A Novel Mechanism to Improve Performance of TCP Protocol over Asymmetric Networks

Wissam Al-Khatib and K. Gunavathi
Department of Electronics and Communication Engineering, PSG College of Technology, India

Abstract: *In this paper, 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 mechanism 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 mechanism has better performance compared to existing algorithms in terms of good utilization of the existing resources and higher throughput.*

Keywords: *asymmetric networks, TCP performance, ACK, bandwidth allocation.*

Received May 1, 2006; Accepted August 8, 2006

1. 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 (ADSL), satellite based networks are aimed at alleviating the "last mile" bottleneck [11], 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 [1]. 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, is often limited compared to the downstream bandwidth towards the customer premises. In certain cases upstream communication may just become impossible [9].

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 [2]. 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 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 [10].

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.

This paper is organized as follows. Section 2 deals with classification of asymmetric networks. In section 3 a brief survey on techniques to improve the TCP performance has been discussed. Section 4 describes the proposed algorithm. Simulation results and performance comparison are discussed in sections 5 and 6. Finally, conclusion about the performance of the proposed algorithm has been stated.

2. Classification of Asymmetric Networks

The asymmetric networks have been classified depending on the characteristics of networks into three kinds [3, 14]: bandwidth asymmetry, media access asymmetry, and link bit error rate asymmetry. In bandwidth asymmetry, the forward link bandwidth is 10-1000 times greater than the reverse link bandwidth. Media access asymmetry is difference of access characteristics in different directions. Such as cellular networks, when the base station sends data to the mobile terminals, the overhead of the media access layer is large, while in reverse direction it is small. 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 [5, 7], which is ratio of raw bandwidth divided by the ratio of the packet sizes used in two directions as given in Equation 1.

$$K = \frac{C_{dn}}{C_{up}} * \frac{L_{ack}}{L_{data}} \quad (1)$$

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 10 Mb downlink and 100 Kb 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 [2]. On average only one acknowledgement will get through for every k data packets transmitted by sender which could degrade performance in several ways. First, the sender can burst out k packets at a time which could lead to loss of data especially when k is large. Secondly, the congestion window grows on counting the number of acknowledgements. Infrequent acknowledgements would result in slower growth. Finally, the loss of acknowledgements elsewhere in networks would cause long ideal periods while the sender waits for subsequent acknowledgements to arrive.

In case of bidirectional data transfer, the data flow occurs in both forward and reverse direction simultaneously. This becomes more complicated [RFC 3449]. 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 [3]. 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 [9].

In summary, the presence of bidirectional data transfer would aggravates 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.

3. Related Work

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 [3]. 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 (AF), ACK Congestion Control (ACC) and ACKs-first scheduling. TCP header compression [RFC 1144] greatly reduces the size of ACKs on the uplink. When losses are infrequent, it is recommended to use over low-bandwidth uplinks. AF is a TCP-aware link layer technique that reduces the number of TCP ACKs sent on the upstream channel [13]. AF removes only certain ACKs without starving the sender by taking advantage of the fact that TCP ACKs are cumulative. The previous two techniques require changes on the receiver side. Other solutions require a change in both sender and receiver side, such as sender adaptation and ACC [1, 3]. It is an alternative to ACK filtering that operates end-to-end rather than at the upstream bottleneck router [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 behind 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.

4. Dynamic Class Based Queuing

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. Towards this area a novel queuing mechanism called Dynamic Class Based Queuing (DCBQ) has been proposed in this paper. DCBQ is an extension of ACQ queuing mechanism [9]. In DCBQ Weighted Round Robin (WRR) scheduling algorithm [6] has been used instead of Fair Queuing (FQ) scheduling between the different classes. WRR scheduling algorithm is simpler to implement and is less complex compared to FQ scheduling algorithm [12].

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.

4.1. DCBQ Algorithm

Figure 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) [4]. 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 WRR scheme.

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.

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

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 as given in Equation 3.

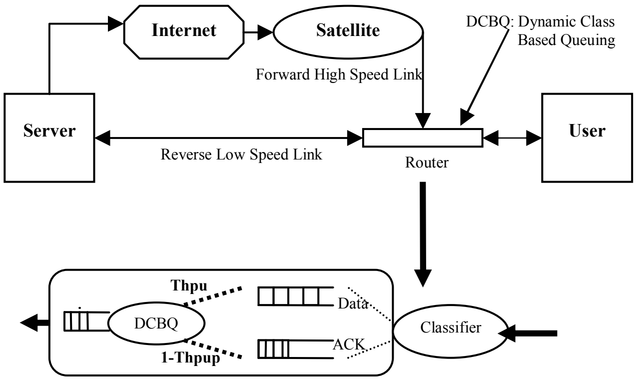

Figure 1. Architecture of asymmetric network.

$$Thp_{dn} = \frac{N_{down} * Size\ of\ Forward\ Data\ Packet}{T} \quad (3)$$

where, N_{down} is the number of acknowledgement packets generated at the receiver and received at ACK queue, which is equal to the number of data packets received in the forward link.

The upstream throughput will be measured as given in Equation 4.

$$Thp_{up} = \frac{N_{up} * \text{Size of Reverse Data Packet}}{\text{T}} \quad (4)$$

where, N_{up} is the number of data packets transmitted in the reverse direction.

The proposed algorithm measures the data rate in both directions and dynamically changes bandwidth allocated for the data queue of upstream traffic. The ACK queue will be allocated the rest of the bandwidth which maintains forward traffic, as explained mathematically below:

Let $R(n)$ be the rate allocated to the data queue in the upstream direction at time nT, $n \in N$. 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.

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).

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

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 Equation 5 the data rate is given by:

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

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

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

Using Equation 7 in Equation 6, the rate allocated to the upstream Data queue will become,

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

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,

$$U_{up}(Thp_{up}) = \frac{Thp_{up}}{C_{up}} \quad (9)$$

$$U_{dn}(Thp_{dn}) = \frac{Thp_{dn}}{C_{dn}} \quad (10)$$

From Equations 9 and 10, the allocated rate of upstream data queue becomes

$$R(n+1) = Thp_{up}(n) + \gamma \frac{1}{C_{up}} + \gamma \frac{1}{C_{dn}} * \frac{Thp_{dn}(n) - Thp_{dn}(n-1)}{Thp_{up}(n) - Thp_{up}(n-1)} \quad (11)$$

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 $Thp_{up}(n) = R(n)/C_{up}$, $Thp_{up}(n) \in [0, 1]$. 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. Since TCP adapts its window size every two Round-Trip Times (RTT), the minimum value of *T* should be twice the bigger round trip time of the connections. 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.

5. Simulation Environment and Results

Figure 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 2 Mbps, 10 ms propagation delay and the link R_2 to R_1 is the reverse link with a bandwidth of 56 Kbps and 5 ms propagation delay.

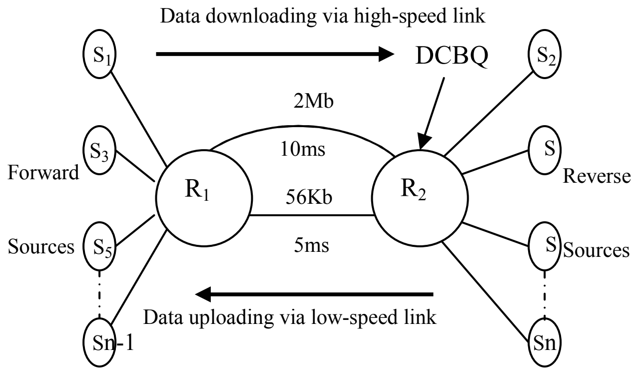

Figure 2. Topology of simulation.

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 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.

5.1. Performance Analysis for CBQ Algorithm

The CBQ is implemented in the link R_2 to R_1 in the network topology shown in Figure 2. Both Packet Round Robin (PRR) and WRR scheduling techniques are implemented. 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 for ACK and data packets.

The downstream throughputs are plotted for PRR and WRR in Figure 3. It can be seen that when the data class bandwidth is 0.2 and ACK class bandwidth is 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. 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. From Figure 3, it can be seen that WRR have more throughput when compared to the throughput of PRR. So WRR scheduling is preferred over PRR scheduling.

It can be seen that the upstream is in contradict with the downstream as shown in Figure 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.

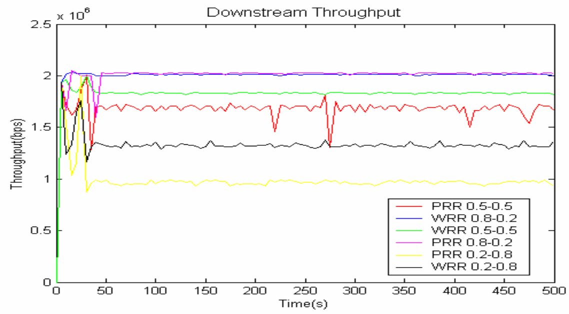

Figure 3. Comparison of throughput in forward link.

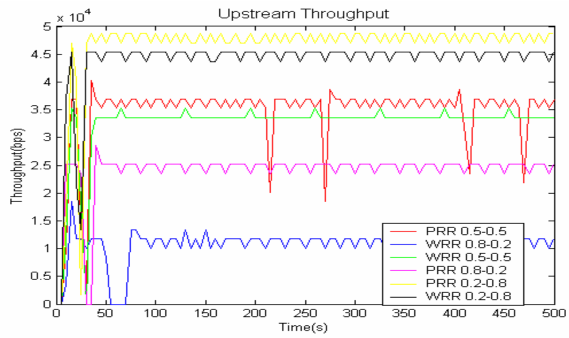

Figure 4. Comparison of throughput in reverse link.

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 vary at every time interval based is required.

5.2. 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 (E-DCBQ). Hence one could expect that (E-DCBQ) would have a better throughput because of lesser flow of ACK packets which again reduces the congestion in reverse link.

5.3. 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 11.

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. Figure 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. As it can be seen from Figure 5, the data class obtained 70% of the available bandwidth, leaving the other 30% to the ACK class. Also it is observed that DCBQ converge rapidly to those allocation ratios. So the DCBQ scheduler is stable. Here the value of T is 5 seconds and 10Kbps for γ.

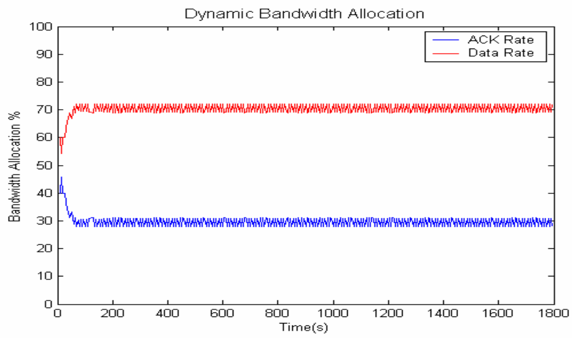

Figure 5. Bandwidth sharing in DCBQ algorithm.

5.4. Throughput Performance

The network designed has a forward link bandwidth of 2 Mb 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 Figure 6. The value of the throughput in forward link is seen to remain around 1.6 Mb. Hence the DCBQ is proved successful as far as forward links are concerned. The throughput in the reverse direction with respect to E-DCBQ has an average throughput of 39 Kbps and DCBQ has around 35 Kbps as seen in Figure 7. 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 is more than 1.40 which is very good. The results are tabulated in Table1.

Table 1. Performance of DCBQ & E-DCBQ algorithms.

ALGORITHM	Avg. Thp_{dn}	Avg. Thp_{up}	$U(thp_{dn})$	$U(thp_{up})$	U_{tot}
DCBQ	1605868	35233	0.7657	0.6292	1.3949
E-DCBQ	1574957	39057	0.7510	0.6974	1.4484

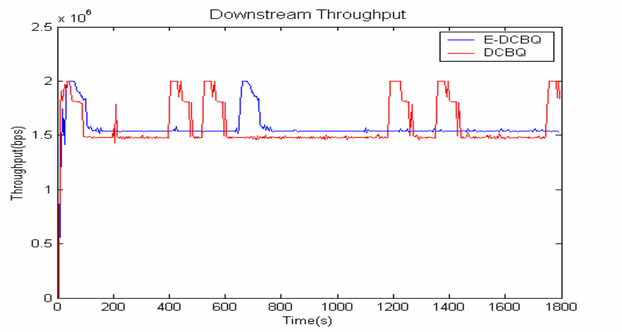

Figure 6. DCBQ & E-DCBQ forward link throughput.

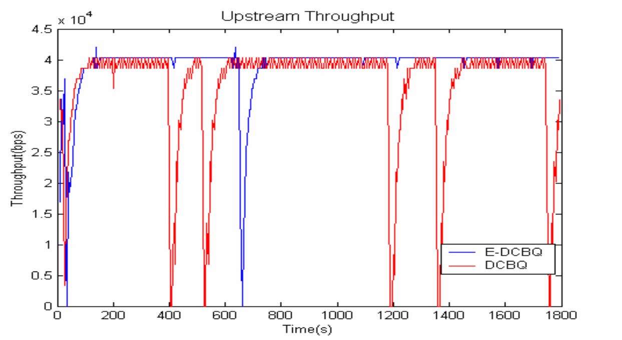

Figure 7. DCBQ & E-DCBQ reverse link throughput.

5.5. 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. Figures 8 and 9 shows the downstream and upstream throughput for varying number of nodes respectively.

Table 2. DCBQ with varying nodes.

ALGORITH -EMS	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

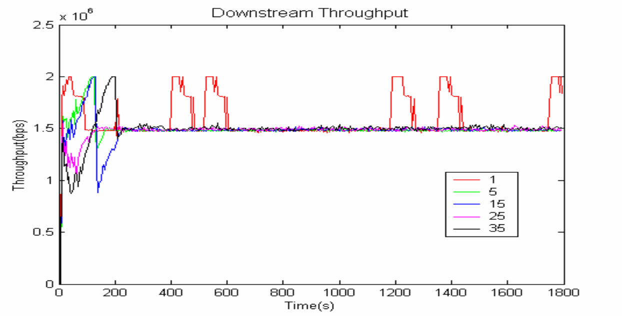

Figure 8. DCBQ forward link throughput vs. number of nodes.

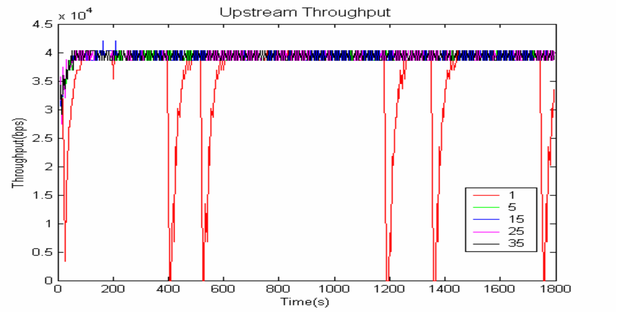

Figure 9. DBCQ reverse link throughput vs. number of nodes.

5.6. 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 0.5 ms. The Packet size is kept as 1000 bytes and 200 Kbps of transmission rate, with 10 sources on both direction of DCBQ link. 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 shows the results for different algorithms with exponential source. It is observed that DCBQ algorithm compared to other algorithms have better throughput in upstream, as well as for downstream the throughput is good. Results are shown in Figures 10 and 11.

5.7. Changing Asymmetric Factor *k*

The numbers of connections in each direction has been set to 10 nodes, the values of T and γ are 5 sec and 10 Kbps respectively and the capacity of reverse link is 56Kbps.

Table 3. Various algorithms with exponential source.

ALGORITHEM	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
PRR+EXP	1962361	17108	0.9357	0.3055	1.2412
WRR+EXP	1833879	33408	0.8745	0.5966	1.4710

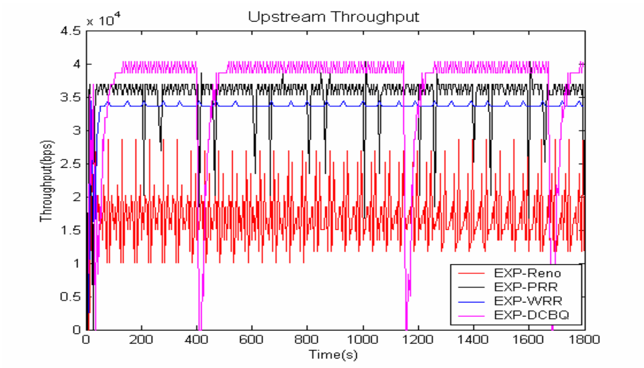

Figure 10. DCBQ forward link throughput vs. on/off traffic.

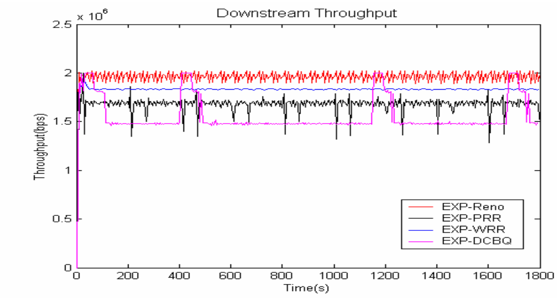

Figure 11. DBCQ reverse link throughput vs. on/ off traffic.

The capacity of forward link has been changed from 1 Mb which is nearly 20 times of reverse link capacity, till 10 Mb which is 182 times of reverse link. Table 4 shows that DCBQ has very good performance while changing the value of forward link when it will be 1 Mbps and 2 Mbps, but when forward link is 10Mb, the DCBQ algorithm behaves well compared to other existing algorithms as it maintains good throughput in both directions, which is not available in other algorithms, as given in Table 4.

Table 4. Effects of asymmetric degree on various algorithms.

ALGOR-ITHEM	C_{dn}	Avg.Thp_{dn}	Avg.Thp_{up}	U(Thp_{dn})	U(Thp_{up})	U_{tot}
DCBQ	1 Mb	932050	30065	0.8889	0.5369	1.4257
Reno		777018	32340	0.74	0.56	1.30
ACK-First		990050	17758	0.94	0.30	1.24
DCBQ	2 Mb	1605868	35233	0.7657	0.6292	1.3949
Reno		1963169	17049	0.9361	0.3044	1.2406
ACK-First		1999523	0	0.9534	0.0000	0.9534
DCBQ	10 Mb	1844121	38722	0.1759	0.6915	0.8673
Reno		473722	50317	0.0452	0.8985	0.9437
ACK-First		4728869	0	0.4509	0.0000	0.0509

In case of changing the reverse direction rate, the proposed algorithm maintains good utilization for upstream and downstream for every case to be nearly same. Table 5 shows the effects of changing C_{up} from very less bandwidth 9.6 Kbps to 56 Kbps, while downstream link capacity is 2 Mbps. As shown by simulation results the user satisfaction factor "U (Thp_{dn})" and "U (Thp_{up})" will be nearly same so there is no monopolization of data at any direction, this gives more robustness to the proposed algorithm.

Table 5. Effects of upstream data rate on DCBQ algorithm.

ALGO-RITHEM	C_{up}	Avg.Thp_{dn}	Avg.Thp_{up}	U(Thp_{dn})	U(Thp_{up})	U_{tot}
DCBQ	9.6 Kb	668982	3378	0.3190	0.3519	0.6709
	28 Kb	1060248	16208	0.5056	0.5789	1.0844
	56 Kb	1605868	35233	0.7657	0.6292	1.3949

6. 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 6 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 highest user satisfaction and then comes DCBQ. The ACK-First Scheduling has the least user satisfaction.

Table 6. Throughput comparison table.

ALGORITHEM	Avg.Thp_{dn}	Avg.Thp_{up}	U(Thp_{dn})	U(Thp_{up})	U_{tot}
Reno	1963169	17049	0.9361	0.3044	1.2406
ACKsFirst	1999523	0	0.9534	0.0000	0.9534
PRR0.5.5	1684318	34709	0.8031	0.6198	1.4229
WRR0.55	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 12 and 13. 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.

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.

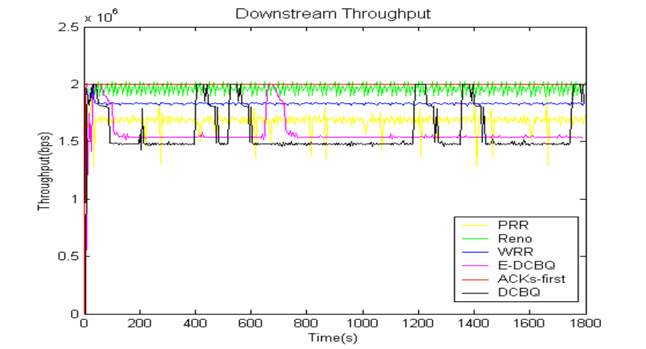

Figure 12. Comparison of throughput in forward link.

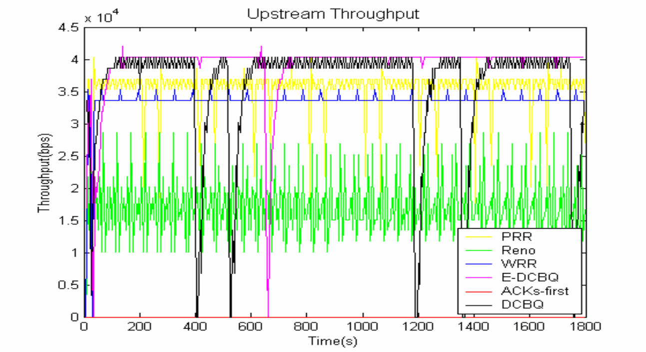

Figure 13. Comparison of throughput in reverse link.

7. 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] Balakrishnan H., Venkata N., and Randy H., "The Effects of Asymmetry on TCP Performance," *in Proceedings of the 3rd IEEE/ACM International Conference on Mobile Computing and Networking*, Hungary, pp. 1-13, 1997.

[2] Balakrishnan H., Padmanabhan V., and Katz R., *RFC 3449 – TCP Performance Implications of Network Path Asymmetry*, December 2002.

[3] Balakrishnan H. and Venkata N., "How Network Asymmetry Affects TCP," *IEEE Communications Magazine*, pp. 60-67, 2001.

[4] Floyd S. and Speer M., "Experimental Results for Class-based Queuing," http://ftp.ee.lbl.gov/floyd/cbq/report.pdf, 1998.

[5] Ju H. and Liao W., "Fast Request Transmission in DOCSIS-based CATV Networks," *in Proceedings of IEEE International Conference on Multimedia and Expo (ICME'02)*, Lausanne, Switzerland, 2002.

[6] Katenis M., Katevenis M., Sidiropoulos S., and Courcoubetis C., "Weighted Round-Robin Cell Multiplexing in a General-Purpose ATM Switch Chip," *IEEE Journal on Selected Areas in Communication*, pp. 1265-1279, 1991.

[7] Lakshman T., Lakshman T., Madhow U., and Suter B., "TCP/IP Performance with Random Loss and Bidirectional Congestion," *IEEE/ACM Transactions on Networking*, pp. 541-555, 2000.

[8] Lakshman T., Madhow U., Suter B., Bell L., Lucent T., and Holmdel N., "Window-based Error Recovery and Flow Control with a Slow Acknowledgement Channel: A Study of TCP/IP Performance," *INFOCOM'97*, pp. 1199-1210, 1997.

[9] Louati F., Barakat C., and Dabbous W., "Handling Two-way TCP Traffic in Asymmetric Networks," *in Proceedings of the 7th IEEE International Conference, High Speed Networks and Multimedia Communications 2004*, Toulouse, France, pp. 233-243, 2004.

[10] Raisinghani V., Patil A., and Iyer S., "Mild Aggression: A New Approach for Improving TCP Performance in Asymmetric Networks," *in Proceedings of Asian International Mobile Computing Conference*, Malaysia, 2000.

[11] Shankar M., Parthasarathy R., and Surendranath, "Quality of Service for the Last Mile," *Final Project,* ECPE 6504, Bradley Department of Electrical and Computer Engineering, Virginia Tech, 2001.

[12] Shreedhar M. and Varghese G., "Efficient Fair Queueing Using Deficit Round-Robin," *IEEE/ACM Transcations on Networking*, pp 375-385, 1996.

[13] Wu H., *et al*, "ACK Filtering on Bandwidth Asymmetry Networks," *in Proceedings of APCC/OECC'99*, pp. 175-178, Beijing, 1999.

[14] Yu L., Minhua Y., and Huimin Z., "The Improvement of TCP Performance in Bandwidth Asymmetric Network," *14th IEEE 2003 International Symposium on Personal, Indoor and Mobile Radio Communication Proceedings*, pp. 482-486, 2003.

Wissam Al-Khatib graduated from Damascus University, Syria, in 1998. Completed PG Diploma in computer engineering and control systems during 2000 from Damascus University, Syria. In 2003, he completed Master Degree in communication systems from PSG College of

Technology under Bharathier University Coimbatore, India. Currently, he is a research scholar at the Department of Electronics and Communication Engineering, PSG College of Technology, Coimbatore, under Anna University. His Areas of interest include broadband computer networks, digital communications, and fiber optical communication.

K. Gunavathi is an assistant professor, Department of Electronics and Communications, PSG College of Technology, Coimbatore, India. Her areas of interest include VLSI design, digital communication, and computer networks.