Advanced Communication Protocol Technologies:

Solutions, Methods, and Applications

Katalin Tarnay
University of Pannonia, Hungary

Gusztáv Adamis
Budapest University of Technology and Economics, Hungary

Tibor Dulai
University of Pannonia, Hungary

Information Science REFERENCE

Senior Editorial Director:	Kristin Klinger
Director of Book Publications:	Julia Mosemann
Editorial Director:	Lindsay Johnston
Acquisitions Editor:	Erika Carter
Development Editor:	Hannah Abelbeck
Production Editor:	Sean Woznicki
Typesetters:	Mike Brehm, Keith Glazewski, Natalie Pronio, Jennifer Romanchak, & Deanna Zombro
Print Coordinator:	Jamie Snavely
Cover Design:	Nick Newcomer

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data
Advanced communication protocol technologies : solutions, methods, and applications / Katalin Tarnay, Gusztáv Adamis, Tibor Dulai, editors.
p. cm.
Includes bibliographical references and index.
Summary: "This book explores the complications and solutions created by communication required between ever-expanding technologies, covering the fundamentals of protocol functions and protocol operations, the controlling protocols of ISDN and mobile networks, the evolution of IP-based protocols, and advanced solutions for routing, mobility and multimedia transmission"-- Provided by publisher.
ISBN 978-1-60960-732-6 (hardcover) -- ISBN 978-1-60960-733-3 (ebook) -- ISBN 978-1-60960-734-0 (print & perpetual access) 1. Telecommunication. I. Tarnay, Katalin. II. Adamis, Gusztáv, 1965- III. Dulai, Tibor, 1979-
TK5101.A35 2011
621.382'12--dc23
2011013392

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The views expressed in this book are those of the authors, but not necessarily of the publisher.

Chapter 14
The TFRC Protocol and Its Usage for Wireless Video Transmission

Christos Bouras
Research Academic Computer Technology Institute, Greece & University of Patras, Greece

Vassilis Papapanagiotou
Research Academic Computer Technology Institute, Greece, & University of Patras, Greece

Kostas Stamos
Research Academic Computer Technology Institute, Greece & University of Patras, Greece & Technical Educational Institute of Patras, Greece

Giannis Zaoudis
Research Academic Computer Technology Institute, Greece & University of Patras, Greece

ABSTRACT

The subject of this chapter is to present the TFRC (TCP-Friendly Rate Control) protocol in the area of efficient wireless video transmission and its possible usage in cross-layer power management mechanisms. The basic aspects of TFRC operation are presented, along with the suitability of TFRC usage for video transmission. The chapter examines related work and presents several mechanisms for efficient wireless video transmission using TFRC that have been proposed. These mechanisms utilize cross-layer approaches for adaptation of the power transmission level of the sender and TFRC feedback information regarding the wireless connection status from the receiver for improved transmission statistics, and therefore user experience, without unnecessary power consumption.

INTRODUCTION

Networking complexity has led to the modularization of network architecture in layers. Traditional approaches focus on wired networks and try to separately optimize each network layer such as the physical, the medium access, the routing and the transport layer. This approach reduces the complexity and makes issues more manageable and architectures more flexible and upgradeable,

DOI: 10.4018/978-1-60960-732-6.ch014

but it may lead to suboptimal designs. Under this layered approach, communication occurs between two adjacent layers without taking into consideration the specific characteristics of multimedia applications. Although this layered approach has been the fundamental factor for the growth of the wired networks and the World Wide Web it seems to pose constrains when attempting to adapt protocol's behavior to multimedia applications characteristics and to wireless network conditions. Therefore, a careful cross-layer approach, where selected communication and interaction between layers is allowed, can have performance advantages without negating the successful layer separation that has guided network design so far. A theoretical discussion of the cross-layer problem framework can be found in (Schaar & Shankar, 2005).

Wireless transmission differs in an important way from wired communication, in that the notion of the link is not as fixed and can vary depending on the movement of the communicating nodes, the intermediate interferences and the transmission characteristics of the communicating nodes, most notably their transmission power. While increased power generally correlates with a stronger signal and therefore improved transmission characteristics, in many wireless scenarios reduced power consumption is desired. This trade-off has been explored by various researchers studying TCP (Transmission Control Protocol) modifications (Tsaoussidis & Badr, 2000, Zhang & Tsaoussidis, 2001, Jones et. al, 2001) trying to combine reduced power consumption with increased data throughput. Wireless standards such as IEEE 802.11 specify power saving mechanisms (IEEE 802.11 PSM), although studies have shown that PSM and other similar mechanisms carry a significant performance penalty in terms of throughput (Molta, 2005, Chen & Huang, 2004, Anastasi et. al, 2004, Simunic, 2005).

An important issue for the efficiency of wireless networks is to accurately determine the cause of packet losses. Packet losses in wired networks occur mainly due to congestion in the path between the sender and the receiver, while in wireless networks packet losses occur mainly due to corrupted packets as a result of the low Signal to Noise Ratio (SNR), the multi-path signal fading and the interference from neighboring transmissions. A second difference between wired and wireless networks is the "mobility factor". Mobility in wireless networks introduces a number of additional barriers in multimedia data transmission. Channel fading and handover time are the most important factors that cause packet losses as they introduce additional delays when the mobile user changes its location from one Access Point (AP) to another.

According to its specification, TFRC (Handley et al, 2003) is a congestion control mechanism for unicast flows operating in a best-effort Internet environment. It aims to be reasonably fair when competing for bandwidth with TCP flows, but at the same time achieving a much lower variation of throughput over time compared with TCP, making it thus more suitable for applications such as telephony or streaming media where a relatively smooth sending rate is important. However, TFRC is slower than TCP in responding to the available bandwidth. TFRC congestion control is appropriate for flows that would prefer to minimize abrupt changes in the sending rate, including streaming media applications with small or moderate receiver buffering before playback. TCP-like congestion control, halves the sending rate in response to each congestion event and thus cannot provide a relatively smooth sending rate.

Several researchers have focused on various issues of cross-layer optimization for wireless ad hoc networks, when there is no infrastructure assumed. Also several efforts have been made in order to combine efficiency and TCP fairness. These works are discussed in the relevant sections of this chapter.

RELATED WORK

As noted above, cross-layer optimization for wireless ad hoc networks, when there is no infrastructure assumed is a very active research field. (Radunovic, 2005) proposes a jointly optimal design of the three layers (physical, MAC (Medium Access Control), routing) for wireless ad-hoc networks and studies several existing rate-maximization performance metrics for wireless ad-hoc networks in order to select appropriate performance metrics for the optimization. In (Shan & Zakhor, 2002) the authors propose an application adaptive scheme based on priority based ARQ (Automatic Repeat Request) together with a scheduling algorithm and FEC (Forward Error Correction) coding combined with RLP (Radio Link Protocol) layer granularity. In (Schaar & Shankar, 2005) the need of a cross-layer optimization is examined and an adaptation framework is proposed amongst the application (APP), the Medium Access Control (MAC) and the Physical (PHY) layers. In the same publication a number of different methodologies for cross-layer adaptation are proposed, named "top-down" approach, "bottom-up", "application centric" and "MAC centric".

(Chen et al., 2004) a joint cross-layer design for QoS content delivery is presented. The central concept of the proposed design is that of adaptation. A new QoS-awareness scheduler with a power adaptation scheme is proposed and is applied at both uplink and downlink Medium Access Control (MAC) layer to coordinate the behavior of the lower layers for resource efficiency. The work in (Lin et al., 2006) summarizes the recent developments in optimization based approaches for resource allocation problems in wireless networks using a cross-layer approach. (Li et al., 2008) deals with 802.16 (WiMax) networks. The 802.16 standard provides four kinds of multimedia data services with QoS parameters but does not define any QoS scheduling algorithm. The above mentioned paper presents an adaptive cross-layer scheduling algorithm for the IEEE 802.16 BWA system. The algorithm uses adaptive modulation and coding (AMC) scheme at the physical layer according to the Signal to Noise Ratio (SNR) on wireless fading channels. In addition the cost function is defined for each kind of multimedia connection based on its service status, throughout the deadline in MAC layer. The simulation results provided show that the scheduling algorithm achieved an optimum trade-off between throughput and fairness. In (Warrier, 2007), the gap between existing theoretical cross-layer optimization designs and practical approaches is examined. The trade-off between increased power consumption and improved signal strength has been explored by various researchers studying TCP modifications (Tsaoussidis & Badr, 2000, Zhang & Tsaoussidis, 2001, Jones et. al, 2001) trying to combine reduced power consumption with increased data throughput. Wireless standards such as IEEE 802.11 specify power saving mechanisms, although studies have shown that PSM (Power Saving Mode) and other similar mechanisms carry a significant performance penalty in terms of throughput (Molta, 2005, Chen & Huang, 2004, Anastasi et. al, 2004, Simunic, 2005). Several researchers have dealt with the issue of optimized video transmission using power management techniques (Kim & Kim, 2003, Zamora et al., 2007), but these approaches do not utilize feedback available from protocols such as TFRC.

TFRC can be viewed as a congestion control technique that trades responsiveness to the network conditions for a smoother throughput variation. TCP-Friendly Rate Control offers an alternative to the method in (Floyd & Kempf, 2004). The key differentiator of TFRC, relative to the Additive Increase Multiplicative Decrease (AIMD) method used in TCP and SCTP (Stewart, 2007), is its smooth response to packet loss. TFRC has been implemented as one of the "pluggable" congestion control algorithms for the Datagram Congestion Control Protocol (DCCP, Kohler et al., 2006 and Floyd et. al., 2006) and as a profile for RTP (Lee & Chung, 2006).

Previous work in (Bouras, 2009) introduced a power management mechanism for TFRC, which operates in a MIMD (Multiplicative Increase, Multiplicative Decrease) fashion. In general, when the packet loss increases above a preset threshold, then the power is also increased, else if the packet loss falls below the threshold, the power consumed is decreased. Moreover the power consumed has a lower bound to prevent the base station from halting the transmission and an upper bound to prevent excessive consumption. This mechanism (called MIMD approach) has been extended with a more sophisticated method for quicker selection of the optimal trade-off, called Binary approach.

THE TFRC PROTOCOL

In this section we provide a short summary of the TFRC operation, in order to demonstrate the way that it tries to achieve UDP-levels of efficiency with TCP friendliness. The TFRC (TCP-friendly Rate Control) protocol presents a modern approach to transport layers protocols, which tread protocols as a set of building blocks – independent components, from which transport protocols are assembled. TFRC provides a sending rate within a factor of two of the sending rate a TCP flow would have under the same condition but with relatively more stable throughput which is a desirable characteristic for a streaming service. TFRC is a receiver-based mechanism where the receiver performs some calculation of the congestion control indicators and reports them back to the server. It relies on the underlying transport protocol such as the DCCP (Kohler et al., 2006) to provide means for the exchange of control information between the server and the client.

The algorithm used to calculate the next sending rate depends on whether the sender is still in the initial Slow Start phase or in the Congestion Avoidance phase. In the Slow Start phase, the sender approximately tries to double its sending rate every time a Receiver Report is received in order to reach the maximum throughput the channel can support which can be detected by increasing *RTT* and losses. Once the first loss has been detected, the sender enters the Congestion Avoidance phase. The next sending rate *X* is now determined from the minimum between twice the previous receiving rate and the sending rate as calculated from the TCP throughput equation.

X = min (TCP throughput, 2*receiving rate)

For its congestion control mechanism, TFRC directly uses a throughput equation for the allowed sending rate as a function of the loss event rate and round-trip time. In order to compete fairly with TCP, TFRC uses the TCP throughput equation, which roughly describes TCP's sending rate as a function of the loss event rate, round-trip time, and packet size. Specifically, TFRC's throughput equation is a slightly simplified version of the throughput equation for Reno TCP:

$$X_{TFRC} = \frac{s}{R\sqrt{\frac{2bp}{3}} + t_{RTO}(3\sqrt{\frac{3bp}{8}})p(1+32p^2)}$$

Where:

- X_{TFRC} is the transmit rate in bytes/second.
- *s* is the packet size in bytes.
- *R* is the round trip time in seconds.
- *p* is the loss event rate, between 0 and 1.0, of the number of loss events as a fraction of the number of packets transmitted.
- *t_RTO* is the TCP retransmission timeout value in seconds.
- *b* is the number of packets acknowledged by a single TCP acknowledgement. The value of b is recommended to be set to 1.

Moreover, it would be beneficial to briefly describe some well-known TCP-like congestion control mechanisms like TCP Vegas, TCP Hybla, TCP Tahoe and Reno. In TCP Vegas, timeouts are set and round-trip delays are measured for every packet in the transmit buffer, while in other TCP versions, are based upon only the last transmitted packet in the transmit buffer. TCP Hybla aims to eliminate penalization of TCP connections that incorporate a high-latency terrestrial or satellite radio link, due to their longer round trip times. It stems from an analytical evaluation of the congestion window dynamics, which suggests the necessary modifications to remove the performance dependence on *RTT*. To avoid congestion collapse, TCP uses a multi-faceted congestion control strategy. For each connection, TCP maintains a congestion window, limiting the total number of unacknowledged packets that may be in transit end-to-end. When the congestion window exceeds a certain threshold, the algorithm enters a new state, called congestion avoidance. The congestion avoidance mechanisms of Tahoe and Reno are not the same, and specifically the behavior of Tahoe and Reno differ in how they detect and react to packet loss. In Tahoe, triple duplicate *ACKs* are treated the same as a timeout, while in Reno, if three duplicate *ACKs* are received, Reno will halve the congestion window.

TFRC defines a loss event as one or more lost or marked packets from a window of data, where a marked packet refers to a congestion indication from Explicit Congestion Notification (Ramakrishnan, 2001). TFRC congestion control mechanism works as follows:

- The receiver measures the loss event rate and feeds this information back to the sender.
- The sender also uses these feedback messages to measure the round-trip time (*RTT*).
- The loss event rate and *RTT* are then fed into TFRC's throughput equation, giving the acceptable transmit rate.
- The sender then adjusts its transmit rate to match the calculated rate.

The dynamics of TFRC are sensitive to how the measurements are performed and applied. Specific mechanisms are used to perform and apply these measurements. Other mechanisms are possible, but it is important to understand how the interactions between mechanisms affect the dynamics of TFRC.

For the purposes of the cross-layer mechanisms detailed later in the chapter, it is very important to understand the mechanism and structure of the feedback packets that the TFRC protocol specifies.

The receiver periodically sends feedback messages to the sender. Feedback packets should normally be sent at least once per *RTT*, unless the sender is sending at a rate of less than one packet per *RTT*, in which case a feedback packet should be send for every data packet received. A feedback packet should also be sent whenever a new loss event is detected without waiting for the end of an *RTT*, and whenever an out-of-order data packet is received that removes a loss event from the history. If the sender is transmitting at a high rate (many packets per *RTT*) there may be some advantages to sending periodic feedback messages more than once per *RTT* as this allows faster response to changing *RTT* measurements, and more resilience to feedback packet loss. However, there is little gain from sending a large number of feedback messages per *RTT*.

Each feedback packet sent by the data receiver contains the following information:

- The timestamp of the last data packet received. We denote this by *t_recvdata*. If the last packet received at the receiver has sequence number i, then *t_recvdata* = *ts_i*. This timestamp is used by the sender to estimate the round-trip time, and is only needed if the sender does not save timestamps of transmitted data packets.
- The amount of time elapsed between the receipt of the last data packet at the receiver, and the generation of this feedback report. We denote this by *t_delay*.

- The rate at which the receiver estimates that data was received since the last feedback report was sent. We denote this by *X_recv*.
- The receiver's current estimate of the loss event rate, *p*.

The sender's behavior specified by TFRC when a feedback packet is received is as follows:

The sender knows its current sending rate, *X*, and maintains an estimate of the current round trip time, *R*, and an estimate of the timeout interval, *t_RTO*.

When a feedback packet is received by the sender at time *t_now*, the following actions should be performed:

1. Calculate a new round trip sample.

```
R_sample = (t_now - t_recvdata)
- t_delay.
```

2. Update the round trip time estimate:

```
if no feedback has been received
before
   R = R_sample;
else
   R = q*R + (1-q)*R_sample;
```

TFRC is not sensitive to the precise value for the filter constant *q*, but a default value of 0.9 is recommended.

3. Update the timeout interval:

```
_RTO = 4*R.
```

4. Update the sending rate as follows:

```
if (p > 0)
   Calculate X_calc using the TCP
   throughput equation.
   X = max(min(X_calc, 2*X_recv),
   s/t_mbi);
else
      if (t_now - tld >= R)
        X = max(min(2*X, 2*X_recv), s/R);
        tld = t_now;
```

Note that *if p is equal to zero*, then the sender is in slow-start phase, where it approximately doubles the sending rate each round-trip time until a loss occurs. The *s/R* term gives a minimum sending rate during slow-start of one packet per *RTT*. The parameter *t_mbi* is 64 seconds, and represents the maximum inter-packet backoff interval in the persistent absence of feedback. Thus, when p is greater than zero, the sender sends at least one packet every 64 seconds. The variable tld is an abbreviation for Time Last Doubled.

5. Reset the nofeedback timer to expire after *max(4*R, 2*s/X)* seconds.

In order the sender to receive the feedback analyzed above, the receiver is responsible for the calculation of the Loss Event Rate (*p*).

Obtaining an accurate and stable measurement of the loss event rate is of primary importance for TFRC. Loss rate measurement is performed at the receiver, based on the detection of lost or marked packets from the sequence numbers of arriving packets. We describe this process before describing the rest of the receiver protocol.

TFRC USED FOR VIDEO TRANSMISSION

A very important issue on video transmission is high fluctuations and oscillations which may damage the video transmission, which demands smooth transmission rates. Most video algorithms such as MPEG2 utilize the three major frame types (I-frames, P-frames, B-frames). The video bit rate tends to vary according to the complexity of the frame data, for example an I-frame would be more complex compared to a P-frame as it results in more bits after compression. The same also applies to scene changes and high motion scenes in a video sequence as they tend to incur a higher prediction error which results in a lower compression efficiency. Thus a typical video bit rate will have occasional "pulses". A smoothed transmission rate will reduce these "pulses" and ends up affecting the video quality. To prevent oscillatory behavior in environments with a low degree of statistical multiplexing it is useful to modify sender's transmit rate to provide congestion avoidance behavior by reducing the transmit rate as the queuing delay (and hence *RTT*) increases. To do this the sender maintains an estimate of the long-term *RTT* and modifies its sending rate depending on how the most recent sample of the *RTT* differs from this value. The long-term sample is *R_sqmean*, and is set as follows:

```
if no feedback has been received
before
   R_sqmean = sqrt(R_sample);
else
   R_sqmean = q2*R_sqmean +
   (1-q2)*sqrt(R_sample);
```

Thus *R_sqmean* gives the exponentially weighted moving average of the square root of the *RTT* samples. The constant *q2* should be set similarly to *q*, and a default value of 0.9 is recommended.

The sender obtains the base transmit rate, *X*, from the throughput function. It then calculates a modified instantaneous transmit rate *X_inst*, as follows:

```
X_inst = X * R_sqmean / sqrt(R_sample);
```

When *sqrt(R_sample)* is greater than *R_sqmean* then the queue is typically increasing and so the transmit rate needs to be decreased for stable operation.

This modification is not always strictly required, especially if the degree of statistical multiplexing in the network is high. However, it is recommended that it is done because it does make TFRC behave better in environments with a low level of statistical multiplexing. If it is not done, it is recommend using a very low value of *q*, such that *q* is close to or exactly zero.

Another important issue is the protocol's transmission rate. TFRC computes its maximum transmission rate as the number of packets per second that a TCP application would receive in similar conditions while breaking up its data into 1480-byte chunks. A TFRC application that is using large packets will experience roughly the same transmission rate in bits per second as a TCP application. However, a TFRC application using small packets will experience a lower transmission rate, in bits per second, than a TCP application. The reasoning for this is that bottlenecks can be the bits per second capacity of links, and also the packets per second capacity of routers. In the subsequent sections of this chapter, we present TFRC mechanisms that still remain TCP-friendly, yet their goal is not to contribute too much to network congestion but to achieve a reasonable video quality gain over the conventional method.

POWER MANAGEMENT MECHANISMS USING TFRC REPORTS

Over the last years a number of new protocols have been developed for multimedia applications in the whole OSI layer's scale. The MPEG protocol family includes the encoding and compression of multimedia data. The MPEG-4 protocol with the enhancements of the FGS (Fine Granularity Scalability), AVC (Advanced Video Coding) and SVC (Scalable Video Coding) provides adaptive video coding by taking into account the available bandwidth and is expected to be used by many multimedia applications. Moreover, congestion control and TCP-friendliness pose additional design requirements as highly fluctuating ("shark teeth"-like) transmission rates may be too difficult to be followed by Audio-Video (AV) encoders and decoders. TCP congestion control produces high fluctuations in the transmission rate which are not suitable for the current audio-video codecs which expect predictive and stable bandwidth allocation. Therefore, the development of protocols such as TFRC can be seen as a step to improve multimedia transmission. One way to cope with transient fluctuations of the transmission rate is with the use of buffers at the clients. However, an initial data pre-fetch in a buffer of more than 8 seconds before the player starts playing the stream is not easily accepted by the end user. Moreover, in real time video applications and conversational media large pre-fetch buffers are not acceptable. For multimedia applications smooth and steady transmission rates and low delay are more important attributes than guaranteed and on order delivery of data packets.

The MIMD Mechanism

The first mechanism is henceforth called the MIMD (Multiplicative Increase, Multiplicative Decrease) mechanism (Bouras et al., 2009). The MIMD mechanism uses the TFRC receiver's reports to the sender in order to calculate the packet loss rate percentage. The algorithm considers only a constant number of previous packet losses, so that it is more adaptive to the most recent conditions of the network. In addition, if the packet loss rate increases above a preset threshold, then the power is also increased by a percentage, else if the packet loss falls way below the threshold, the power consumed is decreased for reasons of power efficiency. Moreover the power consumed has a lower bound to prevent the base station from halting the transmission and an upper bound to prevent excessive consumption. This cross-layer mechanism uses information provided by the TFRC protocol which is a transport layer protocol and needs to act upon the physical layer to adjust the transmission power. The parameters involved by each layer include the transmission power at the physical layer, and the packet loss information at the transport layer. The interaction of these parameters is explained in the pseudo code below.

Below the mechanism is expressed in a more compact form using pseudo code (PL stands for Packet Losses (as a percentage) and TP for Transmission Power, while A>1 and 0<B<1):

```
while (true) {
  retrieve last TFRC report
  set PL = Average of last N reports
  if (PL > Threshold_1) and (TP < Upper_
  Bound) then
    set TP = A * TP
  else if (PL < Threshold_2) and (TP >
  Lower_Bound) then
    set TP = B * TP
}
```

After extensive experimentation with the values *A,B* and the thresholds, (Bouras et al., 2009) conclude that the values *A=1.05, B=0.05*, *Threshold_1=0.1, Threshold_2=0.075* lead to both good PSNR values and limited energy consumption. The values for *Upper_Bound* and *Lower_Bound* are discussed in the Experiments section.

The target of the proposed mechanism is to minimize or eliminate packet losses, since even a small packet loss rate can result to important reduction of multimedia quality in the end user and result to a bad end user experience. Improvements in the above two areas will lead to improved media parameters such as PSNR and MOS, which better represent the end user experience. At the same time, it has to make sure that power consumption will be bounded and will only increase when this results to noticeably improved video quality.

The Binary Mechanism

The second mechanism that is proposed is henceforth called the Binary mechanism. The Binary mechanism uses the TFRC receiver's reports to the sender in order to calculate the packet loss rate percentage. The algorithm considers only a constant number of previous packet losses, so that it is more adaptive to the most recent conditions of the network. This cross-layer mechanism uses information provided by the TFRC protocol which is a transport layer protocol and needs to act upon the physical layer to adjust the transmission power. The parameters involved by each layer include the transmission power at the physical layer, and the packet loss information at the transport layer.

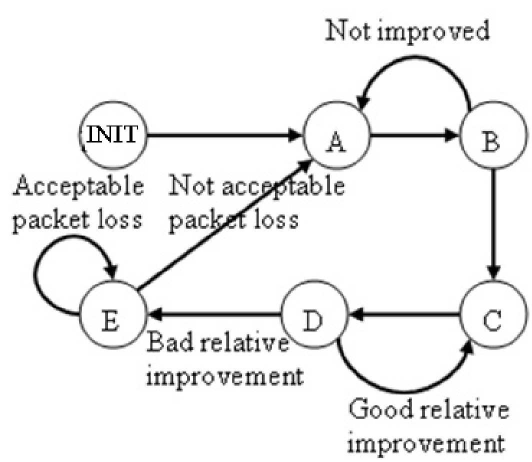
The finite automaton presented in Figure 1 is the mechanism used by the sender of the video via TFRC. Every time the sender receives a TFRC report from the receiver changes its state according to the state it is in and the new data. The mechanism after receiving the first report, if packet loss is not satisfactory, defines a region in which it will try to approximate the optimum power. The optimum power is the one that produces a desired value of packet loss. After defining the region, the sender will increase its power to the maximum possible in that region and send the next TFRC packet with that power (state A). When the sender receives the next report, it tests whether there has been as significant improvement. If there has been an improvement and packet loss is below a predetermined threshold goes to state C or else repeats the actions of state A. In state C, the mechanism sets the power to the middle of the defined region and the sender goes to state D. In state D the algorithm tests whether the packet loss constraints are still satisfied and if this is the case it repeats state C. If this is not the case the algorithm goes to state E where it goes back to the previous known acceptable power value. The mechanism stays at state E while the packet loss value is acceptable, and if not it goes back to state A. Below there is a summary of the states of the automaton:

```
INIT: initializations
A: Expand "power region" and apply region-
maximum power, then goes to state B
B: Improvement and constraint testing. If
qualified, goes to state C, else it goes to
state A
C: Lowers consumption to the middle of the
defined power region and goes to state D
D:If all the constraints are satisfied, goes
to state C, else goes to state E
E: Backtracks to the last known acceptable
power value and stays there while packet
loss is acceptable, else it goes to state A.
```

Testbed Setup

In order to examine the efficiency of the above described mechanisms, the Network Simulator 2 (ns-2.30) was used as a basic tool for simulating multimedia data transmission over wireless networks. The purpose of the experiments was to judge whether TFRC reports could actually

Figure 1. Finite automaton for the proposed mechanism for the sender



lead to better power management techniques in wireless networks.

In order to simulate MPEG-4 video transmission using ns-2, another software package is needed, namely Evalvid-RA (Lie & Klaue, 2008). Evalvid-RA supports rate-adaptive multimedia transfer based on trace file generation of an MPEG video file. A typical trace file provides information for frame number, frame type, size, fragmentation into segments and timing for each video frame. The multimedia transfer is simulated by using the generated trace file and not the actual binary multimedia content. The simulator keeps its own trace files holding information on timing and throughput of packets at each node during simulation. Combining this information and the original video file, Evalvid-RA can rebuild the video file as it would have been received on a real network. Additionally, by using the Evalvid-RA toolset the total noise introduced can be measured (in dB PSNR) as well as Mean Opinion Score (MOS) can be calculated. An example implementation is illustrated in (Haukass, 2007).

Several modifications of the network simulators were needed in order to build a working instance of the proposed mechanism. Firstly, a module that implements the logic of the proposed mechanism was added in the simulator. Then, the module that implements the TFRC protocol was changed so that it provides information about packet losses to our mechanism. The mechanism calculates the power needed to improve PSNR and then this information is passed to the modified wireless physical layer module that is able to increase or decrease power according to the mechanism.

The next sections of the chapter present some experiments that use a simple network topology, consisting of a transmitting and a receiving node. The transmitting node is located behind a wired connection to a router, while the receiving node is connected through the wireless interface of the intermediate router. The akiyo sample video found in Xiph.org was used for video streaming for the purposes of these experiments.

Firstly, the video file was pre-processed and many video files were produced of different quality and resolution using the ffmpeg tool and shell scripts included in the Evalvid-RA toolset. Then, trace files were generated for all these files and by using these trace files the simulation took place. Ns-2 scripts were created to simulate video transmission over a wireless network over TFRC. After simulating the transfer of the video in several different resolutions, ns-2 trace files were obtained which then were used to reconstruct the videos as it would have been sent over a real network. In this phase, several measurements and calculations can be done involving network and video metrics such as PSNR, MOS, jitter, throughput and delay. By using this procedure and another simulation script and algorithm we can make extensive comparisons and reach conclusions about the efficiency of each algorithm.

Performance Evaluation Experiments

In our ns-2 experiments, we transfer H.264 video over TFRC over wireless links, and in particular over a single hop in a wireless ad hoc network. In order to model various instances of network degradation, we have performed experiments where both nodes are stationary, or where the transmitting node remains stationary, while the receiving node moves with steady speed away from the sender. We then compare the achieved throughput in terms of PSNR, packet losses and power consumption. Objective PSNR measurements can be approximately matched to subjective MOS (Mean Opinion Score) according to the standardized Table 1. The MOS scores reported below are derived from the automatic PSNR to MOS mapping according to Table 1.

In the MIMD mechanism the *Lower_Bound* ranged from *0.02* to *0.04* and the *Upper_Bound* from *0.06* to *0.1*. In Experiments 1 and 2 we ran a set of experiments with different *Lower_Bound* and *Upper_Bound* each time in the above range and increasing by *0.01* in each experiment. The results presented in Table 1 are from the average of these experiments.

In order to model various instances of network degradation, we have performed a series of experiments with various scenarios, with both stationary and mobile nodes:

- Scenario 1: Two nodes, both stationary
- Scenario 2: Two nodes, one stationary, one moving away
- Scenario 3: Two nodes, one stationary, one moving closer and then moving away
- Scenario 4: Two nodes, one stationary, one moving closer
- Scenario 5: Two nodes, one stationary, one moving closer and then moving away and then moving closer again
- Scenario 6: Two nodes, one stationary, one moving away and then stops moving
- Scenario 7: Two nodes, one stationary, one moving closer and then stops moving
- Scenario 8: Two nodes, one stationary, one moving randomly

We repeat each scenario three times, one without any power management, one with MIMD power management algorithm and one with Binary power management algorithm. We then compare the achieved throughput in terms of PSNR, packet losses and power consumption. Objective PSNR measurements can be approximately matched to subjective MOS (Mean Opinion Score) according to the standardized Table 1. The MOS scores reported below are derived from the automatic PSNR to MOS mapping according to Table 1.

Table 1. PSNR to MOS mapping

PSNR [dB]	MOS
>37	Excellent (5)
31-37	Good (4)
25-31	Fair (3)
20-25	Poor (2)
<20	Bad (1)

The MIMD method's performance varied according to the values of the thresholds chosen, while the Binary Method is insignificantly susceptible to thresholds' change. The Binary Method's performance however, depends on the initial desired power that one wants to use.

Experiments

We ran the 7 scenarios described above and took the ratio average PSNR over average power per experiment. The purpose is to maximize this ratio as the larger its value the better the performance. Indeed a large value means larger average PSNR or lower average power or both. The Binary method clearly outperforms the MIMD method and the version without mechanism.

We also present a detailed graph for each scenario, and provide trend lines in order to illuminate the behavior of each mechanism under different conditions. It is worthwhile to note that in many cases as shown in the following figures the Binary method achieves Excellent Mean Opinion Score (see Table 2) whereas the other methods achieve at most Good Mean Opinion Score.

In the first scenario, both nodes are stationary, so power requirements do not vary. Nevertheless, power management mechanisms offer a better ratio of PSNR to transmission power. The proposed mechanism proves especially capable in taking advantage of the available transmission power. For a given amount of transmission power, the proposed mechanism significantly outperforms both the MIMD method and the original transmission approach, in terms of achieved video quality (as measured by the PSNR metric). For example, as shown in the graph, for an average available transmission power of about 0.03 all mechanisms achieve a PSNR value of about 34, which is considered good. When however, average available transmission power is doubled, the simple approach yields almost no benefit, the MIMD mechanism achieves a slightly higher PSNR value of 36, while the binary mechanism excels with a PSNR value of over 41 which qualifies as excellent.

The same observations apply also when one of the nodes is moving away. This time, the MIMD mechanism also displays a noticeable performance advantage over the simple approach. The Binary Method converges faster and closer to an optimum value of power needed to decrease packet, therefore achieves better PSNR values for the same average power. This scenario is one of the most beneficial for the proposed MIMD and Binary mechanisms, because the movement is monotonous and they can easily find an excellent tradeoff between energy consumption and video quality.

Because of the increased proximity of the nodes in Scenarios 3 and 4, the simple transmission approach is able to achieve better performance, without however being able to match either the MIMD or the Binary power management approach because of their adjustment of power according to the packet loss. It is interesting to note that in this scenario, the Binary mechanism clearly outperforms the rest implementations even when average transmission power is low, since the variability in the movement of one of the nodes better suits a quickly adapting algorithm.

When a node is moving closer it is natural to achieve a better PSNR value in all methods. By also using rapid adjustment of power even better results occur. Also, the Binary method achieves Excellent Mean Opinion Score for power over 0.04. In this scenario, the performance gain of the MIMD mechanism over the simple one is reduced because as the moving node increases its proximity, less transmission power is required, and therefore a simple implementation can cope.

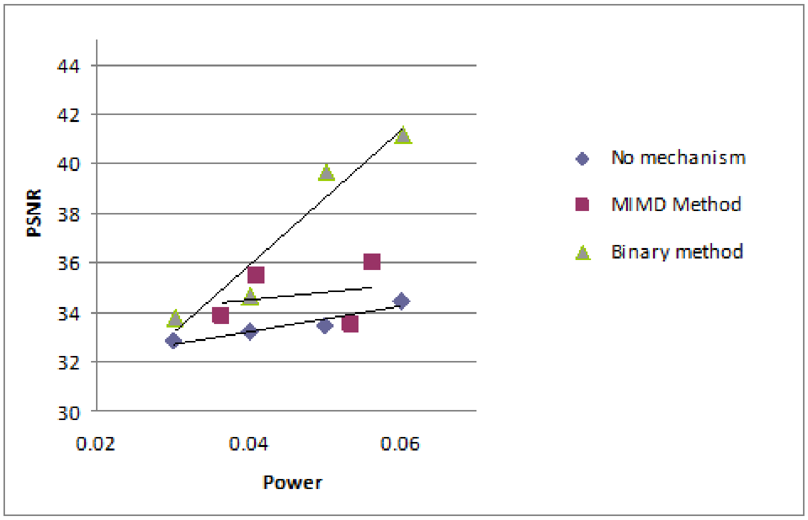
In Scenarios with more complicated movement patterns, the basic conclusion seems to be the same: the proposed Binary approach demonstrates a significant performance lead, which often results in the received video quality to be excellent. The MIMD method provides intermediate benefits, while the original transmission approach without active power management lacks behind both in terms of video quality and consumed power.

In the cases where the nodes stop after moving, the MIMD and Binary methods adjust themselves to be as power saving as possible without making a reduction to the quality of video image transmitted. In fact, for the same PSNR values the MIMD and Binary methods consume less energy than when using no mechanism and when the Binary

Table 2. Scenario results

	Normal	MIMD	Binary
Scenario	**PSNR/Power**	**PSNR/Power**	**PSNR/Power**
1:	669,2	813,1	790,1
2	666,4	769,4	782,5
3:	662,2	759,8	798,8
4:	676,2	798,9	814,8
5:	671,8	800,3	789,7
6:	666,4	769,4	782,5
7:	669,2	813,1	790,1
8:	919,3	902,3	968,4
Average	700,9	803,3	814,6
stddev	88,66	45,06	63,02

Figure 2. Scenario 1: Two nodes, both stationary



method uses power 0.05 and over achieves excellent results.

When one node moves randomly the results show that all mechanisms tend to display similar behavior, for power values up to 0.04. Above this value the Binary method again gains a significant advantage and achieves excellent results. The fact that in this scenario the performance gains are not as pronounced as in previous scenarios can be attributed to the fact that the adaptive methods need some time to adjust (they adjust every time they receive a TFRC report). Random motions tend to quickly change the assumptions upon which adaptive behavior is based. Therefore, the adaptive methods (MIMD, Binary) tend to perform best in situations where there are movement patterns and changes in movement direction occur slower than the round-trip time of a TFRC report.

Figure 3. Scenario 2: Two nodes, one stationary, one moving away

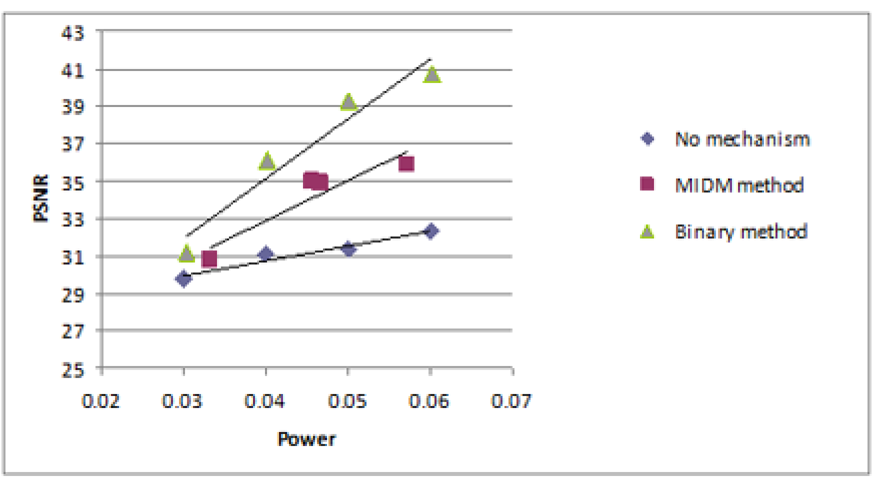
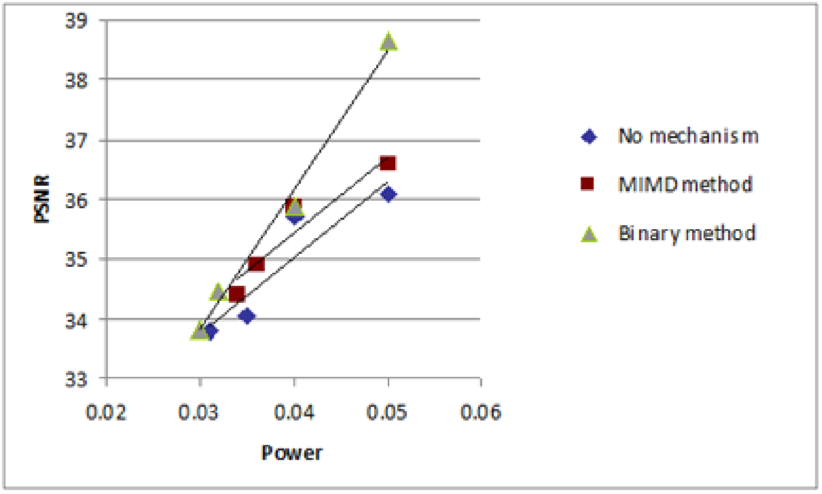


Figure 4. Scenario 8: Two nodes, one stationary, one moving randomly



CONCLUSION

In this chapter we have seen how cross-layer optimization techniques can be used to improve quality and reduce power consumption. More specifically, it has been examined how TFRC can be used for video transmission in wireless networks. The basic aspects of TFRC operation have been presented along with the suitability of TFRC usage for video transmission. Additionally, related work has been presented and two cross-layer mechanisms for wireless transmission via TFRC have been proposed that may be utilized for efficient power management, since the trade-off of power consumption and transmission quality is an important factor for wireless devices.

We have seen that by inserting a simple cross-layer mechanism for power management in wireless TFRC transmission, we can significantly improve both the objective quality of the transmitted video, and make a more optimal usage of available power. The complexity cost of the mechanism is quite small, and slightly larger fluctuations in PSNR measurements seem to be the only remaining trade-off. An advanced power management cross-layer mechanism for power management in wireless TFRC transmission seems to significantly improve both the objective quality of the transmitted video, and makes more optimal usage of available power. A simpler MIMD power management approach also has performance benefits, albeit significantly smaller. The complexity cost of the binary mechanism is relatively small, as the implementation in the ns2 simulator has shown. Ongoing research in this area deals with also taking into account the PSNR metric along with packet loss and using the capabilities of H.264 in order to change video quality dynamically so that there can be adaptation of the transmission rate according to the available bandwidth.

REFERENCES

Anastasi, G., Conti, M., Gregori, E., & Passarella, A. (2004, June). A performance study of power-saving polices for Wi-Fi hotspots. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, *45*(3), 295–318.

Bouras, C., Gkamas, A., Kapoulas, V., Papapanagiotou, V., Stamos, K., & Zaoudis, G. (2009, September). *Video transmission over TFRC using cross-layer power management.* 17th International Conference on Software Telecommunications and Computer Nerworks, Hvar, Croatia.

Chen, H., & Huang, C.-W. (2004). *Power management modeling and optimal policy for IEEE 802.11 WLAN systems*. IEEE Vehicular Technology Conference 2004.

Chen, J., Lv, T., & Zheng, H. (2004, May). *Cross-layer design for QoS wireless communications*. 2004 International Symposium Circuits and Systems, ISCAS '04, 23-26 May 2004 Volume 2, (pp. 217-20).

Evalvid-RA. (n.d.). Retrieved on December 23, 2009, from http://www.item.ntnu.no/~arnelie/Evalvid-RA.htm

FFmpeg project. (n.d.). Retrieved on December 23, 2009, from http://www.ffmpeg.org/

Floyd, S., & Kempf, J. (2004, March). *IAB concerns regarding congestion control for voice traffic in the Internet*. (IETF RFC 3714).

Floyd, S., Kohler, E., & Padhye J. (2006, March). *Profile for datagram congestion control protocol (DCCP) congestion control ID 3: TCP-friendly rate control* (TFRC). (RFC 4342).

Handley, M., Floyd, S., Pahdye, J., & Widmer, J. (2003, January). *TCP friendly rate control (TFRC): Protocol specification.* (IETF RFC 3448).

Haukass, T. (2007). *Rate adaptive video streaming over wireless networks explained & explored.* MSc thesis, Norwegian University of Science and Technology, Department of Telematics.

IEEE 802.11. (2009). *PSM standard. Power management for wireless networks, section 11.11.2*.

ISI. (2009). *ns2 network simulator*. Retrieved on December 23, 2009, from http://www.isi.edu/nsnam/ns/

Jones, C. E., Sivalingam, K. M., Agrawal, P., & Chen, J. C. (2001, August). A survey of energy efficient network protocols for wireless networks. *Wireless Networks*, *7*(4), 343–358. doi:10.1023/A:1016627727877

Kim, I.-M., & Kim, H.-M. (2003, January). An optimum power management scheme for wireless video service in CDMA systems. *IEEE Transactions on Wireless Communications*, *2*(1).

Kohler, E., Handley, M., & Floyd, S. (2006, March). *Datagram congestion control protocol* (DCCP). (IETF RFC 4340).

Lee, S., & Chung, K. (November, 2006). TCP-friendly rate control scheme based on RTP. *Information Networking. Advances in Data Communications and Wireless Networks*.

Li, X., Wu, X., Li, W., & Wang, X. (2008). *An adaptive cross-layer scheduling algorithm for multimedia networks*. 4th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP 2008, art. no. 4604006, (pp. 52-55).

Lie, A., & Klaue, J. (2008, June). Evalvid-RA: Trace driven simulation of rate adaptive MPEG-4 VBR video. [Berlin/Heidelberg, Germany: Springer.]. *Multimedia Systems*, *14*(1), 33–50. doi:10.1007/s00530-007-0110-0

Lin, X., Shroff, N. B., & Srikant, R. (2006, August). A tutorial on cross-layer optimization in wireless networks. *Selected Areas in Communications* . *IEEE Journal*, *24*(8), 1452–1463.

Molta, D. (2005, December). Wi-Fi and the need for more power. *Network Computing*.

Radunovic, B. (2005, July). A cross-layer design of wireless ad-hoc networks. Ph.D. Thesis.

Ramakrishnan, K., Floyd, S., & Black D. (2001, September). *The addition of explicit congestion notification (ECN) to IP.* (IETF RFC 3168).

Shan, Y., & Zakhor, A. (2002). *Cross layer techniques for adaptive video streaming over wireless networks*. IEEE International Conference on Multimedia and Expo, volume 1, (pp 277-280).

Simunic, T. (2005). *Power saving techniques for wireless LANs.* Conference on Design, Automation and Test in Europe - volume 3 (pp. 96-97).

Stewart, R. (2007, September). *Stream control transmission protocol.* (IETF RFC 4960).

Tsaoussidis, V., & Badr, H. (2000 November). *TCP-probing: Towards an error control schema with energy and throughput performance gains.* 8th IEEE Conference on Network Protocols, Japan.

van der Schaar, M., & Sai Shankar, N. D. (2005, August). Cross-layer wireless multimedia transmission: Challenges, principles and new paradigms. *IEEE Wireless Communications*.

Warrier, A., Le, L., & Rhee, I. (September, 2007). *Cross-layer optimization made practical*. 4th International Conference on Broadband Communications, Networks, Systems, BroadNets, art. no. 4550507, (pp. 733-742).

Xiph.org. (n.d.). *Test sequences*. Retrieved on December 23, 2009, from http://media.xiph.org/video/derf/

Zamora, N. H., Kao, J.-C., & Marculescu, R. (2007, April). *Distributed power-management techniques for wireless network video systems*. Design, Automation & Test in Europe Conference & Exhibition, (pp. 1-7).

Zhang, C., & Tsaoussidis, V. (2001). *TCP real: Improving real-time capabilities of TCP over heterogeneous networks.* 11th IEEE/ACM NOSSDAV, NY.

ADDITIONAL READING

Fu Z., Meng X., & Lu S. (2004, December). "A Transport Protocol For Supporting Multimedia Streaming in Mobile Ad Hoc Networks", IEEE JSAC.

Klues, K. (2006). *"Power Management in Wireless Networks", Report, for Advanced Topics in Networking: Wireless and Mobile Networking by R*. Jain, Wash.: Univ. in St. Louis.

Li M., Lee C.-S., Agu E., Claypool M., & Kinicki R (2004, September). "Performance Enhancement of TFRC in Wireless Ad Hoc Networks", Distributed Multimedia Systems (DMS).

Lochin, E., Dairaine, L., & Jourjon, G. (2006). "gTFRC: a QoS-aware Congestion Control Algorithm, International Conference on Networking (ICN)

Rulnick, J. M., & Bambos, N. (1997). Mobile power management for wireless communication networks . *Wireless Networks*, 3. doi:10.1023/A:1019168125436

Tsao S.-C., Lai Y.-C., & Lin Y.-D. (2007, November). "Taxonomy and Evaluation of TCP-Friendly Congestion-Control Schemes on Fairness, Aggressiveness, and Responsiveness", IEEE Network.

Vojnovic, M., & Le Boudec, J.-Y. (2005). "On the Long-Run Behavior of Equation-Based Rate Control", ACM SIGCOMM 2002, also . *IEEE/ACM Transactions on Networking*, 2005.

Welzl M. (2005, June). "User-Centric Evaluation of TCP-friendly Congestion Control for Real-Time Video Transmission", Elektrotechnik und Informationstechnik.

Yan J., May M., Katrinis K., & Plattner B. (2005). "A New TCP-Friendly Rate Control Algorithm for Scalable Video Streams", NETWORKING 2005.

Yan, J., Zheng, X., Liu, J., & Li, J. (2008). *Wireless Loss Detection for TCP Friendly Rate Control Algorithm in Wireless Networks*. Information Networking. Towards Ubiquitous Networking and Services.

KEY TERMS AND DEFINITIONS

B-Frame: A video frame used in a video compression algorithm that requires both previous and forward frames for its decoding (used in MPEG video standards).

Cross-Layer: The method of a network systems design that takes into consideration multiple aspects of system operation across various layers in order to improve functional efficiency.

H.264/MPEG-4: A modern standard for video compression that aims at flexibility and providing good video quality at lower bit rates than previous compression methods.

I-Frame: A video frame used in a video compression algorithm that does not require other frames for its decoding (used in MPEG video standards).

P-Frame: A video frame used in a video compression algorithm that requires previous frames for its decoding (used in MPEG video standards).

TCP-Friendliness: The characteristic of a flow that does not create network congestion so that competing TCP flows are not squeezed out of bottleneck links.

TCP-Friendly Rate Control (TFRC): A: congestion control protocol for TCP-friendly flows that achieve better throughput than TCP.

Wireless: Transmission of information without the usage of wires.

APPENDIX

Table of abbreviations

AIMD	Additive Increase Multiplicative Decrease
AMC	Adaptive Modulation and Coding
AP	Access Point
ARQ	Automatic Repeat Request
AV	Audio-Video
AVC	Advanced Video Coding
DCCP	Datagram Congestion Control Protocol
FEC	Forward Error Correction
FGS	Fine Granularity Scalability
MAC	Medium Access Control
MIMD	Multiplicative Increase, Multiplicative Decrease
MOS	Mean Opinion Score
PL	Packet Loss
PSM	Power Saving Mode
PSNR	Peak Signal-to-Noise Ratio
RLP	Radio Link Protocol
RTT	round-trip time
SNR	Signal-to-Noise Ratio
SVC	Scalable Video Coding
TCP	Transmission Control Protocol
TFRC	TCP-Friendly Rate Control
TP	Transmission Power