Improved SNR for High Data Rate Transmission

D Sureesh, P Kalyanchakravarthi, K Vijayakumar
Department of ECE, GMR Institute of Technology, Rajam, India.

***Abstract*—** Rapid growth of communication improvement needed high data transfer for video transmission. While videos are transmitting through wireless lot of errors occurred in encryption and decryption stages. To overcome that we are going to propose a protocol involved in transmission. In this paper we are discussing about various protocol comparison with proposed protocol. This report gives an overview of different technologies, advancements, advantages and comparison among them. Initially amplitude modulation mobile communication systems were developed and then frequency modulation is used in mobile communication system. Here each system covers distance of over 50km. It uses half duplex mode of operation. It also uses large RF bandwidth. Later a cellular system based on code division multiple access technique is used. Wireless networks replace fiber optic cables and copper lines used for wired networks. That is replacement of wires. Bluetooth is an example of wireless connections. In this paper we are going describe about classifications of generations .And discussed about optimized wireless video transmission protocol.

Keywords- SNR, High Data Rate Transmission

I. 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[1,2]. This approach reduces the complexity and makes issues more manageable and architectures more flexible and upgradeable, 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[3].

Although his 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.[4] 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 at Schaar & Shankar (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.[5] 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.

II. 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[6]. 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.[7] 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.[8,9] 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:

$$X_{TFRC} = \frac{s}{R^{\frac{2bp}{3}} + t\_RTO^{\frac{3bp}{8}} + 32p^2}$$

- XTFRC 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

III. PROBLEM FORMULATION

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.[10,11] 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. [12]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 behaviour 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:
t_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 that 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.

IV. SIMULATION RESULTS

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 behaviour in environments with a low degree of statistical multiplexing it is useful to modify sender’s transmit rate to provide congestion avoidance behaviour 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.

A. 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 below 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

TABLE 1: PSNR TO MOS MAPPING

PSNR [dB]	MOS
>37	Excellent
31-37	Good
25-31	Fair
20-25	Poor
<20	Bad

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 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

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.

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 behaviour 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 1) whereas the other methods achieve at most Good Mean Opinion Score.

CONCLUSION

We also present a detailed graph for each scenario, and provide trend lines in order to illuminate the behaviour 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 1) whereas the other methods achieve at most Good Mean Opinion Score.

REFERENCES

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

[2] 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.

[3] Zhang C., & Tsaoussidis V. (2001). "TCP Real: Improving Real-time Capabilities of TCP over Heterogeneous Networks" *11th IEEE/ACM NOSSDAV*, NY.

[4] 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*, Volume 7, Issue 4, pp. 343-358.

[5] IEEE 802.11 PSM Standard. Power Management for Wireless

[6] Networks. Section 11.11.2.

[7] Molta D. (2005 December). "Wi-Fi and the need for more power". *Network Computing*.

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

[9] 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*. Volume 45 , Issue 3, pp. 295-318.

[10] Simunic T. (2005). "Power Saving Techniques for Wireless LANs", *Conference on Design, Automation and Test in Europe* - Volume 3. pp. 96 -97.

[11] 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.

[12] 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. II- 217-20.

[13] Lin X., Shroff, N.B., & Srikant, R. (2006 August). "A tutorial on cross-layer optimization in wireless networks", *Selected Areas in Communications, IEEE Journal*, Volume: 24, Issue: 8,On page(s): 1452-1463.