Effective Router Assisted Congestion Control for SDN

Sofia Naning Hertiana[1], Adit Kurniawan[2], Hendrawan[3], Udjianna Sekteria Pasaribu[4]
[1,2,3]School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Indonesia
[4]Faculty of Mathematics and Natural Sciences, Institut Teknologi Bandung, Indonesia

Article Info

Article history:

Received Mar 16, 2018
Revised Jul 6, 2018
Accepted Jul 20, 2018

Keyword:

Congestion control
Global knowledge
PACEC
SDN
Traditional Internet

ABSTRACT

Router Assisted Congestion Control (RACC) was designed to improve end-to-end congestion control performance by using prior knowledge on network condition. However, the traditional Internet does not provide such information, which makes this approach is not feasible to deliver. Our paper addresses this network information deficiency issue by proposing a new congestion control method that works on the Software Defined Network (SDN) framework. We call this proposed method as PACEC (Path Associativity Centralized Congestion Control). In SDN, global view of the network information contains the network topology including link properties (i.e., type, capacity, power consumption, etc.). PACEC uses this information to determine the feedback signal, in order for the source to start sending data at a high rate and to quickly reach fair-share rate. The simulation shows that the efficiency and fairness of PACEC are better than Transmission Control Protocol (TCP) and Rate Control Protocol (RCP).

Corresponding Author:

Sofia Naning Hertiana,
School of Electrical Engineering and Informatics,
Institut Teknologi Bandung, Indonesia,
Jl. Ganesha No.10, Bandung 40132, Indonesia.
Email: sofiananing@telkomuniversity.ac.id

1. INTRODUCTION

Congestion control is a process to regulate the sending rate of the source on the network so the sender may adjust the entry of data according to the network condition to ensure sender's Quality of Service (QoS) requirements are fulfilled [1]. A good congestion control will ensure that there is no drop in quality and assure a good quality network [2], [3]. It is widely known that an end-to-end packet loss control such as Transmission Control Protocol (TCP) has several inefficiencies [4]. One reason of inefficiency is that TCP treats packet loss as congestion signal. Therefore, it cannot distinguish between packet loss due to congestion with packet loss due to other causes.

Packet loss as a congestion signal implies that action can only be done after congestion occurs [5]. TCP uses assumption that the network does not provide explicit feedback to the source [6], which makes each source to estimate the nature of the network path, such as round trip time (RTT) or usable bandwidth, to deliver efficient end-to-end congestion. To improve TCP performance, previous research introduced the use of explicit congestion signals to the network [7]. This method is commonly called the Router Assisted Congestion Control (RACC). In the traditional Internet, RACC method is applied to each router where the router provides feedback to the end system about the state of the network and tells the sender to send packets at a specific rate. Figure 1 gives a simple overview of routers that always provide feedback information to the end-user. RACC is modeled using the M/G/1-PS queue theory to calculate the aggregate rate on each router (node). The aggregated rate at each node *l* follows a simple model as described by the following equation [8]:

$$R_l = C_l\,(1 - \rho_l) \tag{1}$$

where R_l is the sending rate for a link, C_l is the link capacity, and ρ_l is the link utilization.

There are three solutions offered for RACC in the traditional Internet. The first solution is to use a router to detect congestion and send this information to the end system. Based on this information, the end system decides to control the congestion on the network. The example of this category is Explicit Congestion Notification (ECN) [9] and Quick-Start [10]. The sending rates in both protocols are decided at the end system based on information received from the network and the characteristics of each application. The second solution is to use a router to detect congestion and available network resources, and at the same time, the router distributes network resources for each information flow. The example of this category is Explicit Congestion Control Protocol (XCP) [11], and Rate Control Protocol (RCP) [12]. In this second approach, the end system only accepts recommendations from the router to adjust its sending rate. This approach allows the router to decide the sending rate for each flow without causing congestion on the network.

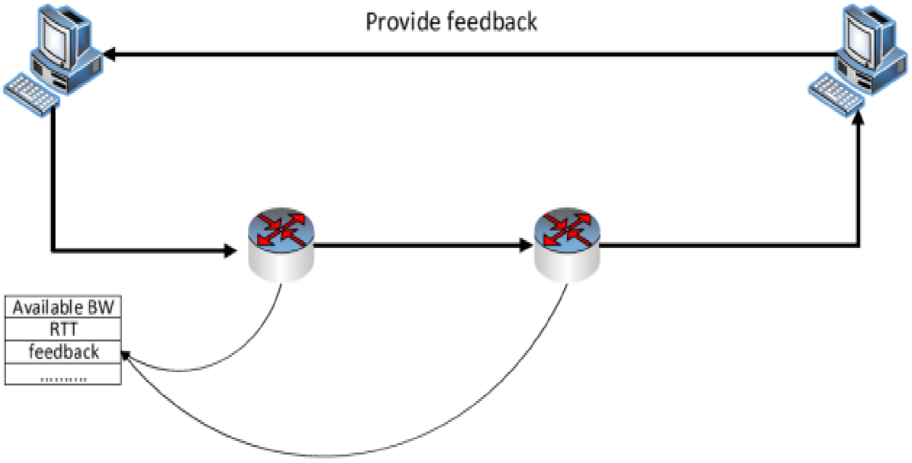

Figure 1. Router assisted congestion control

To calculate the change in flow sending rate, XCP calculates the aggregate bandwidth for each router. XCP uses the following equation to calculate the desired adjustment of the aggregate bandwidth [11].

$$R_l(t) = \alpha(C_l - y_l(t)) - \beta \frac{q_l(t)}{d} \tag{2}$$

In this equation, C_l is the capacity of the outgoing link, $y_l(t)$ is the rate of its outgoing traffic, $q_l(t)$ is the persistent queue during the previous control interval and d is the average RTT. The resulted aggregate feedback $R_l(t)$ can be positive or negative and is distributed among the traversing flows. The fairness controller uses the Additive-Increase/Multiplicative-Decrease (AIMD) principle to allocate the positive or negative feedback. Positive feedback is distributed equally among all flows and negative feedback is distributed proportionally to their current throughput. XCP has two disadvantages: first, the startup flow is working slowly, so the completion time of flow is not smooth for small flow [13]. Second, it requires per-packet calculation; this raises the problem of significant overhead [12].

RCP was developed by Nandita [12] with the aim of providing a simpler congestion control mechanism. RCP assumes that the sender uses a rate-based delivery mechanism. Equal to XCP, RCP requires calculating the aggregate rate to adjust the flow sending rate. RCP calculates the aggregate rate $R_l(t)$ once per interval control with the following equation:

$$R_l(t) = R_l(t-d)\left(1 + \frac{\alpha\big(C_l - y_l(t)\big) - \beta\left(\frac{q_l(t)}{d}\right)}{C_l}\right) \tag{3}$$

Where $R_l(t)$ is the common feedback rate, d is the average RTT and C_l is the capacity of the outgoing link. $y_l(t)$ is the aggregate outgoing traffic which was measured during the last control interval, $q_l(t)$ is the current queue occupation d is the update interval duration with d. It can be shown that RCP is locally stable if certain conditions for α and β are fulfilled [12]. Despite these improvements, XCP and RCP still face many challenges especially in queue support. The problem of the queue support in this system is that the protocol assumes only a single queue, contrary to the design of a high-end router that rarely has only one queue. The implementation can be tricky if there are thousands of flows passing through the routers with different characteristics.

Finally, the third solution is to use a router to detect congestion along the path and provide information to the end system. The example of this category is the Open Box Protocol (OBP) [14]. OBP uses a collaboration router to identify network resources along the path and deliver this information to the end system. The end system made the congestion control decision using the data received from the router. The following equation shows how the sending rate is adjusted. In OBP, The initial sending rate $W(t_0)$ depends on the available bandwidth $AB(t_0)$, the capacity $CB(t_0)$ at the narrow link and the constants *α* and *ß* [14].

$$W(t_0) = \alpha * AB(t_0) + \beta * CB(t_0) \tag{4}$$

Every time a new ACK packet is received, the feedback information inside the packet is used to make adjustments in transmission rate. OBP claims that its computation is simpler than XCP and RCP. However, the sender on OBP should create decision of adjusting the sending rate using only the information on the router along the path. This information is still a local category because the recipient does not understand the actual network conditions. Consequently, the OBP should always be careful in increasing and decreasing the delivery rate to maintain the network stability.

In general, RACC can improve network performance. However, the RACC method is run by using a distributed framework as in traditional networks has some drawbacks. It requires a router (s) that supports signaling bandwidth of the set of parameters of the sending rate [15]. The presence of incomplete information about network conditions makes the issue of efficiency and stability. The congestion control scheme with network support becomes inefficient so that a global information provider mechanism is needed that can be used by congestion control mechanisms to control network congestion. Congestion control using global information has been proposed by Monia et al. [16] proposed OpenTCP, a TCP adjustment dynamics framework based on SDN. The sending rate adjustment in OpenTCP is global-based information managed by the controller. OpenTCP has filled the SDN application gaps in the congestion control field. However, OpenTCP is a new architecture and needs several modifications on some elements of the network such as having to modify the source, forwarding nodes and controllers. OpenTCP also has no specific target rate and adjustment methods. Lingyun et al. [17] present multiple active queue management algorithms. The algorithm is executed according to the location of the congestion on the network. This algorithm is adaptive to link condition.

The link condition is detected by monitoring the statistical center, if there is a congestion link, then the information flow on this link is transferred by the OpenFlow controller. This method claims to be effective for congestion control. Nevertheless, this algorithm does not fully take into account the condition of network globally. Yao [18] proposed an algorithm called Software-Defined Congestion Control (SDCC). This approach has the characteristics of centralized control and can get a global topology for integrated network management. SDCC can optimize link utilization to control network congestion. However, SDCC still only considers network performance and does not pay attention to flow performance. Congestion control using SDN approach is still in early development stage. The proposed works have not discussed the weaknesses, and performances improvements of router assisted control in traditional networks. The most extensive work for the field of congestion control in the SDN is proposed for the data center as in [19]-[21].

In this paper, we proposed a new Router Assisted Congestion Control (RACC) mechanism that we call Path Associativity Centralized Explicit Congestion Control (PACEC). PACEC works on the SDN framework to overcome the weaknesses of the RACC in traditional Internet by providing global network information. We use comprehensive information to improve the accuracy of feedback to determine the source sending rate. By using SDN technology, congestion control utilizes the global knowledge of the network in its decision making. Data monitoring collects network information, and this information is used by the controller to make centralized decisions in response to changing network conditions [22]. This approach produces accurate information so that the sender's sending rate can be customized appropriately according to network conditions.

PACEC calculates the rate by involving all the nodes along the connection path through which the information flows. Therefore, the sending rate of the information flow does not need to change as long as it still passes through the same path and the update timer of the controller has not ended. PACEC can also set the source sending rate starting at high-speed so that network resources can be used more efficiently. This proposed mechanism is the novelty of this research in the congestion control domain. The rest of this paper is structured as follows: section 2 explains the related works within the topic of the RACC. Section 3 describes the design of the proposed mechanism in this paper. Section 4 describes the simulation results of the proposed mechanism, and finally, Section 5 describes the conclusions and possible future research directions.

2. PROPOSED METHOD

This section provides an overview of the PACEC draft [23]. To better understand the motivations behind PACEC, let us remember that RACC uses the basic equations we have written in Equation (1) to determine the aggregate rate on a link. From that equation we can see that information of available bandwidth covers only within local. Consequently, an RACC router will need to coordinate with other routers when implemented into networks with multiple routers. To avoid this drawback, PACEC implements a centralized congestion control mechanism, while the RACC router on the traditional Internet calculates the aggregate rate for a link only. PACEC can calculate the aggregate rate for a path, where the path has been provided to distribute the flow from the source to the destination. With this mechanism, the flow is guaranteed by route and rate when the flow is allowed to enter into the network. Our proposed method is described in the following sections.

2.1. Path Rate (R_p)

Similar to the case of the RACC routers, PACEC requires calculation on aggregate rates to adjust the sending rate. The difference is that the RACC calculates an aggregate rate for one link only. PACEC calculates the aggregate rate for a path. We call aggregate rate on PACEC as path rate (R_p). To get the path rate, we consider a network model whose topology is characterized in Figure 2. The network model is a path consisting of several links (*1, 2,..., H*). The whole link is connected to a centralized controller. End system as the source of information flow is connected to the ingress switch as the gateway to the network. In this case, nodes C_1serves as ingress switch. The source f_i has an associated sending rate of r_i. Flow is transmitted from source to destination via a path that has capacity C_p.

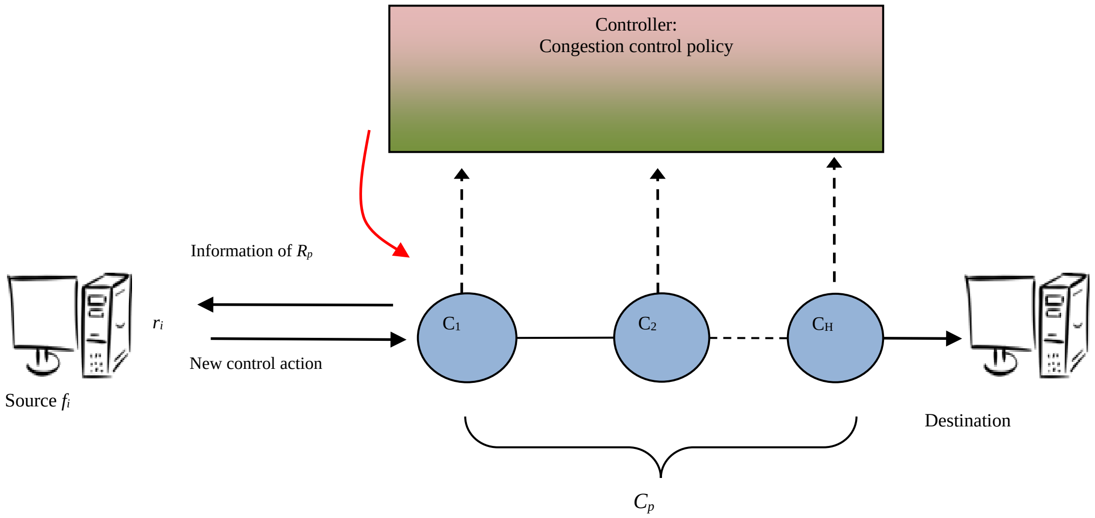

Figure 2. Network model of PACEC

A path containing a series of *H* link(s) will have the path rate (R_p) formulated by the following equation:

$$R_p = \min_{i=1...H} C_i\,.(1-\Gamma_i) \tag{5}$$

Γ_i is the *i*_th average link utilization of collected by controller *i*, where $\Gamma_i = \frac{y_{i+q_i}}{c_i}$, y_i is the *i*_th average traffic link, q_i is the *i*_th queue link, C_i is the*i*_th link capacity. In contrast to RACC in the traditional Internet, PACEC uses Γ_i, which is a utilization matrix whose elements are global information network. Information about Γ_i is obtained by the controller globally from each switch incorporated in the controlled network. The main advantage of Γ_i is that it can be a precise solution for R_p, which is not under-estimated nor over-estimated. Sub-section 3.2 describes how PACEC obtains such information.

2.2. *R*R_p Updating

Network conditions vary from time to time. Therefore the controller needs to update the network information continuously to deliver an effective policy. The controller updates the information of R_p in each update period T_c and the controller will provide updated information to the ingress switch. R_p is affected by the switch utilization Γ_i. R_p changes if there is a change in Γ_i. In this section, we describe how the controller obtains the updated information on Γ_i. Suppose there are H switches that send the updated information to the controller in constant time interval Δt. Thus, each delivery of data is done at$Ts = (\Delta t, 2\Delta t, 3\Delta t, \dots\dots k\Delta t)$, where T_s is time update switch. The controller processes the information for the controller at a period T_c, with $T_c = m.T_s$ so that every time the update is performed, the controller collects information m times with interval T_s. If the total observation time is T and during this time interval there are k times where information is collected, then the controller requires an update of $\frac{k}{m}$ times that are done at time interval index $s_1, s_2, \dots .s_{\frac{k}{m}-1}, s_{\frac{k}{m}}$, where $\frac{k}{m}$ is total information at observation time T as shown in Figure 3. Here s_c denotes the index of time interval where a controller collects information from each switch.

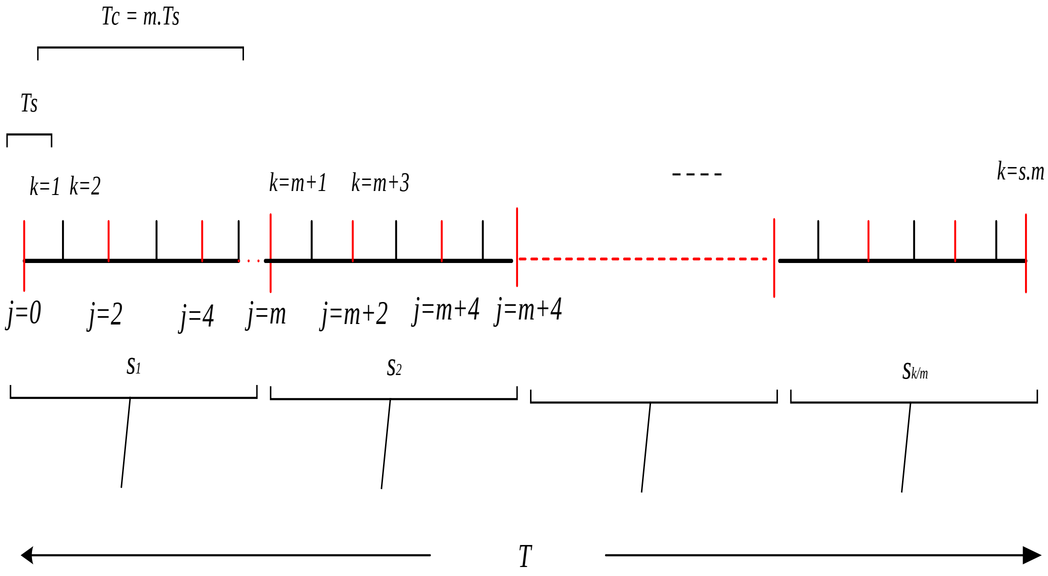

Figure 3. Information update

Link utilization in each time index s_c for each link i can be calculated as

$$\Gamma_i(s) = \frac{1}{m}\sum_{j=1}^{m}\frac{y_{i,j+(s-1)m} + q_{i,j+(s-1)m})}{c_i} \tag{6}$$

Where $y_{i,j}$ represents the packet passing through a switch i at time index j $(t = j.\Delta ts)$, whereas $q_{i,j}$ represents the queue length at switch i at j $(t = j.\Delta t)$. The y_i and q_i are collected every time interval Ts to $k.Ts$. The $y_{i,p}$ is the traffic on node I from time interval to p, where p=1,2, ... k. Furthermore, R_p for every update period s can be written as

$$R_p(s) = \min_{i=1\dots H} C_i.\left(1 - \Gamma_{i,s}\right) \tag{7}$$

or

$$R_p(s) = \min_{i=1\ldots H} C_{i,} . \left(1 - \left(\frac{1}{m}\sum_{j=1}^{m} \frac{y_{i,j(s-1)m} + q_{i,j(s-1)m})}{c_i}\frac{1}{m}\right)\right) \tag{8}$$

Available bandwidth on each link as proposed in Equation (8) differs from traditional available bandwidth model given in Equation (1). In this mechanism, we continuously update the available bandwidth based on actual utilization and queue length at every node in the path as given in Equation (5). Traditional routing, on the other hand, only relies on a specific node as given in Equation (1).

2.3. Sending Rate Update

Sending rate update aims to adjust the source sending rate (r_i) based on current network conditions. As mentioned before, the source sending rate adjustment on PACEC is based on the path rate (R_p) calculated by the SDN controller. Every T_C period, the controller updates R_p. Then in each of these periods, the source gets a new rate corresponding to R_p, which is informed by the controller to the source through the ingress switch. The new rate is independent of the previous rate. If there is N flow to be transmitted on a path and each flow is given the same rate, then R_p on each controller update is divided equally. The following equation shows how the sending rate (r_i) is adjusted.

$$r_{i_{max}}(s) = \frac{\min\limits_{i=1\ldots H} c_i . \left(1 - \left(\sum_{j=1}^{m} \frac{y_{i,j(s-1)m}+q_{i,j(s-1)m})}{c_i}\right)\right)}{N} \tag{9}$$

Equation (9) denotes the maximum sending rate that can be provided to transmit a flow so that the source sending rate r_i meets the limit$r_i \leq r_{i_{max}}$. Here, we conclude that to adjust the sending rate at the source, PACEC uses the following steps:

a. The SDN controller calculates resources on a path at each period Tc. This resource calculation covers bandwidth availability which is then converted into path rate (R_p).
b. Ingress switch receives information from the SDN controller for each period of controlling T_c. Ingress switch passes this information to the source.
c. The source adjusts its sending rate based on the information received from ingress switch. The source transmits the flow at the rater_i , corresponding to the rate-sharing algorithm at the source. The r_i cannot exceed R_p, as the upper limit. The source will stream the flow with the same rate until there is a change of information rate. If there are N flows then $\sum_1^N r_i \leq R_p$.

3. RESULTS AND ANALYSIS

In this section, we report the simulation results collected with our implementation of PACEC in the Mininet Simulator. We compared the simulation results of the proposed method with other congestion control mechanisms such as RCP. Here we compared PACEC with RCP due to the reason that PACEC is a rate based congestion control such as RCP. We also compare PACEC with TCP to see PACEC improvements overs important congestion control protocol on the Internet. To measure the performance of the congestion control mechanism, we evaluated the throughput, efficiency, smoothness, and fairness. As a reference scenario, we write the simulation parameters in Table 1.

Table 1. Parameters of Simulation

parameter	value
Bottleneck Link	C=100 Mbps
Buffer Size	B [10,100]
Simulation Time	100 seconds
Emulator	Mininet
Controller	Ryu
Topology	Abilene
Packet Size	1500 Bytes
RTT	50 ms
Operating system	Ubuntu

3.1. Throughput

Figure 4 shows comparison throughput achievement of congestion control scheme. Our simulation is conducted by entering two kinds of flows consisting of large flow (flow A) and small flow (flow B) to the network. We measure the throughput for both flows. The simulation resulted in the average of throughput for PACEC is 48.9 Mbps, RCP is 46.3 Mbps, and TCP is 31, 45 Mbps for flow A. The simulation results for flow B obtained the average of throughput; PACEC is 12.27 Mbps, RCP is 11.58 Mbps, and TCP is 7.87 Mbps. In this case, PACEC outperformed RCP and TCP. PACEC increased mean throughput achievement over RCP by 5.7% and TCP by 55.7%.

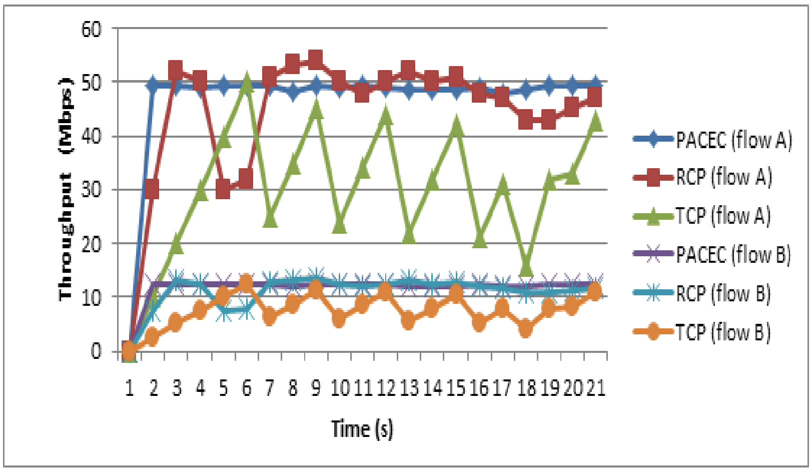

Figure 4. Throughput achievement of congestion control scheme

3.2. Efficiency

The efficiency of the congestion control mechanism can be determined through the power parameters, i.e., the ratio between throughput and delay [24].

$$power = \frac{throughput^{\alpha}}{delay} \tag{10}$$

We summarize the simulation results in Table 2. We can see that PACEC has the highest power value compared to RCP and TCP. That indicated that PACEC is more efficient than RCP and TCP.

Table 2. Efficiency comparison of PACEC, RCP, and TCP

Congestion control scheme	Throughput (Mbps)		Delay (ms)		Power (α=1)	
	Flow A	Flow B	Flow A	Flow B	Flow A	Flow B
PACEC	48,9	12,2	54,2	12,9	0,9	0,9
RCP	46,3	11,5	57,1	14,6	0,8	0,8
TCP	12,2	7,8	84,1	17,1	0,4	0,5

3.3. Smoothness

Smoothness is an essential feature of congestion control. Here we declared smoothness as the ratio of the rate change between two successive update periods to the previous rate and written as [25]

$$Th_{m_i} = \frac{|x_i - x_{i-1}|}{x_{i-1}} \tag{11}$$

where x_i is the average throughput during the *i-th* interval for the flow (each range is T_c sec). A flow smoothness index is defined as the mean throughput-change over its lifetime ratio, where T is the total time interval during the simulation. Smaller smoothness indices show smoother throughput changes [26].

$$\overline{Th_m} = \frac{\left(\sum_{i=1}^{N} Th_{m_i}\right)}{T} \qquad (12)$$

We do a simulation of 10 different flows, a small flow having sizes (1-10 Kbps), medium (10Kbps-1Mbps), and large (1-100Mbps) within 100 seconds. We declare the throughput fluctuation with the smoothness parameter as in (12). Table 3 shows the comparison of smoothness index for each flow. The smoothness index captures the time series of rate changes. A smaller smoothness index indicates smoother throughput change for a flow.

Table 3. Comparison of Smoothness (Th_m)

Flow id	Flow type	PACEC	RCP	TCP
1	small	0,05	0,09	0,13
2	medium	0,03	0,04	0,10
3	medium	0,02	0,09	0,17
4	medium	0,02	0,09	0,16
5	medium	0,02	0,03	0,06
6	large	0,02	0,08	0,06
7	large	0,03	0,04	0,10
8	large	0,02	0,08	0,11
9	large	0,02	0,07	0,12
10	large	0,02	0,06	0,18
	average	0,025	0,067	0,12

The simulation results show that smoothness index for PACEC varies from 0.02 to 0.05, it varies from 0.03 to 0.09 for RCP, and varies from 0.06 to 0.18 for TCP. Based on these values, indicating that PACEC is smoother compared with RCP and TCP.

3.4. Fairness

Fairness is used to decide whether the user or application receives a fair share of system resources. Here, we use the mathematical definition from Jain and Chiu [26]. Fairness can be written as

$$J(x_1, x_2, \ldots\ldots, x_n) = \frac{(\sum_1^n x_i)^2}{n.\sum_1^n x_i^2} \qquad (13)$$

where x_i is the throughput of flow *i* and *n* is the sum of flow. We consider the medium-term fairness of PACEC. To evaluate medium-term fairness, we obtain the average throughput of PACEC flows over the entire time of simulation (100 seconds). We simulated 20 identical flows (10 Mbps). Based on equation (13), we got fairness index for 20 identical flows as shown in Figure 5. The fairness index of PACEC is smaller than RCP and RCP. The fairness index of PACEC is 0.99 (close to one) which indicates that the throughput assignment for a competing flow in PACEC is fair.

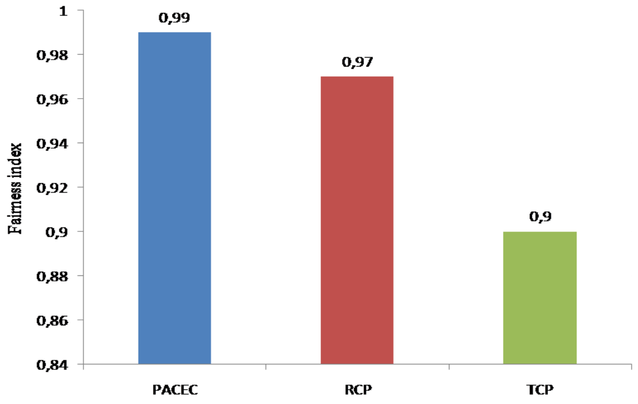

Figure 5. Fairness index of congestion control scheme

4. CONCLUSION

In this paper, we designed a new Router Assisted Congestion Control (RACC) mechanism that works with the SDN framework. This mechanism is designed to overcome weaknesses in traditional Internet networks that cannot provide global information networks. We propose a scheme that uses explicit information from the controller as a network policy determiner. Based on the information from the controller, the sender can adjust the sender rate according to the network conditions. The sender does not need to adjust the sending rate incrementally. We have demonstrated through computer simulation that the scheme is able to use the network bandwidth more efficiently and more consistently, and also able to maintain fairness of any flow that requests network services. For future works, we plan to integrate this scheme with admission control and bandwidth allocation mechanism.

ACKNOWLEDGMENTS

This work is partially supported by the Directorate of Research and Community Service, the General Directorate of Research and Development Strengthening, the Ministry of Research, Technology, and Higher Education of the Republic of Indonesia under the research contract1603/K4/KM/2017.

REFERENCES

[1] Van. J. *Congestion Avoidance and Control*. In SIGCOMM '88. California, USA. 1998; 18(4): 314–329.

[2] John N. Congestion control in IP/TCP internet. *ACM SIGCOMM Computer Communication Review*. 1984; 14(2); 11-17.

[3] Manjur K, Mosleh M, Faiza R. QoS Design Consideration for Enterprise and provider's Network at Ingress and Egress Router for VoIP Protocol. *International Journal of Electrical and Computer Engineering* (IJECE). 2016; 6(1): 235-241.

[4] Nandita D, Masayoshi K, Rui Z, Nick M. *Processor Sharing Flows on the Internet*. Thirteenth International Workshop on Quality of Service (IWQoS). Passau, Germany. 2005; 271-285.

[5] Ramanjaneya R, Chenna R, Padmavathamma M, Performance Enhancement of TCP Friendly Rate Control Protocol over Wired Networks. *International Journal of Electrical and Computer Engineering* (IJECE).2016; 6, (6):2949-2954.

[6] Allman M, paxon V Stevens W, "TCP Congestion Control", RFC2581, 1999.

[7] Sally F. TCP and explicit congestion notification. *ACM SIGCOMM Computer Communication Review*. 1994; 24(5): 8-23.

[8] Juan P, Mounir H. Enhancements on Router-Assisted Congestion Control for Wireless Networks. *IEEE Transactions on Wireless Communications*. 2008; 7(6): 2253 – 2260.

[9] Ramakrishnan K, floyd S, Black D. The Addition of Explicit Congestion Notification (ECN) to IP. RFC 3168, 2001.

[10] Allman M, Jain A, Sarolahti P. Quick-Start for TCP and IP. RFC 4782, 2007.

[11] Dina K, Mark H, Charlie R. *Congestion Control for High Bandwidth-Delay Product*. Proceedings of The Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. New York, USA. 2002; 32(4): 89-102.

[12] Nandita D, Nick. M. Why Flow-Completion Time is the Right Metric for Congestion Control and Why This Means We Need New Algorithms. *ACM SIGCOMM Computer Communication Review*. 2006; 36(1): 59-62.

[13] Simon H, Michael S, Jochen K, Chawapong S. *Quick-Start and XCP on a network processor: Implementation issues and performance evaluation*. In Proc. High Performance Switching and Routing. Shanghai, China. 2008; 241 – 246.

[14] Paulo L, Saverio M, Edmundo M. *Open Box Protocol (OBP).* Proceedings of the 3rd international conference on High-Performance Computing and Communications. Houston, Texas. 2007; 496 – 507.

[15] Slim B, Thomas B, Alexandre P, Gwenaël R, James R. *Statistical bandwidth sharing: A study of congestion at flow level*. ACM SIGCOMM Computer Communication Review. New York, USA. 2001; 31, (4): 111–122.

[16] Monia G, Soheil H, Yashar G. *Rethinking End-to-End Congestion Control in Software-Defined Networks*. in proc HotNets-IX, Redmond, Washington. 2012; 61-66.

[17] Lingyun L, Yang X, Haifeng D. *OpenFlow Control for Cooperating AQM Scheme*. Signal Processing (ICSP), IEEE 10th International Conference. Beijing, China. 2010; 2560–2563.

[18] Yao H, Ting P, Lianming Z. Software-Defined Congestion Control Algorithm for IP Networks, *Hindawi Scientific Programmin.* 2017; 2017; 1-9.

[19] Masoumeh G, Behzad. G. *Congestion control in software-defined data center networks through flow rerouting*. in proc. Electrical Engineering (ICEE), Tehran, Iran. 2015.

[20] Yifei L, Shuhong Z. *SDN-based TCP congestion control in data center networks*. *Computing and Communications Conference (IPCCC), 2015 IEEE 34th International Performance*, 2015; 1-7.

[21] Lu Y, Ling Z, Zhu S, Tang L.SDTCP: Towards Datacenter TCP Congestion Control with SDN for IoT Applications", *in Sensors.* 2017; 17(1): 109.

[22] I.D Irawat, Sun Y, Hariyani, Hadiyoso S. Link Aggregation Control Protocol on Software Defined Network. *International Journal of Electrical and Computer Engineering* (IJECE). 2017; 7(5): 2706-2712.

[23] Sofia N, Adit K, Hendrawan, Udjianna P. *Path Associativity Centralized Explicit Congestion Control (PACEC) for SDN*. Control, Electronics, Renewable Energy and Communications (ICCREC), 2017 International Conference on ICCEREC. Yogyakarta, Indonesia. 2017.

[24] Raj J, Ramakrishnan, Dah-Ming C. *Congestion Avoidance in Computer Networks with a Connectionless Network Layer*. Proceedings. Computer Networking Symposium, Washington, DC, USA. 1987.

[25] Chen K, Nahrsted K. *Limitations of Equation-based Congestion Control in Mobile Ad Hoc Networks*. Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on, Tokyo, Japan, Japan. 2005.

[26] Jain R, Dah-Ming Chiu, W. Hawe. A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared System", *Research Report TR-301*, 1984.

BIOGRAPHIES OF AUTHORS

Sofia Naning graduated from Brawijaya University, Indonesia, in 1995 and received the M.E. degree from the Bandung Institute of Technology (ITB), Indonesia, in 2004, both in Electrical Engineering, majoring in Telecommunication Engineering. Currently, she is pursuing the doctoral degree at the same university. Since 1999, she joined the Faculty of Electrical, Telkom University, Indonesia, as a Lecturer. Her research interests cover traffic engineering and software-defined network.

Adit Kurniawan graduated from the Bandung Institute of Technology (ITB), Indonesia, in 1986 and received the M.Eng. degree from RMIT, Australia, in 1996 and the Ph.D. degree from the University of South Australia, in 2003, both in telecommunication engineering. He became a faculty member of the Department of Electrical Engineering, ITB, in 1990. His research interests are antenna and wave propagation, cellular communication system, and radio communication. He is currently a Professor and serves as the Head of Telecommunication Engineering Department at the Bandung Institute of Technology.

Hendrawan graduated from the Bandung Institute of Technology (ITB), Indonesia, in 1985 and received the M.Sc. degree from University of Essex, UK, in 1990and the Ph.D. degree from the University of Essex, UK, in 1994, both in telecommunication engineering. He became a faculty member of the Department of Electrical Engineering, ITB, in 1987. He is a member of IEEE Communication Societies (ComSoc). His research interests are teletraffic engineering and multimedia network.

U. S. Pasaribu is an Associate Professor in Department of Mathematics and Natural Sciences, Bandung Institute of Technology, Indonesia. She graduated from the Bandung Institute of Technology (ITB), Indonesia, in 1985and received her Ph.D. from the European Business Management School, University of Wales, Swansea, UK. Her research interests include stochastic process, space-time analysis, and Markovian models.