In Proc. of European Wireless 2004, Barcelona, Spain, February 2004.

TCP-Friendly Congestion Control over Wireless Networks

Vicente E. Mujica V., Dorgham Sisalem
FhG FOKUS, Berlin,Germany
{mujica,sisalem}@fokus.fhg.de

Radu Popescu-Zeletin and Adam Wolisz
FhG FOKUS/University of Berlin, Germany
zeletin@fokus.fhg.de / awo@ieee.org

Abstract: **In this paper, we present an end-to-end adaptation scheme, called the wireless loss-delay based adaptation algorithm (WLDA+). WLDA+ adapts the transmission behaviour of multimedia senders in accordance with the network congestion state in wireless environments. WLDA+ is based on the loss-delay based adaptation scheme [1] which adjusts the transmission behaviour of the senders in a manner similar to TCP connections suffering from equal losses and delays. To take the specific characteristics of wireless links into account, WLDA+ incorporates error differentiation schemes to detect the loss nature in the wireless channel. The performance of WLDA+ is then investigated by simulating the behaviour of this algorithm under different network topologies.**

1 Introduction

While congestion controlled TCP connections carrying time insensitive FTP or WWW traffic still constitute the major share of the Internet traffic today [2], 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. 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 [3]. Various studies and papers such as [4, 5, 6, 7] already describe TCP-friendly congestion control schemes to be used for real-time communication. However, those schemes are optimized for fixed networks and are generally not easily applicable to wireless environments. That is, in a fixed network a packet loss can in general be considered as an indication of overload and congestion situation. In a wireless environment losses could also occur due to bad channel characteristics or interference for example. Based on these losses, congestion control mechanisms would lead to a reduction of the transmission rate and would only decrease the link utilization unnecessarily.

In this paper, we describe a new scheme called the wireless loss-delay based adaptation algorithm (WLDA+), that adapts the transmission rate of UDP-based multimedia flows to the congestion situation in wireless networks in a TCP-friendly manner. This scheme is based on the loss-delay based adaptation algorithm (LDA+). Basically, LDA+ regulates the transmission rate of a sender according to end-to-end feedback information about losses, delays and the bandwidth capacity measured by the receiver. WLDA+ further utilizes loss differentiation schemes to distinguish between losses caused by overload situations (congestion losses) and those occurring due to temporary interference of channel disturbances (wireless losses). With no observed congestion losses, the sender can increase its transmission rate additively otherwise it needs to reduce it multiplicatively. For differentiation, we investigate the usage of two schemes namely spike-trains scheme [8] and the Inter-arrival scheme [9].

In the context of wireless communication, TCP-Friendly indicates that TCP and UDP flows have similar bandwidth shares under similar delay and congestion. In Sec. 2 we take a brief look at some of the available TCP-friendly algorithms and loss differentiation schemes in the literature. WLDA+ is described in sec. 3. Depending on the use of loss differentiation schemes we distinguish two WLDA+ versions. Finally, the performance of WLDA+ and these loss differentiation schemes is evaluated using different simulation models in Sec. 4.

2 Background and Related Works

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 [7] 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.

Congestion schemes designed in a similar manner to TCP such as RAP or the TCP Emulation at Receiver (TEAR) [6] will not be considered here. Works done in [10][11] for example show the poor performance of TCP in wireless environments. In such environments, losses occur in clusters resulting in losses of multiple packets from a single TCP sender window. When this happens, TCP performs poorly, unnecessarily retransmitting packets and thus reducing the throughput.

Padhye et al. [12] present an analytical model for the average bandwidth share of a TCP connection (r_{TCP}). 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.

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

Using this model Padhye et al. [13] 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. This equation was used by Floyd et al. to develop the TCP-Friendly Rate Control (TFRC) congestion control mechanism [5].

2.1 The Enhanced Loss-Delay Based Adaptation Algorithm

In contrast to other adaptation schemes such as [13, 7] that introduce a new protocol for establishing the flow of feedback messages from the receiver to the sender, the enhanced loss-delay based adaptation algorithm (LDA+) [1] relies on the real time transport protocol (RTP) [14].

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 (RTCP), each receiver periodically sends control reports to the sender containing information about losses and delay noticed in the network. The RTCP traffic is scaled with the data traffic so that it makes up a certain percentage of the data rate (usually 5%) with a minimum interval of 5 seconds between sending two RTCP messages.

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. 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 faster increase their transmission rates than competing flows with higher shares, A is determined in dependence of the bandwidth share ($r_{\text{m-1}}$) the sender is currently consuming relative to the bottleneck bandwidth (R_i) of the path connecting the sender to the receiver. A would evolve as follows:

$$A_{\text{add}_\text{m}} = A_{\text{m-1}} + \left(1 - \frac{r_{\text{m-1}}}{R}\right) \times A_{\text{m-1}} \quad (2)$$

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

$$A_{\text{exp}_\text{m}} = \left(1 - \exp^{-(1-\frac{r_{m-1}}{R})}\right) \times r_{m-1} \quad (3)$$

Finally, an RTP flow should not increase its bandwidth share faster than a TCP connection sharing the same link. With an average value of 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

$$P = \sum_{q=0}^{T/\tau} q = \frac{(\frac{T}{\tau}+1) \times \frac{T}{\tau}}{2} \quad (4)$$

with the window size being increased by one packet each round trip delay. With a packet size of M and averaged over T, the RTP receiver should maximally increase its estimation of its bandwidth share by

$$A_{\text{TCP}_\text{m}} = M \times \frac{P}{T} \rightarrow M \times \frac{\frac{T}{\tau}+1}{2 \times \tau} \quad (5)$$

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

$$A_{\text{m}} = \min(A_{\text{add}_\text{m}}, A_{\text{exp}_\text{m}}, A_{\text{TCP}_\text{m}}) \quad (6)$$

Finally, the receiver determines r_{m} as

$$r_{\text{m}} = r_{\text{m-1}} + A_m \quad (7)$$

Loss situation: For the case of losses, this model suggests that TCP adjusts its transmission rate inversely proportional to the square root of the losses. Thus, if losses

(l) were indicated in the RTCP messages then the transmission rate (r_{m}) is reduced to:

$$r_{\text{m}} = \max(r_{\text{m-1}} \times (1 - \sqrt{l}), r_{\text{TCP}}) \quad (8)$$

with r_{TCP} as the rate calculated using Eqn. 1. In case the current transmission rate is already lower than r_{TCP} the sender is allowed to further increase its transmission rate up to r_{TCP}.

The work done in [1] as well as the extensive simulations and measurements reported in [4, 15] suggest LDA+ to be efficient in terms of bandwidth utilization and loss reduction as well as being friendly to competing TCP-traffic. A sample of those simulations describing the TCP-friendliness aspects of LDA+ are presented in section 4.

2.2 Error Differentiation Schemes

While in fixed networks losses usually occur due to overload situations, in wireless networks packets might be dropped due to various reason such as interferences, handoffs, fading channels and so forth. Thereby in the context of a wireless environments, the throughput can be reduced unnecessarily if wireless tranmission errors are not previously discriminated. More precisely, congestion control actions should only be triggered in wireless networks when a packet loss is caused by congestion. Based on the TCP protocol, TCP Westwood (TCPW) [16] implements a window congestion control algorithm taking into account a differentiation scheme for mixed wired/wireless networks. To manage the efficiency/friendly tradeoffs in TCPW, Wang et al. propose in [17] the estimation differentiation technique called Combined Rate and Bandwidth (CRB) estimation scheme. CRB infers the predominant cause of packet loss (buffer congestion or random error) using the information obtained from ACKs received at the sender. More precisely, CRB is a hybrid method that combines the both TCPW bandwidth estimators. The Bandwidth Sampling Estimator (BSE) considers each ACK pair separately to obtain a bandwidth sample, filters the samples into a low pass filter and returns as result the bandwidth share that the TCP sender is estimated to be getting from the network. The other estimator called Rate Estimator (RE), monitors the amount of data acknowledged during a fixed interval of time T, then feeds such samples into an appropriate low-pass filter to get the estimated rate. To take into account differentiation losses scheme over UDP protocol, we propose the study of the spike-trains scheme and the Inter-arrival scheme. These two schemes can be adapted using TCP as well as UDP, due to the ability to be implemented as generic mechanism in end-terminals without any modification of the mentioned protocols.

2.2.1 Inter-arrival scheme

This scheme proposed by Biaz et al. [9], uses the inter-arrival time between consecutive packets for differentiating between losses caused by network congestion and losses due to wireless specific reasons. More precisely, this scheme discriminates wireless losses (l_{WRLS}) and congestion losses (l_{CONG}) using the minimum inter-arrival time (T_{min}) between two consecutive packets. Let T_g denote the time between the arrival of two packets at the receiver side and n as the number of packet losses during this time interval, then:

$$\begin{array}{c} If\ ((n+1)T_{min} \leq T_g \leq (n+2)T_{min}) \\ then\ l_{\text{WRLS}} = n \quad else\ l_{\text{CONG}} = n \end{array} \quad (9)$$

Note that the scheme assumes the following conditions: -Only the last link on the path is wireless.
-The wireless link is the bottleneck for the connection.
-The sender performs a bulk data transfer.

2.2.2 Spike-train Scheme

The spike-train scheme was developed by Tobe et al. in [8]. This scheme uses the Relative One-way Trip Time (ROTT) as congestion signal in the network. ROTT is the time that a packet needs to be transported from the sender to the receiver. The spike-train scheme derives its name from the fact that plotting ROTT vs. time tends to show spikes during periods of congestion. By continuously measuring the ROTT value, it is possible to use this value as measurement of overloading of the network. For example, a packet with a large ROTT value would possibly indicate congestion in a link.

To classify losses, the WR_O-LDA+ algorithm defines two major thresholds, $B_{\text{spike}_{\text{start}}}$ and $B_{\text{spike}_{\text{end}}}$. Tobe et al. in [8] recommend to fix the values of these thresholds using the following algorithm with $K = 1/2$ and $B = 1/3$.

$$\begin{array}{l} B_{\text{spike}_{\text{start}}} = ROTT_{\text{min}} + K(ROTT_{\text{max}} - ROTT_{\text{min}}) \\ B_{\text{spike}_{\text{end}}} = ROTT_{\text{min}} + B(ROTT_{\text{max}} - ROTT_{\text{min}}) \end{array} \quad (10)$$

Where $ROTT_{\text{max}}$ and $ROTT_{\text{min}}$ represent the maximum and minimum ROTT in the network.

3 Wireless Loss-Delay Based Adjustment Algorithm (WLDA+)

The LDA+ algorithm was designed and optimized for fixed environments. That is, packet losses are interpreted as network congestion. To reduce these congestion losses, senders would reduce their transmission rate and thereby reduce the network load situations. However, in wireless environments losses could also occur due to handoffs, fading channels, noise and transient random errors. As those wireless losses are not related to the network load situation, reducing the transmission rate after such losses would not improve the loss situation and would only reduce the bandwidth utilization level of the wireless links.

To deploy LDA+ in wireless environments, in this paper we extended LDA+ with mechanisms for differentiating between congestion and wireless losses. The extended algorithm called Wireless Loss-delay based adjustment algorithm (WLDA+) that we present here comes in two versions, Wireless-Interarrival-LDA+ and Wireless-ROTT-LDA+, respectively.

Basically, WLDA+ takes into account only the losses caused by congestion. With this purpose of the parameter l in eqn. 8 is changed to l_{cong}, which represents only the congestion losses detected by each scheme. Both schemes determine a loss based on the sequence number of two consecutive received packets. If the loss is caused by congestion, the parameter l_{cong} in the receiver is incremented. On the other hand, in the presence of wireless losses the l_{cong} parameter is kept to the same value. The loss differentiation schemes are integrated with LDA+ by extending the eqn. 8 with the following rules:

Wireless-Interarrival-LDA+: Based on the inter-arrival scheme, the Wireless-Interarrival-LDA+ (WI_N-LDA+) uses the time between consecutive RTP packets for differentiating between congestion and wireless losses. In the configuration of Wireless-Interarrival-LDA+, the minimum inter-arrival time (T_{min}) is detected during the observation time T_{obs}. The observation time is defined as the period of time where receivers monitor constantly the inter-arrival time without the application of any Wireless-Interarrival-LDA+ scheme only to detect the minimum value. After T_{obs} time, Wireless-Interarrival-LDA+ uses the determined T_{min}. Ts_i is the RTP timestamp from packet i and Tr_i is the arrival time, respectively. For two consecutive packets i and $i-1$ the interarrival time (T_g) is measured as follows:

$$T_g = Tr_i - Tr_{i-1} - (Ts_i - Ts_{i-1}) \tag{11}$$

Wireless-ROTT-LDA+: Wireless-ROTT-LDA+ (WR_O-LDA+) is an extension of the original spike-train scheme. Basically, we create a Markov chain machine that alternates between two states, Congestion(C) and Underload (U), respectively. The events that determine the transition between these two states, are defined with the mentioned thresholds from equation 10. Using the current ROTT measured by receivers ($ROTT_{\text{c}}$), the current state (S_{c}) is determined with the following conditions:

$$\begin{array}{l} if(ROTT_{\text{max}} \geq ROTT_{\text{c}} \geq B_{\text{spike}_{\text{start}}})\ then\ S_{\text{c}} = C \\ if(B_{\text{spike}_{\text{start}}} > ROTT_{\text{c}} > B_{\text{spike}_{\text{end}}})\ then\ S_{\text{c}} = S_{\text{p}} \\ if(B_{\text{spike}_{\text{end}}} \geq ROTT_{\text{c}} \geq ROTT_{\text{min}})\ then\ S_{\text{c}} = U \end{array} \tag{12}$$

The network is overloaded, when the measured $ROTT_{\text{c}}$ is inside the ”Congestion State” interval. So that the major cause for packets dropped is due to congestion in the network. On the other hand, wireless losses are represented in the “Underload State”, where it is assumed that the congestion level of the network is low and packet drop is due to wireless transmission errors. In order to be consequent with the increasing or decreasing behavior of the ROTT parameter, the connection maintains the previous state S_{p} when the $ROTT_{\text{c}}$ is between the $B_{\text{spike}_{\text{start}}}$ and $B_{\text{spike}_{\text{end}}}$ values. When receivers detect a loss, the current state of the receiver determine the kind of losses (wireless or congestion losses).

The analysis of one-way delay as alternative to detect congestion in a network, should study the difference caused by the skew and offset between the sender‘s and receiver‘s clocks. To obtain a precise one-way delay measurement, Tobe et al. proposed in [18] a scheme to remove the skew and offset in the receiver site. It is important to keep in mind the following observation for the results: In the reality, wireless error transmissions appear in the channel independent of the load in the wired links. This means that the wireless losses can occur when the wired link is overload or underload. With the WR_O-LDA+ approach, wireless error losses are considered as congestion losses during the congestion state.

4 Performance Evaluation of WLDA+

4.1 Simulation Topology and Environment

The simulation tool network simulator (ns) version ns-2.1b9 [19] was used to implement a dumb-bell topology such as illustrated in fig. 1.

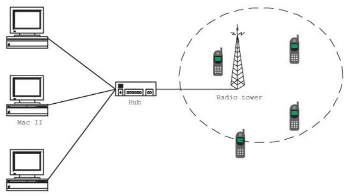


Figure 1: Simulation Topology

With the purpose of testing both versions of WLDA+ (WR_O-LDA+ and WI_N-LDA+), we conducted several simulations with different scenarios. Each scenario was simulated five times and the presented values in section 4.2 are the average of the results reached in the different runs. The first 100 seconds (T_{obs}) have been ignored to take into account the effect of a transient period. The packet size was held constant to 1000 bytes which is a size often used in video conferencing applications.

The link between the router and Base Station(BS) has a capacity of 10 Mb/s and a propagation delay of τ. The gateway (router) and BS were modeled using a RED router to ensure that all flows receive the same loss rate and avoid synchronization among them. The bottlenecked router and BS can introduce a maximal additional delay ($\dot{\tau}_{\text{router}}$) of 0.1 seconds due to the buffering of the data. The Base Station (BS) serves only as a distributer of the data to the end mobile devices and introduces losses with a certain error probability ($\dot{P}_{\text{e}}$). Senders are located in the wired hosts and start with a transmission rate of 10 kb/s.

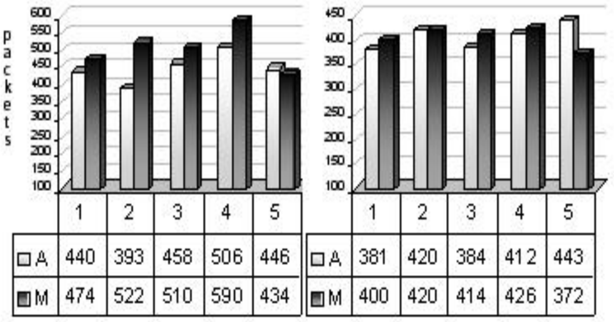
The wireless network is supposed to operate in the 2.4 GHz frequency band using the original 802.11 standard. The “aggregate bandwidth” in the wireless channel is set to 11 Mbps. Nguyen et al. [20] present a two-state Markov wireless error model based on collected WaveLAN error traces. Loss behavior of this WaveLAN reveals a maximal error rate average of 3 percent based on a distance below 100 meters. Results taking into account a packet size of 1000 bytes, show an error rate around 0.5%. To evaluate different scenarios, the error probability $\dot{P}_e$ in the channel is changed between 0.5% and 5%. These parameters are presented as low and high channel error rates (corresponding to packet loss rates of 0.5% and 5% respectively). Finally, the number of senders is varied from 20 to 100.

4.2 Simulation results

4.2.1 Accuracy of the Error Differentiation Schemes

As a first step we evaluate the accuracy of both loss differentiation schemes. The topology of fig. 1 was configured with 20 senders and a propagation delay τ of 150 msec. The error probability is set to 0.5%. I.e., 0.5% of the packets carried over the wireless link can be lost due to inference, disturbance or other wireless specific reasons. The number of actual wireless transmission errors (A) was compared with the number of wireless losses, which are measured by schemes (M).

Based on the results depicted in fig. 2, the accuracy of discrimination (Ac) is calculated as the ratio of the number of wireless transmission errors correctly identified ("measurement") over the total number of packets really dropped in the wireless channel ("actual"). For example, if 100 wireless losses occurred and WI_N-LDA+ distinguished only 95 losses. Then, the accuracy in this case is 0.95.



(a)WR_O-LDA+ (b)WI_N-LDA+

Figure 2: Accuracy of WR_O-LDA+(a) and WI_N-LDA+(b)

Table 1 illustrates the accuracy of both WR_O-LDA+ and WI_N-LDA+. If the accuracy (A_c) is close to one, the algorithm has detected almost all the wireless transmission errors in the channel.The WI_N-LDA+ algorithm shows a better precision to detect correctly the nature of the packet loss. Biaz et al. in [9] have demonstrated the improvement in TCP performance using the accuracy metric parameter. In the case of WI_N-LDA+ accuracies, results depict an efficiency for UDP flows, where the accuracy oscillates close to 1. In addition, descriptive statistical have been used to verify the precision and exactitude for both schemes. The determined wireless transmission errors have been expressed as the sample mean(M_x) plus some uncertainty(S_x). For the case of WI_N-LDA+, the total number of wireless error packets for "actual" values were 408 ± 26 and "measurement" values were 407 ± 21. On the other hand, WR_O-LDA+ has detected 449 ± 40 packets for the "actual" and 506 ± 58 packets for the "measurement", respectively. The above results demonstrate a good accuracy for both schemes. However, the WI_N-LDA+'s descriptive statistical shows more precision than WR_O-LDA+.

Table 1: Accuracy for WR_O-LDA+ and WI_N-LDA+

$Runs$	WI_N-LDA+	WR_O-LDA+
1	1.05	1.07
2	1	1.33
3	1.08	1.11
4	1.03	1.17
5	0.84	0.97

4.2.2 Evaluation of WR_O-LDA+ and WI_N-LDA+

Using the configuration of the sec. 4.2.1, fig. 3 depicts the tranmission rate for WLDA+ versions and LDA+ respectively. Based on the transmission rates from WI_N-LDA+ and WR_O-LDA+, it is important to note that the two schemes converge around 200 KBps. However, using LDA+ without any error differentiation algorithm the transmission rate is close to 175 KBps.

In addition, the two WLDA+ versions show values of fraction lost, around 0,42% by WI_N-LDA+ and 0,43% by WR_O-LDA+. In the case of LDA+ the total fraction lost was 0,56%. The effective discrimination of the wireless losses decreases the total fraction lost, thereby the transmission rate is optimally increased such as illustrated results in fig. 3.

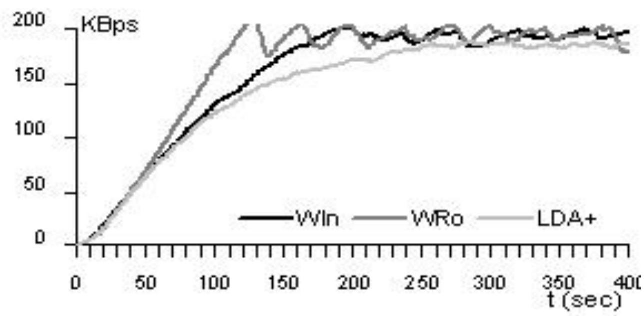


Figure 3: Transmission Rate Comparison for 20 senders

4.2.3 Evaluation of the effects of different error rates in the wireless channel:

Scenarios with different loss probabilities (P_e) in the wireless channel were considered using 0.5% and 5%, respectively. The number of senders was set to 50. Using $P_e = 0.5\%$ fig. 4 depicts the number of wireless (WRLS) and congestion (CONG) losses, which are discriminated by WR_O-LDA+ during 400 seconds.

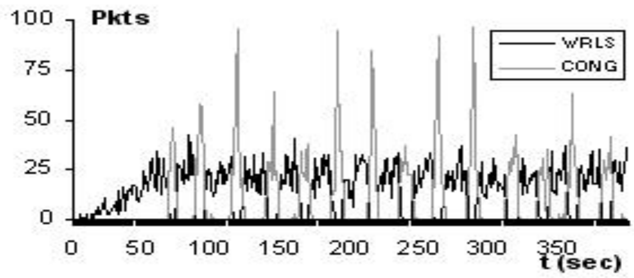


Figure 4: Losses discriminate by WR_O-LDA+

WR_O-LDA+ detects wireless losses only during the "underload state" and spikes represent packets dropped due to congestion. With this scheme is impossible to detect any wireless error packets when the state machine is in the congestion state. On the other hand, WI_N-LDA+ can continously distinguish wireless and congestion losses such as described fig. 5.

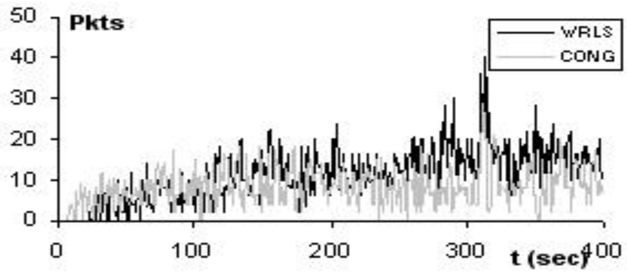


Figure 5: Losses discriminate by WI_N-LDA+

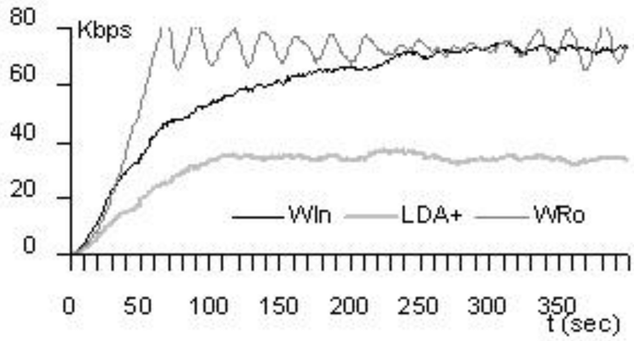


Figure 6: Transmission Rate Comparison using P_e=5%

Fig. 6 shows the transmission rate comparisons taking into account an error rate $P_e = 5\%$. WI_N-LDA+ manages a conservative transmission rate with respect to the WR_O-LDA+ scheme. However, both schemes converge to 75 KBps after 230 seconds. LDA+ achieves a transmission rate close of only 40 KBps. This figure (Fig. 6) shows very clear the advantage in the usage of an error differentiation algorithm. Applying WLDA+ the transmission rate can be improve upto 100% considering that with only LDA+ the transmission rate is around 40 KBps.

Taking into account $P_e = 0.5\%$ table 2 shows the descriptive statistic for wireless losses, which are determined by the WLDA+ versions. "Measurement" expressions in the form $M_x \pm S_x$ tend to be similar to the "actual" values. The precision and accuracy of both schemes has been again demonstrated for the case of topologies with more senders (50 senders). Finally, transmission rates are improved as is depicted in fig. 6, using the implementation of an error differentiation algorithm (WI_N-LDA+ and WR_O-LDA+). More precisely, the advantage is more notable when the error probability in the wireless channel is relatively high ($P_e \sim 5\%$).

Table 2: Statistics of the error differentiation schemes

	P_e=0.5%	
$M_x \pm S_x$	WI_N-**LDA+**	WR_O-**LDA+**
Actual	456 ± 18	375 ± 24
Measurement	446 ± 56	393 ± 49

4.3 Scalability of WLDA+

To test the scalability of the WLDA+ implementation, scenarios with 20 and 50 senders have been simulated in the last section. In this section we evaluate 100 senders with a error rate in the wireless channel of Pe=0.5%. Fig. 7 compares the transmission rate for WI_N-LDA+, WR_O-LDA+ and LDA+, respectively. Transmission rates fluctuate abruptly around 40 KBps using both WLDA+ versions. Thereby, 100 sender overload the base station buffer. On the one hand, it translates into a large collision probability which delays the access to the radio channel. On the other hand, when the channel is shared by a large number of sources, the channel capacity perceived by individual sources is smaller.

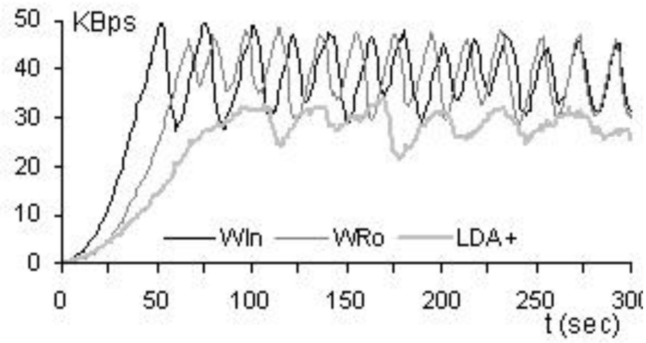


Figure 7: Transmission Rate Comparison for 100 senders

For this simulation we detect that the number of packets dropped due to congestion increases considerably. Using only one base station it is not enough to guarantee a coverage area of 100 senders. However, results in fig. 7 show that the WLDA+ versions can maintain a transmission rate in overloaded networks. Applying only LDA+ the transmission rate is around 25 KBps.

5 Conclusions

The simulation results show a clear improvement of the performance of the LDA+ congestion control algorithm when an error differentiation scheme is integrated. WI_N-LDA+ and WR_O-LDA+ presented a good accuracy taking into account low error rate probability in the wireless channel, i.e. below 0.5%. The performance of WLDA+ is partially decreased using probability error of 5% over high congestion networks, which was considered as high error rate in the wireless channel. Using WI_N-LDA+ and WR_O-LDA+, transmission rates show an improvement of 100% with respect to LDA+ results. Due to the ability to discriminate a great percentage of wireless tranmission errors, WI_N-LDA+ and WR_O-LDA+ present good properties to be implemented as end-to-end congestion control schemes for wireless environments.

Comparing the performance between both schemes, WI_N-LDA+ can smoothly manage the tranmission rate. On the other hand, WR_O-LDA+ throughput presented an oscillating behavior. It is caused by wireless losses not being discriminated when the network is in the congestion state.

The performance of the WR_O-LDA+ decreases especially in congested networks. Because with more senders the network load increases and the length of the queue is constantly high leading to measuring large ROTTs frequently. Thereby, there are less variations in the values of ROTT and wireless losses are classified wrongly as congestion losses.

References

[1] D. Sisalem and H. Schulzrinne, "The loss-delay based adjustment algorithm: A TCP-friendly adaptation scheme," in *nossdav*, England, July 1998.

[2] K. Thompson, G. J. Miller, and R. Wilder, "Wide-area Internet traffic patterns and characteristics," *ieeenet*, vol. 11, no. 6, pp. –, December 1997.

[3] S. Floyd and F Kevin, "Router mechanisms to support end-to-end congestion control," Tech. Rep., LBL-Berkeley, Feb. 1997.

[4] D. Sisalem, *TCP-Friendly Congestion Control for Multimedia Communication in the Internet*, GMD-FOKUS, Bonn, Germany, Oct. 2000, PhD Thesis.

[5] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *sigcomm*, Sweden, Aug. 2000.

[6] I. Rhee, V. Ozdemir, and Y. Yi, "TEAR: tcp emulation at receivers - flow control for multimedia streaming," Tech. Rep., Dpto. of Computer Science, NCSU-USA, Apr. 2000.

[7] R. Rejaie, M. Handley, and D. Estrin, "RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the internet," in *INFOCOM (3)*, 1999, pp. 1337–1345.

[8] Y. Tobe, Y Tamura, A. Molano, S. Ghosh, and H. Tokuda, "Achieving moderate fairness for udp flows by path-status classification," in *25th Annual IEEE Conference on Local Computer Networks (LCN'00)*, Tampa, Florida, Nov. 2000, IEEE, p. 252.

[9] S. Biaz and N Vaidya, "Discriminating congestion losses from wireless losses using inter-arrival times at the receiver," *IEEE Symposium ASSET'99, Richardson, TX, USA*, 1999.

[10] G. Holland and N. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," in *Mobile Computing and Networking*, 1999, pp. 219–230.

[11] Z. Fu, X. Meng, and S. Lu, "How bad tcp can perform in mobile ad hoc networks," in *Seventh International Symposium on Computers and Communications (ISCC'02)*, Italy, July 2002, IEEE, p. 296.

[12] 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.

[13] J. Padhye, J. Kurose, D. Towsley, and R. Koodli, "A model based TCP-friendly rate control protocol," in *nossdav*, Basking Ridge, NJ, June 1999.

[14] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a transport protocol for real-time applications," Tech. Rep. RFC 1889, IETF, Jan. 1996.

[15] D. Sisalem and A. Wolisz, "Towards TCP-friendly adaptive multimedia applications based on RTP," in *Fourth IEEE Symposium on Computers and Communications (ISCC'99)*, Egypt, July 1999, IEEE.

[16] C. Casetti, M. Gerla, S. Mascolo, M.Y. Sansadidi, and R Wang, "Tcp westwood: End-to-end congestion control for wired/wireless networks," *In Wireless Networks Journal*, , no. 8, oct 2002.

[17] R. Wang, M. Valla, M.Y. Sanadidi, B.K.F Ng, and M. Gerla, "Efficiency/friendliness tradeoffs in tcp westwood," *In Proc. of the Seventh IEEE Symposium on Computers and Communications*, jul 2002.

[18] Y Tobe, Y. Tamura, and H. Aida, "Detection of change in one-way for analyzing the path status.," in *Proc. of the First Passive and Active Measurement Workshop*, CamHamilton, New Zealand, Apr. 2000.

[19] L. Yamamoto and G Leduc, "Network simulator," Apr. 1998.

[20] G.T. Nguyen, R.H. Katz, B. Noble, and M. Satyanarayanan, "A trace-based approach for modeling wireless channel behavior," in *Winter Simulation Conference*, Dec. 1996, pp. 597–604.