LETTER

A QoS-Aware Differential Processing Control Scheme for OpenFlow-Based Mobile Networks

Yeunwoong KYUNG[†], **Taihyong YIM**[†], *Nonmembers*, **Taekook KIM**[†], **Tri M. NGUYEN**[†], *Student Members*, *and* **Jinwoo PARK**[†a)], *Nonmember*

SUMMARY This paper proposes a QoS-aware differential processing control (QADPC) scheme for OpenFlow-based mobile networks. QADPC classifies the input packets to the control plane by considering end terminal mobility and service type. Then, different capacities are assigned to each classified packet for prioritized processing. By means of Markov chains, QADPC is evaluated in terms of blocking probability and waiting time in the control plane. Analytical results demonstrate that QADPC offers high priority packets both lower blocking probability and less waiting time.
key words: *OpenFlow, quality of service (QoS), mobile flow, SDN*

1. Introduction

In contrast to the tight coupling structure of conventional networks, OpenFlow separates the control and data plane. This decoupled architecture allows an effective method of managing the data plane from the control plane by an open interface (i.e., OpenFlow). In the data plane, a switch matches the incoming packets with the flow entries. If an entry is matched, the instructions are performed according to the flow entry [1]. The flow entries are installed reactively or proactively by the control plane. The reactive flow setup approach, in which entries are added in response to the first packet of each flow, is usually assumed in mobile networks due to policy changes and host mobility [2]. Furthermore, statistical information related to each flow is reported to the control plane. Such flow based fine-grained control supports traffic optimization with visibility over all flows.

However, the centralized OpenFlow architecture leads to a scalability issue (i.e., bottleneck in the control plane due to the flow entry installation for all flows). The bottleneck can cause problems such as long flow entry installation delay or blocking of input packets. These problems are critical for the packets right after a handover or for the initial delay sensitive service packets because users will experience severe QoS degradation when these packets are blocked or delayed [3].

Several works have been performed on the scalability issue in OpenFlow-based networks. Earlier works [2], [4] could reduce the number of packets to the control plane because only a few or no packets are processed by the control plane. In addition, another work [5] used multiple controllers to distribute the load on each controller. However, these approaches only focused on reducing the load in the control plane. They did not consider the QoS of the incoming packets to the control plane. As explained earlier, if the load is concentrated in the control plane, QoS degradation will result because of long flow entry installation delay or blocking of the packets. Therefore, QoS-aware processing is required.

Several works were performed for QoS support in OpenFlow-based networks. A fine-grained automated QoS control scheme [6] is proposed using rate shaping and priority queue mapping. Also, a dynamic rerouting scheme [7] is proposed to ensure delivery of high priority flows within specified constraints. However, these works only considered how to deliver the packets for QoS provision in the data plane.

To tackle the above problem, we propose a QoS-aware differential processing control (QADPC) scheme for OpenFlow-based mobile networks. We introduce a packet classifier to classify the packets according to the QoS and differentially allocate the capacity to process each classified packet. By means of Markov chains, we evaluate QADPC in terms of the blocking probability and waiting time in the control plane. Analytical results show that QADPC offers high priority packets both lower blocking probability and less waiting time.

2. System Model

Figure 1 shows the proposed system model. The first packets of every flow are sent from the data plane to the control plane. Then, the packet classifier classifies the packets into three kinds: handover packets with high priority (HP packets), just high priority packets (P packets, and not handover

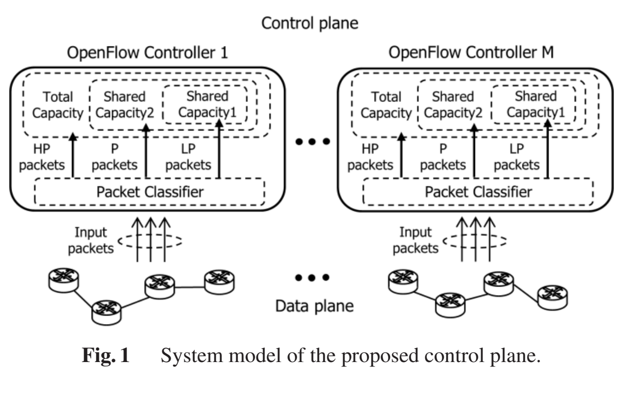

Fig. 1 System model of the proposed control plane.

Manuscript received March 6, 2014.
Manuscript revised April 9, 2014.
[†]The authors are with the School of Electrical Engineering, Korea University, Seoul 136–701, Korea.
a) E-mail: jwpark@korea.ac.kr
DOI: 10.1587/transinf.E97.D.2178

packets), and other packets with low priority (LP packets). We consider services with stringent QoS requirements for flow initiation delay and loss as high priority services, such as command/control and video telephony. In addition, the packets of the high priority services are defined as high priority packets. For example, the command/control packets require a loss ratio of 0% for service management, and the video telephony packets are very sensitive to the initial delay for playback performance [3].

We define a handover packet as the first packet after handover to a new access node, which means that the flow entry of the packet exists in the previous serving nodes on the path before the handover. As explained earlier, protecting the handover packets from being blocked or delayed is important especially in high priority services. Therefore, we consider the HP packets as the first priority packets and give second priority to the P packets. The LP packets have the lowest priority.

After the classification, each packet is processed using different resources in the control plane. Our system achieves QoS provisioning by controlling the capacity of the LP and P packets.

We also consider multiple controllers for scalability. When we use multiple controllers, we assume that the switches are divided into groups and each group is statically managed by different controller as shown in Fig. 1. Moreover, only one controller per each switch processes every message from the switch for safety [8].

Ingress Port	Ethernet Dst	Ethernet Src	IP Src	IP Dst	...	QoS type	Priority	
*	00:1E:68:A5:21	02:6A:D4:13:50	1.2.3.4	5.6.7.8	...	10	2	HP packets
*	01:2B:2A:51:21	03:7K:A1:32:76	9.10.11.1	13.14.15.1	...	13	2	
...	...	...	...	...	...	...	...	
*	*	*	*	*	...	10	1	P packets
*	*	*	*	*	...	13	1	
...	...	...	...	...	...	...	...	
*	*	*	*	*	...	*	0	LP packets

Fig. 2 Example of a table in a packet classifier.

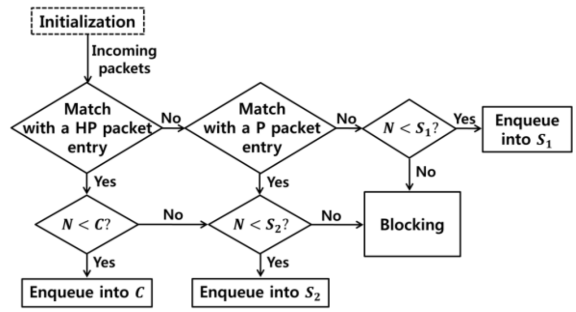

Fig. 3 Differential processing operation.

3. QoS-Aware Differential Processing Control

In this section, we present the two phases of QADPC: packet classification and differential processing.

Figure 2 shows an example of the table entries in the packet classifier. The table consists of matching fields, including the ingress port, Ethernet destination and source addresses, IP destination and source addresses, QoS type field, etc. and priority field. We define the ingress port as the logical port number from the access node. The QoS type field can be any field supported by OpenFlow, which represents the QoS of the packets such as TCP port number, DSCP, and ToS field. In other words, the priority field can be modified adaptively and also multiple QoS type fields can be supported by using TLV format [1] based on the QoS policy of network operator. In Fig. 2, we assume that 10 and 13 are the numbers that denote the QoS of the high priority services. The packet classifier matches the packets in priority order. Therefore, we set the priority fields as 2, 1, and 0 for the HP, P, and LP packets, respectively. In the packet classifier, P packet entries are pre-set only with pre-defined QoS type values for matching with the P packets. After P packets are processed, HP packet entries are installed with P packets' header fields (flow information). HP packet entries are exact matching entries that all the fields are filled with specific value. However, only the ingress port fields are wildcarded because the access node changes after the handover. Therefore, packets matched with HP packet entries are recognized as HP packets from the point of controller's view. Each entry is maintained using the statistical reports from the data plane and removed when it has matched no packets in the idle-timeout [1]. Packets not matched with HP and P packet entries, are classified as LP packets.

After matching with the packet classifier, input packets are classified as HP, P, or LP packets. Each classified packet is differentially processed employing a cutoff priority scheme [9] by dividing the total capacity (C) of each controller into three parts: reserved capacity ($C - S_2$) exclusively used for HP packets, shared capacity ($S_2 - S_1$) for both HP packets and P packets, and the remaining shared capacity (S_1) for all kinds of packets. The overall procedure for the differential processing of each controller is shown in Fig. 3. N is the number of packets currently served in a controller. When a packet is processed, N is decreased by one, and an incoming packet is determined whether it can be enqueued. If the packet is matched with an HP packet entry, it is enqueued when the total capacity is not full. On the other hand, if the packet is classified as a P packet, it can be accepted when N is less than S_2. Finally, in the case of an LP packet, the packet is enqueued only when N is less than S_1.

4. Performance Analysis

In this section, we develop an analytical model of QADPC. We assume that the arrival process per system (entire controllers) of the HP, P, and NP packets is modeled as Poisson distributions with rates λ_{HP}, λ_P, and λ_{LP}, respectively. The service time of the control plane follows an exponential distribution with mean $1/\mu$.

Figure 4 shows the state transition diagram of QADPC. In state (i, j), i is the controller number when multiple con-

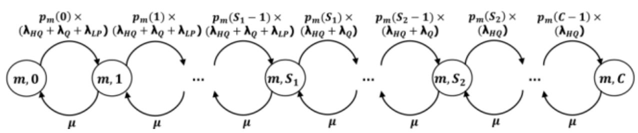

Fig. 4 Markov chain for QADPC.

trollers exist, and j represents the number of packets currently served in the ith controller. $p_i(j)$ is the probability that a packet can be inserted into the ith controller when j packets exist in the controller. We let M be the number of controllers and C, S_1, and S_2 are defined earlier. Therefore, the state space is given as $\mathrm{S} = \{(i, j)|i \in \{1, 2, \ldots, M\}, 0 \leq j \leq \mathrm{C}\}$. We let $p(i, j; k, l)$ be the transition rate from state (i, j) to state (k, l). Then, we have

$$\begin{aligned} p(i, j; i, j+1) &= p_i(j) \cdot (\lambda_{HP} + \lambda_P + \lambda_{LP}), \\ & \qquad for \quad 0 \leq j < S_1 \\ p(i, j; i, j+1) &= p_i(j)(\lambda_{HP} + \lambda_P), \\ & \qquad for \quad S_1 \leq j < S_2 \\ p(i, j; i, j+1) &= p_i(j)(\lambda_{HP}), \\ & \qquad for \quad S_2 \leq j < C \\ p(i, j; i, j-1) &= \mu, \\ & \qquad for \quad 1 \leq j \leq C \end{aligned} \tag{1}$$

We let $\pi(i, j)$ be the steady state probability that j packets exist in the ith controller. Deriving $\pi(i, j)$ is computationally difficult because we consider the probability at each state that a packet is inserted into the ith controller when j packets exist (i.e., $p_i(j)$). Therefore, for simplicity, we assume that this probability only depends on the number of controllers and is equal at each state irrespective of the state condition (i.e., $1/M$). The analysis that considers each state condition will be one of our future works. On the basis of the above assumptions, we obtain the following steady state probability from the detailed balance equation.

$$\pi(i, j) = \begin{cases} \left(\dfrac{\lambda_{HP} + \lambda_P + \lambda_{LP}}{\mu \cdot M}\right)^{j} \cdot \pi(i, 0), \\ \qquad\qquad for \quad 0 \leq j \leq S_1 \\ \left(\dfrac{\lambda_{HP} + \lambda_P + \lambda_{LP}}{\mu \cdot M}\right)^{S_1} \cdot \left(\dfrac{\lambda_{HP} + \lambda_P}{\mu \cdot M}\right)^{j-S_1} \cdot \pi(i, 0), \\ \qquad\qquad for \quad S_1 + 1 \leq j \leq S_2 \\ \left(\dfrac{\lambda_{HP} + \lambda_P + \lambda_{LP}}{\mu \cdot M}\right)^{S_1} \cdot \left(\dfrac{\lambda_{HP} + \lambda_P}{\mu \cdot M}\right)^{S_2-S_1} \\ \quad \cdot \left(\dfrac{\lambda_{HP}}{\mu \cdot M}\right)^{j-S_2} \cdot \pi(i, 0), \quad for \quad S_2 + 1 \leq j \leq C \end{cases} \tag{2}$$

By using the steady state probability, the LP packet blocking probability (LPBP), P packet blocking probability (PBP) and HP packet blocking probability (HPBP) at the ith controller are obtained as

$$LPBP = \sum_{j=S_1}^{C} \pi(i, j) \tag{3}$$

$$PBP = \sum_{j=S_2}^{C} \pi(i, j) \tag{4}$$

$$HPBP = \pi(i, C) \tag{5}$$

To derive the average HP packet waiting time (T_W) at the ith controller, the average number of packets (N_{avg}) at the ith controller should be first computed as

$$N_{avg} = \sum_{j=1}^{C} j \cdot \pi(i, j) \tag{6}$$

The effective arrival rate (λ_{eff}) is expressed as

$$\begin{aligned} \lambda_{eff} &= \sum_{j=1}^{S_1-1} (\lambda_{HP} + \lambda_P + \lambda_{LP}) \cdot \pi(i, j) \\ &\quad + \sum_{j=S_1}^{S_2-1} (\lambda_{HP} + \lambda_P) \cdot \pi(i, j) + \sum_{j=S_2}^{C-1} (\lambda_{HP}) \cdot \pi(i, j) \end{aligned} \tag{7}$$

By using the Little's law [10], we obtain the average waiting time (T_W) at each controller as

$$T_w = \frac{N_{avg}}{\lambda_{eff}} + \frac{1}{\mu} \tag{8}$$

5. Numerical Results

In this section, we evaluate the performance of QADPC compared with that of the conventional non-prioritized processing method in terms of the blocking probability and average HP packet waiting time. For the numerical results, we set S_1 and S_2 to 60% and 80% of the total capacity C. Although we fix the size of S_1 and S_2, they can be flexible depending on the network policy. In addition we assume that the mean service time ($1/\mu$) is 1ms. Furthermore, we choose the following set of parameters: $\lambda_P = 350$, $\lambda_{LP} = 350$ when $M = 1$, and $\lambda_P = 700$, $\lambda_{LP} = 700$ when $M = 2$. On the other hand, λ_{HP} is varied from low to high considering the traffic characteristics of real LTE network [11].

Figure 5 shows the blocking probabilities as the HP packet arrival rate increases when M is equal to one and two. The blocking probabilities of the conventional scheme become higher with the increase in the HP packet arrival rate irrespective of the types of packets. On the other hand, the blocking probabilities of QADPC depend on the types of packets. Because QADPC reserves the $C - S_2$ capacities only for the HP packets, the HPBP exhibits the lowest blocking probability regardless of the HP packet arrival rate and the number of controllers. In addition, the PBP of QADPC appears lower than that of the conventional scheme because of the reserved $C - S_1$ capacities. However, QADPC has a higher LPBP than the conventional scheme because of the minimum capacities. Further, we can notice that multiple controllers can support a far larger network with acceptable blocking probability.

Figure 6 shows the average waiting time for processing an HP packet as the HP packet arrival rate increases

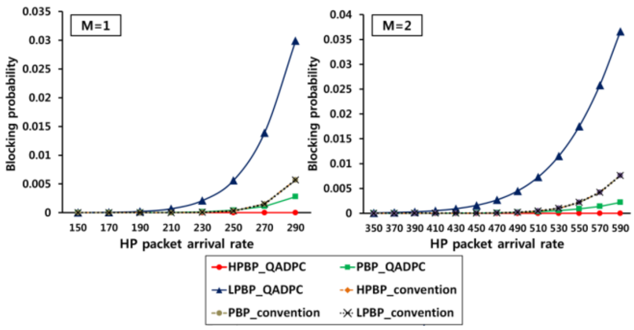

Fig. 5 Blocking probability according to the HP packet arrival rate.

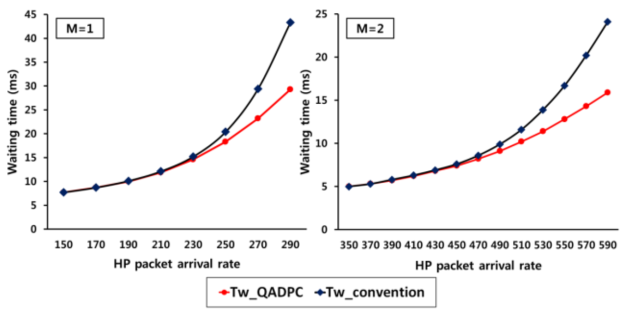

Fig. 6 HP packet waiting time according to the HP packet arrival rate.

when M is equal to one and two. Average waiting times of both QADPC and conventional scheme are shorter when M equals 2 compared to those when M equals 1 because the number of input packets per controller is reduced when M increases. We can notice that QADPC has a lower average waiting time than the conventional scheme because of the preserved $C - S_2$ capacities for the HP packets. Moreover, the difference between them becomes higher with the increase in the HP packet arrival rate because the conventional scheme equally admits all types of packets without considering the QoS even in bottleneck situations. Actually, the packets can be deliverd after the flow entry for the HP packet is installed at the new serving nodes on the new path after the handover. In other words, QADPC can minimize the handover latency for high priority services.

6. Conclusion

This letter presents QADPC to provide prioritized processing for OpenFlow-based mobile networks. The incoming packets to the control plane are classified according to the predefined priority. Then, the capacities are assigned differentially to guarantee the QoS of each classified packet. Analytical results show that QADPC offers high priority packets both lower blocking probability and less waiting time compared with the conventional scheme. In our future work, we will extend QADPC for non PACKET-IN messages such as state change messages and we will handle them with higher priority than PACKET-IN messages because they are used for global network view and traffic optimization.

Acknowledgments

This work was supported partly by the IT R&D program of MOTIE/KEIT [10043462, The development of Gigabit Wireless Backhaul Transmission System Connecting EPC Network & Small Cell BS for NLOS/LOS Environment], and partly by the MSIP (Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2013 [2013-005-031-001, Development of Device Collaborative Giga-Level Smart Cloudlet Technology].

References

[1] "Openflow switch specification, version 1.4.0." online. http://www.openflowswitch.org/documents/openflow-spec-v1.4.0.pdf, Open Networking Foundation, Oct. 2013.
[2] M. Yu, J. Rexford, M.J. Freedman, and J. Wang, "Scalable flow-based networking with difane," Proc. SIGCOMM Comput. Comm. Rev., vol.40, no.4, pp.351–362, Oct. 2010.
[3] R. Stankiewicz and A. Jajszczyk, "A survey of QoE assurance in converged networks," Elsevier Computer Networks, vol.55, pp.1459–1473, 2011.
[4] A.R. Curtis, J.C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, "Devoflow: Scaling flow management for high-performance networks," Proc. SIGCOMM Comput. Comm. Rev., vol.41, no.4, pp.254–265, Aug. 2011.
[5] S. Yeganeh and Y. Ganjali, "Kandoo: A framework for efficient and scalable offloading of control applications," Proc. HotSDN'12, pp.19–24, Aug. 2012.
[6] W. Kim, P. Sharma, J. Lee, S. Banerjee, J. Tourrilhes, S.J. Lee, and P. Yalagandula, "Automated and scalable qos control for network convergence," Proc. USENIX INM/WREN, pp.1–6, April 2010.
[7] H.E. Egilmex, S. Civanlar, and A.M. Tekalp, "An optimization framework for qos-enabled adaptive video streaming over openflow networks," IEEE Trans. Multimedia, vol.15, no.3, pp.710–715, April 2013.
[8] A. Dixit, F. Hao, S. Mukherjee, T.V. Lakshman, and R. Kompella, "Towards and Elastic Distributed SDN Controller," Proc. ACM HotSDN, pp.7–12, Aug. 2013
[9] Y. Fang and Y. Zhang, "Call Admission Control Schemes and Performance Analysis in Wireless Mobile Networks," IEEE Trans. Veh. Technol., vol.51, no.2, pp.371–382, March 2002.
[10] D. Gross and G.M. Harris, Fundamentals of queueing theory, 3rd ed., John Wiley & Sons, 1998.
[11] X. Jin, L.E. Li, L. Vanbever, and J. Rexford, "SoftCell: Scalable and Flexible Cellular Core Network Architecture," Proc. ACM CoNEXT, pp.163–174, 2013.