Rate Adaptation for Conversational 3G Video

Varun Singh, Jörg Ott
Helsinki University of Technology (TKK)
Espoo, Finland
Email: {varun,jo}@netlab.tkk.fi

Igor D.D. Curcio
Nokia Research Center
Tampere, Finland
Email: igor.curcio@nokia.com

***Abstract*—Wireless cellular environments, such as UMTS, are often affected by congestion and errors, which are inherent to wireless transmission channels due to fading, interference, resource scarcity, mobility, etc. For a conversational video application to be successful i.e., to provide good viewing quality to the receiver at all times, the sender must be able to quickly adapt its sending/encoding rate (and other related parameters) to that offered by the link. Moreover, for a rate adaptation scheme to be successful, the receiver must provide timely feedback in order to mitigate further losses due to congestion. In this paper, we investigate different rate adaptation mechanisms and redefine them for 3GPP networks, reusing existing RTCP extensions standardized in the IETF and in 3GPP where possible.**

I. Introduction

The third generation mobile system provides conversational video communication in the *Media Telephony Service for IMS* (MTSI) [1]. This 3GPP standard supports the use of H.264/AVC [2] encapsulated in RTP for carrying video traffic. A typical conversational mobile multimedia system, such as MTSI, require that end-to-end delays do not exceed values in the order of 400ms [3] for providing acceptable media quality for playback and a good user experience. Fading, interference, mobility, handovers, cell loading and other factors often cause the available bandwidth for each user to fluctuate, which causes congestion in the network. Moreover, packet losses may occur due to radio effects causing bit errors, congestion-induced drops from router queues, and packets discarded due to late arrival at the receiver. Since packet losses are detrimental to video quality perception and expensive to repair, they need to be avoided as much as possible.

Mobile multimedia applications thus need to adapt to the bandwidth constraints by adjusting their encoding and/or transmission rate. However, congestion control in wireless 3G networks for conversational video applications is challenging because the application-defined maximum delay (400ms) and the minimal network-incurred latency leave only very little room for a congestion control algorithm to operate. Traditional congestion indicators such as packet losses are not applicable because 1) air interface losses and congestion losses may be hard to differentiate and, more importantly, 2) increased queuing delays in the network may cause the receiver to discard packets even before congestion losses occur. Therefore, a sender has to anticipate upcoming congestion from various cues—including but not limited to the per-packet delay used in many delay-based congestion control algorithms—to prevent network queues from building up in the first place. This requires extreme sensitivity to the reported transmission characteristics.

In this paper, we choose a suitable operating environment as defined by the 3GPP in [3] to help evaluate the performance of our new algorithm and our enhancements of existing rate adaptation signaling schemes against those already defined like TCP Friendly Rate Control (TFRC) [4] [5], [6], Temporary Maximum Media Stream Bit rate Request (TMMBR) [1], [7]. We introduce these and discuss them and the related work in section II. In section III we introduce our new rate adaptation algorithm and explain the features and configuration of the simulation environment in section IV. Section V presents the results, compares them against each other and we draw conclusions and discuss directions for future work in section VI.

II. Overview of Rate adaptation techniques

The decision making process of rate adaptation can be made at the sender, the receiver, or at some intermediate node (edge or core) in the network. Sender-driven rate adaptation requires that the receiver be aware of the current network situation i.e., latency experienced by a packet, current jitter buffer state at the receiver, current decoding rate, packets lost, etc., and signal this information to the sender which decides to adapt the rate based on the received parameters. In a receiver-driven rate adaptation scheme, the receiver gauges the current situation based on the parameters available to it, and signals the new required bandwidth to the sender that, on receiving the new rate, adapts to it. In a network-driven rate adaptation, an element in the mobile network will signal to the sender/receiver that the rate is going to drop or increase due to better or worse network conditions arising from handovers, cell-loading, etc. In these cases the network is aware of the conditions beforehand and can therefore signal to the appropriate node the new data rate.

TCP Friendly Rate Control (TFRC) is an equation based congestion control algorithm implemented at the sender [4] and is a profile in the Datagram Congestion Control Protocol (DCCP) to fairly compete for bandwidth with other flows. TFRC uses knowledge at the sender to calculate the new bandwidth based on average packet size, RTT, loss-rate [8]. [5] extends [6] for multimedia applications by using RTP/RTCP feedback loop to control the algorithm and redefines the timing rules in [9] for very short RTTs ($< 20ms$).

Temporary Maximum Media Bit-rate Request (TMMBR) / Temporary Maximum Media Bit-rate Notification (TMMBN):

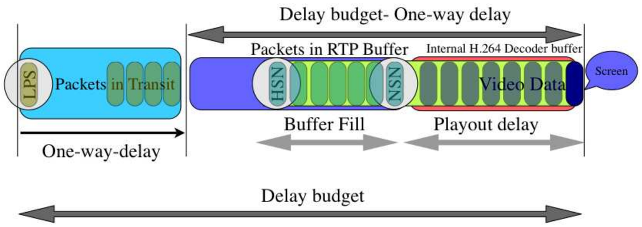


Fig. 1. Receiver-side Queuing model

In addition to the Feedback Control Information defined in RFC4585 [9], RFC5104 [7] (codec control messages for AVPF) defines several more codec-related feedback messages such as the TMMBR and TMMBN. TMMBR is generated by the receiver in a point-to-point (PtP) scenario and is sent by the receiver to request the sender to limit its maximum bandwidth to that value i.e., the sender may choose the value received in TMMBR or a lower value. TMMBN is a notification sent by any entity (sender, receiver, network) to the other to notify the bounding rate it is using.

Next Application Data Unit (NADU) for Streaming video: The maximum delay budget (400ms) and the the minimal network latency provide a small opportunity for the receiver to queue packets for a very short period of time. For example, in the 3G Simulation framework [3] the system allows a maximum $400ms$ delay budget (from video capture to display) and $240ms$ static one way delay. If there is no congestion due to queuing at the intermediate nodes in the core network then the maximum time the receiver can cache is $max\_cache\_time = 400 - 240 = 160ms$. This potentially means that the receiver can queue upto 2-3 frames of a 15 fps video stream). NADU is a signaling mechanism which intimates the sender of Playout delay of the first packet in the RTP queue and its sequence number [10], [11]. NADU is already defined in 3GPP [12] for the video streaming scenario where it provides the sender with playback buffer information. Figure 1 shows the receiver side queuing model and some of the terms associated with the signaling of NADU, like Highest Sequence Number (HSN), Next Sequence Number (NSN), buffer fill-level, number of Packets in Transit (PiT), number of Packets in the Buffer (PiB), Last RTP packet sent from the sender (LPS) just before receiving the RR, Playout Delay, etc.

III. New Rate Adaptation Schemes

In this section, we introduce two rate adaptation algorithms for conversational video. The first one consists of a superset of NADU signaling based on the RTCP XR Discard Metrics packet reports and ordinary RTCP RR information. Also a new sender-side algorithm is described. We will denote algorithm and signaling as C-NADU, which stands for Conversational NADU. The second one consists of variants of TMMBR that couple Uplink information and Downlink information coming from the network. TMMBR using both Uplink and Downlink information will be denoted in the following as TMMBR-A, using only downlink information will be denoted as TMMBR-B, and unassisted TMMBR will be denoted as TMMBR-U. Finally, reactive scheduling of RTCP reports is also a way to improve system performance, and is part of our solution.

A. Conversational NADU (C-NADU)

We define two modes of operations for this new sender-side rate adaptation scheme: Congestion Avoidance and Congestion Mitigation. In congestion avoidance, the sender (or receiver) tries to detect if the link is undergoing light congestion and, based on the input, it slightly increases or decreases the sending rate. For example, slight reductions or increases in round-trip time (RTT), jitter, packets in transit (PiT), etc. can be indicators for light congestion or under-utilization. However, in the case of congestion mitigation, the rate adaptation module realizes that there is already heavy congestion, and needs to take a corrective action immediately. For example, high packet loss might provide indication for the presence of heavy congestion. Moreover, one might also associate that in congestion avoidance only small changes to the bandwidth might occur. However, in congestion mitigation more drastic changes might be made to mitigate the congestion. The rate adaptation algorithm takes input from many parameters signaled from the receiver to the sender via various extensions defined for RTCP, namely:

- Normal RTCP Receiver Report (RR) [13].
  - Fraction Loss (FL)
  - Inter-arrival Jitter ($Jitter$)
  - Calculated RTT (RTT)
  - Highest Sequence Number, (HSN)
- NADU Packet [12] reports
  - Next Sequence Number (NSN) is the RTP sequence number of the next packet to be decoded from the receiver queue. If no packets are available for playout then, $NSN = HSN + 1$ (this packet has not been received by the receiver yet).
  - Playout Delay of NSN (PD_{NSN}) is the difference between the scheduled playout time of the NSN packet and the time the receiver sends the RTCP report [12]. If no packets are available for playout then the receiver can signal $PD_{NSN} = 0 \times FFFF$.
- RTCP XR Discard Metrics [14] packet reports
  - the number of bytes discarded ($bytes_{discarded}$) related to the packets dropped at the receiver due to late arrival of packets.

In addition to the above signaling information, the sender maintains a ring buffer with the size of all video packets sent since the last RR received. Figure 1 describes the receiver side queuing model and visually represents some of the entities received in the RR and local state information.

Using the aforementioned parameters the sender is able to calculate the following:

1) PiT = $LPS - HSN$ packets.
2) Receiver Buffer Fill-level, if $NSN < HSN$

- number of Packets in Receiver Buffer, $PiB = (HSN - NSN) + 1$ packets.
- Time to drain the buffer, $Buffer\ Filllevel_{in\_ms} = RTP\_TS_{HSN} - RTP\_TS_{NSN}$ ms.
- size of packets from ring-buffer, $Buffer\ Fill\text{-}level_{in\_bytes} = \sum_{i=NSN}^{HSN} sizeof(i)_{in\_bytes}$ bytes.

3) Playout Delay experienced by the HSN if no underflow or losses occurs, $PD_{HSN} = PD_{NSN} + Buffer\ Fill\text{-}level_{in\_ms} + \frac{RTT}{2}$[1] ms.
4) Perceived Receiver Rate at the sender, $ReceiverRate_{perceived}(inkbps) = \frac{\sum_{i=HSN_{last\_RR}}^{HSN+PiT} sizeof(i)_{in\_bytes} \times (1.0 - FL) \times 8}{1000 \times (t_{now} - t_{last\_RR})}$
5) Perceived Receiver Goodput, the perceived rate that was played back,

$$Goodput_{perceived}(inkbps) = \frac{((\sum_{i=HSN_{last\_RR}}^{HSN} sizeof(i)_{in\_bytes} \times (1.0 - FL)) - bytes_{discarded}) \times 8}{1000 \times (t_{now} - t_{last\_RR})}$$

In addition to the ring-buffer, the sender keeps a short history of some of the above parameters, namely PiT, PiB, Jitter, and RTT by calculating the correlation of the current value with the moving average of the last 3 values or the $90^{th} - percentile$ values of lossless reports.

1) Correlated RTT, by using the $90^{th} - percentile$ value of all loss-less RTTs it is possible to calculate the correlation of the current RTT, $CorrRTT = \frac{90 - percentile_{lossless}(RTT)}{RTT_{now}}$
2) Correlated PiT and PiB are calculated to ascertain if the queues in the network and at the receiver are increasing or decreasing. $CorrPiT = \frac{PiT_{avg\_last3}}{PiT_{now}}$ and $CorrPiB = \frac{PiB_{avg\_last3}}{PiB_{now}}$

In the Algorithm 1: line 15, $\delta_{undershoot}$ is calculated only for the first loss event of a new downward trend, and is done to quickly mitigate congestion because of higher rate packets in transit and $lines\ 9,\ 22,\ and\ 45$ use constants (α, β, Ψ) when no conclusive information is available in cases of extreme congestion, or underflow.

B. TMMBR-A, TMMBR-B and TMMBR-U

In TMMBR-A, the network notifies the sender and receiver of the Uplink and Downlink rates respectively. The sender is aware of the downlink capacity, but this information arrives at the sender delayed by an order of a one-way delay from the receiver. However, the downlink may not be the constraining link. Therefore, the sender receives also information about the Uplink rate. In TMMBR-B, the network notifies the receiver of the Downlink rate. As before, the sender is notified about the current downlink capacity by the receiver; however the sender is not aware of the Uplink rate. Hence, the TMMBR messages from the receiver are considered as an upper bound for the current encoding rate and the TMMBR message rate is never exceeded.

In TMMBR-U, the network does not assist the sender nor the receiver. The receiver sends the new bandwidth request to

Algorithm 1 Sender-side Rate Adaptation Algorithm

```
Require: Encoder maintains a ring-buffer with sizes of packets sent since the HSN of Last RR
Ensure: Reception of Latest RR from receiver
    Parse $(RR) \Rightarrow (RTT_{now}, Jitter, FL, HSN_{now})$
 2: if available, Parse $(NADU) \Rightarrow (NSN, PD_{NSN})$
    if available, Parse (RTCP XR Discard Metric) $\Rightarrow (bytes_{discarded})$
 4: Calculate $PiB_{now}, PiT_{now}, CorrRTT, CorrPiT, CorrPiB$ and $ReceiverRate_{perceived}, GoodPut_{perceived}, PD_{HSN}$
 6: if $(HSN_{now} = HSN_{last\_RR})$ then
      //No Packets were received!
 8:   $NewBw \leftarrow CurrentBw \times \alpha;$
      $\forall\ \alpha \in (0, 1)$, we use $\alpha = 0.5$
10: else
      if $((FL > 0)||(bytes_{discarded} > 0))$ then
12:     //Congestion mitigation!
        if $(CurrentBw > GoodPut_{perceived}$ then
14:       $NewBw \leftarrow GoodPut_{perceived} \times \delta_{undershoot}$
          $\forall \delta_{undershoot} \in (0, 1]$
16:     else
          //High congestion!
18:       if $(CorrRTT < 1.0)$ then
            $NewBw \leftarrow CurrentBw \times CorrRTT$
20:       else
            $NewBw \leftarrow CurrentBw \times \beta$
22:         $\forall\ \beta \in (0, 1)$, we use $\beta = \frac{\sqrt{2}}{2}$
          end if
24:     end if
      else
26:     //Congestion Avoidance!
        if $(CorrPiT < 1.0)$ then
28:       $NewBw \leftarrow CurrentBw \times CorrPiT$
        else if $(CorrPiB < 1.0)$ then
30:       $NewBw \leftarrow CurrentBw \times CorrPiB$
        else if $((CorrPiT > 1.0) AND (CorrPiB > 1.0))$ then
32:       $NewBw \leftarrow CurrentBw \times corrPiT$
        end if
34:     if $(PD_{HSN} \neq 0 \times FFFF)$ then
          $NewBw \leftarrow CurrentBw \times \frac{PD_{max}}{PD_{HSN}}$
36:       $\forall\ PD_{max} = 400ms$
        else
38:       //Underflow!
          if $(CurrentBw < ReceiverRate_{perceived})$ then
40:         $NewBw \leftarrow ReceiverRate_{perceived}$
          else if $(CorrRTT > 1.0)$ then
42:         $NewBw \leftarrow CurrentBw \times CorrRTT$
          else
44:         $NewBw \leftarrow CurrentBw \times \Psi;$
            $\forall\ \Psi \in (1, 2)$, we use $\Psi = 1.1$
46:       end if
        end if
48:   end if
    end if
```

[1] one-way delays are presumed to be symmetric even though video data is flowing only in one direction which makes the delays asymmetric.

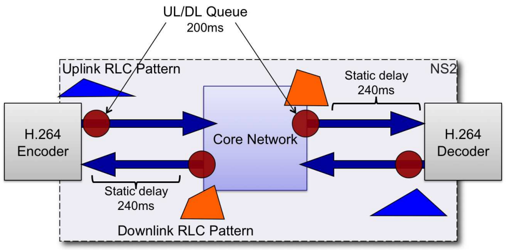


Fig. 2. Simulation environment

the sender using TMMBR based on the average inter-arrival time of RTP packets between two RTCP RRs. The receiver also enhances the performance of TMMBR in all scenarios (TMMBR-A, TMMBR-B and TMMBR-U) by signaling the number of discarded bytes [14] to the sender as it helps in undershooting and thus temporarily alleviating the stress on the network queues. Due to link induced losses, the sender implements some light congestion avoidance techniques based on increasing RTT, discarded bytes [14] and packet loss.

C. Reactive scheduling of RTCP RRs from receiver to sender

RFC4585 [9] allows throttling of RTCP to 2.5% of the available bandwidth to each end-point in a point-to-point scenario, which is quicker than the $5 \pm 2.5sec$ restriction described in [13]. [15] describes that sending feedback at every 200ms or up to 380ms helps in quicker adaptation to congestion but uses non-compound RTCP [16] to conserve RTCP bandwidth. However, we do not use non-compund RTCP [16] reporting, as normal RTCP packet contains essential information such as RTT, HSN, Jitter, etc. to the sender.

For reactive scheduling of RTCP we consider the bad packet rate which takes into account both the lost and discarded packets at the receiver. We define a threshold of between 20-30% to reduce the feedback rate by half. However, while reducing the feedback rate we limit our lower-bound to the minimum RTCP interval set by [9].

IV. Simulation Environment

Our simulation environment is built using ns2 [17] for the core network. The simulator interfaces with the Nokia H.264 codec [18] so that the rate adaptation algorithms can be evaluated in a real-world setting. We have extended ns2 as described in [19] to provide real-time exchange of RTP/RTCP messages between the codec and ns2 by routinely synchronizing their clocks. Furthermore, the receiver RTP layer is extended to generate feedback messages while the sender is extended to respond to them. Fig 2 represents an overview of the simulation environment. The sender/receiver generate video data encapsulated in RTP packets [13]. The decoder generates RTCP feedback based on [7], [13], [14] [20], and conforms to the timing rules described in RFC4585 [9]. Furthermore, the 3G core network is presumed to be a well provisioned error-free network. The four 3G Links are used as access links between the codec and core network. The 3G links conform to the behavior described in [21]. The Radio Link Control (RLC) [22] frame sizes and their scheduling controls the amount of data (inclusive all headers) that can flow on the 3G links. The RLC frame sizes and scheduling opportunity of the frames conform to those defined by 3GPP for evaluation of rate adaptation [3].

There are four different RLC pattern files. Two for the sender side: uplink (UL) / downlink (DL); and two for the receiver side: uplink/downlink. The simulation environment can also produce 0.5% to 1.5% link layer losses (3G Link) using error patterns defined in [21]. To simulate the 0.5% losses, the RLC frames [3] are further broken down into 40-byte frames and sent over the 3G link. If a 40-byte frame is dropped, reconstruction of the associated IP packets fails, therefore, 0.5% loss rate may cause higher IP layer packet loss [19]. It should also be noted that no header compression was used over the 3G links.

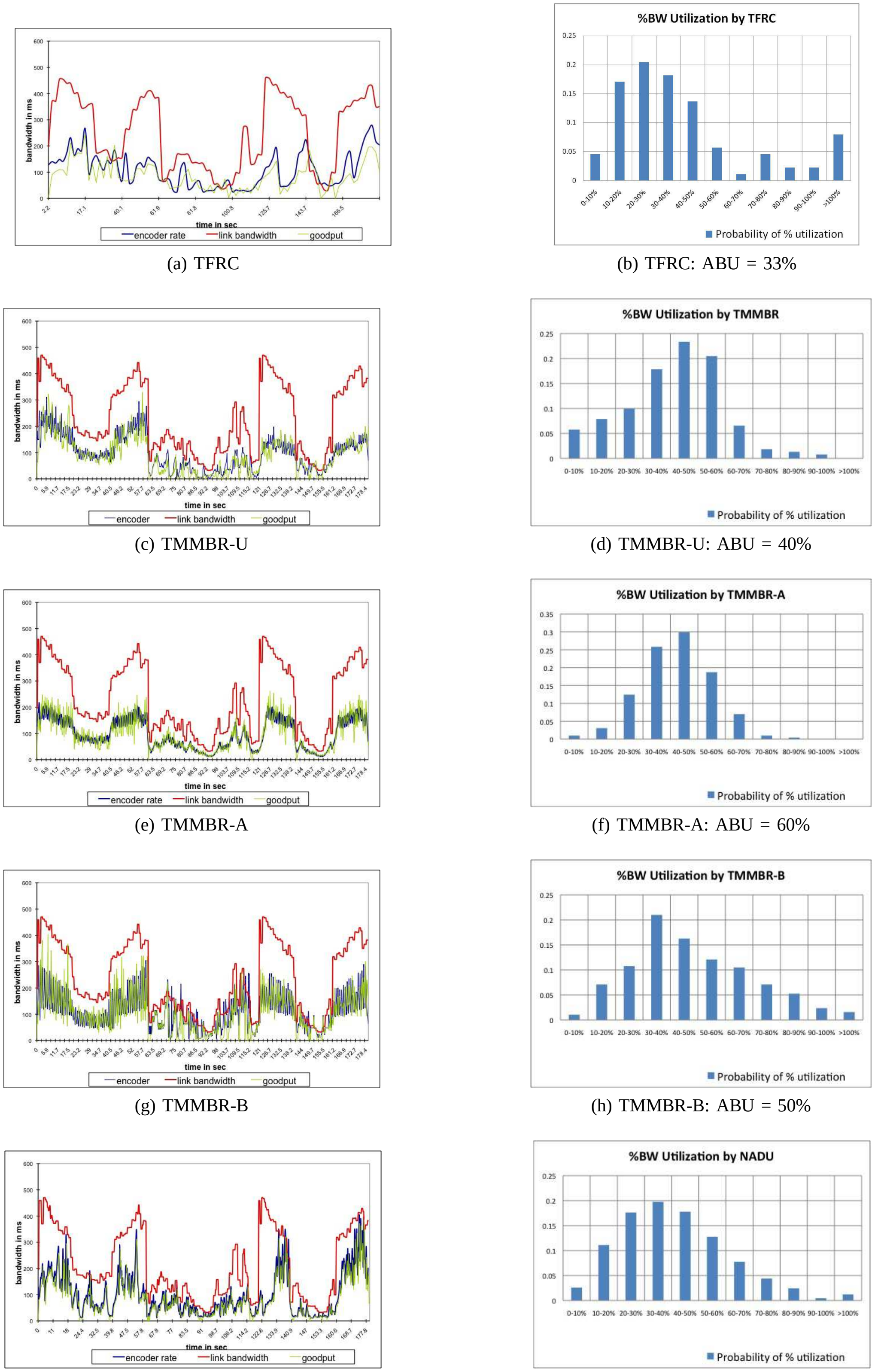
The uplink and downlink queues in the network are long queues with $200ms$ time-to-live for a packet in the queue. Therefore, only complete IP packets are transmitted through the core network. Apart from the queuing delay caused by the RLC scheduling of each packet at the UL/DL queues, the packets are queued for a further $240ms$ as static one-way delay just before they are delivered to the receiver. Instead of using fixed packet sizes as described in [3], we use a medium motion media sequence ("Foreman" QCIF sequence) encoded at $15frame/second$ and the sender encapsulates $1frame/IP\ packet$ (for simplicity, even though the H.264 codec [18] supports slicing of frames). Furthermore, in all scenarios the sender begin with an initial sending rate of 128 kbps and are not restricted by a maximum encoding rate.

We have chosen two types of scenarios to evaluate the rate adaptation scheme. The first is a highly dynamic 3G link based on the 3G traces [3]; the sender's uplink is a concatenated pattern based on excellent, poor, and elevator scenarios (60s each) while the receiver uses the elevator RLC pattern file concatenated three-times. The second is a more stable link with slowly changing links with link bitrate changing at 0, 20, and 40 seconds to 192, 96, 128 kbps respectively at all links. The second scenario is chosen to test the stability of the algorithms. In the case of TMMBR, bandwidth updates are generated at the end of every $1s$ interval in the dynamic scenario (by averaging the available RLC bytes in that interval) and in the more stable scenario it is generated every time the bandwidth changes.

V. Performance Evaluation

TFRC is implemented as defined in [8]. Not all scenarios and extensions mentioned in [6] were developed. However all extensions suggested in [5] were implemented along with the signaling mechanism for conveying loss event rate, timestamp of the last received packet and current decoding rate at the receiver (TFRC-FB). TFRC-FB is sent along with each RR and is sent every $500ms$.

For TMMBR, we introduce three cases: TMMBR-A, TMMBR-B and TMMBR-U (see sec. III-B). In TMMBR-A and TMMBR-B the network assists the sender or receiver



(a) TFRC

(b) TFRC: ABU = 33%

(c) TMMBR-U

(d) TMMBR-U: ABU = 40%

(e) TMMBR-A

(f) TMMBR-A: ABU = 60%

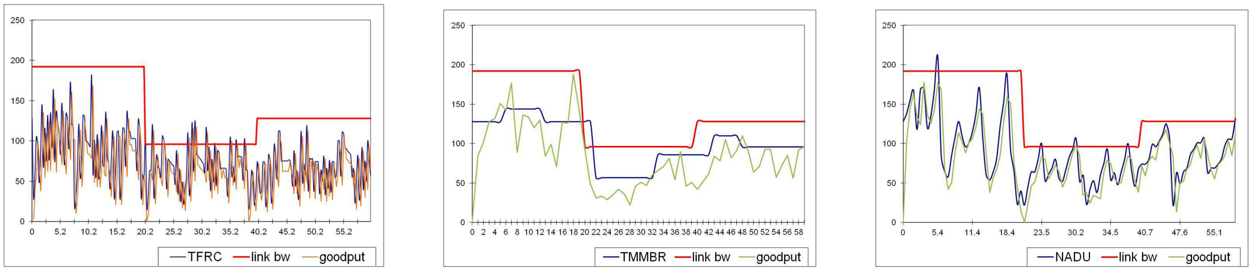
(g) TMMBR-B

(h) TMMBR-B: ABU = 50%

(i) NADU-C

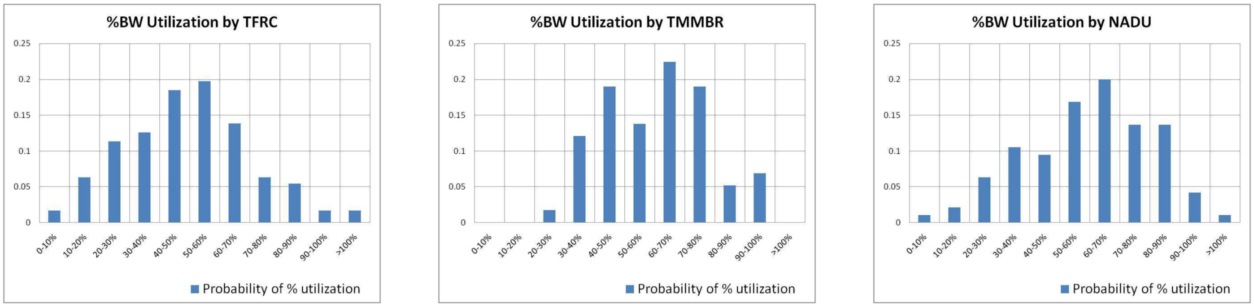
(j) NADU-C: ABU = 55%

Fig. 3. Plot of Link rate, encoder rate, goodput (left column) and Histogram of Probability of per-instance %Utilization (right column) and Average BW Utilization (ABU) of Dynamic 3G Links



(a) TFRC (b) TMMBR-A (c) NADU-C

Fig. 4. Plot of Link rate, encoder rate, goodput in a stable and slow bandwidth changing scenarios.



(a) TFRC: ABU = 40% (b) TMMBR-A: ABU = 70% (c) NADU-C: ABU = 60%

Fig. 5. Histogram of Probability of per-instance %Utilization of the stable, slow bandwidth changing links

or both. In TMMBR-U there is no network assistance, and the receiver notifies the sender with a recommendation for the sending rate based on losses or increase in inter-arrival time of packets. In all the cases, the receiver signals the number of bytes discarded to the sender. Furthermore, due to link induced losses, the sender implements some light congestion avoidance techniques based on increasing RTT and packet loss. We do not run simulations for TMMBR-B in the second scenario (slowly changing bandwidth) because the uplink and downlink traces in this scenario are exactly the same. Therefore, adaptation of TMMBR-B follows that of TMMBR-A.

C-NADU uses the algorithms describes in section III and the signaling defined in [12]. The NADU feedback is sent with every RTCP RR, even if the buffer is empty due to no new packets arriving or underflow. However, the bytes discarded extension [14] is only sent by the receiver when it actually discards packets due to late arrival. Feedback messages are sent every $500ms$ except when interval losses exceed 30% after which the RRs are sent every $250ms$, however the SR sending rate is not affected and is sent at $500ms$.

Figures 3 (left column) and 4, show the instantaneous variation of the encoder rate and decoder goodput to the link bandwidth which is the minimum of the UL and DL bandwidth for the rate adaptation schemes. Tables I and II present the average encoder rate, average goodput, average PSNR and the delta loss rate (DLR) for the two scenarios. The latter is defined as the additional loss rate caused by the operation of the rate adaptation algorithm. This delta loss rate occurs whenever the uplink and downlink network buffers overflow, and it is therefore induced by congestion losses on the top of inherent losses caused by the wireless nature of the link. It has to be pointed out that in our simulations the air interface loss rate in normal conditions was 1.9% for TFRC%, 1.8% for TMMBR-U, 1.9% for TMMBR-A, 2% for TMMBR-B and 1.8% for NADU-C in the dynamic 3G link scenarios. In Figures 3 (right column) and 5, we present the percentage of bandwidth utilization in terms of probability. i.e. $\%BW\ Utilization = \frac{goodput}{actual\ linkrate}$.

TABLE I
SCENARIO 1: 3G LINKS USING RAN TRACES (180 MS SIMULATION)

	Avg. enc. rate	Avg. goodput	DLR	Avg. PSNR
	(kbps)	(kbps)	(%)	(dB)
TFRC	98.6	84.1	6.9%	29.3
TMMBR-U	99.7	89.8	3.7%	30.5
TMMBR-A	97.7	90.1	1.3%	32.3
TMMBR-B	98.5	90.5	2.9%	31.7
C-NADU	99.4	92	2.2%	31.9

TABLE II
SCENARIO 2: 3G LINKS WITH STABLE AND SLOW BW CHANGES

	Avg. enc. rate	Avg. goodput	DLR	Avg. PSNR
	(kbps)	(kbps)	(%)	(dB)
TFRC	75.7	66.1	4.4%	30.5
TMMBR-A	87.6	82.9	0%	31.8
C-NADU	88.5	80.9	2.1%	31.2

TMMBR-A due to its knowledge about the network conditions at the UL and DL provides the best adaptation $(1.3\%\ and\ 0\%$ delta loss rate and $60\%\ and\ 70\%$ Average

BW utilization (ABU)[2]) while TFRC, basing its knowledge solely on normal RRs, suffers from the maximum packet loss ($6.9\% \ and \ 4.4\%$) and under utilizes the link ($33\% \ and \ 40\%$ ABU) in both the scenarios. In the dynamic 3G scenario, TMMBR-B receives the upper-bound bandwidth information of the downlink, and is therefore able to provide better utilization (50%) of the link when compared to TFRC. However, due to probing (based on RTT, inter-arrival times of packets at the receiver) it causes a delta loss rate of 2.9%. C-NADU on the other hand, without any assistance from the network produces better results in terms of delta loss rate ($2.2\% \ and \ 2.1\%$) and ABU ($55\% \ and \ 60\%$) when compared to TFRC and unassisted TMMBR (TMMBR-U) which produces 3.7% delta loss rate and only 40% ABU.

VI. Conclusion

Network-assisted rate adaptation provides the best adaptation, which can be useful in scenarios such as handovers, cell-loading where the operator has knowledge of an event before it takes place. In this case TMMBR-A (TMMBR with network assisted adaptation) has shown the best performance. When no direct information about the uplink and downlink bit rates is available from the network, our new algorithm (C-NADU) has shown performance close to that of TMMBR-A and better than the unassisted TMMBR (TMMBR-U). Moreover, by using cross layer technologies it could be possible to get some of this information from within the device instead of signaling it explicitly. Results also show that TFRC adapted for real time media is still not well suited for multimedia applications as it under utilizes the link. We believe that C-NADU can be extended to operate in the general internet because it does not get link updates like TMMBR and makes decisions based on perceived network conditions.

Extension to the current work will involve adapting the algorithms to consider video slices, proactive RTCP scheduling to send feedback early, considering scenarios with short intermediate queue. Furthermore, develop these rate adaptation mechanisms for the general internet environment with cross-traffic.

References

[1] 3GPP TS 26.114, “IP Multimedia Subsystem (IMS): Multimedia telephony; media handling and interaction.” [Online]. Available: http://www.3gpp.org/ftp/specs/html-info/26114.htm

[2] ITU-T Rec. H.264, “Advanced video coding for generic audiovisual services.”

[3] 3GPP S4-080771, “MTSI video dynamic rate adaptation: Evaluation framework ver 1.0.” 3rd Generation Partnership Proejct (3GPP), Proposal S4-080771, Oct. 2008. [Online]. Available: http://www.3gpp.org/FTP/tsg_sa/WG4_CODEC/TSGS4_51/Docs/S4-080771.zip

[4] S. Floyd, M. Handley, J. Padhye, and J. Widmer, “Equation-based congestion control for unicast applications,” in *SIGCOMM ’00: Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*. New York, NY, USA: ACM, 2000, pp. 43–56.

[5] L. Gharai, “RTP with TCP Friendly Rate Control,” work in progress, January 2008. [Online]. Available: http://tools.ietf.org/id/draft-ietf-avt-tfrc-profile-10.txt

[6] S. Floyd, M. Handley, J. Padhye, and J. Widmer, “TCP Friendly Rate Control (TFRC): Protocol Specification,” Internet Engineering Task Force, RFC 5348, Sep. 2008. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5348.txt

[7] S. Wenger, U. Chandra, M. Westerlund, and B. Burman, “Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF),” RFC 5104 (Proposed Standard), Feb. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5104.txt

[8] M. Handley, S. Floyd, J. Padhye, and J. Widmer, “TCP Friendly Rate Control (TFRC): Protocol Specification,” Internet Engineering Task Force, RFC 3448, Jan. 2003. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3448.txt

[9] J. Ott, S. Wenger, N. Sato, C. Burmeister, and J. Rey, “Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF),” RFC 4585 (Proposed Standard), Jul. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4585.txt

[10] I. Curcio and D. Leon, “Application rate adaptation for mobile streaming,” *WOWMOM ’05: Proceedings of the Sixth IEEE International Symposium on World of Wireless Mobile and Multimedia Networks*, pp. 66–71, 13-16 June 2005.

[11] ——, “Evolution of 3gpp streaming for improving qos over mobile networks,” *ICIP 2005: IEEE International Conference on Image Processing, 2005.*, vol. 3, pp. III–692–5, 11-14 Sept. 2005.

[12] 3GPP TS 26.234, “Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs.” [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/26234.htm

[13] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” RFC 3550 (Standard), Jul. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt

[14] J. Ott, I. Curcio, and V. Singh, “Real-time Transport Control Protocol Extension Report for Run Length Encoding of Discarded Packets,” work in progress, June 2009. [Online]. Available: http://tools.ietf.org/id/draft-ott-avt-rtcp-xt-discard-metrics-00.txt

[15] H. Garudadri, H. Chung, N. Srinivasamurthy, and P. Sagetong, “Rate adaptation for video telephony in 3g networks,” *Packet Video 2007*, pp. 342–348, Nov. 2007.

[16] I. Johansson and M. Westerlund, “Support for Reduced-Size RTCP, Opportunities and Consequences,” work in progress, May 2009. [Online]. Available: http://tools.ietf.org/id/draft-ietf-avt-rtcp-non-compound-08.txt

[17] “Homepage of the Network Simulator (ns2) and the Network Animator (nam),” http://nsnam.isi.edu/nsnam.

[18] Nokia, “Nokia’s public h.264 codec.” [Online]. Available: http://www.nokia.com

[19] J. Devadoss, V. Singh, J. Ott, C. Liu, Y.-K. Wang, and I. Curcio, “Evaluation of error resilience mechanisms for 3g conversational video,” *IEEE International Symposium on Multimedia,*, vol. 0, pp. 378–383, 15-17 December 2008.

[20] T. Friedman, R. Caceres, and A. Clark, “RTP Control Protocol Extended Reports (RTCP XR),” RFC 3611 (Proposed Standard), Nov. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3611.txt

[21] 3GPP S4-050560, “Software Simulator for MBMS Streaming over UTRAN and GERAN,” 3rd Generation Parnership Proejct (3GPP), Proposal S4-050560, Sep. 2005. [Online]. Available: http://www.3gpp.org/FTP/tsg_sa/WG4_CODEC/TSGS4_36/Docs/S4-050560.zip

[22] 3GPP, “Radio Link Control (RLC) protocol specification,” 3rd Generation Partnership Project (3GPP), TS 25.322, Sep. 2008. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/25322.htm

[2] this is weighted average utilization and $> 100\%$ utilization is considered as 100% utilization.