ROBUST ADAPTIVE CONGESTION CONTROL FOR NEXT GENERATION NETWORKS

Francesco Delli Priscoli, Antonio Pietrabissa

Dipartimento di Informatica e Sistemistica (DIS), University of Rome "La Sapienza"
Via Eudossiana 18, 00184 Rome (Italy); e-mail addresses: {dellipriscoli, pietrabissa}@dis.uniroma1.it

Abstract: This paper deals with the problem of congestion control in a next-generation heterogeneous network scenario. The algorithm runs in the 'edge' routers (the routers collecting the traffic between two different networks) with the aim of avoiding congestion in both the network and the edge routers. The proposed algorithm extends congestion control algorithms based on the Smith's principle: i) the controller, by exploiting on-line estimates via probe packets, adapts to the delay and rate variations; ii) the controller assures robust stability in the presence of time-varying delays. *Copyright © 2005 IFAC*

Keywords: congestion control, robust control, adaptive control, Smith's principle.

1. NEXT-GENERATION NETWORKS[1]

The current frontier in the communication area is represented by the so-called next-generation networks, characterized by a heterogeneous network scenario: the packet flow between a source and a receiver involves several networks, different in terms of topology and access technique. Networks providing the users with the access to network resources are called Access Networks, whereas networks connecting different Access Networks are called Core Networks.

For instance, in Fig. 1, a typical next-generation network scenario is depicted: the source of the packet flow is a Personal Digital Assistant (PDA) located in Access Network A; the destination is a PC located in Access Network B; the packet flow is routed through a Core Network connecting the two Access Networks. Within this scenario, the research is focused on Quality of Service (QoS) provision, i.e., on providing the traffic flows with guarantees in terms of bandwidth, delay and losses constraint.

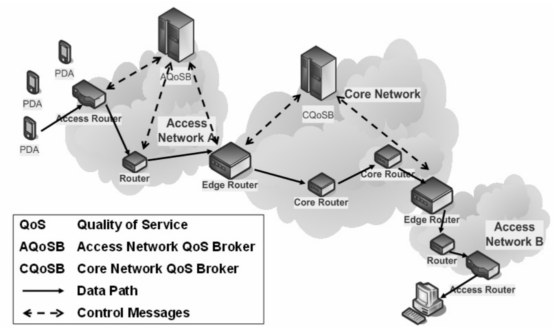

Fig. 1. Next-generation network framework.

For this purposes, new entities have to be defined, named QoS Brokers (QoSB) – both in the Access (AQoSB) and in the Core Networks (CQoSB) – in charge of managing the network resources through admission control, congestion control, policing and traffic shaping procedures.

Two kinds of routers are defined: Network Routers and Edge Routers. Network Routers are placed within a single network, and have the function of correctly forwarding the packet flows from one edge of the network to another one; conversely, Edge Routers connect two different networks and have the function of collecting and regulating the packet flows between the two networks. Thus, Edge Routers are likely to support heavy aggregates of traffic, which, if transmitted without a well-defined policy, might congestion the destination network.

As shown in Fig. 1, the fundamental difference between AQoSB and CQoSB is that the former can directly regulate all the routers of the Access

[1] The work described in this paper is based on results of IST FP6 Integrated Project DAIDALOS. DAIDALOS receives research funding from the European Community's Sixth Framework Programme. Apart from this, the European Commission has no responsibility for the content of this paper. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

Network, whereas, for scalability reason, the latter can only act on the Edge Routers. Thus, to control the traffic entering the Core Network, the CQoSB can only limit the traffic carried by the Edge Routers.

In many papers (Breslau et al., 2000; Choi and Bettati, 2001), this problem has been analyzed from the connection admission control viewpoint: the problem is to check whether a new traffic flow (connection) can be set up, depending on the available network resources and on the declared parameters of the new connection (typically, bandwidth and delay requirements). Thus, at connection setup, the problem is to perform an estimate of the actually available bandwidth. For this purpose, current proposals converge to the definition of special packets, named probes – see (Breslau et al., 2000; Choi and Bettati, 2001) –: probes are transmitted by the source Edge Routers towards the destination ones, and collect some measures on the network status.

However, since admission control procedures are performed at connection setup only, i.e., they are *static*, congestion situations might arise where network resources are over-utilized; in these cases, packets are dropped by the Core Network Routers, which, for scalability reasons, manage aggregate traffic only. The solution is to add a congestion control to dynamically slow down the packet flow during congestions; in this way, packets are accumulated in the source Edge Routers, which can enforce the most appropriate policies on per-flow basis.

If we want to control the network dynamically, one problem is that the probes might significantly increase the network load, even causing congestions (Breslau et al., 2000); this fact limits the amount of information which can be carried by the probes. Another problem related to the heterogeneous network scenario and addressed by the proposed congestion control scheme is that the buffers of the network routers are dimensioned taking into account the bandwidth-delay product of the network they belong to – in fact, it is known that the buffer requirements are tightly related to the bandwidth-delay product –. If the flow path involves different networks, as, for instance, a terrestrial network and a satellite one (which is a high latency network), the buffer dimensioning of the terrestrial network are inadequate for the end-to-end delay. This is a well-known situation arising when TCP connections involve high-latency networks. On the contrary, the edge router between the terrestrial and the satellite networks is aware of the delays involved by both networks, so that it can be correctly dimensioned.

The objective of the present paper is to dynamically control the packet flow entering a certain Core Network, without controlling directly the Core Network Routers and without flooding it with probe packets, by continuously estimating the bandwidth availability, with the aim of limiting the occurrences of overflows of the network buffers.

Section 2 introduces the congestion control problem; Section 3 details the discrete-time network model and the congestion controller; in Section 4 some simulation results are presented; finally, in Section 5 conclusions and on-going research are described.

2 EDGE ROUTER CONGESTION CONTROL.

Edge router congestion control is an emerging topic, since it is tightly related to the heterogeneous network scenario. Congestion control is required when either the Access Network of the destination Edge Router cannot support the transmission rate of the source Edge Router (destination Edge Router congestion), or the Core Network Routers cannot support the transmission rate of the Source Edge Router (Core Network congestion). Congestion control objective is to set the transmission rate of the aggregate flows so that buffer overflows of the destination Edge Router and of the Core Routers are avoided (as far as possible).

In the considered scenario, the CNQoSB has the following functions: i) it can perform delay measures in the Core Network; ii) it communicates with the Edge Routers; iii) it can retrieve information about Edge Router status; iv) it can enforce the transmission rates of the flows aggregate by the Edge Routers. Moreover, thanks to the admission control functionality, in the core network links some capacity is reserved for the aggregate flows of the Edge Routers; this means that these flows are separated from local traffic within the Core Network, and that congestions are due to aggregate traffic fluctuations and to interaction between aggregate flows.

With respect to usual congestion control protocols, the main differences are that the destination and the source Edge Routers exchange control messages via the CQoSB (and not via the network), and that the transmission delay between the Source and the Edge Routers is estimated on-line by the CQoSB. Thus, to take advantage of these characteristics, the congestion controller must be adaptive and robust with respect to variable delays.

There is vast literature dealing with congestion control: for instance, in (Altman, Başar and Srikant, 1999), the congestion control problem is formulated as a stochastic control problem where the controls of different users are subject to different delays; in (Mascolo, 1999), the congestion control law is based on the Smith's principle; in (Quet et al., 2002), an $\mathcal{H}^\infty$ controller is designed guaranteeing stability robustness with respect to uncertain time-varying multiple time-delays; in (Tarbouriech et al., 2001), the congestion control problem is formulated as a robust tracking control problem, in which the target is a constant threshold on the queue length; in (Jagannathan and Talluri, 2002), a neural network-based adaptive control methodology is developed to prevent congestion. We decided to develop a Smith predictor-based controller (see (Palmor, 1996)): in particular, we extended the algorithm in (Mascolo,

1999) to provide robust stability in the presence of time-varying delays, and we used the on-line delay estimates to render the controller adaptive; the reason of this choice is that the resulting control law is simple and easy to implement: this is a crucial requirement in the considered dynamic scenario.

2. NETWORK MODEL AND CONTROLLER.

2.1 Network Model.

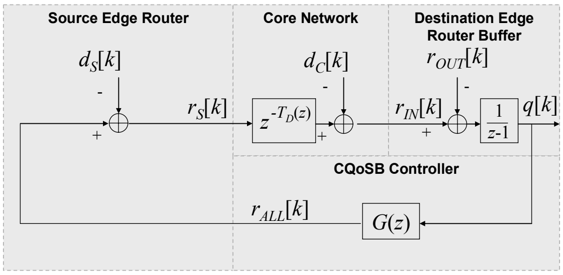

Fig. 2. Network Model.

As shown in Fig. 2, the network is modelled by a linear, discrete-time system with time-varying delay, composed by 4 elements:

1) The destination Edge Router buffer is modelled by a summation:

$$q[k] - q[k{-}1] = r_{IN}[k] - r_{OUT}[k] \qquad (1)$$

where $r_{IN}[k]$ and $r_{OUT}[k]$ are the rates of the packets feeding/departing from the buffer, respectively (due to space reason, saturations are neglected; see (Delli Priscoli and Pietrabissa, 2004) for details).

2) The CQoSB controller, $G[k]$, based on $q[k]$, set the allowed transmission rate, $r_{ALL}[k]$, of the source Edge Router (also in this case, due to space reason, saturations are neglected; see (Delli Priscoli and Pietrabissa, 2004) for further details).

3) The source Edge Router transmits with source rate $r_S[k]$, equal at maximum to $r_{ALL}[k]$; this is modelled by introducing the additive source disturbance, $d_S[k]$:

$$d_S[k] = r_{ALL}[k] - r_S[k] \qquad (2)$$

so that if the source can transmit at $r_{ALL}[k]$, $d_A[k] = 0$, whereas, in the opposite case, $d_A[k] > 0$.

4) The Core Network introduces a time-varying transport delay, $T_D[k]$; furthermore, in case of congestion, some packets are dropped, so that $r_{IN}[k] \leq r_S[k]$; this is modelled by the Core Network disturbance, $d_C[k]$:

$$d_C[k] = r_S[k - T_D[k]] - r_{IN}[k] \qquad (3)$$

so that, in case of congestion, $d_C[k] > 0$, whereas, in the opposite case, $d_C[k] = 0$.

Note that the process $P(z)$ of the model has a time-delay ($P_0(z)$ indicates the delay-free model):

$$P(z) = \frac{1}{z-1} e^{-T_D(z)} = P_0(z) e^{-T_D(z)} \qquad (4)$$

Thanks to the linearity of the model, in the following, $d_C[k]$ and $d_S[k]$ will be modelled by a single additive disturbance, $d[k]$:

$$d[k] = r_{ALL}[k - T_D[k]] - r_{IN}[k] \qquad (5)$$

which, is non-negative.

2.2. Congestion Controller.

$G[k]$ is based on the Smith predictor. Fig. 3 shows the basic controller, derived from (Mascolo, 1999), where K is the proportional primary controller, T_D is constant, $r_{MAX}[k]$ is the maximum transmission rate of the source router avoiding overflows of the destination router buffer. By denoting with S the buffer size, $r_{MAX}[k]$ is temporarily set as:

$$r_{MAX}[k] = S \,/\, (T_D{+}1/K) \qquad (6)$$

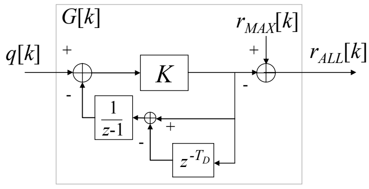

Fig. 3. Non-robust, non-adaptive controller.

The resulting rate command is the discrete-time counterpart of the one of (Mascolo, 1999):

$$r_{ALL}[k] = r_{MAX}[k] - K\left\{ q[k] - r_{MAX}[k]T_D + \sum_{i=1}^{T_D} r_{MAX}[k-i] \right\} \qquad (7)$$

The following theorem holds:

***Theorem* 1:** If $T_D[k]$ is constant and equal to T_D, the transfer function between $r_{MAX}(s)$ and $q(s)$ is equal to a first-order system followed by a pure delay:

$$q(z) / r_{MAX}(z) = \left[K / (z - 1 + K)\right] z^{-T_D} \qquad (8)$$

and $q[k] \leq S \; \forall k$.

***Proof*:** Equivalent to the one in (Mascolo, 1999). □

Source/core bottleneck. The first problem arises since, with the controller of Fig. 3, $G[k]$ is always aimed at transmitting at the maximum rate, $r_{MAX}[k]$; if the source Edge Router or the Core Network cannot transmit at the allowed rate, $r_{ALL}[k]$, the control becomes ineffective: $r_{IN}[k]$ is independent of $r_{ALL}[k]$. The solution is that the CQoSB estimates the rate sustainable by the source and by the network by comparing $r_{IN}[k]$ and $r_{ALL}[k]$. Periodically, the CQoSB computes the measure of the sustainable transmission rate, $r_{SUS\_meas}[\kappa]$, by counting the number of packets received in the last measurement period, and comparing it with the allowed rate command during the same period, denoted with

$r_{ALL}[\kappa]$. If $r_{SUS\_meas}[\kappa] < r_{ALL}[\kappa]$, the CQoSB assumes that $r_{SUS\_meas}[\kappa]$ is the source/core sustainable rate; if $r_{SUS\_meas}[\kappa] = r_{ALL}[\kappa]$, the CQoSB assumes that the source/core would have been capable of sustaining the full rate, and $r_{SUS\_meas}[\kappa]$ is set equal to $r_{MAX}[k]$ (6). The sustainable rate is then estimated as follows:

$$\hat{r}_{SUS}[\kappa] = h\hat{r}_{SUS}[\kappa-1] + (1-h)r_{SUS,meas}[\kappa] \qquad (9)$$

where $\hat{r}_{SUS}[\kappa]$ is the sustainable rate estimate updated with the κ^{th} measure and h is a tuning parameter. The estimate is used in the control law (7) in place of $r_{MAX}[k]$, unless the estimate is greater than $r_{MAX}[k]$ itself:

$$\begin{cases} r_{ALL}[k] = \hat{r}_{MAX}[k] - \\ \qquad -K\left\{q[k] - \hat{r}_{MAX}[k]T_D + \sum_{i=1}^{T_D} \hat{r}_{MAX}[k-i]\right\} \\ \hat{r}_{MAX}[k] = \min(\hat{r}_{SUS}[k], r_{MAX}) \end{cases} \qquad (10)$$

where $\hat{r}_{SUS}[k]$ is the most recent rate estimate.

Adaptation to variable transport delay. The second problem is that the transport delay is time-varying. The CQoSB can estimate it by periodically sending probes from the source to the destination Edge Router. Traffic overhead is almost negligible, since the only required information carried by the probes is a sequence number. By denoting the delay measures with $T_{D,meas}[\kappa]$, the following estimate is computed:

$$\hat{T}_D[\kappa] = g\hat{T}_D[\kappa-1] + (1-g)T_{D,meas}[\kappa] \qquad (11)$$

where $\hat{T}_D[\kappa]$ is the delay estimate updated with the κ^{th} measure and g is a tuning parameter. The delay estimate is used in place of T_D both in the computation of $r_{MAX}[k]$ (6) and in the control law (10):

$$r_{MAX}[k] = \frac{S}{\hat{T}_D[k] + 1/K} \qquad (12)$$

$$\begin{cases} r_{ALL}[k] = \hat{r}_{MAX}[k] - \\ \qquad -K\left\{q[k] - \hat{r}_{MAX}[k]\hat{T}_D[k] + \sum_{i=1}^{\hat{T}_D[k]} \hat{r}_{MAX}[k-i]\right\} \\ \hat{r}_{MAX}[k] = \min(\hat{r}_{SUS}[k], r_{MAX}) \end{cases} \qquad (13)$$

where $\hat{T}_D[k]$ is the most recent delay estimate. Notice that eq. (11) is the well-known updating law of the TCP delay estimate (see Von Jacobson et al. (1992)). Note that when queues build up in the Core Network the delay increases; the algorithm then reacts by diminishing the maximum rate which, in turn, diminishes the allowed rate.

Robustness. Finally, robust stability must be granted in presence of time-varying delay. In the considered scheme, the only uncertainty is in the time delay, denoted with $\delta[k]$, which is given by the difference between the actual delay and the delay estimate used by the controller:

$$\delta[k] = T_D[k] - \hat{T}_D[k] \qquad (14)$$

By following the procedure of (Morari and Zafioru, 1989), based on the Nyquist criterion, it can be stated that the system is robust if the following relations hold:

$$\left|\eta(e^{j\omega})l(e^{j\omega})\right| < 1 \quad \omega \in [0,2\pi) \qquad (15)$$

where $\eta(e^{j\omega})$ is the complementary sensitivity function and $l(e^{j\omega})$ is the multiplicative model error.

By denoting with $\bar{\delta}$ the maximum value of the uncertainty, in the considered control scheme $\eta(e^{j\omega})$ and $l(e^{j\omega})$ are given by:

$$\eta(e^{j\omega}) = \frac{P_0(z)G(z)}{1+P_0(z)G(z)} = \frac{K}{z-1+K} \qquad (16)$$

$$l(e^{j\omega}) = \frac{P(z)-\hat{P}(z)}{\hat{P}(z)} = e^{j\omega\bar{\delta}} - 1 \qquad (17)$$

where $G(z)$ is the controller of Fig. 3 with $T_D = \hat{T}_D[k]$.

Eq. (16) is a low-pass filter, whereas the module of eq. (17) is upper-bounded by

$$l_m(e^{j\omega}) = \begin{cases} \left|e^{j\omega\bar{\delta}} - 1\right| & \omega \in [0, \pi/\left|\bar{\delta}\right|) \\ 2 & \omega \in [\pi/\left|\bar{\delta}\right|, \pi) \end{cases} \qquad (18)$$

which is a high-pass filter (see Fig. 4). To verify eq. (15), it is then sufficient to find the appropriate value of K to tune the low-pass filter. It can be verified that K can be set according to the following law, which depends on $\bar{\delta}$ only:

$$K(\bar{\delta}) = 1 - e^{-\frac{\pi}{3|\bar{\delta}|}} \qquad (19)$$

Fig. 4 shows $l(e^{j\omega})$, $l_m(e^{j\omega})$, $\eta(e^{j\omega})$ and $|\eta(e^{j\omega})l(e^{j\omega})|$ for $\bar{\delta}$ = 5 sample periods.

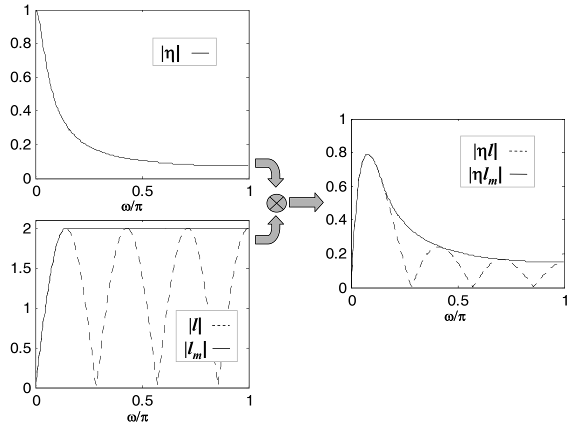

Fig. 4. Robust stability condition.

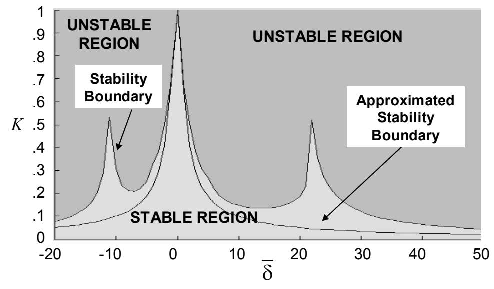

Fig. 5. Stability and approximated stability boundary.

Fig. 5 shows the maximum values of K ensuring system stability vs. $\bar{\delta}$, with transmission delay equal to 20 sample periods. The figure shows that the approximated stability boundary, described by eq. (17), is always within the actual stability boundary, obtained via the exact numerical criterion by (Jury, 1962) – which requires heavy computations and depends on both $\bar{\delta}$ and T_D.

From the above mentioned considerations, *Theorem* 2 follows:

***Theorem* 2:** By setting the controller gain K as in eq. (19), and by assuming $\bar{\delta} \geq \delta[k] = \left|T_D[k] - \hat{T}_D[k]\right| \; \forall k$, the system shown in Fig. 1 with the controler of equations (12) and (13) is stable.

The approximated boundary is not so conservative as it seems from Fig. 5 for particular values of $\bar{\delta}$: for instance, if the maximum uncertainty were 22 sample periods, we could not set $K = 0.5$, since the system is not stable for uncertainty values between 4 and 21.

The maximum uncertainty estimate $\hat{\delta}[k]$ is computed by the CQoSB as the maximum difference between the measured delay and the estimated delay within a temporal window equal to w samples. $\hat{\delta}[k]$ is used in the control law (13) and in the maximum rate computation (12).

Robust Adaptive Congestion Controller. Finally, the robust adaptive control scheme is depicted in Fig. 6.

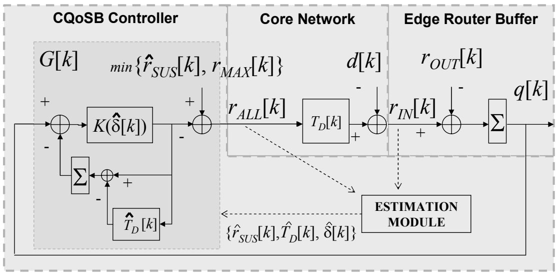

Fig. 6. Robust Adaptive Congestion Control Scheme.

The resulting rate command is:

$$\begin{cases} r_{ALL}[k] = \hat{r}_{MAX}[k] - \\ \qquad - K(\hat{\delta}[k]) \left\{ q[k] - \hat{r}_{MAX}[k]\hat{T}_D[k] + \sum_{i=1}^{\hat{T}_D[k]} \hat{r}_{MAX}[k-i] \right\} \\ \hat{r}_{MAX}[k] = \min\left(\hat{r}_{SUS}[k], r_{MAX}[k]\right) \end{cases} \tag{20}$$

where $\hat{\delta}[k]$ is the most recent rate estimate, $K(\hat{\delta}[k])$ is given by eq. (19), and:

$$r_{MAX}[k] = \frac{S}{\hat{T}_D[k] + 1/K(\hat{\delta}[k])} \tag{21}$$

3. SIMULATION RESULTS.

Two simulation runs have been setup to test the congestion control scheme under destination Edge Router congestion and Core Network congestion.

Destination Edge Router congestion occurs when the destination Access Network cannot sustain the transmission rate of the source Edge Router. In this situation, packets are accumulated in the destination router buffer: the congestion control algorithm objective is to set the transmission rate of the source router (i.e., the input rate of the buffer) equal to the destination Access Network sustainable rate (i.e., the output rate of the buffer).

Fig. 7 shows the simulation results, obtained with Matlab with the following parameters: sampling period $T_C = 0.01$ s; r_{OUT}[k] = ($7u_{-1}[k]$ – $4u_{-1}[k\text{-}2000]$) kbps; variable actual delay with ramp-like increases/decreases and white noise with 0 mean and variance 0.0025s^2; source and delay estimates performed every 0.1 s = 10 T_C; $g = h = 0.875$; uncertainty window $w = 2$ s; S is set sufficiently high not to limit the allowed rate. Fig. 7 shows i) that K is diminished every time the difference between the actual delay T_C and the estimated one $\hat{T}_D$ increases; ii) that the allowed rate $r_{ALL}[k]$ reproduces the output rate $r_{OUT}[k]$ (note that the low-pass filtering effect of the controller is more evident when the value of K diminishes); iii) that the queue length $q[k]$ is effectively controlled: the lower plot shows that if the gain K is not adaptive, (i.e., is set equal to 1, as if there were no uncertainties), the queue length $q_{NON\_ROBUST}[k]$ is not stable.

Core Network congestion occurs when one of the Core Routers cannot sustain the transmission rate of the source Edge Router, allowed by the destination Edge Router. In this situation, packets are accumulated in the Core Router buffer: the congestion control algorithm objective is to set the transmission rate of the source router equal to the Core Router sustainable rate in order to stabilize the queue length of the Core Router.

Fig. 8 shows the simulation results, obtained with OPNET, a packet-level simulation tool, with the following parameters: sampling period $T_C = 0.01$ s; source and delay estimate performed every 0.1 s = 10 T_C; $g = h = 0.875$; uncertainty window $w = 2$ s.

At time $t = 0$, the packets transmitted by the reference source starts feeding the core router buffer. Network congestion is simulated by a sudden decrease of the output rate of the core router buffer from 500 to 100 kbps at time $t = 30$ s; at time $t = 150$ s the reference source stops transmitting. Fig. 7 a) shows that the queue length grows as soon as congestion starts at time $t = 30$ s; if there is no edge router control, buffer overflows occur until congestion ends time $t = 150$ s.

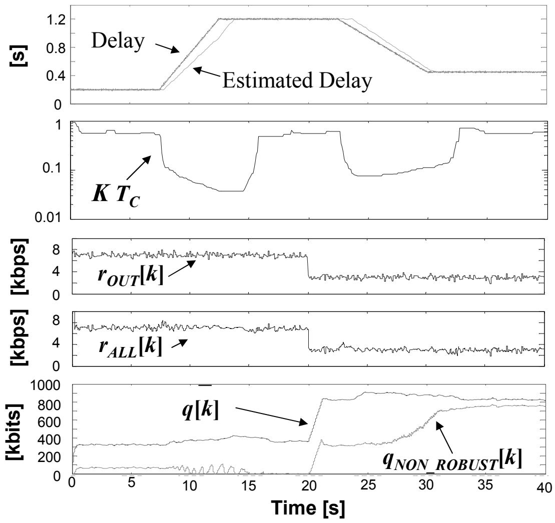

Fig. 7. Destination Access Network congestion.

On the contrary, if the congestion control is present, it can detect the congestion by estimating the transmission delay, which increases due to the queue length growth, as shown in Fig. 7 b). Then, according to the congestion control algorithm, $\hat{r}_{MAX}[k]$ and, consequently, $r_{ALL}[k]$ are lowered, so that $r_S[k]$ – which is the rate feeding the Core Network – is lowered as well (see Fig. 7 c)). In tis way, as shown in Fig. 7 a), the queue length in the Core Router is regulated and packet losses are reduced to 44 packets. Notice that buffer overflows can be effectively reduced only if the buffer size of the core routers are dimensioned according to the bandwidth-delay product of the core network, since this is also the reacting time of the protocol; however, as mentioned in Section 1, this is a realistic assumption.

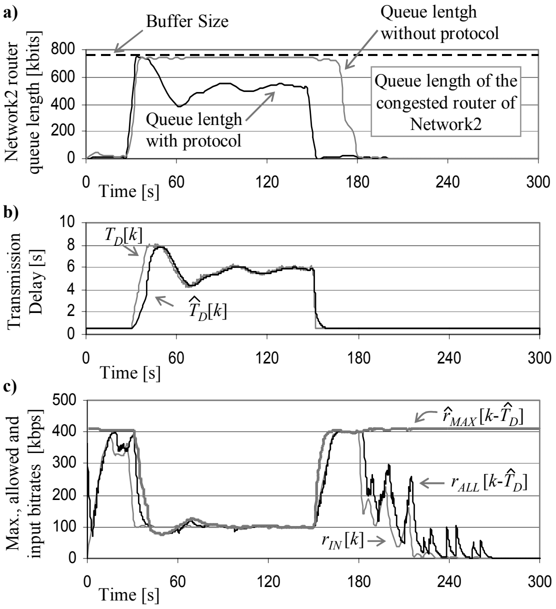

Fig. 8. Core Network congestion.

3. CONCLUSIONS AND FUTURE WORK.

An adaptive and robust congestion control has been proposed in the framework of next-generation networks, which can provide the controller with estimates of the network-induced delay. Delay compensation is provided by a Smith predictor controller; the delay estimate – considered in the controller to compute the allowed transmission rate – is updated by an on-line measurement system, wich does not introduce significant traffic overhead. Moreover, the gain of the controller is varied depending on the estimate of the uncertainty in the delay estimate, in order to obtain robust stability in the presence of time-varying delays.

On-going and future work is aimed at analysing the protocol behaviour in presence of multiple aggregate flows, with the purpose of better evaluating the effectiveness of the protocol and the achievable fairness.

REFERENCES

Altman, E., Başar, T. and Srikant, R. (1999). Congestion control as a stochastic control problem with action delays. *Automatica*, **35(15)**, 1937-1950.

Breslau, L., Knightly, E., Shenker, S., Stoica, I. and Zhang, H (2000) Endpoint Admission Control: Architectural Issues and Perormance, *in Proc. AC'M SIGCOMM*, Stockholm, Sweden, 57-69.

Choi, P. and Bettati, R. (2001) Endpoint Admission Control: Network Based Approach, *in Proc. of IEEE ICDCS*, Mesa, AZ, 227-235.

Delli Priscoli F. and Pietrabissa A. (2004) Design of a bandwidth-on-demand (BoD) protocol for satellite networks modelled as time-delay systems, *Automatica*, **40(6)**, 729-741.

Jacobson, V., Braden, R. and Borman, D. (1992) TCP Extensions for High Performance, *IETF-RFC 1323*

Jagannathan, S and Talluri, J. (2002) Predictive congestion control of ATM networks: multiple sources/single buffer scenario, *Automatica* **38(6)**, 815 – 820.

Jury, E. I. (1962) A simplified Stability Criterion for Linear Discrete Systems, *IRE Proc.*, **50(7)**, 1493-1500.

Mascolo, S. (1999) Congestion control in high-speed communication networks, *Automatica*, **35**, 1921-1935.

Morari, M. and Zafiriou, E. (1989) *Robust Process Control. Prentice Hall,* Englewood Cliffs, New Jersey.

Palmor, Z. J. (1996). Time-delay compensation - Smith predictor and its modifications. In: *The Control Handbook* (W. S. Levine (Ed.)), 224-237. CRC Press.

Quet, P. F., Ataşlar, B., İftar, A., Őzbay, H., Kalyanaraman, S. and Kang, T. (2002) Rate-based flow controllers for communication networks in the presence of uncertain time-varying multiple time-delays, *Automatica*, **38(7)**, 917-928.

Tarbouriech, S., Abdallah, C. T. and Ariola, M. (2001) Bounded Control of Multiple-Delay Systems with Applications to ATM Networks, *in Proc. of CDC 2001*, 2315-2220.