ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 2006. Nr. 7(71)
ELEKTRONIKA IR ELEKTROTECHNIKA

TELECOMMUNICATIONS ENGINEERING
T 180
TELEKOMUNIKACIJŲ INŽINERIJA

A New Approach to Improve TCP Performance over Asymmetric Networks

Wissam Al-Khatib, K. Gunavathi
Department of Electronics and Communication Engineering, PSG College of Technology, Coimbatore-641004, India, phone: -0091-9894187009, e-mail:wissampsg1@yahoo.com

Introduction

The ever growing demand for broadband data communication has lead to the development of many new network access technologies. A few of these, such as cable modem, Asymmetric Digital Subscriber Line (DSL), satellite based networks are aimed at alleviating the “last mile” bottleneck [1], while others, such as wireless packet radio networks, are motivated by the need to provide users with continuous access to the internet, particularly to their mobile devices.

One of the most evident characteristics of the networks mentioned above is asymmetry [2]. An asymmetric network is one in which the network characteristics in one direction is quite different than those in the other direction. For example in cable modem and ADSL networks, the downstream bandwidth is very much more than the upstream bandwidth. The upstream bandwidth of the cable plant, from the customer premises out to the ISP (Internet Service Provider), is often limited compared to the downstream bandwidth towards the customer premises. In certain cases upstream communication may just become impossible [3].

Network asymmetry adversely impacts the performance of TCP which heavily relies on feed back mechanism to assure consistent and smooth transmission as TCP is an ACK-Clocked protocol [4]. Though the bandwidth in forward direction is adequate and data can reach the receiver quickly, the acknowledgements that flow in the reverse direction suffers delay due to congestion and can’t arrive at the sender in time, leading to a great degradation in the whole performance. If acknowledgements get bunched up, the sender may burst out data which would over flow some queues. Further more TCP sender depends on the received acknowledgements to increase the size of the congestion window and hence the lack of acknowledgements will induce slow growth in congestion window, which degrades the TCP performance under the utilization of network available bandwidth [5].

In this paper the performance problems caused by network asymmetry in the context of TCP are discussed. A new technique is developed to address these problems. The objective of this work is to maximize the link utilization and user satisfaction. The proposed scheme named Dynamic Class Based Queuing (DCBQ) adapts the available bandwidth between the opposite direction traffics, and manages to obtain the best utilization of a shared asymmetric link. The simulation results show that DCBQ is a robust scheduling mechanism when confronted to changes in network settings.

Asymmetric Networks Classification and Related Work

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. In Bandwidth asymmetry, the forward link bandwidth is 10-1000 times the reverse link bandwidth. Examples include cable modem, asymmetric DSL (ADSL) and satellite based networks. Media Access asymmetry is difference of access characteristics in different directions. In case of Link bit error rate asymmetry the quality of link is good in one direction but in reverse direction it is very bad.

The asymmetric networks may also be classified depending on the way of data transfer. They are unidirectional and bidirectional data transfer. In the case of unidirectional, the data transfer occurs only in the forward direction. The acknowledgement path is assumed to be the bottleneck of transmission. A parameter called bandwidth normalized ratio k for connection [7], which is ratio of raw bandwidth divided by the ratio of the packet sizes used in two directions as given in equation (1)

$$k = \frac{C_{dn}}{C_{up}} * \frac{L_{ack}}{L_{data}}, \qquad (1)$$

where C_{dn} and C_{up} are Bandwidth of forward and reverse Channel respectively, L_{ack} and L_{data} are the length of ACK Packet and Data Packet respectively. For example, in 10Mb downlink and 100Kb uplink, the raw bandwidth ratio is 100, for 1000 bytes data packets and 40 bytes ACKs packets, the ratio of packet size is 25. For such parameters, k is 100/25 = 4. The value of k is the most crucial parameter for asymmetric network. If the receiver transmits more than one acknowledgement for every *k* data packets, the upstream bottleneck gets saturated before the downstream does. This will force the sender to clock out data more slowly than optimal, thus decreasing the throughput [4]. On average only one acknowledgement will get through for every *k* data packets transmitted by sender which could degrade the performance.

In case of bidirectional data transfer, the data flow occurs in both forward and reverse direction simultaneously. An example of this is user sending an e-mail (Uploading) and simultaneously viewing web pages (Downloading). This becomes more complicated. On the bottleneck reverse link the data packets of reverse direction will monopolize bandwidth which will result in acknowledgement compression and hence stops it from reaching the sender which would result in throughput degradation in the forward link [2]. In other cases the ACKs are not responsive to congestion, so the available bandwidth and the rate of uploaded data will drop to zero due to ACK monopolizing of reverse channel [3]. In summary, the presence of bidirectional data transfer would exacerbate the problem due to bandwidth asymmetry. The reason is adverse interaction between the data packets of the upstream connection and acknowledgements of the downstream connection.

Link bandwidth and buffer space are two main problems that occurs because of contention in the bottleneck resources of reverse direction. It is clear that there are two key issues that needed to be addressed in order to improve TCP performance over asymmetric networks [2]. The first issue is managing bandwidth usage on the uplink, used by ACKs. So many techniques, work by reducing the number of ACKs that flow over the upstream channel. Thus, the second issue is to avoid any adverse impact of infrequent ACKs.

Uplink bandwidth management may be performed by controlling the degree of compression, frequency, and scheduling of upstream ACKs. The available techniques are TCP Header Compression, ACK Filtering, ACK Congestion Control and ACKs-First Scheduling etc., [RFC 3449]. To overcome the problem of bidirectional traffic over asymmetric networks, the commonly used algorithm is ACKs- First Scheduling, which gives higher priority to ACKs over data packets. The motivation is that it minimizes the ideal time for the forward connection by minimizing the time that ACKs spend queued behind data packets in upstream link. The drawback of this algorithm is that ACKs are not reactive to congestion. So they will monopolize the whole reverse bandwidth and the rate of uploaded data will fall nearly to zero. This may not correspond to the optimal allocation of the scarce resources on the reverse path. The other technique for bidirectional TCP traffic called priority based multiplexing [RFC 3135], uses a queuing strategy combined with a scheduling mechanism at the upstream link. A simple scheme may be implemented using per flow queuing with a fair scheduler (e.g., Round robin service to all flows or priority schemes). The problem here is the static fixation of bandwidth allocation at the beginning, which cannot be changed in accordance to the changes of traffic rates in both directions. Moreover if the priority for the ACKs is higher than the data rate of uploading would suffer and vice versa.

In this context a new dynamic class based queuing (DCBQ) which use a class based queuing is proposed. This allocates the available bandwidth between the opposite direction traffics dynamically and that manages to obtain the best utilization of the shared asymmetric link.

Proposed Dynamic Class Based Queuing

As two-way asymmetric connections will probably become common case in the future with the widespread use of ADSL, satellites and other high-speed technologies. It is important to make sure that congestion will be properly handled in these environments. To this end, Fatma Louati, et al proposed a new Queuing mechanism called Adaptive Class-based Queuing (ACQ) for handling two-way TCP traffic over links that exhibit bandwidth asymmetry [3]. In ACQ algorithm Fair Queuing (FQ) scheduling algorithm between the different classes has been used. A modified scheme of ACQ, named Dynamic Class Based Queuing (DCBQ) has been proposed in this paper, which uses Weighted Round Robin (WRR) scheduling algorithm. WRR scheduling algorithm is simpler to implement and has lesser complexity compared to FQ scheduling algorithm used in ACQ [9]. DCBQ algorithm runs at the entry of the slow link and relies on two separate classes, one for ACK packets and one for Data packets. DCBQ proposes to adapt the weights of both classes according to the crossing traffic in order to maximize some utility function defined by the user or the network operator. The proposed technique DCBQ shows good performance compared to existing algorithms. In the following section the mathematical model of the algorithm is presented.

DCBQ Algorithm

Fig. 1 shows the architecture of asymmetric network. The downstream link will be a high speed link and the reverse link will be a low speed link. For bidirectional traffic the router used at the reverse direction is Class Based Queuing (CBQ). The two flows of ACKs and Data in reverse direction have to be separated by some kind of two classes CBQ. A simple algorithm to adapt the rates of the ACKs and Data class queues at the input of the reverse link has been proposed. In DCBQ the ACKs and data packets are queued in separate buffers and served in Weighted Round Robin scheme.

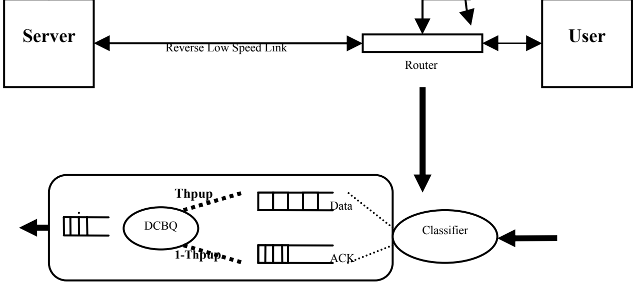

Fig. 1. Architecture of asymmetric network

To explain the model, let Thp_{dn} be the downstream data rate, and Thp_{up} the upstream data rate. The user satisfaction of the downstream data is U_{dn} (Thp_{dn}) and the user satisfaction of the upstream data is U_{up} (Thp_{up}). The need is then to find Thp_{dn} and Thp_{up} that maximize the global utility function, which is given in equation (2)

$$U\,(Thp_{dn}, Thp_{up}) = U_{dn}\,(Thp_{dn}) + U_{up}\,(Thp_{up})\,. \tag{2}$$

It is also considered that the DCBQ buffer is able to measure the rate of data in both directions. This can be done by getting the data rate in the forward direction from that of ACKs in the reverse direction. At every time interval T, the proposed algorithm measures the data rate in both directions and dynamically changes upstream data flow rate. The ACK flow will be allocated the rest of the bandwidth.

Let R(n) be the rate allocated to the data flow in the upstream direction at time nT, n Є N. This rate remains allocated until time (n + 1) T. Let $Thp_{up}(n)$ and $Thp_{dn}(n)$ be the measured upstream data rate and downstream data rate respectively between nT and (n + 1)T as shown in equations (3) and (4)

$$Thp_{dn} = \frac{No.\ of\ \ ACK_{received\ at\ ACKqueue}\ *\ Size\ of\ \ Data\ Packet}{T}\ , \tag{3}$$

$$Thp_{up} = \frac{No.\ of\ \ Data\ Packet_{received\ at\ Dataqueue}\ *\ Size\ of\ \ Data\ Packet}{T}\,. \tag{4}$$

Finally, let "U (Thp_{dn}, Thp_{up})" be the total user satisfaction. By using the gradient projection method the value of R(n), is updated as given in equation (5)

$$R(n+1) = Thp_{up}(n) + \gamma\ *\ \frac{dU(Thp_{up}(n), Thp_{dn}(n))}{dThp_{up}(n)}\,. \tag{5}$$

This method assumes the existence of one maximum in the definition region of Thp_{up} and Thp_{dn}, which assumed true here. γ is a constant that tradeoffs stability and convergence rate. The optimal value of γ will be changed with respect to the experiment parameters (no of nodes, data rate etc.). From (5) the data rate is given by,

$$R(n+1) = Thp_{up}(n) + \gamma\frac{dU(Thp_{up}(n))}{dThp_{up}} + \gamma\frac{dU_{dn}(Thp_{dn}(n))}{dThp_{dn}} * \frac{dThp_{dn}(n)}{dThp_{up}}\,. \tag{6}$$

Since the clear relation between Thp_{dn} and Thp_{up} is unknown, the following approximation is made

$$\frac{dThp_{dn}(n)}{dThp_{up}} = \frac{Thp_{dn}(n) - Thp_{dn}(n-1)}{Thp_{up}(n) - Thp_{up}(n-1)}\,. \tag{7}$$

Using the equation (7) in (6) the rate allocated to the upstream data flow will become

$$R(n+1) = Thp_{up}(n) + \gamma\frac{dU(Thp_{up}(n))}{dThp_{up}} + \gamma\frac{dU_{dn}(Thp_{dn}(n))}{dThp_{dn}} * \frac{Thp_{dn}(n) - Thp_{dn}(n-1)}{Thp_{up}(n) - Thp_{up}(n-1)}\,. \tag{8}$$

Here the utility function is equal to the bandwidth utilization. The user wants to maximize the total utilization in both directions. Let C_{dn} be the available bandwidth in the downstream direction, and C_{up} the available bandwidth in the upstream direction. Hence

$$U_{up}(Thp_{up}) = \frac{Thp_{up}}{C_{up}}, \qquad U_{dn}(Thp_{dn}) = \frac{Thp_{dn}}{C_{dn}}\,.$$

The allocated rate of upstream data flow becomes

$$R(n+1) = Thp_{up}(n) + \gamma\frac{1}{C_{up}} + \gamma\frac{1}{C_{dn}} * \frac{Thp_{dn}(n) - Thp_{dn}(n-1)}{Thp_{up}(n) - Thp_{up}(n-1)}\,. \tag{9}$$

As said earlier in DCBQ, the queue will be separated into ACKs queue and data queue. Each queue is allocated a fraction of the bandwidth C_{up} and it dynamically changes according to the crossing traffic in both directions.

The aim of DCBQ is to find in a minimum number of intervals T, the allocation scheme that allows the optimal user satisfaction. The weight of the Data class is defined as the variable $Thp_{up}(n) = R(n)/C_{up}$, $Thp_{up}(n)$ Є [0,1]. The weight of the ACK class is then equal to 1- $Thp_{up}(n)$. At each interval T the weight $Thp_{up}(n)$ is updated. T represents the bandwidth allocation updating interval; its value is a tradeoff between stability and responsiveness of the system. T must be long enough to permit the traffic to respond to a change in the bandwidth allocation. At the same time, T cannot be very long since this alters the consistency of the system and slows its response to any change in traffic conditions. γ factor decides on the amount by which the rates of DCBQ are updated every T. Giving a big value for γ will quickly lead us to an unstable state, and a too small value of γ will necessitate very long convergence time. γ must help the system to avoid big oscillations and to converge to the stable state as fast as possible. The choice of γ involves then a clear tradeoff. The unit of γ is $Kbps^2$.

Simulation environment and results

Fig. 2 shows the simulation topology used to model a network bandwidth asymmetry, the bandwidth and delay parameters have been chosen to closely model the bandwidth asymmetric networks. The model network with n number of nodes was simulated using the NS-2.29. The link R_1 to R_2 is the forward link and is given a bandwidth of 2Mbps, 10ms propagation delay and the link R_2 to R_1 is the reverse link with a bandwidth of 56Kbps, 5ms propagation delay.

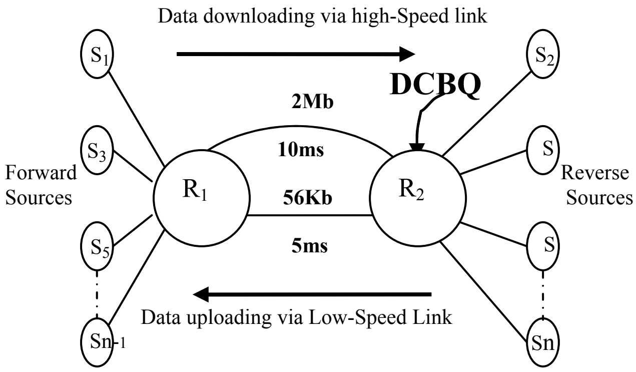

Fig. 2. Topology of Simulation

Downloading takes place using the forward link and the uploading takes place with the reverse link. The nodes $S_1 - S_{n-1}$ are assumed to be downloading sources (left side of R_1) and the other nodes $S_2 - S_n$ are uploading sources, both sources are using TCP connection. All flows are long-lived. Using TCP Reno, packets are 1000 bytes for data and 40 bytes for ACKs. The forward buffer R_1 is 100 packets and the reverse buffer R_2 is 20 packets.

Performance analysis for CBQ-WRR Algorithm

The CBQ-WRR scheduling is implemented in the link R_2 to R_1 in the network topology shown in Fig. 2. In class based queuing, the reverse queue is divided in to two virtual queues, One for data packets and the other for the ACK packets. The bandwidths of the virtual queues are assigned with a percentage bandwidth of the original queue. The performance for CBQ has been evaluated with three sets of virtual queue bandwidths, (0.2, 0.8), (0.5, 0.5) and (0.8, 0.2) for ACK and data packets respectively.

The downstream throughputs are plotted in Fig. 3. It can be seen that when the data class value is 0.2 and ACK class is given a value of 0.8, the downstream throughput is very high. This is because ACK class occupies major part of the queue and never gets interrupted due to the data packets. As ACKs reach in time, the TCP flow in the forward direction is continuous and uninterrupted. But when it is other way round (i.e.) the ACK class taking 0.2 and data class taking 0.8, the downstream throughput is low. This is because; the ACK class has lesser bandwidth. In case of 0.5 for both the classes, it can be seen that the throughput falls in between the other two downstream throughputs.

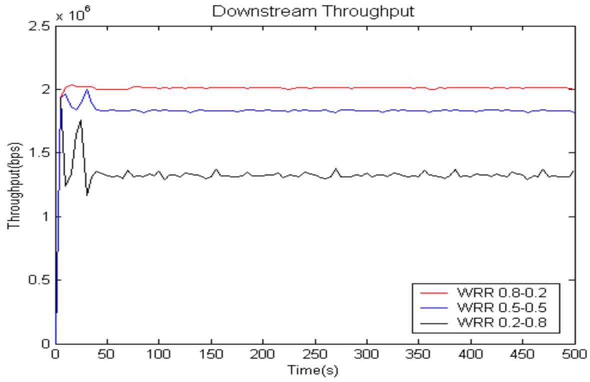

Fig. 3. Comparison of Throughput in Forward Link

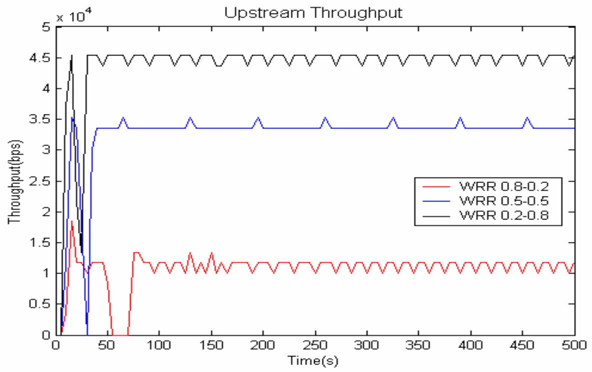

Fig. 4. Comparison of Throughput in Reverse Link

It can be seen that the upstream is in contradict with the downstream as shown in Fig. 4. It is very clear that there exist a tradeoff between the downstream and upstream throughputs. So it is very important to fix up the bandwidth for the virtual queues based on the network characteristics.

CBQ has its own limitations. The data rate in the network is unpredictable in real time. So it is very necessary to adjust the queue rate dynamically. Static queue rate fixation can lead to wastage and inefficient use of bandwidth. Hence dynamic allocation of class based queuing in which both the queue bandwidth rates varies at every time interval based on the data rate in forward and reverse link is required. The proposed DCBQ algorithm would overcome the limitations of CBQ algorithm, by adjusting the available bandwidth between the opposite directions dynamically.

Performance analysis for DCBQ Algorithm

To meet DCBQ requirements, NS-2 Simulator has been modified. The DCBQ algorithm is known to adapt itself to the traffic flow in the network. When more ACKs flow through the ACK-queue, the bandwidth is raised and in contrary the bandwidth decreases in the Data-queue. Hence whenever there is a need to have more bandwidth the algorithm realizes it and allocates accordingly. The DCBQ algorithm is implemented for both Reno and Reno with delayed ACKs called (Enhanced-DCBQ). Hence one could expect that Enhanced-DCBQ (E-DCBQ) would have a better throughput because of lesser flow of ACK packets which again reduces the congestion in reverse link.

Bandwidth sharing in DCBQ

Once the simulation starts, first the static parameters such as bandwidth of forward link (C_{dn}), reverse link (C_{up}), time interval (T) for which the ratio is calculated and the convergence factor (γ) are set. Both the ACK-queue and Data-queue are initially set to 0.5 and the algorithm is run. Now the bandwidth for ACK-queue and Data-queue are calculated at every time interval using the equation (9). Hence it can be seen that the virtual queue value changes at every time period (T) and adjusts itself in accordance with the networks flow rate. Fig. 5 shows the variation of data and ACK queue rates versus the simulation time, the simulation time duration is chosen to let DCBQ to adapt to the best utility function.

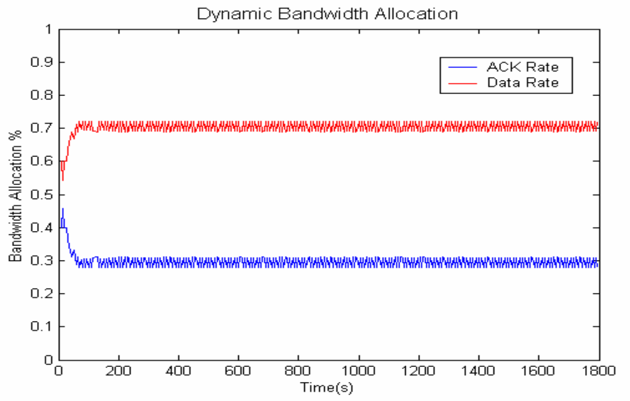

Fig. 5. Bandwidth sharing in DCBQ Algorithm

Throughput performance

The network designed has a forward link bandwidth of 2Mb and reverse link of 56 Kb and 10 nodes in each direction. The downstream throughput can be seen to be extremely good as shown in Table 1. The average value of the throughput in forward link is seen to remain constant around 1.6 Mb. Hence the DCBQ is proved successful as far as forward links are concerned. The data throughput in the reverse direction with respect to E-DCBQ has an average throughput of 39 Kbps and Reno has around 35 Kbps. The utilization is seen around 69 % and 63 % respectively which is very acceptable at upstream direction. Hence it can be seen that both the forward and reverse link provide good throughput which is the main aim for the network. The overall utilization of the resources is more than 1.40 which is very good.

Table 1. Performance of DCBQ & E-DCBQ Algorithms

ALGORITHM	**No. of Nodes**	**Avg. Thp_{dn}**	**Avg. Thp_{up}**	**$U(thp_{dn})$**	**$U(thp_{up})$**
DCBQ	1605868	35233	0.7657	0.6292	1.3949
E-DCBQ	1574957	39057	0.7510	0.6974	1.4484

Changing the network load

To check the robustness of DCBQ algorithm, the number of connections having simultaneous transfer of data is varied. Simulations are done by varying number of nodes on each side of the DCBQ Link. The previous simulations are for 10 connections on both sides of DCBQ link. Simulations are done by changing the number of nodes to 5, 15, 25 and 35 on each side of DCBQ link. It is seen that the value of the total user satisfaction U_{tot} is maintained around 1.41. This result confirms that DCBQ algorithm works well with varying loads/nodes, proving the robustness of the algorithm. The results are tabulated in Table 2.

Table 2. DCBQ with varying nodes

ALGORITHM	**No. of Nodes**	**Avg. Thp_{dn}**	**Avg. Thp_{up}**	**$U(thp_{dn})$**	**$U(thp_{up})$**	**U_{tot}**
DCBQ	1	1605868	35233	0.7657	0.6292	1.3949
	5	1505636	39241	0.7179	0.7007	1.4187
	15	1489817	39332	0.7104	0.7024	1.4128
	25	1484411	39329	0.7078	0.7023	1.4101
	35	1496941	39340	0.7138	0.7025	1.4163

Changing the Traffic type

The DCBQ algorithm is also analyzed using Exponential traffic to handle short-lived files. TCP connections with On/Off flows that have an exponential distribution are considered. The average period for On/Off sources is kept as 0.5ms. The Packet size is kept as 1000 bytes and 200Kbps of transmission rate, with 10 sources on both direction of DCBQ link. Table 3 shows the results for different algorithms with exponential source. DCBQ manages well with short lived data packets, by maintaining a total customer satisfaction U_{tot} value more than 1.40. Thus the results confirm that DCBQ works well with short lived traffic of TCP also.

Table 3. Various algorithms with exponential source

ALGORITHM	Avg.Thp_{dn}	Avg.Thp_{up}	$U(thp_{dn})$	$U(thp_{up})$	U_{tot}
DCBQ + EXP	1579969	36214	0.7534	0.6467	1.4001
Reno + EXP	1680947	35345	0.8015	0.6312	1.4327
WRR + EXP	1833879	33408	0.8745	0.5966	1.4710

Comparison between DCBQ and other algorithms

The superiority of the DCBQ algorithm can be demonstrated by comparing it with all the algorithms implemented so far. The throughput analysis of the algorithms together could give a clear picture about the network performance with respect to each algorithm implemented. The parameters of the network are kept uniform for all the algorithms so that comparisons can be charted clearly in graphs.

Table 4 charts out the average downstream and the upstream throughputs of all the algorithms that were simulated. The tabular representation gives a very clear idea about the user satisfaction of all the algorithms. It can be seen that the E-DCBQ algorithm has the best user satisfaction and then comes DCBQ. The ACK-First Scheduling has the least user satisfaction.

Table 4. Throughput Comparison

ALGORITHM	**Avg. Thp_{dn}**	**Avg. Thp_{up}**	**$U(thp_{dn})$**	**$U(thp_{up})$**	**U_{tot}**
Reno	1963169	17049	0.9361	0.3044	1.2406
ACKs First	1999523	0	0.9534	0.0000	0.9534
WRR 0.55	1838324	32559	0.8766	0.5814	1.4580
DCBQ	1605868	35233	0.7657	0.6292	1.3949
E-DCBQ	1574957	39057	0.7510	0.6974	1.4484

The performances of all algorithms discussed are compared and is shown in the Figures 6 and 7. It can be seen that TCP Reno with ACK first has highest downstream throughput but the lowest upstream throughput as noted earlier. WRR algorithm has a good level of throughput in both upstream and downstream directions but its well known fact that it has its own limitations and setting the size of virtual queues is most crucial factor. DCBQ gives the best results in the forward and reverse link and the utilization rate is around 75% which is very good. Hence it can be very clearly observed that DCBQ is superior compared to all algorithms discussed. Moreover E-DCBQ has better results as only one ACK is sent for every two data packets. This reduces the congestion in the reverse link to a good extent and provides excellent performance.

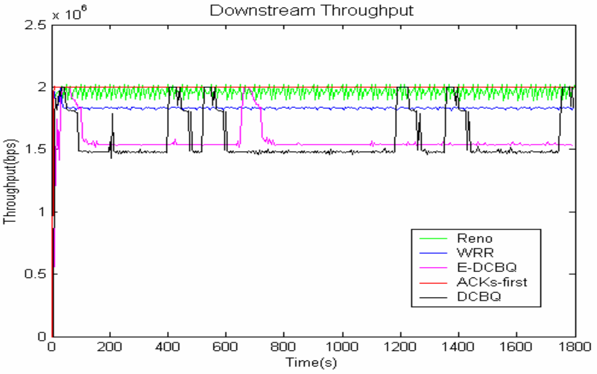

Fig. 6. Comparison of Throughput in Forward Link

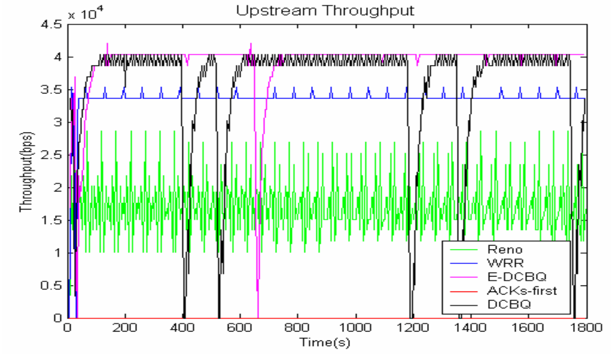

Fig. 7. Comparison of Throughput in Reverse Link

Moreover any algorithm is said to have good performance only when user satisfaction is high. It can be seen that the utility function has better value which ensures about the good performance of the algorithm. The WRR 0.5-0.5, DCBQ and E-DCBQ have the highest user satisfaction with more than 1.41. The algorithm is capable of adapting itself to any kind of data rate and provides necessary bandwidth for packet transmission.

Conclusion

The TCP performance in asymmetric networks has been discussed, especially in bidirectional bandwidth asymmetric networks. The data rate in the network is unpredictable in real time. Hence a novel algorithm for handling bidirectional traffic has been proposed and analyzed. The simulated results show that the proposed technique provides better performance than the existing techniques in terms of throughput ad user satisfaction factor. The results show that DCBQ is a robust scheduling mechanism for efficient TCP in a variety of asymmetric conditions.

References

1. **Mohan Madav Shankar M., et al.** Quality of Service for the last Mile: Final Project, ECPE 6504, Bradley Department of Electrical and Computer Engineering. – Virginia Tech, Spring, 2001. – P. 1–6.
2. **Balakrishnan H., Venkata N. Padmanabhan.** How Network Asymmetry Affects TCP // IEEE Communications Magazine. – 2001. – P. 60–67.
3. **Fatma Louati, et al.** Handling two-way TCP traffic in asymmetric networks // Proceedings of 7th IEEE International Conference, HSNMC 2004. – Toulouse, France, 2004. – P. 233–243.
4. **Balakrishnan H., et al.** RFC 3449 – TCP performance implications of Network path Asymmetry, December 2002.
5. **Lakshman T.V., et al.** TCP/IP Performance with random Loss and Bidirectional Congestion, IEEE/ACM Transactions on Networking, October 2000. – P. 541–555.
6. **Liu Yu, Ye Minhua, Zhang Huimin.** The Improvement of TCP Performance in Bandwidth Asymmetric Network // 14th IEEE 2003 International Symposium on Personal, Indoor and Mobile Radio Communication Proc., 2003. – P. 482–486.
7. **Huei-Jiun Ju and Wanjiun Liao.** Fast Request Transmission in DOCSIS-based CATV Networks // Proceedings of IEEE International Conference on Multimedia and Expo (ICME) '02. – Lausanne, Switzerland, Aug. 2002.
8. **Haitao Wu, et al.** ACK Filtering on Bandwidth Asymmetry Networks // Proceedings of APCC/OECC'99, Beijing, 1999. – P. 175–178.
9. **Shreedhar M. and George Varghese.** Efficient Fair Queueing Using Deficit Round-Robin // IEEE/ACM Transactions on Networking, June 1996. – P. 375–385.

Submitted for publication 2006 06 06

Wissam Al-Khatib, K. Gunavathi. A New Approach to Improve TCP Performance over Asymmetric Networks // Electronics and Electrical Engineering. – Kaunas: Technologija, 2006. – No. 7(71). – P. 13–18.

TCP performance issues in bandwidth asymmetric networks are investigated. Related works to improve TCP performance in bandwidth asymmetric networks are introduced and analyzed. A new approach to improve TCP performance in two way traffic over asymmetric networks is also proposed in this paper, which can further improve TCP performance by assigning the bandwidth rates over the reverse link dynamically. Simulation results show that the new scheme has better performance compared to existing algorithms in terms of good utilization of the existing resources and higher throughput**.** Ill. 7, bibl. 9 (in English; summaries in English, Russian and Lithuanian).

Виссам Аль-Кхатиб, К. Гунаватхи. Новый метод увеличения TCP эффективности в асимметрических сетях // Электроника и электротехника. – Каунас: Технология, 2006. – № 7(71). – С. 13–18.

Анализируются вопросы TCP эффективности в асимметричных сетях. Представлены и рассматриваются вопросы улучшения TCP эффективности. Рассмотрен новый метод увеличения TCP эффективности при передаче двунаправленного потока данных, на основе которого и дальше может увеличаться TCP эффективность при динамическом изменении скорости обратного потока данных. Результаты моделирования показали, что новый метод отличается большей эффективностью при сравнении с уже имеющимися алгоритмами. Ил. 7, библ. 9 (на английском языке; рефераты на английском, русском и литовском яз.).

Wissam Al-Khatib, K. Gunavathi. Naujas metodas TCP efektyvumui asimetriniuose tinkluose padidinti // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2006. – Nr. 7(71). – P. 13–18.

Nagrinėjamas TCP efektyvumas asimetriniuose tinkluose. Iškeliami ir analizuojami TCP efektyvumo gerinimo klausimai. Teikiamas naujas metodas TCP efektyvumui padidinti perduodant dvikryptį duomenų srautą asimetriniuose tinkluose, kuriuo remiantis toliau gali būti didinamas TCP efektyvumas dinamiškai keičiant atgalinio duomenų srauto perdavimo spartą. Modeliavimo rezultatai rodo, kad naujas metodas pasižymi didesniu efektyvumu palyginti su esamais algoritmais, įvertinant geresnį turimų išteklių panaudojimą ir didesnį pralaidumą. Il. 7, bibl. 9 (anglų kalba; santraukos anglų, rusų ir lietuvių k.).