INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Handling Two-Way TCP Traffic in Bandwidth Asymmetric Networks

Fatma Louati — Chadi Barakat — Walid Dabbous

N° 4950

October 2003

THÈME 1

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--4950--FR+ENG

Handling Two-Way TCP Traffic in Bandwidth Asymmetric Networks

Fatma Louati , Chadi Barakat , Walid Dabbous

Thème 1 — Réseaux et systèmes
Projet PLANETE

Rapport de recherche n° 4950 — October 2003 — 25 pages

Abstract: The TCP congestion control protocol assumes bandwidth symmetric paths. As two-way asymmetric connections will probably become common case in the future with the widespread use of ADSL and satellites technologies, it will be important to assure that congestion is properly handled in these environments. A lot of researches were done to find way to do this, but most of them are ad hoc solutions unable to go far in solving the general problem.
In this context, we propose two new approaches for handling two-way traffic over links that exhibit bandwidth asymmetry: an Adaptive Class-based Queuing mechanism called ACQ and a Virtual Ack-based Queuing mechanism called VAQ. Both mechanisms runs at the entry of the slow link. ACQ relies on two separate classes, one for Ack packets and one for Data and proposes to adapt the weights of both classes according to the crossing traffic and to a Utility Function defined by the user or the network operator. VAQ relies on two virtual classes, one for Ack packets and one for Data. It grants a credit to the Data class and adapts this credit according to the crossing traffic through a WFQ scheduling scheme. We show by simulations that our mechanisms are able to reach a good utilization of the available resources managing then to maximize the satisfaction of the user of such asymmetric connections. We also demonstrate by simulations the robustness of ACQ and VAQ when confronted to different network settings.

Key-words: TCP, bandwidth asymmetry, two-way traffic, ack compression, resource utilization, performances, utility function, packet scheduling

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Gestion efficace de trafic bidirectionnel passant par des liens asymétriques

Résumé : Le protocole TCP (Transmission Control Protocol) constitue les fondements de l'Internet d'aujourd'hui. En effet entre autres importantes fonctions, il assure un contrôle fiable de la congestion du réseau permettant ainsi une utilisation efficace de la bande passante disponible et une stabilité du réseau. Le protocole TCP se base sur les acquittements pour réguler son trafic et contrôler la congestion. On appelle réseau asymétrique tout réseau ayant un chemin de retour non suffisament rapide pour porter le flux d'acquittements générés par le récepteur TCP. Ces réseaux sont de plus en plus présents dans l'Internet avec le développement des réseaux satellites et de l'ADSL. Un lent chemin de retour entraîne une congestion au niveau du trafic d'acquittements. Cette congestion résulte en un retard important ou carrément une perte d'acquittements. Le réseau est alors en présence du problème de "Ack-Compression" qui cause des difficultés pour le bon fonctionnement de TCP. De plus, l'ajout d'un traffic bidirectionnel pourtant de plus en plus fréquent, exacerbe les effets du problème de Ack-Compression puisque acquittemnets et paquets de données partagent la même ressource alors qu'ils sont si différents de part leur taille et leur réaction face à des notifications de congestion.

Dans ce contexte nous proposons deux nouvelles approches pour la gestion et l'ordonnancement de trafics bidirectionnels passant par des liens asymétriques: ACQ pour Adapative Class-based Queuing et VAQ pour Virtual Ack-based Queuing. Les deux mécanismes se placent à l'entrée du lien asymétrique. ACQ se base sur deux classes distinctes, une pour les paquets de données et une pour les acquittements et propose d'adapter les poids des classes selon le trafic. VAQ de son coté se base sur des files d'attente virtuelles, une pour les paquets de données et une pour les acquittements. VAQ accorde un crédit à la file de données et propose d'adapter ce crédit selon le trafic tout en suivant une politique de WFQ. Nous montrons grâce à des simulations fiables, que ACQ et VAQ permettent d'améliorer considérablement l'utilisation des liens du réseau et d'atteindre ainsi la satisfaction de l'utilisateur. De plus, nous testons la robustesse et la stabilité de ACQ et de VAQ face à des changements dans les paramètres du réseau. Nos résultats montrent que VAQ est très stable tandis que ACQ se trouve un peu influencé par certains changements tels que le nombre de liens asymétriques par lequel le trafic passe.

Mots-clés : TCP, asymétrie de bande passante passante, trafic bidirectionnel, utilisation du lien, ordonnancemnet de paquets, satisfaction de l'utilisateur, fonction d'utilité

Contents

List of Figures

1 Introduction

The huge success of the Internet and its transformation into an important commercial infrastructure is in fact a double edge weapon. In fact, new applications have appeared, they are accessible to everyone and it is now necessary for Internet designers to consider new consumer expectations in term of performance, services and bandwidth demand [7, 2]. Satellite networks are on rise and it is usual nowadays the case of a user who downloads data via a high speed link (e.g. downloading a web page via satellite link) while uploading data to the Internet via a low speed link (e.g. sending a mail via modem dial-up line). Asymmetric Digital Subscriber Line (ADSL [14]) is another new technology promising significant bandwidth increases over download paths of existing Internet traffic. Unfortunately TCP, the Transport Control Protocol of the Internet [3] is not prepared yet to face such cases [6]. TCP relies on the Congestion control and avoidance mechanism to regulate the rate of Internet traffic and avoid collapses. The problem is that this mechanism itself relies on the feedback of acknowledgments. Consequently, in case of bandwidth asymmetric links when the reverse bandwidth is considerably less important than forward bandwidth, acknowledgments may be delayed or lost and TCP is then faced to catastrophical situations. Moreover, the presence of bidirectional traffic implies data packets and acknowlegements sharing the same resources. These two kinds of packets are so different in size and in reaction to congestion notifications. that's why the interaction between the two kind of packets is impredictable for TCP. All this leads to bad behavior of the most important protocol of the Internet.
In this context, we present in this report two mechanisms for handling two way traffic over bandwidth asymmetric link. They are called ACQ for Adaptive Class-based Queuing and VAQ for Virtual-Ack-based Queuing. Both run at the entry on the asymmetric link and rely on two classes one class for Data packet traffic and one for Ack packet traffic. The diffrence is that classes are real for ACQ and virtual for VAQ. ACQ proposes to adapt the weights of its classes according to the crossing traffic whereas VAQ grants a credit for the virtual Data packets queue and proposes to adapt this credit according to the crossing traffic.

In the next section, we describe more in details the motivations of our study and the problematic that we propose to solve. In section 3 and 4, we introduce respectively the ACQ model and parameters and the VAQ model. We also report some simulations results concerning the bandwidth share and the average utility function reached with the use of ACQ and VAQ. In section 5, we see the impact of some changes in the network settings on the behavior ACQ and VAQ and finish by giving an efficient comparison of ACQ and VAQ, showing the advantages and inconvenients of every mechainism.

2 Motivations and Related work

When faced to bandwidth asymmetry, TCP encounters some difficulties to control its congestion and to recover from losses. Moreover, the adding of two-way traffic exacerbates these problems since Data and Acknowledgments (Acks) packets share the same resources while being very different in size and reaction to congestion notification.

We describe all this in this section, showing then the motivations of our study. We also report some of the most significant propositions published in order to alleviate the problem of bandwidth asymmetry and bidirectional traffic in the Internet.

2.1 Bandwidth asymmetry

In TCP, Acks serve among other as a reliable clock for packet transmissions: upon every Ack arrival at the source, there is one or more packet transmissions into the network. This clocking is based on the idea that, when arriving at the source, Ack packets are separated in time by the transmission time of a data packet at the bottleneck router on the Data path.
When the bandwidth available in one side of a link is considerably different than the one available in the reverse side of the link, it results in congestion on the Ack path, yielding delays and losses of Acks. This results in a bad behavior of TCP, based essentially on the Ack-Clocked property. The problem is called *Ack-Compression* [15] and results among other in the following consequences:

- Slowness of the congestion window increase: this constitutes the most important consequence of the Ack-Compression problem. In fact since Acks are considerably delayed or even lost, the TCP source is unable to increase efficiently its congestion window and the connection rate remains low.
- Difficulties to recover from losses: TCP recover from losses thanks to its congestion control and avoidance mechanism. Algorithms of this mechanism rely on the flow of Ack packets. A delay or loss of Acks have then direct impact on the behavior of the mechanism.
- Increase in delays and further burstiness in the TCP traffic.

We can then conclude that an asymmetry in the bandwidth available in forward and reverse direction of a link have very bad effects on the behavior of TCP protocol. It is then important to alleviate the problems caused by such environments.

2.2 Effects of two-way traffic

The problem of Ack compression is exacerbated in the case of two-way traffic. Consider a user that downloads data from the Internet via a high speed link and that uploads data to the Internet via a low speed link. The objective of the user is to maximize its satisfaction. The problem is less complex when both traffics are independent of each other. On asymmetric link the problem is challenging since traffics are dependent on each other. The dependence comes from the flow of Acks that share the reverse path with the uploaded data, when the downloaded data is carried by TCP. The available bandwidth on the slow reverse link has to be shared by Acks and data. The simultaneous presence of Acks and data is known to cause several problems:

The major problem is that Acks are not responsive to congestion, so they will finish by monopolizing the available bandwidth and the rate of uploaded data drops to zero. This may not correspond to the optimal allocation of the scarce resource on the reverse path. Moreover Acks and Data packets are really different in size: Data packets are voluminous and when small Acks have to wait behind Data, Acks risk to wait for long time and to be bunched together and transmistted into bursts. This leads to large bursts of Data packets in the opposite direction.

We are then faced to a dead end: giving priority to Ack packets that are unresponsive to congestion notification or giving priority to Data packets that are considerably voluminous.

2.3 Related work

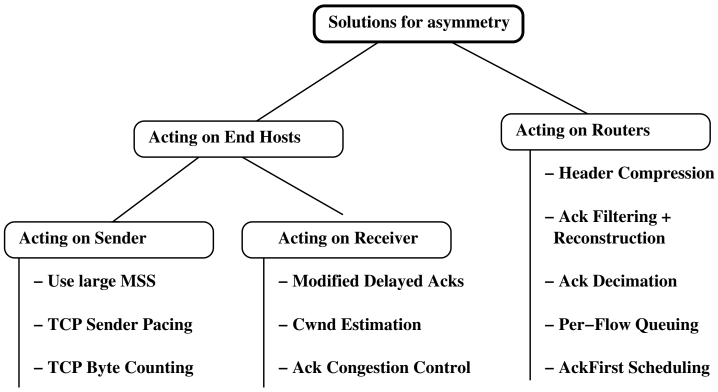

Figure 1: Some solutions for asymmetry and two-way traffic

We report most solutions in Figure 1 and for more details on each solution we refer to [15]. Some solutions that require a change in the TCP stack have been proposed, such as Sender Adaptation or Ack Congestion Control[10, 11]. Balakrishnan and al. [1, 11, 10], Kalampoukas [8] and the PILC group [15] show that performance can be substantially increased by making two operations on Ack flow: simply suppressing Acks on reverse channel (*Ack Filtering* (AF) [8, 5]) and regenerating them after the reverse link has been crossed (*Ack-Reconstruction* (AR) [4]). We retain these two last mechanisms since they do not change the TCP stack.
To alleviate the problem of bidirectional traffic over asymmetric links, RFC 3449 recommends the use of AF/AR combined with an AckFirst scheduling in the router. In [12] Shekar and al. show that if Acks are given priority, the rate of packets on the reverse low channel drops to unacceptable values since the channel will be monopolized by Acks. We agree with their study and we will not apply AckFirst scheduling in our work.

$\Longrightarrow$ Motivations

In this section we saw the impact of bandwidth asymmetry combined with two-way traffic on TCP protocol. We saw that such environnements leads to the Ack-compression problem and all its bad consequences. Moreover we saw that the presence of two-way traffic makes the situation even more difficult and problems more challenging.
We propose in the continuation two packets schedulers which address these problems.

3 ACQ: Adaptive Class-based Queuing

We present in this section our first coarse-grained scheduler for two-way traffic in bandwidth asymmetric networks. The aim is to efficiently schedule between Ack and Data packets so that we obtain an optimal utilisation of the asymmetric link. We present ACQ, Adaptive Class-based Queuing mechanism.

3.1 The model

Let x_f be the rate of downloaded data, and x_r the rate of uploaded data. Denote by $U_f(x_f)$ the satisfaction of the user from the downloaded data, and $U_r(x_r)$ his satisfaction from the uploaded data. The problem is then to find x_f and x_r that maximize the function $U_f(x_f)$ + $U_r(x_r)$.

The two flows of Acks and data in the upload direction have to be separated, by some kind of two-class CBQ buffer. The question that one asks here is how to choose the rates of the CBQ buffer. Giving advantage to the data flow will penalize the flow of Acks, which may penalize the flow of data in the forward direction and conversely giving advantage to Acks will penalize the uploaded data flow. The allocation of rates to Acks and data has to be intelligently done, so that the satisfaction of the user is maximized. We propose here a simple adaptive algorithm to adapt the rates of the two-class CBQ buffer at the input of the reverse link. We consider that Acks and data packets are queued in separate buffers, and are served in a weighted round-robin way.

We consider that the CBQ buffer is able to measure the rate of data in both directions. We can measure the rate of data in the forward direction by using the information carried by Acks, or by some signaling between the host and the CBQ buffer at the input of the slow link. Every time T, our algorithm measures the flow rate of data in both directions, and adapts the rate allocated to the uploaded data flow.

Let r_n be the rate allocated to the data flow at time nT. This rate remains allocated until time (n + 1)T. Let x_r(n) (resp. x_f(n)) be the measured rate of the uploaded data (resp. downloaded data) between nT and (n + 1)T. Finally, let $U(x_r,x_f)$ be the total satisfaction of the user. We use the gradient projection method to update the value of r_n ,

$$r_{n+1} = x_r(\mathrm{n}) + \gamma \frac{dU(x_r(n),x_f(n))}{dx_r}$$

This method assumes the existence of one local maximum, in the definition region of x_r and x_f, which we assume true. γ is a constant that trade offs stability and convergence rate. The experiments will lead us to the optimal value of γ.

We write,

$$r_{n+1}=x_r(\text{n})+\gamma\ \frac{dU(x_r(n))}{dx_r}+\ \gamma\frac{dU_f(x_f(n))}{dx_f}\ \frac{dx_f(n)}{dxr}$$

Since we don't know the explicit relation between x_r and x_f,, we make the following approximation

$$\frac{dx_f(n)}{dxr}\ =\ \frac{x_f(n)-x_f(n-1)}{x_r(n)-x_n(n-1)}$$

We obtain the following rule for setting the rate allocated to the uploaded data flow,

$$r_{n+1}=x_r(\text{n})+\gamma\ \frac{dU(x_r(n))}{dx_r}+\ \gamma\frac{dU_f(x_f(n))}{dx_f}\ \frac{x_f(n)-x_f(n-1)}{x_r(n)-x_n(n-1)}$$

Utility functions The utility function depends on the application type. It may also depend on the cost of bit/s in each direction. We consider here a particular function, which is equal to the bandwidth utilization. The user wants to maximize the sum of utilization in both directions. Let C_f be the available bandwidth in the downstream direction, and C_r the available bandwidth in the reverse direction. Hence, $U_r(x_r)= \frac{x_r}{C_r}$ and $U_f(x_f)= \frac{x_f}{C_f}$. The rule of updating CBQ rates becomes,

$$r_{n+1}=x_r(\text{n})+\gamma(\frac{1}{C_r}+\frac{1}{Cf}\frac{x_f(n)-x_f(n-1)}{x_r(n)-x_r(n-1)}) \tag{1}$$

3.2 ACQ: Architecture and Principles

In order to achieve the user satisfaction and maximize the utility function described in last section, we propose to use a CBQ-like scheduler with two queues, one for Acks packets and one for Data Packets. Both have the same priority and they are served according to the Weighted Round Robin algorithm. Each queue is allocated a fraction of the bandwidth C_r. As represented in figure 1, ACQ runs at the entry of the slow link and proposes to adapt the weights of both classes according to the crossing traffic.

ACQ parameters: T and γ

T represents the bandwidth allocation refreshment interval, its value is then a trade-off between stability and responsiveness of the system. T must be long enough to permit to the traffic to react to a change in the bandwidth allocation. At the same time T cannot be very long since this alters the stability of the system and slows its reaction to any change in the traffic conditions. Since TCP adapts its window size every two RTTs, the minimum value

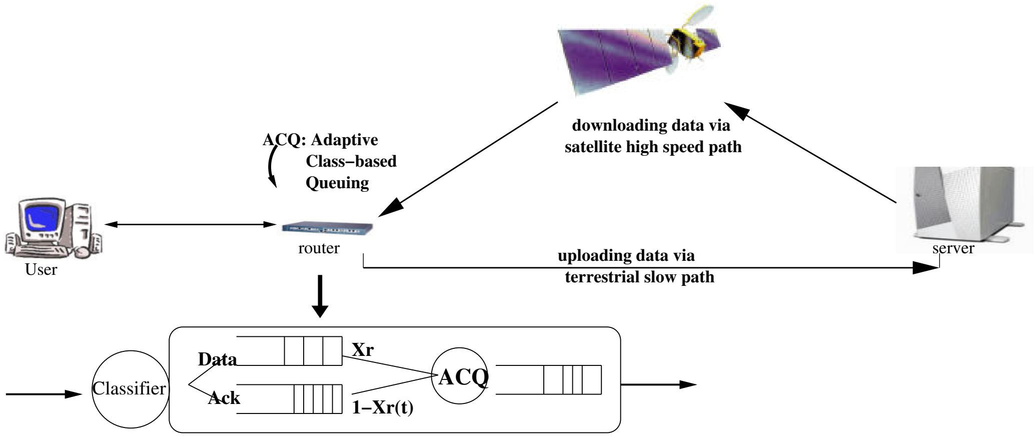

Figure 2: ACQ architecture

for T is obviously twice the bigger RTT of the involved connections.
Concerning γ, this factor decides on the amount by which the rate of CBQ are updated every T. Giving a big value for gamma will rapidly lead us to unstable state, and a too small value of Gamma will necessitate very long time to converge. γ is a variable that must help the system to avoid big oscillation and to converge to the stable state as quick as possible. The choice of γ involves then a clear trade-off. The unit of γ is $Kbps^2$.
In the section 3.1, we report the variation of the utility function in function of values of γ and T, and we give a strategy of choosing these two parameters according to the traffic settings.

3.3 Simulations and results

Simulations presented in this paper have been performed with ns-2[9], figure 3. N TCP flows start simultaneously on each side of the asymmetric link. All flows are FTP-like flows, using TCP Reno as transport protocol. Packet size are 1500 bytes for Data packets and 40 for Acks. All routers have FIFO queue management and buffers of 40 packets. In the reminder of this paper, we will call traffic T_f the forward traffic and T_r the reverse traffic.

3.3.1 Values of T and γ

As said in 3.1, it is important to fix the pair of T and γ that enables ACQ to gives the asymmetric link its maximum utilization. In figure 4, We have plotted the variations of Utility function with different values of T and γ. Simulations were done with N=10 and results are the average of 10 different simulations. We note that for every pair of T and γ, it exists a maximum value for the utility function and the more T is high the less is the value of γ . Figure4 shows three pairs : T=20seconds/ γ =50, T=10seconds/ γ =100 and

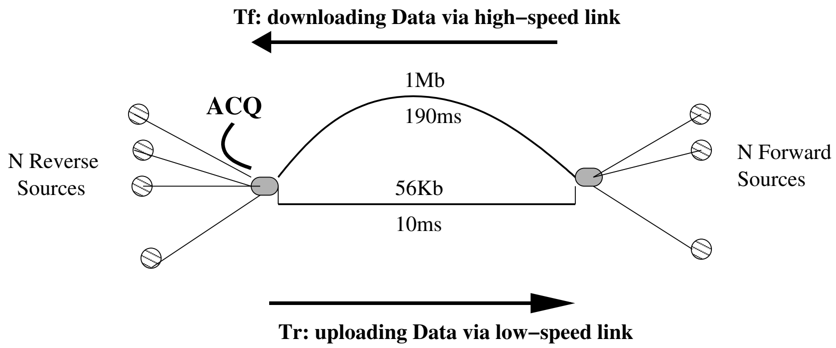

Figure 3: Topology of the simulations

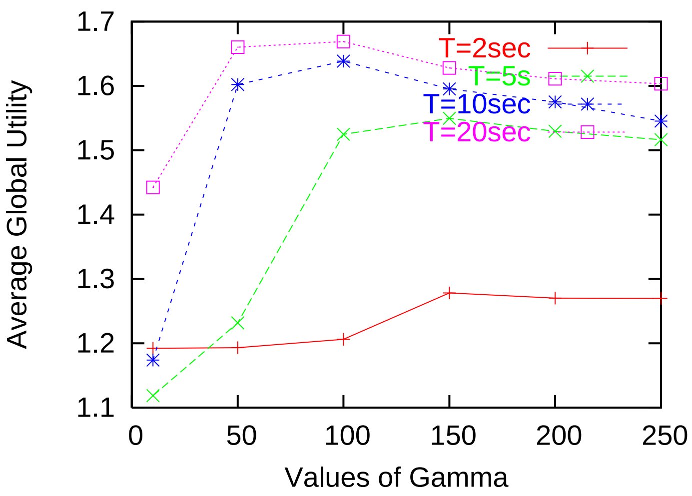

Figure 4: Values of γ and T

T=5seconds/ γ =150. In fact giving a value of T, γ must adapt the system to be reactive to a change in bandwidth allocation, so if we have a small value of T, γ must be big. If in the contrary the value of T is sufficient to make the system converge, γ must not be too big in order to not oscillating.

We have also done simulations involving other topologies and scenarios, and we report the variation of the utility in function of the values of T and γ.

In figure 5(a) and (b) we change the number of connections in each direction, we dealt with the minimum, which is 1 connection on each side, and an important number which is

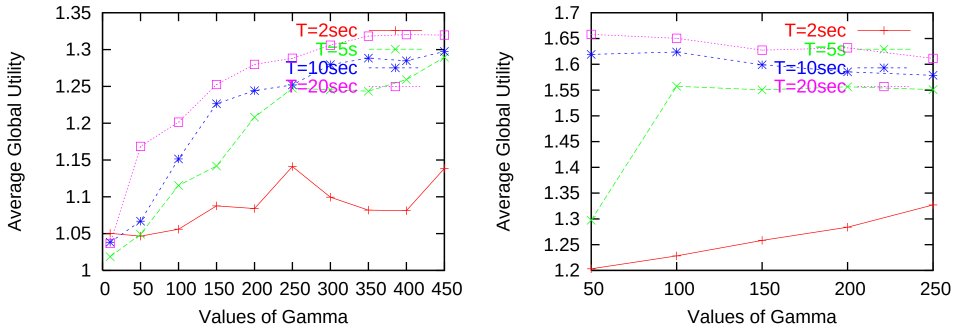

Figure 5: Variation of the Utility Function with the number of connections (1 and 20)

20 connections. For figure5(a), we have one connection in each side, the load is then very light and T must be big enough (around 20seconds) so that the system can react. Once we choose big values of T, we must also fix γat a big value because the system is still not able to reach the maximum utility function.

For 20 connections in each side, we have enough load in the path, and for T $>=$ 5seconds we have good results, even for small values γ. γ in this case will be the degree of sensitivity of the system. So we can affirm that T and γ have different values according to the topology and to the network load:

- If the network is underloaded both the refreshment interval T and γ must be important so that sources can react,
- Otherwise if T is big, γ must be small to avoid important oscillations and if T is small γ must be big to converge rapidly to the optimal utility function.

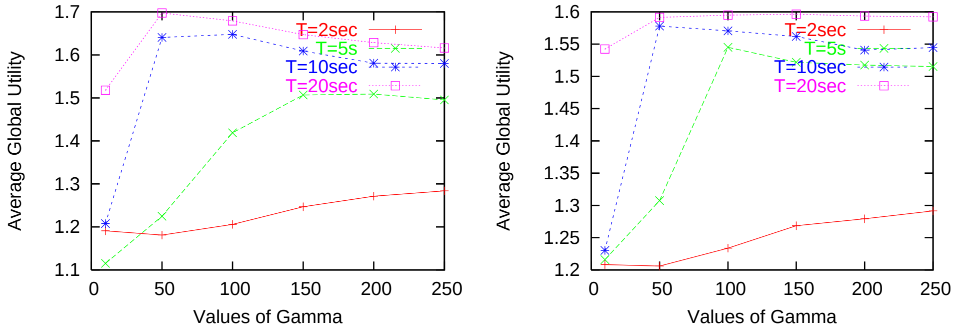

Figure 6: Variation of the Utility Function with the RTT (40 and 70 ms)

We have also plotted the evolution of T and γ with the value of RTT, results are reported in figure 6 (a) and (b). Here again, plots are the same. We can affirm that ACQ is independent of the RTT of connections. It is another advantage for deployment issues.

We do not claim that there are optimal values of T and γ, but for every environment it is possible to have the ideal pair of T and γ, and this pair is between 2 seconds and 20 for T and 50 and 150 for γ. In the rest of this paper we used 10 seconds for T and 50 for γ in order to have like an average of these values.

3.3.2 Bandwidth sharing

We report in the next figures the characteristics of the behavior of ACQ regarding the bandwidth sharing between the two classes and the utility that is reached using ACQ. We have 10 connections in each direction that start all randomly between 0ms and 5ms and stop at t=10000sec. We chose an important duration of the simulations to let ACQ the time to adapt to the best utility function, but we notice that it converge quite rapidly.

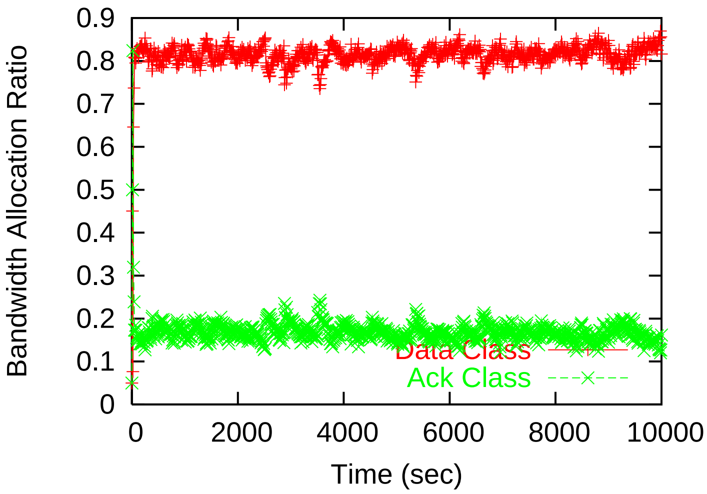

Figure 7: Bandwidth share

In the figure 7 we plot the variation of Data and Ack queue weights in function of the simulation time. As we can see that Data class obtain 80% of the available bandwidth, leaving the other 20% to the Ack class. We also remark that ACQ converges rapidly to these two allocation ratios.

The use of ACQ as a scheduling scheme gives an average utility function around 1.6. We plot in figure 8 the variation of the average utility as a function of the time for 10 different

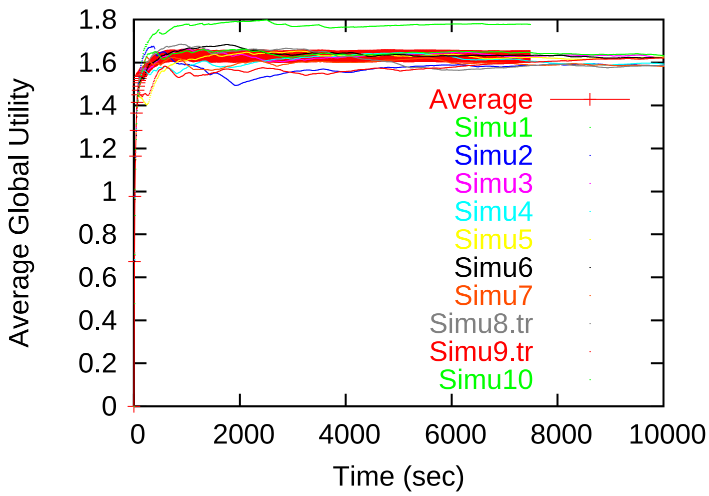

Figure 8: Utility Function

simulations. All of them give strongly near results. We have then a very small confidence interval, in fact there is 95% of chance that the average utilization is in the confidence interval of [1.621481 - 0.005186, 1.621481 + 0.005186] which means that the interval of confidence is in 95% between [1.616195 , 1.626667]. This allows us to affirm that results of our simulations are reliable and that the use of ACQ gives good performance.

3.3.3 Utilisation of the link

We have compared the average utilization of the link computed in three scheduling schemes:

- CBQ scheduling, with and without AF/AR in the Ack queue,
- ACQ scheduling with and without AF/AR in the Ack queue and
- simple Drop-Tail (FIFO) scheduling with AF/AR mechanism in the shared buffer.

Again we have N=10 connections in each side and we choose the values of T=10 and γ=50 as explained earlier. In figure 9 we reports our simulations results. ACQ manage to reach 1.62 as average utility function outpassing the 1.4 giving by CBQ. Certainly the difference is not so important but this means that ACQ has found the bandwidth share between the two classes that gives more satisfaction for the user. In the rest of this paper we have compared with the simple CBQ that gives 50 per cent of the bandwidth equally to Ack and Data classes.

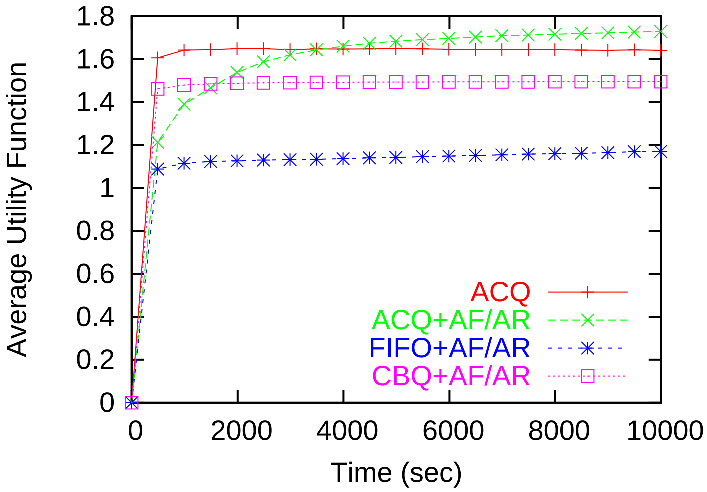

Figure 9: Comparison of link utilisation

In figure 9 we add AF/AR in the Ack queue and we also compare with FIFO. We can see that applying ACQ with the AF/AR mechanism in the Ack queue allows to reach a quasi maximum utility function i.e 1.72 compared to 1.5 for CBQ with AF/AR and 1.28 for FIFO with AF/AR. We also note that the addition of AF/AR mechanism in ACQ has slowed the speed convergence to the optimal utility function of ACQ, this is due to the time spent in filtering and especially reconstructing Acks.

Simulations done in this section show explicitly that with or without the use of AF/AR in the Ack queue, ACQ manage to reach the optimal utility function. Adding AF/AR will increase the final value of this utility function but will necessitate longer time to reach it.

3.4 Conclusions about ACQ

In this section we presented ACQ: a mechanism to handle two-way traffic over bandwidth asymmetric link. We have done simulations that show the increase of performances reached by the use of ACQ. ACQ necessitates the setting of two parameters, and we have have explained how to choose these parameters according to the traffic on the network.

We can easily imagine the gain that ACQ will allow in term of link utilisation. This gain is especially important for operators who will have the opportunity to improve the utilisation of their resources. In fact, as explained the good performances of ACQ depend strongly on the parameters settings. Operators know exactly the traffic on their networks, they are then able to optimally set ACQ parameters according to the predictable traffic and get better utilization of their resources.

On the other hand, the interest of ACQ is also to be able to answer exactly to the user's expectation. In the ACQ model, our Utility Function is the utilisation of the asymmetric link. It is possible however to have other Utility Functions, according to the user's demands; for example the user may want to give priority to forward or reverse traffic, he may require more fairness, and so on. Consequently, we can at each time, model the ACQ scheduler according to the user satisfaction changing only the Utility Function.

4 VAQ: Virtual Ack-based Queuing

In the last section, we described the ACQ scheduler. However setting the ACQ parameters can be considered a drawback, since in some cases, network settings may change often, and it becomes complicated to always fix the best values of T and γ.
In this section we propose a fine-grained mechanism to optimise the utilization of available resources. We present VAQ for Virtual Ack-based Queuing mechanism. VAQ does not necessitate the setting of any parameters and manages however to reach good utilization of the asymmetric links.

4.1 The VAQ Model

VAQ runs in the entry of the asymmetric link and uses two virtual queues: one for Data packets and one for Ack packets. It grants a credit to the Data packets and proposes to adapt this credit according to the crossing traffic in order to reach an optimal utilization of the asymmetric link. VAQ obeys the following rule:

The Data queue is allowed to send packets only when it has sufficient credit.

4.1.1 The VAQ scheduling policy

We keep the same notations as in the last section: T_f for the forward traffic, T_r for the reverse traffic and C_f and C_r for the available bandwidth respectively in the forward and reverse path.We add these following notations:

- QueueDatas: virtual queue associated to the Data packets
- QueueAcks: virtual queue associated to the Ack packets
- NbrBytesAcked(p): total number of Bytes acked by a given Ack packet p
- Size(p): size in bytes of one given Data packet p

The utilization of the asymmetric link is represented by the sum of Data packets belonging to the reverse traffic that cross the reverse link plus the Data packets belonging to the forward traffic that cross the forward link. When an Ack packet (belonging to the forward traffic) is transmitted through the reverse link, that means that in the earlier RTT, the forward link has been used by one or more Data packets of the forward traffic. So, in order

to maximise the global utilization of the link, it is now necessary to obtain a good utilization of the reverse link and to accord bandwidth to the reverse traffic.
When an Ack packet is transmitted through the reverse link, VAQ looks into the total number of bytes that this Ack acknowledges and accords an "equivalent" credit to the Data packets of the reverse traffic. The ratio of the total number of bytes acked by this Ack and the total capacity of the forward link represents the utilization of the forward link. So we have:

$$\text{Forward utilization} = \frac{NbrBytesAcked}{C_f}$$

This gain in forward utilization implies a loss in the reverse utilization since Acks and Data share the slow link. To compensate this loss, VAQ must allow to the reverse traffic to reach a reverse utilization equal to:

$$\frac{NbrBytesAcked * C_r}{C_f}$$

This is exactly the value by which the credit accorded to QueueData will be incremented for any passage of an Ack packet.
On another hand, on the passage of one Data packet in the reverse link, VAQ looks to the value of the credit. If the credit allow it, QueueData is able to send one Data pasket. VAQ considers then that QueueData has used a part of its allowed credit and decreases it by Size(p). The credit will then be updated by:

$$\text{credit} = \text{credit} - \frac{Size(p)}{C_r}$$

4.1.2 VAQ parameters

VAQ is a scheduler that sets two virtual queues for the Data and Acks packets, grants a credit for the Data queue and adapts this credit according to the crossing traffic in order to obtain the optimal utilization of the asymmetric link.
The only parameters that must be given to VAQ are the sizes of both virtual queues. Intuitively the queues must be given a sufficient size to store at the minimum one packet from each active connection, the minimum size is then the number of active connections. In fact the maximum sizes have bigger importance on the performances of VAQ, and more specifically the Ack queue maximum size:
Data packets are responsive to congestion notification, so even if they experience important queuing delays, TCP is able to regulate itself and to alleviate bad effects of such delays. However Acks are not responsive to any congestion notification, so if they are stored in a too long queue, some of them will be considerably delayed. This will disturb the behavior of TCP and consequently will decrease the utilization of the link. That's why it is advised to have small virtual Ack queues.

The size of Data and Ack packets is another factor that may have influence on the behavior of VAQ. In fact, the credit allocated to the Data queue is decremented or incremented proportionnaly to the Data packet size. Simulations we have done show that the size of Ack and Data packet has no significant importance on the performance of VAQ see figure, the best pair is: 40 bytes for the ack packet and 1500 bytes for Data.

4.2 VAQ simulations results

In this section we report the results of simulations done on the same topology of the figure 3. According to the last section, we choose 10 and 100 respectively for the AckQueue and DataQueue size, and 40 and 1500 for the packet sizes. In order to let the system enough time to react and stabilise itself, simulations last 10000 seconds.
Our objective is always to satisfy the user, so we consider as our Utility Function the utilization of the asymmetric link. In these simulations, we want to know the sharing of the reverse available bandwidth between Data and Ack packets and of course the variation of the average Utility Function with time.

4.2.1 Bandwidth sharing

We plot the distribution of the bandwidth between the two traffics. This will allow us to have an idea of the distribution of the available reverse bandwidth between Data and Ack packets when we use VAQ.

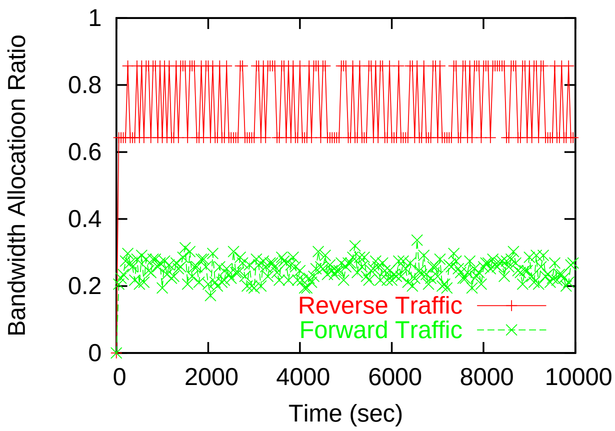

Figure 10: Bandwidth utilization with VAQ

As the Figure 10 shows, after some seconds, VAQ manage to find the best sharing of the bandwidth between the reverse and forward traffic. It gives around 80 per cent of the reverse to Data belonging to the reverse traffic and the rest is for Acks of the forward traffic.

4.2.2 Utilisation of the link

We take again the Figure 9 and add the variation of the utility function with VAQ as a scheduler at the entry of the asymmetric link. VAQ will then decide which of the Data or Ack packets will have access to the asymmetric link. VAQ gives the optimal value for the Utility Function: results are on Figure ??.

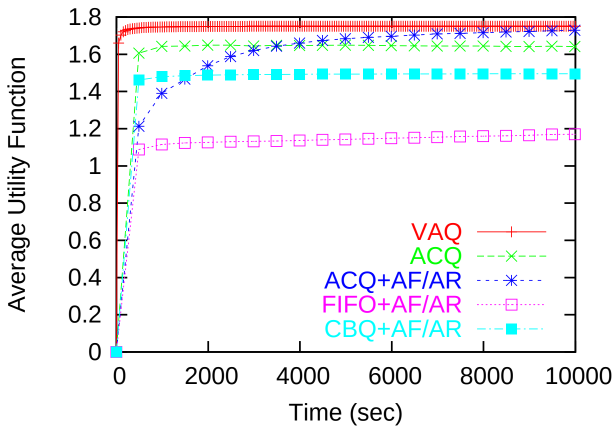

Figure 11: Comparison of link utilisation

With the use of VAQ, we have a global utility function around 1.75 and the system reaches this values in a very short period of time. So VAQ outpasses all the other schedulers even if it does not involve any mechanism of handling the Ack Compression problem caused by the asymmetry of the link. By just adapting the credit of the Data packets queue, VAQ manage to reach rapidly the best satisfaction of the user.

Like with ACQ in section 3, this bandwidth sharing gives the best global utilization results. The difference with VAQ is that there is no need to setting T and γ parameters, and that VAQ is able to reach around 1.75 as utility function comparing to 1.62 to ACQ.

4.3 Conclusions about VAQ

VAQ is a packet scheduler which maximises the utilisation of an asymmetric link when it is crossed by two-way traffic. VAQ runs at the entry of the shared link, affects a credit to Data packets and adapts the credit according to crossing Ack and Data traffic. VAQ is then an implementation of the Weighted Fair Queuing scheduling mechanism with a Leaky Bucket policing.

Our simulations show clearly that VAQ manages to reach an optimal utilisation of the asymmetric link and with an efficient sharing of the bandwidth, allows the user satisfaction.

It is then really recommended to use VAQ on the entry of every link where Ack and Data packets may share the available bandwidth.

We have seen that VAQ gives better results than ACQ. The problem now is to verify that VAQ and ACQ are stable to some changes in the network settings. In the next section, we test the robustness of VAQ and ACQ.

5 Robustness of ACQ and VAQ with changes in some network settings

In the last sections, we have done simulations which are certainly realistic but static and obeying to fixes network settings. In this section we change some of the network settings and see the behavior of ACQ and VAQ. We wonder whether ACQ and VAQ are able to adapt and in how long. We end by reviewing quickly ACQ and VAQ and give a comparison of these two new schedulers for handling two-way traffic over asymmetric networks. We answer the question of for whom such schedulers are addressed and how complex or simple it may be to deploy these mechanisms.

5.1 Changing the number of connexions involved

We begin by changing the number of connexions involved in the simulations, in each side of the asymmetric link. However, we always keep the same number on connections for the forward and the reverse traffic in order to be the most fair as possible. In fact, with a higher number of connexions belonging to one traffic or the other we may have some particular behaviors of ACQ and VAQ.

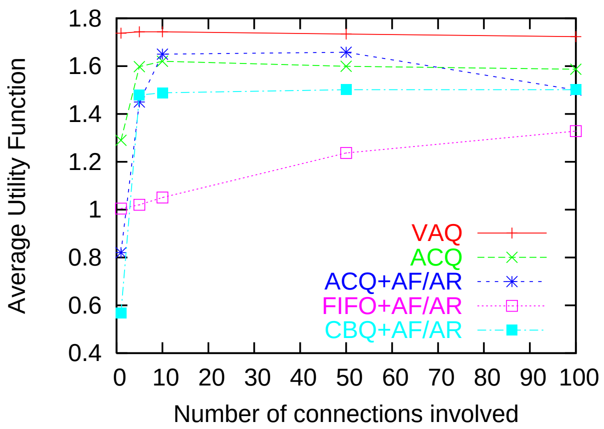

Figure 12: Influence of the number of connexions

We make simulations reported in Figure 12 with minimum number of connections involved (one for T_f and one for T_r), small number of connections involved (5 and 10 respectively for T_f and 5 and 10 for T_r) and large number of connections involved: 100 for T_f and 100 for T_r.

Concerning ACQ, since we have seen that the use or not use of AF/AR in the Ack Class may have impact on the behavior of ACQ, we also want to know the impact of the number of connexions on ACQ with AF/AR and without AF/AR. At each simulation scenario, we set the parameters T and γ according to the strategy of the section 3.3.1. For example, we set T to 20 seconds and γ to 400 $kbps^2$ for the case of 1 connexion for each traffic.

We see clearly that the number of connexions involved does not have any impact on the good behavior of VAQ. In fact, the charge of the network does not alter the ability of VAQ to adapt and to give the optimal utility function. However for ACQ, impact of the number of connexions involved depends on whether we use AF/AR or not in the Ack queue. Figure 12 shows that, if we use AF/AR, an increase in the number of connections implies a decrease in the average utility function. This is explained by the fact that the more we have connexions, the more delays added by mechanism of filtering and reconstructing Acks will be important. And an increase in the delay lead to a rate decrease. However for just ACQ, we have acceptable robustness of ACQ faced to a change in the number of connexions. That's why, for ACQ, it is better not to use AF/AR in the case of important number of connexions involved.

5.2 Changing the asymmetry degree

The asymmetry degree may also have impact on the good or bad behavior of ACQ and VAQ. In fact, when the difference between available bandwidth in forward and in the reverse direction become really important, the effect of two-way traffic is exacerbated and ACQ and VAQ may not be able to adapt and give the best utilisation of the link.

We here do some simulations changing the value of the capacity of the reverse asymmetric link, going from a case of very important asymmetry to the symmetric case.

Results are on figure **??**. We see that the more the degree of asymmetry increases (i.e. the more the capacity of the reverse link decreases) the more the utility function achieved by ACQ decreases. And this valid for the use of AF/AR and for the non use of AF/AR. One remark however on the variation of the utility function for the symmetric case and with AF/AR. We see that in this case applying the Ack Filtering and then the Ack Reconstruction at only one side of the symmetric case will decrease the performances. This is quite normal since AF/AR must only be deployed in an asymmetric environment, otherwise it must not be deployed only in one side of a symmetric link.

Concerning VAQ, we see that there is a small change in the global utility function when the degree changes, especially we reach the maximum utility function when the link is symmetric. Globally, results of this simulations were really satisfying since VAQ give an optimal Utility Function even in the case of important asymmetry.

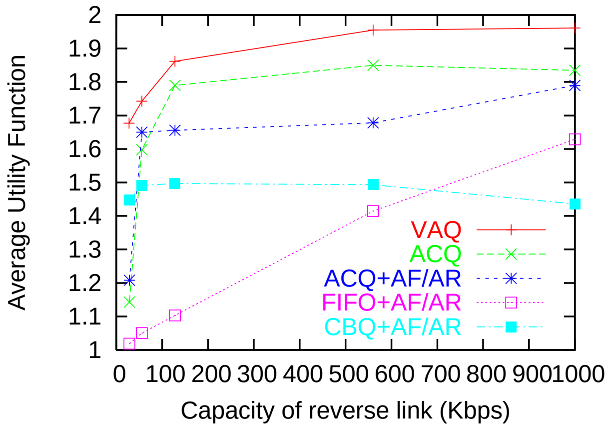

Figure 13: Influence of the asymmetry degree

5.3 Passing through more than one asymmetric link

In the real word, a traffic may pass through one or more asymmetric link. The question is then whether ACQ and VAQ are able to adapt two-way crossing traffic even in the case of passing through more than one asymmetric link. That's the subject of simulations done in this section.

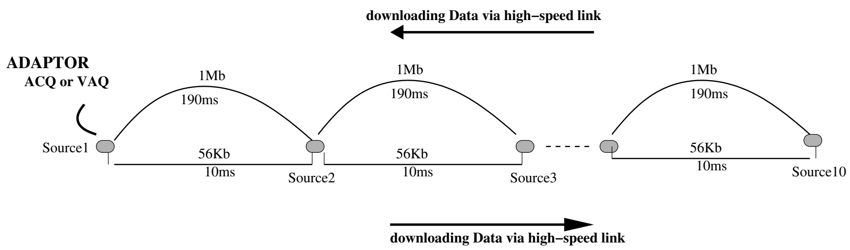

Figure 14: Topologie of the simulations

Topologies of the simulation is reported in Figure 14. We make simulations with traffics passing through one, two, three, five and ten asymmetric links. At each edge router in the entry of one asymmetric link we apply ACQ or VAQ, according to the case. And we plot the variation of the average Utility Function with the time.

Results of the simulations are on Figure 15. We see that ACQ is influenced by the number of asymmetric links, and if traffic pass through more than one asymmetric link, ACQ become unable to good value of the average Utility Function.

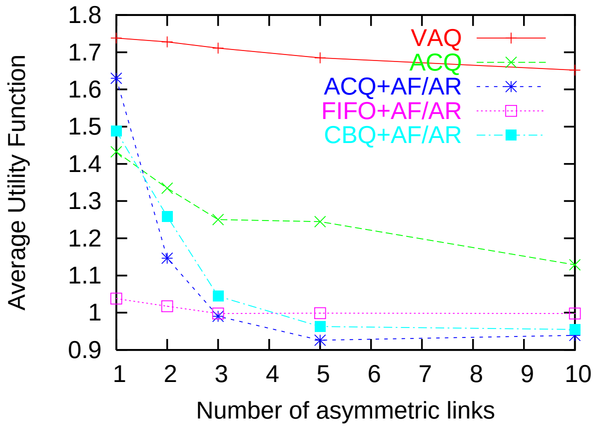

Figure 15: Influence of the number of asymmetric links

However VAQ is really stable and even for 10 asymmetric links, the Utility Function reached is optimal. The VAQ mechanism is then stable when passing through more than one asymmetric link. Its deplyement is then really possible.

5.4 Robustness of ACQ and VAQ

In this section, we did simulations that involve both ACQ and VAQ schedulers in diffrenet network settings, we saw the variation of the average utilization of the asymmetric link shared between Data and Acks packets.
We began by varying the number of connexions of forward and reverse traffic and see the impact on the behavior of ACQ and VAQ. Results showed that both schedulers are robust to such change in the topology. However if ACQ is applied with AF/AR mechanisms for facing asymmetry, the more we increase the number of connexions, the more we decrease the Utility Function reached.
We then varied the asymmetry degree of the shared link. Obviously if we have symmetric link, the average utilisation is maximum, especially for VAQ where we reach 1.95. However even in the case of strong asymmetry, VAQ is stable. For ACQ, a strong symmetry gives poor average Utility Function.
Finally we change the number of asymmetric link through which passes the forward and reverse traffic. Here again VAQ gives excellent results concerning the first link utilisation. In fact VAQ is not influenced by the number of asymmetric links. However, ACQ is really sensible to a change in the asymmetric links, and the more we have asymmetric links, the more we decrease the value of the Utility Function.

These study gives us an idea of the possible deplyement of ACQ and VAQ schedulers. Obviously, ACQ is not ready yet to be deployed into a real network, where number of asymmetric links is large and where average degree of asymmetry is important. However, in all other networks, it has the advantage to obey exactly to the user expectation, by simply changing the Utility Function. For VAQ, we are very happy of the simulations results. In fact VAQ scheduler has showed a great stability to changes in the network settings. This encourages us to continue studying this promising mechanism for handling two-way traffic over bandwidth asymmetric links.

6 Conclusion

In this report, we examined the performances of bidirectional TCP connections over a bandwidth asymmetric network, and we gave two solutions for alleviating serious problems caused by such situations. In fact, the bandwidth asymmetry has been shown to result in an effect called "Ack-Compression", and in two-way traffic, the TCP segment transmitted by the connection in one direction share the same physical path with the acknowledgment of connections in opposite direction. The "‘Ack-Compression"’ effect is then exacerbated and results in reduced overall throughput.

We first introduced ACQ for Adaptive Class-based Queuing, a mechanism that relies on two different classes, one for Data and one for Ack packets and that adapt the weigths of classes according to the crossing traffic. Simulations showed that ACQ manages to give better results for the global utilisation than with FIFO or CBQ. Associated with AF/AR mechanism for handling asymmetry, it gives more better results, but necessitates more time to converge. However our study showed some limitations of this scheduler in that case of different network settings, like passing through many asymmetric links.

We also introduced VAQ for Virtual-Ack-based Queuing, a mechanism that relies on two virtual queues, one for Data and one for Ack packets, grants a credit to the Data queue and adapt this credit according to the crossing traffic. Simulations results were really encouraging and promising. VAQ gives optimal results for the average utilisation of the link and above all it is stable to many changes in the network settings. We then will immediately test this scheduler in a real testbed and work hard to see it deployed shortly.

References

[1] H. Balakrishnan, V.N. Padmanabhan, and R.H. Katz. “The Effects of Asymmetry on TCP Performance” in Proc. 3rd.

[2] T.V. Lakshman, B. Suter. “TCP/IP Performance with Random Loss and Bidirectional Congestion” , IEEE/ACM transactions on networking, Vol. 8, no 5, Oct. 2000.

[3] V. Jacobson. “Congestion avoidance and control”, ACM SIGCOMM, Aug. 1998.

[4] T.V. Lakshman, U. Madhow, and B. Suter. "Window-based error recovery and flow control with a slow acknowledgment channel: a study of TCP/IP performance", IEEE INFOCOM, Apr. 1997.

[5] C. Barakat, and E. Altman. "On ACK Filtering on a Slow Reverse Channel", proceedings of the first international workshop on Quality of future Internet Services (QofIS), Berlin, Germany, September 2000.

[6] L. Zhang, S. Shenker, and D.D. Clark. "Observations on the dynamics of a congestion control algorithm: The effects of two-way traffic", In Proc. SIGCOMM '91 Symposium on Communications Architectures and Protocols, pages 133–147, Zurich, September 1991.

[7] U. Madhow, "Dynamic congestion control and error recovery over a heterogeneous Internet" (invited paper), IEEE CDC, 1997.

[8] L. Kalampoukas, A. Varma, and K. K. Ramakrishnan. "Improving TCP throughput ovtwo-way asymmetric links: analysis and solutions", In Proc. of Sigmetrics, 1998.

[9] Ns network simulator, available via http://www-nrg.ee.lbl.gov/ns/

[10] H. Balakrishnan PhD Thesis "Challenges to Reliable Data Transport over Heterogeneous Wireless Networks" Aug 1998.

[11] H. Balakrishnan, V. N. Padmanabhan, G.Fairhurst, M. Sooriyabandara "TCP Performance Implications of network Path Asymmetry" IETF RFC 3449 December 2002 .

[12] D. Shekhar, H. Qin, S. Kalyanaraman, K. Kidambi, "Performance Optimization of TCP/IP over Asymmetric Wired and Wireless Links," Invited paper at European Wireless 2002, February 2002.

[13] V. Jacobson. "Compressing TCP/IP Headers for Low-Speed Serial Links", RFC 1144, Feb 1990

[14] S Kalyanaraman, D. Shekhar, K. Kidambi "TCP/IP Performance Optimization over ADSL" GI2000.

[15] PILC: Performance Implications of Link Characteristics Working Group, URL: http://www.ietf.org/html.charters/pilc-charter.html.

[16] S. Kunniyur, R. Srikant "End-to-End Congestion Control Schemes: Utility Functions, Random Losses and ECN Marks" INFOCOM 2000.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399