Evaluation of Combination of ACK Filtering and ACK Congestion Control with Pure TCP/NewReno protocol in Asymmetry Network

Haydar Abdulameer Marhoon
Computer Science, Kerbala University

Evaluation of Combination of ACK Filtering and ACK Congestion Control with Pure TCP/NewReno protocol in Asymmetry Network

Haydar Abdulameer Marhoon
Computer Science, Kerbala University

Abstract:-

The user demand for high speed and ubiquitous connectivity has led to the development and deployment of many new technologies, Given that these networks are increasingly being deployed as high-speed access networks, it is highly desirable to achieve good network performance over such networks. These networks show different characteristics (asymmetry) in uplink and downlink directions. Network asymmetry (uneven bandwidth) can negatively affect the performance of feedback-based transport protocol such as Transmission Control Protocol (TCP). ACK Filtering and ACK Congestion Control techniques are used to diminish the congestion on the upstream link. These techniques suffer from sender burstiness and a slowdown in congestion window growth problems. This paper addresses the TCP performance problems caused by network asymmetry and evaluate the performance of combination ACK Filtering and ACK Congestion Control with Pure TCP/NewReno using the Network Simulation 2 (NS2), based on actual queue length, packet loss, packet arrival rate, congestion window size, and throughput. Even these tows techniques are reduced the number of the ACKs, the TCP/NewReno still have good performance as a result of this evaluation. The results showed that TCP NewReno has faster, smoother, and less queue length fluctuation, higher packet arrival rate, less packet drop, but with fluctuated throughput that does not reach the optimal level.

Keywords: ACK Filtering; ACK Cngetion Control; TCP/NewReno; Asymmetry Network

I. INTRODUCTION

The increase in the user application that require high speed network with ubiquitous connection network has derived inventors to come up with many new technologies. According to Mahbub and Jain [1], some of these technologies, including cable modems, direct broadcast satellite with interactive return channels, Very Small Aperture satellite Terminals (VSAT), Asymmetric Digital Subscriber Line (ADSL), Hybrid Fiber-Coaxial (HFC), several packet radio networks, digital subscriber line (DSL), and satellite-based network, were developed to moderate the last mile bottleneck. Other technologies, such as wireless and packet radio networks, are motivated by the need to providing the user with boundless access to their mobile devices across the Internet.

Asymmetric characteristics are exhibited by the mentioned above network technologies. Given that these technologies were deployed to provide high-speed access networks while achieving good network performance, the asymmetry of the networks often makes this challenging. The characteristics in uplink direction might be totally different from the characteristics in the downlink direction.

For example, many of existing connections (cables) were designed for broadcasting (unidirectional) from the head end out to user. The bandwidth of the upstream connection from the user out to the Internet is limited compared to the bandwidth of the downstream connection toward the user. Sometimes the upstream communication can be impossible on the same technology [2].

Usually, satellite sub networks employ shared frequency channels and arbitrate use of the satellite bandwidth by using MAC protocols which employ a Bandwidth on Demand (BoD) Scheme [3]. These links may show a per packet transmission overhead (each packet sent consumes satellite resource which could otherwise be used to transfer useful data) [4]. Consequently, several Very Small Aperture satellite Terminals (VSATs) employ techniques to mitigate the asymmetry effects in order to improve the performance. The subnetwork MAC contention problem is a major function of the number of ACKs transmitted rather than their size [5]

A. Asymmetry Classification

Network asymmetry has several types. The asymmetric networks have been classified depending on the characteristics of networks into three kinds [6]: bandwidth asymmetry, media access asymmetry, and link bit error rate asymmetry.

Bandwidth Asymmetry: The bandwidth of the downstream link is 10-1000 times the bandwidth of upstream link. ADSL, satellite, and cable modem networks are example of this configuration that depends on a dialup link for upstream connectivity. For example network has a 400Kbps downstream connection and a 56Kbps dialup upstream connection.

Media-Access Asymmetry: Media-access asymmetry shows itself in many forms, such as cellular wireless network, a centralized base station incurs a lower medium access control (MAC) overhead in transmitting to distributed mobile end nodes. In this type of Asymmetry, the transmitter may experience larger (smaller) MAC delay than the receiver.

Loss Rate Asymmetry: Packet loss probability in the uplink may be different than that of downlink. This can happen if one of the links is more congested than the other. Loss-rate asymmetry can occur in any network, and it may be a transient phenomenon.

This imposes the use of different network technology for upstream connection; dialup modem line is example of this. An example of such a setup is shown in Figure 1. The figure depicts the bandwidth asymmetry in wireless cable modem network where users receive data from Internet through a connection of high bandwidth, say 100Mbps, and send out data through a low bandwidth downstream connection.

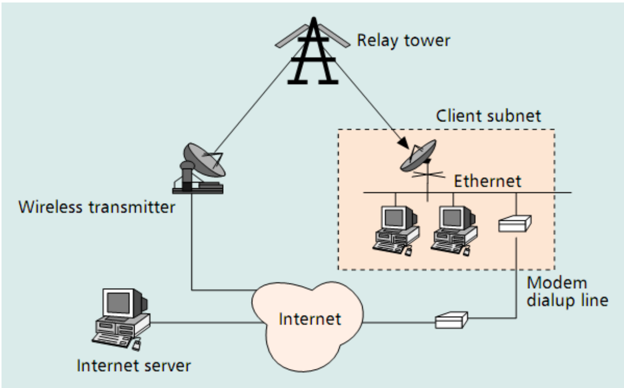

Figure 1: Wireless cable modem network (adopted from [1])

Network asymmetry can affect the performance of transport protocols, which are feedback-based, such as Transmission Control Protocol (TCP). This is because that even the connection of the data stream is uncongested, the congestion occurs on the other direction can easily disrupt the feedback flow, which can degrade the performance of transport protocols.

II. SOLUTIONS TO IMPROVE TCP PERFORMANCE

In order to improve TCP performance over asymmetric networks, it is important to manage the bandwidth usage on the uplink, used by ACKs (and possibly other traffic). It can be done by reducing the number of ACKs that are transferred over the upstream channel, which has the potential of destroying the (desirable) self-clocking property of the TCP sender where new data transmissions are started by incoming ACKs. Thus, the second issue is to avoid any adverse impact of infrequent ACKs. In addition, desired TCP performance can be achieved by avoiding any negative impact of infrequent ACKs.

According to Balakrishnan et al. [7], the uplink bandwidth can be managed by controlling the compression level, frequency, and scheduling of ACKs flow through the upstream link. As stated in [8], TCP header compression is used over low-bandwidth links running Point-to-Point Protocol (PPP) due to that fact that it can reduces the size of ACKs on the uplink in case losses are infrequent in a condition where the compressor and decompression are synchronized. TCP header compression was proposed to reduce the size of ACKs in order to increase the rate of the reverse channel in terms of ACKs per unit of time.

Nevertheless, TCP header compression cannot solve all of the problems because in some networks there is a large per-packet MAC overhead which can be considered as independent of the packet size. In addition, a reduction in the ACKs size cannot prevent adverse interaction with large upstream data packets where bidirectional traffic takes place. Therefore, to efficiently address the TCP performance problems cause by network asymmetry, others techniques are needed

Several solutions were proposed to match the rate of ACKs to the rate of the reverse channel. This can be done by either delaying ACKs at the receiver side, or Filtering them in the reverse buffer (called ACK Filtering (AF)) [9]. The technique for delaying ACKs at the receiver adaptively is called ACK Congestion Control (ACC). For both solutions, it is required to implement the technique at the receiver together with some feedback from the network. The purpose is to adapt the generation rate of ACKs as a function of the reverse buffer occupancy.

Another technique is called ACKs-first Scheduling [10] which is designed to improve performance of bidirectional transfers where case data packets and ACKs compete for resources at the upstreambottleneck link. If a single First-In First-Out, FIFO, queue for both data packets and ACKs, performanceof forward transfers can be affected significantly.

A. ACK Filtering

To effectively address the performance problems caused by asymmetry, there is a need for techniques over and beyond TCP header compression. The second type of techniques performed only at one point on the return path, within the upstream router/host. It uses class or per-flow queues at the upstream interface of the router/host to manage the queue of packets waiting for transmission on the bottleneck return link.

In this type of technique, the queue size is limited, and an algorithm employed to remove (discard) excess ACK packets from each queue. This relies on the cumulative nature of TCP ACKs.

ACK Filtering (AF) [11] is a TCP-aware link-layer technique that reduces the number of TCP ACKs sent on the reverse link. The challenge is to ensure that the sender does not stall waiting for ACKs, which can happen if ACKs are removed indiscriminately on the reverse path. AF removes only certain ACKs without starving the sender by taking advantage of the fact that TCP ACKs are cumulative. As far as the sender's error control mechanism is concerned, the information contained in an ACK with a later sequence number subsumes the information contained in any earlier ACK. When an ACK from the receiver is about to be enqueued at a reverse direction router, the router or the end-host's link layer (if the host is directly connected to the bottleneck upstream link) checks its queues for any older ACKs belonging to the same connection. If any are found, it removes them from the queue, thereby reducing the number of ACKs that go back to the sender. The removal of these "redundant" ACKs frees up buffer space for other data and ACK packets.

B. ACK Congestion Control (ACC)

ACK congestion control (ACC) is an end-to-end mechanism that enables TCP receiver to detect the ACK packets loss and it send the ACKs at a slower rate. The queue management mechanism used at the uplink bottleneck router, such as RED

(Random Early Detect), is responsible for detecting the ACK packet loss and notifying the TCP receiver about the loss, and as a reaction, the receiver slows the ACKs sending rate [12]

Therefore, ACC has two parts: (i) network feedback to inform the receiver that the ACK path is congested, and (ii) the receiver's response to such information. The router detects congestion by monitoring the average queue length. If the average exceeds a specific threshold, the router selects a packet at random and marks it, by setting an Explicit Congestion Notification (ECN) bit in the packet header; otherwise the packet is normally dropped. The ECNnotification received by the end host is reflected back to the sending TCP end host forcing it to start congestion avoidance. Note that routers implementing RED with ECN, do not eliminate packet loss, and may drop a packet (even when the ECT bit is set). It is also possible to use an algorithm other than RED to decide when to set the ECNbit.

The TCP receiver maintains a dynamically varying delayed-ACK factor, d, and sends one ACK for every d data packets received [13]. When it receives a packet with the ECN bit set, it increases d multiplicatively, thereby decreasing the frequency of ACKs also multiplicatively. Then for each subsequent round-trip time (determined using the TCP timestamp option) during which it does not receive an ECN, it linearly decreases the factor d, thereby increasing the frequency of ACKs. Thus, the receiver mimics the standard congestion control behavior of TCP senders in the manner in which it sends ACKs.

ACC remains a research issue as it should not be used in its current form because not all TCP senders usually use ECN in their packets

III. EVALUATION & RESULTS

In this section, we provide the evaluation of the observed performance of TCP where the ACK Filtering and ACK

Congestion techniques are involved comparing to the case where only TCP NewReno is used.

As this performance evaluation model considers the unidirectional data stream where 20 TCP sources uploading data to a single server over a link of 10Mbps and receiving ACK over a link of 300Kbps, bandwidth asymmetry indicates that buffering of the ACKs will take place. Thus, the peak rate of the transmitting TCP will be determined by the time intervals between transmissions of the ACKs over the downlink from the server. As show in figure

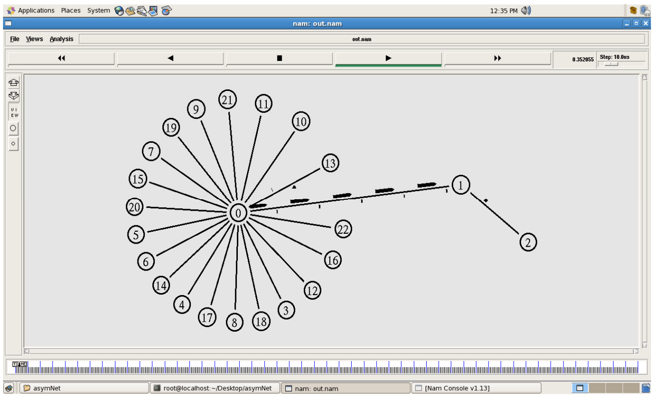

Figure 2: Topology from nam

A. Actual Queue Length

This section provides the performance of ACK Filtering and ACK congestion techniques in terms of the actual queue length. The purpose is to show the occupancy level of buffer where the bandwidth of the network is asymmetric which affects the data traffic transfer over the network. Also, the actual queue length can help realizing how much the buffer is utilized.

The queue management mechanism used at the routers imposes its policies, such as packet drop or mark, to keep the queue length as small as possible to accommodate the sudden increases in the data traffic.

For the case of using the ACK Filtering and ACK congestion techniques, Figure 3 shows the actual queue length of the uplink buffer towards the destination (server) while Figure 4 shows the actual queue length of the uplink buffer using TCP NewReno, respectively. Note that the traffic shown in the figure is the FTP sent by TCP resources which supposed to be accommodated in the router's buffer.

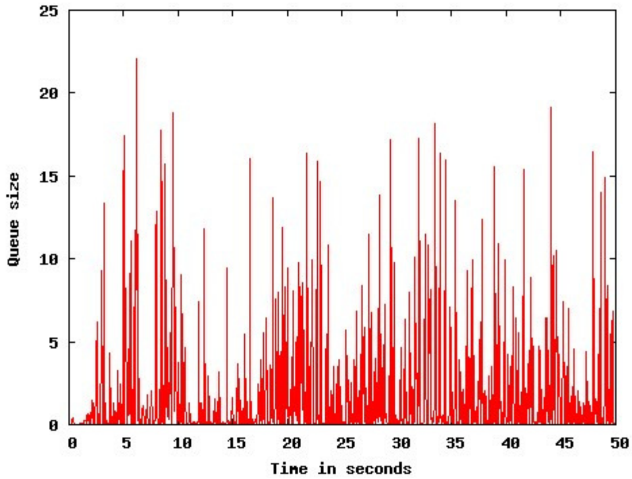

Figure 3: The actual queue length of the uplink buffer using ACK Filtering and ACC techniques

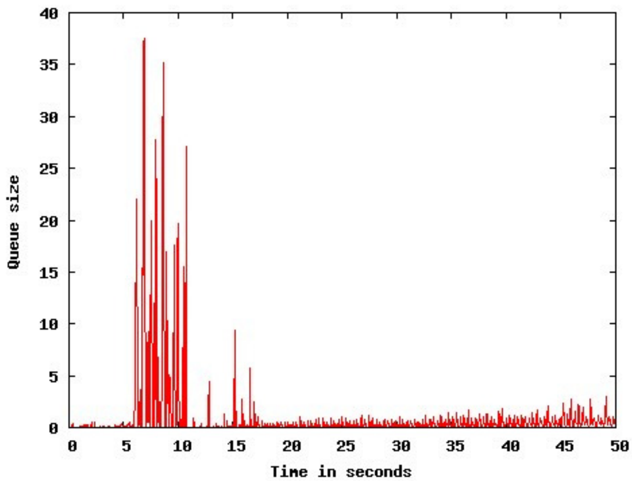

Figure 4: The actual queue length of the uplink buffer using TCP NewReno

From the Figure 3, we can realize how the number of received ACK through the uplink affects the transmission rate of TCP sources. As ACK Filtering technique is used on the downlink which reduces the number of the TCP ACKS sent on the downlink (by removing only the older ACKs belonging to a specific TCP connection without starving the TCP source considering the fact that TCP ACKs are cumulative), TCP sources has to reduce their transmission rates from time to time which justify the fluctuations in the number of packets (Figure 3) in the uplink buffer. The fluctuation in the queue length of the uplink buffer is much more compared to the case when only TCP NewReno is used in network with asymmetric bandwidth. As shown in Figure 4, the queue length fluctuates widely from time 6 to 10 only while most of the time it show faster and smoother queue length.

B. Packet Loss

Packet loss results in very noticeable performance issues, packet drops degrades the performance of the TCP applications significantly when the packet loss is high. For the case of using ACK Filtering and ACK congestion techniques, Figure 5 shows the total number of packets dropped at the uplink buffer which is around 270 packets compared to the case where only TCP NewReno is used (see Figure6) .

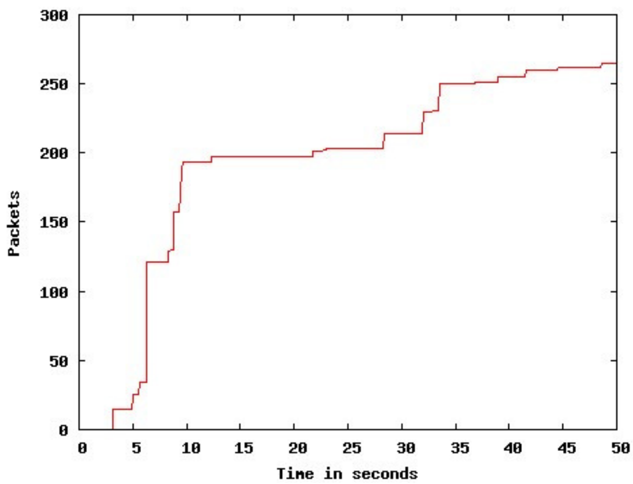

Figure 5: Packets dropped at the uplink buffer using ACK Filtering and ACC techniques

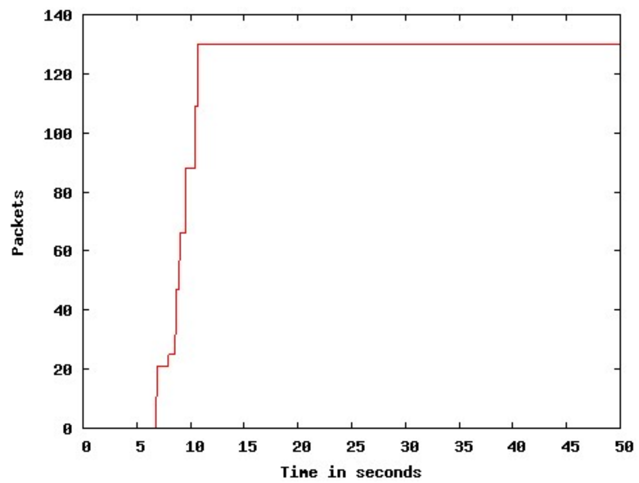

Figure6: Packets dropped at the uplink buffer using TCP NewReno

Figure6 shows the total number of packets dropped at uplink buffer which is about 130 packets. This implies that TCP NewReno behaves better in terms of packet lost, in the absence of the ACK Filtering and ACK congestion techniques in the network. The number of lost packets is small compared to the case with TCP Asym(involving ACK Filtering and ACK congestion techniques).

Figure 7 shows the packet loss on the downlink. It is around 14000 ACKs packets got dropped by RED queue management that employ ACK Filtering technique which enforces dropping the old packets when a new ACK arrives for the same TCP source, as mentioned earlier.

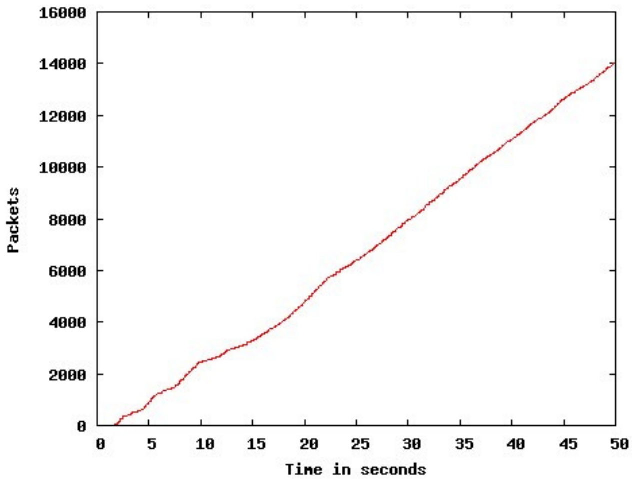

Figure 7: Packets dropped at the downlink buffer using ACK Filtering and ACC techniques

Figure 8 shows the packet loss on the downlink in the case of using TCP NewReno. It is around 22000 ACKs packets got dropped at the router shows how fast the packets are arrived to the server that triggers sending the ACKs frequently.

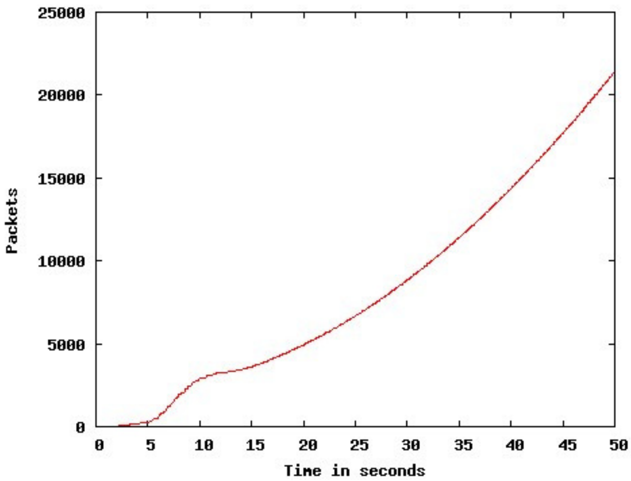

Figure 8: Packets dropped at the downlink buffer using the TCP NewReno

C. Congestion Window Size

Figures 9 is shows evaluation of the window sizes of the first five TCP sources (based on their starting time) where ACK Filtering and ACK congestion techniques are used. From the figure, it is clear that TCP window fluctuates more and has long time outs occasionally which it degrades the performance of TCP. It is important to mention that using the strategy of random drop for the queued packets (as in RED mechanism) is much better than drooping packets from tail (as in Drop Tail mechanism). Drops from tail leads to long and consecutive sequences of ACKs packet drops on the downlink which results in sudden increase in TCP sources traffic.

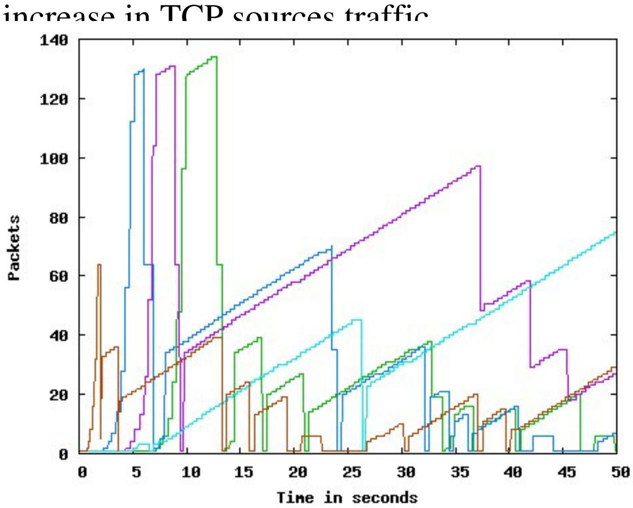

Figure 9: The window size for the first five TCP Asym sources using ACK Filtering and ACC techniques

Figures 10 shows evaluation of the window sizes of the first five TCP sources (based on their starting time) in the case of using the TCP NewReno. It is obvious that TCP window fluctuation much more less and some TCP sources do not experience time outs at all. From the figure, TCP NewReno source increase their window size gradually and behave gently when the packet loss in occurs . This implies that TCP NewReno shows better performance than the case where ACK Filtering and ACK congestion techniques are used.

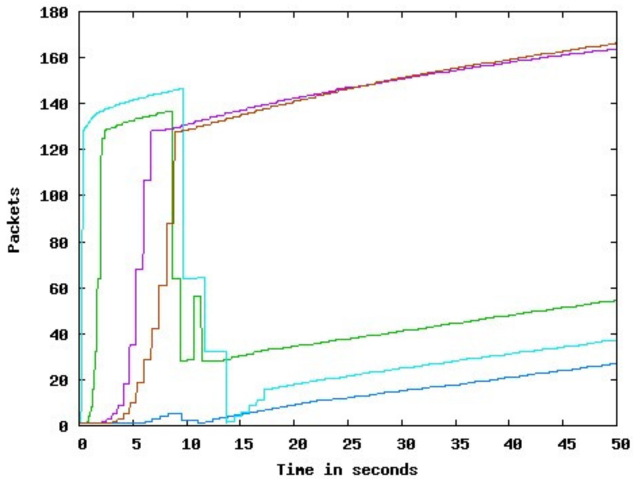

Figure 10: The window size for the first five TCP NewReno sources

D. Throughput

In data network, throughput is defined as the amount of data transferred successfully from host to another in a given time period. Throughput, which is essentially bound by the Bandwidth Delay Product (BDP), is measured in number of bits per second (bps). In this project, throughput is measured as the number of data packets received correctly at the server host in a unit of time (in bit per second). Throughput is the significant performance measure for short and long-lived TCP connections.

Figure 11 illustrates the throughputs gained for the first five TCP connections (based on their transmission times) for the case of using ACK Filtering and ACK congestion techniques. It shows that at the beginning, the throughput was high till TCP resources got chocked because of the high packet loss happened on the uplink. The main reason is that TCP end hosts, in this case, use the ACK congestion control technique which imposes notification to the TCP sender that the downlink that ACKs use is congested, and as a result, TCP sources are forced to respond to this notification by reducing their transmission rate.

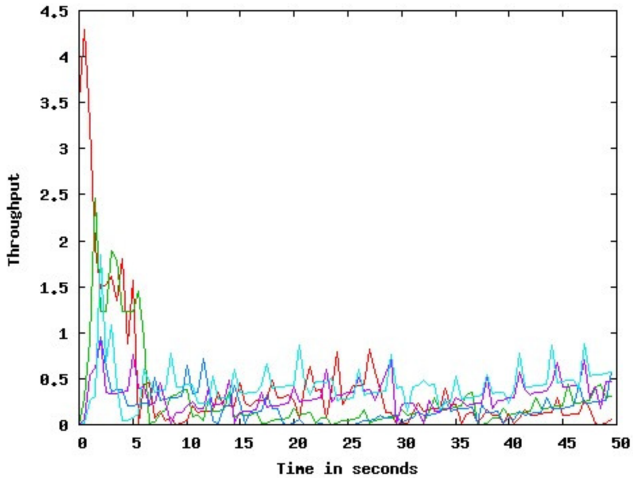

Figure 11: Throughput gained for the first five TCP Asymmetry sources using ACK Filtering and ACC techniques

The average throughput for the first TCP resources (based on transmission starting times), in the case of using ACK Filtering and ACK congestion techniques, are shown below.

Avg throughput (source 18) = 0.4134784 MBytes/sec

Avg throughput (source 21) = 0.2867584 MBytes/sec

Avg throughput (source 7) = 0.1887424 MBytes/sec

Avg throughput (source 19) = 0.3097984 MBytes/sec

Avg throughput (source 20) = 0.4215424 MBytes/sec

Figure 12 shows the throughputs gained for the first five TCP connections (based on their transmission times) for the case of using TCP NewReno. It is clear that the throughput of TCP NewReno sources is higher, fluctuate almost constantly, and has less transmission rate reduction or idle time compared to the case of ACK Filtering and ACK congestion techniques.

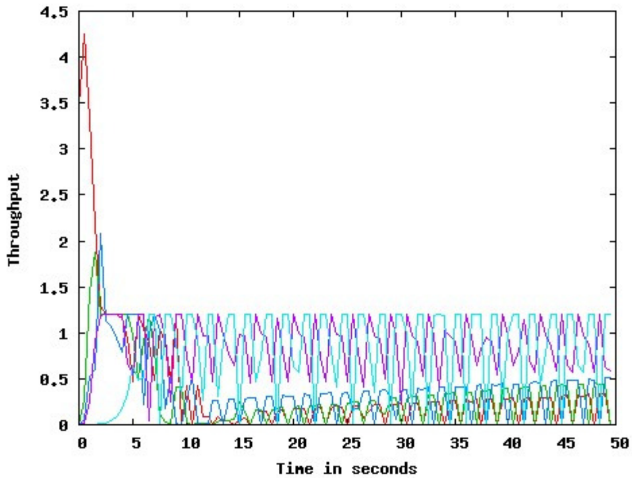

Figure 12: Throughput gained for the first five TCP NewReno sources

The average throughput for the first TCP resources (based on transmission starting times), in the case of using TCP NewReno, are shown below.

Avg throughput 6.0 - 2.3 = 0.4090416MBytes/sec

Avg throughput 22.0 - 2.19 = 0.3201168MBytes/sec

Avg throughput 3.0 - 2.0 = 0.3668448MBytes/sec

Avg throughput 13.0 - 2.10 = 0.8564976MBytes/sec

Avg throughput 5.0 - 2.2 = 0.7833376MBytes/sec

IV. CONCLUSION

The development of the Internet accessing technologies raises some performance issues, especially in TCP/IP network. The difference in the characteristics of the uplink and downlink links degrades the performance of TCP.

Network asymmetry can affect the performance of transport protocols, which are feedback-based, such as Transmission Control Protocol (TCP). This is because that even the connection of the data stream is uncongested, the congestion occurs on the other direction can easily disrupt the feedback flow, which can degrade the performance of transport protocols.

Improving TCP performance over asymmetric networks was a serious matter that drove researchers to find out solutions to overcome this issue. They found that, it is important to manage the bandwidth usage on the uplink, used by ACKs (and possibly other traffic). This can be made by decreasing the number of ACKs that are transferred over the uplink channel.

Solutions for improving the TCP performance over asymmetric networks include Managing the Link Bandwidth and Handling Infrequent TCP ACKs drove researchers to develop several techniques for these solutions

Therefore, this paper concerns developing a simulation model to study the effectiveness of ACK Filtering and ACK congestion control techniques in lessening the impacts of bandwidth asymmetry on TCP performance. The performance was examined in terms of actual queue length, packet loss, packet arrival rate, link utilization, and throughput. The developed model was implemented using network simulator 2 (ns-2).

Based on the simulation results, it was shown that the combination of ACK congestion control and ACK Filtering techniques underutilizes the network bandwidth due to frequent decrease in the window sizes of the TCP sources compared to the TCP NewReno case where trying to utilize the bandwidth carefully; despite of the fact that the performance of TCP NewReno was not optimum.

The results showed that the combination of the ACK congestion control and ACK Filtering techniques have negative impact on the performance of TCP when DropTail queue management mechanism is used on the uplink. In contrast, TCP NewReno showed better performance based on its particular behavior in utilizing the network resources to achieve good throughput. The results showed that TCP NewReno has faster, smoother, and less queue length fluctuation, higher packet arrival rate, less packet drop, but with fluctuated throughput that does not reach the optimal level.

الخلاصة:ـ

الطلبات المتزايدة على السرعة العالية من قبل مستخدمين الشبكات تقود المطورين للتفكير بتقنيات جديدة باستمرار، وهذه التطورات قادة الشبكات إلى التطور بصورة سريعة وعالية الأداء لتقديم ما هو أفضل للمستخدمين. هذه الشبكات تبين اختلاف في الخصائص بين عملية التحميل والتصعيد للبيانات وهذا ما يدعى بالشبكة الغير متماثلة الخصائص. الشبكات الغير متماثلة الخصائص من الممكن ان تؤثر بشكل سلبي في أداء بعض البروتوكولات التي تحمل خاصية الاستفادة من الاتجاهات العكسية مثل (TCP). ACK filtering and ACK Congestion Control هي من التقنيات التي تستخدم في تقليل الازدحامات في التي تحصل في جهة تصعيد البيانات وهذه التقنيات تجعل الشبكة تعاني من عملية نمو حزم البيانات من الجهة المرسلة لها. هذا البحث يحدد المشكلة التي من الممكن ان تحصل في شبكات الغير متماثلة الخصائص وتصميم نموذج محاكات للمقارنة بين استخدام هلتين التقنيتين معا في مقابل استخدام TCP\NewReno صافية بدون اي تقنية في جهة التصعيد للبيانات باستخدام محاكي الشبكان من الجيل الثاني NS2، بالاعتماد على عدة خصائص مهمة مثل طول الطابور الحقيقي، كمية حزمة البيانات، عدد البيانات المفقودة، نسبة البيانات الواصلة والانتاجية، وعلى الرغم من ان هاتين التقنيتين تقومان بتقليل عدد البيانات الواصلة من جهة التصعيد الا انها في نفس الوقت لا تقوم باستغلال الشبكة من حيث ان حزمة البيانات في جهة المصدر تعاني من عدم النمو وذلك لأن المصدر يستخدم عدد التعليمات الواصلة لزيادة حجة حزمة البيانات المرسلة.

مما تقدم يستطيع البحث القول بان استخدام TCP\NewReno وحده يعود على الشبكة بالنفع اكثر من استخدام هاتين التقنيتين وعلى الرغم من هذا فإنها لم تصل الى المستوى المثالي بعد.

V. REFERENCES

[1] H Mahbub and j Raj, High Performance TCP/IP Networking: Concepts, Issues, and Solutions., 2003.

[2] R Mattson and S Ghosh, "A performance analysis of page and object retrieval using HTTP-MPLEX in symmetric network environments," in 6th IEEE/ACIS International, 2007.

[3] M. Haoling and C. Lin, "Performance analysis of randomized MAC for satellite telemetry systems.," in 5th International ICST Conference on, China (CHINACOM), 2010.

[4] A. Narula-Tam et al., "l. QoS considerations for future packet switched satellite communication systems with dynamic resource allocation," in IEEE, Milita, 17-20 Oct. 2005.

[5] E. Aryafar, O. Gurewitz, and E. W Knightly, "Distance-1 Constrained Channel Assignment in Single Radio Wireless Mesh Networks.," in the INFOCOM, 27th Conference on Computer Communications. IEEE., 2008.

[6] Y. Liu, M. Ye, and H. Zhang, "The improvement of TCP performance in bandwidth asymmetric network.," in the Personal, Indoor and Mobile Radio Communications, PIMRC 2003. 14th IEEE Proceedings on., 2003.

[7] H. Balakrishnan and V. N. Padmanabhan, "How network asymmetry affects TCP.," in Communications Magazine, IEEE, 2001, pp. 60-67.

[8] W. Gengyu, R. Sato, A. Fujii, and Y Nemoto, "Improvement of TCP performance on asymmetric channels based on enhancements of compressing TCP/IP header algorithm. ," in the Communication Technology Proceedings,ICCT '98. International Conference on., 1998.

[9] B. Chadi and A Eitan, "On ACK Filtering on a Slow Reverse Channel.," in the Proceedings of the First COST 263 International Workshop on Quality of Future Internet Services, 2000.

[10] P. Jiyong, P. Daedong, H. Seongsoo, and P. Jungkeun, "Preventing TCP performance interference on asymmetric links using ACKs-first variable-size queuing.," in Comput. Commun., 2011 , pp. 730-742.

[11] G. Fairhurst et al., "Performance issues in asymmetric TCP service provision using broadband satellite," in Communications, IEE Proceedings, 2001, pp. 95-99.

[12] F. Keceli, I. Inan, and E. Ayanoglu, "TCP ACK Congestion Control and Filtering for Fairness Provision in the Uplink of IEEE 802.11 Infrastructure Basic Service Set.," in the Communications ICC '07. IEEE International Conference on., 2007.

[13] C. Jiwei, L. Yeng Zhong, M. Gerla, and M. Y. Sanadidi, "TCP with Delayed Ack for Wireless Networks. ," in the Broadband Communications, Networks and Systems, BROADNETS 3rd International Conference on., 2006.