INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES

PUBLICATION INTERNE
N° 1792

IMPROVEMENT OF MULTIMEDIA STREAMING USING ESTIMATION OF WIRELESS LOSSES

KAMAL DEEP SINGH, DAVID ROS, LAURENT TOUTAIN AND CÉSAR VIHO

ISSN 1166-8687

IRISA

CAMPUS UNIVERSITAIRE DE BEAULIEU - 35042 RENNES CEDEX - FRANCE

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES
Campus de Beaulieu – 35042 Rennes Cedex – France
Tél. : (33) 02 99 84 71 00 – Fax : (33) 02 99 84 71 71
http://www.irisa.fr

Improvement of Multimedia Streaming using Estimation of Wireless losses

Kamal Deep Singh,* David Ros,** Laurent Toutain*** and César Viho****

Systèmes communicants
Projet ARMOR

Publication interne n ° 1792 — Mars 2006 — 18 pages

Abstract: Many networked applications use end-to-end congestion control protocols to avoid the occurrence of congestion collapse phenomena. However, such transport protocols may suffer from link underutilization when there are wireless links along the path: the inability to distinguish wireless losses from congestion losses often results in unnecessary throughput decreases. Since networks are becoming increasingly heterogeneous, consisting of a mix of wired and wireless links, it is important to have some means of correctly identifying the cause of a packet loss, so as to adapt the response of the transport protocol.

In this paper, we present an end-to-end solution to this problem and propose a novel method for Wireless Loss Estimation in IP DiffServ networks (WLED). WLED aims at enhancing the performance of equation-based rate control protocols for multimedia applications, by means of directly estimating the wireless and congestion loss rates. We discuss the integration of WLED with a recent, equation-based rate control protocol and we evaluate WLED's performance through simulation. Our results illustrate a new benefit of using DiffServ-like mechanisms for multimedia streaming in wireless networks.

Key-words: Multimedia, Differentiation, Wireless loss estimation, WLED, Congestion, Diffserv, Multimedia Streaming

(Résumé : tsvp)

This work was partially financed by the Conseil Régional de Bretagne, under grant B/1042/2004/MOTIPV6.

* kamal.singh@irisa.fr
** david.ros@irisa.fr
*** laurent.toutain@enst-bretagne.fr
**** cesar.viho@irisa.fr

Centre National de la Recherche Scientifique (UMR 6074) Université de Rennes 1 – Insa de Rennes
Institut National de Recherche en Informatique et en Automatique – unité de recherche de Rennes

Estimation des pertes dans les réseaux sans-fil pour améliorer la transmission du multimédia

Résumé : Les applications multimédia doivent en général utiliser des protocoles de contrôle de congestion de bout en bout afin d'éviter l'écroulement du réseau par congestion. Cependant, lorsque de tels protocoles sont employés il y a une sous-utilisation de la bande passante lorsqu'il y a des liaisons sans-fils le long du chemin : l'incapacité de distinguer des pertes dues au sans-fil de celles dues à la congestion entraîne une diminution inutile du taux de transmission.

Les réseaux deviennent de plus en plus hétérogènes et utilisent des liaisons filaires et sans-fil. Il est donc important de disposer de moyens permettant d'identifier correctement la cause de la perte des paquets afin d'adapter le mécanisme de contrôle de congestion du protocole de transport.

Dans cet article, nous présentons une solution de bout en bout à ce problème et proposons une méthode d'estimation de pertes dues au sans-fil dans les réseaux IP DiffServ (WLED: Wireless Loss Estimation in IP Diffserv networks). WLED augmente la performance des protocoles de contrôle de débit basés sur des modèles mathématiques pour des applications multimédia, par estimation directe des taux de pertes dues au sans-fil et à la congestion. Nous étudions l'intégration de WLED dans un protocole récent de contrôle de débit et nous évaluons la performance de WLED par simulation. Nos résultats montrent un nouvel avantage d'utiliser les mécanismes DiffServ pour la transmission multimédia dans les réseaux sans-fil.

Mots clés : Multimédia, Différenciation, Estimation de perte sans fil, WLED, Congestion, Diffserv, Transmission de Multimédia

1 Introduction

The popularity of multimedia streaming applications is leading to an increasing demand of data services over wireless networks, such as third-generation (3G) cellular networks. Multimedia applications may utilize congestion control protocols to adapt their sending rate to the path conditions and prevent the network from suffering from congestion collapse. However, TCP-derived rate-control protocols, like TFRC [1, 2], may show a significant degradation of performance over wireless links. Such protocols usually take packet loss as an indication of congestion; they confuse wireless losses with congestion losses, so they unnecessarily reduce the throughput. In wireless links, a significant percentage of packets may be lost due to bad channel conditions. Thus, wireless losses can significantly degrade the performance of congestion control mechanisms for multimedia flows. Approaches like link-layer retransmissions (ARQ) and Forward Error Correction (FEC) can reduce the impact of wireless losses. Nonetheless, these schemes can never eliminate such an impact completely.

The ability of correctly discriminating among wireless and congestion losses can significantly enhance the performance of congestion control mechanisms, and also can be beneficial to applications that can adapt their error coding to the channel conditions.

In the literature, some end-to-end approaches to differentiate wireless losses have used Round Trip Time (RTT) variations to predict the nature of losses. Biaz and Vaidya [3] tested some congestion predictors based on RTT variations to decide whether congestion was present or not. They showed that the performance of these congestion predictors was not satisfactory. Parsa and Garcia-Luna-Aceves [4] proposed a variant of TCP that uses a state machine that changes TCP's congestion window size based on RTT variations. Tobe et al. [5] proposed a scheme called Spike which differentiates among degrees of congestion, instead of directly identifying wireless and congestion losses. It uses the Relative One-way Trip Time (ROTT) to identify the state of congestion. ROTT is used instead of delay, because of the fact that the delay value can be incorrect due to clock skew between the sender and the receiver; hence, a "relative" measurement is preferred. Cen et al. [6] proposed the so-called ZigZag scheme, which uses a combination of the values of ROTT and the number of lost packets to classify losses as wireless-related or congestion-related. They also proposed a hybrid loss discriminator, which uses the Biaz and Vaidya's [7], ZigZag and Spike methods and switches among them depending on the observed network conditions. Liu et al. [8] proposed a scheme using Loss Pairs and Hidden Markov Modeling techniques. This work also uses the changes on RTT over time to infer the cause of losses.

The above type of loss differentiation can be tricky and unreliable over wireless networks, because large delay fluctuations are inherent in such types of networks. Moreover, some of the previously-cited works report inaccuracies in these differentiators. Also, let us remark that other approaches exist, based on using explicit congestion notification (ECN) [9] to allow conveying non-ambiguous, explicit congestion signals to the end-hosts. We however limit our discussion to "pure" end-to-end approaches, in the sense that we exclude the possibility of any kind of signaling from the routers back to the traffic sources.

We propose a more reliable algorithm for end-to-end wireless loss estimation using DiffServ (WLED), which is based on the fact that DiffServ networks [10] may provide differential dropping to packets according to their drop precedences [11]. If overall loss rate for lower priority packets is not very high, then we can safely assume that the congestion loss rate for the highest priority packets will be insignificant. In such a case, the loss of highest priority packets will be mainly due to wireless errors. Thus, it is to be expected that, in general, there is a good correlation between wireless packet loss rate and the total loss rate of highest priority packets. We exploit this correlation for estimating the wireless loss rate. Note that this basic concept of using DiffServ's biased queuing behavior has been used in a similar way in [12], for improving the throughput of TCP flows by "de-randomizing" the congestion losses.

The remainder of this paper is organized as follows. Section 2 reviews some key concepts of the DiffServ architecture. Our proposal for estimating wireless and congestion losses in DiffServ networks is discussed in Section 3, together with its integration with a rate control protocol. Section 4 presents the results of a simulation-based study of WLED. Finally, Section 5 concludes the paper.

2 The DiffServ Architecture

The IETF (Internet Engineering Task Force) has developed some standards and technologies in order to enable Quality of Service (QoS) support in IP networks. The most widely accepted among them is the Differentiated Services architecture (DiffServ). In DiffServ, the user traffic is separated into different Classes of Service based on their individual requirements [10]. At the edge of the network, each packet is marked according to the treatment that it would like to receive inside the network. Packet marking may be done by edge routers, based on rate-metering or flow identification mechanisms, and also by the source application in order to request a differentiated (enhanced) treatment for some of its data [13]. The mark or tag is a coded value in the DiffServ Code Point field (DSCP) of the IP header. Packets with the same DSCP value belong to the same class and will receive the same treatment inside the network. The different treatments a particular router can implement are called Per-Hop Behaviors (PHB).

One of the standardized PHB is Assured Forwarding (AF) [11]. The AF PHB allows a network provider supporting DiffServ to offer different levels of forwarding assurances, for IP packets accomplishing a target throughput for each network aggregate. It provides four AF forwarding classes with three dropping priorities each that define the relative importance inside an AF class. These drop priorities are usually identified with colors: *green* for the lowest drop precedence (the highest forwarding priority), *yellow* for the middle one and *red* for the highest one.

RIO (RED with In and Out) [14] is the basic queue management mechanism suitable for the setup of the AF PHB. Active queue management (AQM) mechanisms provide congestion avoidance at the router. RIO is derived from RED [15], which is one of the best-known AQM mechanisms. RED avoids congestion by controlling the average queue size and comparing it against two thresholds, $\min_{th}$ and $\max_{th}$. Inside this "congestion-avoidance" interval, packets are discarded with a probability that grows linearly with the average queue size.

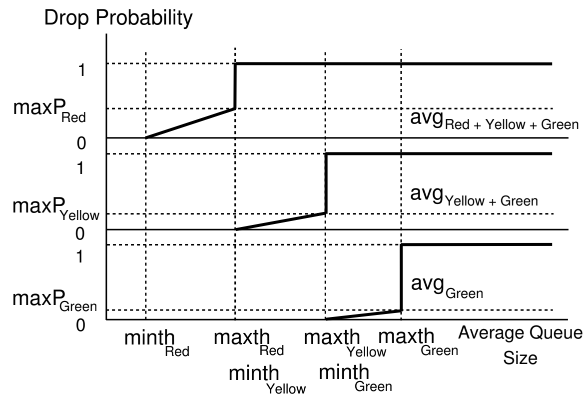


Figure 1: RIO "staggered" discard probability functions.

Several variants of RIO exist, of which Fig. 1 shows the "staggered thresholds" model [16]. It can be seen that the packets of lower priority are more likely to be discarded as compared to higher priority packets. Moreover, incoming higher priority packets are only dropped after discarding all the lower priority packets when queue length starts to increase, i.e., during congestion.

3 The WLED Scheme

WLED is a wireless loss estimation scheme that is designed to help congestion control schemes, in DiffServ networks supporting AF-based services. The idea of WLED is to exploit additional information regarding the character of losses, which can be obtained through the use of the AF PHB. The protection of higher-priority packets inherent in the staggered RIO algorithm means that, if the loss rate of low-priority packets is not significant, then we may assume that the loss of high-priority packets is highly correlated with the wireless loss rate.

3.1 WLED Algorithm

The pseudo-code of the WLED algorithm is shown in Fig. 2. Some details like whether (and how) to estimate the congestion probability are in fact separate from the generic algorithm; the particular choices shown in Fig. 2 will be discussed in Sections 3.2 and 3.3.

At the sender side, WLED maintains a separate sequence number N_j for the stream of packets marked with color $j \in \{green, yellow, red\}$, as well as a common sequence number N for all the packets sent[1]. These per-precedence sequence numbers are essential for the

[1] Other implementations are possible which do not require sending two sequence numbers in every packet, nor having the receiver compute the loss rates; we however adopted this method for the sake of simplicity.

```
SENDER:
  for every packet of color j {
    increment N and N_j;
    send the packet, using sequence
      numbers N and N_j;
  }

RECEIVER:
  every RTT {
    calculate l_green, l_yellow and l_total;
    if l_yellow < θ then w = l_green else w = 0;
    p̂_c = EWMA_Estimator(l_total, w);
    send an ACK with the value p̂_c;
  }

  EWMA_Estimator(l_total, w) {
    p_c = (l_total − w)/(1 − w);
    p̂_c ← αp_c + (1 − α) · p̂_c;
    return p̂_c;
  }
```

Figure 2: WLED algorithm.

receiver to be able to compute the *per-precedence* (per-color) loss rates l_j. At the receiver side, WLED estimates the losses every RTT. It computes the value of wireless loss rate w using the loss rate l_{green} of *green* packets. When the loss rate of lower (*yellow*) priority packets is less than a threshold $0 < \theta \leq 1$, then WLED takes w equal to the loss rate of *green* packets. If the loss rate of lower priority packets exceeds the threshold, then WLED conservatively assumes that all losses are due to congestion, so $w = 0$.

In the variant of the algorithm that we tested, the total loss rate l_{total} is calculated taking into account all the packets, irrespective of color. Besides, the receiver sends back to the sender an ACK with a (smoothed) estimated value $\hat{p}_c$ of the congestion loss probability.

3.2 Integration of WLED with Multimedia Rate-Control Protocols

There are several congestion control protocols that have been proposed in the literature. We will focus on equation-based rate control protocols[2]. We first considered both TFRC [1, 2] and ARC [17] as they both use mathematical models of TCP for rate control; we decided to initially work with ARC for the reasons discussed below.

[2] Note that WLED can work with any rate-control protocol that uses loss probability to calculate throughput.

TFRC is a mechanism for equation-based congestion control for unicast traffic, designed to be TCP-friendly. A TFRC sender adjusts its rate as a function of the measured rate of *loss events*, where a loss event consists of one or more packets dropped within a single round-trip time. The *loss event fraction* will obviously differ from the packet loss rate (for a thorough discussion see e.g. [2].) However, WLED estimates the wireless *packet loss rate*, thus, it is incompatible with the loss event rate used in the TFRC equation. WLED in fact needs to be coupled to a model of TCP that captures the *ideal* behavior of TCP in wireless scenarios, that is, to back off only if losses are due to congestion and to do nothing if losses are due to wireless-related phenomena. The required model should take into account the wireless loss probability in addition to the congestion loss probability, which is not the case with TFRC.

To the best of our knowledge, ARC [17] is the first to model this desired, ideal behavior of TCP facing wireless losses. ARC is a rate-control scheme that uses the following equation:

$$S = \frac{1}{4RTT}\left(3 + \sqrt{25 + \frac{24}{p_c}}\right), \tag{1}$$

where S is the sending rate in packets per second, RTT is the round trip time, and p_c is the congestion loss probability. The latter is related to the total packet loss probability π and the wireless loss probability w through the expression:

$$p_c = \left(\frac{\pi - w}{1 - w}\right). \tag{2}$$

ARC needs a way to calculate π and w, in order to further compute p_c; note that the value of π is easily estimated from the total packets received and the total packets lost, which in turn can be known by the receiver by looking at the sequence numbers. Calculating w is tricky and ARC relies on the MAC layer to get this loss probability. However, this approach violates the end-to-end paradigm and will not work if there is no way to obtain the wireless loss probability from lower layers.

For these reasons, we propose using WLED in conjunction with ARC over DiffServ-enabled networks. As discussed before, WLED calculates w by using the loss rate of *green* packets, which in turn can be known by looking at the sequence number N_{green} of such packets. WLED uses a loss estimation method to infer the values of p_c, w and π. At present, we only estimate the value of p_c, as shown in Fig. 2, because only that quantity is needed to control the sending rate. The value of w can also be inferred in the same way if needed by applications, say, for determining the level of forward error correction (FEC) to be used.

In addition to WLED, we also have integrated some mechanisms from TFRC (discussed in [2]) into ARC[3]. For example, we initially increase the sending rate in a fast way similar

[3]From now on, when we say "WLED-ARC" we refer to the ARC rate-control scheme in combination with WLED, TFRC's Slow Start and TFRC's increase and decrease methods; likewise, by "WLED users" we mean applications using such a scheme.

to TCP's Slow Start. Later, the rate is determined by the TCP model equation (1). If the model suggests increasing the rate, then we augment the sending rate by at most one packet per RTT; otherwise, it is directly reduced to the computed value.

3.3 Loss Estimation

Loss estimation is perhaps the most important part of any equation-based congestion control mechanism. Several loss estimators have been discussed in the literature. The loss estimator that we would like to use with WLED should have the following characteristics: it should take loss history into account, it should yield a "smooth" estimate of the loss probability (in order to avoid fast oscillations in the sending rate) and, at the same time, it should be fast enough to detect congestion build-up.

We first tested a trivial loss estimation method by counting losses over a time window of $n > 1$ RTTs. As will be discussed in Section 4, we found that this method results in strong oscillations in the sending rate. Thus, we do not recommend the use of the time-window method. We finally selected the well-known EWMA (exponentially-weighted moving average) estimation method, which uses the following update equation for computing the estimated quantity $\hat{E}$ from the current sample E:

$$\hat{E} \leftarrow \alpha E + (1 - \alpha)\hat{E}, \tag{3}$$

where the weight $\alpha \in (0, 1)$ determines the "responsiveness" of the estimator.

As shown in Fig. 2, we used the EWMA method to estimate p_c. The estimated, smoothed value of p_c is sent to the sender which utilizes it directly in the ARC rate equation (1).

4 Performance Evaluation

We will now present a simulation study of WLED in combination with ARC. We are interested in looking at the link utilization in wireless networks and to compare the WLED-ARC scheme with other congestion control protocols facing the same wireless loss conditions. Moreover, we tested the stability of sending rate and the TCP-friendliness of the scheme.

4.1 Experimental Setup

In this section we describe the general settings that we used to evaluate WLED. These settings remain the same for all the simulations unless specified otherwise.

All tests were done with the well-known ns-2 simulator [18]. The general topology that we used is as shown in Fig. 3. Simulated time is 120 seconds. All the links are of the same capacity, equal to 6 Mbps. The value of RTT (not considering the queuing delay) is 240 ms. The topology represents a network where the last link is a wireless one and congestion can occur just before the packets are sent over the link (e.g., UMTS and 802.11 networks). The "wireless links", which are present at the end of the "bottleneck link", mainly serve to introduce wireless packet losses. This is done using a simple drop model that discards

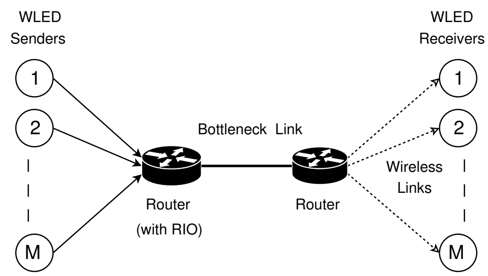


Figure 3: Simulation topology.

Table 1: RIO settings (Q = total buffer length).

Color	th_{min}	th_{max}	max_p
green	0.50 Q	0.70 Q	0.02
yellow	0.30 Q	0.50 Q	0.1
red	0.10 Q	0.30 Q	0.2

the packets randomly with the given wireless loss probability. The end points are the M WLED-ARC senders and M WLED-ARC receivers. The senders are not data-limited, i.e., they can send at high rates depending on the bandwidth availability. Moreover, the senders are responsible to mark their packets as *green*, *yellow* or *red*. In most of the tests we have used an assured rate[4] of 50%. We will call this a “marking profile of 50%”. The effect of the marking profile on the performance of WLED is also discussed later.

We used the RIO AQM to set up the AF PHB in our simulations. Configuring RIO and its parameters is in itself a separate area of research [16, 19–21]. Our RIO settings, as shown in Table 1, are qualitatively close to the values used in the above works. Moreover, we have chosen the “staggered” RIO model (shown in Fig. 1) that is recommended in [16]. The RIO algorithm uses an EWMA filter to estimate the average queue length over time in the same manner as RED. The smoothness of the filter output depends on a weight w_q; we have followed the recommendations of [22] and used $w_q = 0.0013$.

The value of Q, the total length of the RIO buffer should be as small as possible to minimize the queuing delay. On the other hand, it should be large enough so as to avoid excessive losses of *green* packets. After some tests we fixed Q to 300 kbytes, which corresponds to about three times the bandwidth-delay product.

[4]Assured rate is the guaranteed rate to the user by the DiffServ network provider. It is roughly equal to the rate of *green* packets because it signifies the percentage of the user's traffic that will get the best treatment.

4.2 Choice of Loss Estimator and Threshold θ

For loss estimation, we have selected the EWMA method because the time-window method produces oscillations in the sending rate, as seen in Fig. 5(a). Rate oscillations appear when the loss estimation is not smooth, as shown in Fig. 4(a). EWMA gives a better performance by yielding a smoother estimate of p_c, thus leading to a smoother sending rate; nonetheless, the smoothness depends on the value of the weight α. Selecting the right value of α is not obvious, because if this weight is too high then the sending rate will tend to oscillate (see Figs. 5(b) and 4(b)), whereas if it is too low then the sender will be slow to react to congestion. We choose $\alpha = 0.005$ as it seems to represent a good compromise between rate stability and responsiveness. This value also seems to work well for different wireless loss rates, as shown in Figs. 4(c) and 5(c).

We found the performance of WLED is not very sensitive to the value of θ, however, a low value may lead to underestimation of w. On the other hand, a higher value of θ can lead to overestimating w when *green* packets get lost due to congestion (though this should not be frequent since RIO offers a good protection to the *green* packets). The value of θ was chosen to be 0.5 in all the simulations except in Figs. 4 and 5, where $\theta = 0.8$ was used: we did so because were more concerned about finding a good setting for α and did not want noise in the estimation due to (a possibly too low) θ.

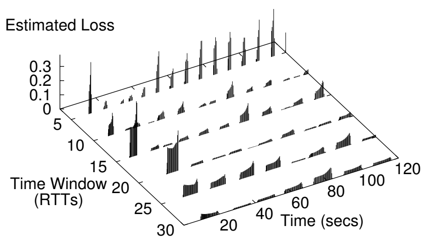
4.3 Link Utilization

One goal of WLED is to improve link utilization despite wireless losses. We measured the link utilization with WLED-ARC with different values of w. We set the number of users M to 20. For comparison purposes, we also measured the performance of TFRC and TCP NewReno under the same conditions with varying wireless loss probability.

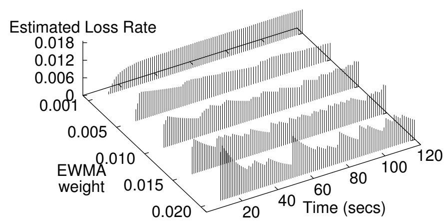
Fig. 6 shows the link utilization for different values of the wireless loss probability w. It can be seen that the link utilization with WLED-ARC is always $> 90\%$, even when w is as high as 0.2, whereas utilization drops sharply with TFRC and TCP for relatively low values of w. WLED performs better than TFRC and TCP NewReno because the latter two fail to distinguish among congestion and wireless losses.

4.4 TCP Friendliness

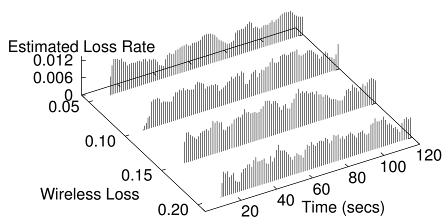
Link utilization alone is not a good indicator of performance for congestion control mechanisms; such mechanisms should also share the available bandwidth in a fair way. Since ARC was designed to be TCP-friendly, we tested WLED-ARC for its TCP-friendliness using the topology shown in Fig. 7. The difference from the original settings is that M TCP users are present in addition to the M WLED users. All users compete for their share of the bottleneck bandwidth. In addition, the TCP flows do *not* travel through wireless links. This was done due to the fact that TCP does not differentiate between wireless and congestion losses and its performance may be bad over wireless links. Thus, no wireless losses were introduced to TCP flows to give fairer conditions to them.



(a) Time Window estimator.

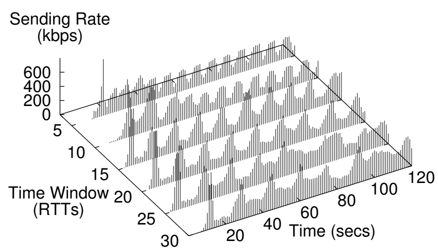


(b) EWMA with different weights.

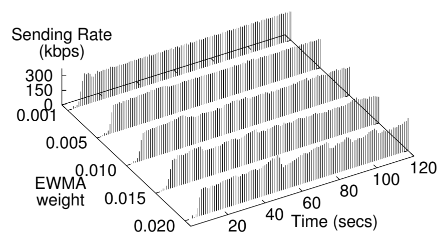


(c) EWMA with different wireless losses and weight = 0.005.

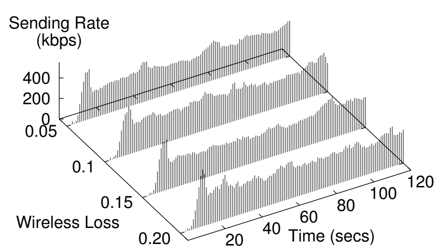
Figure 4: Estimated Congestion Loss.



(a) Time Window estimator.



(b) EWMA with different weights.



(c) EWMA with different wireless losses and $\alpha = 0.005$.

Figure 5: Sending Rate variation.

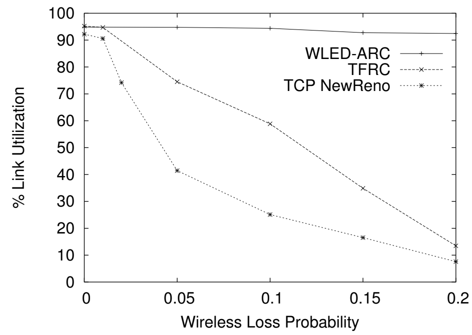


Figure 6: Link Utilization for Different Wireless Loss Probabilities.

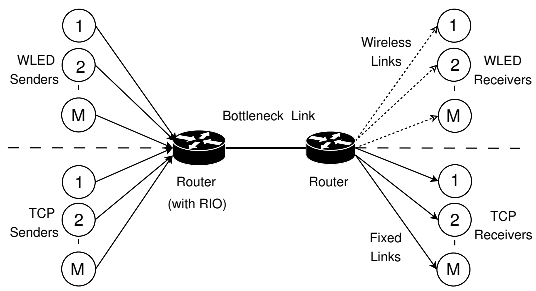
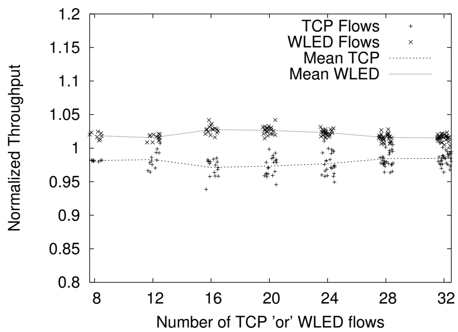


Figure 7: Topology used for testing TCP friendliness.

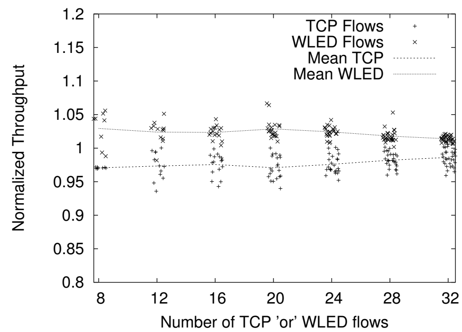
Fig. 8 shows the fairness of WLED-ARC when it competes with TCP NewReno flows, the latter subject to no wireless losses. The number of flows M and the wireless loss probability w for WLED was varied, and the normalized throughput of each flow was plotted (shown as individual plot marks). Note that WLED-ARC is indeed fair with respect to TCP: on the average, the throughput of a WLED-ARC flow is no more than 10% greater than that of a TCP flow, irrespective of the values of M and w. The behavior observed in Fig. 8(c) for low values of M might be due to the fact that, at higher values of w, the running estimate of l_{yellow} crosses the threshold θ more often. This may lead to underestimation of w, which in turn decreases the aggressiveness of WLED-ARC senders to probe for available bandwidth.

4.5 Marking Scheme

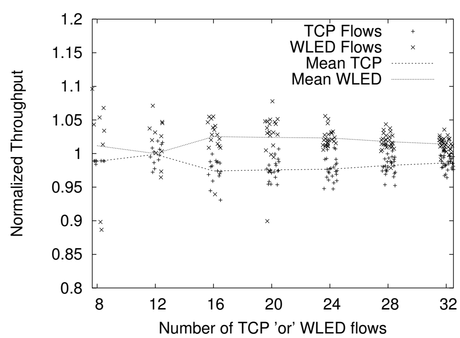
Applications can decide to mark their packets in arbitrary ways, meaning that the proportion of *green* packets may change from one application to the other. In the context of video streaming, the marking ratio may be very flexible when hierarchical, scalable video codecs [23] are used.



(a) No wireless losses ($w = 0$).



(b) $w = 0.1$.



(c) $w = 0.2$.

Figure 8: TCP NewReno and WLED.

To test the impact of the marking scheme on WLED-ARC we used the topology shown in Fig. 7 with $M = 12$. We measured the stability over time of the sending rate by computing its coefficient of variation (CoV), that is, the ratio of the standard deviation to the mean value; a low CoV value means higher rate stability. In order to calculate the CoV, throughputs were measured[5] using a time window of 1 s. The values of CoV obtained with different marking profiles and wireless loss probabilities are shown in Fig. 9(a). The error bars show the deviation of CoV values between different users. It can be seen that the CoV tends to increase (meaning a decrease in rate stability) for lower values of assured rate. This is because lower values of assured rate ($< 30\%$) mean that less *green* packets are available for estimation per RTT, hence, the quality of the estimation deteriorates.

In order to see the effect of the marking profile on TCP friendliness, we plotted the ratio of throughput of total WLED flows and throughput of total TCP flows in Fig. 9(b). The percentage of assured rate and wireless loss probability were varied and, for each case, the simulation was run 10 times. The average and standard deviation, shown by the error bars in the graph, obtained from these simulations is plotted. Similar to the above case, it can be seen that the lower values of assured rate ($< 30\%$) give a poorer TCP friendliness.

5 Conclusion

In this paper we have proposed a wireless loss estimation scheme for DiffServ-enabled networks that works together with equation-based congestion control schemes. This novel scheme, WLED, estimates the wireless probability in an end-to-end fashion. This significantly helps congestion control protocols because they can use this information and reduce their throughput accordingly, otherwise they would unnecessarily drop their sending rates, resulting in a very low utilization of links. Estimation of wireless loss probability can also help the applications, as some of them can optimize the amount of forward error correction based on this information.

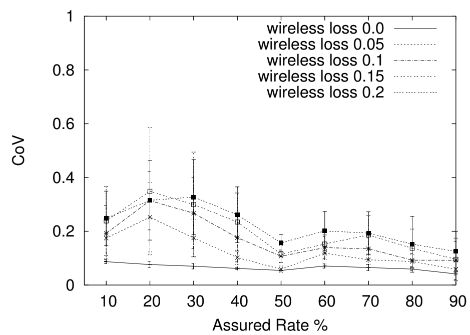
WLED was evaluated with the ARC congestion control scheme. In comparison with TFRC and TCP, we found that the link utilization of WLED-ARC was far better, while showing stability of sending rates and TCP friendliness for a diverse range of scenarios.

Concerning our future work, we are currently exploring some outstanding issues like the impact of re-marking ("re-coloring") of packets by edge routers and of time-varying RTTs and wireless drop rates, as well as the use of adaptive loss estimation methods.

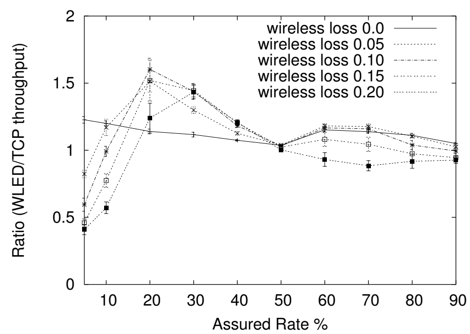
References

[1] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification," Internet Standards Track RFC 3448, IETF, Jan. 2003.

[5] Simulations were run for 140 s, and the rate values for the first 20 s were discarded to filter out the start-up phase.



(a) CoV of Throughputs with Different Assured Rates and Measuring Window of 1 s.



(b) Ratio of WLED and TCP Throughputs with Different Assured Rates.

Figure 9: Stability and TCP-Friendliness of WLED

[2] S. Floyd, M. Handley, J. Padhye, and J. Widmer, “Equation-Based Congestion Control for Unicast Applications,” *Technical report TR-00-03, International Computer Science Institute*, March 2000.

[3] S. Biaz and N. Vaidya, “Distinguishing congestion losses from wireless transmission losses: A negative result,” in *Proceedings of IEEE IC3N ’98*, Washington, DC, USA, 1998, p. 722.

[4] C. Parsa and J. Garcia-Luna-Aceves, “Differentiating congestion vs. random loss: A method for improving TCP performance over wireless links,” in *IEEE WCNC’2000*, 2000, pp. 90–93.

[5] Y. Tobe, Y. Tamura, A. Molano, S. Ghosh, and H. Tokuda, “Achieving moderate fairness for UDP flows by path-status classification,” in *Proceedings of IEEE LCN ’00*, Washington, DC, USA, 2000, p. 252.

[6] S. Cen, P. C. Cosman, and G. M. Voelker, “End-to-end differentiation of congestion and wireless losses,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 703–717, 2003.

[7] S. Biaz and N. H. Vaidya, “Discriminating congestion losses from wireless losses using inter-arrival times at the receiver,” CS Dept., Texas A & M University, Technical Report 98-014, Aug. 1998.

[8] J. Liu, I. Matta, and M. Crovella, “End-to-end inference of loss nature in a hybrid wired/wireless environment,” in *Proceedings of WiOpt’03*, 2003.

[9] K. Ramakrishnan, S. Floyd, and D. Black, “The Addition of Explicit Congestion Notification (ECN) to IP,” Internet Standards Track RFC 3168, IETF, Sept. 2001.

[10] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An architecture for differentiated services,” Internet Standards Track RFC 2475, IETF, Dec. 1998.

[11] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, “Assured Forwarding PHB Group,” Internet Standards Track RFC 2597, IETF, June 1999.

[12] S. Biaz and N. H. Vaidya, “”De-randomizing” congestion losses to improve TCP performance over wired-wireless networks,” *IEEE/ACM Transactions on Networking*, vol. 13, no. 3, pp. 596–608, 2005.

[13] J. Orozco and D. Ros, “Diffserv-Aware Streaming of H.264 video,” in *Proceedings of the 14th International Packet Video Workshop (PV2004)*, December 2004.

[14] D. D. Clark and W. Fang, “Explicit allocation of best-effort packet delivery service,” *IEEE/ACM Transactions on Networking*, vol. 6, no. 4, pp. 362–373, 1998.

[15] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[16] R. Makkar, I. Lambadaris, J. Salim, N. Seddigh, B. Nandy, and J. Babiarz, "Empirical study of buffer management scheme for DiffServ Assured Forwarding PHB," in *Proceedings of IEEE ICCCN 2000*, 2000.

[17] O. B. Akan and I. F. Akyildiz, "ARC: the analytical rate control scheme for real-time traffic in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 634–644, 2004.

[18] S. McCanne and S. Floyd, "`ns` Network Simulator," http://www.isi.edu/nsnam/ns/.

[19] C.-H. Chien and W. Liao, "A self-configuring RED gateway for quality of service (QoS) networks," in *International Conference on Multimedia and Expo, 2003 ICME'03.*, vol. 1, July 2003.

[20] J. Orozco and D. Ros, "An Adaptive RIO (A-RIO) queue management algorithm." in *Proceedings of QoFIS 2003*, 2003.

[21] W. Liu, Z. Yang, J. He, C. Le, and C. T. Chou, "Analysis and improvement on the robustness of AQM in Diffserv networks," in *Proceedings of IEEE ICC 2004*, vol. 4, June 2004, pp. 2297–2301.

[22] S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: An algorithm for increasing the robustness of RED," ICIR," Technical Report, 2001.

[23] W. Li, "Overview of Fine Granularity Scalablility in MPEG-4 Video Standard," *IEEE Transactions on Circuit and Systems for Video Technology*, vol. 11, no. 3, March 2001.