Permissible Throughput Network Feedback for Adaptive Multimedia in AODV MANETs

Manthos Kazantzidis, Mario Gerla
University of California, Los Angeles,
Computer Science Department
4732 Boelter Hall, Los Angeles, CA 90095-1596
{kazantz, gerla} @cs.ucla.edu

Sung-Ju Lee
Internet & Mobile Systems Labs
Hewlett-Packard Laboratories
Palo Alto, CA 94304-1126
sjlee@hpl.hp.com

Abstract-- **Higher layer protocols in wireless networks need to dynamically adapt to observed network response. A common approach is for each protocol to employ an end-to-end monitoring and measuring mechanism and estimate quantities of interest, usually related to delay, latency or bandwidth. A less conventional approach is to employ network layer feedback mechanisms in place or in aid of end-to-end efforts. In this paper we use an 802.11-based *link throughput* and *permissible throughput* measurement. In our experiments these source-destination pair permissible throughput measurement are propagated in a multi-hop network using the AODV algorithm. In this setting, we are investigating how source rate adaptive multimedia applications can make use of this network feedback. We find that the investigated network feedback approach provides a strong network control feature, an accurate and timely available measurement at the sources, while the overhead of supporting and maintaining it is minimal. Finally, we argue that network based feedback approaches are particularly promising for higher layer protocol support and soft-QoS support in wireless networks.**

Index terms-- **wireless, AODV, permissible throughput, multi-hop, ad-hoc, network feedback, end-to-end feedback, call acceptance, bandwidth measurement, throughput measurement.**

I. Introduction

The 802.11 standard for medium access control may allow link performance measurements by using the acknowledgement to existing unicast traffic. Provided that meaningful and robust throughput and delay measurements can be obtained, the effort to support and propagate them in the network may be well worthwhile, since such techniques are likely to enhance the efficiency and design simplicity of higher protocols. Such measurements may play significant role in QoS support mechanisms such as call acceptance control, QoS routing, reservation mechanisms, adaptive multimedia and other transports. These problems are particularly hard in multihop ad-hoc networks (MANETs).

Link bandwidth measurement has been attempted in the past for networks with wireless links (including multi-hop) using a variety of approaches including link by link or end-to-end, on demand or proactive, and using existing traffic or extra traffic. In [5] a maximum available bandwidth measurement per node is proposed for QoS support in MANETs. The proposed measurement is however expressed in a formula. Translation of the formula to a distributed algorithm would likely result in a complex scheme mainly because it requires the knowledge of neighborhood for each node, as well as a protocol for exchange of local measurements (in order for a node to compute its own available bandwidth. In [1] a technique called packet tailgating is presented for the internet, which is based and claimed as accurate as packet pair but consuming much less bandwidth. In the same paper it is mentioned that no end-to-end measurement can be accurate enough after a few hops. The same authors have used a kernel density estimator in [2]. In this work they identify the problems of existing measurement techniques as poor accuracy, poor scalability, lack of statistical robustness, poor agility to adapting to bandwidth changes, lack of flexibility in deployment and inaccuracy in presence of variety of traffic types with existing techniques. In networks with wireless links these problems become even more critical due to the high response variance of the medium as well as the nature of contention in multi-hop links. Also, in general, mobility resets statistical properties of the measured values and limits the applicability of statistical methods.

We combine the link by link (source, destination pair) throughput and link utilization measurements to calculate the permissible throughput. We apply three techniques to the instant link by link throughput in order to make it robust, relatively comparable and absolute. Propagation of the permissible throughput values can be accomplished using a number of techniques. It can be performed either by piggybacking the values to routing messages or as a separate application. The former constrains the two mechanisms, with possibly different cost functions, to work on relevant time scales. Their propagation can also be distance vector based, link state based or on demand. In this paper we are investigating an on-demand, event triggered propagation algorithm based on AODV.

We believe that in most wireless link layers such measurements are feasible since when they are asynchronous they have to employ inband control mechanisms to acquire the channel, they can sense the channel and may have an explicit acknowledgement as in 802.11.

In the next section we present our approach for 802.11 links. In section 3, we perform simulations with network feedback and analyze the results. In 4 we conclude and next we mention future work that is opened by this study.

II. Measurement And Propagation Of Permissible Throughput

A. Measurement

Measuring the link permissible throughput in multi-hop networks requires taking into account the unique contention for each (source, destination) link pair, the distributed knowledge of the contention and the rapid changes due to mobility. Consequently a permissible throughput measurement can be written:

$$r_{(source,destination)} = (1-u)*Throughput_{(source,destination)}$$

where u is the link utilization. The source-destination pair throughput is measured for a window of packets using existing traffic. Each successful packet transmission contributes its bits to the numerator of the throughput measurement and the time from when it was ready for transmission at the head of the link queues, to the acknowledgement receipt. This interval is packet size dependent as shown in the next equation:

$$Throughput_{packet} = \frac{S}{t_q + (t_s + t_{CA} + t_{overhead}) * R + \sum_{r=1}^{R} B_T}$$

$$where, t_q = MAC\ queuing\ time,$$

$$t_s = transmission\ time\ of\ S\ bits$$

$$t_{CA} = Collision\ Avoidance\ phase\ time$$

$$t_{overhead} = Control\ Overhead\ time$$

$$(e.g. RTS, CTS, ACK, Hdr, 4\ prop\ delay\ etc)$$

$$R = Number\ of\ necessary\ transmissions$$

$$B_T = Backoff\ time\ for\ r$$

In [8], we chose a simple approach to subtract an overhead constant from the denominator of each measurement because it does not require any extra memory to keep the different measurement values per different packet size and source-destination pair. As shown in [8], we still achieve an accurate measurement interpretable enough to be used in multimedia adaptation or even call admission control. In cases where congestion has not built up R is close to 1. In those cases the denominator of the link throughput measurement is indeed a constant that depends on specific MAC implementation parameters. In fact, if c is close to $t_{overhead}$ at low loads ($t_{overhead}$ is paid in most packet transmissions, depending on RTS/CTS related parameters) then at these loads our throughput measurements will be almost immune to packet size. At high loads the throughput will be sufficiently low, and consequently more immune to packet size, due to the longer collision avoidance and backoff phases. An example calculation of c is shown below:

$$c \approx t_{overhead} \approx t_{RTS} + t_{CTS} + t_{HDR} + \tilde{t}_{waitACK} + t_{transmission\ overhead}$$

A typical calculation of this may include the transmission times of the overhead bits, RTS, CTS, Header, ACK, the hold for ACK time (see also[6]), the extra propagation times and SIFS for the RTS and the CTS and one DIFS that the transmission will be delayed etc. (see also [7]). For our simulations we use $c = 1200\ \mu \sec$.

Another measurement issue is smoothing and filtering using a packet window. We use a packet window in order to increase statistical robustness of the measurements. A packet window of 16 or 32 samples (packets) is adequate to produce fast enough, noise immune measurements. For an illustration of this see fig. 1 where the high variance per packets measurements are aggregated on a window of 32 packets. The idle time and window duration are calculated to produce the link utilization factor and the permissible throughput measurement as

$$\frac{idle\ time\ in\ window}{window\ duration} * Throughput\ measured.$$

The measurement summarized so far does not necessarily has to be explicitly supported by a link layer implementation. In [8] an algorithm to obtain this measurement at the network layer is provided. The disadvantage of a network implementation is that errors are introduced to the measurement but the obvious advantage is that it can be implemented software, as opposed to firmware, making its deployment easier.

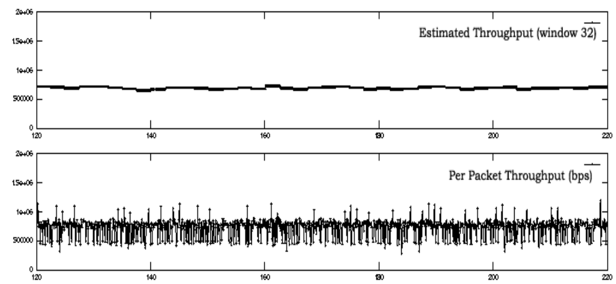

Fig. 1: Window operation

B. Propagation of Permissible Throughput

We have designed and implemented a propagation of the measurement based on AODV [4] routing algorithm for MANETs.

In AODV only routes to destinations each node communicates with are recorded. The uniqueness of AODV is its two dimensional routing metric. The hop count metric accounts for building optimal routes, and the sequence number ensures the freshness of the route. When a source has data packets to transmit to the destination but does not have any routing information, it floods the entire network with the Route Request (RREQ) packet to discover a route. A node can respond to this RREQ by sending a Route Reply (RREP) to the source if one of the following two conditions is satisfied. The node must either be the destination itself, or it must have the ‘‘fresh enough” routing information to the destination. When the source receives a RREP, it learns the route to the destination and can hence send data packets. When a node detects a route disconnection, the upstream node of the broken link sends a Route Error (RERR) message to the upstream direction and to the source to notify a route break. Nodes that receive this packet remove the routing table entries that use the broken link. The source can re-discover a route when needed.

We made implementation changes to AODV in order to support propagation of the measurement values. We have kept the AODV philosophy in all the decisions taken for the propagation of permissible throughput.

Firstly, two fields are added to the RREP packet; minimum measurement value field and bottleneck link field. The former is used to keep the minimum permissible throughput along the discovered path (bottleneck) while the latter is used to identify the distance of the current node to the bottleneck link, the link along the path for which the permissible throughput measurement is the smallest. This is necessary to enable detection of change of the route's bottleneck link without route change.

Additionally, hop based AODV naturally reacts only to path changes (link failure messages), whereas our setting requires a more frequent update to detect changes in permissible throughput measurements along an unchanged path. In order to accommodate this while keeping the on-demand philosophy we added the "event triggered RREP." When the measurement values is changed for more than RELAY_FACTOR % than the previously relayed measurement for the link (source-destination pair) then an RREP is triggered. A RELAY_FACTOR of 1.0 (never trigger RREP from change in measurement) naturally minimizes the control overhead but allows learning of new measurements only following route changes. A RELAY_FACTOR of e.g. 0.2 (trigger an RREP when the measurement has changed by at least 20%) increases control overhead but results in learning one permissible throughput change along a path within at most a round-trip time. During an RREP the node sends this information to the upstream node of the route. Upon receiving this event triggered RREP, a node will not relay the measurement packet unless the event triggering condition holds between his own last transmitted value for the link and the newly received.

The routing still operates without being affected by the propagation of these values, other than the increase in the size of messages, and calculates the minimum hop routes. In general it may be desirable that the mechanism be implemented in parallel with the routing, operating in de-coupled processes and time scales.

We are now ready to put this scheme to the test and find out how and how much multimedia applications can benefit from it.

III. SIMULATIONS

A. Experiments

The audio is encoded at 9 rates of 256 to 8 Kbps as in [3]. The server tries to match half of the reported throughput by choosing the appropriate rate for the codec, in order to maintain stability and fairness in the network.

1) Correctness: The following figures show that the measurement is correct in CBR and VBR traffic.

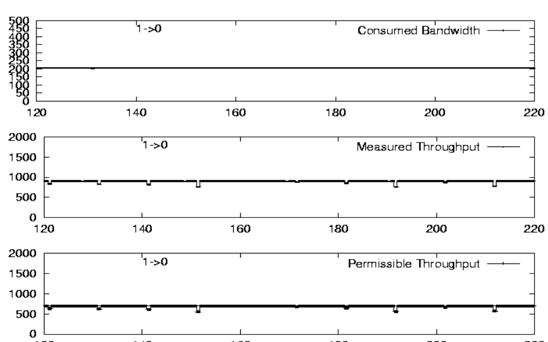

Fig. 2 Exact measurements for CBR

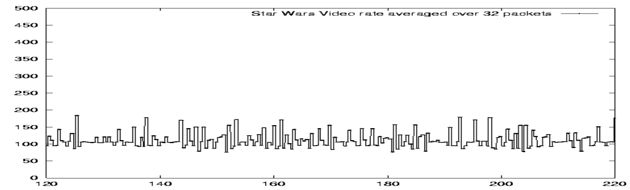

Fig. 3 The VBR source (H263 clip) in time averaged over 32 packets

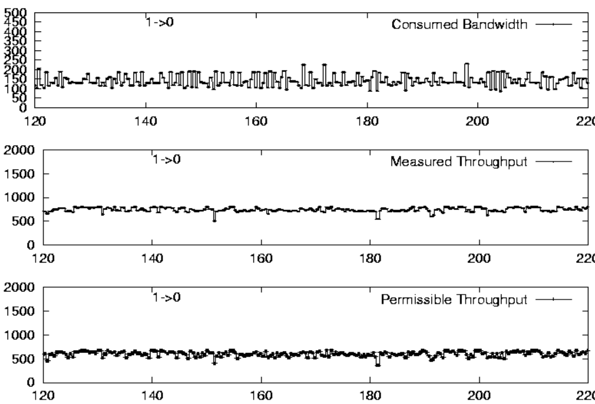

Fig. 4 Accurate measurements for VBR traffic.

2) Individual Connection: In this section we take some connections from the experiments presented aggregately in the following paragraph and look at them closer. In all figures the top graph shows the RTP 1 second loss rates, the middle graph shows the permissible throughput to the destination and the bottom graph shows the rate adaptation (not necessarily quality adaptation) as is received by client (and chosen by the server). These graphs show the behavior of the application in relation to the measurement. They also show that even in most cases of high loss rates the measurement is correct and indicates a close to zero permissible throughput.

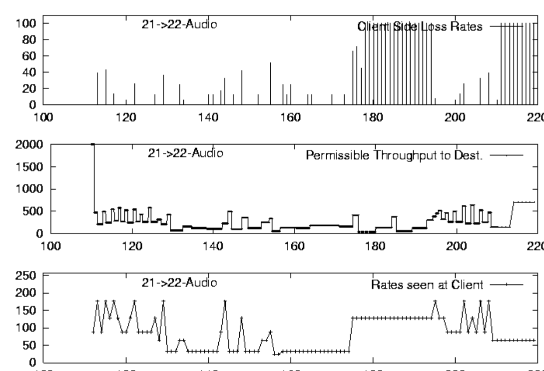

Fig. 5: 1 hop connection (in experiment with 10 connection pairs), no mobility

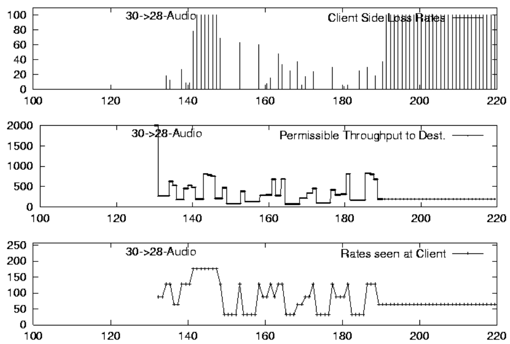

Fig. 6: 2 hop connection (10 connection pairs), no mobility

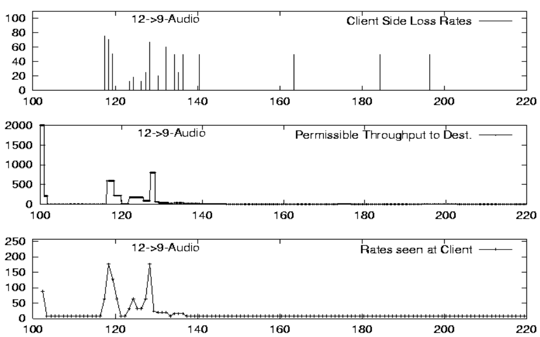

Fig. 7: An initially 3 hop connection in a 25 m/sec experiment.

3) Performance: In this section we are experimenting a mesh topology of 36 (6 by 6) nodes. In the mesh topology each node is within range with the 4 neighbors across but cannot reach the diagonal nodes. We use a 170x170 terrain with a 40-unit range. The links are 1Mpbs links. We distribute 2 to 10 connection pairs of 1 through 3 hop connections throughout the network. We gradually introduce mobility up to very high speeds of 65 m/sec. In high mobilities the initial hop count naturally becomes less and less relevant. Figures 8-15 and show that the mechanism is able to correctly control the application flows. The aggregate server consumed bandwidth is not increasing at high loss rates when adding new connections. Note that in absence of call admission the applications manage to share the network resources. For example, in the case of 3-hop connections adding more than 5 pairs actually decreases the overall bandwidth attempted to be transmitted by the servers. Note also that the loss rates in the figure are aggregated over all connections. In fact, it is always the case that high aggregate loss rates are due to individual suffering connections. For example the 35% loss rate of 10 pairs is mostly due to 2 connections unable to get their lower layer through (because there is no call admission in these experiments). The loss rates for each connection were 0%, 1.4%, 0.15%, 1.4%, 42%, 92%, 0.9%, 24%, 20%, 97%.

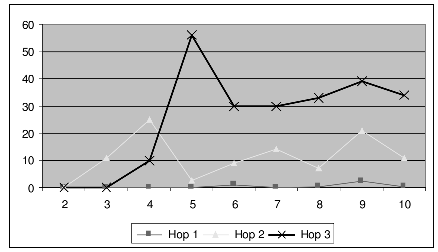

Fig. 8: Loss rates aggregated over all connections in each of 30 experiments.

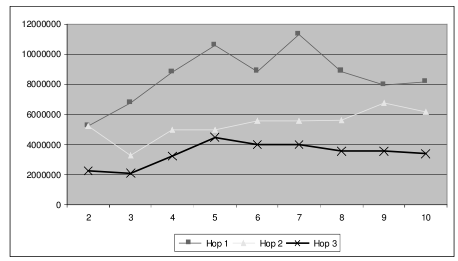

Fig. 9: Server Consumed Bandwidth for the network feedback case.

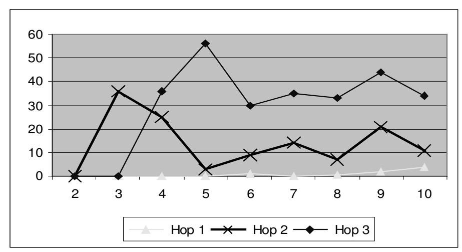

Fig. 10: Aggregated Loss Rates for Mobility 3m/sec

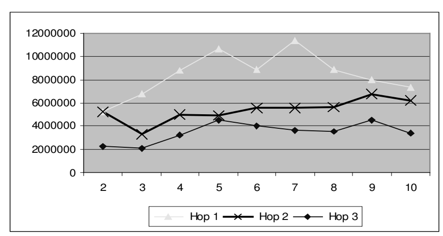

Fig. 11: Server Consumed Bandwidth for mobility 3 m/sec.

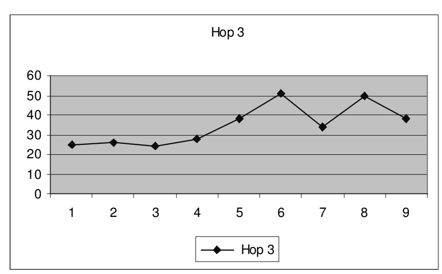

Fig. 12: Loss Rates for mobility 15 m/sec. Connections are initially 3-hop.

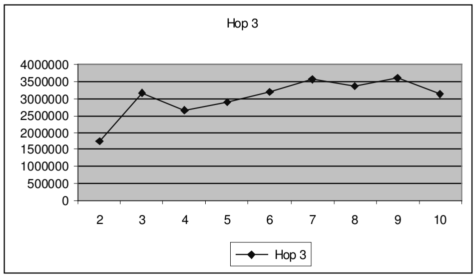

Fig. 13: Server Consumed Bandwidth for Mobility 15 m/sec.

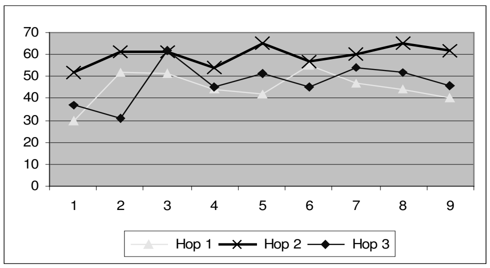

Fig. 14: Aggregate Loss Rates for mobility 40 m/sec. Hop distance is only initial.

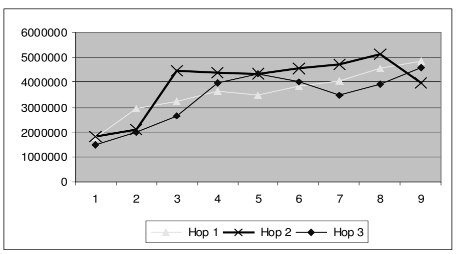

Fig. 15: Server Consumed Bandwidth for mobility 40 m/sec.

Note in the figures that the scheme works well for even high mobility. The hop distance changes frequently and consequently there is no separation of the 1,2,3 hop cases. The server consumed bandwidth slightly increases as connection pairs are added. This is because of the delay to obtain a valid measurement and the delay for the valid measurement to be propagated towards the source. When a link breaks, the nodes reset their local measurements and have to wait for a window of packets to obtain their first estimate.

IV. CONCLUSIONS

We have used the 802.11 acknowledgement and link failure events to measure throughput and permissible throughput on a link by link basis in 802.11 networks. We have enhanced the simple bytes/time unit approach with techniques that increase the measurement statistical robustness, correctness and interpretability. We provided a mechanism for measuring the link utilization within a window of packets, by calculating the idle time and the window duration as time progresses, and showed that these techniques combined provide a robust and correct permissible throughput measurement. At the expense of errors treated as statistical noise we have used a network layer mechanism and implementation which measures these quantities independently of link layer specific implementation, and allows for independent development and ease of deployment. We developed a propagation mechanism for these values that timely discovers (and cancels) the bottleneck permissible throughput to any destination based on the AODV routing protocol. After making those values available at the end points of a multi-hop ad-hoc network, we showed that the measurements propagated with AODV provide an attractive alternative to end-to-end techniques. We showed by simulation that the measurements are accurate and timely available and that can be used very effectively in rate-adaptive multimedia applications. We showed that even in cases of high loss rates, the measurement is accurate and available. This indicates that it may be suitable for call admission techniques. Our simulations show that the values propagated provide a powerful network control mechanism at the endpoints. Finally we note that more than one transport and application can share the benefit of network measurements and feedback in parallel to their own specific measurements. In general network supported measurements and feedback present a promising approach for soft QoS in ad-hoc multi-hop wireless networks.

V. FUTURE WORK

The scheme showed promise in using it to perform call acceptance strategies. Also, propagation of the measurements mostly fits the link state paradigm. Furthermore, even though propagation of the measurement need not be piggybacked in routing packets, we have only considered this so far.

REFERENCES

[1] Kevin Lai; Mary Baker Measuring Link Bandwidths Using a Deterministic Model of Packet Delay. *Proceedings of SIGCOMM 2000*

[2] Kevin Lai and Mary Baker, Measuring Bandwidth, *Proceedings of IEEE INFOCOM '99*, March 1999.

[3] Kazantzidis, M.I.; Wang, L.; Gerla, M. On fairness and efficiency of adaptive audio application layers for multihop wireless networks. *1999 IEEE International Workshop on Mobile Multimedia Communications (MoMuC'99) IEEE, 1999*. p.357-62. xii+390 pp.

[4] Perkins, C.E.; Royer, E.M. *Ad-hoc on-demand distance vector routing*. Proceedings WMCSA'99.

[5] Cansever, D.H.; Michelson, A.M.; Levesque, A.H. Quality of service support in mobile ad-hoc IP networks. *MILCOM 1999. IEEE* p.30-4 vol.1.2vol. xxxv+1499 pp.

[6] IEEE Std 802.11 - Wireless LAN Medium Access Control and Physical Layer specifications

[7] Hadzi-Velkov, Z.; Gavrilovska, L. Influence of hidden terminals over the performance of the MAC protocol for IEEE 802.11 wireless LANs. ITG-Fachbericht, (no.157), (*European Wireless '99)*

[8] Manthos Kazantzidis "Wireless Multimedia using Network Measurements" - UCLA CS Technical Report No. 200102