

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 10(3):403-419, June, 2015.

Performance Comparison of TCP Spoofing and End to End Approach to Enable Partial QoS on IP Based Network

Y. Suryanto, R.R. Nasser, R.F. Sari

Y. Suryanto*, **R.R. Nasser**, **R.F. Sari**

1. Department Electrical Engineering Faculty of Engineering,
University of Indonesia, Depok, 16424, INDONESIA.

yohan.suryanto@ui.ac.id, rizki.reynaldo@ui.ac.id, riri@ui.ac.id

*Corresponding author: yohan.suryanto@ui.ac.id

Abstract: Implementation has a purpose to give adequate guarantee for multimedia application to be able delivered according to the priority and the class of services. But basically, end to end QoS guarantee is very difficult to be realized, especially when it involves a lot of operators with a variety of interconnection networks such as Internet. To overcome that difficulty of implementing end-to-end QoS in IP-based networks, we propose a partial QoS approach through TCP spoofing technique. Partial QoS is implementation of QoS subset like bandwidth parameter in certain ip based network segment. TCP spoofing is a technique to intercept TCP connection between user and content server to be further manipulated. Each TCP connection will be intercepted by spoofing gateway and will be adjusted to the appropriate window size parameter that has been approved for each user. Spoofing Gateway will forward the request from the client to the content server and change the original TCP protocol to TCP Linux Highspeed. Our Simulation results, using NS-2.35, show that for some cases of Partial QoS through spoofing technique produced better performance, such as completion time and average throughput for each user class priority, compared to the end-to-end QoS approach. Even in cases where the Internet network delay product characteristics are relatively high, partial QoS spoofing technique has a much better performance than end-to-end QoS.

Keywords: TCP Spoofing, partial QoS, End to End Approach, Performance Comparison, Wireless Network, Internet, IP based network.

1 Introduction

Rapid growth of Internet users today, bringing a significant impact to the network load including access network. A lot of users compete to get access to the Internet network through ISP. They are willing to pay higher to get a higher class of service. For that, ISPs need to implement QoS [1] parameters, such as maximum bandwidth, which is in accordance with the level of service provided.

Setting different priority level needs QoS implementation for full or partial QoS parameters. [2] It requires appropriate bandwidth settings for the level of service provided by the operator. Unfortunately the implementation of the end-to-end QoS for the Internet network is very difficult, because many networks interconnected without common QoS standard and there is no single operator which controls the end-to-end Internet network.

The technology used, especially in the access network, does not always support the QoS capabilities. For example in the case of WiFi 802.11 which is widely used today as an access network, it does not allow the operator to provide the service class based on adequate bandwidth. Through user requires a priority based on the level of service they have paid to the operator. Although setting QoS in 802.11 networks can also be done by implementing 802.11e standard, but these techniques require modification of the media access layer (MAC layer) of WiFi 802.11. [3]

Priority is generally given to the user in the form of different access speeds for each class of service or total bytes accessed by users each month. The 802.11 WiFi network can not

distinguish between the speed per user because there is no mechanism in layer 2, which could limit the throughput per given time frame. While the Remote Access Server (RAS) which is commonly used as a WiFi gateway, can only limit the bandwidth capacity that is used for a relatively long period of time, such as for a period of a day or a month.

We propose a bandwidth setting at TCP level because it is independent to hardware of the access network being used by the customer. Due to the fact that the TCP connection is end to end from the client to the server, it is not easy to implementing bandwidth control through TCP manipulation. Therefore, we propose the use of spoofing techniques to overcome these problems. TCP connection will be intercepted on the gateway, which is spoofing server, so that a subset of the QoS parameters can be added. This connection is then converted into a highspeed TCP in order to be utilize as quickly as possible the available capacity backbone networks that has been designed using traffic engineering process. [4]

Partial spoofing techniques to enable QoS that we propose, can be applied not only on the wireless network, but also can be applied to the wireline network. This is in contrast to the application of the 802.11e EDCA that can only be applied to the WiFi network by changing the 802.11 MAC layer. [3]

We simulate the application of the partial QoS wired or wireless network by implementing spoofing techniques using NS-2.35, a tool commonly used to simulate TCP performance. [5,6] The performance of partial QoS through spoofing approach will be compared with the performance of end-to-end approach. Although partial QoS can be applied also to the UDP protocol, in this simulation we only apply to TCP protocol. Therefore spoofing technique used is TCP spoofing.

Further in this paper, we will discuss the theoretical background in section II, including the QoS and needs [4], Internet traffic flow model, TCP, Highspeed TCP and throughput analysis. [7] The proposed solution is discussed in section III. Meanwhile the partial QoS simulation scenarios through TCP spoofing approach will be presented in section IV. Performance analysis of partial QoS using TCP spoofing techniques will be presented in section V, and then the conclusion in section VI.

2 Supporting theory

2.1 QoS Overview

In communication using the Internet network, network bandwidth generally has a maximum limit for distributing multimedia content over time. Specific content in order to be delivered according to the priority level required for the implementation of QoS. [8] QoS implementation is possible, because even if the user wants all the applications to arrive at the destination as quickly as possible, however:

1. Not all applications must be delivered within limited time duration.
2. Only certain applications that require limited time.
3. Only certain applications that sensitive to jitter.
4. Not all content is very sensitive to packet lost.

On IP-based networks, QoS services provided by the operator can be measured according to the following performance criteria:

- a. Throughput
- b. Service availability
- c. Packet loss

- d. Delay
- e. Variation of Delay or jitter

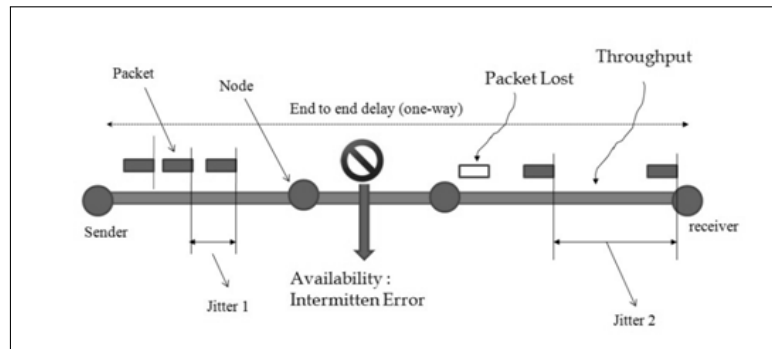


Figure 1: QoS Parameter

Quality of service (QoS) means to guarantee the commitment of those criteria according to the awarded contract. The given contract may be based on something that is explicitly mentioned, such as network access at the speed of 2 Mbps. But it can also be something that is not explicitly presented, such as the multimedia package, 2 Mbps office package, 2 Mbps of easy browsing and so on.

Each group applications such as voice, video streaming and browsing has a different sensitivity to the QoS parameters. Some approaches have been proposed to improve QoS. ([1]- [4], [9]- [12]) In practice, users are also difficult to determine how many time in a month they will use the voice applications, video streaming, browsing and download. Therefore in practice, only a subset of QoS parameters that often become criterion, generally only a maximum throughput that can be obtained by the customer when connected to the network.

2.2 Internet Application Traffic Flow Model

Commonly most popular Internet activities are based on client-server traffic model. List of the most popular activity on the internet, can be categorized in the following activities:

1. Email : webmail like gmail or yahoo, and also offline email like outlook.
2. Browsing : common website server located at Data Center.
3. Search Engine : google, yahoo, etc
4. Social Networking : facebook
5. Video streaming : youtube
6. Audio streaming : mp3 server
7. Audio Communication : skype
8. Chatting : Text message conversation such as YM and whatsapp.

When explored further, such activity has a traffic pattern from the client to the server, not the pattern of the user to the user directly. That means most of the Internet traffic in the network is a communication from the client to the server which usually located in a server farm. Traffic patterns just as hub-spoke or star topology is depicted in Figure 2 below.

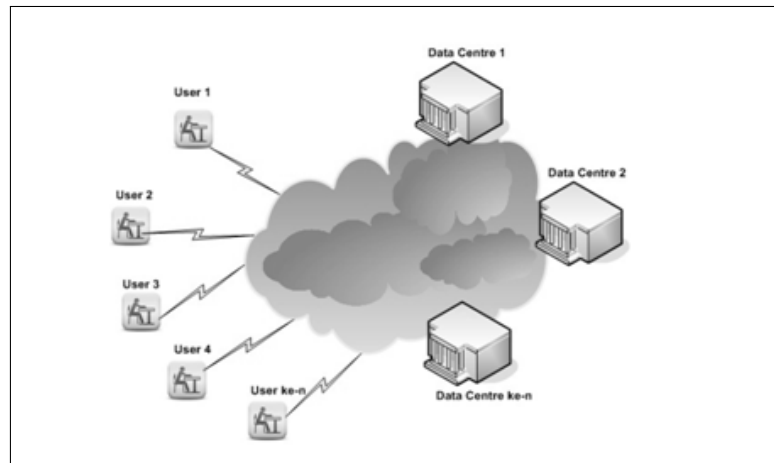


Figure 2: Traffic Flow Client-Server Model

With data flow pattern from the server farm to the customer or vice versa, the traffic flow can optimally be intercepted on the access gateway. Connections from the access gateway to the server farm usually connected through the Internet network backhaul or backbone. The backbone capacity can be adjusted to follow design of the maximum traffic demand from customers to a server farm. With this pattern, partial QoS can be applied in access network.

2.3 Transmission Control Protocol (TCP)

Transmission Control Protocol (TCP) is a connection-oriented protocol which can be relied upon. It provides a transport service for applications that run in a network based on IP protocol.

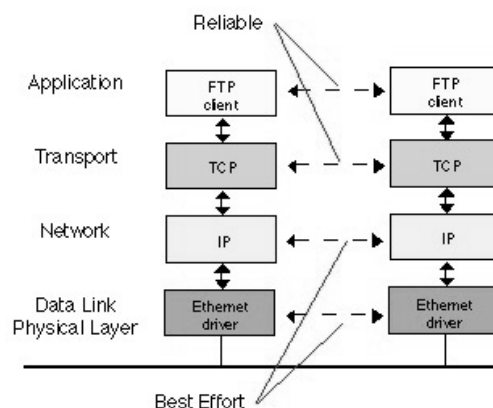


Figure 3: TCP providing reliable data transfer to FTP over an IP network using ethernet

TCP is a stream-oriented protocol, which means that the TCP protocol entities exchange data streams. Every byte of data from the application layer protocol or session layer is placed in the memory buffer and then transmitted by the TCP transport protocol in the data unit. Reliability of service TCP flow control is far more complex than UDP, since UDP only provides

best effort service. To implement such a reliable service, TCP protocol uses a timer to ensure reliability and to synchronize between the two ends of the TCP connection.

For most networks approximately 90% of current traffic uses this transport service. It is used by such applications as telnet, World Wide Web (WWW), FTP, electronic mail. The transport header contains a Service Access Point which indicates the protocol which is being used (23=telnet, 25=mail, 69=TFTP, 80=WWW).

There are many kind of TCP:

- TCP Selective Acknowledgments (SACK)
- TCP Reno
- TeP Vegas
- HSTCP (HighSpeed TCP)
- XCP (eXplicit Control Protocol)

Some paper suggest the way to improve TCP performance ([5], [13]- [15]), in this paper we propose a TCP spoofing technique that uses TCP normal in access network and forwards packets received by the spoofing gateway to the server destination with highspeed connections TCP (HSTCP). [15] HSTCP is an update of TCP that react better when using large windows on high bandwidth, high-latency networks. HSTCP alters how the window is opened each round trip and closed on congestion events as a function of the absolute size of the window. [15]

2.4 Average Throughput and Completion Time

Performance comparison between partial QoS through spoofing techniques and QoS through end-to-end approach is done by looking at the average throughput and completion time for each user using both methods. Completion time or transfer time is the time the last byte is received by the receiver subtracted by start time packet sent by the sender. Meanwhile, in our NS2 simulation, we capture the throughput of $Th(i)$ on each TCPSink node for the unit time of $\Delta t(i)$. The example graph of $Th(i)$ for each $\Delta t(i)$ can be seen in Figure 4.

To make it easier to compare the throughput performance, we need the average throughput at time $t(i)$ during the transfer process. This can be obtained using equation (1):

$$Avg_Th(i) = \frac{\sum Th(i) x \Delta t(i)}{\sum \Delta t(i)} \quad (1)$$

Where $Avg_Th(i)$ is average throughput from the starting transfer time to $t(i)$. $Th(i)$ is throughput at $\Delta t(i)$.

for the example For Δt of 0.02 second, the average throughput for each time $t(i)$ based on the equation (1), the average throughput can be depicted in Figure 5:

To determine the average throughput as a whole, from the start transfer time until the transfer is complete, also can use the following equation:

$$Avg_Th = \frac{Packet\ Size \ x \ 8}{time\ completion} \quad (2)$$

Where Avg_Th is the average throughput since the transmitter starts sending packets until the receiver received the entire packet.

Meanwhile, the maximum capacity of TCP link can be calculated from the equation (3) below:

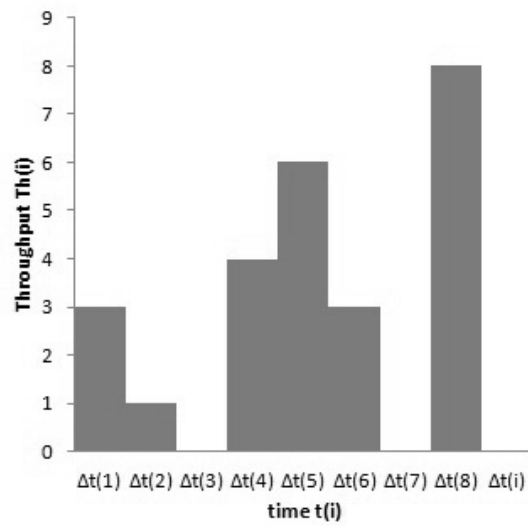


Figure 4: Throughput for each Δt captured at TCPSink

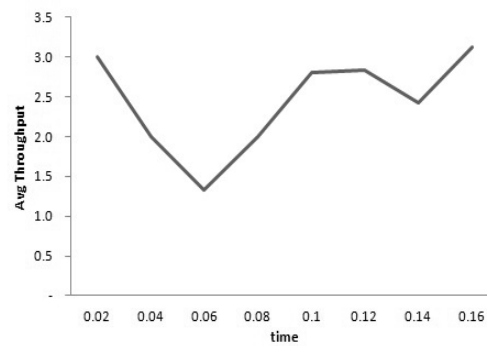


Figure 5: The normalized average Throughput

$$Maxs_Th = \frac{Window_Size(i) \times 8 \times Packet_Size(i)}{2 \times Delay(i)} \quad (3)$$

Where the Maxs_Th is the maximum throughput, Window_size(i) is TCP window size at frame i, Packet_size(i) is the TCP packet size at i and Delay(i) is the link delay at frame i.

3 Proposed solution

In this paper we propose the use of TCP spoofing techniques to enable the partial QoS. TCP Spoofing is a method to intercept TCP. TCP connection which has been intercepted can be converted into another TCP protocol type and can also be added to a subset of the QoS parameters. By utilizing the TCP spoofing, the partial QoS can be implemented independent to the type of network used. We can implement the partial QoS on wired or wireless networks.

Partial QoS is an implementation technique of the subset QoS parameters in a particular network segment. In this paper, a partial QoS implementation proposed is done at the network access. With a partial QoS through TCP spoofing approach, access technology currently installed can support the bandwidth control without the need to change the MAC layer. This is different from the approach of the 802.11e solution [3, 12]. Thus the TCP spoofing approach has better flexibility than the implementation of QoS by changing the access network hardware. Partial QoS configuration using spoofing techniques can be seen in Figure 6.

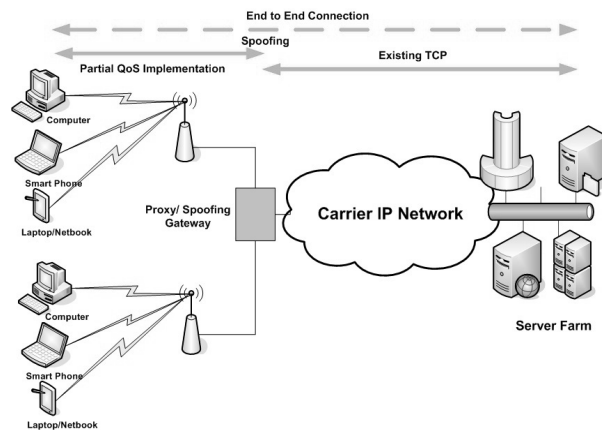


Figure 6: Partial QoS through spoofing technique

Spoofing gateway charged to intercept TCP connection from the user to the content server. In access network side from the user to the gateway spoofing, partial QoS will be applied. This is done by adding parameters window size and congestion window to the corresponding TCP connection. Then by spoofing gateway, packets will be forwarded to the destination server by first changing into highspeed TCP connection. [15]

4 Simulation scenario

This paper analyzed the completion time performance and average throughput between partial QoS using TCP spoofing and end-to-end approach. Simulations performed on wireless and

wireline networks. In the wireline network simulation, we use a dumb bell topology with 3 user configurations, each with priority classes low, medium and high. Node 3 is a spoofing gateway, while node 5, node 6 and node 7 are acted as content server as shown in Figure 7 below.

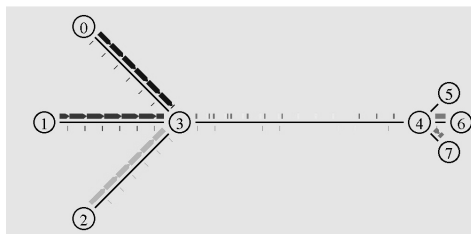


Figure 7: Topology of wireline simulation

For simulations in wireless networks, we used a single 802.11 base station with three users, each configured with priority low, medium and high. Node 1 is a spoofing gateway and node 0 act as a content server, as shown in Figure 8 below.

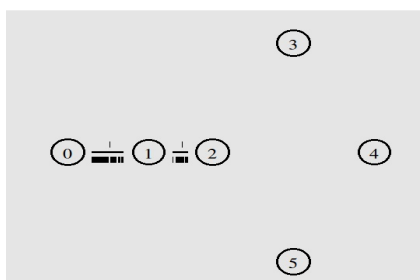


Figure 8: Topology of wireless simulation

Chosen to measure the performance is average throughput and completion time. This is selected because the QoS subset used is bandwidth parameter. Users will transfer the FTP file with file sizes M bytes to the server. Comparative performance begins by comparing the average throughput and the transfer time for spoofing techniques and end-to-end approach for single user without setting the window size. This is done to determine the speedup ratio of spoofing techniques for various total delay conditions. Validation is made by measuring the size of packets received by the server in TCP spoofing approach should be the same as the end-to-end approach.

The next comparison is done for a single user with a window size limitation. This simulation is run to determine the effect of window size restrictions that will be used as a QoS subset. The next simulation is the performance comparison of 3 users without the window size, which aims to measure the average throughput without additional subset of QoS. The next scenario is to make a comparison of simulated single user and multiuser with window size under varied backbone delay conditions. This is done to measure the QoS performance with partial spoofing approach compared with end-to-end approach.

In the NS2 simulations we set a subset of QoS parameters by adjusting the window size using `tc` commands:


```
set TCP0 [new Agent/TCP]
$TCP0 set packetSize_ $pktsize
$TCP0 set window_ $window1
$TCP0 set cwnd_ $cwnd1
$ns attach-agent $n0 $TCP0
```

TCP connection from the spoofing gateway to the server is converted into TCP highspeed linux with the command:

```
set TCP4 [new Agent/TCP/Linux]
$ns at 0 "$TCP4 select_ca highspeed"
$ns attach-agent $n3 $TCP4
```

Packets received by the spoofing gateway will be forwarded to the server with the command:

```
set relay1 [new Application/FTP]
$relay1 attach-agent $TCP3
set bw0 [$sink0 set bytes_]
$relay1 producemore [expr $bw0/$pktsize]
```

5 Result analysis

When we do a file transfer from the user to the server through a wireline network with dumb bell topology as shown in Figure 8, we can estimate the maximum transfer speed which can be achieved. The maximum throughput between the user and the server on end-to-end approach is the minimum value of TCP connection capacity and a media capacity of a segment which can be written as:

$$Th_maxs = Min(MaxsTCP, minB_segment) \quad (4)$$

Where are:

- Th_maxs : Maximum Throughput
- Maxs TCP : Maximum capacity of TCP connection according to equation (3)
- Min B_segment : Minimum bandwidth of each segment of transmission media between user and server.

Meanwhile, the maximum throughput between the user and the server on TCP spoofing approach is the minimum throughput of the access and backbone section that can be written as:

$$Th_maxs = Min(Th_maxs_access, Th_maxs_backbone) \quad (5)$$

Where are:

- Th_maxs access : maximum throughput of access network according to equation (4)
- Th_maxs backbone : maximum throughput of backbone network according to equation (4)

The network in TCP spoofing approach if broken down into two segments, which means that each segment delay \leq TCP connection end-to-end delay. Based on equation (3), equation (4) and equation (5), in general it can be concluded that the Th_maxs spoofing \geq Th_maxs end

to end. This can be seen for example in the first simulation, where we compare the average throughput between TCP spoofing approach and end-to-end approach.

For the case of access bandwidth set to 1 Mbps, access delay 10 ms and backbone bandwidth 2 Mbps, 1 Mbps bandwidth server, backbone and server delay 95 ms, we get:

$$\begin{aligned} \text{Th_maxs end to end} &= \min(0.76 \text{ Mbps}, 1 \text{ Mbps}) \\ &= 0.76 \text{ Mbps} \end{aligned}$$

$$\begin{aligned} \text{Th_maxs spoofing} &= \min(\min(8 \text{ Mbps}, 1 \text{ Mbps}), \min(0.84 \text{ Mbps}, 1 \text{ Mbps})) \\ &= 0.84 \text{ Mbps} \end{aligned}$$

The result show that average speed of the end-to-end approach \leq Th_maxs end to end, and average speed of spoofing \leq Th_maxs spoofing approach. The comparison result of the average throughput of the two approaches for the case is consistent with the above analysis. This can be seen in Figure 9.

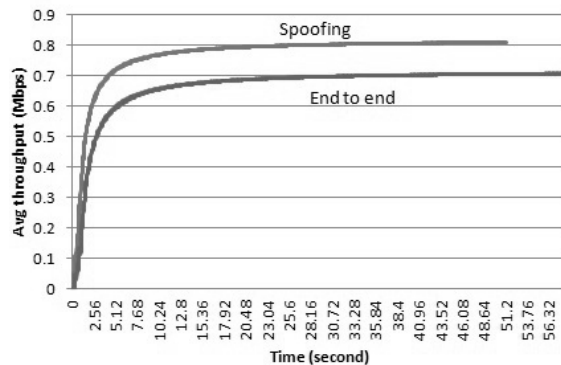


Figure 9: The average throughput comparison between end to end approach and TCP spoofing approach

In the simulation of TCP spoofing approach in wireline network topology as depicted in Figure 7, the backbone delay is varied, comparative relationship between the average throughput and total delay in the end-to-end approach and the spoofing approach can be seen in Figure 10. When the capacity of TCP \geq capacity of a media segment, according to equation (4) and (5), this shows that

the average throughput is limited by the available bandwidth limit of transmission media. The average throughput between TCP spoofing approach and the end-to-end approach is not very different. But when TCP connection capacity \leq segment capacity of transmission media, according to equation (4) and (5), this shows that the performance of spoofing approach is much better than the end-to-end approach.

Figure 11 confirm that the analysis of the completion time of the end-to-end approach \geq completion time of TCP spoofing approach. The completion time difference between the end-to-end approach and TCP spoofing approach for 5 MB FTP packets to the total delay is clearly according to equation (4) and (5).

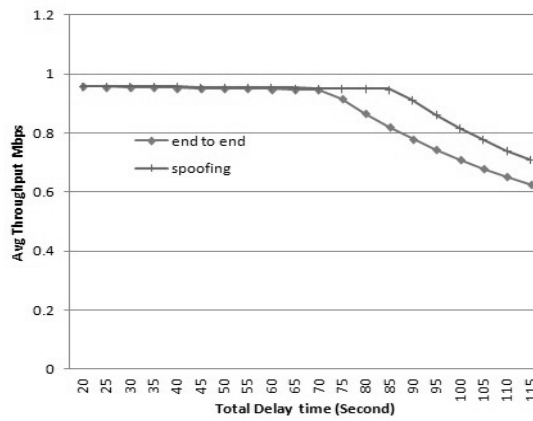


Figure 10: Average throughput to total delay

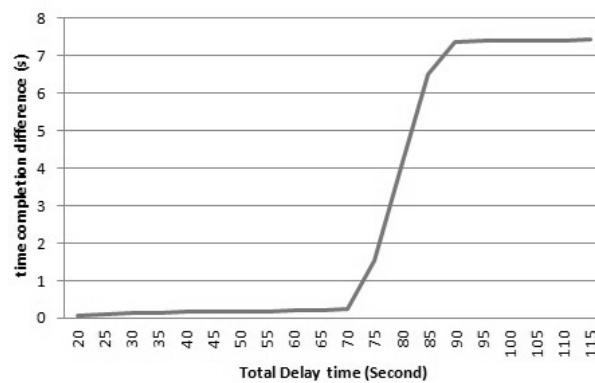


Figure 11: Completion time difference to total delay

As expected, the result shows that the average throughput of the end-to-end approach \leq TCP spoofing approach. The difference in the completion time becomes quite large when TCP capacity \leq available bandwidth of media segment. In this case, the total delay is 70 ms in which a TCP connection capacity approaching media segment bandwidth limitations, the completion time of the TCP spoofing is much faster than the end-to-end approach.

When we choose to run the simulation in condition where TCP connection capacity $<$ bandwidth of media segment, the completion time is more influenced by the capacity of the TCP connections. The comparison of the completion time to the FTP packet sizes, under the condition where bandwidth of media segment is still sufficient such as in the case when we select the delay of backbone and the server 105 ms and the access delay 50 ms. It is clear that completion time by spoofing approach is better than the completion time by the end approaches. This can be seen in Figure 12 below.

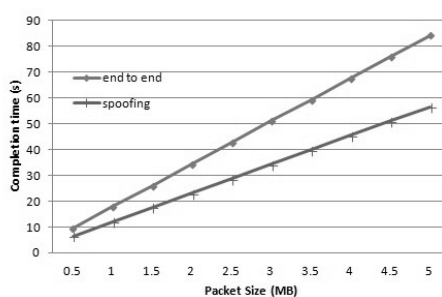


Figure 12: Comparison of completion time to the size of FTP file

In the next simulation, the access and backbone segments bandwidth is set to 10 Mbps so that the media connection does not become a bottleneck, because in a multi-user scenario with a QoS subset we want to show a completion time for each user priority based on the TCP connection capacity to the delay. The implementation of QoS subset for the low user priority is represented by the window size 5, the medium user priority with window size 10 and the high user priority with window size 15. For the condition of delay access 50 ms, delay backbone plus delay server is 55 ms, the average throughput graphs for each user priority using end-to-end approach can be seen in Figure 13.

It appears that the user with lower priority has the lowest average throughput and the user with high priority has the highest throughput. The duration of 5 MB file transfer time for each scenario with multi-user network delay for each priority users with end-to-end approach can be seen in Table 1. We can see that the completion time of user with high priority $<$ completion time of medium user $<$ completion time of low priority user.

The average throughput of each user priority with a media connection of 10 Mbps using TCP spoofing approach can be seen in Figure 14. The average throughput for each priority user using TCP spoofing approach, either for low priority, medium priority and high priority has a similar pattern with the end-to-end approach. However it can be seen from the graph, that the average throughput of TCP spoofing approach is higher than the end-to-end approach.

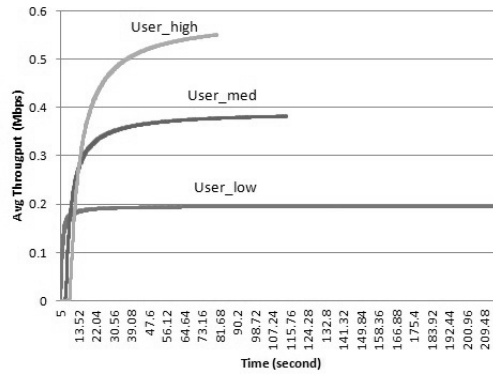


Figure 13: The average throughput for each user priority for end to end approach

Table 1: Completion time each User Priority for the End to End Approach

No	Description	Delay		completion time end to end		
		Access	Backbone +server	user 1	user 2	user 3
1	Scenario 1	10	15	52.68	26.46	17.72
2	Scenario 2	10	55	132.8	66.64	44.64
3	Scenario 3	50	55	212.92	106.84	71.56

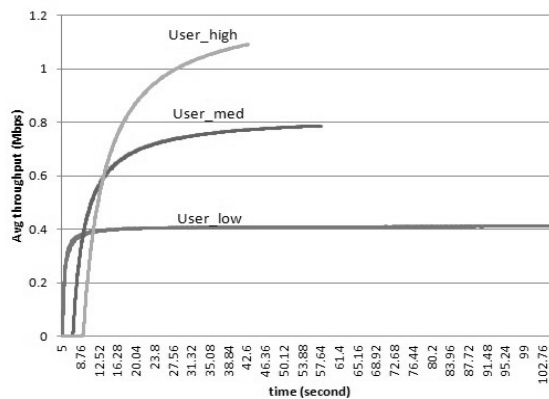


Figure 14: The average throughput for each user priority for TCP spoofing approach

The maximum throughput of spoofing approach can be obtained from equation (5), where in the access network using TCP with window size parameter and in backbone side using TCP highspeed. In the access side, for access delay of 10 ms, based on equation (5) we obtained that:

$$\text{Th_maxs access side user low} = \min(2 \text{ Mbps}, 10 \text{ Mbps})$$

$$\text{Th_maxs access side user med} = \min(4 \text{ Mbps}, 10 \text{ Mbps})$$

$$\text{Th_maxs access side user high} = \min(6 \text{ Mbps}, 10 \text{ Mbps})$$

Meanwhile, for the backbone delay of 10 ms and the server delay of 5 ms, where highspeed TCP protocol with window size 20 as the default is used, we obtained:

$$\text{Th_maxs backbone} = \min(5.33 \text{ Mbps}, 10 \text{ Mbps})$$

It can be seen from Table 2, that for the backbone delay of 10 ms and access delay of 10 ms, the transfer rate is determined by the settings window size of access side. This is because the backbone throughput > access throughput for each user priority. When the backbone and server delay of 55 ms, Th_maxs backbone is 1.45 Mbps. Under these conditions, the backbone becomes a bottleneck so that the completion time is determined by the speed of transfer over the backbone side. As soon as the access delay changed to 50 ms, which means Th_maxs access side of the low priority user = 0.4 Mbps, Th_maxs user side access med = 0.8 Mbps and the user side access Th_maxs high = 1.2 Mbps, the speed of the transfer process is more determined by the network access. Further completion time for all three spoofing scenarios can be seen in Table 2.

Table 2: Completion Time Each User Priority for The Spoofing Approach

No	Description	Delay		completion spoofing-qos		
		Access	Backbone +server	user 1	user 2	user 3
1	Scenario 1	10	15	20.92	10.54	10.08
2	Scenario 2	10	55	28.28	28.28	28.26
3	Scenario 3	50	55	101.08	50.78	34.04

Overall, based on Table 1 and Table 2, the completion time of each level of user priority using spoofing approach is faster than the completion time of end-to-end approach. The comparison of the completion time between the spoofing approach and the end-to-end approach, i.e. for high priority users, to the total delay can be seen in Figure 15.

The simulation of the partial QoS implementation in wireless networks is done using a topology as seen in Figure 9, with 54 Mbps wireless bandwidth share, access delay of about 10 ms, backbone bandwidth of 10 Mbps and backbone delay 30 ms. The average throughput of each user priority with end-to-end approach can be seen in Figure 16 below.

Using the same wireless network topology, partial QoS implementation through TCP spoofing approach for each class of user priority, can be seen in Figure 18. The interesting result for the user with the lowest priority is the average throughput flat when there is a user with a higher priority. But when users with higher priority reaches its optimum speed, the average user with low priority throughput also rise if the bandwidth is still available. Thus the overall average throughput of partial QoS through spoofing approach has a higher average throughput compared with end-to-end approach.

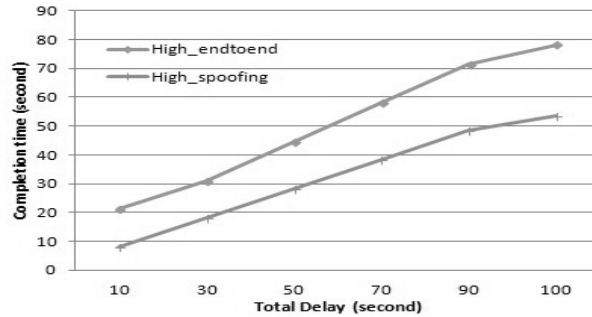


Figure 15: Comparison of Completion time for high priority user to total delay

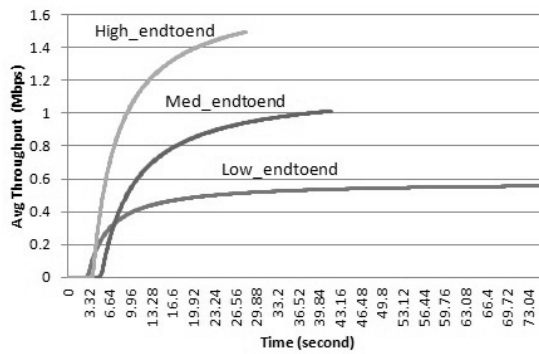


Figure 16: The average throughput of each priority user for end to end approach on wireless network

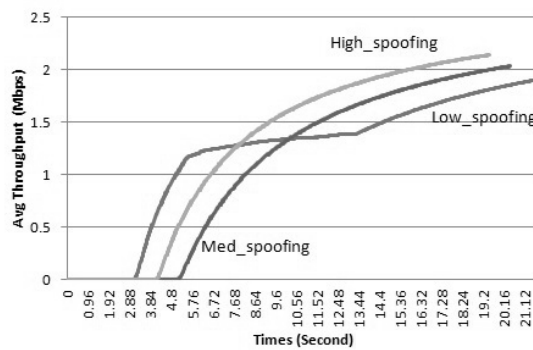


Figure 17: Average throughput each priority user for spoofing approach on wireless network

To get an idea of how much faster the FTP transfer 5 MB in wireless networks for all priority users by spoofing approach compared with end-to-end approach, we created wireless simulation network with two scenarios. Scenario 1 is the backbone delay of 10 ms and scenario 2 with the backbone delay of 30 ms delay. In Table 3 it appears that, for the both scenario, spoofing approach is faster than end-to-end approach to factor > 1 . Speed up user with lower priority is higher than speed up user with a higher priority. It occurs because the forwarding mechanism by highspeed TCP has higher ratio for lower throughput of the access side. Speed up can reach a significant number in certain cases. For example the speed up of low priority user for scenario 2 achieves 3.49 times.

Table 3: Speedup of The Spoofing and The End to End Approach on Wireless Network

No	User Priority	Completion Time				Speed Up	
		End to End		Spoofing		Scr 1	Scr 2
		Scr 1	Scr 2	Scr 1	Scr 2		
1	Low	36.26	76.16	20.56	21.8	1.76	3.49
2	Medium	19.88	39.82	14.6	18.46	1.36	2.16
3	High	11.26	24.34	9.34	15.5	1.21	1.57

6 Conclusion

In general the average throughput of TCP spoofing approach is higher than the average throughput of end-to-end approach both in wireline network and the wireless network. This is in accordance with equation (4) and (5) of section analysis. TCP spoofing approach allows the implementation of a subset of the QoS parameters applied partially in access networks. Thus TCP spoofing approach is more flexible than the end-to-end approach to enable partial QoS. TCP spoofing approach is also more flexible when compared to QoS implementations that require MAC layer changes such as 802.11e EDCA solution on WiFi network. Simulation results show that the TCP spoofing techniques allow implementation of partial QoS in the access network with better performance under to various delay conditions compared to the implementation of end-to-end QoS. The completion time speedup of the spoofing approach varies which value ≥ 1 compared with end-to-end approach. For certain cases, completion time speed up value is significant, more than 3 times. However, in the spoofing approach we need to consider the buffer availability of the spoofing gateway. This is especially the case where the maximum throughput of access network side much higher than the throughput of the backbone side. In this condition, the spoofing gateway needs to buffer the received data from the access side because the forwarding speed is less than the input rate from access network.

Bibliography

- [1] A. Tuoriniemi, G. Eriksson, N. Karlsson, and A. Mahkonen (2002); QoS concepts for ip-based wireless systems, *3G Mobile Communication Technologies, 2002. Third Int. Conf. on (Conf. Publ. No. 489)*, 229-233.
- [2] G. Rong and G. Lei (2010); A Partial Bandwidth Reservation Scheme for QoS Support in Ad Hoc Networks, *Computer Engineering and Applications (ICCEA), 2010 Second International Conference on*, 380-384.

-
- [3] L. Jae Young, L. Hwang Soo, and M. Joong Soo (2009); Model-based QoS parameter control for IEEE 802.11e EDCA, *Communications, IEEE Trans. on*, 57: 1914-1918.
 - [4] D. Miras (2002); Network QoS needs of advanced internet applications: A survey, *Internet2 QoS Working Group*.
 - [5] D. X. Wei and P. Cao (2006); NS-2 TCP-Linux: an NS-2 TCP implementation with congestion control algorithms from Linux, Proc. from the 2006 workshop on ns-2: the IP network simulator, Pisa, Italy, 2006.
 - [6] S. Xu and Y. Yang (2011); Protocols simulation and performance analysis in wireless network based on NS2, *Multimedia Technology (ICMT), Int. Conference on*, 638-641.
 - [7] S. Floyd (2003); HighSpeed TCP for Large Congestion Windows, IETF, Network Working Group RFC 3649, December 2003.
 - [8] J. W. Evans and C. Filsfils (2010); *Deploying IP and MPLS QoS for Multiservice Networks: Theory & Practice*: Morgan Kaufmann.
 - [9] K. L. Dias, D. F. Sadok, and J. Kelner (2003); A hierarchical resource management approach for QoS provisioning in IP-based wireless and mobile networks, *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, 3498-3502.
 - [10] S. H. Bouk, I. Sasase, S. H. Ahmed, and N. Javaid (2012); Gateway discovery algorithm based on multiple QoS path parameters between mobile node and gateway node, *Communications and Networks, Journal of*, 14: 434-442.
 - [11] V. B. Iversen (2012); Systems with selective overflow and change of bandwidth, *Communications in China (ICCC), 2012 1st IEEE Int. Conference on*, 694-697.
 - [12] R. Verschae, M. Koeppen, and K. Yoshida (2009); User Modeling for Improving QoS Using Partial User-Supplied Information," in *Intelligent Networking and Collaborative Systems, 2009. INCOS '09. International Conference on*, 387-390.
 - [13] P. Kulkarni, M. Sooriyabandara, and L. Li (2009); Improving TCP performance in wireless networks by classifying causes of packet losses, *Wireless Communications and Networking Conference, WCNC 2009. IEEE*, 1-6.
 - [14] Y. Li, R. Zhang, Z. Liu, and Q. Liu (2009); An improved TCP congestion control algorithm over mixed wired/wireless networks, *Broadband Network & Multimedia Technology, IC-BNMT '09. 2nd IEEE Int. Conf. on*, 786-790.
 - [15] D. Lopez-Pacheco and C. Pham (2004); Performance comparison of TCP, HSTCP and XCP in high-speed, highly variable-bandwidth environments, *Proc. of the IEEE 3rd Int. Conf. on Network Protocols (ICNP 2004). Berlin*, 2004.