2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems

Short Cycle Conversion Scheduling Model for Flexilink Architecture

Yi Guo*†, Jing Liu*†§, Yonghao Wang‡, Wei Hu*†
*College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China
†Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan, China
‡Digital Media Technology Lab, Birmingham City University, BirmingHam, UK
Email: gavinpoet@gmail.com, luijing_cs@wust.edu.cn, yonghao.wang@bcu.ac.uk, huwei@wust.edu.cn
§Correspondence author

***Abstract*—Nowadays, most of the data traffic on the Internet is audio and video content, such as streaming media, Internet TV and so on. High-quality interactive multimedia content requires a high-quality real-time network. The traditional network protocol cannot meet all of the low latency, reliability, and time deterministic requirements for real-time data transmission. Furthermore, the currently packet-switched network cannot avoid the disadvantages of delay, large overhead, and time uncertainty. Flexilink is a kind of new dynamic time division multiplexing protocol architecture that provides a low-latency, unified network infrastructure for real-time traffic and traditional best-effort traffic. Based on the Flexilink protocol, this paper proposes a new data flow scheduling model called "Short Cycle Conversion (SCC) " to further ensure deterministic transmission and low latency. Experimental results show that SCC provides better real-time performance than the existing RMS and NP-RMS algorithms and can meet requirements of professional multimedia data transmission.**

***Index Terms*—Flexilink, latency, scheduling, deterministic**

I. Introduction

With the increasing network bandwidth, multimedia data such as audio video content has been recently exceed the ordinary file transmissions on the Internet. High-quality multimedia data has higher complexity and variability, which causes the data transmission with higher complexity and unpredictability. This will have an negative impact on high-quality multimedia data transmission which required to be low-latency, reliable, and real-time.

Due to retransmission and congestion control mechanisms, TCP network transmission is not suitable for low-latency real-time transmission [1]. For network voice communication, network video chat, etc., people often have bad experience on unstable data transmission. Not to mention high-definition interactive multimedia applications and other commercial applications which have high requirement on low latency, like intercontinental high frequency transactions and e-banking, as well as the emerging internet of things. Existing solutions, such as complex quality of service (QoS) management [2] and redundant bandwidth, do not fundamentally solve the problem [3]. Instead, they greatly increase complexity, cost, and energy of network systems [4] [5].

Flexilink is a new network framework protocol that supports both best-effort data and efficient real-time traffic (real-time data) [6]. It can make full use of the effective network bandwidth and considers the time management of real-time traffic. Also it has the advantages of high throughput, low latency, time certainty and avoiding high packet loss rate.

Based on Flexilink, this paper proposes a novel data flow transmission scheduling model SCC to achieve low-latency, reliable, and real-time data transmission. The model perfectly fits the transmission strategy of the Flexilink protocol and can improve the throughput of data transmission, making SCC more efficient to reduce transmission delay, avoid packet loss, and predict the transmission time of critical data.

The rest of this paper is organized as follows. The second section describes the Flexilink protocol and the RMS scheduling algorithm [7]. The third section proposes the scheduling model of SCC. In the fourth quarter we conduct experiments to test the performance of SCC. Finally in the fifth section, we summarize the paper.

II. Related work

A. Flexilink protocol

Flexilink is a Layer 2 based network protocol that combines the advantages of Time Division Multiplexing (TDM) and best effort networks [8]. As a result, it can effectively support best-effort general data as well as time-critical audio data. Flexilink's hybrid transmission mode predicts the transmission delay of time-critical data, improves transmission efficiency and guarantees quality of service (QoS). Because Flexilink is developed with the prototype network processor architecture and interface cards, it has maximum compatibility with existing network infrastructure and related network protocols. Flexilink supports full-duplex mode operation, avoiding the uncertainty of CSMA/CD.

In Flexilink network, there are mainly three types of traffic data: Synchronous Flow (SF), Asynchronous Flow (AF) and Control Message (CM). SF refers to data that has a fixed period and requires predictable delays, that is, time deterministic data [6]. AF refers to data that has no high requirement for latency but is transmitted as early as possible, that is, best-effort data. CM is a kind of initial interrupt signal and present in session control and link management to ensure the reliability of data transmission. Among these traffic data, CM has the highest transmission priority.

The data link layer (MAC layer) of Flexilink uses the second layer of network. For CM, since it controls connections and interrupts, the amount of data contained is minimal. Moreover, due to its highest priority, it is not encapsulated into frames for transmission but transmitted directly. The real-time performance of SF and AF is not as high as that of CM, but the amount of data is much larger than that of CM, so the method of transmitting frames is used.

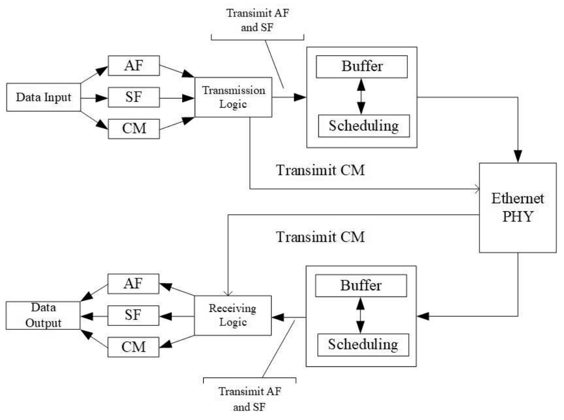

Fig. 1. The transmission architecture of Flexilink

The transmission architecture of Flexilink is shown in Fig. 1. It has the following main parts: transmission logic, receiving logic, and buffer. The transmission logic layer includes point-to-point identification, real-time transmission of CM data, and transmission of information content and traffic encryption. The receiving logic layer identifies the transmitter, receives the CM data in real time, and analyzes the encrypted content [3]. The design of the transmission logic and the receiving logic effectively improves data security. The buffer can reduce data jitter during transmission, degrade the probability of packet loss, and ensure stable transmission under high load.

The data transmission consists of two stages. First, data enters the transmission logic, and the transmission logic identifies the data. If the data is the CM data, it is transmitted directly to the receiving logic over the network. If the data is the SF data or the AF data, it is transmitted to the buffer. When a certain amount of data is buffered in the buffer, data is scheduled and sent to the network. Second, the buffer of the receiving logic end buffers the transmitted SF and AF data and performs scheduling. After scheduling, the receiving logic identifies and output data.

For Flexilink, by using a Simplified Jumbo Frame (RJF) as a transmission frame, the transmission time of multiple RJFs are regarded as an Allocation Period (AP), which makes each data flow have a fixed position in each AP [7]. For time division, a periodic uniform distribution of SF data is formed. As shown in Fig. 2, it is an ideal transmission link. In a link with sufficient bandwidth, we can pre-allocate space or time slots for SF data flows. AF data flow can be transmitted in the gap of the SF data flows during transmission process.

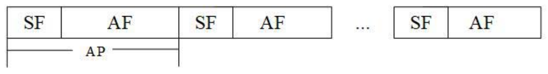

Fig. 2. An ideal traffic link

The purpose of SCC proposed in this paper is to convert the AP into shorter periods to schedule as much SF data as possible, which can ensure the time certainty of the data and make the AP cache time shorter.

B. The RMS algorithm

The Ratio Monotonic Scheduling (RMS) is a static real-time scheduling algorithm and known to be optimal [9]. It assigns a fixed priority for each task on the basis of their periods. The shorter the period of a task, the higher its priority. For example, there are two tasks u and v, and the period of u is smaller than that of v, then u has a priority higher than v. RMS typically makes decisions at the least common multiple (LCM) period of all tasks, where the LCM period is repeated until all tasks are scheduled. RMS can not only solve preemptive task scheduling problem, but also non-preemptive task scheduling problem. When it is used to solve non-preemptive task scheduling problem, it is called NP-RMS [10].

III. The proposed SCC scheduling model

In this section, we will propose the SCC scheduling model. Since the CM data is directly transmitted through the transport logic layer, only the transmission of the SF data and the AF data are considered in the Flexilink protocol.

A. Data flow task set

For SF data streams which are usually periodic, we can represent them with a set of periodic data flow tasks, and each task $Flow_i$ is a data flow with 3 parameters C_i, T_i, and D_i. Where

- C_i is the worst transmission time of task $Flow_i$.
- T_i is the transmission period of task $Flow_i$.
- D_i is the cutoff transmission time of task $Flow_i$(Usually equal to the end of the current cycle).

For the AF data stream which is non-real-time and the data stream length is variable, we transmit it in the gap of SF data streams during transmission in the model.

B. The period of the data flow task set

Periodic task scheduling algorithms usually make decisions based on the LCM period of all tasks, and the subsequent task scheduling is repeated in each periodic LCM. SF data stream in Flexilink is periodic, so our SCC scheduling model also adopts this LCM period decision method.

Let TS be the initial period of a set of real-time data flow tasks, then we have

$$TS = LCM\{T_i\}, 1 \leq i \leq n, \tag{1}$$

where n is the number of data flows (i.e., tasks) in the given set.

C. Scheduling priority

The SCC scheduling model uses a hierarchical priority allocation strategy to divide data streams into two layers. The first layer is the real-time data stream SF, and the second layer is the non-real-time data stream AF. Since the SF data stream is real-time and the AF data stream is non-real-time, the first layer has a higher priority than the second layer.

For each data flow in the SF data stream set in the first layer, we use the priority strategy of the RMS algorithm to assign priority for it. The shorter the data flow period, the higher its priority. In order to avoid the interruption which increases the burden on the network transmission system, preemption behaviors cannot occur between data streams [11].

The priority strategy of the SCC scheduling model can effectively eliminate data interference during the mixed transmission process of the SF data stream and the AF data stream, and improve the security of data transmission.

D. SCC scheduling strategy

Generally, the period of a data flow task set is the least common multiple period of all data flows in the data flow set. This often makes the LCM too large and results in too much data being cached in the buffer. Excessive cached data imposes an excessive burden on network devices that it is prone to overload, high latency, and time uncertainty during the transmission process, causing a poor network transmission experience. SCC reduces the buffer pressure by narrowing the period TS of the data flow task set, conforms to the time deterministic idea of Flexilink, and schedules the buffered data flow in the buffer. By doing so, it makes the data transmission mechanism have the characteristics of estimating the transmission time of data.

In the Flexilink protocol, multiple simplified jumbo frames are cached as one transmission frame. The AP of the transmission frame is the period TS of the data flow task set. SCC proposes a strategy of adding virtual data flow tasks to greatly reduce the TS. Let TS' be the narrowed task set period, and we set TS' to be the period of the data flow with the largest period in the real-time data flow task set, that is

$$TS' = max\{T_i\}, 1 \leq i \leq n. \tag{2}$$

In a TS, let f_i be the frequency that the real-time data flow $Flow_i$ appears, then

$$f_i = TS/T_i, 1 \leq i \leq n. \tag{3}$$

In a TS, suppose that the number of TS' is N, we have

$$N = min\{f_i\}, 1 \leq i \leq n. \tag{4}$$

In the SCC scheduling model, we propose a concept called ideal frequency which refers to the number of occurrences of each real-time data flow. Denote f_i^{ideal} to be the ideal frequency data flow $Flow_i$. We have

$$f_i^{ideal} = \lceil \frac{f_i}{N} \rceil, 1 \leq i \leq n. \tag{5}$$

Since the value of f_i^{ideal} is rounded up, a situation may occur in a TS': some data flows can reach the ideal frequencies, and other data flows have a difference from their ideal frequency value by one. To solve this problem, we add virtual tasks (invalid data flows that occupy transmission resources but have no effect on the system) to replace the corresponding real-time data flows, so that the frequency of each data flow can reach the ideal frequency fantasy in each TS'. After virtual tasks are added, the number of the same data flow in each TS' is the same. Then, we only need to schedule data flow tasks in each TS' according to the same scheduling scheme. The period of the data flow task set can be changed from the initial period TS = LCM$\{T_1, T_2, \ldots, T_n\}$ to TS', so that it is shortened several times of TS'.

For the problem of how to schedule the data flows in each TS', we assign a priority for each data flow according to the priority allocation strategy of RMS: the shorter the data flow period, the higher its priority. Then we schedule all the data flows in a TS' by the order of their priorities. After data streams in each TS' are released from the buffer, they will be transmitted according to their scheduling orders. In each TS', all data streams may not be able to fill the entire TS' peirod. In order to avoid wasting system resources and bandwidth, we choose to perform non-real-time best effort data flow AF in the remaining time slot.

The caching mechanism of the Flexilink protocol enables Flexilink to provide time-deterministic data transmission. SCC is time-deterministic in accordance with the Flexilink protocol. When the entire TS' is buffered in the buffer, the buffer will release all the data flows in a TS' for transmission. Since SCC has no preemption mechanism, we can accurately calculate the locations of all data flows during transmission, which allows to estimate the time and delay of all data flows.

RMS has been proved to be optimal for static priority scheduling. SCC follows RMS' priority allocation strategy to schedule data flows in a TS'. Therefore, it is also optimal.

Fig. 3 gives an example generated by the SCC scheduling model. In the example, there are three real-time data flow tasks $Flow_1$, $Flow_2$, and $Flow_3$ with periods satisfying $T_1 < T_2 < T_3$. Initially, TS = LCM$\{T_1, T_2, T_3\}$ and $TS' = T_3$. Suppose that $f_1 = 15$, $f_2 = 10$, and $f_3 = 6$, and we obtain $N = 6$. Furthermore, we get $f_1^{ideal} = 3$, $f_2^{ideal} = 2$, and $f_3^{ideal} = 1$. In the first TS' of a TS, $Flow_1$ occurs three times, $Flow_2$ occurs twice, and $Flow_3$ occurs once, so the first three data flows are $Flow_1$, then two $Flow_2$, then one $Flow_3$. After that, TS' has remaining time slot and we use it to perform a AF data flow. In the second TS' of the TS, $Flow_1$ occurs twice, $Flow_2$ occurs twice, and $Flow_3$ occurs once. To keep $f_1^{ideal} = 3$, a virtual flow is added with the same transmission time as that of $Flow_1$. Herein, the first two data flows are $Flow_1$, then one virtual flow, then two $Flow_2$, then one $Flow_3$, then one AF in the second TS'. Table 1 summarizes the number of virtual flows added in each TS' of the TS.

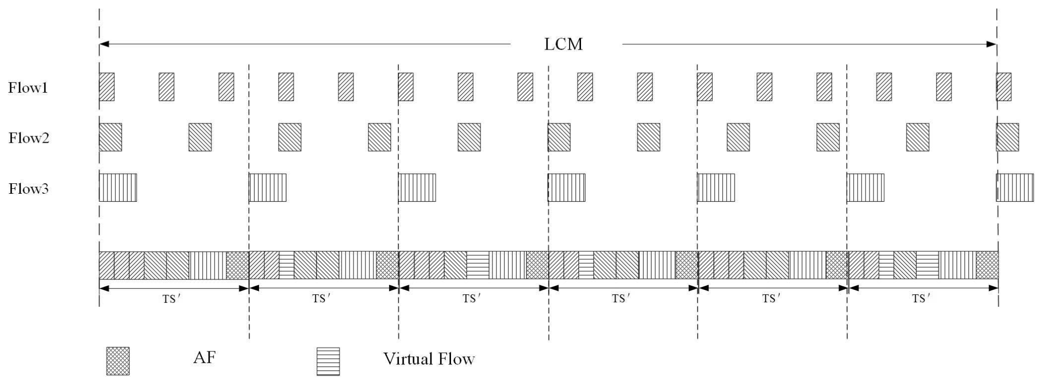

Fig. 3. An example generated by the SCC scheduling model

TABLE I
NUMBERS OF VIRTUAL FLOWS ADDED IN EACH TS'

	1st TS'	2nd TS'	3rd TS'	4th TS'	5th TS'	6th TS'
$Flow_1$	0	1	0	1	0	1
$Flow_2$	0	0	1	0	0	1
$Flow_3$	0	0	0	0	0	0

IV. EXPERIMENTS AND RESULT ANALYSIS

In this section, we conduct experiments to evaluate the performance of SCC from three aspects: transmission delay, high load transmission, and delay rate. All experiments in this paper are simulated on Matlab.

A. Transmission delay

Let B be the bandwidth of the transmitted network, S be the total amount of data of one TS, and S_i is the amount of data of a data flow $Flow_i$, then

$$S = B \times TS, \tag{6}$$

$$S_i = B \times C_i. \tag{7}$$

In order to facilitate testing, we used a fixed bandwidth of $B = 1Gbit/s$. We also suppose there are three real-time data flows $Flow_1$, $Flow_2$, and $Flow_3$. The values of three parameters of each data flow are randomly generate by the random function in Matlab as following.

- $Flow_1(C_1 = 1*10^{-6}s, T_1 = 6*10^{-6}s, D_1 = 6*10^{-6}s)$
- $Flow_2(C_2 = 2*10^{-6}s, T_2 = 1.2*10^{-5}s, D_2 = 6*10^{-6}s)$
- $Flow_3(C_3 = 6*10^{-6}s, T_3 = 2.1*10^{-5}s, D_3 = 6*10^{-6}s)$

Through (6), we can calculate the amount of data in a TS of S = 10500 bytes. Through (7), the data volume of $Flow_1$, $Flow_2$, and $Flow_3$ is separately S_1 = 125 bytes, S_2 = 250 bytes, and S_3 = 750 bytes.

Fig. 4 shows the scheduling results of above three data flows generated by different scheduling methods simulated on Matlab. It consists of 4 parts. The first part shows the number of each data flow occurred in each TS, and it is not a true data transmission scheme. The second part shows the data transmission scheme produced by RMS, the third part by NP-RMS, and the fourth part by our proposed SCC. The triangle symbol represents the Initial Transmission Time (ITT) of data flow in the TS, and the pink data flow is the added virtual flow. The jitter of real-time network transmission is mostly solved by the buffer mechanism. Since the data flow task set period of SCC is much shorter than that of RMS and NP-RMS, it has lower requirements for the buffer mechanism of the network system. As a result, SCC can reduce the cost of the network system.

In practical applications, the delayed arrival and early arrival of data packets have opposite effects. Generally, the packets arrived in advance have a positive effect on network transmission. During transmission, it will significantly improve the transmission quality of network that the data flow with higher priority arrives in time. In Fig. 4, $Flow_1$ data flows can be transmitted without exceeding the ITT using RMS and SCC; Only 71.4% $Flow_2$ data flows can be transmitted before the ITT using SCC; $Flow_3$ data flows cannot be completed on time and have a slight delay using all three algorithms RMS, NP-RMS and SCC. Therefore, SCC has significant advantage in transmission delay.

B. High load transmission

Flexilink's architecture supports high load transmission that SCC performs better than traditional priority scheduling methods in high load situations. We perform transmission simulation through a set of three real-time data flow tasks $Flow_1$, $Flow_2$, and $Flow_3$ with real-time data flow transmission efficiency of up to 94.28%. The information of the three data flows are as following.

- $Flow_1(C_1 = 1*10^{-6}s, T_1 = 6*10^{-6}s, D_1 = 6*10^{-6}s)$
- $Flow_2(C_2 = 2*10^{-6}s, T_2 = 1.2*10^{-5}s, D_2 = 6*10^{-6}s)$
- $Flow_3(C_3 = 6*10^{-6}s, T_3 = 2.1*10^{-5}s, D_3 = 6*10^{-6}s)$

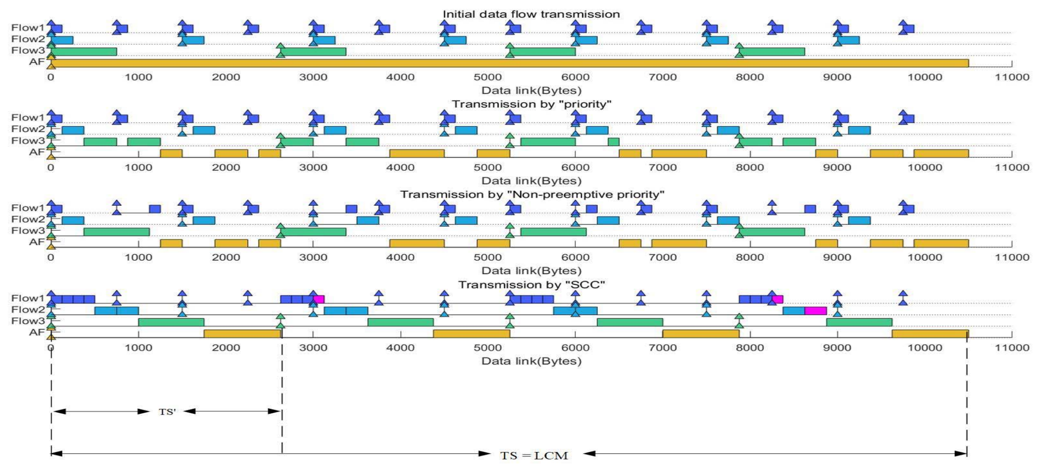

Fig. 4. Transmission schemes generated by RMS, NP-RMS and SCC

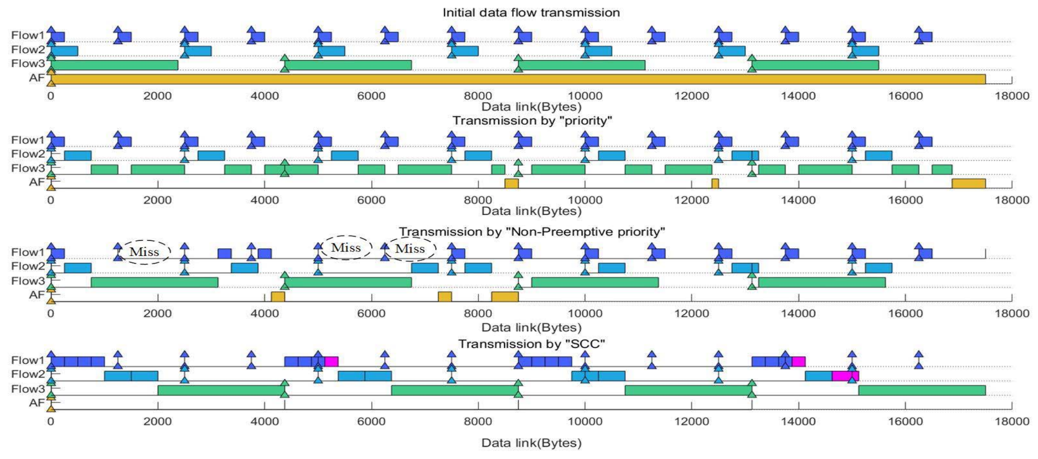

Fig. 5. Transmission schemes generated by RMS, NP-RMS and SCC for high load transmission

Fig. 5 shows the transmission schemes generated by different scheduling methods for data flows $Flow_1$, $Flow_2$, and $Flow_3$ in data flow task set. It also consists of 4 parts. The first part is not a true data transmission scheme. The second part shows the data transmission scheme produced by RMS, the third part by NP-RMS, and the fourth part by our proposed SCC. The triangle symbol represents ITT of data flow in the TS, and the pink data flow is the added virtual flow. We can clearly see that during transmission process using the NP-RMS model, $Flow_1$ with the highest priority has multiple loss, which seriously affects the transmission quality. For the RMS model, the number of data flow interruptions is significantly increased, which cause extra burden to the network transmission system. However, the SCC model does not undergo any interruption or data flow loss during high load transmission, showing extremely high network transmission stability.

Fig. 6 shows the delay attained by SCC, RMS and NP-RMS respectively for data flows $Flow_1$, $Flow_2$, and $Flow_3$ in one TS of theirs. We set the delay of the data flow arrived in advance to 0 and the delay of the lost data flow to infinity. The above subgraph compares delay attained by SCC and RMS, and it can be seen that the delay of SCC is lower than that of RMS. The below subgraph compares delay attained by SCC and NP-RMS, and the NP-RMS cause much larger delay that SCC, especially for $Flow_1$ data flow.

C. Delay rate

Early arrival of data flows has a positive effect on the network system during transmission, while delayed arrival has a serious negative impact. In order to intuitively reflect the low latency advantage of SCC in the transmission process,

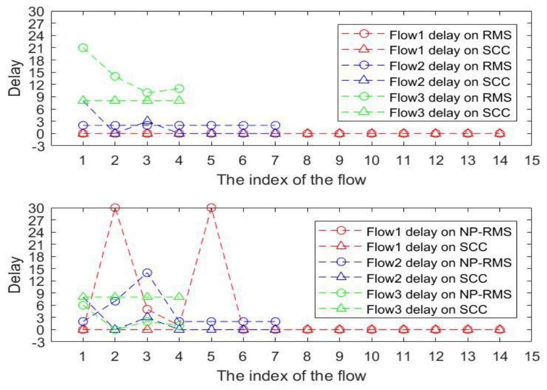

Fig. 6. Delay Comparison of SCC with RMS and NP-RMS

we propose the concept of delay rate, denoted by P. It is the percentage of the number of data flows that are delayed divided by the total number of data flows during a data flow task set period. Therefor, the delay rate is calculated as follows.

$$P = \frac{\sum_1^n f_i^{delay}}{\sum_1^n f_i} \times 100\%, \tag{8}$$

where f_i^{delay} the number of delays of the data stream $Flow_i$.

We randomly generated 100 sets of data, and their transmission efficiency are gradually increased from 30% to 90%. The test results are shown in Fig. 7. We can see that with the increase of transmission efficiency, the delay rate of NP-RMS increase fast, however the delay rate of RMS and SCC increase more slowly. We also observe that SCC has a lower delay rate than RMS. That is, SCC has significant advantage in network transmission delay.

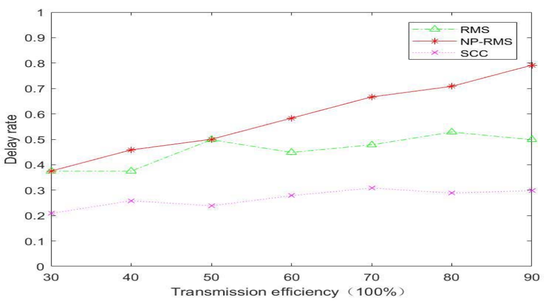

Fig. 7. Delay rate obtained by RMS, NP-RMS, SCC

In summary, compared with two existing scheduling models RMS and NP-RMS, our proposed SCC is high efficient in low latency, reliability, and time deterministy.

V. Conclusion

In this paper, we present a real-time data flow transfer model SCC based on the Flexilink protocol architecture to solve the problem of high latency, uncertainty in data transmission in network. Experimental results show that SCC performs real-time data stream transmission high efficiently in Flexilink architecture. Compared with two traditional priority transmission modes RMS and NP-RMS, SCC has lower delay, supports high load transmission, with small buffer period and no packet loss. Addtionally, due to the Flexilink architecture design, SCC can estimate the transmission time and delay of each data flow when it performs real-time data transmission, which is of great significance for some key time-deterministic data. To our best knowledge, this is a feature that other transport scheduling models do not have.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61602350, 61702382).

References

[1] S. Tullimas, T. Nguyen, R. Edgecomb, and S.-c. Cheung, "Multimedia streaming using multiple tcp connections," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 4, no. 2, p. 12, 2008.

[2] R. Koodli and M. Puuskari, "Supporting packet-data qos in next generation cellular networks," *IEEE Communications Magazine*, vol. 39, no. 2, pp. 180–188, 2001.

[3] N. Laoutaris and I. Stavrakakis, "Adaptive playout strategies for packet video receivers with finite buffer capacity," in *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No. 01CH37240)*, vol. 3. IEEE, 2001, pp. 969–973.

[4] W. Tu and W. Jia, "Adaptive playback buffer for wireless streaming media," in *Proceedings. 2004 12th IEEE International Conference on Networks (ICON 2004)(IEEE Cat. No. 04EX955)*, vol. 1. IEEE, 2004, pp. 191–195.

[5] L. Dai and T. You, "Real-time streaming media playout algorithm based on rbf delay prediction," in *2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT)*. IEEE, 2010, pp. 757–761.

[6] 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.

[7] 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 *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 2017, pp. 23–30.

[8] T. Ma, Y. Wang, W. Hu, D. El-Banna, and K. Zhang, "Evaluation of flexilink as unified real-time protocol for industrial networks," in *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 2018, pp. 123–128.

[9] L. Sha, T. Abdelzaher, K.-E. Årzén, A. Cervin, T. Baker, A. Burns, G. Buttazzo, M. Caccamo, J. Lehoczky, and A. K. Mok, "Real time scheduling theory: A historical perspective," *Real-time systems*, vol. 28, no. 2-3, pp. 101–155, 2004.

[10] M. Nasri and B. B. Brandenburg, "Offline equivalence: A non-preemptive scheduling technique for resource-constrained embedded real-time systems (outstanding paper)," in *2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, 2017, pp. 75–86.

[11] M. Park, "Non-preemptive fixed priority scheduling of hard real-time periodic tasks," in *International Conference on Computational Science*. Springer, 2007, pp. 881–888.