RCS: A Rate Control Scheme for Real-Time Traffic in Networks with High Bandwidth-Delay Products and High Bit Error Rates

Jin Tang Giacomo Morabito Ian F. Akyildiz
Broadband & Wireless Networking Laboratory
School of Electrical & Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332–0250
Email: {jtang, giacomo, ian}@ece.gatech.edu

Marjory Johnson
NASA
Ames Research Center
Moffett Field, CA 94035-1000
Email: mjj@riacs.edu

Abstract **- Currently there is no control for real-time traffic sources in IP networks. This is a serious problem because real-time traffic can not only congest the network but can also cause unfairness and starvation of TCP traffic. In this paper, a new rate control scheme, RCS, is introduced for real-time traffic in networks with high bandwidth-delay products and high bit error rates. RCS is based on the concept of using dummy packets to probe the availability of network resources. Dummy packets are treated as low priority packets and consequently they do not affect the throughput of actual data traffic. Therefore, RCS requires all the routers in the connection path to support some priority policy. Simulation experiments show that in environments with high bandwidth-delay products and high bit error rates, RCS achieves good throughput performance without penalizing TCP connections.**

Key Words: *Real Time Protocols, Flow Control, High Bandwidth-Delay Products, High Bit Error Rates.*

I. Introduction

REAL-TIME applications have strict requirements on end-to-end delay. For this reason the *Differentiated Service* and *Integrated Service* models have been proposed in recent years [4], [25]. Both of them try to guarantee lower bounds on the bandwidth allocated to each flow and, consequently, upper bounds on the end-to-end delay. Both Differentiated and Integrated Service models require a high amount of resources and, as a result, the services relying on them are expected to have high cost. On the other hand, there will be a large number of users interested in using real-time applications at a low cost. These users will share network resources without any reservation.

In a shared network, such as the Internet, all traffic flows are expected to be *good network citizens* or *TCP friendly* [23], i.e.,

- *Rule 1*: Their transmission rate can increase as long as the network is not congested, and
- *Rule 2*: Their transmission rate must decrease immediately when the network is congested.

Next generation IP-routers will penalize traffic flows not compliant with these rules [10].

In case of real-time streams, transmission rate, S, can be adjusted by adapting the quality of transmitted stream based on the available bandwidth. For example,

- Layered encoding is used and the source transmits the maximum number of layers that can fit in S [23].
- Encoding parameters are changed so that the output traffic rate is not higher than S.

Consequently, the quality of transmitted streams adapts to network condition.

Recently, much research work has been done to define TCP friendly rate control protocols for real-time applications in terrestrial networks. For example, in [13] a TCP-like scheme that does not perform retransmissions is proposed. The Streaming Control Protocol (SCP) is introduced in [6]. SCP is a modified version of TCP [14], [15] that performs TCP-Vegas-like rate adjustment [23]. In [19] and [28] transmission rate is adjusted based on TCP throughput model [11], [20]. In [27] and [23], two rate adaptation protocols, namely LDA and RAP, are presented. Both of them perform flow control for real-time streams by means of mechanisms very similar to those of TCP [14].

Packet losses are the only congestion sign in the current Internet. Accordingly, all previous schemes decrease transmission rate when packet losses are detected. However, some links, such as wireless and satellite links, are characterized by high link error rates and thus, packet losses can occur due to link errors with probability even higher than 10^{-2} [3], [1]. If the source decreases its transmission rate when a packet loss occurs due to link errors, then network efficiency decreases drastically [17], [21], i.e., it can be lower than 20%. This problem is amplified by the long delays involved in most Internet communications. For example, in Table 1 we show the round trip time, RTT, values measured when the mail server of the University of Catania (Italy) is connected to the WEB servers of other universities. Moreover, even higher RTT values have been observed in Wireless Wide Area Networks. For example, typical values of the round trip times in Cellular Digital Packet Data (CDPD) networks range between 800 msec and 4 sec [26].

Delay can be high because of the high hop distance between the two end systems. In fact, each hop causes a new queueing and processing delay. In Figure 1 we show the current hop distance probability distribution given in [18]. The current average hop distance is about 16 and is

The work of JT, GM, and IFA was supported by NASA-Ames under contract # NAG 2-1262.

University	Country	RTT
Georgia Tech	USA	200 msec
University of Campinas	Brasil	420 msec
Korea University	Korea	430 msec
Beijing University	China	800 msec

TABLE I

RTT VALUES FOR LONG DISTANCE CONNECTIONS.

expected to increase in the future [18].

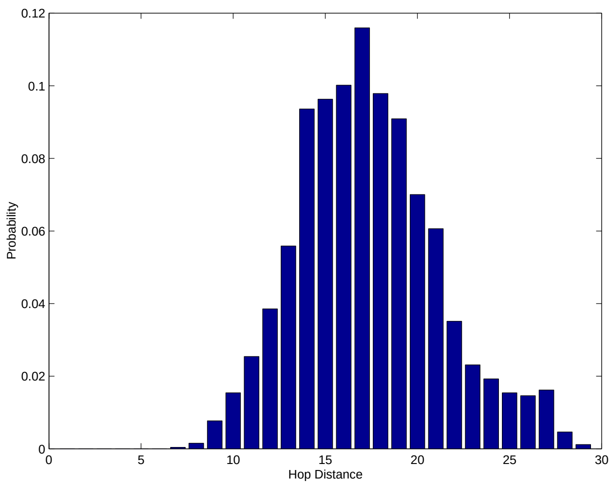

Fig. 1. Hop Distance Distribution.

We propose a Rate Control Scheme (RCS) for real-time traffic in networks with high bandwidth-delay products and high bit error rates. RCS is an end-to-end scheme which produces TCP-friendly traffic flows and improves throughput performance in networks with high bandwidth-delay products and high bit error rates. The new scheme requires all the routers in the connection path to apply some priority policy. In fact, RCS is based on the idea of using low priority packets called *dummy packets* [2]. Low priority packets are also used in [5] for call Admission Control. In our scheme, dummy packets are used by the source to probe the availability of network resources in the connection path. If a router on the connection path is congested, then it discards dummy packets first. If the routers are not congested, then dummy packets can reach the destination which then sends ACKs back. When the source receives an ACK for a dummy packet, this is the evidence that there are still unused resources in the network. RCS source can then set its transmission rate accordingly.

The paper is organized as follows. In Section II we introduce RCS and in Section III we show its behavior in two different cases: first, when a packet loss occurs due to link errors; second, when a packet loss occurs due to network congestion. Simulation results in Section IV show that in case of networks with high bandwidth-delay products and high bit error rates, RCS improves the efficiency without penalizing TCP traffic. Finally, we conclude the paper in Section V.

II. RCS: Rate Control Scheme

RCS is an end-to-end rate control scheme which uses additive-increase, multiplicative decrease (AIMD) [7], in order to produce TCP-friendly traffic flows while maintaining high throughput performance in networks with high bandwidth-delay products and high bit error rates.

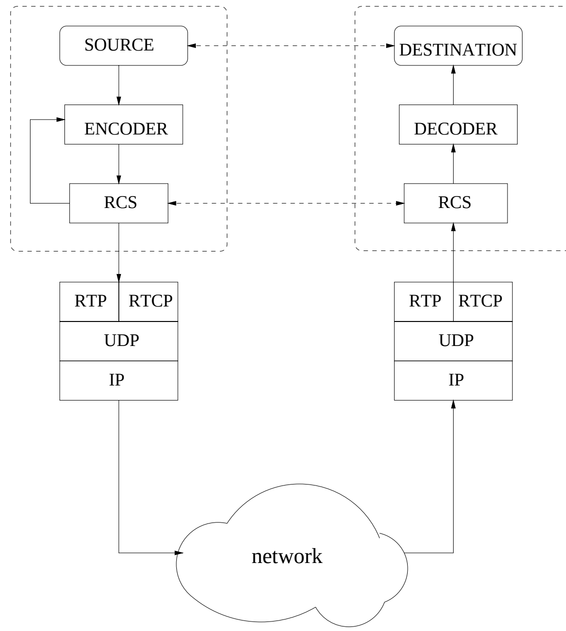

Fig. 2. RCS Architecture.

RCS is mainly implemented at the source but also needs some functions at the destination as shown in Figure 2. The value of the transmission rate, S, is determined by the rate control functions performed by RCS. The proposed scheme will run on top of RTP/RTCP [24] and UDP as shown in Figure 2.

At the destination RCS layer sends back an acknowledgement (ACK) for any received packet, as also suggested in [23]. Note that these ACKs are used only for flow control as will be explained in the following. No retransmissions are performed. At the destination, RCS layer passes the received data packet to the decoder and discards dummy packets.

A. Dummy Packets

RCS is based on the use of dummy packets. Dummy packets are low priority packets used by the source to probe the availability of network resources [2]. If a router on the connection path is congested, then it discards the IP packets carrying dummy packets first. Consequently, the transmission of dummy packets does not cause a decrease of throughput of actual *data packets*, i.e., the traditional packets. If the routers are not congested, then the dummy packets can reach the destination which sends the related acknowledgements. The ACKs for dummy packets are carried by low priority IP packets. The source interprets the

ACKs for dummy packets as the evidence that there are unused resources in the network and accordingly, can increase its transmission rate. Observe that dummy packets produce some overhead, but we outline that they use resources which otherwise would be unutilized.

The new scheme requires all routers in the connection path support some priority discipline. In fact, RCS injects dummy packets into the network regardless of the current traffic load. As a consequence, dummy packets may congest routers and effect data packet throughput if a router on the connection path does not apply any priority policy. Note that in traditional IP [22] networks the IP *type of service* (TOS) can be used for this purpose. In fact, one of the eight bits of the TOS field in the IP header gives the priority level of the IP packet [22]. Instead, more recent IP versions, e.g., IPv6 [9], explicitely provide several priority levels.

Currently, some routers in the Internet do not apply any priority policy. However, in the near future, Internet will support quality of service through the *Differentiated Service Model* (DiffServ) [4], which requires all routers to support multiple service classes. As a consequence, all recent commercial routers, e.g., Cisco series 7000 and 12000 [8], support at least the IP TOS.

Applications generating low priority traffic may be penalized by dummy packets even if priority is supported by routers. In fact, dummy packets may cause congestions for low priority traffic. Those congestions, however, last for a short period. In fact, dummy packets are transmitted only in two cases:

- *In the beginning of a new connection.* This occurs only once for each connection and the RCS source transmits dummy packets for a period long one round trip time approximately.
- *When a data packet loss is detected.* Packet losses can be due to network congestion or link errors. In both cases RCS transmits dummy packets for a period no longer than one round trip time. Moreover,
  - If the packet loss was due to network congestion, then high priority traffic is already using all the resources, i.e., no low priority traffic is passing through the connection.
  - If the packet loss was due to link errors, the transmitted dummy packets may harm low priority data traffic. However, we will show that the payback for this problem is a high increase of the throughput of high priority traffic.

As shown in Figure 3, RCS source is a finite state machine model with three states: *Initial*, *Steady* and *Detected*. In the following we present the behavior of RCS source in each of the above states.

B. Initial State Behavior

In the beginning of a new connection, the source has to set the initial transmission rate value, $S_{Initial}$. Let $S_{Available}$ be the transmission rate sustainable by the network. The choice of $S_{Initial}$ is important, because

- If $S_{Initial} >> S_{Available}$, then the new connection will cause network congestion.
- If $S_{Initial} << S_{Available}$, then resource utilization is low and will be low for a time interval proportional to the bandwidth-delay product.

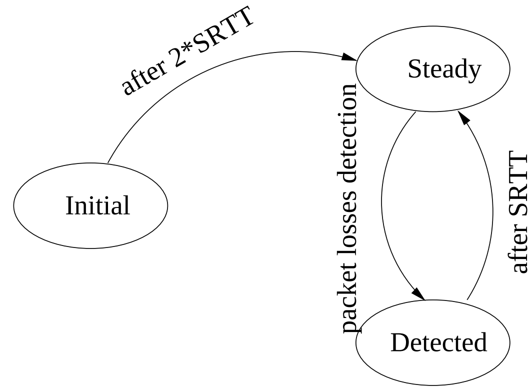

Fig. 3. Finite State Machine Model of the RCS Source.

RCS starts a new connection in the Initial state and remains there for a time interval, $t_{Initial}$, equal to two times the estimated round trip time, $SRTT$, i.e., ($t_{Initial} = 2 \cdot SRTT$). RCS maintains an $SRTT$ value as in case of TCP [14]. There are different methods to compute $SRTT$ [14], [16], [30]. In our scheme, we use $SRTT$ only as a reference timescale, thus, RCS performance is independent of $SRTT$ selection.

During the Initial state phase, RCS source executes the `Initial()` algorithm shown in Figure 4.

```
Initial()
    t_0 = t;
    t_1 = t_0 + SRTT;
    t_END = t + 2 · SRTT;
    IPG_Dummy = 1/S_Target;
    t_next_dummy = t_0 + IPG_Dummy;
    n_ACK = 0;
    while (t ≤ t_END)
        while (t ≤ t_1)
            while(t < t_next_dummy)
                if (DUMMY_ACK_ARRIVAL)
                    n_ACK = n_ACK + 1;
                end;
            end;
            send(DUMMY_PACKET);
            t_next_dummy = t_next_dummy + IPG_Dummy;
        end;
        if (DUMMY_ACK_ARRIVAL)
            n_ACK = n_ACK + 1;
        end;
    end;
    wdsn = −1;
    S = max(1, n_ACK)/SRTT;
    state=Steady;
end.
```

Fig. 4. `Initial()` Algorithm.

Let t represent the current system time and t_0 the initial time instant. The Initial phase lasts for two times $SRTT$, as a result, the `Initial()` algorithm will terminate at the time t_{END}, given by:

$$t_{END} = t_0 + 2 \cdot SRTT \tag{1}$$

Let S_{Target} represent the value of the data transmission rate needed to transmit the real-time stream with the highest quality. For example,

• If layered encoding is used, then S_{Target} is the transmission rate needed to transmit all the layers of the encoded stream.
• If adaptive encoding is used, then S_{Target} is the transmission rate needed to transmit the real-time stream encoded with the highest definition.
During $t_0 \leq t \leq (t_1 = t_0 + SRTT)$, RCS source sends dummy packets (`send(DUMMY_PACKET)`) at rate S_{Target}, i.e., the inter-transmission time is ($IPG_{Dummy} = 1/S_{Target}$). RCS uses the variable t_{next_dummy} to indicate the time when the next dummy packet has to be sent. After each dummy packet transmission, t_{next_dummy} is updated.

RCS source counts the number, n_{ACK}, of ACKs received for dummy packets. In fact, whenever it receives an ACK for a dummy packet (`DUMMY_PACKET_ARRIVAL`), it increases n_{ACK} by one. Note that n_{ACK} is used to estimate the available network resources, in fact

$$\frac{n_{ACK}}{SRTT} \approx \min\{S_{Target}, S_{Available}\} \tag{2}$$

Before leaving the Initial State, RCS sets the variable $wdsn = -1$ and the data transmission rate, S, as follows

$$S = \frac{\max\{1, n_{ACK}\}}{SRTT} \tag{3}$$

As will be clarified in the following section, RCS uses the variable $wdsn$ in order to be TCP friendly.

C. Steady State Behavior

In the Steady state, RCS source assumes that the network is not congested. Thus, according to the additive-increase scheme [7], it increases its transmission rate in a step-like fashion periodically. Moreover, upon receiving an ACK for a dummy packet, the RCS source checks the value of the variable $wdsn$. If $wdsn$ is greater than zero, then RCS source decreases $wdsn$ by one, i.e., ($wdsn = wdsn - 1$). Otherwise, RCS source increases its transmission rate by one packet per estimated round trip time, $SRTT$. We use the variable $wdsn$ in order to match RCS source behavior with TCP behavior when the network is congested. RCS source leaves the Steady state for the Detected state when a data packet loss is detected. Note that RCS source uses the same mechanism of TCP Reno [15] to detect data packet losses.

During the Steady phase, RCS source executes the `Steady()` algorithm shown in Figure 5. The algorithm uses the following variables:
• `END`: it is a boolean variable which is set to 1 when a data packet loss is detected (`PACKET_ LOSS_DETECTION`) to indicate that the algorithm must be terminated and RCS source must go to the Detected State, i.e., `state`=Detected.
• t_0: it gives the time instant when the current Steady phase started.
• t_{next_data}: it is the time instant when the next data packet has to be sent. When the current time, t, is greater than or equal to t_{next_data}, a data packet is sent (`send(DATA_PACKET)`), and t_{next_data} is updated.

```
Steady()
    END=0;
    t_0 = t;
    t_next_data = t_0;
    t_next_increase = t_0 + SRTT;
    while (END == 0)
        if (PACKET_LOSS_DETECTION)
            END=1;
        end;
        if (t >= t_next_data)
            send(DATA_PACKET);
            t_next_data = t_next_data + IPG;
        end;
        if (t >= t_next_increase)
            S = min(S + 1/SRTT, S_Target);
            IPG = 1/S;
        if (DUMMY_ACK_ARRIVAL)
            if (wdsn == 0)
                S = min(S + 1/SRTT, S_Target);
                IPG = 1/S;
            else
                wdsn = wdsn - 1;
            end;
        end;
    end;
    state=Detected;
end.
```

Fig. 5. `Steady()` Algorithm.

• IPG: it is the time interval between two successive data packet transmissions and is given by $IPG = 1/S$, where S is data transmission rate.
• $t_{next\_increase}$: it is the time instant when the transmission rate, S, must be increased. According to the additive increase scheme [7], S is increased periodically in a step-like fashion. In order to match the behavior of RCS source with the behavior of TCP [14], [15], the period is $SRTT$ and the step height is $1/SRTT$. However, the transmission rate, S, never exceeds the value S_{Target}. As a consequence, if $t \geq t_{next\_increase}$, the transmission rate is updated as follows

$$S = \min\{S + 1/SRTT, S_{Target}\} \tag{4}$$

and $t_{next\_increase}$ is updated

$$t_{next\_increase} = t_{next\_increase} + SRTT \tag{5}$$

• $wdsn$: when the ACK for a dummy segment is received, i.e., `DUMMY_ACK_ARRIVAL`, RCS source checks the value of $wdsn$. If $wdsn == 0$, then the transmission rate is increased as in eq. (4), otherwise $wdsn$ is decreased by one, i.e., $wdsn = wdsn - 1$.

D. Detected State Behavior

RCS source enters the Detected state when a data packet loss is detected. Packet losses are the only indication of network congestion in the current Internet. As a result, RCS source keeps the TCP [14], [15] conservative assumption that all packet losses are due to network congestion and, accordingly, halves its data transmission rate, S. However, it also starts transmitting dummy packets in order to probe availability of network resources. If the data packet loss was due to link errors, i.e., the network is not congested, then these dummy packets will be acknowledged and the data transmission rate, S, will be increased accordingly.

At the end of the Detected phase, RCS source goes back to the Steady state as shown in Figure 3.

`Detected()`
$t_0 = t$;
$t_{END} = t_0 + SRTT$;
$S = S/2$;
$IPG = 1/S$;
$t_{next\_data} = t_0$;
$wdsn = SRTT \cdot S$;
`while` $(t \leq t_{END})$
`while` $(t \leq t_{next\_data})$;
`send(DATA_PACKET)`;
`while` $(t \leq t_{next\_data} + IPG/3)$;
`send(DUMMY_PACKET)`;
`while` $(t \leq t_{next\_data} + 2 \cdot IPG/3)$;
`send(DUMMY_PACKET)`;
$t_{next\_data} = t_{next\_data} + IPG$;
`end`;
`state`=Steady;
`end`.

Fig. 6. `Detected()` Algorithm.

In the Detected state, RCS source executes the `Detected()` algorithm shown in Figure 6. The Detected phase lasts for a time interval equal to the estimated round trip time, $SRTT$, thus, it will finish at time ($t_{END} = t_0 + SRTT$), where t_0 is the time instant when the Detected algorithm is initiated. Moreover,

- The transmission rate, S, for data packets is halved, i.e., ($S = S/2$), and IPG is updated accordingly, i.e., ($IPG = 1/S$).
- The variable $wdsn$ is set to the value $wdsn = (SRTT \cdot S)$.
- The time, t_{next_data}, of the next data packet tranmission is set.

In the Detected phase, data packets are sent with rate S and two dummy packets are transmitted for each data packet. Packet transmissions are uniformly distributed, thus, the time interval between two successive transmissions is $IPG/3$.

At the end of the Detected phase, RCS source goes back to the Steady state, i.e., `state`=Steady as shown in Figure 3.

III. RCS Behavior

In this section we show how the `Detected()` and `Steady()` algorithms cooperate when a data packet loss is detected. More in detail, in Section III-A we show RCS behavior when a data packet loss occurs due to link errors, while in Section III-B we show RCS behavior when the cause of the data packet loss is network congestion.

A. Packet Loss Due to Link Errors

Let t be time instance in Figure 7.

- $t = t_0$

($S = S_0$, state=Steady).
Suppose that the data transmission rate is S_0 at time t_0.

- $t_0 < t \leq t_1$ (where $t_1 = t_0 + SRTT$)

($S = S_0/2$, $wdsn = SRTT \cdot S_0/2$, state=Detected).

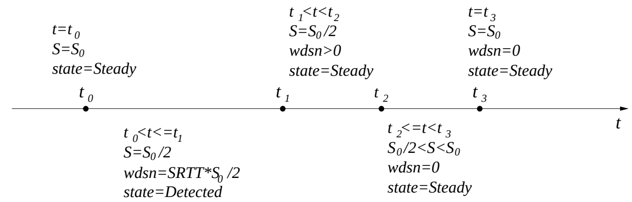

Fig. 7. RCS Behavior when a Packet Loss occurs due to Link Errors.

Suppose at time t_0 the source detects a packet loss. The source enters the Detected state, halves its transmission rate, i.e., $S = S_0/2$, and sets $wdsn$ to ($wdsn_0 = SRTT \cdot S_0/2$). Moreover, it transmits ($SRTT \cdot S_0$) dummy packets with rate equal to S_0.

- $t_1 < t < t_2$ (where $t_2 = t_1 + 0.5 \cdot SRTT$)

($S = S_0/2$, $wdsn > 0$, state=Steady).
At time $t = t_1$, the RCS source returns to the Steady state and starts to receive the ACKs for the dummy packets transmitted in the time interval $[t_0, t_1]$. If the network is not congested and thus, dummy packets are not lost, then the number of ACKs for dummy packets received in time interval $t_1 < t < t_2$ is ($SRTT \cdot S_0/2$). Consequently, $wdsn > 0$ and the transmission rate is not increased.

- $t_2 \leq t < t_3$ (where $t_3 = t_1 + SRTT$)

($S_0/2 \leq S \leq S_0$, $wdsn = 0$, state=Steady).
In this time interval, the source receives the ACKs for the other ($SRTT \cdot S_0/2$) dummy packets transmitted during the Detected phase. Since $wdsn$ value is 0, the source can increase its transmission rate by $1/SRTT$ each time it receives an ACK for a dummy packet.

- $t = t_3$

($S = S_0$, $wdsn = 0$, state=Steady).
At time t_3, the source has received the ACKs for all the ($n_{Dummy} = SRTT \cdot S_0$) dummy packets transmitted in the time interval $[t_0, t_1]$. Accordingly, the transmission rate has been increased by ($\Delta S = S_0/2$); in fact

$$\begin{aligned} \Delta S &= (n_{Dummy} - wdsn_0) \cdot \frac{1}{SRTT} \\ &= (SRTT \cdot S_0 - SRTT \cdot S_0/2) \cdot \frac{1}{SRTT} \\ &= \frac{S_0}{2}. \end{aligned} \quad (6)$$

Thus, the data transmission rate resumes the value it had before the data packet loss was detected.

In Figure 8, we show the transmission rate, S, dependent on time when a packet loss is due to link errors. Figure 8 is obtained by simulation assuming $S_0 = 22$ packets/sec and $RTT = 550$ msec. We see that the RCS source returns to its previous rate within approximately two round trip times when a packet loss occurs due to link errors.

B. Packet Loss Due to Network Congestion

Here we show that RCS sources halve their transmission rate when the network is congested.

Consider a single connection and let t be time instance given in Figure 9.

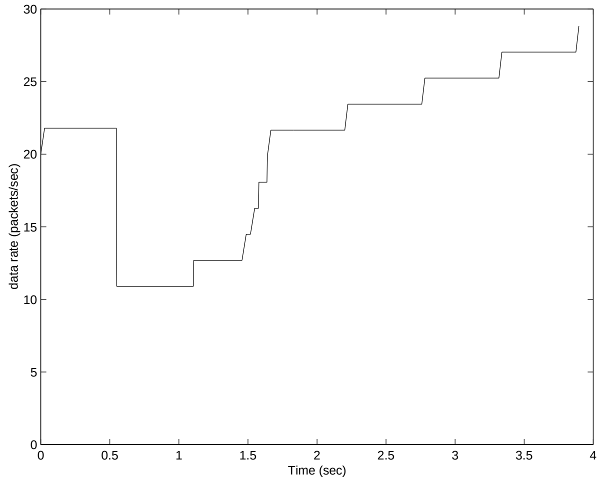

Fig. 8. Transmission Rate when a Packet Loss is due to Link Errors.

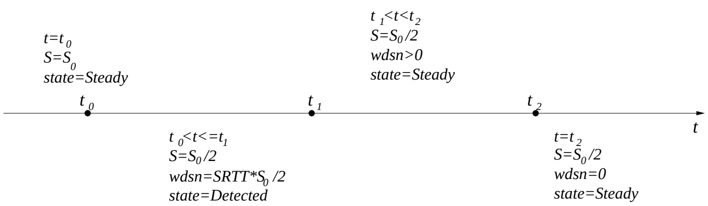

Fig. 9. RCS Behavior when a Packet Loss occurs due to Network Congestion.

• $t = t_0$
$(S = S_0$, state=Steady).
Let S_0 denote the data transmission rate at time t_0.
• $t_0 < t \le t_1$ (where $t_1 = t_0 + SRTT$)
$(S = S_0/2$, $wdsn = SRTT \cdot S_0/2$, state=Detected).
Suppose that at time t_0 the source detects a packet loss. Also suppose that the above loss is due to network congestion, i.e., the connection path can accommodate at most a transmission rate given by S_0. The source enters the Detected state, halves its transmission rate, i.e., $S = S_0/2$, and sets $wdsn$ to $wdsn_0 = SRTT \cdot S_0/2$. Moreover, it transmits $(SRTT \cdot S_0)$ dummy packets at a rate equal to S_0. Consequently, the overall transmission rate is $(3 \cdot S_0/2)$. However, the connection path can accommodate at most a rate given by S_0. Since data packets have high priority they are not discarded, whereas the half of the dummy packets (because they have low priority) will be discarded.
• $t_1 < t < t_2$ (where $t_2 = t_1 + SRTT$)
$(S = S_0/2$, $wdsn > 0$, state=Steady).
At time $t = t_1$, the RCS source returns to the Steady state and starts to receive the ACKs for the $(SRTT \cdot S_0/2)$ dummy packets transmitted in the time interval $[t_0, t_1]$ which are not discarded by the congested router. Since $wdsn > 0$ in the time interval $[t_1,\ t_2]$, then the transmission rate will not be increased.
• $t \ge t_2$
$(S = S_0/2$, $wdsn = 0$, state=Steady).
By the time t_2, all the ACKs for dummy packets which are not dropped by the network reach the source. The value of $wdsn$ has always been higher than zero when the ACKs for dummy packets were received. Consequently, the transmissions of the dummy packets during the detected state do not cause any increase in the transmission rate, which means that RCS is TCP friendly.

In Figure 10, we show the data transmission rate, S, dependent on time when a data packet loss occurs due to network congestion. Figure 10 was obtained through simulation assuming $S_0 = 22$ packets/sec and $RTT = 550$ msec. At time $t_0 = 0.5$ sec, a packet loss is detected, accordingly, the transmission rate, S, is set to $S = 11$ packets/sec. Then, for $t > t_0$, the transmission rate increases by $\{$one packet$\}/RTT$ each round trip time as in the TCP [14] case, i.e., RCS behavior is TCP-friendly.

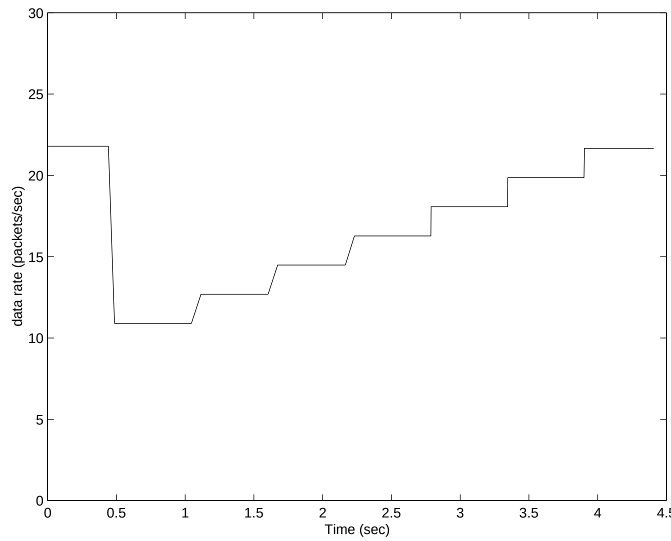

Fig. 10. Transmission Rate when a Packet Loss is due to Network Congestion.

IV. Performance Evaluation

In this section, we show throughput improvement (Section IV-A) and fairness (Section IV-B) of RCS. Performance results were obtained through simulation experiments.

A. Throughput Performance

Satellite networks are typical examples of networks with high bandwidth-delay products and high bit error rates. As a consequence, we simulate the system in Figure 11 where N sources transmit data to N destinations. The N streams are multiplexed in the Earth Station A, whose buffer accommodates K packets. Both data and dummy packets may get lost due to link errors with probability P_{Loss} in the satellite link. We assume that $N = 10, K = 50$ packets and the link capacity is $c = 1300$ packets/sec, which is approximately equal to 10 Mb/sec for packets of length 1000 bytes. We consider a Geostationary (GEO) satellite system, consequently, the RTT value is $RTT = 550$ msec.

In Figure 12, we show RCS throughput for different values of loss probabilities, P_{Loss}[1]. In Figure 12, RCS

[1] The bit error rate (BER) in satellite networks can be as high as

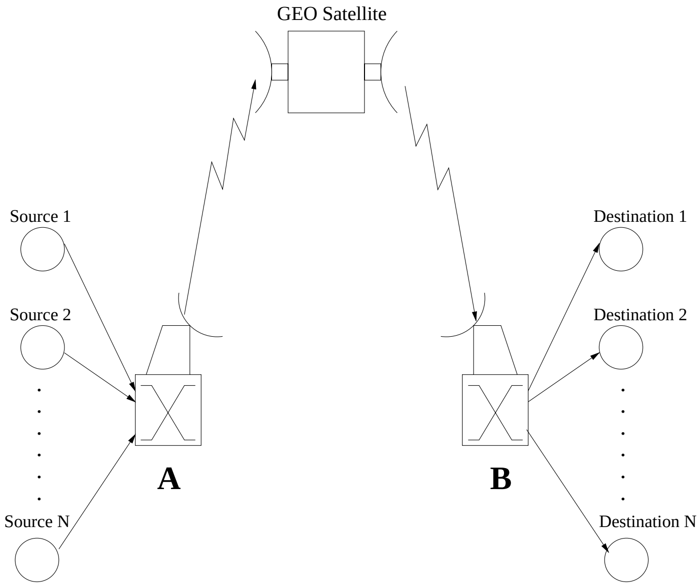

Fig. 11. Simulation Scenario.

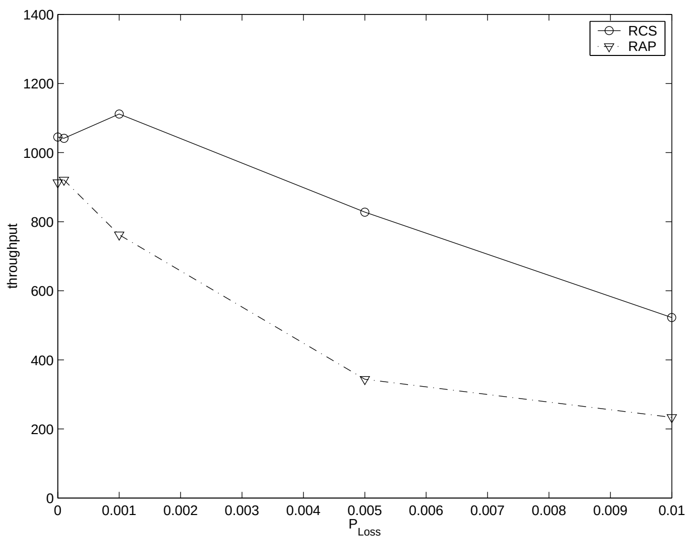

Fig. 12. Throughput Performance of RCS (solid lines) and RAP (dashed lines) for Different Values of Loss Probability.

throughput is maximum when $P_{Loss} = 10^{-3}$. This is because source no congestion related losses decrease the *phase effect* [12] characterizing the behavior of AIMD schemes. When the P_{Loss} further increases, the throughput decreases drastically. However, the throughput obtained by using RCS is always higher than the throughput obtained by using RAP [23]. RAP is a rate adaptation protocol for real-time applications applying AIMD scheme in order to be TCP-friendly. We have chosen RAP because, to the best of our knowledge, it gives the best performance.

In Figure 13, we show the overhead due to the transmission of dummy packets. Let

- N_{Dummy} be the total number of transmitted dummy

10^{-4}, i.e., one bad bit out of 10000 bits. For packets of 1000 bytes, the BER 10^{-4} gives a packet loss probability, P_{Loss} higher than 10^{-2} even if powerful FEC algorithm is applied.

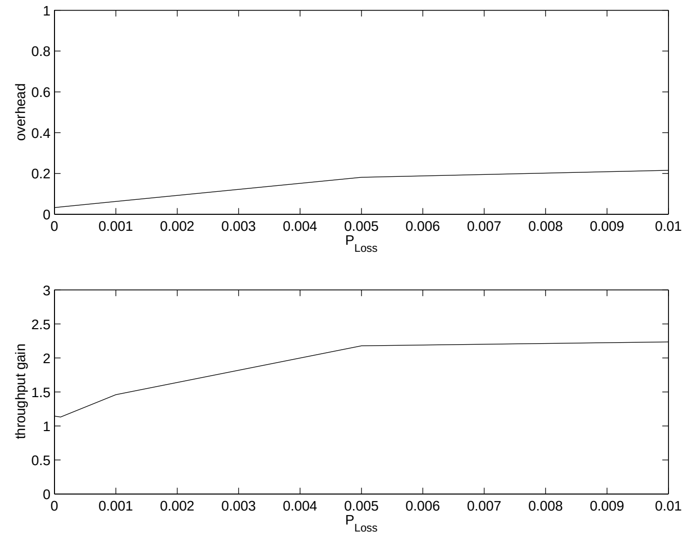

Fig. 13. Comparison of Bandwidth Overhead and Throughput Gain between RCS and RAP.

packets.

- N_{Data} be the total number of transmitted data packets.

We define the overhead as:

$$overhead = \frac{N_{Dummy}}{N_{Data} + N_{Dummy}}. \tag{7}$$

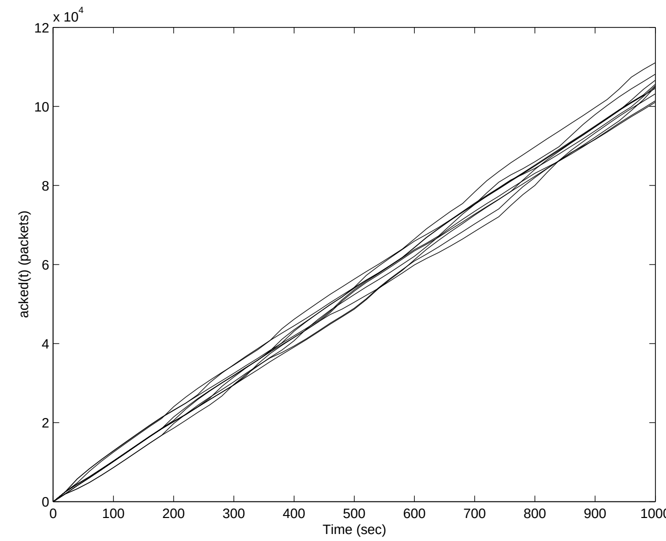

Fig. 14. Fairness in the Homogenous Case.

In the upper plot of Figure 13 we show the overhead dependent on the loss probability, P_{Loss}. Obviously, the overhead increases when P_{Loss} increases. Obesrve that the overhead can be as high as 21.5% when $P_{Loss} = 10^{-2}$. However, using dummy packets RCS gives much higher throughput than other rate control schemes for real-time traffic. For example, in the bottom plot of Figure 13, we show the throughput gain obtained using RCS. We evaluated the throughput gain as the ratio between the throughput given by RCS and the throughput given by RAP. Note that when

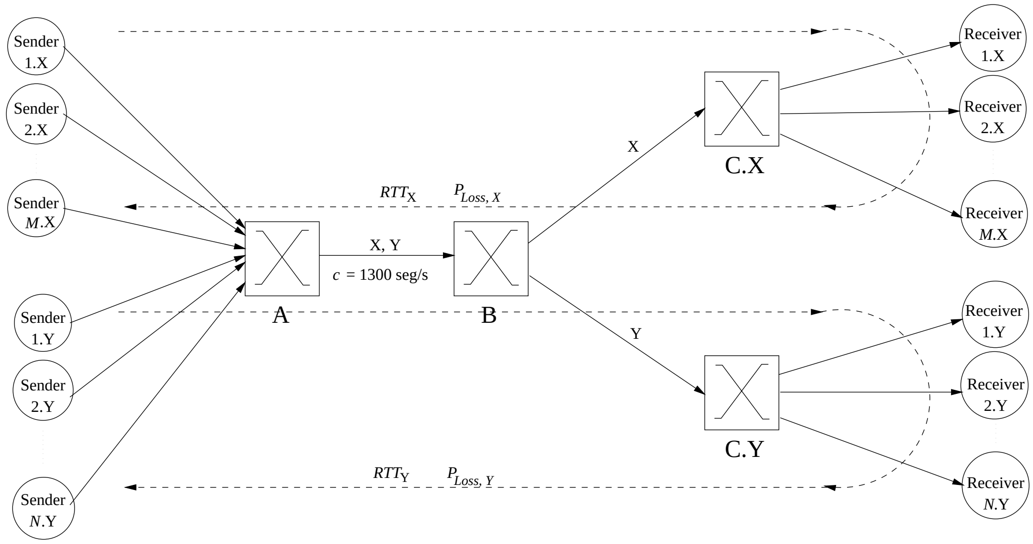

Fig. 15. Network Model for the Fairness Evaluation in the Heterogeneous Cases.

$P_{Loss} = 10^{-2}$, the overhead is 21.5%, but the throughput gain is higher than 200%.

B. Fairness

First we consider the homogenous case, i.e., the fairness between RCS connections with the same path, and later we evaluate the fairness of RCS in heterogeneus cases, i.e., connections can use different protocols or flows through different network paths.

B.1 Homogeneous Case

Here all connections pass through the same path and run RCS. Let $acked_i(t)$ represent the number of data packets acknowledged in the time interval $[0, t]$ for connection i, for $i = 1, 2, ..., N$. In Figure 14, we show $acked_i(t)$ dependent on time t for $i = 1, 2, ..., N$. We see that at any time, t, all connections have been acknowledged the same number of data packets approximately, i.e., $acked_{i'}(t) \approx acked_{i''}(t)$, for any i' and i''. This means that each RCS connection is given a fair share of network resources. Here, we assumed $N = 10$, $K = 50$ packets, $c = 1300$ packets/sec, $P_{Loss} = 0$ and $RTT = 550$ msec. Note that we obtained similar results for several other cases.

B.2 Heterogeneous Case

We consider the case shown in Figure 15. There are M connections of type X and N connections of type Y. Connections of type j, for $j = X, Y$, are characterized by round trip time equal to RTT_j, and loss probability for link errors equal to $P_{Loss,j}$. All connections pass through the link connecting the routers A and B, which is assumed to be the bottleneck and whose capacity is assumed to be $c = 1300$ packets/sec. The fairness, ϕ, is the ratio between the average throughput of connections of type X, r_X, and the average throughput of connections of type Y, r_Y, i.e.,

$$\phi = \frac{r_X}{r_Y} \tag{8}$$

It is obvious that the fairness becomes higher as ϕ approaches 1.

In Figure 16, we show the fairness when connections X use RCS and pass through a GEO satellite system, $RTT_X = 550$ msec, while connections Y use RCS but pass through a Medium Earth Orbit (MEO) satellite system, $RTT_Y = 250$ msec. We assumed $N = M = 5$. In Figure 16, we see that

$$\phi \approx \frac{RTT_Y}{RTT_X} \tag{9}$$

This fairness behavior has already been observed and analyzed in all AIMD disciplines [17], [29].

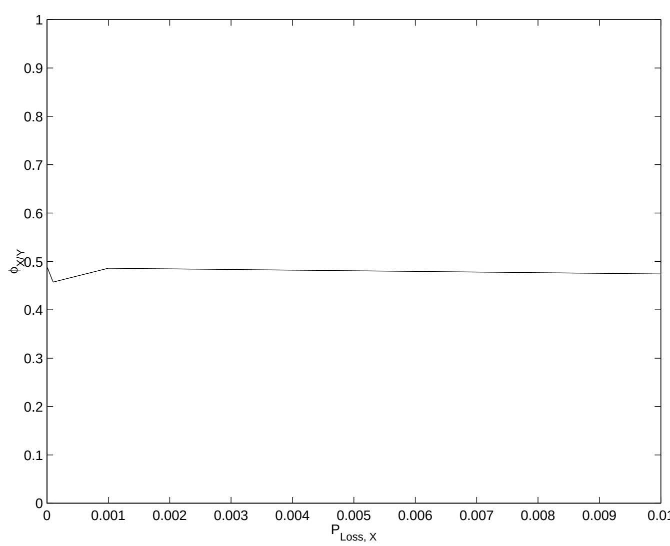

Fig. 16. Fairness between RCS Connections with Different Paths.

Now, we assume that both connections X and Y pass through a GEO satellite link, i.e., $RTT = 550$ msec, but connections of type X are TCP connections and connections of type Y are RCS connections. Moreover, we assume $N = M = 5$. In Figure 17, we see that r_Y values are higher than r_X thus, resources are not shared equally between TCP and RCS connections. This is mostly due to the problems of TCP in networks with high bandwidth-delay product and high link error rate [21]. In fact in Figure 17, we show that $r_X \approx r_{TCP}$, where r_{TCP} is the average throughput when all connections, i.e., both X and Y connections, use TCP. This means that RCS significantly improves network efficiency without penalizing TCP flows.

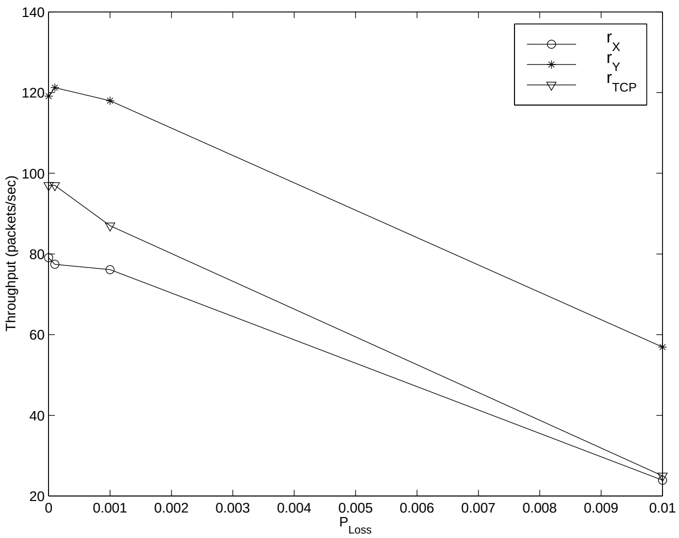

Fig. 17. Fairness between RCS and Traditional TCP Connections [14].

V. Conclusion

In this paper we introduced RCS, a new rate control scheme for real time traffic in networks with high bandwidth-delay products and high link error rates. RCS improves throughput using dummy packets to probe available network resources. Dummy packets are low priority packets which do not carry any new information to the destination. Therefore, RCS requires the routers along the connection to implement some priority mechanism. The main feature of RCS is that it is an end-to-end protocol, i.e., it needs to be implemented only at the source and destination. Simulation results show that RCS gives good throughput performance in networks with high bandwidth-delay product and high link error rate while providing TCP-friendly behavior.

References

[1] I. F. Akyildiz and S-H Jeong, "Satellite ATM Networks: A Survey," *IEEE Communication Magazine*, vol. 35, no. 7, pp. 30–43, July 1997.

[2] I. F. Akyildiz, G. Morabito, and S. Palazzo, "TCP-Peach: A New Flow Control Scheme for Satellite IP Networks," *Submitted for publication*, July 2000.

[3] H. Balakrishnan, S. Seshan, E. Amir and R. H. Katz, "Improving TCP/IP Performance over Wireless Networks," in *Proc. of ACM Mobicomm'95*, November 1995.

[4] Y. Bernet *et al.*, "A Framework for Differentiated Services," *Internet draft, draft-ietf-diffserv-framework-02.txt*, February 1999.

[5] G. Bianchi, A. Capone and C. Petrioli, "Throughput Analysis of End-to-End Measurement-Based Admission Control in IP," in *Proc. of IEEE Infocomm'2000*, March 2000.

[6] S. Cen, C. Pu and J. Walpole, "Flow and Congestion Control for Internet Media Streaming Applications," in *Proc. Multimedia Computing and Networking*, January 1998.

[7] D. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithm for Congestion Avoidance in Computer Networks," *Computer Networks and ISDN Systems Journal*, vol. 17, no. 1, pp. 1–14, June 1989.

[8] Cisco Systems' Web Page, available at *http://www.cisco.com*, April 2000.

[9] S. Deering and R. Hinden, "Internet Protocol Version 6 (IPv6) Specification," *IETF RFC 2460*, December 1998.

[10] S. Floyd and K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet," *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 458–472, August 1999.

[11] S. Floyd and K. Fall, "Router Mechanisms to Support End-to-End Congestion Control," *Technical Report*, February 1997.

[12] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transmission on Networking*, 1(4):397–413, August 1993.

[13] S. Jacobs and A. Eleftheriadis, "Real-Time Dynamic Rate Shaping and Control for Internet Video Applications," in *Proc. of the Workshop on Multimedia Signal Processing 1997*, June 1997.

[14] V. Jacobson, "Congestion Avoidance and Control," *ACM SIGCOMM*, pp. 314–329, August 1988.

[15] V. Jacobson, "Congestion Avoidance and Control," *Technical Report*, April 1990.

[16] P. Karn and C. Partridge, "Improving Round-Trip Time Estimates in Reliable Transport Protocol," *ACM Transactions on Computer Systems*, vol. 9, no. 4. November 1991.

[17] T. V. Lakshman and U. Madhow, "The Performance of TCP/IP for Networks with High Bandwidth-Dalay Products and Random Loss," *IEEE/ACM Transactions on Networking*, vol. 5, no. 3, June 1997.

[18] M. W. Maeda, "Next Generation Internet: SuperNet Technology," available at *http://www.darpa.mil/ito/research/ngi/su–pernet.html*, 1999.

[19] J. Mahdavi and S. Floyd, "TCP-friendly Unicast Rate-based Flow Control," *Technical Note*, available at *http://ftp.ee.lbl.gov/floyd/papers.html*, June 1997.

[20] T. Ott, J. Kemperman and M. Mathis, "Window Size Behavior in TCP/IP with Constant Loss Probability," in *Proc. of IEEE HPCS'97*, June 1997.

[21] C. Partridge and T.J. Shepard, "TCP/IP Performance over Satellite Links," *IEEE Network Magazine*, pp. 44–49, September/October 1997.

[22] J. Postel, "DoD Standard Internet Protocol," *IETF RFC 760*, January 1980.

[23] R. Rejaie, M. Handley and D. Estrin, "RAP: An End-to-End Rate-based Congestion Control Mechanism for Realtime Streams in the Internet," in *Proc. of IEEE Infocom'99*, April 1999.

[24] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport for Real-Time Applications," *IETF RFC 1889*, January 1996.

[25] S. Shenker, C. Partridge, and R. Guerin, "Specification of Guaranteed Quality of Service," *RFC 2212*, available at *ftp://ftp.isi.edu/in-notes/rfc2212.txt*, September 1997.

[26] P. Sinha, N. Venkataraman, R. Sivakumar, and V. Bharghavan, "WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks," in *Proc. of ACM Mobicom'99*, August 1999.

[27] D. Sisalem and H. Schulzrinne, "The Loss-Delay Based Adjustment Algorithm: A TCP-Friendly Adaptation Scheme," *Workshop on Network and Operating System Support for Digital Audio and Video*, July 1998.

[28] T. Turletti, S. F. Prisis, and J.-C. Bolot, "Experiments with a Layered Transmission Scheme over the Internet," *Rapport de recherche 3296, INRIA*, November 1997.

[29] M. Vojnovic, J.-Y. Le Boudec and C. Boutremans, "Global Fairness of Additive-increase and Multiplicative-decrease with Heterogeneous Round-Trip Times," in *Proc. IEEE Infocom 2000*, April 2000.

[30] L. Wolf, R. G. Herrtwich and L. Delgrossi, "Filtering Multimedia Data in Reservation-based Networks," *Kommunikation in Verteilten Systemen 1995 (KiVS)*, February 1995.