and file transfer.

Traditional TCP protocol is connection oriented. It has retransmission and congestion control mechanism that is not suitable for transmission of real-time low delay streams [1]. The design of Flexilink takes current ISO network model and infrastructure into account, using different flows and protocols to make logic decision. The resulting architecture can avoid high packet loss, achieve high throughput [2]. When using Flexilink to transmit real-time data stream, it demonstrates good delay and jitter performance [3].

This paper extends the work presented in into these main aspects, section II reviews some of the existing transmission QoS and some real-time Ethernet protocols. Section III discuss the structure of Flexilink and the crucial technology. Section IV establishes a simulation model, which theoretically defines the transmission scheduling scheme. section V compares the simulation model with traditional and prioritized networks. Finally the section VI, we explain the advantages and future development of Flexilink.

II. ASSESSMENT OF THE REAL-TIME TRANSMISSION

A. The QoS Management for Real-Time Transmission

The three aspects of QoS parameters are delay, packet loss rate and transmission rate. The combination of the three can be used to construct a hierarchical analysis matrix, which has different weights and decision factors for different system requirements.

Time delay is the most important part of real-time network transmission. For real-time transmission, the significant delay will lead to poor user experience, and even the crash of the overall system that causes serious security concerns.

The packet loss is tolerable in some systems. However, in today's highly demanding network environments,packet loss can be closely link to the security problem. In some situation, every single byte or every frames are not allowed to be lost. However, packet loss and delay are often the contradictory performance measures.

Transmission rate is an important measure of QoS. The transmission rate is not only related to the different types of data but also to the choice of the underlying framing structure. To solve the conflict between privacy protection and efficiency. KK Gai and MK Qiu proposed using multi-channel communication for high-level secure transmission [4], or using blend arithmetic operations on tensor-based fully homomorphic encryption over real numbers to increase security [5].

In order to guarantee the stability of information delivery over the network, it is important to measure these three aspects of network: delay, packet loss rate and transmission rate.

2324-9013/18/31.00 ©2018 IEEE
DOI 10.1109/TrustCom/BigDataSE.2018.00014

IEEE computer society

B. Real-Time Transmission Scheme

The OSI reference model divides the function of the entire network communication into seven layers. There are also different protocols for different types of transmission to achieve better results. It is difficulty and expensive to manage QoS of real-time communication under traditional network architecture.

There are many compelling features of traditional Ethernet, including the well integration with the TCP/IP stack so become part of Internet naturally. It is easy to implement of network automation to have network monitoring, control and troubleshooting. However, when using TCP to processes real-time signals, it might retransmits both real-time and non-real-time signals. It then causes the bandwidth congestion, network loop and packet loss [1].

The most common protocols for real-time network transmission are IntServ (Integrated services) and DiffServ (Differentiated services). IntServ uses the RSVP that monitors each service of the stream. However, when the data flow is in considerably large size in the network, the storage and processing capacity of the equipment will suffer a lot of pressure [6]. The Differentiated Services (DiffServ) model brings the flexibility of the definition of a variety of QoS services through PHBs (Per Hop Behaviours) and Traffic Conditioners. [7]. However, due to the lack of end-to-end bandwidth reservation and the service guarantee may be impaired on the congested links [8].

There are real-time transmission protocols proposed for industrial Ethernet. In 2001, Popp & Wenzel [9] introduced PROFINET based on the IEEE 802.3 Ethernet standard. It was first used for distributed automation systems with different types of data are sent over the same channel using TDMA with a highly precise synchronized cycle based on the IEEE1588 standard for precision clock synchronization protocol [9]. EtherCAT is a master-slave environment for transmitting Ethernet frames, using summed frames to correspond to multiple nodes, making low delay for the transmission, processing and retransmission. However in a large network, the large amount of computation of the node, the tremendous data, and the logical ring topology network transmission will bring enormous delay.

In summary, most of the protocols dealing with real-time transmission by defining priorities, introducing pre-emption mechanism and changing the transmission frame size. Flexilink, which is described next, uses a similar but a more elegant approach.

III. THE FLEXILINK ARCHITECTURE

A. Data Classification

In the initial Flexilink proposals [2], [3], it classified the internet traffics into two type: Synchronous Flow (SF) and Asynchronous Flow (AF). Since then, the IETF's DetNet group (https://datatracker.ietf.org/wg/detnet/documents/) has defined three types of network traffic. (1) Synchronous: fixed latency service where all the links a flow passes over are synchronised and transmission opportunities are according to a schedule. (2) Asynchronous: bounded latency service, with capacity (but not specific transmission slots) reserved for the flow. (3) Non-detnet/best-effort: no guarantees on latency, or even whether the packet will be delivered at all.

The AF definition of Flexilink is more like the (3) non-detnet/best-effort data that is defined in by DetNet Group. The SF data in Flexilink can be made to perform as (2) Asynchronous data if it is tunnelled via well managed traffic engineering with guaranteed throughput. In addition, for using Flexilink in local area control network for industry we also propose a Instantaneous Trigger Message (ITM) type that has highest priority that can interrupt SF and best effort traffic to act as emergency control message. These are described below:

- Synchronous flow (SF): SF is a real-time streaming signal, such as a multimedia signal. It has a special frame structure, consisting of short header with payload. The latter part is the overall transmission data. The header includes the entire addressing information, data size information and transmission type identification. It can be implemented after scheduling, only to read the header field to determine the transmission order and type. This means that the delay in reading the entire data filed is eliminated.
- Background flow (BF): BF is a non-real-time signal. Using the best-effort transmission method, insert transmission as soon as possible between the gaps of SF data.
- Instantaneous Trigger Message (ITM): ITM is an emergency signal, in order to deal with the information security issues of the transmission. Such signals can interrupt the entire transmission under abnormal conditions, and can also serve as interception signals, which can be used to protect critical information and react for emergency situations.

B. Transmission Period

Flexilink can use layer-2 transmission frame in the MAC layer. When Flexilink proposed, it is designed to use existing Ethernet infrastructure. So it choose a Reduced Jumbo Frame (RJF) as the transmission frame. Multiple RJFs can be made to an allocation period (AP), or use equal proportions to make small cycles. This partitioning can guarantee that each real-time streaming position is fixed, so that the delivery speed is at the fastest, and easy for hardware to determine the SF flow position. This kind of hyper-periodic conversion is called periodic segmentation and mapping, forming a uniform distribution of SF data.

In this paper, we propose a periodic conversion that is called 'as tight as possible' (ATAP). We arrange SF packets as closely as possible, densely scheduled in the transmission frame. According to different types real-time streaming flows, we choose different ordering methods in order to achieve maximum bandwidth utilisation, the transmission rate, and the throughput rate. There are also many ways to partitioning the cycle, as well as the special handling solutions for transmission flows that are difficult to transport or have extremely high payloads.

In the transmission data link, the idea of transmission cycle selection scheme is to use LCM of SF periods, to ensure that each transmission cycle is the same. The position of each SF packets in cycle is fixed, so the transmission speed can be made fast. and all the position of SF packets can be predicted.

C. Transmission Architecture of the Flexilink

Flexilink classify the data traffic at data link layer. As shown in Figure 1. It gives different priorities to different types of traffics and wrap it with a simple header structure put them into different buffers. It does the similar approach at the receiving end. For ITM message, it has highest priority that can interrupt other types of traffic.

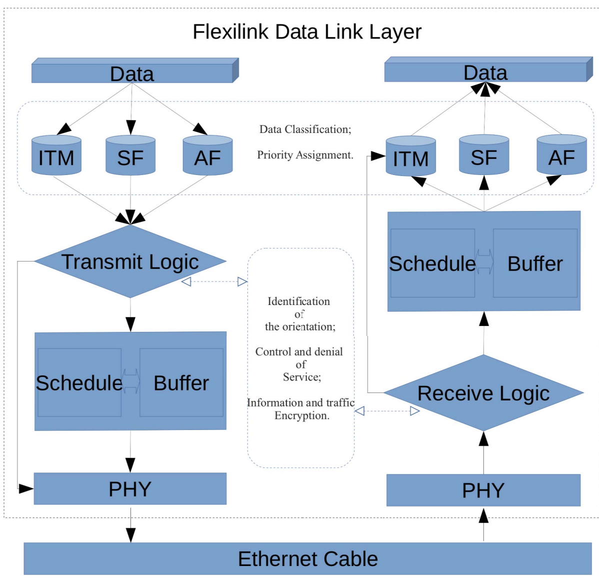

Fig. 1. Transmission architecture of the Flexilink

The transport logic and the receive logic of Flexilink architecture are designed to enhance the information security. The transmission logic layer includes the transmission of point-to-point identification, control or denial of service, information content and traffic encryption. At the receiving logic layer, there will be the transmitter identification, control and denial of service, analysis of encrypted content, provide feedback when there is problem in the transmission.

According to the selected scheduling method, the transmission cycle can be decided. Buffer is reserved for resources in advance to reduce the consumption of transmission. This part is to ensure the stable real-time streaming in heavily loaded transmission environment and to ensure there is no packet loss. Using RJF instead of traditional Ethernet frame, this is aimed at reducing the overhead by eliminating unnecessary fields from the frame header.

IV. SIMULATION MODEL

A. Synchronous Flow Parameters

Flexilink is a network transmission scheduling protocol for real-time task. So we used a group of periodic tasks to represent the SF traffic. Each task is defined in terms of three main parameters as $\tau_i = (C_i, T_i, BW)$ where:

- C_i is the execution time of task *i*.
- T_i is the period of task *i*.
- BW is the bandwidth of the used link. BW is a fixed constant that is equivalent to the network transmission rate, determined by different networks.

In order to calculate the feasible schedule, and can visually see the position of each real-time task in the transmission bandwidth. we define another two parameters:

- E_i amount of data sent during the execution time of task τ_i. E_i can be calculated using equation (1)
- P_i total amount of data sent during the period of task τ_i. Because the first is stored in the buffer, and then planning for the transmission frame by frame. It is equivalent that each period of task in the transmission frame reserved position. Similar to E_i, P_i can be calculated using equation (2).

$$E_i = BW \times C_i \tag{1}$$

$$P_i = BW \times T_i \tag{2}$$

B. Transmission Period Conversion Algorithms

Currently we consider real-time streams are periodic signals. So it is schedulable within the least common multiple of all task cycles. The next transmission cycle is to repeat this large cycle. Here we can define a large period Major Cycle (MJC) , as shown in equation (3).

$$MJC = lcm(P_1, P_2, ..., P_i,) \tag{3}$$

The idea of Flexilink is that SF data are transmitted on the same cycle, and the position of the SF task is the same position in each period. We propose a smaller cycle, that is, the maximum period of the transmission tasks.we call it a Minor Cycle (MNC), such as in equation (4).

$$MNC = max(P_1, P_2, ..., P_i) \tag{4}$$

When schedule a Minor Cycle, arrange the transmission of real-time tasks in the the Minor cycle. Add the tasks to fill the task node that was originally empty, so that the real-time flow is the same position for each small period.

In each MJC, each task τ_i has a number of times $f_i = \frac{MJC}{P_i}$.

In each MNC, each task τ_i has a number K_i of allocated slots. $K_i = \lceil \frac{MNC}{P_i} \rceil$.

Let $M = min(f_i) = \frac{MJC}{MNC}$, we have M number of Minor cycles in one Major Cycle.

In order to ensure that the number of tasks in each Minor cycles is the same, expanded the redundant virtual tasks. After expansion, each task appears times $f_i^{'}$ in the MJC. The number

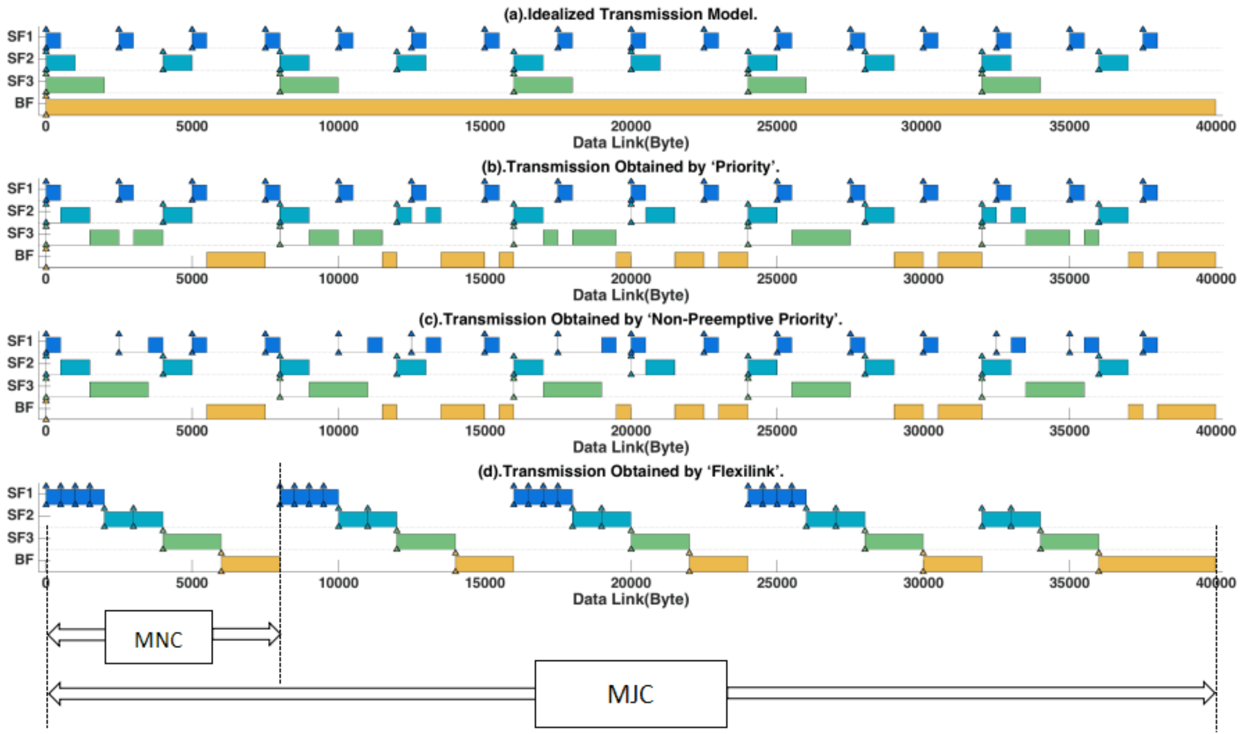

Fig. 2. The transmission scheduled simulation mode

of tasks that need to be expanded is f_i^{empty}. Equation 5 shows how to calculate f_i^{empty}.

$$f_i^{empty} = K_i \times M - f_i \tag{5}$$

During the actual transmission, only these spare locations are reserved to ensure that each MNC cycle is the same, but there is no real transmission of data here. An example of simulation shown in Figure 2.

C. Overload Transmission Scheme

Flexilink is superior to many prioritized real-time transmissions in that it is capable of transmitting without packets loss, in a heavy-loaded environments or beyond the bounds of transmission limits of RMS. We propose high-compression transmission scheme to guarantee full utilisation of bandwidth without any packet loss with the cost of small increasing of delay.

The previous occupied slots of the empty tasks can be calculated TS from equation 6, that need to condense.

$$TS = \sum_{n=1}^{N} f_i^{empty} \times c_i \tag{6}$$

Averaging the TS, and increase to every MNC. Replace the original Minor cycle with a new one, T_c.

$$T_c = MNC + \frac{TS}{M} \tag{7}$$

In order to ensure that the size of the entire Major Cycle unchanged, so cut off the original complement. Make j_{expand} convenient for judgement, it represents the number of small cycles that need to be compressed.

$$j_{expand} = min\left\{\lfloor \frac{f_i}{K_i} \rfloor\right\} \tag{8}$$

Within each of the original period MNC, the tasks number that need to be complemented is different. So occupied slots $L_j(i)$ represents the ith task in the jth cycle, like equation 9.

$$L_j(i) = \begin{cases} (j \times K_i - f_i) \times C_i & j \times K_i > f_i > (j-1) \times K_i \\ K_i \times C_i & j \times K_i - f_i > K_i \end{cases} \tag{9}$$

Therefore the uneven cycle $T_c^{'}(j)$ can be calculated as equation 10. The simulation results shown in Figure 3, indicating that each small cycle size is slightly different in this method.

$$T_c^{'}(j) = \begin{cases} TS & j \leq j_{expand} \\ TS - \sum_{n=1}^{|\tau|} L_j(i) & j > j_{expand} \end{cases} \tag{10}$$

This approach maximises bandwidth utilization by compressing the spare task flow gaps. To ensure that the efficiency of real-time streaming can theoretically reach 100%. And it also does not change the size of the original major cycle. If you need the same period for each minor cycle, the TS of the Equation 6 can be used. However, looping through each minor cycle, there will be some time elapse and delay. Both of

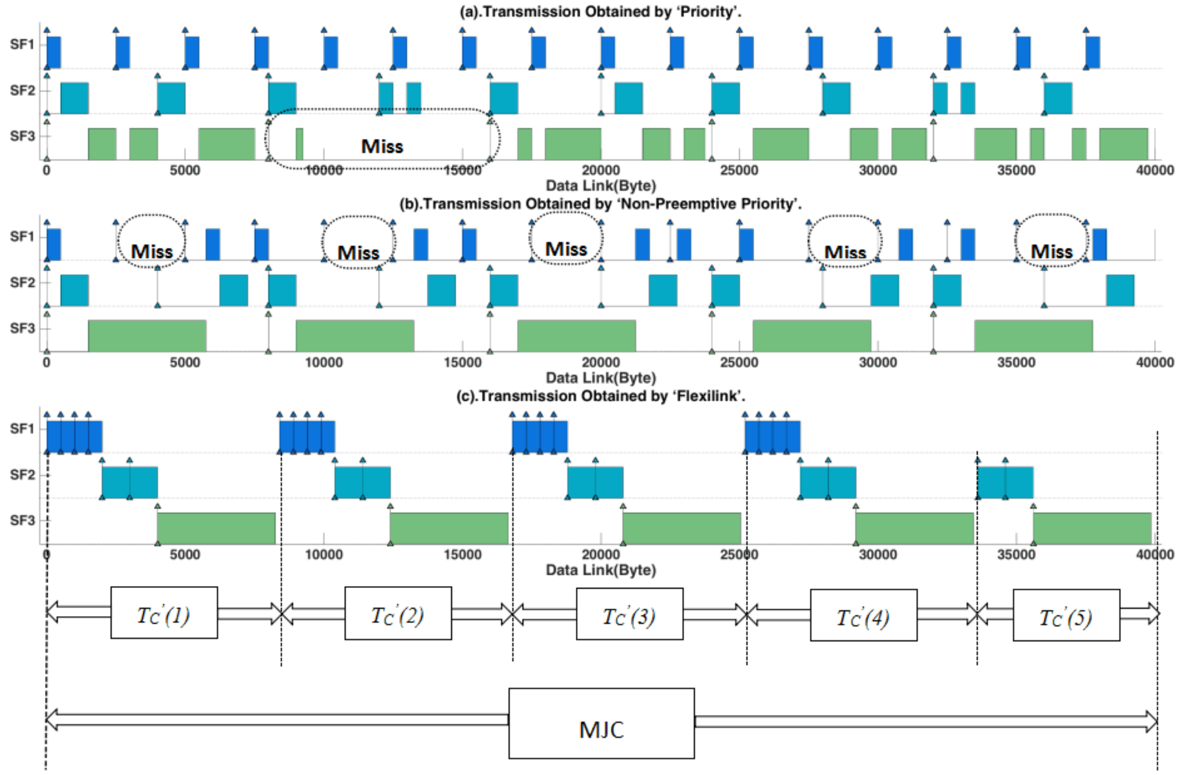

Fig. 3. The high utilization transmission scheduled simulation mode

these ideas can achieve safe and complete transmission under heavy loads, but the which method is better for what situation remains to be studied.

V. SIMULATION EVALUATION

A. The Transmission of SF and AF in Congested Networks

In the case of non-real-time and real-time streaming in network transmission. We use an unchanging real-time stream of 75 KHz. At the same time, by increasing the size of non-real-time stream, building the bandwidth from 0%, 20% to 100%, and create a network congestion model up to 120%. The figure 4 can show the different real-time flow delay from Ethernet, Priority network and Flexilink transmission.

In simulation, use traditional Ethernet transmission, when both real-time and non-real-time signals are sent at the same time, they hang together and wait for retransmission. After repeated 16 times, they will be completely lost. The transmission time of non-real-time signals is random, so the result is produced with averaging 100 random sets of data. Experiments show that in the traditional network transmission, real-time signal will suffer non-real-time flow interference, will result in a very large delay or even a large packet loss rate. Priority transmission will effectively avoid non-real-time interference, but still there will be slight fluctuations. Flexilink shows a very stable performance for the real-time traffic even under network congestion with delay of 3×10^{-5}s.

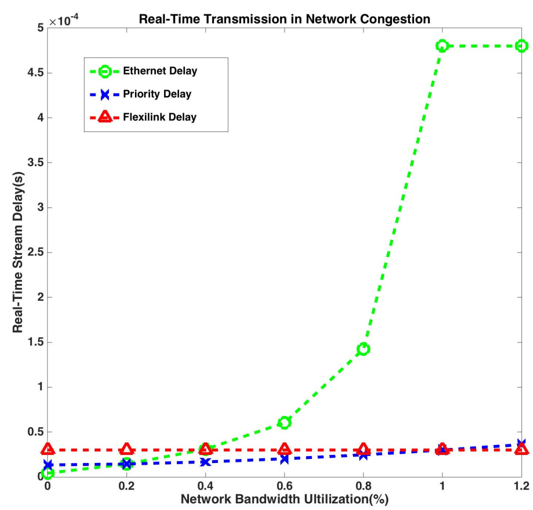

Fig. 4. Compare the delay in the network congestion

B. Specific Simulation

There are defined two tasks sets that represent real-time transmission of different utilization.The Set 1 is:

- $\tau_1(C_1 = 4 \times 10^{-6}s, T_1 = 2 \times 10^{-5}s, E_1 = 500Bytes, P_1 = 2500Bytes)$
- $\tau_2(C_2 = 8 \times 10^{-6}s, T_2 = 32 \times 10^{-6}s, E_1 = 1000Bytes, P_1 = 4000Bytes)$
- $\tau_3(C_3 = 16 \times 10^{-6}s, T_3 = 64 \times 10^{-6}s, E_1 = 2000Bytes, P_1 = 8000Bytes)$

Here we consider the transmission of multiple real-time stream, the priority network and Flexilink transmission. Priority network can be divided into preemptible and non-preemptible mode. The first simulation can see the obvious transmission sequence. Intuitively we can see our proposed Flexilink scheduling of Major Cycles and Minor Cycles. And the other set is :

- $\tau_1(C_1 = 4 \times 10^{-6}s, T_1 = 2 \times 10^{-5}s, E_1 = 500Bytes, P_1 = 2500Bytes)$
- $\tau_2(C_2 = 8 \times 10^{-6}s, T_2 = 32 \times 10^{-6}s, E_1 = 1000Bytes, P_1 = 4000Bytes)$
- $\tau_3(C_3 = 34 \times 10^{-6}s, T_3 = 64 \times 10^{-6}s, E_1 = 4250Bytes, P_1 = 8000Bytes)$

Figure 3 shows the results of Set 2, and compare the two examples, with only changing the transmission utilisation of the SF. The first experiment shows the utilisation ratio of 70%, the second is as high as 98.13%. It can be seen that the priority transmissions have limits. If the load exceed the transmission limit, there will be packet loss phenomenon. And this time will be used to the above Flexilink overload transmission scheme, it can achieve a safe and full load transmission, there is no packet loss.

C. Experimental Evaluation of Throughput

This experiment is to test the transmission throughput. Firstly, the utilization of real-time streaming is from 60%, gradually increased to 100%. Under the same transmission utilization circumstances, the use of random numbers to create 100 real-time data stream sets. Simulate and calculate the number of real-time streams that can be effectively transmitted. Then calculate the proportion of successful transmission of the original transmission number.

In figure 5, it can be seen that non-preemptive priority transmission is enormous affected by the utilization of real-time streams. In the case of real-time traffic congestion, the packet loss rate is high. Preemptive priority transmissions are slightly better. but when the utilisation is more than 85%, real-time transmission will have packet loss. Only Flexilink can be transmitted completely and safely.

VI. CONCLUSION

In this article, we present a new Flexilink scheduling algorithm that is called “as tight as possible”. This is specifically targeted for real-time transport over the network. Experiments that compared with traditional and prioritized transmission shows Flexilink using our algorithm can achieve excellent latency, real-time streaming without packet loss, and good throughput.

Currently Flexilink is still in the research phase, there are still many issues need to be explored. Such as buffer, the optimal division major cycle or minor cycle, and transmission scheduling. And Flexilink can be extended to more areas.

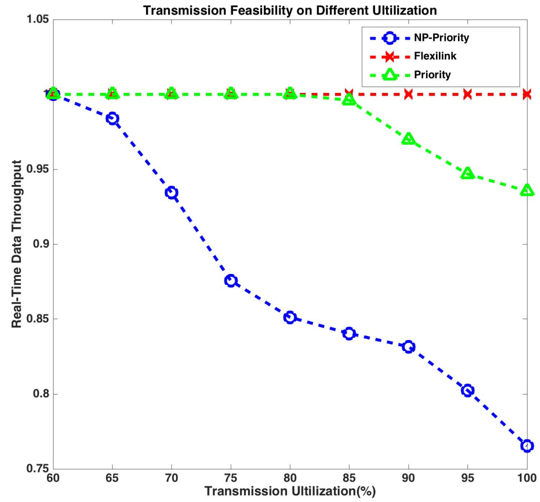

Fig. 5. The fraction of the transmission throughput

REFERENCES

[1] J. Silvestre-Blanes, L. Almeida, R. Marau, and P. Pedreiras, “Online qos management for multimedia real-time transmission in industrial networks,” *IEEE Transactions on Industrial Electronics*, vol. 58, no. 3, pp. 1061–1071, 2011.

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

[3] Y. Song, Y. Wang, P. Bull, and J. D. Reiss, “Performance Evaluation of a New Flexible Time Division Multiplexing Protocol on Mixed Traffic Types,” in *Advanced Information Networking and Applications (AINA), 2017 IEEE 31st International Conference on*. IEEE, 2017, pp. 23–30. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/7920885/

[4] K. Gai, M. Qiu, Z. Xiong, and M. Liu, “Privacy-preserving multi-channel communication in edge-of-things,” *Future Generation Computer Systems*, 2018.

[5] K. Gai and M. Qiu, “Blend arithmetic operations on tensor-based fully homomorphic encryption over real numbers,” *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1–1, 2017.

[6] D. Kandlur, A. Mehra, and D. Saha, “Rsvp and integrated services in the internet,” 2007.

[7] C. J. Barenco, A. A. Salona, and J. I. Moreno, “An architecture of qos services for a core internet network over dtm,” in *European Conference on Universal Multiservice Networks*, 2000, pp. 351–359.

[8] J. M. Batalla, “Advanced multimedia service provisioning based on efficient interoperability of adaptive streaming protocol and high efficient video coding,” *Journal of Real-Time Image Processing*, vol. 12, no. 2, pp. 443–454, 2016.

[9] M. Popp and P. Wenzel, “Profinet-linking worlds,” in *Emerging Technologies and Factory Automation, 2001. Proceedings. 2001 8th IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 519–522.