LDA+ TCP-Friendly Adaptation: A Measurement and Comparison Study

Dorgham Sisalem
GMD FOKUS
Berlin, Germany
sisalem@fokus.gmd.de

Adam Wolisz
TU Berlin/GMD Fokus
Berlin, Germany
wolisz@ee.tu-berlin.de

Abstract— **In this paper, we present an end-to-end adaptation scheme, called the enhanced loss-delay based adaptation algorithm (LDA+) for regulating the transmission behavior of multimedia senders in accordance with the network congestion state. LDA+ uses the real-time transport protocol (RTP) for collecting loss and delay statistics which are then used for adjusting the transmission behavior of the senders in a manner similar to TCP connections suffering from equal losses and delays. The performance of LDA+ is then investigated by running several simulations as well as measurements over the Internet. Additionally, by conducting simulations, the performance of LDA+ is compared to that of other TCP-friendly congestion control schemes presented in the literature.**

I. Introduction and Motivation

While congestion controlled TCP connections carrying time insensitive FTP or WWW traffic still constitute the major share of the Internet traffic today [1], recently proposed real-time multimedia services such as IP-telephony and group communication will be based on the UDP protocol. While UDP does not offer any reliability or congestion control mechanisms, it has the advantage of not introducing additional delays to the carried data due to retransmissions as is the case with TCP. Additionally, as UDP does not require the receivers to send acknowledgments for received data, UDP is well suited for multicast communication. However, deploying non-congestion controlled UDP in the Internet on a large scale might result in extreme unfairness towards competing TCP connections as TCP senders react to congestion situations by reducing their bandwidth consumption and UDP senders do not. Therefore, UDP flows need to be enhanced with control mechanisms that not only aim at avoiding network overload but are also fair towards competing TCP connections, i.e, be *TCP-friendly*. TCP-friendliness indicates here, that if a TCP connection and an adaptive flow with similar transmission behaviors have similar round trip delays and losses both connections should receive similar bandwidth shares. As an oscillative perceived QoS is rather annoying to the user, multimedia flows require stable bandwidth shares that do not change on the scale of a round trip time as is the case of TCP connections. It is, thus, expected that a TCP-friendly flow would acquire the same bandwidth share as a TCP connection only averaged over time intervals of several seconds or even over the entire life time of the flow and not at every time point [2].

In this paper, we describe a new scheme called the loss-delay based adaptation algorithm (LDA+), that adapts the transmission rate of UDP-based multimedia flows to the congestion situation in the network in a TCP-friendly manner. Basically, LDA+ regulates the transmission rate of a sender based on end-to-end feedback information about losses, delays and the bandwidth capacity measured by the receiver. With no observed losses, the sender can increase its transmission rate additively otherwise it needs to reduce it multiplicatively.

The work presented here is an extension of previous work presented in [3]. We have updated this work based on more recent proposals for analytical models of TCP [4], improved the used approach for dynamically determining the additive increase rate and finally the new algorithm avoids using parameters that had to be statically set by the user in the older version.

In Sec. II we take a brief look at some of the available TCP-friendly schemes in the literature. LDA+ is then presented in Sec. III. The performance of LDA+ is then investigated using simulations in Sec. V and measurements over the Internet in Sec. V. LDA+'s performance is compared to that of other schemes in Sec. VI.

II. Background and Related Work

Recently, there has been several proposals for TCP-friendly adaptation schemes that either use control mechanisms similar to those of TCP or base the adaptation behavior on an analytical model of TCP.

Rejaie et al. present in [5] an adaptation scheme called the rate adaptation protocol (RAP). Just as with TCP, sent packets are acknowledged by the receivers with losses indicated either by gaps in the sequence numbers of the acknowledged packets or timeouts. The sender estimates the round trip delay using the acknowledgment packets. If no losses were detected, the sender periodically increases its transmission rate additively as a function of the estimated round trip delay. After detecting a loss the rate is multiplicatively reduced by half in a similar manner to TCP.

Jacobs [6] presents a scheme called the Internet-friendly protocol that uses the congestion control mechanisms of TCP, however, without retransmitting lost packets. In this scheme, the sender maintains a transmission window that is advanced based on the acknowledgments of the receiver which sends an acknowledgment packet for each received data packet. Based on the size of the window the sender estimates then the appropriate transmission rate.

Padhye et al. [4] present an analytical model for the average bandwidth share of a TCP connection (r_{TCP})

$$r_{\text{TCP}} = \frac{M}{t_{\text{RTT}}\sqrt{\frac{2Dl}{3}} + t_{\text{out}} \min\left(1, 3\sqrt{\frac{3Dl}{8}}\right) l\left(1 + 32l^2\right)} \quad (1)$$

with M as the packet size, l as the loss fraction, t_{out} as the TCP retransmission timeout value, t_{RTT} as the round trip delay and D as the number of acknowledged TCP packets by each acknowledgment packet.

Using this model Padhye et al. [7] present a scheme in which the sender estimates the round trip delay and losses based on the receiver's acknowledgments. In case of losses, the sender restricts its transmission rate to the equivalent TCP rate calculated using Eqn. 1 otherwise the rate is doubled.

Additionally, various schemes have been proposed for the case of multicast communication such as [8], [9], [10] that aim at using a TCP-friendly bandwidth share on all links traversed by the multicast stream.

III. The Enhanced Loss-Delay Based Adaptation Algorithm (LDA+)

With most of the adaptation schemes presented in the literature [5], [4] the sender adapts its transmission behavior based on feedback messages from the receiver sent in short intervals in the range of one or a few round trip delays. This is particularly important for the case of reliable transport where the sender needs to retransmit lost packets. Additionally, with frequent feedback messages the sender can obtain up-to-date information about the round trip delay and, hence, use an increase in the round trip delay as an early indication of possible congestion.

On the contrary, LDA+ was designed to use the real time transport protocol (RTP) [11] for exchanging feedback information about the round trip time and the losses at the receiver. As RTP is currently being proposed as an application-level protocol for multimedia services over the Internet, using RTP would ease the introduction of adaptation schemes in the context of such services. RTP defines a data and a control part. For the data part RTP specifies an additional header to be added to the data stream to identify the sender and type of data. With the control part called RTCP, each member of a communication session periodically sends control reports to all other members containing information about sent and received data. However, with RTP, the interval between sending two RTCP messages is usually around five seconds. The in-frequency of the RTCP feedback messages dictates that an RTP sender can not benefit fast enough from rapid changes in the network conditions. Thus, the goal of RTCP-based adaptation is to adjust the sender's transmission rate to the average available bandwidth and not react to rapid changes in buffer lengths of the routers for example. This might be actually more appropriate in some cases than rapidly changing the transmission rate at a high frequency.

A. Measurement of Characteristics of Internet Paths

From Eqn. 1 it is obvious that for determining a TCP-friendly bandwidth share losses as well as delays on the links between the sender and receiver need to be taken into account. Additionally, the sender should not increase its transmission rate above the bottleneck rate of the link, i.e., the bandwidth of the smallest router on the path connecting the sender to the receiver.

As already mentioned, LDA+ uses RTP for transporting control information between the sender and the receiver. RTCP messages already include information about the losses and delays noticed in the network. Losses are estimated at the receiver by counting the gaps in the sequence numbers included in the RTP header of the data packets. The round trip delay is estimated by including a timestamp in the sender reports indicating the time the report was generated. The receiver includes in its reports the timestamp of the last received sender report (T_{LSR}) and the time elapsed in between receiving that report and sending the receiver report (T_{DLSR}). Knowing the arrival time (t) of the RTCP receiver report the sender calculates the round trip time (τ) as follows:

$$\tau = t - T_{\text{DLSR}} - T_{\text{LSR}} \quad (2)$$

This calculation requires no synchronization between the clocks of the sender and receiver and is therefore rather accurate.

Further, we enhanced RTP with the ability to estimate the bottleneck bandwidth of a connection based on the packet pair approach [12]. With this enhancement, the sender periodically transmits a number of data packets in bursts. Based on the time gaps (G) between two packets of size S the receiver can estimate the bottleneck bandwidth (B) as follows:

$$B = \frac{S}{G} \quad (3)$$

For detailed information about the probing process and methods for filtering out wrong estimates see [13], [14]. The information about which packets constitute the probing burst are then included in the sender RTCP messages. The results of the bottleneck estimation are reported in the receiver reports. The choice of running the measurement of the bottleneck bandwidth after the sending of each RTCP sender report, i.e., in intervals of minimally 5 seconds, or every few ones should be left to the application.

B. Rate Adjustment with LDA+

LDA+ is an additive increase and multiplicative decrease algorithm with the addition and reduction values determined dynamically based on the current network situation and the bandwidth share a flow is already utilizing. During loss situations LDA+ estimates a flow's bandwidth share to be minimally the bandwidth share determined with Eqn. 1, i.e., the theoretical TCP-friendly bandwidth share determined using the TCP model. For the case of no losses the flow's share can be increased by a value that does not exceed the increase of the bandwidth share of a TCP connection with the same round trip delay and packet size.

In the detail, after receiving the mth receiver report the sender estimates the bandwidth share (r_{m}) it should be using as follows:
No loss situation: In this case, the sender can increase its estimation of its TCP-friendly bandwidth share by an additive increase rate (A). To allow for a smooth increase of A and to allow flows of smaller bandwidth shares to increase their transmission rates faster than competing flows with higher shares, A is determined in an inverse relation to the ratio of the bandwidth share ($r_{\mathrm{m\text{-}1}}$) the sender is currently consuming and the bottleneck bandwidth (R) of the path connecting the sender to the receiver. Thus with an initial transmission rate of (r_0), an initial additive increase value of $\dot{A}$, A would evolve as follows:

$$A_{\mathrm{add}_1} = (2 - \frac{r_0}{R_i}) \times \dot{A} \tag{4}$$

$$A_{\mathrm{add_m}} = (2 - \frac{r_{\mathrm{m\text{-}1}}}{R}) \times A_{\mathrm{m\text{-}1}} \tag{5}$$

Both r_0 and $\dot{A}$ are set by the user but should be kept small relative to the bottleneck bandwidth.
To limit the rate increase maximally to the bottleneck bandwidth a second value of A is determined that converges to 0 as the bandwidth share of the flow converges to the bottleneck bandwidth. One function that fulfills this requirement is the exponential function in the form of

$$A_{\mathrm{exp_m}} = (1 - \exp^{-(1-\frac{r_{\mathrm{m\text{-}1}}}{R})}) \times r_{\mathrm{m\text{-}1}} \tag{6}$$

Finally, an RTP flow should not increase its bandwidth share faster than a TCP connection sharing the same link. With T seconds between the reception of two receiver reports and a round trip delay of (τ) a TCP connection would increase its transmission window by P packets with P set to

$$P = \sum_{p=0}^{T/\tau} p = \frac{(\frac{T}{\tau} + 1) \times \frac{T}{\tau}}{2} \tag{7}$$

with the window size being increased by one packet each round trip delay. Averaged over T, the sender should maximally increase its estimation of its bandwidth share by

$$A_{\mathrm{TCP_m}} = \frac{P}{T} \rightarrow \frac{\frac{T}{\tau} + 1}{2 \times \tau} \tag{8}$$

The additive increase value (A_{m}) is then set to

$$A_{\mathrm{m}} = \min(A_{\mathrm{add_m}}, A_{\mathrm{exp_m}}, A_{\mathrm{TCP_m}}) \tag{9}$$

Finally, r_{m} is determined as

$$r_{\mathrm{m}} = r_{\mathrm{m\text{-}1}} + A_m \tag{10}$$

Loss situation: For the case that the receiver reports a loss value of l, the transmission rate (r_{m}) is determined as follows:

$$r_{\mathrm{m}} = \max(r_{\mathrm{m\text{-}1}} \times (1 - \sqrt{l}), r_{\mathrm{TCP}}) \tag{11}$$

with $r_{\mathrm{m\text{-}1}}$ as the rate currently used by the sender and r_{TCP} determined using the TCP-model of Eqn. 1. Additionally, the increase factor (A) is reset to $\dot{A}$.

IV. Performance Investigation of LDA+ Using Simulations

For testing the performance of LDA+ we have conducted a large set of simulations investigating various aspects that might influence the behavior of LDA+. Among those tests we have investigated the robustness of LDA+ to various settings like number of competing flows, type and load of competing traffic, round trip delays, buffer management and length as well as the initial values of the scheme itself such as initial transmission rate and increase rate. Due to the lack of space we present here only a set of results describing the behavior of LDA+ in the presence of WWW traffic under different delays and the effects of the length of the report intervals on the efficiency of LDA+.

For testing the performance of LDA+ we used a simple topology, see Fig. 1, consisting of a bottlenecked link connecting m RTP-based senders and receivers, n TCP-based FTP connections and k WWW servers. The TCP connections were based on the Reno TCP specifications with fast retransmission and fast recovery algorithms [15]. The LDA+ and FTP sources were modeled as greedy sources that always had data to send at the rate allowed by the congestion control mechanism. The WWW servers generated short TCP connections with each connection lasting for the time required to carry a number of packets drawn from a Pareto distribution with the factor of 1.1 and a mean of 20 packets. After the end of a TCP connection the server paused for a period of time drawn from a Pareto distribution with a factor of 1.8 and a mean of 0.5 seconds [16] before starting a new connection.

The link had a capacity of 10 Mb/s and a propagation delay of τ. The bottlenecked router (Router_1) introduced a maximal additional delay of τ_{q} seconds due to the buffering of the data. The second router (Router_2) served only as a distributer of the data to the end systems and did not introduce losses or delays to the transported data. The routers used random early drop (RED) [2] for buffer management. A RED gateway detects incipient congestion by computing the average queue size. When the average queue size exceeds a preset minimum threshold the router drops each incoming packet with some probability. Exceeding a second maximum threshold leads to dropping all arriving packets. This approach not only keeps the average queue length low but ensures that all flows receive the same loss ratio and avoids synchronization among the flows. Actually, the measurements presented later and simulation studies conducted in [17] suggest that the performance of LDA+ was not affected by the used buffer management scheme. The maximum and minimum thresholds were set in this study to 0.8 and 0.3 of the maximum buffer size.

The packet size was set to 1000 bytes, $\dot{A}$ to 5 kb/s and all the RTP connections started at the same time with an initial transmission rate ($\dot{R}$) of 10 packets/sec.

Similar to [5], the TCP-friendliness (F) of LDA+ was deter-

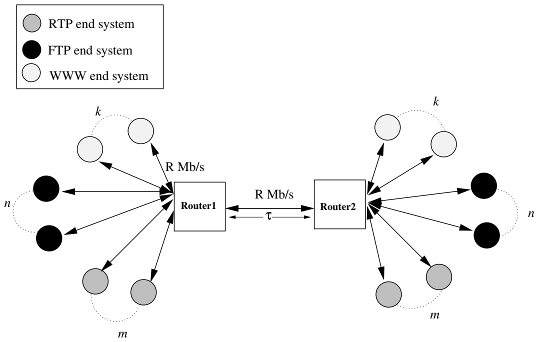


Fig. 1. Simulated performance testing topology

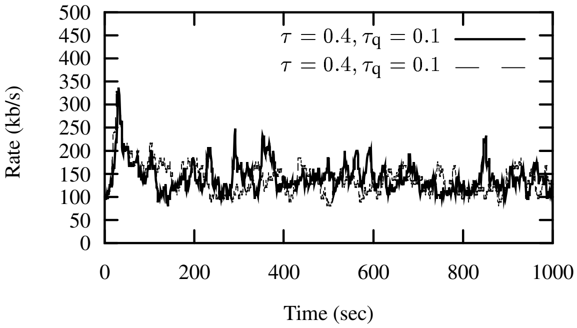


Fig. 2. Temporal behavior of LDA+

mined as

$$F = \frac{r_{\mathrm{RTP}}}{r_{\mathrm{TCP}}} \tag{12}$$

with r_{RTP} as the goodput of the RTP connection and r_{TCP} as the goodput of the TCP connection.

A. Performance of LDA+ in the Presence of WWW Traffic

For testing the performance of LDA+ when competing with long-lived TCP connections and bursty WWW traffic we used the simulation topology depicted in Fig. 1 with ($n = m = k = N = 27$) and various round trip propagation delays (τ) and maximum buffering delays (τ_{q}). Each simulation was repeated four times with each run lasting for 1000 seconds. The first 200 seconds were considered as a transient phase and were not taken into account in the here presented results.

Tab. I depicts the average values for the rate (r), standard deviation among the average rates of the flows ($\overline{\sigma}$), the fairness index (F) and the router utilization level (u) achieved for the simulations. The results presented in Tab. I suggest that LDA+ achieves acceptable friendliness values between 0.7 and 1 for a wide range of round trip delays (τ) and buffering delays (τ_q).

τ_{q} (sec)	0.10				0.50			
τ (sec)	r	$\overline{\sigma}$	F	u	r	$\overline{\sigma}$	F	u
0.1	155.9	4.0	0.85	0.87	148.5	7.3	0.82	0.87
0.2	134.7	4.9	0.72	0.84	156.3	10.3	0.88	0.88
0.4	134.0	8.5	0.78	0.80	168.6	15.1	1.02	0.88

TABLE I

ACHIEVED AVERAGE RATE AND FAIRNESS FOR LDA+ WITH 27 TCP, 27 LDA+ AND 27 WWW SERVERS COMPETING FLOWS

Fig. 2 displays the temporal behavior of two arbitrarily chosen LDA+ flows for the case of two different values of τ. We notice that while the LDA+ flows show some oscillations, these oscillations are only around $\pm 30\%$ of the average values and occur on a slow time scale.

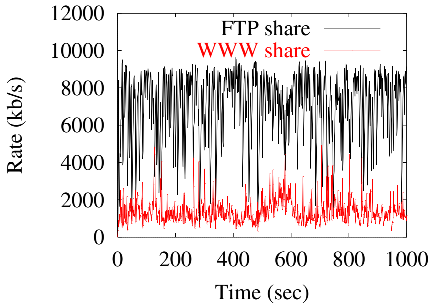
When investigating the interaction between adaptive flows and long-lived TCP connections one aims at achieving an equal bandwidth distribution among the competing flows. For the case of short lived connections, however, those performance metrics can not be used. Short lived connections do not carry enough data in order to fully utilize a bandwidth share similar to that of a long lived TCP connection. With the WWW server model we are using in this paper the average connection carries about 20 packets. Hence, those short lived TCP connections will usually send their data in a few round trip times. To test the effects of introducing LDA+ flows to a network inhibited by short lived TCP connections we compare the bandwidth share those short lived connections can assume when competing with LDA+ flows on the one hand and when competing with long lived TCP connections on the other. In this case, we expect that the bandwidth share of the short lived TCP connections should be in both cases rather similar.

For the simulation topology depicted in Fig. 1 with a RED router, round trip propagation delay (τ) of 0.4 seconds, queuing delay of 0.1 seconds and a link bandwidth of 10 Mb/s we ran two simulations with 27 FTP connections and 27 WWW servers in the first case and 27 LDA+ flows and 27 WWW servers in the second.

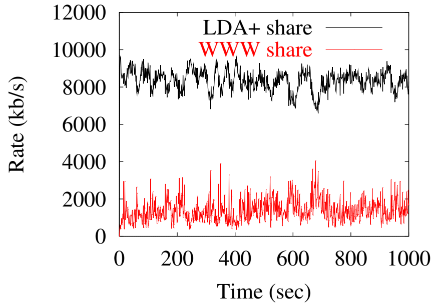
Fig. 3 shows the bandwidth distribution for the case of WWW traffic competing with LDA+ flows on the one hand and with long lived TCP connections, i.e., FTP connections on the other. While the LDA+ flows receive a much higher bandwidth share than the WWW traffic, the share of the LDA+ flows is similar to that of the TCP connections under the same conditions. This suggests that adding LDA+ flows to the network does not affect the performance of the WWW traffic under the simulated situation any different than long lived TCP connections do.

B. Effects of the RTCP Intervals on the Performance of LDA+

To investigate the possible benefits of using smaller intervals we simulated the topology depicted in Fig. 1 with 27 LDA+ flows competing for a bottleneck of 10 Mb/s. As background traffic we used 27 WWW servers. The round trip propagation delay τ was set to 0.4 seconds and the maximum queuing delay (τ_q) to 0.1 seconds. Each simulation was run for 500 seconds and



(a) FTP and WWW

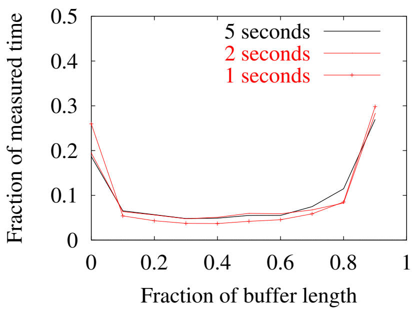


(b) LDA+ and WWW

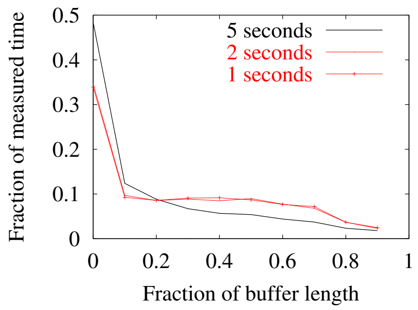
Fig. 3. Bandwidth sharing for the case of WWW traffic competing with FTP or LDA+ flows

the presented results are the average values of 10 simulation runs with different seed values.

Congestion in packet switched networks is manifested by buffer overflows at the routers. That is, a constantly filled buffer is a good indication of a highly congested network. Network underutilization is on the other hand indicated by empty buffers. To investigate the effects of changing the RTCP interval on the network congestion state, in our simulations we measured the time the buffer was observed to be occupied to a certain percentage of the maximum buffer, see Fig. 4. That is, for the case of RED buffer management, the results depicted in Fig. 4(b) reveal that 0 to 0.1 of the buffer was occupied for 48% of the simulation time for the case of an RTCP sending interval of 5 seconds and the range between 0.1 and 0.2 of the buffer was occupied 17% of the time. Fig. 4 suggests that for both cases of FIFO and RED buffer management, using larger RTCP intervals results in reduced network congestion as indicated by the lower occupancy times of the higher ranges of the buffer. So for the case of an RTCP interval of one second, the buffer was 2.4% of the simulation time above 90% occupied, whereas for intervals of five seconds this value is reduced to 1.8%. It is also interesting to see that with RED the buffer occupancy in the high region of above 80% is much lower than that of FIFO, indicating a more balanced load situation and therefore lower queuing delays.



(a) FIFO router



(b) RED router

Fig. 4. Effects of the length of RTCP intervals on the buffer occupancy

Tab.II presents the average bandwidth share of the flows (r) as well as the average utilization of the router (u) for both cases of using FIFO and RED for buffer management. The rather equal utilization and rate results suggest that reducing the RTCP interval does not improve the performance of LDA+ significantly. Actually, the reduced buffer occupancy values indicate that with larger RTCP values LDA+ is more conservative in its adaptation behavior and hence less affected by network instabilities and traffic burstiness.

	FIFO		RED	
Interval (sec)	r	u	r	u
5	126.88	0.991	105.031	0.971
2	122.430	0.993	118.066	0.991
1	121.206	0.997	107.361	0.997

TABLE II

PERFORMANCE OF LDA+ FOR DIFFERENT RTCP INTERVALS

V. Measuring the Performance of LDA+ over the Internet

The simulation results in Sec. IV suggest the efficiency of LDA+ in reducing network congestion while maintaining a high network utilization level as well as its friendliness towards competing TCP connections over a wide range of parameters. In this part of the work, we investigate the performance of LDA+ when used over the wide area Internet.

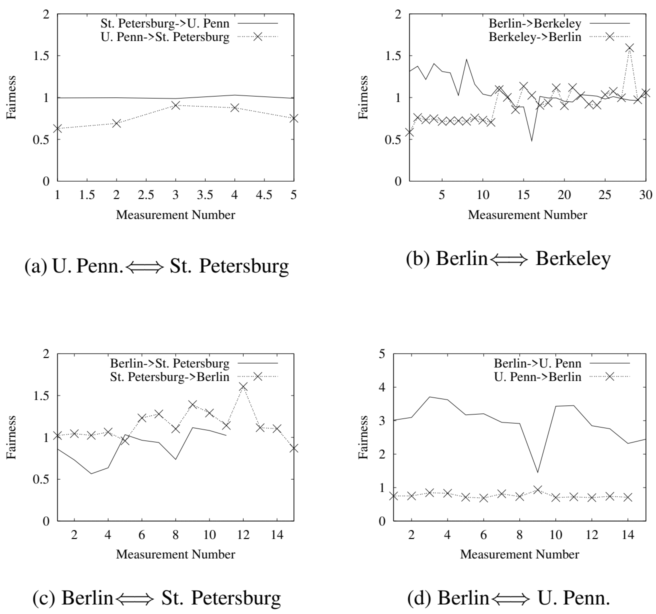
For the purpose of testing LDA+ we have conducted several measurements on different parts of the Internet. Each measurement consisted of a host sending data packets over a TCP connection to some other destination as fast as it can. Simultaneously, the host sends UDP packets to the same destination with the transmission rate determined using LDA+. Each measurement was done several times over different times of the day.

Host name	Domain	Operating System	Location
donald	fokus.gmd.de	SunOS 5.5	Berlin
verba	stu.neva.ru	SunOS 5.6	St. Petersburg
systems	seas.upenn.edu	SunOS 5.5	U. Pennsylvania
ale	icsi.berkeley.edu	SunOS 5.6	Berkeley, CA.

TABLE III
HOSTS USED IN THE EXPERIMENTAL TESTS

The host names and domains as well as their operating systems and locations are listed in Tab. III. Each measurement consisted of sending 10000 packets on both the TCP and UDP flows. The packet size was held constant to 1000 bytes which is a size often used in video conferencing applications. For networks that do not support this packets size it might be appropriate to extend LDA+ with path MTU discovery mechanisms [18] to optimize its performance and avoid fragmentation. The initial additive increase rate ($\dot{A}$) was set to 5 kb/s and the initial transmission rate of the UDP flows was set to 10 packets/s. The maximum transmission rate of the UDP flow was limited to 1 Mb/s. The friendliness factor (F) of LDA+ is determined here as in Eqn. 12. To obtain a detailed view of the performance of LDA+ we collected information about the sent and received RTP and TCP data in intervals of one second. Additionally, we collected the loss, round trip delay and bottleneck bandwidth information carried by the RTCP protocol. To estimate the average bottleneck bandwidth we relied on an approach similar to that used in the BPROBE tool [19] for filtering out incorrect estimates. That is, similar bottleneck bandwidth values estimated by the receiver using Eqn. 3 were clustered into intervals and the average of the interval with the highest number of estimates was chosen. The estimated value was then sent back to the sender with the receiver reports.

Obviously, this approach has lots of drawbacks. We did not include the time between the transmission of two probe packets. But, as we sent the probe packets at the sender's network access speed which in our case was 10 Mb/s we can usually ignore this time. Also, we did not consider packet drops or competing traffic. However, testing this approach on different Internet connections we achieved results comparable to those estimated by a more complicated tool such as the PATHCHAR tool [20].

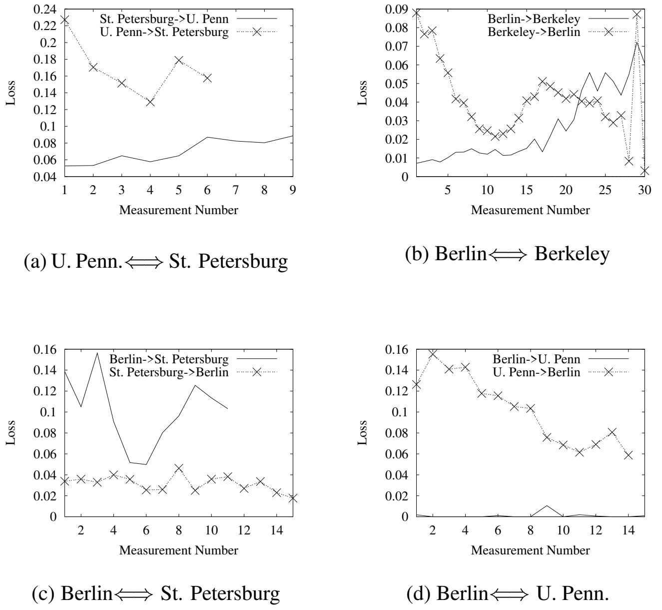


(a) U. Penn. ⟺ St. Petersburg
(b) Berlin ⟺ Berkeley
(c) Berlin ⟺ St. Petersburg
(d) Berlin ⟺ U. Penn.

Fig. 5. TCP friendliness measured over different Internet paths

Fig. 5 shows the friendliness results as measured over different links in Europe and the States. Fig. 6 depicts the average loss values observed during these measurements.

The fairness results depicted in Fig. 5 show great variations among the different flows and even on the different directions of the same flow. The links between St. Petersburg and Berlin as well as St. Petersburg and U. Penn. are rather lossy in both directions. Under these conditions the measured friendliness index (F) varies between 0.6 and 1.4 with most of the measured values in the range of 0.8 and 1.2. These results are rather similar to the results achieved using the simulation model in Sec. IV and indicate the TCP-friendliness of LDA+. The results depicted in Fig. 5(d) are, however, contradictory. In the direction form Berlin to U. Penn. we have a friendliness factor of around 0.8 on the average which means that the LDA+ controlled flow actually receives a smaller share of the bandwidth than the competing TCP connection. The measurements on the opposite direction indicate, however, that the LDA+ controlled flow receives four times as much bandwidth as the competing TCP connection. Actually, the LDA+ controlled flow manages to achieve the maximum transmission rate and to stay at this level. While these results sound contradictory on the first sight, they actually resulted from the asymmetry of the Internet link between Europe and the States. Looking at the loss results depicted in Fig. 6(d) reveals that while the traffic from Berlin to U. Penn. suffered on the average from losses below 1%, the traffic sent in the other direction



(a) U. Penn.⟺ St. Petersburg

(b) Berlin⟺ Berkeley

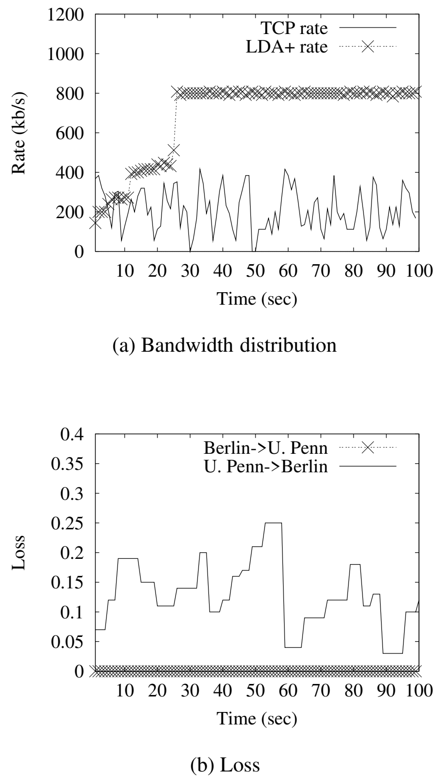
(c) Berlin⟺ St. Petersburg

(d) Berlin⟺ U. Penn.

Fig. 6. Losses measured over different Internet paths

had an average loss of more than 10%. Fig. 7 shows a detailed snapshot of the measurement labeled 2 in Fig. 5(d) and displays the transmission rate of both the TCP connection and LDA+ controlled flow and the losses measured over intervals of one second in both directions on the link connecting U. Penn. and Berlin. In Fig. 7(b) we notice that while in the direction from Berlin to U. Penn. no losses were observed during the entire measurement period the traffic from U. Penn. to Berlin faced losses ranging from 5% up to 25% during the same period. This asymmetry leads to the well known *ack compression* problem [21], [22]. On symmetric links TCP acknowledgment packets arrive at the sender with a similar rate to the arrival rate of the data packets at the receiver. During the congestion avoidance phase, each arriving acknowledgment clocks one or two new packets out at the sender and leads to an increase in the congestion window (CWND) by $\frac{1}{\mathrm{CWND}}$ each round trip delay. For the case of a slower or lossy back link, acknowledgments might arrive in clusters or might get lost. In the worst case, loosing several acknowledgment packets in a row might cause a timeout at the sender which leads to a severe reduction of the congestion window and the sender retransmitting packets that have already reached the receiver. In any case, the clustering or loss of acknowledgments can result in idle states at the sender. That is, after sending a complete window worth of packets, for example from packet (P_1) up to packet (P_y) the sender needs to wait for an acknowledgment for packet (P_1). If the acknowledgment for this packet was lost the sender will only be able to start sending data after receiving the acknowledgment for packet ($P_x : y \geq x > 1$). Further, after receiving the acknowledgment for (P_x) the sender can send up to x packets instead of one or two at once at the access speed of the sending station. Sending a large burst of back to back data packets increases the probability of one of those packets being dropped as the speed of the transmission and the number of packets might exceed the capacities of some router on the path to the receiver.

So while the transmission rate of UDP flows is adapted only to the losses on the way from the sender to the receiver, the TCP connections' transmission behavior is affected by the loss conditions on the direction from the receiver to the sender as well. Thus, in this situation setting the transmission rate of the UDP senders exactly to the equivalent TCP rate would be ineffective. The transmission rate of the UDP sender would be artificially restricted to a level that actually leads to underutilization of the network and might not fulfill the needs of the user.



(a) Bandwidth distribution

(b) Loss

Fig. 7. Bandwidth distribution and losses measured on both directions from Berlin to U. Penn.

Adaptation schemes that adjust the transmission rate based on acknowledgment packets from the receivers such as [23], [5], [6] have the same problem as TCP and can not benefit from the available network resources appropriately. Additionally, note that during the entire observation period in Fig. 6 no losses were measured on the link from the sender to the receiver. Thus, in such a situation relying solely on the TCP model of Eqn. 1 is not appropriate for determining the transmission rate.

Fig. 8 shows the goodput, measured in intervals of 2 seconds, of the competing TCP and LDA+ streams during a period of 200 seconds of the measurement shown as point 18 in Fig. 5(b). LDA+ shows a less oscillatory behavior than TCP and has in general a comparable rate to that of TCP.

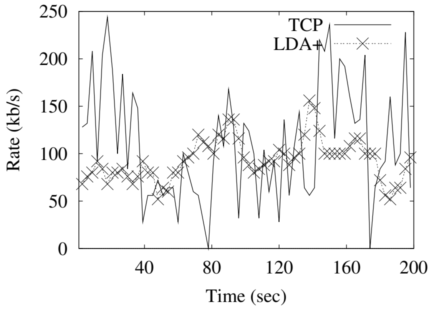


Fig. 8. Temporal behavior of competing TCP and LDA+ streams

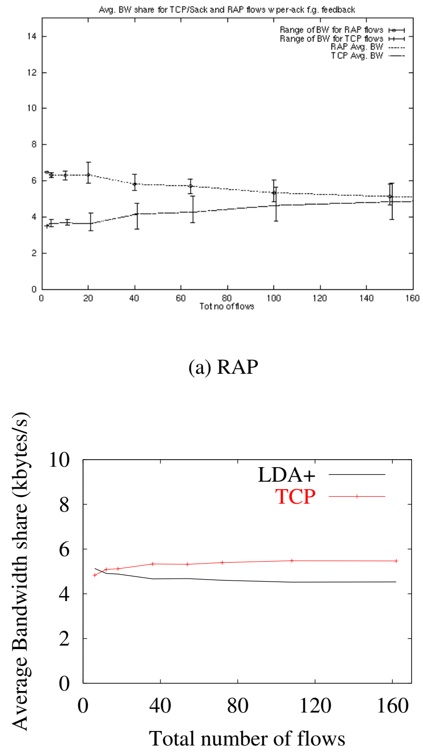
VI. Comparison of the Performance of LDA+ to Other Congestion control Schemes

In this part of the work, we compare between the performance of LDA+ and a row of recent proposals for TCP-friendly congestion control. In order to cover a wide range of possible approaches for TCP-friendly adaptation we compare LDA+ to a rate-based scheme and a scheme based on an analytical model of TCP. For comparing the schemes, we picked out some representative test cases as were described in the papers presenting those algorithms. We re-simulated those cases in our simulation environment and compared the achieved results using LDA+ with the results achieved by the other schemes as were reported by their authors. This approach reduces possible errors in the comparisons due to misinterpretations or wrong implementation of the algorithms.

A. Comparison of LDA+ with a Rate-Based TCP-Friendly Adaptation Scheme

With the rate adaptation protocol (RAP) [5], already briefly described in Sec. I, the receiver acknowledges the reception of each incoming packet. Based on those acknowledgments the sender can estimate the round trip delay and losses. In the absence of packet losses the sender increases its transmission rate by an additive amount determined using the estimated round trip delay. Additionally, the transmission rate is subject to fine grain adaptation related to variations of the short-term average round trip delay compared to the long-term average round trip delay. After detecting a packet loss, the rate is reduced by half in a similar manner to TCP.

For testing the performance of RAP Rejai et al. used in [5] a simulation topology similar to that depicted in Fig. 1 with N TCP connections sharing a single bottleneck with N UDP flows. The round trip delay (τ) was set to 0.04 seconds and the bottleneck bandwidth (R) was set to ($2 \times N \times 40$ kb/s). The router used FIFO buffer management with the buffer length set to ($R \times \tau \times 4$). The packet size was set to 100 bytes. To avoid synchronization among the flows, each transmitted packet was delayed randomly by an amount varying between 0 and the bottleneck service time. In [5] SACK-TCP connections were used. As our simulation environment does not support SACK TCP we referred to using Reno-TCP which delivers similar results in general but might show a lower performance for the case of bursty losses.



(a) RAP

(b) LDA+

Fig. 9. Bandwidth distribution with RAP and LDA+

Fig. 9 depicts the results of the average bandwidth share for the adaptive flows and the TCP connections. We can observe that even though RAP requires a higher control overhead due to the frequent acknowledgments, equal bandwidth shares are only reached for higher numbers of competing flows. For a low number of competing flows the RAP flows actually receive a 50% higher bandwidth share than the TCP connections. Additionally, note that Fig. 9(a) describes the results reached with SACK-TCP which is more robust to multiple losses than Reno-TCP which we have used. Actually, the authors of [5] report that with Reno-TCP the fairness results of RAP are slightly worse. With LDA+, the TCP connections receive on the average a bandwidth share of around 5.4 kbytes/s compared to 4.5 kbytes/s for the LDA+ flows. Hence, LDA+ is under this configuration more conservative than TCP.

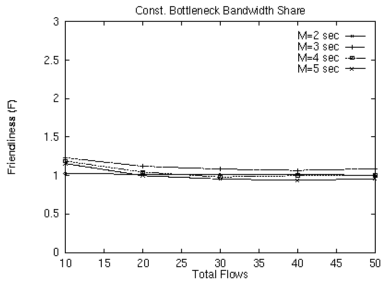
B. *Comparison of LDA+ with an Equation-Based TCP-Friendly Adaptation Scheme*

Using the analytical model of TCP in Eqn. 1 Padhye et al. present in [23] a scheme called TCP-friendly rate control protocol (TFRCP). With TFRCP the sender estimates the round trip delay and losses based on the receiver's acknowledgments. In the case of losses, the sender restricts its transmission rate to the equivalent TCP rate calculated using Eqn. 1 otherwise the transmission rate is doubled. The sender updates its rate periodically in intervals of M. In [23] different results were reported for adaptation intervals between two and five seconds.

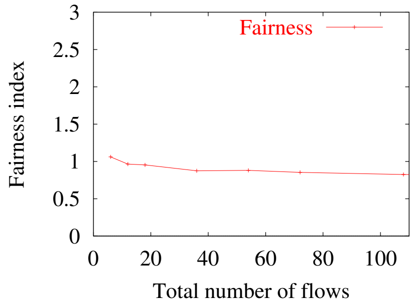
Similar to our previous investigations a simple test topology was used in [23] with N TCP connections sharing a single bottleneck with N UDP flows. The round trip delay (τ) is set to 0.04 seconds and the packet size was set to 100 bytes. The router uses FIFO buffer management with the buffer length set to ($4 \times R \times \tau$).

Fig. 10 depicts the fairness results of a test setting with the bottleneck bandwidth set to ($2 \times N \times 40$ kb/s) for a varying number of flows. The fairness index is determined as the average goodput of the adaptive flows to the average goodput of the TCP connections. Both LDA+ and TFRCP achieve in this case a fairness index of around one indicating a high degree of TCP-friendliness. With TFRCP this is especially evident when an adaptation interval (M) of two seconds is used. For higher values of M TFRCP is especially for the case of fewer competing flows more aggressive than TCP.

Fig. 11 depicts the fairness results for the case of a constant bottleneck bandwidth of 1.5 Mb/s. Depending on the chosen adaptation interval, TFRCP achieves different fairness values ranging from 0.7 to around one. For higher numbers of competing flows, LDA+ achieves a fairness index of around one as well indicating its TCP-friendliness in cases of higher network loads. However, for the case of a lower number of competing flows a fairness factor higher than one is achieved. This is especially evident for the case of six LDA+ flows competing with six TCP connections. In this case, a fairness index of 2.3 is achieved indicating that the LDA+ flows receive on the average twice as much bandwidth as the TCP connections. However, this is not caused by the adaptation algorithm itself but by the way the loss information are transported in RTP. The receiver reports use an eight bit field for the loss information and hence the smallest loss information that can be indicated is approximately 0.004. However, under the used topology with 0.04 seconds round trip delay and a bottleneck of 1.5 Mb/s to be divided among 12 competing flows, using Eqn. 1 indicates that the average loss rate each flow needs to suffer in order for it to get a bandwidth share of ($\frac{1.5}{12} = 0.125$ Mb/s) is around 0.002. In this case, the RTP receivers round the loss value down to 0 and the RTP flows can increase their transmission rate in situations where they are supposed to reduce it. Fig. 11(b) depicts the fairness results achieved when using a 32 bit field in the receiver reports for loss indication instead of eight bits. In this case, we can observe that LDA+ is rather conservative and achieves for the case of smaller numbers of competing flows a fairness index of less than one. For the case of higher numbers of competing flows the behavior of LDA+ is identical for both loss representations.



(a) TFRCP



(b) LDA+

Fig. 10. Fairness index with TFRCP and LDA+ for a constant bandwidth share per flow

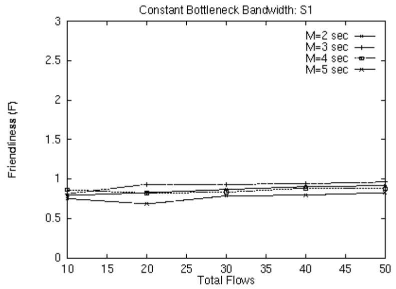
VII. Summary and Future Work

In this paper, we presented an algorithm called LDA+ for adapting the transmission rate of multimedia senders in a TCP-friendly manner. LDA+ relies on RTP and does not require the introduction of a new control protocol for establishing a closed feedback loop between the sender and receiver.

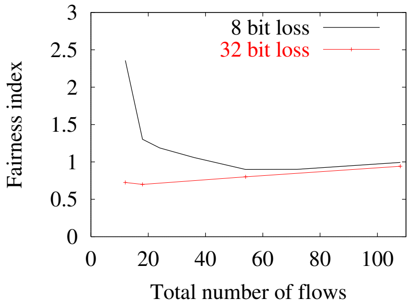
The measurement and simulation results presented in this paper suggest that while RTCP generates feedback messages on a much slower scale than the control protocols used for RAP [5] or TFRCP [7] LDA+ still achieves similar fairness results over a wide range of parameters.

The here presented results constitute only a subset of the test results collected for LDA+. More extensive tests documented in [17] confirm the TCP-friendliness of LDA+ and its efficiency in achieving high network utilization and avoiding losses.

While LDA+ was not designed as a multicast congestion control scheme, basing the adaptation actions on infrequent control messages makes it suitable to be used within a multicast congestion control framework. In [10] we present such a framework



(a) TFRCP



(b) LDA+

Fig. 11. Fairness index with TFRCP and LDA+ for a constant bottleneck bandwidth

called MLDA in which the receivers use LDA+ to estimate their TCP-friendly bandwidth share and inform the sender about this share. The sender can then use this information to optimize its transmission behavior. While with MLDA the receivers do the rate estimation and the control protocol is slightly different than the current specification of RTCP the increase and decrease behavior of LDA+ is still preserved.

VIII. Acknowledgments

The comments of Henning Sanneck and Henning Schulzrinne are gratefully acknowledged and were the basis for various aspects of this work. The RTP/RTCP simulation models were mainly implemented by Timur Friedman and improved by Frank Emanuel. Special thanks to Andrey Vasilyev and Martin Reisslein for providing the measurement end points.

References

[1] K. Thompson, G. J. Miller, and R. Wilder, “Wide-area Internet traffic patterns and characteristics,” *IEEE Network*, vol. 11, no. 6, pp. –, November/December 1997.

[2] Sally Floyd and Fall Kevin, “Router mechanisms to support end-to-end congestion control,” Tech. Rep., LBL-Berkeley, Feb. 1997.

[3] Dorgham Sisalem and Henning Schulzrinne, “The loss-delay based adjustment algorithm: A TCP-friendly adaptation scheme,” in *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Cambridge, England, July 1998.

[4] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, “Modeling TCP throughput: A simple model and its empirical validation,” in *ACM SIGCOMM ’98*, Vancouver, Oct 1998.

[5] Reza Rejaie, Mark Handley, and Deborah Estrin, “An end-to-end rate-based congestion control mechanism for realtime streams in the Internet,” in *Infocom’99*, New York, March 1999, IEEE.

[6] Stephen Jacobs and Alexandros Eleftheriadis, “Providing video services over networks without quality of service guarantees,” in *RTMW’96*, Sophia Antipolis, France, Oct. 1996.

[7] J. Padhye, J. Kurose, D. Towsley, and R. Koodli, “A model based TCP-friendly rate control protocol,” in *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Basking Ridge, NJ, June 1999.

[8] Lorenzo Vicisano, Luigi Rizzo, and Jon Crowcroft, “TCP-like congestion control for layered multicast data transfer,” in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, San Francisco, USA, Mar. 1998.

[9] Thierry Turletti, Sacha Fosse Prisis, and Jean-Chrysostome Bolot, “Experiments with a layered transmission scheme over the Internet,” Rapport de recherche 3296, INRIA, Nov. 1997.

[10] Dorgham Sisalem and Adam Wolisz, “MLDA: A TCP-friendly congestion control framework for heterogeneous multicast environments,” in *Eighth International Workshop on Quality of Service (IWQoS 2000)*, Pittsburgh, PA, June 2000.

[11] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: a transport protocol for real-time applications,” Tech. Rep. RFC 1889, Internet Engineering Task Force, Jan. 1996.

[12] Jean-Chrysostome Bolot, “End-to-end packet delay and loss behavior in the Internet,” in *SIGCOMM Symposium on Communications Architectures and Protocols*, Deepinder P. Sidhu, Ed., San Francisco, California, Sept. 1993, ACM, pp. 289–298, also in *Computer Communication Review* 23 (4), Oct. 1992.

[13] Kevin Lai and Mary Baker, “Measuring bandwidth,” in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, New York, USA, Mar. 1999.

[14] Vern Paxon, *Measurements and Analysis of End-to-End Internet Dynamics*, Ph.D. thesis, Lawrence Berkeley National Laboratory, University of California, Berkeley, California, Apr. 1997.

[15] W. Richard Stevens, *TCP/IP illustrated: the protocols*, vol. 1, Addison-Wesley, Reading, Massachusetts, 1994.

[16] Kihong Park, Gitae Kim, and Mark Corvella, “On the relationship between file sizes, transport protocols and self-similar network traffic,” in *International Conference on Network Protocols (ICNP)*, Columbus, Ohio, Oct 1996.

[17] Dorgham Sisalem, “TCP-friendly congestion control for multimedia communication,” Technical report, GMD, Fokus, Germany, Apr. 2000.

[18] S. Knowles, “IESG advice from experience with path MTU discovery,” Tech. Rep. RFC 1435, Internet Engineering Task Force, Mar. 1993.

[19] Robert L. Carter and Mark E. Crovella, “Measuring bottleneck link speed in packet-switched networks,” Tech. Rep. BU-CS-96006, Computer Science Departement, Boston University, Mar. 1996.

[20] Allen B. Downey, “Using pathchar to estimate internet link charachteristics,” in *Special Interest Group on Data Communication (SIGCOMM’99)*, Cambridge, USA, Aug. 1999.

[21] Lixia Zhang, Scott Shenker, and David D. Clark, “Observations on the dynamics of a congestion control algorithm: The effects of two-way trffic,” in *SIGCOMM Symposium on Communications Architectures and Protocols*, Z rich, Switzerland, Sept. 1991, ACM, pp. 133–147, also in *Computer Communication Review* 21 (4), Sep. 1991.

[22] Lampros Kalampoukas and Anujan Varma, “Two-way TCP traffic over ATM: Effects and analysis,” in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, Kobe, Mar. 1997.

[23] J. Padhye, J. Kurose, D. Towsley, and R. Koodli, “A TCP-friendly rate adjustment protocol for continuous media flows over best effort networks,” Technical report, University of Massachusetts, Amherst, Massachusetts, Oct. 1998.