CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2016; **00**:1–22
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cpe

Efficient QoS-aware Packet Chunking Scheme for M2M Cloud Services

Anan Sawabe[1,2], Kazuya Tsukamoto[2] and Yuji Oie[2]

[1]*System Platform Research Laboratories, NEC Corporation.*
[2]*Department of Computer Science and Electronics, Kyushu Institute of Technology.*

SUMMARY

With the recent advances in machine-to-machine(M2M) communications, huge numbers of devices have become connected and massive amounts of traffic are exchanged. M2M applications typically generate small packets, which can profoundly affect the network performance. Namely, even if the packet arrival rate at the router is lower than the link bandwidth, bits per second(BPS), it can exceed the router forwarding capacity, which indicates the maximum number of forwarded packets per second(PPS). This will cause the decrease in the network throughput. Therefore, eliminating the PPS limitation by chunking small packets will enable M2M cloud services to spread further. This paper proposes new packet-chunking schemes aimed at meeting both application requirements and improving achievable router throughput. In our schemes, multiple buffers, each of which accommodates packets classified based on their delay requirement, are installed in parallel. Herein, we report on analysis of the theoretically performance of these schemes, which enabled us to derive some important features. We also propose a scheme whereby a single chunking buffer and parallel multiple buffers were arranged in tandem. Through our simulation and numerical results, we determined that these schemes provide excellent performance in reducing the number of outgoing packets from the router while meeting various delay requirements.

Received . . .

KEY WORDS: M2M cloud services; Packet chunking; QoS application requirement; Buffer management.

1. INTRODUCTION

Numerous machine-to-machine (M2M) applications are currently emerging and some reports estimate that as many as 50 billion M2M devices will be interconnected by 2020 [1]. In M2M communication systems, a huge number of interconnected devices are embedded in environmental observation systems, manufacturing systems, automobiles, and so on, where they exchange enormous amounts of data on wireless and wired networks [2][3]. Such heterogeneous data, known as “big data”, are collected, stored, and analyzed as part of efforts to explore new approaches that could solve a wide spectrum of issues [4][5].

The Internet needs the ability to efficiently handle these rapidly increasing data traffic flows while simultaneously accommodating the ever-increasing mobile traffic generated from smartphones and other devices. Looking to facilitate this, we begin by focusing on Internet architecture. The Internet is mainly composed of two types of networks: (1) access networks to which an end-user first connects, and (2) core networks connecting access networks. Core networks are very high-speed

*Correspondence to: System Platform Research Laboratories, NEC Corporation, Kawasaki, Kanagawa, 211-8666, Japan.
E-mail: a-sawabe@cb.jp.nec.com
†This project is supported by the Ministry of Internal Affairs and Communications of Japan.

Prepared using cpeauth.cls *[Version: 2010/05/13 v3.00]*

networks conveying huge numbers of relatively large packets generated from traditional services such as file transfer and video streaming [6]. Achievable router throughput is usually measured in terms of bits per second (BPS). However, since the maximum number of packets a router can handle is limited, another measure, packets per second (PPS), needs to be considered as well.

Generally speaking, M2M applications generate very small packets [7][8][9]. However, when vast numbers of small packets are generated from a huge number of M2M devices, the packet arrival rate at access networks routers can exceed their PPS number, which means the router will be unable to forward all of the arriving packets. This further results in low BPS throughput [10]. For example, when routers can transfer packets at 10 Gbps and handle 5M PPS , where all packets are 40 bytes in length, their throughput is just 1.6 Gbps (5M packets of 40 bytes per second). In fact, some network careers (such as KDDI and DoCoMo) have reported encountered problems dealing with the small signaling packets periodically generated from large numbers of mobile devices [11][12][13][14]. Since core routers handle incoming packets from a large number of access routers, the same issue will eventually impact them as well.

One straightforward solution to the issue would be to replace currently deployed routers with high-speed models capable of handling larger PPS numbers, while an alternative edge router-based solution would be to provide routers with a function for decreasing the number of outgoing packets by chunking several packets. However, since comparing these two solutions, in terms of their implementation complexity and implementation/deployment cost, is beyond the scope of this paper, we will instead focus on edge router-based solutions and several effective methods of chunking packets.

In this paper, we propose several packet-chunking schemes aimed at decreasing the number of outgoing packets from edge routers while simultaneously meeting several different quality of service (QoS) application requirements. These schemes force incoming packets to wait in the buffer until a set amount of packets are stored, after which they are bound together into large packets, such as the maximum jumbo frame size [15], that are designed to satisfy their diverse requirements, such as acceptable delay. The performance of these schemes will be examined through analysis and simulation experiments in order to determine how efficiently they chunk packets while meeting the acceptable delay requirements under a wide range of practical scenarios. In addition, several cases of traffic imbalance and traffic overload are examined.

The remainder of this paper is organized as follows. Section 2 describes related work, and the network model is described in Section 3. Our new packet chunking schemes are discussed in Section 4, and the router and traffic models employed here are described in Section 5. The analytical approach provided to a part of our proposed schemes is discussed in Section 6, while Section 7 discusses the numerical and simulation results. Finally, we conclude this paper in Section 8.

2. RELATED WORK

As stated in the Introduction, the packet chunking schemes proposed in this paper attempt to reliably satisfy QoS application requirements. Chunk packets are composed of multiple packets picked up from multiple buffers that have been established with different priorities. The following two fundamental technologies are essential for achieving these proposed schemes: (1) packet chunking and (2) multiple buffer management. Accordingly, we will begin by surveying the existing studies focusing on these two technologies.

Numerous existing studies have focused on packet chunking technology. For example, several papers including [16][17] have proposed layer 2 frame aggregation schemes to decrease the signaling overhead over wireless local area networks (WLANs) in order to improve throughput. Reference [18] applied data aggregation technology to improve the throughput performance over multi-hop wireless networks, while reference [19] focused on sensor networks and proposed a data aggregation scheme to reduce energy consumption by reducing the amount of transmitted data in the sensor network. However, since these methods do not consider the concatenation of packets obtained from multiple buffers with different priorities, they do not directly relate to our study.

DOI: 10.1002/cpe

Meanwhile, numerous active queue management technologies have been studied, some of which have already been implemented in actual products. Recently, [20] conducted a wide survey that classified these technologies from a bird's-eye view. However, since these studies do not focus on the performance degradation caused by exceeding the maximum PPS numbers that core routers can handle, and since they do not discuss the combined method of the packet chunking and multiple buffer management at all, they are essentially irrelevant to our present paper.

On the other hand, several studies [21][22] have paid attention to optical burst switching (OBS) networks and have proposed several packet scheduling schemes in which multiple Internet protocol (IP) packets received from Ethernets are concatenated by following a number of predetermined rules, after which a resulting chunk packet is transmitted to the optical core network as a "burst". Since these schemes are able to concatenate multiple IP packets with different priorities into a "burst", they utilize the two abovementioned essential technologies (chunking packets while considering different priorities). That is, we can say that these studies are deeply related to our present study. Furthermore, from the viewpoint of triggers for packet concatenation, two different triggers are employed in parallel: transmission intervals (timer-based) and the length of "burst" (threshold-based). These triggers are also employed in our studies, as will be discussed later.

However, since these existing studies have focused on OBS networks, buffering (i.e., the ability to delay transmission timing) they cannot be utilized when packet transmission timing contention arises due to different priorities. As a result, low priority packets stored in a buffer may frequently be discarded. On the other hand, since our study assumes use of a conventional IP network, packet buffering can be guaranteed, even when the packet transmission timing contention exists among multiple buffers. Furthermore, the performance of packet loss ratio could be improved (i.e., packet losses decreased) and the delay performance may be minimized.

Next, we will review the existing packet chunking methods. References [21][22] proposed a naive packet chunking scheme that individually composes a "burst" based on multiple packets per buffer. Furthermore, Reference [22] proposed a particularly interesting enhanced packet chunking method that composes a "burst" based on packets buffered in all buffers, each of which have different priorities. These two chunking schemes are similar to our proposed method, which will be explained later. However, these papers evaluated end-to-end performance such as "burst" length, packet loss ratio, and transmission delay, through simulation experiments alone. In addition, References [23][24][25] evaluated the performance of the OBS network (from the ingress edge to the egress edge routers) in terms of packet loss ratio and network delay, using both a mathematical approach and simulation experiments. However, our major concern is how well the QoS requirement can be satisfied and how efficient packet chunking can be achieved. Unfortunately, these issues were not examined in the above references.

Taken as a whole, it is clear that none of existing studies evaluate both "the chunking performance" (the number of chunk packets and the size of each chunk packet) and "the QoS performance" (packet loss ratio and the ratio of packets violating the QoS requirement) at the ingress edge router, while simultaneously considering performance limitations caused by exceeding the maximum number of packets that core router can handle.

Our earlier work in reference [26], published in WAINA2014, dealt with a conventional IP network whose waiting mechanism in response to packet transmission contention was totally different from that of an optical network. Therein, the performance of our proposed scheme in terms of both packet chunking efficiency and the QoS satisfaction ratio (packet loss ratio and the ratio of packets violating the QoS requirement) was investigated through simulation experiment. However, the feasibilities of existing chunking schemes were not clarified in that work because the pattern of incoming traffic was assumed to be limited.

In this study, we have enhanced our earlier work in the following three aspects: (1) by executing performance evaluations in a wide range of traffic arrival patterns, such as homogeneous traffic, imbalanced traffic, and overload cases, not only via simulation experiment, but also using an analytical approach [†], (2) by proposing new chunking efficiency aware packet chunking schemes

[†]Portions of our proposed schemes are analyzed by a mathematical approach.

DOI: 10.1002/cpe

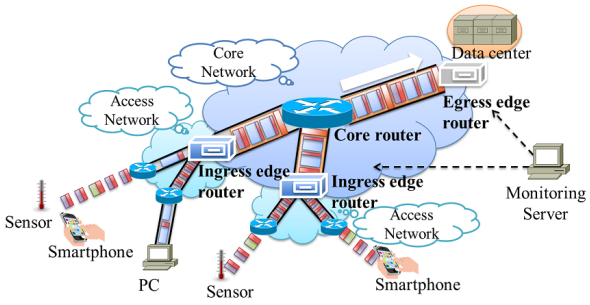


Figure 1. Target network model.

that improve the chunking efficiency, and (3) by investigating the impact of the number of buffers on scheme performance levels.

Consequently, in comparison to the abovementioned existing studies, the performance characteristics of our proposed chunking methods will be discussed exhaustively. Furthermore, a wide range of traffic arrival patterns are examined in this paper in order to evaluate the proposed schemes in terms of feasibility and effectiveness.

3. TARGET NETWORK MODEL

In this section, first we explain the target network model discussed in this paper, and then define the maximum chunking waiting time.

3.1. Types and roles of nodes

Here, we will focus on core networks. Currently, core networks generally consist a small number of core routers and relatively large numbers of edge routers. Core routers provide high-speed communications in the core network and edge routers connect the access network and the core network. Note that one core router is connected to multiple edge routers.

In this paper, we focus on the edge routers in the core network, which are equipped with a new "packet chunking/de-chunking" function that is designed to facilitate high throughput in the core network while meeting the packet delay requirements. As shown in Fig. 1, the ingress edge routers chunk small packets, while the egress edge routers de-chunk the received chunk packets and send them on to the cloud servers. Moreover, we prepared a monitoring server to detect abnormal events in the core network and to support the message exchange of network information, such as throughput and round trip time (RTT), between the ingress and the egress edges. The roles of the edge routers are as follows:

- Ingress edge router:
  Identifies incoming packets and grasps their QoS application requirements. The router then makes chunk packets based on the maximum transmission unit (MTU) and their QoS-related information.
- Egress edge router:
  Unpacks the received chunk packets and then forwards the unpacked packets to cloud servers (such as data centers) or the destination device.

3.2. Maximum waiting time for chunking

We begin by presuming that the application types of incoming traffic on an ingress edge router can be recognized in some way, such as via deep packet inspection (DPI), which identifies the application types of incoming packets by searching through packets [27]. Then, their application's

DOI: 10.1002/cpe

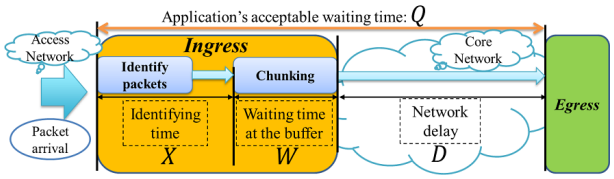


Figure 2. One-way delay between an ingress and an egress router.

acceptable waiting time (Q) can be obtained based on the identified information. Let us assume the time required for identifying the application types to be X and the maximum waiting time for chunking to be W. Furthermore, we assume that the ingress edge router can grasp the network delay (D) reported by the monitoring server. Note that, since this paper assumes that the network delay (D) is already given, the method of obtaining D is not discussed. From this, as shown in Fig. 2, one-way delay is calculated as follows:

$$One - way\, delay = X + W + D.$$

The application's maximum acceptable waiting time Q should be larger than the "one-way delay" in order to guarantee the application's QoS requirement is met. Therefore, the maximum amount of acceptable waiting time for chunking W_{max} is $W_{max} = Q - (X + D)$.

4. PACKET CHUNKING SCHEME

Next, we will discuss the "packet chunking" function added to the ingress edge routers. The purpose of packet chunking schemes is to make the best use of a core router's forwarding capacity while meeting the applications' QoS requirement for packets. We propose a scheme for the ingress edge router that buffers incoming packets and packs them into large-sized chunk packets. Note that, in this paper, we will limit our focus to packets destined for same egress edge router.

Large-sized chunk packets reduce the number of packets forwarded to the core network, and thus avoid exceeding the maximum PPS number the core routers can handle. This permits the core routers to provide their maximum achievable throughput in terms of BPS. Thus, in order to prevent both packet fragmentations and drops at the intermediate routers in the core network, the maximum packet size should be limited to the MTU size of the core routers. As a result, the MTU size of a jumbo frame (9000 bytes [28][29]) is employed as the maximum size of a chunk packet.

Under these assumptions, the straightforward scheme is as follows: an ingress edge router continues to buffer incoming packets until the queue length reaches the MTU size. The router then creates a chunk packet and transmits it. However, the following two problems may arise in terms of the QoS guarantee:

- Problem 1
  In the case of the lightly loaded traffic, the buffer waiting time for chunking increases and the waiting time can exceed W_{max}, defined previously.
- Problem 2
  In case of the heavily loaded traffic, the queue length can increase to the point where the waiting time also exceeds W_{max}.

In order to satisfy the maximum acceptable waiting time for chunking, we propose the concept of "residual waiting time", which is defined as the "(maximum waiting time (W_{max})) $-$ (the time that packets have already been held at the buffer)" for the priority control mechanism. Hereinafter, we will propose several chunking schemes based on residual waiting time.

DOI: 10.1002/cpe

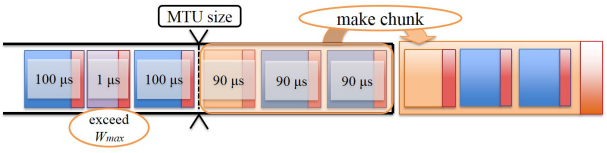


Figure 3. BasiCQ problem.

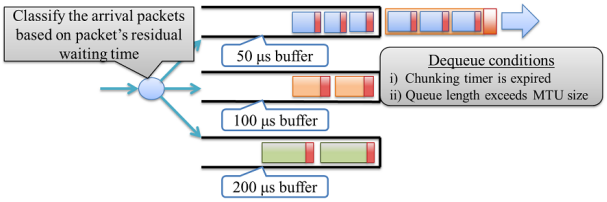


Figure 4. Queue management in ChuMQ ($N = 3$).

4.1. Basic Chunking scheme with single Queue (BasiCQ)

First of all, a straightforward chunking scheme consisting of a single buffer and chunking timer, which will be thus hereafter referred to as the basic chunking scheme with single queue (BasiCQ), will be addressed below.

- Enqueue method
  - Packets are queued in order of arrival.
  - The chunking timer is updated to the minimum packet residual waiting time every time a packet arrives.
- Dequeue method
  - If either of the following two conditions is satisfied, a chunk packet is created from the buffered packets.
    1. Chunking timer has expired: i.e., chunking timer expiration.
    2. Queue length exceeds the MTU size: i.e., MTU size excess.

BasiCQ can solve Problem 1, and can also be expected to achieve efficient chunking with the QoS guarantee under various moderate traffic conditions by using both the chunking timer expiration and the MTU size excess rules.

On the other hand, in actual environments, traffic fluctuates and is often imbalanced. In such cases, some specific packet classes may arrive very frequently or the packet arrival rate may exceed the router forwarding capacity for some period of time. In these cases, some additional mechanisms will be needed.

4.2. Chunking scheme with Multi-Queues (ChuMQ)

When the packet arrival rate exceeds the router forwarding capacity, the router cannot handle the QoS requirements of all arriving packets, as shown in Fig. 3. When this occurs, priority packets should be given the precedence, thus ensuring their QoS requirements are met, even under the stated condition.

Accordingly, a new chunking scheme, hereafter referred to as chunking scheme with multi-queues (ChuMQ), is proposed. In this scheme, the QoS requirements of some high-priority packet classes are given precedence, even in overloaded traffic conditions, by utilizing a priority queue management.

Thus, packets are classified into N classes, each of which is provided with a dedicated buffer and chunking timer. Specifically, N buffers are equipped in parallel and N chunking timers are used in

DOI: 10.1002/cpe

total. Again, the cost of implementing N buffers to the ingress edge routers is considered out of the scope of this paper and will not be addressed.

Fig. 4 shows a system overview of ChuMQ and describes queue management when $N = 3$. The maximum chunking timer for each buffer is predefined as $T_1, T_2, \cdots, T_N$: T_1 =50 μs, T_2 =100 μs, and T_3 =200 μs in Fig. 4.

That is, this system cannot guarantee a delay requirement of less than T_1. Although the timer value depends significantly on both the class and the QoS application requirements, determining the timer value remains a subject of our future work.

Next, the enqueue process classifies the incoming packets according to the rules provided below. Note that packet classification should be executed on the safe side (i.e., the packet's residual waiting time should always be greater than or equal to the queue's chunking timer) in order to ensure the application's QoS requirement is met.

- For $1 \leq i < N$, if the residual waiting time is less than T_{i+1} and greater than or equal to T_i, ith buffer is selected.
- If the residual waiting time is greater than or equal to T_N, Nth buffer is selected.

Details of the queue management procedures for ChuMQ are described below:

- Enqueue method
  - Multiple buffers with different chunking timers are placed in parallel. Then, each arriving packet is classified and stored in one of the buffers based on both the packet's residual waiting time and the buffer's chunking timer.
  - The chunking timer is initiated whenever the queue length is increased from zero to one, i.e., when a packet newly arrives at the empty buffer.
- Dequeue method
  - The dequeue conditions are the same as those used in BasiCQ. In addition, since the first buffer (which has the smallest chunking timer value) has the highest priority and is checked first among all the buffers, the ith buffer is first served when ith and jth ($j > i$) buffers expire simultaneously.

ChuMQ is expected to provide priority packets with better performance even in heavily loaded traffic cases. Nonetheless, the following features will be discussed later.

1. Chunk packet transmission contention.
   Chunk packet transmission contention occurs when a chunking timer expires while a chunk packet from other buffers are being transmitted. When this occurs, the chunk packet is forced to wait until the end of ongoing transmission, which means that its delay requirement is violated.
2. Inefficiency in chunking.
   In imbalanced traffic conditions, some lightly loaded queues cannot reach MTU length. This will lead to the creation of small chunk packets (less than the MTU size).

4.3. Chunking scheme with Tandem-Queue (ChuTQ)

As part of efforts to solve the problems arising in ChuMQ, we propose an additional new chunking scheme, hereafter referred to as chunking scheme with tandem-queue (ChuTQ). In this scheme, we place a single buffer (called the chunking buffer) after N parallel buffers in order to eliminate chunk packet transmission contention. In other words, a tandem queue system, shown in Fig. 5, is introduced.

ChuTQ checks head-of-line (HoL) packets in all N buffers, and then selects one based on one of the policies that will be discussed later. The chosen packet is forwarded to the chunking buffer, whose chunking timer has been updated to the minimum residual waiting time of all buffered packets. This single chunking buffer aggregates the packets from all the class, thereby improving chunking packet efficiency as well.

DOI: 10.1002/cpe

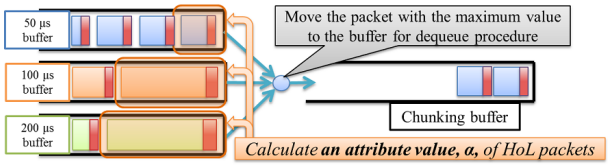


Figure 5. Queue management in ChuTQ ($N = 3$).

There are numerous possible policies that can be used when selecting a packet from among the HoL ones in the N buffers for the chunking buffer. Here, we will examine the following three policies: delay-constraint-first (DCF), throughput-performance-first (TPF), and chunking-efficiency-first (CEF). In addition, we will introduce attribute values that can be attached to each of the packets in order to implement the abovementioned policies. Since defining an attribute value that achieves high performance while meeting the delay constraints is a major concern, the attribute value, α, is defined in each of the following policies:

- **Delay-constraint-first policy (DCF)**
  - Prioritizes packets with short residual waiting times
  - $\alpha = \frac{1}{(residual\ waiting\ time)}$
- **Throughput-performance-first policy (TPF)**
  - Prioritizes large packets while meeting their delay constraints
  - $\alpha = \frac{1}{(residual\ waiting\ time)} \times \frac{(packet\ size)}{(MTU\ size)}$
- **Chunking-efficiency-first policy (CEF)**
  - Prioritizes small packets while meeting their delay constraints
  - $\alpha = \frac{1}{(residual\ waiting\ time)} \times \frac{1}{(packet\ size)}$

Detailed queue management procedures based on the attribute value are described below:

- Enqueue method
  - Same as in ChuMQ
- Dequeue method
  - An attribute value of the HoL packets in each buffer is calculated in the same manner mentioned earlier, and the packet with the maximum value is moved to the chunking buffer for the dequeue procedure.
  - When the chunking buffer receives new packets, the chunking timer is updated to the minimum residual waiting time of all packets in the chunking buffer.
  - The chunk packet is transmitted when the same condition is satisfied as in BasiCQ and ChuMQ; i.e., chunking timer expiration or MTU size excess.

4.4. Chunking efficiency aware scheme in ChuMQ and ChuTQ

Up to this point, we have discussed packet-chunking schemes that both meet the delay requirement and improve the router's achievable throughput. In this section, we will discuss how to further improve ChuMQ and ChuTQ in terms of chunking efficiency.

First, in ChuMQ, we will consider a case in which all the packets stored in a buffer are stuffed in a chunk packet after the chunking timer expires, even if the size of chunk packet does not reach the MTU size. In that condition, the chunking efficiency aware (CEA) scheme adds more packets from other buffers until the chunk packet size exceeds the MTU size, or until all of the buffers have been emptied. When this scheme is implemented, the buffer with the shortest chunking timer is first chosen. This scheme is referred to as ChuMQ + CEA.

DOI: 10.1002/cpe

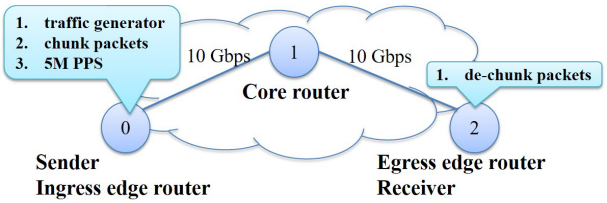


Figure 6. Network model.

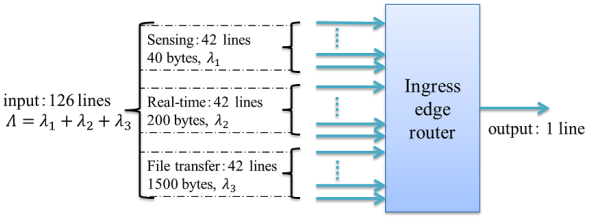


Figure 7. Ingress edge router hardware component.

In contrast, since the ChuTQ scheme creates chunk packets at one chunking buffer, it can provide good chunking efficiency, even under light traffic conditions, at the cost of a slight decrease in chunking efficiency due to the inefficient packet concatenation mechanism, especially under heavy traffic conditions. This is because if the chunk packet size exceeds the MTU size due to some packet, ChuTQ cannot flexibly substitute that packet with a different packet from one of the other buffers. The CEA scheme flexibly adds more packets with short delay requirements as long as the size of chunk packet is less than the MTU size.

5. SYSTEM MODEL FOR PERFORMANCE EVALUATION

In this section, we describe the system model (see Fig. 6) used to evaluate the effectiveness of our proposed schemes. This model includes the network model and the traffic model. In addition, important performance measures obtained in the model will be given. Note that Table I shows the notation used in following sections.

ChuMQ is a mathematically tractable scheme, and will be analyzed in Section 6. The evaluation of other schemes relies on the Network Simulator-3 (NS-3) simulation used [30].

5.1. Network and router model

In our model, the network topology consists of three nodes: node-0 as the sender/ingress edge router, node-1 as the core router, and node-2 as the egress edge router/receiver.

We assume that the packet handling capability of the core router is set to 250M PPS by referring to the value of the "Cisco Catalyst 4500" [31]. Generally, one core router is connected to a relatively large number of the ingress edge routers. Core routers of this type can accommodate packets coming from fifty 5M PPS ingress edge routers. Note that 5M PPS is a typical packet handling capability of an ingress edge router (e.g., "HP MSR3000 Router series" [32]). Moreover, we set the link bandwidth of the core network to 10 Gbps.

The hardware component of an ingress edge router is shown in Fig. 7. This ingress edge router has 126 input lines and 1 output line. In reality, each packet from the ingress edge router has different destinations. In this paper, in order to evaluate fundamental performance levels, we assume that all packets are being sent to the same egress edge router. Moreover, unless otherwise stated, the

DOI: 10.1002/cpe

ingress edge router has three buffers (i.e., $N = 3$), and each of the three different traffic types arrives independently from the 42 input lines.

5.2. *Traffic model*

Since the end-to-end one-way delay in the core network is known as microsecond order [33], we set different microsecond order maximum waiting times for each of the applications. Detailed traffic parameters are listed in Table II.

The incoming traffic model on the ingress edge router is shown in Fig. 7. The ratio of the incoming traffic to the link bandwidth of output line is defined as Λ, and a ratio of traffic of class i is denoted by λ_i $(1 \leq i \leq N)$. That is, these parameters always satisfy the following condition:

$$\Lambda = \lambda_1 + \lambda_2 + \lambda_3 + \cdots + \lambda_N.$$

We examined the effectiveness of the proposed schemes when the values of Λ and $\lambda_i (1 \leq i \leq N)$ are varied. When $i < j$, the value of chunking timer of the ith buffer is smaller than that of the jth buffer. We assume the value of acceptable waiting time and the size of the packets are increasing in size with the increase in the class number. In our simulations, as the packet length is increased, the waiting time is increased as well. As a result, the performance of DCF in ChuTQ will be similar to that of CEF. Therefore, we investigated the performance of DCF and TPF without CEF.

Moreover, we define η as the ratio of the number of outgoing packets from the edge router to the maximum number of packets that the core router can handle per second (i.e., PSS). In addition, ζ is defined as the ratio of the number of incoming packets to the edge router to the PPS value.

We employ the following three traffic patterns in which data communication starts at 1 second and stops at 2 seconds.

(a) Case for homogeneous traffic ($\lambda_1 = \lambda_2 = \lambda_3$)
Packets from each class arrive at the same rate. This case will be referred to as the basic traffic case below. When the traffic load is relatively high (i.e., $\Lambda = 0.9$), the total arrival rate is set to 8.97 Gbps and the ratio of the number of incoming packets, ζ, is set at 2.28. Thus, the packet arrival rate exceeds the router forwarding capacity in PPS.

(b) Case for imbalanced traffic
The following two extreme cases will be examined.

(a) **Small (sensing) packets are dominant** ($\lambda_1 : \lambda_2 : \lambda_3 = 0.88 : 0.01 : 0.01$)

(b) **Large (file transfer) packets are dominant** ($\lambda_1 : \lambda_2 : \lambda_3 = 0.01 : 0.01 : 0.88$)

(c) Case for overloaded traffic ($\Lambda = 0.8 \rightarrow 1.2 \rightarrow 0.8, \lambda_1 = \lambda_2 = \lambda_3$)
At the beginning of communication, Λ is 0.8, and is then increased up to 1.2 at 1.3 seconds. After that, at 1.5 seconds, Λ is decreased to 0.8 again. Note that λ_1, λ_2, and λ_3 are always same (i.e., homogeneous traffic).

5.3. *Performance measures*

We evaluate the performance of proposed schemes from the viewpoint of the following performance measures:

1. Ratio of the number of outgoing packets from the edge router to the maximum number of packets that the router can handle per second (η):
   that is obtained under the basic traffic pattern (i.e., uniform distribution) in order to estimate the efficiency of chunking packets.
2. Output data rate:
   also obtained under the basic traffic pattern (i.e., uniform distribution) in order to estimate the throughput in terms of BPS in all packet chunking schemes.

DOI: 10.1002/cpe

3. Packet loss ratio:
   obtained under all traffic patterns in order to show the packet loss ratio of all schemes.
4. Ratio of packets violating the delay requirement at ingress edge router:
   obtained under all traffic patterns in order to show the possibility that none of the schemes can guarantee the QoS application requirement.
5. Average size of chunk packets:
   obtained under all traffic patterns in order to estimate the efficiency of chunking packets in all schemes.

6. PERFORMANCE ANALYSIS

In this section, we discuss how to evaluate the performance of a part of our proposed schemes using an analytical approach.

6.1. Analysis of packet loss ratio

In our proposed schemes, each buffer is based on a tail drop policy, which means that all arriving packets are dropped when the buffer is full.

- When $\Lambda < 1$, arriving packets are stored in a related buffer and dequeued due to timer expiration or MTU excess. The queue length will not exceed the MTU size. Therefore, packet loss does not occur.
- When $\Lambda \geq 1$, the queue length increases. When the queue length is lower than the maximum size that the buffer can store, arriving packets are stored in the buffer. In contrast, when the buffer is full, all arriving packets are dropped. Thus, the packet loss ratio (P_{loss}) is calculated by $P_{loss} = \Lambda - 1$.

6.2. Performance analysis of ChuMQ

In this section, we analytically obtain the two ChuMQ performance measures that were listed in the previous section: the ratio of packets violating the delay requirement, denoted by R_v, and the average size of chunk packets, $\overline{L_c}$. Packets are assumed to arrive according to the Poisson distribution. The probability that k packets will arrive for duration of T is given by the following equation [34]:

$$P(X = k) = \frac{(\lambda T)^k \mathrm{e}^{\lambda T}}{k!}. \tag{1}$$

6.2.1. Analysis of the ratio of packets violating the delay requirement

In order to obtain the ratio R_v, we first focus on a case where packets violate the delay requirement. Specifically, the case in which packets are chunked due to timer expiration and the resulting chunk packet cannot be transmitted immediately due to chunk packet transmission contention. In that case, the chunk packet is forced to wait for a period of time, which means that the HoL packet of chunk packet violates the delay requirement, and that its succeeding packets may do so as well.

The probability that the chunk packet is made due to the timer expiration is given by

$$P(Timer\ expiration) = \sum_{k=0}^{K-1} P(X = k), \tag{2}$$

in which T is the acceptable waiting time; i.e., T is replaced with T_i for class i (see subsection 4.2), so that T=50 μs for the sensing buffer, T=100 μs for the real-time buffer, and T=200 μs for the file transfer buffer. In addition, K is the upper bound on the number of packets stuffed in a chunk packet: i.e., K=225 (=9000/40) for the sensing buffer, K=45 (=9000/200) for the real-time buffer, and K=6 (=9000/1500) for the file transfer buffer. This probability indicates that, at most, only $K - 1$ packets will arrive before the acceptable waiting time elapses after the HoL packet arrival.

DOI: 10.1002/cpe

Hence, the probability of the number of packets involved in the resulting chunk packet, $P(M_c = m \mid Timer\ expiration)$, is as follows. Its size is denoted by M_c.

$$P(M_c = m \mid Timer\ expiration) = \frac{P(X = m)}{\sum_{k=0}^{K-1} P(X = k)} \tag{3}$$

In this case, among the m packets, those packets whose waiting times exceed their acceptable limits by the end of the ongoing packet transmission will eventually be in violation of their delay requirements. It is very difficult to exactly analyze the average number of such packets, but it is possible to obtain the upper and lower bounds, which are as follows. The average number of such packets in the worst case becomes $1 + \overline{M_c}_{T=7.2\mu s}$, in which '1' indicates the HoL packet and 7.2 μs is the transmission time of a 9000 byte chunk packet. From the above discussion, R_v can be bounded for class i as follows:

$$\frac{1}{\overline{M_c}_{T=T_i}} * \rho * P(Timer\ expiration) < R_v < \frac{(1 + \overline{M_c}_{T=7.2\mu s})}{\overline{M_c}_{T=T_i}} * \rho * P(Timer\ expiration). \tag{4}$$

Here, ρ is the traffic rate minus the rate of the relevant class. Although the exact solution of R_v is difficult to obtain, $(1 + \overline{M_c}_{T=3.6\mu s})/\overline{M_c}_{T=T_i} * \rho * P(Timer\ expiration)$ can be proposed as one candidate for an approximate solution, because the HoL packet arrives uniformly during the chunk packet transmission of 7.2 μs. The simulation and numerical results will be compared to examine the validity of this approximation in the following section.

6.2.2. Analysis of the average size of chunk packets

Next, we will produce an analysis of the average size of chunk packets in ChuMQ, which can be obtained by considering two cases separately: one case in which chunking is triggered by timer expiration, and another in which chunking is triggered by MTU size excess (i.e., 9000 bytes). The case in which the chunk packet size becomes the number of arriving packets $\times$ the size of the relevant class packets, h_i, in the former case: i.e., h_1=40 bytes for the sensing buffer, h_2=200 bytes for the real-time buffer, and h_3=1500 bytes for the file transfer buffer. Therefore the average size of chunk packet, $\overline{L_{ci}}$, from each buffer is given by

$$\overline{L_{ci}} = \sum_{k=0}^{K-1} P(X = k) * k * h_i + \sum_{k=K}^{\infty} P(X = k) * 9000. \tag{5}$$

Finally, the average size of chunk packet from all the classes, $\overline{L_c}$, can be obtained in the following manner. The average size of chunk packet from class i is given by the above equation, so that the ratio of chunk packets from each of the classes for a weighted average is required to calculate $\overline{L_c}$. The average number of packets in the chunk packet from class i, M_{ci}, is $\overline{L_{ci}}/h_i$. Packets arrive at a rate of λ_i(packets/second), so chunk packets are made at a rate of λ_i/M_{ci}(chunk packets/second). Consequently, the average size of chunk packet from all buffers is calculated by

$$\overline{L_c} = \frac{\sum_{i=1}^{N} \overline{L_{ci}} * \lambda_i / M_{ci}}{\sum_{j=1}^{N} \lambda_j / M_{cj}}. \tag{6}$$

In the following section, two ChuMQ performance parameters can be obtained numerically. Specifically, the ratio of packets violating the delay requirement, R_v, is given by Eq. (4), and the average size of chunk packets, $\overline{L_c}$, is given by Eq. (6).

7. NUMERICAL ANALYSIS AND SIMULATION RESULTS

In this section, we execute performance evaluations of our proposed packet chunking schemes. First, preliminary examination of the effect of packet chunking will be provided, after which we will

DOI: 10.1002/cpe

discuss the performance of our packet chunking schemes based on the numerical and simulation results.

7.1. Preliminary examination of the effect of packet chunking

Before extensively studying the performance of the chunking schemes, a preliminary examination will be provided below. Here, we deal with a case where the number of incoming packets per second exceeds the maximum number of packets the ingress edge router can handle per second, namely, $\zeta = 2.28 > 1$.

Our preliminary results, consisting of the ratio, denoted by η, of the number of packets the ingress edge router actually forwarded per second (PPS) to the maximum number of packets the ingress edge router can handle per second (maximum PPS) and the output data rate at the ingress edge router, are listed in Table III. For reference purposes, the table also includes η of the usual queue management based on first in, first out (FIFO) service without chunking; see the performance of "basic scheme without chunking".

As can be seen from the table, all of the chunking schemes achieve a throughput of 8.97 Gbps, which is exactly the same as the incoming traffic rate. In other words, all the incoming packets have definitely been forwarded. This is due to the fact that the chunking schemes can successfully reduce the number of outgoing packets to approximately 1% (=0.025/2.28) of the number of incoming packets. As a result, η is approximately 0.025, which is much less than the maximum PPS. On the other hand, without packet chunking, only 40% (=0.902/2.28) of the incoming packets can be forwarded, and the achievable throughput is limited to just 3.52 Gbps.

From this result, we can state that packet chunking allows the ingress edge router to fully utilize the available physical bandwidth. Note that our simulation results show that since there is no packet loss, the packet loss ratio (P_{loss}) is equal to zero. In the following section, we will discuss two performance measures, namely, ratio of packets violating the delay requirement and chunking efficiency.

7.2. Ratio of packets violating the delay requirement of each class

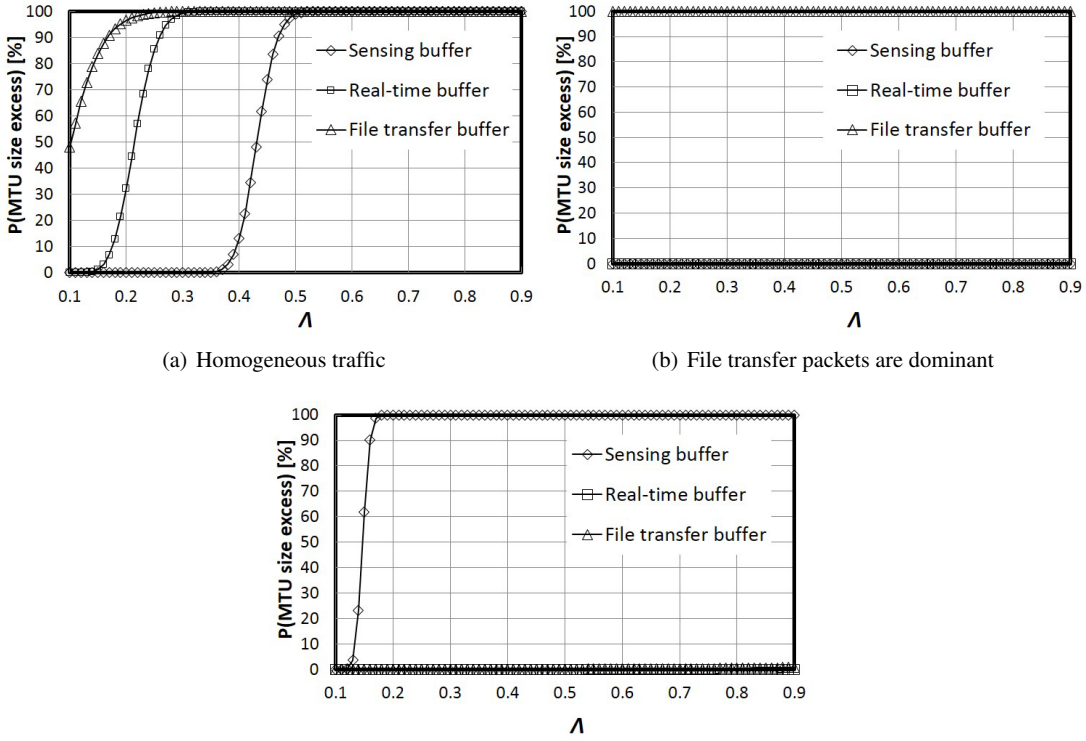
As shown in subsection 7.1, the proposed chunking schemes can successfully reduce the number of outgoing packets, thereby achieving high BPS throughput. At the same time, the QoS of packets should be considered. Here, in order to focus on the delay requirement (i.e., acceptable waiting time) as a QoS metric, we evaluate the ratio of packets violating the requirement in our proposed schemes when the values of Λ and $\lambda_i (i = 1; 2; 3)$ change. The delay requirements of class i are different from each other. Note that Λ changes from 0.1 to 0.9, except in the case of overloaded traffic.

7.2.1. Fundamental property in case of $\Lambda < 1.0$: Homogeneous and imbalanced traffic cases

Herein, the homogeneous traffic case ($\lambda_1 = \lambda_2 = \lambda_3$) and two kinds of imbalanced traffic case are discussed. The performance of ChuMQ is derived through the theoretical analysis described in Section 6 and the performance levels of the other schemes are discussed by simulation results. It should be noted that our simulation results indicate that all the packets meet the delay requirement in BasiCQ and ChuTQ in both the homogeneous and imbalanced cases.

First, as explained earlier, the chunking is triggered by two conditions: chunking timer expiration and MTU size excess. A chunk packet made by chunking timer expiration is intolerant of any further delay, and thus would be in violation of the timing requirement if chunk packet transmission contention occurs. Thus, $P(Timer\ expiration)$ given by Eq. (2) in ChuMQ affects R_v in the manner shown in Eq. (4), and $P(MTU\ size\ excess) = 1 - P(Timer\ expiration)$ is thus shown in Fig. 8. Basically, queue length is very likely to exceed the MTU size before the chunking timer expiration when the arrival rate is relatively large. In fact, as can be seen from the figure, in the homogeneous case, $P(MTU\ size\ excess)$ increases with Λ up to 100% when Λ is larger than 0.5 for all the classes. In addition, in the case of imbalanced traffic, $P(MTU\ size\ excess)$ is almost 100% for a wide range of Λ in the dominant traffic class, while it is zero in the low traffic rate class. Therefore, we can expect the dominant traffic class will meet the delay requirement over a wide (or the entire range) of Λ. Furthermore, in the homogeneous case, in which the traffic rate is the same

DOI: 10.1002/cpe



(a) Homogeneous traffic

(b) File transfer packets are dominant

(c) Sensing packets are dominant

Figure 8. Ratio of dequeue conditions of each buffer in ChuMQ.

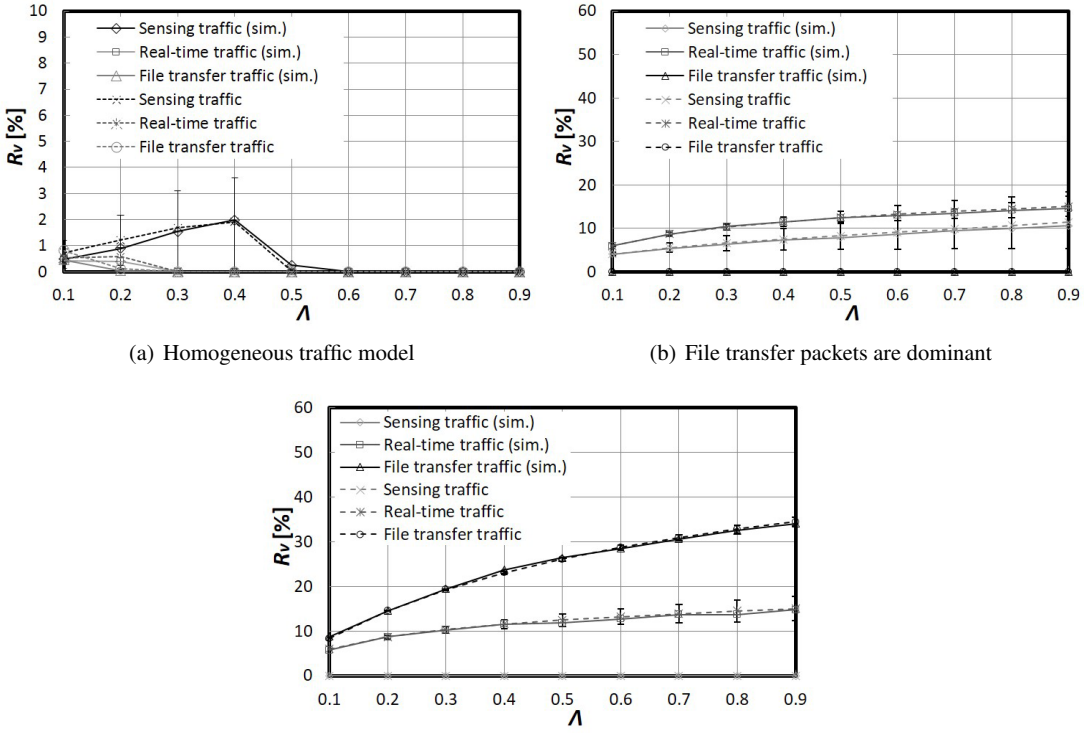
in all the classes, sensing packets are chunked based on their timer expirations over a wide range of Λ because they possess the shortest chunking timer.

Next, the ratio of packets violating the delay requirement, R_v, will be examined. In Fig. 9, we show simulation results on R_v of ChuMQ in the homogeneous traffic case and two imbalanced traffic cases. In addition, in ChuMQ, the upper and lower bounds on R_v are derived through inequality (4), and shown in the figure. The approximate solution to R_v is also indicated. The figure shows that R_v is tightly bounded by inequality (4), and the approximation is in excellent agreement with the simulation results.

In the cases of imbalanced traffic, the dominant class traffic does not violate the requirement at all. On the other hand, the R_vs of other classes increase with Λ because chunking occurs due to the timer expiration, and chunk packet transmission contention is more likely to occur for larger Λ. Furthermore, the R_v of the longer packet class is larger because only a smaller number of packets are concatenated in a chunk packet transmitted from the class with longer packet, and because one or more of the packets among them will be in violation of their delay requirements if chunk packet transmission contention occurs. Actually, when Λ=0.9, the average number of packets included in chunk packet, $\overline{M_c}$, is 16.63 and 7.25 in sensing traffic and real-time traffic when the file transfer traffic is dominant, and 7.25 and 2.66 in real-time traffic and file transfer traffic when the sensing traffic is dominant. Therefore, the ratio R_v of the longer packet becomes larger.

In the homogeneous case, the R_v of all the classes is almost zero. The sensing packets is not zero, but is still quite small over a narrow range of Λ. The reason is as follows. R_v is an increasing function of Λ, and only the sensing packets suffer from the chunk packet transmission contention for Λ of approximately 0.2 to 0.5, whereas the $P(Timer\ expiration)$ decreases when Λ is becoming larger than approximately 0.4, as shown in Fig. 8. Thus, the R_v increases very slightly with Λ from 0.2 to 0.4, and then decreases to zero.

DOI: 10.1002/cpe

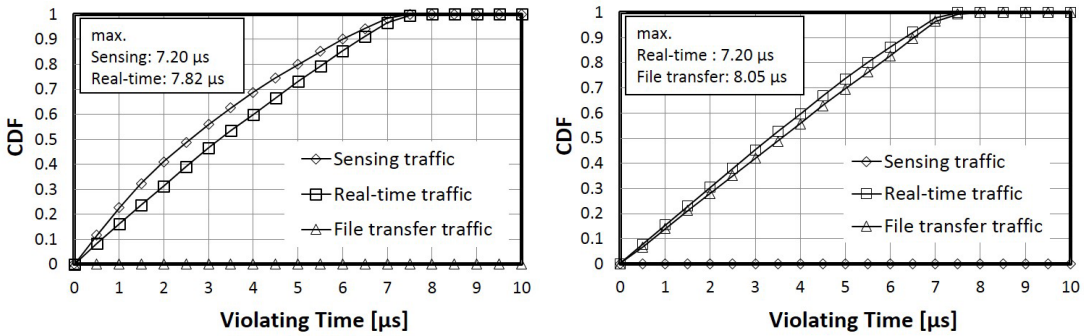


(a) Homogeneous traffic model

(b) File transfer packets are dominant

(c) Sensing packets are dominant

Figure 9. Ratio of packets violating the requirement (R_v) in ChuMQ.



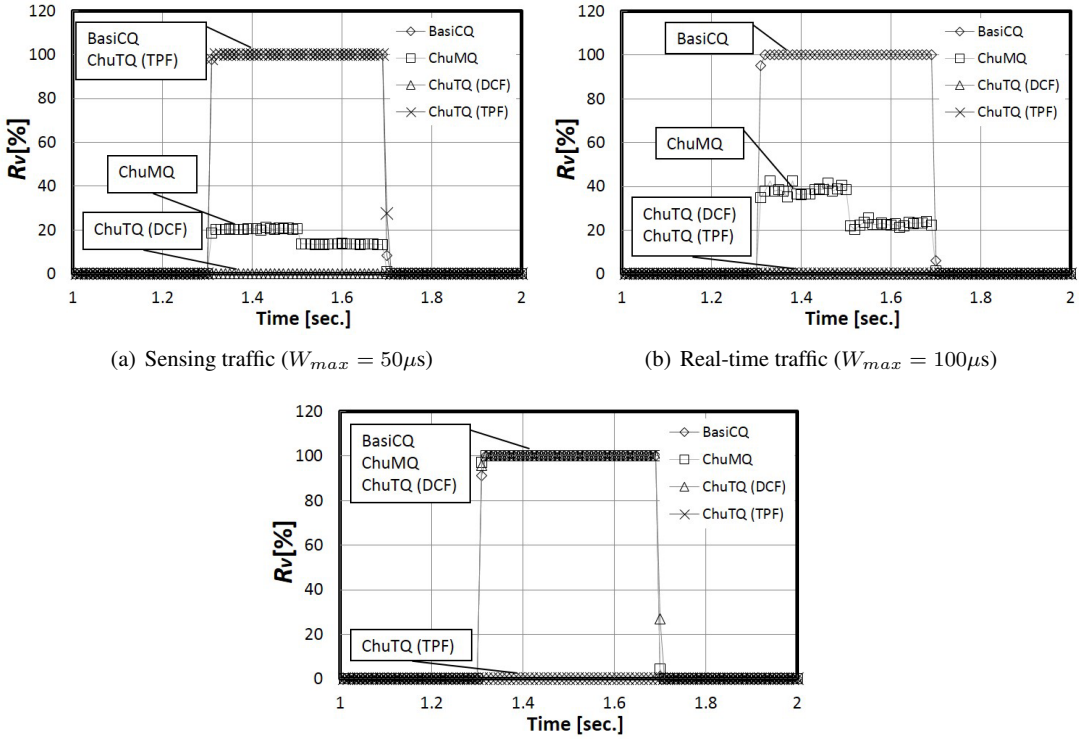
(a) File transfer packets are dominant

(b) Sensing packets are dominant

Figure 10. Cumulative distribution function (CDF) of violating time in ChuMQ when $\Lambda = 0.9$ (Simulation results).

We will now examine how the violation affects the delay time. The cumulative distribution function (CDF) of the excess of delay time over the acceptable waiting time is depicted in Fig. 10. Clearly, the excess is 0 in the dominant class traffic. On the other hand, the maximum excess time is 7.82 μs and 8.05 μs, as shown in Figs. 10(a) and 10(b), respectively. As explained earlier, when a chunk packet is being transmitted, the chunking timer of other classes can expire. In such cases, the buffered packet will be forced to wait for, at most, the transmission duration time of the chunk

DOI: 10.1002/cpe



(a) Sensing traffic ($W_{max} = 50\mu$s)

(b) Real-time traffic ($W_{max} = 100\mu$s)

(c) File transfer traffic ($W_{max} = 200\mu$s)

Figure 11. Ratio of packets violating their delay requirement (R_v) during a period of overloaded traffic.

packet. The MTU size used here is 9000 bytes and the transmission rate is 10 Gbps, so that the transmission duration time of a chunk packet is 7.2 μs.

The reason why the maximum of excess time is larger than 7.2 μs is that the chunking timers of the two classes (not dominant class) can expire while a chunk packet of the dominant class is transmitted. In this case, the packet with the shorter acceptable waiting time is first chunked and transmitted, which means that the other packet will wait for at most 7.2×2 = 14.4 μs, even though the simulation results are much less than that.

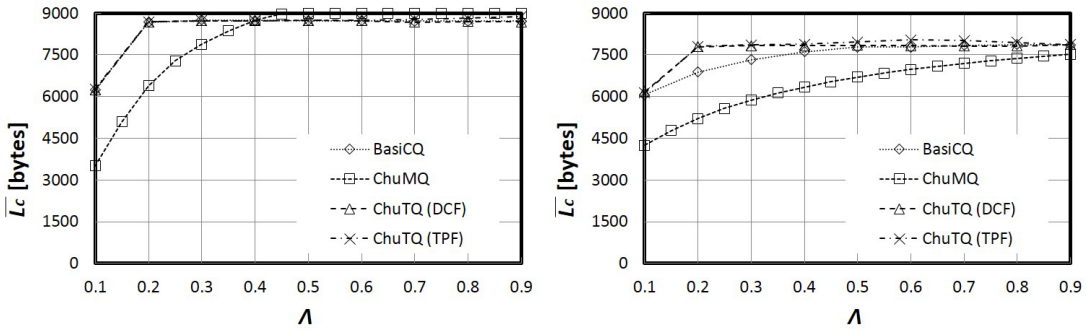
7.2.2. Examination of case of $\Lambda > 1.0$: Overloaded traffic case

Next, the overloaded traffic case of $\Lambda > 1.0$ will be discussed. Since the traffic actually fluctuates, Λ can be greater than 1.0 for a period of time. The traffic case examined, which was previously explained in subsection 5.2, is one in which a large number of packets are accommodated in the router.

The performance measure of our concern is the ratio of packets whose waiting time exceeds their acceptable time, which is shown in Table IV and Figs. 11(a)-11(c). From this table and these figures, we can see that, in BasiCQ, almost all packets of every class transmitted during the overloaded period (i.e., $\Lambda > 1$) are forced to wait for a period larger than their acceptable waiting times. That is, BasiCQ cannot handle incoming packets in a way that meets their delay requirements when the buffer is filled with a large number of packets.

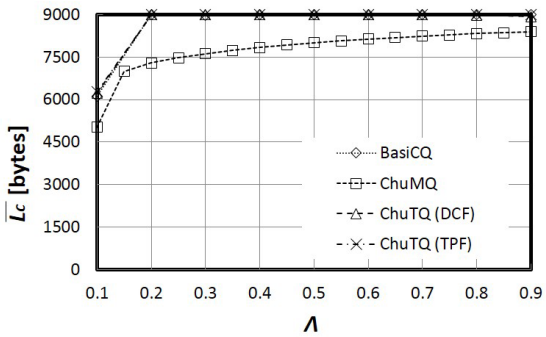
In contrast, ChuMQ can decrease R_v to 20.42% of sensing traffic and 38.33% of real-time traffic, even though it cannot reduce that of file transfer traffic. Therefore, we can say that ChuMQ provides better delay performance, especially for sensing and real-time traffic, than BasiCQ. This is because the value of the timer for packets with short acceptable waiting times is set to a small value, thereby allowing priority control to work for sensing and real-time applications.

DOI: 10.1002/cpe



(a) Homogeneous traffic model

(b) File transfer packets are dominant



(c) Sensing packets are dominant

Figure 12. Average size of chunk packet ($\overline{L_c}$) in BasiCQ, ChuMQ, ChuTQ (DCF), and ChuTQ (TPF).

Next, we will focus on the performance of ChuTQ. ChuTQ with DCF satisfies the QoS requirement for sensing and real-time applications, but it cannot guarantee the requirement for file transfer. Since the attribute value, α, of packets with short acceptable waiting time is likely to be large, those packets are preferentially chunked. On the other hand, ChuTQ with TPF guarantees the QoS requirement of real-time communication and file transfer applications, but does not completely guarantee those of sensing applications. In this scheme, the α of large packets tends to be large, so those packets are transmitted preferentially.

From these results, we can say that ChuMQ and ChuTQ can, at least to some extent, reduce the number of packets violating the delay requirement. In particular, ChuTQ with DCF and TPF provide excellent performance even under overloaded traffic conditions.

7.3. Chunking efficiency: How are large packets made through chunking?

As shown in subsections 7.1 and 7.2, our proposed chunking schemes can successfully achieve high throughput in BPS, while simultaneously guaranteeing the QoS requirement. Our major concern here is the chunking efficiency achieved by our schemes, which results in solving the PPS limitations issue and providing high BPS throughput: the chunk packet size is employed as a criterion. Thus, we will now compare the averaged chunk packet size of ChuMQ and ChuTQ.

7.3.1. Fundamental packet size property after chunking

Fig. 12 depicts the average chunk packet size ($\overline{L_c}$) in our proposed schemes. It is noted that the performance levels of ChuTQ (DCF) and ChuTQ (TPF) are the same as that of BasiCQ in Fig. 12 (a) and (c), and that the performance of ChuTQ (DCF) is the same as that of ChuTQ (TPF) in Fig. 12 (b). $\overline{L_c}$ in ChuMQ can be obtained by Eqs. (5) and (6).

DOI: 10.1002/cpe

As shown in Fig. 8, in ChuMQ, chunk packets are made by the timer expiration, in particular, when λ_i is small, which indicates that the size of resulting chunk packets does not always reach the MTU size. On the other hand, in BasiCQ and ChuTQ, a single buffer, which is the chunking buffer in ChuTQ, accommodates any packet from any class, so that the amount of packets buffered there is very likely to exceed the MTU size before the timer expiration, except in the case of a very small Λ, as shown in Fig. 12. Thus, we can see that the chunking efficiencies of BasiCQ and ChuTQ outperform ChuMQ over a wide range of Λ in each of the three traffic cases.

Nevertheless, since the single buffer accommodates packets of different sizes from different classes, a chunk packet will contain those differently sized packets and will not always match the MTU size. This is because chunk packets are made by sequentially adding packets buffered from the HoL packet, with the last packet removed when the chunk packet size exceeds the MTU size, and it is currently not possible to substitute the removed packet with any other packet following packets in order to produce a chunk packet of that is close to the MTU size.

7.3.2. How to further improve ChuMQ and ChuTQ: Chunking efficiency

As explained in Section 4.4, we proposed a number of CEA schemes that can be used to improve ChuMQ and ChuTQ from the viewpoint of chunking efficiency. Fig. 13 shows the averaged size of a chunk packet ($\overline{L_c}$) provided by ChuMQ + CEA and ChuTQ + CEA methods. From this figure, we can see that ChuMQ + CEA drastically increases $\overline{L_c}$ by up to 90%, especially in case of the small traffic (i.e., $\Lambda < 0.4$ in case of heterogeneous traffic). That is, chunking efficiency is drastically improved by the introduction of the CEA method.

Next, we will discuss how the ChuTQ + CEA method can improve the chunking efficiency in all the traffic cases. In particular, as shown in Fig. 13(b), we find the averaged chunk packet size could be increased 500 bytes by introducing the CEA method. From these results, we can state that the CEA method significantly improves the chunking efficiency of ChuMQ in particular, which is almost as efficient as ChuTQ.

7.4. Impact of the number of buffers on the performance: Examining cases of 3 and 10 buffers

In the sections above, we have shown that our chunking schemes can achieve high chunking efficiency by exploiting multiple buffers, while still guaranteeing the QoS requirement. In this subsection, we will evaluate how changes in the number of incoming applications and the number of buffers impact router performance. Here, in order to understand systems accommodating more diverse services, 10 application types with different packet sizes and acceptable waiting times are considered, as shown in Table V. Furthermore, we will assume that these applications are classified into three groups based on the QoS requirement: Group 1 (composed of Applications 1 to 4) is for sensing applications, Group 2 (composed of Applications 5 to 7) is for real-time applications, and Group 3 (composed of Applications 8 to 10) is for file transfer applications.

We then prepared two sets of buffer architecture and examined how changes in the number of buffers impacted the ratio of packets violating the delay requirement when Λ is fixed at 0.9.

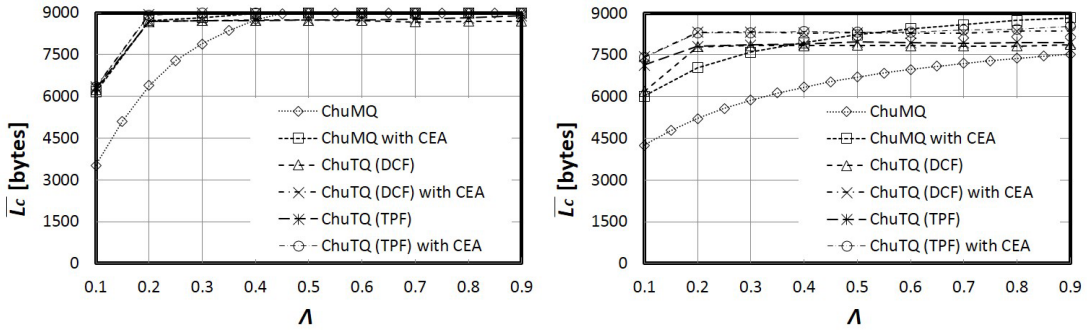
- Case of three buffers: each buffers was assigned to an application groups (not application).
- Case of ten buffers: each buffer was assigned to an application.

From the results shown in subsection 7.3, we can obviously expect both ChuMQ + CEA and ChuTQ + CEA to provide excellent performance in the case of homogeneous traffic. Therefore, in the following two subsections, our performance evaluation will consider severe cases of imbalanced traffic.

7.4.1. Performance when small packets with short acceptable waiting time are dominant ($\Lambda = 0.9$, $\lambda_1 = 0.81, \lambda_2 = \cdots = \lambda_{10} = 0.01$)

Table VI shows the ratio of packets violating the delay requirement (R_v), when the number of buffers is changed from three to ten. From this result, it can be seen that ChuMQ + CEA cannot satisfy the QoS requirement of Applications 2 to 10, whose traffic ratios are extremely small ($\lambda = 0.01$), when one dedicated buffer is assigned for each application (i.e., ten buffers). In this

DOI: 10.1002/cpe



(a) Homogeneous traffic model

(b) File transfer packets are dominant

(c) Sensing packets are dominant

Figure 13. Average size of chunk packet ($\overline{L_c}$) in ChuMQ and ChuTQ with CEA.

case, since chunking process is executed at each buffer, the buffer for application 1 always exceeds the MTU size and creates chunk packets due to its large traffic, but other buffers only make chunk packets at timer expiration. As explained in subsection 7.2.1, chunking timer expiration can occur in two or more classes over the duration of the chunk packet transmission from the dominant class. In such cases, the packets made due to chunking timer expiration are forced to wait until the chunk packet transmission from the dominant class is completed, even though their timers have expired. Since we assume the value of acceptable waiting time to increase with an increase in the application number, applications with long acceptable waiting times inherently experience long waiting time because they must compete for transmission with packets with shorter acceptable waiting times, which are preferentially served. That is why R_v is increased in proportion to the value of acceptable waiting time.

In the three buffer case, since the traffic levels of multiple applications are aggregated, the frequency of chunk packet transmission contentions caused by timer expirations will be decreased. However, applications with the shortest acceptable waiting time in each Group (e.g., application 5 in Group 2) are very likely to be forced to violate their delay requirement due to the waiting times caused by simultaneous timer expirations. As a result, the R_v for applications with the shortest acceptable waiting time in each group (i.e., applications 5 and 8) is increased, but less than that of the case of ten buffers. On the other hand, the QoS requirement for other applications, except for those with the shortest acceptable waiting time in each group, is sure to be satisfied.

It should also be noted that ChuTQ + CEA sequentially selects packets from HoL packets in multiple buffers, and then makes chunk packets using these selected packets, so that the chunk packet transmission is very likely to be triggered by MTU size excess. Furthermore, since chunk packets are made at one chunking buffer, chunk packet transmission contention never occurs. Therefore, ChuTQ + CEA can completely satisfy the QoS requirement.

DOI: 10.1002/cpe

Table VII shows the chunking efficiency of our proposed schemes. From this table, we can see that both ChuMQ + CEA and ChuTQ + CEA achieve excellent chunking efficiency irrespective of the number of buffers under cases where small packets are dominant. Therefore, we will further examine the ratio of packets violating QoS requirement only (without chunking efficiency).

7.4.2. Performance when large packets with long acceptable waiting times are dominant ($\Lambda = 0.9$, $\lambda_1 = \lambda_2 = \cdots = \lambda_9 = 0.01, \lambda_{10} = 0.81$)

Table IX shows the ratio of packets violating their QoS requirements when the large packets with long acceptable waiting time are dominant. From this table, in the case of ten buffers, ChuMQ + CEA cannot meet the relatively short acceptable waiting time QoS requirements of applications 1 to 3, although they can completely satisfy the QoS requirement of applications with long acceptable waiting times, including the dominant one (application 10). Since applications 1 to 3 have relatively short acceptable waiting times, chunk packet transmission occurs frequently. In parallel, the buffer assigned for a dominant class (application 10) also frequently transmits chunk packets triggered by MTU size excess. In such cases, packets of applications 1 to 3 are very likely to violate their delay requirement due to chunk packet transmission contention.

In the case of three buffers, because packets from applications 1 to 3 are classified into the same buffer, chunk packet transmission is triggered by MTU size excess. More specifically, chunk packet transmission is less frequently triggered by timer expiration due to the decrease in the number of buffers. As a result, R_v is clearly alleviated, compared with the case of ten buffers. As in the previous subsection, the QoS requirement of other applications, except for cases involving the shortest acceptable waiting time in each group, can be satisfied. It should be noted that ChuTQ + CEA can satisfy the QoS requirements of all of applications, regardless of the change in the number of buffers. In terms of chunking efficiency, ChuMQ + CEA and ChuTQ + CEA can construct large chunking packets.

7.4.3. Performance in overloaded traffic cases ($\Lambda = 1.2$, $\lambda_1 = \lambda_2 = \cdots = \lambda_{10}$)

As described in subsection 7.2.2, in case of $\Lambda > 1.0$, none of the schemes, including ChuTQ + CEA were found to be capable of meeting the QoS requirement for all applications. Therefore, we will now evaluate how the change in the number of buffers impacts QoS. It should be noted that packet chunking efficiency in this case was assumed to be excellent due to sufficient traffic levels.

As shown in Table VIII, ChuMQ + CEA cannot meet the QoS requirement for all applications when ten buffers are used. In this case, the traffic amounts of each buffer are actually so small ($\lambda_{1-10} = 0.12$) that almost all chunk packets are made due to timer expiration, as can be seen from Fig. 8(a). Furthermore, chunk packets transmitted from buffers with larger chunking timer expirations are likely to be forced to wait for chunk packet transmission completion from buffers with smaller chunking timer expirations. As a result, the ratio of violating packets gradually increases with the increase in delay requirement values. Here, applications 9 and 10 show the worst performance.

In contrast, in the three buffer case, from Fig. 8(a), we can see that the buffer for Group 1 makes chunk packets primarily due to the timer expiration, whereas the buffer for Group 3 makes chunk packets primarily due to MTU size excess. In this situation, as shown in Fig. 11 earlier, the QoS requirement of applications in Group 3 cannot be guaranteed at all. Additionally, the QoS requirements of applications with the shortest acceptable waiting time in Groups 1 and 2 (i.e., application 1 in Group 1 and application 5 in Group 2) are also not guaranteed because the additional waiting time results in chunk packet transmission contention among multiple buffers. However, the QoS requirement of other applications, except for the shortest ones, is completely satisfied. From these results, we can state that ChuMQ + CEA has the potential to meet the QoS requirement, to some extent, in a router equipped with a relatively small number of buffers.

Finally, ChuTQ + CEA selects packets from all buffers and creates chunk packets at the chunking buffer, which means that both chunk packet transmission due to timer expiration and unnecessary waiting times due to simultaneous timer expirations can be avoided effectively, irrespective of the number of buffers. Furthermore, since the priorities determined by attribute values can be

DOI: 10.1002/cpe

considered, the QoS requirement of applications, except for the one with the lowest priority, is sure to be satisfied. From these results, the effects of change in the number of buffers can be summarized as follows:

- In ChuMQ + CEA, it is more likely that packets will be in violation of the QoS requirement due to chunk packet transmission contention when a larger number of buffers are used. The issue can be alleviated by grouping multiple applications together.
- ChuTQ + CEA creates chunk packets by selecting packets from all buffers based on the attribute values. This ensures that the QoS requirement can be guaranteed, except in overloaded traffic conditions. When overloaded traffic occurs, there is no significant difference between the performance in ten and three buffers.

8. CONCLUSION

In this work, we focused on networks accommodating a huge number of small packets generated by numerous devices for M2M communication. This emerging traffic can cause a problem with router forwarding capability; specifically, the arriving packet rate can exceed the number of packets the router can handle per second (PPS). In an attempt to cope with the issue, the edge router-based solution, in which a function for chunking packets is added to the router in order to reduce the number of outgoing packets, was explored. Herein, we proposed a number of schemes for chunking packets, and compared their performance through simulation experiments and theoretical analysis. Their performance was examined to determine how well they could meet delay requirements and how efficiently they chunk packets.

Additionally, we proposed the chunking scheme with multiple queues (ChuMQ), and a scheme with tandem queue (ChuTQ), as well as one with a single queue (BasiCQ). ChuMQ was then analyzed theoretically, and the behavior of that chunking scheme was clearly shown. From simulation and numerical results, we showed that, unlike BasiCQ, ChuMQ and ChuTQ can guarantee the delay requirement of prioritized packets, even in overloaded traffic conditions. Furthermore, we found that ChuTQ outperforms other schemes in both of the performance measures mentioned above. Finally, the chunking efficiency aware (CEA) scheme was introduced in ChuMQ and ChuTQ, and changes to their effectiveness were shown. The effect of the number of buffers provided was discussed as well.

Our study has shown that chunking schemes can remarkably reduce the number of outgoing packets while meeting the different delay requirements for several classes. Some important features of chunking schemes could be made clear by our theoretical analysis.

Based on all our results, we recommend ChuTQ due to its strong ability to meet the delay requirement and achieve high chunking efficiency. In addition, conclude that three buffers provided in parallel are sufficient to achieve an adequate level of performance, and that the CEA scheme contributes significantly to the improvement of ChuMQ.

REFERENCES

1. Machine-to-Machine Communications: Connecting Billions of Devices. OECD Digital Economy Papers. 2012 January:45. DOI: 10.1787/5k9gsh2gp043-en
2. Machine-to-Machine communications (M2M); Use Cases of M2M applications for Connected Consumer. ETSI TR 102 857 v1.1.1. 2013 August:19.
3. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013-2018. Cisco White Paper. 2014 February.
4. IBM big data and information management. IBM; [2014 November]. Available from: `http://www-01.ibm.com/software/data/bigdata/`.
5. Big Data: A New World of Opportunities. NESSI White Paper. 2012 December:25.
6. Cisco Visual Networking Index: Forecast and Methodology, 2013-2018. Cisco White Paper. 2014 June.
7. Mohammad Tauhidul Islam, Abd-Elhamid M. Taha, and Selim Akl. A Survey of Access Management Techniques in Machine Type Communications. Communications Magazine, IEEE. 2014 April;52(4):74-81.
8. Thomas Potsch et al. Influence of Future M2M Communication on the LTE System. Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP. 2013 April:1-4. DOI: 10.1109/WMNC.2013.6549000.

9. Esa Piri, Jarno Pinola. Performance of LTE uplink for IoT backhaul. 2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC). 2016 January:6-11. DOI: 10.1109/CCNC.2016.7444723.
10. Will Small Packets Degrade Your Network Performance? STOKE Tech Insights, 2012:4.
11. Huawai's small data packet storm solution. [2012 August]. Available from: http://www.mobileeurope.co.uk/News-Analysis/huaweis-small-data-packet-storm.
12. Isabelle Deumont. Storm Watch 2012: Devastating Signaling Front Ahead. [2012 April]. Available from: http://blog.seven.com/2012/04/26/storm-watch-2012-devastating-signaling-front-ahead/.
13. Caroline Gabriel. DoCoMo demands Google's help with signalling storm. [2014 June]. Available from: http://www.rethink-wireless.com/2012/01/30/docomo-demands-googles-signalling-storm.htm.
14. Harish Vadada. Understanding the Signalng Tsunami. [2014 August]. Available from: http://www.telecom-cloud.net/understanding-the-signaling-tsunami/.
15. P. Arberg et al. Accommodating a Maximum Transit Unit/Maximum Receive Unit (MTU/MRU) Greater Than 1492 in the Point-to-Point Protocol over Ethernet (PPPoE). IETF RFC 4638. [2006 September]. Available from: http://tools.ietf.org/html/rfc4638.
16. A. Saif et al. An Enhanced A-MSDU Frame Aggregation Scheme for 802.11n Wireless Networks. Springer Wireless Personal Communications. 2012;66(4):683-706. DOI: 10.1007/s11277-011-0358-8.
17. D. Skordoulis et al. IEEE 802.11n MAC frame aggregation mechanisms for next-generation high-throughput WLANs. Wireless Communications, IEEE. 2008;15(1):40-47. DOI: 10.1109/MWC.2008.4454703.
18. R. Raghavendra et al. IPAC: IP-based adaptive packet concatenation for multihop wireless networks. Proceedings of IEEE Conference Signals, Systems and Computers (ACSSC 06). 2006;2147-2153. DOI: 10.1109/ACSSC.2006.355148.
19. E. Fasolo et al. In-network aggregation techniques for wireless sensor networks: a survey. Wireless Communications, IEEE. 2007;14(2):70-87. DOI: 10.1109/MWC.2007.358967.
20. R. Adams. Active Queue Management: A Survey. Communications Surveys & Tutorials, IEEE. 2013;15(3):1425-1476. DOI: 10.1109/SURV.2012.082212.00018.
21. V.M. Vokkarane, Q. Zhang, J.P. Jue, B. Chen. Generalized burst assembly and scheduling techniques for QoS support in optical burst-switched networks. Proceedings of IEEE Globecom'02. 2002;3:2747-2751. DOI: 10.1109/GLOCOM.2002.1189129.
22. Yijun Xiong, Marc Vandenhoute, and Hakki C. Cankaya. Control Architecture in Optical Burst-Switched WDM Networks. Selected Areas in Communications, IEEE Journal. 2000;18(10):1838-1851. DOI: 10.1109/49.887906.
23. Takuji Tachibana, and Shoji Kasahara. Performance analysis of timer-based burst assembly with slotted scheduling for optical burst switching networks. Performance Evaluation, Elsevier. 2006;63(9):1016-1031. DOI: 10.1016/j.peva.2005.11.005.
24. M. Klinkowskia et.al. An overview of routing methods in optical burst switching networks. Optical Switching and Networking, Elsevier. 2010;7(2):41-53. DOI: 10.1016/j.osn.2010.01.001.
25. Kouji Hirata, Takahiro Matsuda, and Tetsuya Takine. Dynamic burst discarding scheme for deflection routing in optical burst switching networks. Optical Switching and Networking, Elsevier. 2007;4(2):106-120. DOI: 10.1016/j.osn.2007.01.001.
26. Anan Sawabe, Kazuya Tsukamoto, Yuji Oie. QoS-Aware Packet Chunking Schemes for M2M Cloud Services. Proceedings of AINA Workshops 2014:166-173. DOI: 10.1109/WAINA.2014.36.
27. Tamer AbuHmed, Abedelaziz Mohaisen, and DaeHun Nyang. A Survey on Deep Packet Inspection for Intrusion Detection Systems. Magazine of Korea Telecommunication Society. 2007;24(11):25-36.
28. Ethernet Jumbo Frames. ethernet alliance White Paper. 2009 November:10.
29. Shaneel Narayan, Paula Raymond Lutui. Network Performance Evaluation of Jumbo Frames on a Network. 2013 6th International Conference on Emerging Trends in Engineering and Technology. 2013 December:69-72. DOI: 10.1109/ICETET.2013.16.
30. ns-3. ns-nam. http://www.nsnam.org.
31. Cisco Catalyst 4500. Cisco. [Aug. 2014]. Available from: http://www.cisco.com/web/JP/product/hs/switches/cat4500/.
32. HP MSR3000 Router Series. Hewlett-Packard Development Company. [2013 October]. Available from: http://h17007.www1.hp.com/us/en/networking/products/routers/HP_MSR3000_Router_Series/index.aspx#.UmIj-PnIak0.
33. E. Nygren, R. K. Sitaraman, J. Sun. The Akamai Network: A Platform for High-Performance Internet Applications. Newsletter ACM SIGOPS Operating Systems Review. 2010;44(3):2-19. DOI: 10.1145/1842733.1842736.
34. Ronald W. Wolff. Stochastic modeling and the theory of queues. Prenticce-Hall International. 1989:14.

Table I. Notation list for performance analysis.

Notation	Definition
W_{max}	The maximum amount of acceptable waiting time for chunking
T_i	The maximum chunking timer for ith buffer
N	The number of application class
Λ	The ratio of the incoming traffic to the link bandwidth of output line
λ_i	The ratio of incoming traffic of class i
η	The ratio of the number of outgoing packets from the edge router to the maximum number of packets which the core router can handle per second
ζ	The ratio of the number of incoming packets to the edge router to the PPS value
R_v	The ratio of packets violating their requirement
$\overline{L_c}$	The average size of chunk packets
h_i	The packet size in class i
$\overline{L_{ci}}$	The average size of chunk packet in class i
M_{ci}	The average number of packets in chunk packet from class i

Table II. Traffic model.

app. type	pkt size	max. acceptable waiting time (W_{max})
Sensing	40 bytes	50 μs
Real-time	200 bytes	100 μs
File transfer	1500 bytes	200 μs

Table III. Output traffic from the ingress edge router.

scheme	η	Output data rate [Gbps]
Basic scheme without chunking	0.902	3.52
BasiCQ	0.026	8.97
ChuMQ	0.025	8.97
ChuTQ (DCF)	0.026	8.97
ChuTQ (TPF)	0.025	8.97

DOI: 10.1002/cpe

Table IV. Ratio of packets violating the delay requirement (R_v) during a period of overloaded traffic.

scheme	Sensing [%]	Real-time [%]	File transfer [%]
BasiCQ	99.88	99.76	99.55
ChuMQ	20.42	38.33	99.84
ChuTQ (DCF)	0.00	0.00	99.79
ChuTQ (TPF)	99.90	0.00	0.00

Table V. Traffic model.

app. type	app. No.	packet size [bytes]	max. acceptable waiting time (W_{max}) [μsec]
Group 1 (sensing)	1	40	50
	2	40	60
	3	40	70
	4	40	80
Group 2 (real-time)	5	200	100
	6	200	120
	7	200	140
Group 3 (file transfer)	8	1500	160
	9	1500	180
	10	1500	200

Table VI. Ratio of packets violating their delay requirements (small packets with short W_{max} are dominant).

scheme	app.	R_v [%]	
		3 buffers	10 buffers
ChuMQ + CEA	1	0.000	0.000
	2	0.000	8.519
	3	0.000	7.316
	4	0.000	6.580
	5	8.601	13.778
	6	0.000	12.622
	7	0.000	10.748
	8	24.727	38.360
	9	0.000	34.040
	10	0.000	33.821

Table VII. Chunking efficiency (small packets with short W_{max} are dominant).

scheme	buffer	$\overline{L_c}$ [bytes]
ChuMQ + CEA	3	8997.39
	10	8992.90
ChuTQ (DCF) + CEA	3	8968.08
	10	8995.22

DOI: 10.1002/cpe

Table VIII. Ratio of packets violating their delay requirements (R_v) during a period of overloaded traffic.

scheme	app.	R_v [%]	
		3 buffers	10 buffers
ChuMQ + CEA	1	23.389	4.982
	2	0.000	6.198
	3	0.000	9.292
	4	0.000	11.646
	5	36.438	17.147
	6	0.000	22.253
	7	0.000	31.219
	8	99.828	64.641
	9	99.876	99.800
	10	99.569	99.772
ChuTQ (DCF) + CEA	1	0.000	0.000
	2	0.000	0.000
	3	0.000	0.000
	4	0.000	0.000
	5	0.000	0.000
	6	0.000	0.000
	7	0.000	0.000
	8	99.731	0.000
	9	99.594	99.670
	10	99.538	99.555

Table IX. Ratio of packets violating their delay requirements (large packets with long W_{max} are dominant).

scheme	app.	R_v [%]	
		3 buffers	10 buffers
ChuMQ + CEA	1	7.970	10.828
	2	0.000	0.789
	3	0.000	0.007
	4	0.000	0.000
	5	0.400	0.000
	6	0.000	0.000
	7	0.000	0.000
	8	0.000	0.000
	9	0.000	0.000
	10	0.000	0.000

DOI: 10.1002/cpe