A method for improving real-time communication of switched Ethernet

Jun Zhao*, Zhong Ma, Xiangjun Wu

Wuhan Digital Engineering Institute, No.718, Luoyu Road, Wuhan, China, 430074

Received 12 June 2014, www.tsi.lv

Abstract

A method has been proposed for improving real-time communication of Switched Ethernet. Based on virtual link ideas, this method offline plans the whole network traffic under traditional Switched Ethernet hardware conditions, improves network terminal TCP/IP protocol by adding real-time communication interface, traffic shaping and priority queuing etc., and uses IEEE802.1p protocol on the switches of communication path. And it employs the network calculus theory to deduce the equation of calculating maximum end-to-end delay of real-time traffic. Meanwhile, it receives a simulation test of OPNET software. Both theoretical calculation and simulation results show that this method can effectively improve real-time communication of Switched Ethernet.

Keywords: Switched Ethernet, virtual link, real-time, traffic shaping, network calculus

1 Introduction

There has been increasingly extensive application of Switched Ethernet in aerospace, shipping, factory control and many other fields in recent years with its real-time performance as research focus. Although Avionics Full Duplex Switched Ethernet (AFDX) [1] can guarantee real-time communication of key data, it requires a particular network switch and interface card (NIC). In addition, traditional best-effort traffic is not compatible. Although Time-Triggered Ethernet (TTE) [2] can integrate both real-time traffic and best-effort traffic, it also needs a particular switch and demands time synchronization of the whole network. As a master-slave synchronization network, Flexible Time-Triggered Ethernet (FTT-SE) [3] regulates network traffic on the basis of the time division multiple access (TDMA) principle, but it also calls for a separate controller. To improve real-time communication of key information under traditional Switched Ethernet hardware conditions, and make best-effort traffic compatible, this paper proposes a method of improving real-time communication of Switched Ethernet. The first part will elaborate design ideas of this method; the second will employ the network calculus [4-5] theory to deduce the equation of calculating maximum end-to-end delay of real-time traffic; the third part will prove the effectiveness of this method through conducting a simulation test of OPNET software; the final part will draw a conclusion.

2 Method design

Based on virtual link [1] ideas, this method categorizes whole real-time traffic and best-effort traffic into different virtual links and arranges bandwidth accordingly, then it offline plans the whole network traffic to avoid traffic overload and network congestion. To guarantee effective communication as planned above, it adds real-time communication interface, traffic shaping and priority queuing to the network terminal TCP/IP protocol, and adopts IEEE802.1p standardized as two kinds of network traffic priority assignment on network terminals and switches. Here are some definitions of this method:

Definition 1: suppose that the whole network virtual link set is $VL = \{vl_1, vl_2, \ldots, vl_n\}$, the virtual link is $vl_i = (sma_i, dma_i, pr_i)$ in which i is the ordinal of vl_i; sma_i is the source MAC address of vl_i; dma_i is the destination MAC address of vl_i; pr_i is the priority $\in \{\mathrm{p}_{rt}, \mathrm{p}_{be}\}$ of vl_i, p_{rt} is the real-time traffic priority and p_{be} is the best-effort traffic priority and $\mathrm{p}_{rt} > \mathrm{p}_{be}$.

Definition 2: suppose the whole network terminal set is $ND = \{nd_1, nd_2, \cdots, nd_w\}$, and terminals are $nd_j = (nsma_j, rbma_j, rmma_j, svl_j, rvl_j,$ $LB_j\{LB_j^1, LB_j^2, \cdots, LB_j^{jm}\}, SQ_j^{rt}, SQ_j^{be}, RQ_j^{rt}, RQ_j^{be}$, (C_j^{send}, C_j^{recv}), in which j is the terminal number of nd_j; $nsma_j$ is the source MAC address of nd_j; $rbma_j$ is the MAC address to receive broadcast of nd_j; $rmma_j$ is the MAC address to receive multicast of nd_j;

* ***Corresponding author*** e-mail: 57278201@qq.com

$svl_j = \{vl_i \mid sma_i = nsma_j\}$ refers to the virtual link set sent by nd_j; $rvl_j = \{vl_i \mid dma_i = nsma_j \,||\, dma_i = rbma_j \,||\, dma_i = rmma_j\}$ refers to the virtual link set received by nd_j; $LB_j = \{LB_j^k, k \in svl_j\}$ stands for the leaky bucket regulator set of nd_j, which is responsible for shaping the traffic which has been sent to the virtual link. $LB_j^k = (lbt_j^k, lbr_j^k, lbb_j^k)$ refers to the leaky bucket regulator of vl_k on nd_j; lbt_j^k is the transmission cycle, lbr_j^k is the arranged bandwidth and lbb_j^k is the maximum packet length; SQ_j^{rt} is the queue of sending real-time traffic of nd_j; SQ_j^{be} is the queue of sending best-effort traffic of nd_j; RQ_j^{rt} is the queue of receiving real-time traffic of nd_j; RQ_j^{be} is the queue of receiving best-effort traffic of nd_j; C_j^{send} is the bandwidth of sending traffic of nd_j and C_j^{recv} is the bandwidth of receiving traffic of nd_j.

Definition 3: suppose that the virtual link configuration table offline is $VLTable$, then network planners use it to arrange bandwidth for the whole network virtual link to avoid traffic overload and ensure correct and real-time communication.

2.1 IMPROVED MODEL OF NETWORK TERMINAL TRANSMISSION

Compared with the standard TCP/IP protocol, there are two improvements on both the upper layer and the lower layer of the transmission model. First, add real-time traffic transmission interfaces to the application layer to isolate real-time traffic and best-effort traffic, then encapsulate real-time traffic into standard UDP packets for identification and forwarding of the commercial Ethernet switches, finally buffer real-time traffic according to virtual like ideas. Second, add classifiers, leaky bucket regulators [6] and traffic schedulers to the data link layer. To begin with, data stream is fist classified and labelled by classifiers, then respective virtual links are calculated and saved into according LB_j^k; next, traffic shaping is carried out by leaky bucket regulators through the bucket algorithm, and data stream is forwarded to SQ_j^{rt}, SQ_j^{be}; finally, data stream in SQ_j^{rt}, SQ_j^{be} is scheduled through the strict priority queue scheduling algorithm. The improved model of network terminal transmission is illustrated in detail in Figure 1.

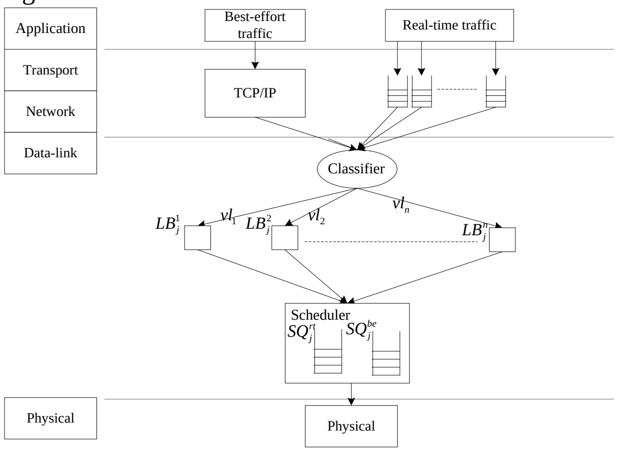

FIGURE 1 The improved model of transmission

Real-time traffic and best-effort traffic are classified according to the Algorithm 1 as follows:

Algorithm1. Packets are classified before transmission

Input: Send the packet $SendPack$, LB_j before classification

Output: LB_j after transmission

Algorithm steps:

1) Calculate vl_k it belongs to according to the source MAC address and destination MAC address of $SendPack$, and determine LB_j^k, which must be saved.

2) If $SendPack$ is sent by real-time traffic protocol, its $priorityType$ will be labelled as p_{rt} according to IEEE802.1p; otherwise, if it is sent by TCP/IP protocol, its $priorityType$ will be labelled as p_{be} instead.

3) Save $SendPack$ into LB_j^k.

After classification, the traffic in LB_j will be shaped by leaky bucket regulators through the bucket algorithm. Then, LB_j^k will arrive at the curve [4] $\alpha_j^k = lbt_j^k t + lbb_j^k$. To avoid overload of sending traffic, $VLTable$ should meet the requirements of the following formula.

$$\forall nd_j, \sum_{k \in svl_j} lbr_j^k \le C_j^{send}. \quad (1)$$

To avoid overload of receiving traffic, $VLTable$ should meet the requirements of the following formula.

$$\forall nd_j, \sum_{x \in ND, x \ne j} \sum_{k \in rvl_j} lbr_x^k \le C_j^{recv}. \quad (2)$$

When virtual link bandwidth is offline arranged, the above two formulas should be abided by. After traffic shaping is realized, data is forwarded into SQ_j^{rt} and SQ_j^{be}. Traffic in SQ_j^{rt}, SQ_j^{be} is scheduled through the strict priority queue scheduling algorithm and is then sent to the physical layer.

2.2 THE IMPROVED MODEL OF NETWORK TERMINAL RECEIVER

TCP/IP protocol, the relative standard of network terminal receiver, consists of improvements on two parts from the upper layer to the lower layer: 1) add dedicated interface to receive real-time traffic at the application layer and isolate the best-effort traffic from real-time traffic with the real-time traffic of each virtual link buffering respectively. 2) add classifier and traffic scheduler at the data link layer. Data traffic firstly passes through the classifier for classification. Then the scheduler uses the strict priority queuing scheduling algorithm for traffic scheduling. See algorithm 2 and algorithm 3 for the specific. The improved model of network terminal receiver is shown in figure 2.

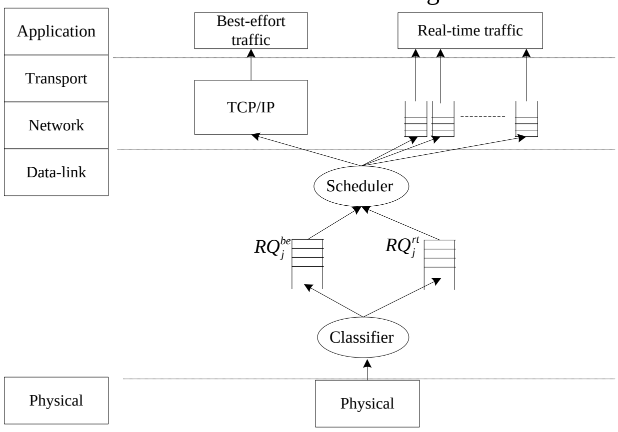

FIGURE 2 The improved model of receiver.

Algorithm 2: Classify and receive messages

Input: Receive the message $\mathrm{Re}\,cvPack$.

Output: RQ_j^{rt}, RQ_j^{be}.

Algorithm steps:

1) Analyse the received message $\mathrm{Re}\,cvPack$ according to the IEEE802.1p protocol and determine the packet priority $priorityType$.

2) When $priorityType$ is p_{rt}, if RQ_j^{rt} is non-null, store $\mathrm{Re}\,cvPack$ at the queue tail of RQ_j^{rt}, Otherwise, abandon $\mathrm{Re}\,cvPack$.

3) When $priorityType$ is p_{be}, if RQ_j^{be} is non-null, store $\mathrm{Re}\,cvPack$ at the queue tail of RQ_j^{be}, Otherwise, abandon $\mathrm{Re}\,cvPack$.

Algorithm 3: The scheduling mainly receives queue packet.

Input: RQ_j^{rt}, RQ_j^{be}

Output: Scheduling packet

Algorithm steps:

1) If RQ_j^{rt} is non-null, determine vl_k, the message belonging, based on source MAC address and destination MAC address of the RQ_j^{rt} head message and store it in corresponding real-time traffic buffer.

2) If RQ_j^{be} is non-null, store the RQ_j^{be} head message in the TCP/IP protocol buffer.

The switch uses the priority queuing scheduling based on IEEE802.1p.

3 The calculation of end-to-end delay upper bound of real-time traffic

The calculation formula of the maximum end-to-end delay of real-time traffic used in this approach can be deduced according to the network calculus theory. The end-to-end delay of real-time traffic of vl_j is the sum of delays of network transmitting terminal, link transmission, switch and network receiving terminal.

The switch delay $D_{sw}^{rt,j}$ consists of the technical delay $tD_{sw}^{rt,j}$ and the queuing delay $\mathrm{q}D_{sw}^{rt,j}$. With respect to specific hardware, $tD_{sw}^{rt,j}$ is a bounded constant and $\mathrm{q}D_{sw}^{rt,j}$ relies on the queuing strategy. Suppose that $RTVL_{sw_i}^{x}$ is the real-time traffic concentration at the output port x of the switch sw_i and the arrival curve of real-time traffic of vl_j is $\alpha_j^{rt} = r_j^{rt}t + b_j^{rt}$. Then the arrival curve of real-time data streams is $\alpha^{rt} = \sum_{j \in RTVL_{sw_i}^{x}} \alpha_j^{rt}$. Suppose $l_{\max}^{be}$ is the maximum length of Ethernet data frame of the best-effort traffic and $C_{sw_i}^{x}$ is the total service bandwidth of the output port x of the switch sw_i. Then the total service curve of the output port x of the switch sw_i is $\beta_{sw_i}^{x}(t) = C_{sw_i}^{x}\left[t-0\right]^{+}$. According to the conclusion 1 of corollary 6.2.1 from literature [7], we can get the service curve of real-time traffic provided by the output port x of the switch sw_i:

$$\beta_{sw_i}^{x,rt}\ t = \left[\beta_{sw_i}^{x}\ t - l_{\max}^{be}\right]^{+} = C_{sw_i}^{x}\left[t - \frac{l_{\max}^{be}}{C_{sw_i}^{x}}\right]^{+}. \quad (3)$$

Then the service rate and service delay of real-time traffic at the output port x of the switch sw_i are as follows:

$$R_{sw_i}^{x,rt} = C_{sw_i}^{x}, \quad T_{sw_i}^{x,rt} = \frac{l_{\max}^{be}}{R_{sw_i}^{x,rt}}. \quad (4)$$

Suppose the maximum data backlog of transmission time is $\sum_{j \in RTVL_{sw_i}^{x}, j \neq k} b_j^{rt}$ and the maximum frame length of real-time traffic of vl_j is $l_{max}^{rt,j}$. According to the corollary 6.2.1 from literature [7], then the service rate and service delay of real-time traffic of sw_i at the output port x of the switch sw_i are as follows:

$$R_{sw_i}^{x,rt,j} = R_{sw_i}^{x,rt} - \sum_{k \in RTVL_{sw_i}^{x}, k \neq j} r_k^{rt}, \quad T_{sw_i}^{x,rt,j} = T_{sw_i}^{x,rt} + \frac{\left(\sum_{k \in RTVL_{sw_i}^{x}, k \neq j} b_k^{rt} + l_{max}^{rt,j}\right)}{R_{sw_i}^{x,rt}}. \quad (5)$$

Suppose SW is the collection of all switches that the virtual link vl_j passes through. Then the service rate and service delay of real-time traffic of vl_j throughout the cascaded network system are as follows:

$$R_{sw}^{rt,j} = \min(R_{sw_i}^{x,rt,j}), \quad T_{sw}^{rt,j} = \sum_{sw_i \in SW} T_{sw_i}^{x,rt,j} \quad (6)$$

The switch queuing delay upper bound of real-time traffic of vl_j is:

$$qD_{sw}^{rt,j} = T_{sw}^{rt,j} + \frac{b_j^{rt}}{R_{sw}^{rt,j}} \quad (7)$$

Then the switch delay upper bound $D_{sw}^{rt,j}$ of real-time traffic of vl_j is:

$$D_{sw}^{rt,j} = \sum_{sw_i \in SW} tD_{sw_i}^{rt,j} + qD_{sw}^{rt,j}. \quad (8)$$

Assume that $maxl_{src}^{be}$ and $maxl_{src}^{rt,j}$ represent the maximum frame length of best-effort traffic of the network transmitting terminal src and the maximum frame length of real-time traffic of vl_j respectively. The calculation of network transmitting terminal delay upper bound $D_{src}^{rt,j}$ of real-time traffic of vl_j resembles that of switch delay upper bound. The specific formula is as follows:

$$R_{src}^{rt} = C_{src}^{send}, \quad T_{src}^{rt} = \frac{maxl_{src}^{be}}{R_{src}^{rt}}, \quad R_{src}^{rt,j} = R_{src}^{rt} - \sum_{k \in svl_{src}, pr_k = p_{rt}, k \neq j} r_k^{rt}, \quad T_{src}^{rt,j} = T_{src}^{rt} + \frac{\left(\sum_{k \in svl_{src}, pr_k = p_{rt}, k \neq j} b_k^{rt} + maxl_{src}^{rt,j}\right)}{R_{src}^{rt}}, \quad D_{src}^{rt,j} = T_{src}^{rt,j} + \frac{b_j^{rt}}{R_{src}^{rt,j}}. \quad (9)$$

Suppose that real-time traffic of vl_j passes a total of n_l physical links whose available bandwidth is $\{plb_1, plb_2, \ldots, plb_{n_l}\}$ and that the frame length is f_j. Then the link transmission delay $D_{link}^{rt,j}$ of real-time traffic of vl_j is:

$$D_{link}^{rt,j} = \sum_{i=1}^{n_l}\left(\frac{f_j}{plb_i}\right). \quad (10)$$

Assume that the receiving network terminal delay of real-time traffic of vl_j is $D_{des}^{rt,j}$. Then the maximum end-to-end delay $D^{rt,j}$ of real-time traffic of vl_j is:

$$D^{rt,j} = D_{src}^{rt,j} + D_{link}^{rt,j} + D_{sw}^{rt,j} + D_{des}^{rt,j}. \quad (11)$$

Therefore, the maximum end-to-end delay of real-time traffic of vl_j is calculable and the real-time communication under that network bears certainty.

4 The simulation result and analysis

We use the OPNET software to conduct simulation experiments on the method proposed in this paper to verify its instantaneity. The whole simulation network topology is a tree structure. The specific is shown in figure 3. Three switches initiate the IEEE802.1p protocol. The network node includes the client node and the server node which are improved specifically using the method put forward in the first section. The client node which ranges from 1 to 30 is linked with the switch on the second layer while the only one server node is connected with the switch on the first layer. Each client node sends on-demand messages comprised of real-time traffic and best-effort traffic to the server node. Two transmitting modules are established at the client node and two receiving modules established at the server node to distinguish between real-time traffic and best-effort traffic.

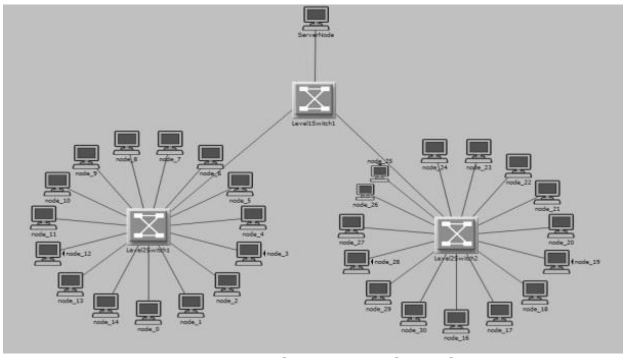

FIGURE 3 Simulation network topology

The flow simulation of client node employs the ON/OFF model. Specific parameters are shown in table 1.

TABLE 1 The parameters of flow simulation

Parameter	Real-time traffic	Best-effort traffic
Traffic priority	7	0
Start time (s)	constant(0)	constant(0)
ON state time (s)	constant(100)	constant(100)
OFF state time(s)	constant(0)	constant(0)
Packet interval time (s)	exponential(0.00038)	exponential(0.00704)
Packet size (bytes)	constant(46)	constant(1500)
Leaky bucket time interval (s)	constant(0.00038)	constant(0.00704)

In experiments, we used the present Ethernet model and this paper's improved model to conduct simulation experiments with the client ranging from 1 to 30. Figure 4 and 5 present respectively the comparison of average end-to-end delays of real-time traffic, and the comparison of maximum end-to-end delays of real-time traffic when the number of clients ranges from 1 to 26. When the number of clients extends from 27 to 30, the difference between the traditional Ethernet model and the improved Ethernet model is larger.

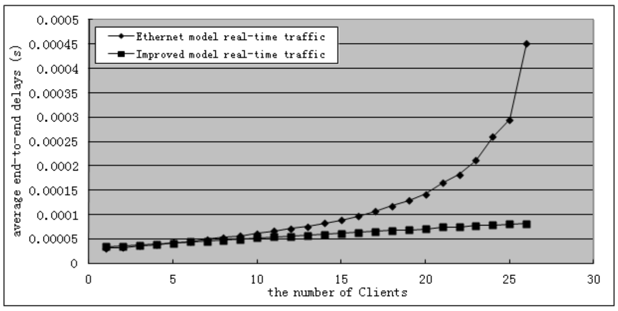

FIGURE 4 Comparison of average end-to-end delays of real-time traffic

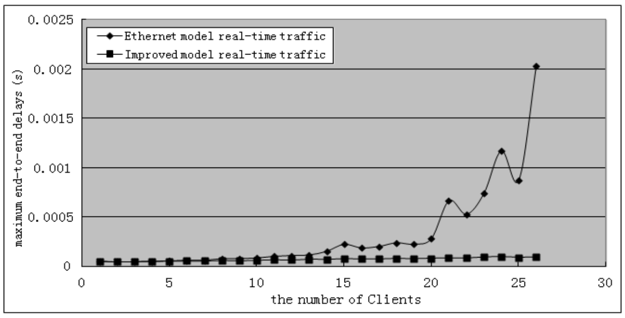

FIGURE 5 Comparison of maximum end-to-end delays of real-time traffic

From figure 4 and 5, we can see that the method proposed in this paper can improve real-time traffic performance over Switched Ethernet and at the same time be compatible with the best-effort traffic.

5 Conclusions

This paper firstly put forward a method for improving real-time communication over Switched Ethernet. Then it employed the network calculus theory to deduce the calculation formula of the maximum end-to-end delay of real-time traffic under this method. Ultimately, it used the OPNET software to conduct simulation experiments. Theoretical calculation and simulation results both show that this method can effectively improve real-time communication over Switched Ethernet. Further research on the traffic scheduling algorithm of network terminal is needed in future.

Acknowledgment

This work was financially supported by Wuhan Digital Engineering Institute, China.

References

[1] ARINC. "Avionics Full Duplex Switched Ethernet (AFDX) Network" ARINC SPECIFICATION 664P7. 2005 June
[2] SAE. "Time-Triggered Ethernet" AS6802 2011 Nov
[3] Marau R, Almeida L, Pedreiras P 2006 Enhancing Real-Time Communication over COTS Ethernet switches *IEEE International Workshop on Factory Communication Systems Torino* 295-302
[4] Cruz R L 1991 *IEEE Transactions on Information Theory* **37**(1) 114-31

[5] Cruz R L 1991 *IEEE Transactions on Information Theory* **37**(1) 132-41

[6] Tranter W, Taylor D, Ziemer R 1993 A generalized processor sharing Approach to flow control in integrated services networks: in the single node case *IEEE Network* **1**(3) 344-57

[7] Boudec J-Y L, Thiran P 2004 *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet* Berlin: Springer Verlag

Authors

Jun Zhao, born in October, 1980, Shulan, Jilin Province, China

Current position, grades: He is now a PhD candidate in Wuhan Digital Engineering Institute.
University Studies: He received a master in computer architecture from Wuhan Digital Engineering Institute, China.
Scientific interest: network architecture theory and cloud computing.
Publications: more than 10 articles.
Experience: He had been a researcher and an engineer from 2002 to 2014 at Wuhan Digital Engineering Research Institute in China.

Zhong Ma, born in January, 1962, Wuhan, Hubei Province, China

Current position, grades: A senior engineer of Wuhan Digital Engineering Research Institute.
University Studies: He received a master in computer architecture from Wuhan Digital Engineering Institute, China.
Scientific interest: network architecture theory and computer architecture.
Publications: more than 30 articles.
Experience: He had been a researcher and an engineer from 1991 to 2014 at Wuhan Digital Engineering Research Institute in China.

Wu Xiangjun, born in May, 1971, Wuhan, Hubei Province, China

Current position, grades: A senior engineer of Wuhan Digital Engineering Research Institute.
University Studies: He received a Ph.D. in Information and Communication Engineering from Huazhong University of Science and Technology, China.
Scientific interest: network architecture theory, network system design and integrating.
Publications: more than 20 articles.
Experience: He had been a researcher and an engineer from 2000 to 2014 at Wuhan Digital Engineering Research Institute in China.