Benefits of Packet Aggregation in Ad-Hoc Wireless Network

Ashish Jain, Marco Gruteser, Mike Neufeld, Dirk Grunwald
Department of Computer Science
University of Colorado at Boulder
Boulder, CO 80309

CU-CS-960-03 August 2003

University of Colorado at Boulder

Technical Report CU-CS-960-03
Department of Computer Science
Campus Box 430
University of Colorado
Boulder, Colorado 80309

Benefits of Packet Aggregation in Ad-Hoc Wireless Network

Ashish Jain, Marco Gruteser, Mike Neufeld, Dirk Grunwald
Department of Computer Science
University of Colorado at Boulder
Boulder, CO 80309

August 2003

Abstract

Emerging mesh networks build on IEEE 802.11 wireless local area networking technology to deliver Internet access to mobile users or to areas where traditional wired access technologies are unprofitable. Such deployments are highly bandwidth constrained; in dense urban environments many users are competing for the wireless medium and long-haul wireless backbone connections have significant contention. In addition, IEEE 802.11 networks suffer from a high packet overhead—much higher than Ethernet, for example. This overhead consumes a significant share of the theoretical capacity. As networks advance to higher bitrates the proportion of this overhead increases, because packet preambles are transmitted at the lowest bitrate. We address this challenge through an adaptive, connection-agnostic aggregation mechanism that can reduce this overhead by combining multiple smaller packets into larger ones. One method uses a small delay on packets to enable the distributed aggregators to collect smaller packets from different network connections for assembling a larger packet. The second method automatically increases aggregation activity with network congestion; thus, it does not impose delays on network traffic in situations of low media utilization and also *reduces* delays under periods of high load. A prototype on a wireless testbed and simulation results for larger community network deployments show significant improvements in actual network capacity, ranging from a 60-270% improvement for traditional "ad-hoc" scenarios and to a 100-500% improvement for wireless networks using unidirectional antennas.

1 Introduction

Wireless networking provides an opportunity to facilitate new uses for networks while also increasing the number of users who can access broadband services. There has been explosive growth in the use of inexpensive, license-free networking technology for applications that previously would have been almost impossible to implement. An example of this is the High Performance Wireless Research and Education Network (HPWREN [1]), a high speed network that both connects rural Native lands and serves as a high speed sensor network for seismic measurements and other instrumentation. Concurrently, numerous "cooperatives" [2, 3, 4] are forming that provide wide-area wireless networking, either for areas that previously had limited connectivity or simply to improve data access in urban environments.

Many of these networks are examples of what we term "*community networks*" – networks that are jointly organized and administered by a community of users rather than the traditional hierarchical organizational model used in the Internet. These networks include a combination of fixed wireless services and mobile networking. There are a number of challenges presented by community networks absent in hierarchically-organized networks. Our research group is addressing several of those challenges.

One such challenge is *network scalability*. The community networks we are deploying use 802.11b networking hardware in the unregulated ISM band. In urban and suburban environments, such as that shown in Figure 1, we use a combination of omnidirectional and unidirectional antennas; in rural areas, we mainly use unidirectional antennas. In each case, we have built a "mesh network" using *ad hoc* routing. Our prototype systems use a "host access point" implementation to provide conventional access point functionality, but routes packets between the nodes using a wireless distribution service. This means that legacy applications, such as wireless devices, Windows or Macintosh system, can use the custom access points, but that traffic between the access points is routed using our router implementations. We currently have a 10-node experimental network and are working with a rural wireless internet service provider to extend their current point-to-point wireless network to include mesh networking.

Several studies have shown that *ad hoc* networks suffer from limited capacity(*e.g.* [5]). Most of these studies do not actually address limits in *ad hoc* networks; rather, the limits they encounter are caused by assumptions of isotropic propagation models for omnidirectional antennas. There are many existing wireless networks that use directional antennas and contain several hundred customers, and these networks can use *ad hoc* routing for increase robustness.

Figure 1: Suburban neighborhood used to generate a NS2 scenario file

One capacity problem that *does* occur in wireless networks is the poor efficiency of higher bandwidth configurations in 802.11b networks due to overhead. In this paper, we describe how *packet aggregation* can be used to increase the delivered bandwidth in community networks including fixed and mobile stations. Packet aggregation is precisely that – we aggregate or combine packets at intermediate nodes and forward them using an encapsulation layer. Our aggregation implementation works at either the MAC or network layer and combines packets from several flows. We show that for scenarios similar to a conventional "adhoc" community wireless network, packet aggregation can improve throughput by 60-270%, although overall throughput is disappointingly low. However with the use of unidirectional antennas and careful utilization of different channels, substantial gains in throughput was noticed. For environments with less interference for the wireless media, performance improvements of 100-500% are possible.

A common objection to packet aggregation is that larger packets are more prone to error, and the increased packet size will result in a lower overall performance. We used common interference devices (*e.g.* a 2.4GHz wireless phone) and determined that while such interference reduces throughput by 15-40% with increasing packet size, aggregation still provided higher throughput. Another concern when using packet aggregation is that it will increase packet delay, reducing the suitability of the network for media services. We implemented two packet aggregation techniques. One uses an explicit packet delay at intermediate nodes to increase aggregation. The other technique seeks simply to increase media capacity and uses the natural queueing that occurs under periods of high demand to provide packet delay. We show this second technique is superior. Further, both forms of aggregation show significant reduction in packet latency compared to non-aggregated case in congested scenarios.

The rest of this paper is structured as follows: next, we describe the overhead in 802.11b networks at high modulation rates. We then describe the two aggregation mechanisms in §3. We then discuss the experimental methodology and tools used to evaluate the aggregation techniques, including simple analysis of induced noise. We then survey related work and discuss our results.

2 Packet Overhead in IEEE 802.11 Networks

The requirements for wireless networks differ from their wired counterparts in several important points that lead to a higher overhead per packet. First, transmissions over a wireless medium are inherently less reliable and suffer more from interference. Therefore, the 802.11 standard specifies a checksum for the physical layer encapsulation (preamble and PLCP header) in addition to the checksum for the MAC header and the frame

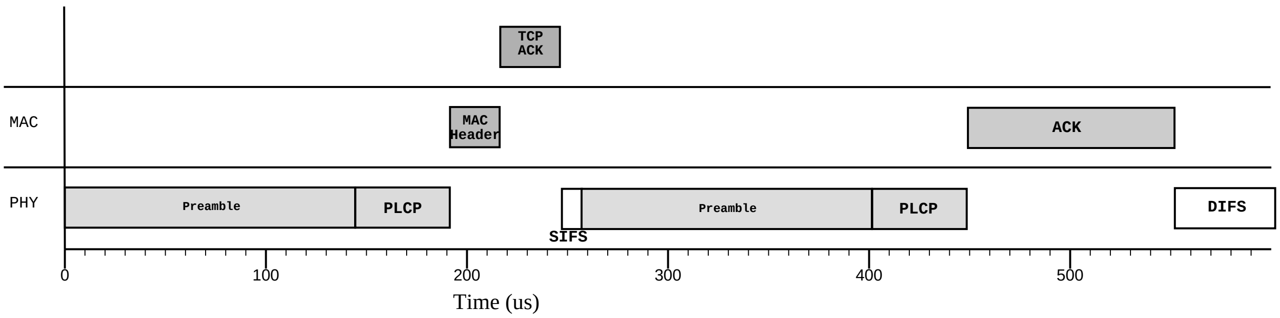


Figure 2: Physical and Mac Layer overhead for IEEE 802.11b packet transmission. The back-off period is not shown to simplify the diagram; this time is variable based on the contention experienced by the transmitter. The minimum back-off period is selected from a uniform ranging between 0 and 31 timeslots, each of which are $20\mu s$ – *i.e.* the average back-off period is $300\mu s$. The upper range of the back-off period increases exponentially as collisions are detected.

body (payload). More important, the receiver of a data frame answers with an acknowledgment to indicate a successful transmission.

Second, a wireless station cannot transmit and listen to the medium simultaneously. This means, that the sender of a message cannot determine a collision on the network until the transmission of the frame is finished. This is further exacerbated by the "hidden node" problem. The 802.11 specification can use a "request-to-send / clear-to-send" protocol to arbitrate access; in practice, this is not usually used.

The 802.11b standard offers several different transmission speeds, including 1, 2, 5.5 and 11 mbit/s. These speeds use different modulation techniques (BPSK, QPSK, CCK, CCK). The higher transmission rates use encodings that are more spectrally efficient but more susceptible to error. Each 802.11b access point or station can use any of the modulation techniques and automatically changes to different modulations depending on the bit error rate and signal strength. For example, the Linksys WCF11 802.11b card advertises an out-door range of 152 meters at 11mb/s, 270m at 5.5mb/s, 396m at 2mb/s and 457m at 1mb/s using the provided antenna. Actual range is determined by a number of factors, including the antenna quality, ambient noise and multipath interference.

Different stations served by the same access point may use different modulation techniques. To support the plethora of modulation techniques, each message uses a common preamble that encodes the configuration options. The common modulation technique uses the lowest transmission speed (1mb/s) for the common header. Message transmissions can essentially be broken into two parts $O + D$, representing the overhead and data. Since the 11mb/s modulation technique is, essentially, 11 times more efficient than the 1mb/s, the transmission duration for the contents of a frame at 11mb/s is 1/11th the duration of a message transmitted at 1mb/s.

Figure 2 depicts a typical exchange for the transmission of a TCP acknowledgment (ACK) to a mobile stations. Assuming the standard long preamble and an 11Mbps configuration, the size of the blocks indicates the time consumed for each activity. The darker blocks are transmitted by the sender of the TCP packet, the light gray blocks belong to the acknowledgement from the receiver, and the white blocks are inter frame spaces. The two darkest blocks indicate 11Mbps transmission while all other blocks use a 1 Mbps bitrate. Evidently, the required MAC and physical layer transmissions occupy the medium for an order of magnitude longer than the actual TCP packet. A large part of this overhead is caused by the requirement to transmit the preamble, PLCP header and control frames in the MAC layer at the Basic Service Set Basic Rate, typically 1 Mbps.

Figure 3 shows the measured minimum transmission time for packets in an 802.11b network running at 11mb/s, measured in milliseconds per packet. For each sample point, we manually varied the transmission rate

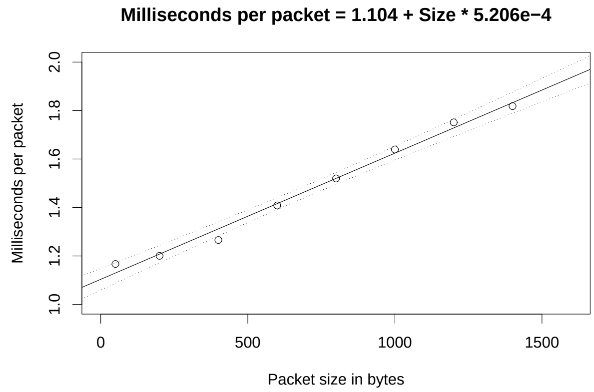


Figure 3: A linear regression model with 95% confidence intervals relating the minimum packet transmission time (in milliseconds) *vs.* packet size. The coefficient of regression is $R = 0.99$.

to find the maximum transmission rate and thus the minimum transmission time. The packet size include any protocol overhead (*i.e.* IP and UDP or TCP overhead).The basic latency is slightly larger than that indicated by Figure 2, which did not explicitly include the contention overhead.

The large overhead makes it significantly more efficient to transmit larger messages – it takes 1.36ms to transmit a 500 byte message, but only 1.88ms to transmit a 1500 byte message, or an increase of 38% to transmit three times more data.

2.1 Effect of aggregation on throughput

In order to evaluate the effect of aggregation on throughput, we consider the following simple analysis. Let us consider a fully-connected topology where all sender-receiver pairs have statistically identical and independent channels. Let t_{con} represent the average time (in seconds) per packet spent (see figure 4) in contention for a single rate IEEE 802.11 protocol for an experiment spanning T seconds. Let T_o represent the overhead time spent per packet. Thus T_o is

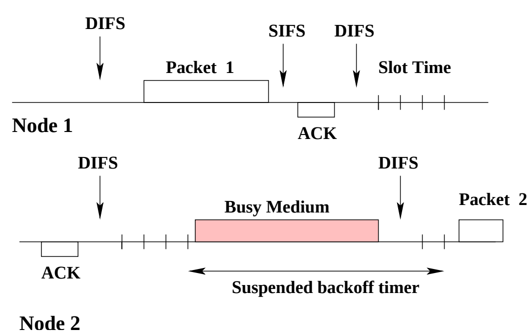


Figure 4: Effect of IEEE 802.11 backoff timer

$$T_o = \begin{cases} \text{DIFS + SIFS + ACK + 2PHY\_HDR + 2MAC\_HDR} & \textit{if basic scheme is used} \\ \text{DIFS + 3 SIFS + 4PHY\_HDR + 4MAC\_HDR + ACK + RTS + CTS} & \textit{if RTS/CTS is used} \end{cases}$$

For simplicity, let us assume that nodes are exchanging constant size packets of l bytes with a constant date rate for payload and overhead of r Mbs. Then the number of packets delivered, η, is given by $\eta \approx \frac{r.T}{l+r.t_{con}+r.T_o}$

Now we deliver the same data via aggregated packets (where each aggregated packet is encapsulating *n* packets). Thus, the size of aggregated packets would be $L = \sum_i^n l_i$. Thus if η_{base} and η_{agg} represents number of original data size packet delivered to the application without and with the aggregation respectively,

$$\eta_{agg}/\eta_{base} \propto (\sum_{i}^{n} l_i + r.\sum_{i}^{n} t_{con} + n.T_o)/(L + r.t_{con\_agg} + r.T_o) \quad (1)$$

In equation 1, t_{con_agg} represents contention time for aggregated packets. Thus for aggregation to deliver more packets, $t_{con\_agg} < \sum_i^n t_{con} + (n-1).T_o$. Happily, this is a natural consequence of aggregating packets since the aggregated packets use the media for significantly less time.

3 Implementation of Aggregation

Our motivation for packet aggregation is clear - by aggregating many smaller packets, we can increase the available bandwidth.We implemented our packet aggregation system using the Click Modular Router framework [6]. As the name suggests, the Click Modular Router provides a framework for constructing and connecting "routing elements"; a full router is configured by composing element in different configurations. The existing Click implementation supports a wide range of router elements supporting a number of wired routing configurations. We extended the existing Click tools by constructing a set of routing components that support different *ad hoc* routing protocols. In this paper, we only use our implementation of the AODV protocol [7], but packet aggregation is applicable to any underlying wireless distribution system. Our AODV implementation is designed to be used either at layer-3 (as a routing algorithm for a conventional network) or layer-2 (as a MAC-layer wireless distribution system that would replace the commonly used spanning tree distribution mechanism). In the experiments described in this paper, we only use the layer-3 implementation because it simplifies measurement. Our AODV implementation does not handle broadcast or multicast traffic, and our actual deployment uses a Rapid Spanning Tree Protocol [8] for those packets. This is also implemented as a Click module.

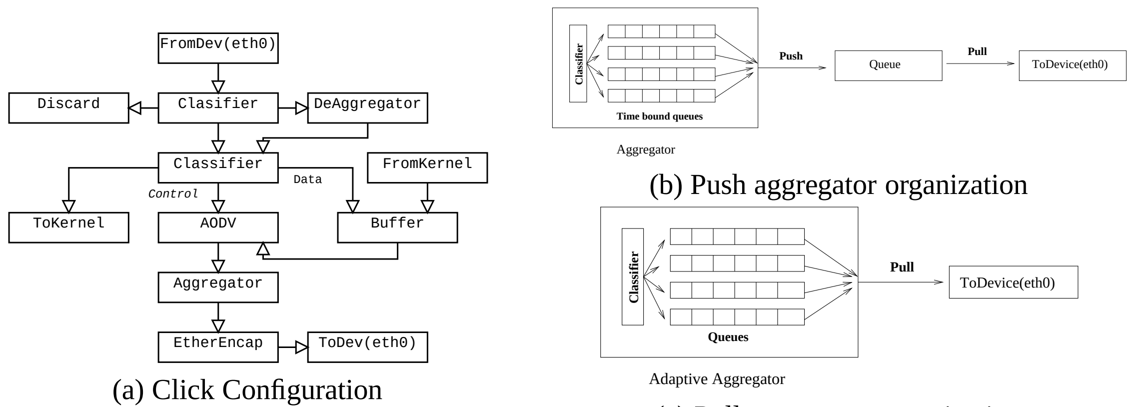
We used a combination of measurements on our prototype community wireless network and the *nsclick* [9] simulation infrastructure, which allows us to use components developed for the Click router in the Ns2 network simulator without modification. This means the same implementation is used both for our deployment and our simulation.

Recent versions of *ns-2* include a physical layer model that attempts to model the 802.11b wireless Ethernet MAC layer and power constraints. The *nsclick* simulation uses this extended 802.11 physical layer model. The experimental results use ambient noise and noise induced by explicit interference. When possible, we validate our simulation model using our experimental deployment.

3.1 Implementing Aggregation

Figure 5(a) shows the general framework for each of the aggregation schemes; this illustration is schematic since it does not show the full Click routing configuration actually used. Packets arrive and initial processing is performed by elements that are not shown. Previously aggregated packets are de-aggregated and then processed as normal; in our experimental network, aggregation uses an explicit Ethernet or frame type, and we discard all frames not explicitly used in our environment. Data packets are then forwarded through an AODV routing element that determines the next routing step for that packet. Packets intended for the local host are delivered locally (after appropriate IP-level processing). Following that, packets may be sorted into distinct classes based on packet properties. Those packets are then forwarded to one or more `Aggregator` elements.

We explored two aggregation techniques. The first is an extension of a previously patented method [10] we call *forced delay aggregation*. In this method, shown in Figure 5(b), packets are placed into individual internal queues based on the next routing step indicated by the AODV component. Each incoming packet is given an



(a) Click Configuration

(b) Push aggregator organization

(c) Pull aggregator organization

Figure 5: Schematic Illustration Show General Aggregation Mechanism

explicit timestamp; when that packet has been delayed for a period longer than a statically configured *maximum aggregation delay*, it is marked as available for transmission. When sufficient packets arrive with the same next routing step such that their combined size meets the MTU of the outgoing link, they are combined into an encapsulation packet and placed in an explicit transmission queue. The size of each internal queue can not exceed that of the MTU for the outgoing link; any available packets are aggregated and marked for transmission when an incoming packet would increase the sum of the packets sizes to be larger than the MTU.

The forced delay aggregation method is simple to implement, but causes additional packet delay due to the forced delay. But consider why packet aggregation is being performed – in an otherwise idle network, there is little incentive to aggregate packets. In a congested network, packets naturally queue waiting for transmission. A rapid stream of small packets will cause a queue in the sender because each packet has significant overhead. This observation resulted in the second method we call *congestion triggered aggregation*, illustrated in Figure 5(c). In this method, packets are sorted into queues depending on the next hop as before, but the Click module operates as a "pull" module – then the ethernet device has finished transmitting its prior packet, it requests ("pulls") a packet from the aggregator object. At this point, the aggregator determines if there are any packets that can be aggregated. This eliminates the delay for most packets, and even when packets are aggregated, the mean delay should decrease – packets other than the first in queue would have to wait more than a millisecond for the packet at the head of the queue to be transmitted.

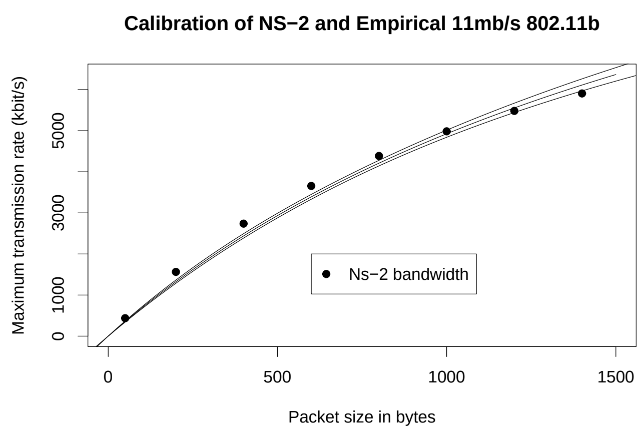


Figure 6: Calibration of maximum throughput achieved by NS-2 simulator scaled to operate at 11mb/s

4 Analysis and Simulation Methodology

We used our deployment environment mainly to calibrate our NS-2 simulations and assess the impact of noise on packet loss rates. These measurements were done using version 2.1.24 of the Click Modular Router installed on a Linux system using kernel 2.4.20. The system use a Dell Inspirion 3800 laptop with Lucent Technologies Orinoco "gold" 802.11b wireless PCMCIA cards. All calibration tests uses the internal diversity antennas and the standard wavelan kernel driver. Load generation tools were constructed as Click modules. To simplify measurement, we used the user-level Click configuration rather than the kernel-based configuration. This equipment was used for the measurements and maximum throughput regression model described in §2. This configuration of kernel and drivers were unable to use packets great than 1500 bytes, although the 802.11 standard allows 2300 byte PDSUs.

All simulations used version 2.1b9 of the Ns-2 simulator. The Click modules in the Ns-2 also used the 2.1.24 Click router software with the *nsclick* patches installed. Simulations mainly used the "wireless extensions" in the Ns-2. We also extended the Ns-2 simulator to include support for unidirectional antennas.

We scaled the parameters of the 802.11 model in the wireless extensions to simulate a 802.11b network operating at 11mb/s. Figure 6 shows the performance of the Ns-2 wireless model scaled to 11 mb/s (the solid points) overlayed on the maximum bandwidth measured on our deployment platform; the dotted lines indicate the 95% confidence intervals for the predicted bandwidth based on the regressed data. While not perfect, the agreement is sufficiently accurate that we did not feel the need to implement a more detailed 802.11b model. Prior work [5] indicated similar agreement when the Ns-2 model was scaled to 2mb/s.

4.1 Mobility/Traffic Scenarios Used In Evaluation

We used a number of scenarios to assess the value of packet aggregation particularly in networks similar to community networks. Figure 1 shows one community network environment we have considered; the map is a 1m resolution map extracted using the Microsoft TerraServer web services [11] for portions of a local suburban area. We identified 381 houses and these are indicated by marks; these were visually determined from the map data and confirmed by visiting the indicated neighborhood. We also model traffic flow using information from the USGS *spatial data transfer standard (SDTS)* [12]. We extracted vector coordinates of primary, secondary, minor roads from the transportation layer of the $1 : 24,000$ scale *digital line graph (DLG)* [13] data files. To emulate moving traffic, we used mobile nodes moving at 10 m/s which, on finding an intersection, chooses a random direction and turns.

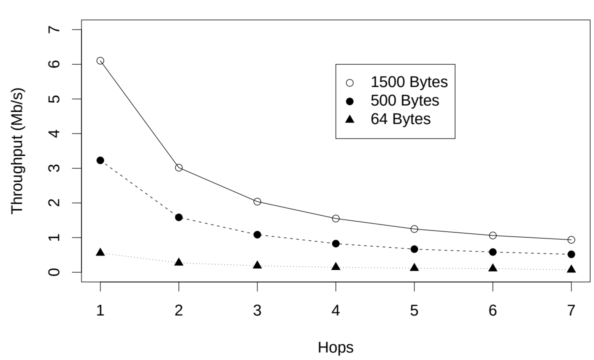


Figure 7: Impact of isotropic propagation model on throughput

The "suburban scenario" includes 381 houses and 30 vehicles. Overlaid on this were two communication scenarios. The first was a simple client-server model where five centrally placed servers each communicate with five clients; the remaining 351 stationary nodes and 30 vehicles act as routers, but do not explicitly communicate. This would be representative of a set of isolated interactive gaming or distributed music servers. Our second communication model combined the client/server with a gateway, where the servers in the prior client-server model communicate with the gateway that represents an uplink to the Internet.

As we'll see, the throughput improvement for this scenario is significant, but the overall bandwidth is fairly low. This occurs because NS-2 uses an isotropic propagation model on a single channel. As previously reported [5], the MAC layer media arbitration limits the available bandwidth as the size of the network increases.

Figure 7 shows the performance of 802.11b at 11mb/s in a linear arrangement. Although 1500-byte packets achieve a significant throughput improvement over 64 byte packets, the delivered bandwidth is low ($\approx$ 1.1mbs).

Current wireless internet service providers typically do not use omnidirectional antennas (*i.e.* those with an isotropic propagation pattern) for customer premise equipment (CPE). A common model used by local wireless internet service providers is to use a backbone network with unidirectional links interconnecting stations with omnidirectional or sectored antennas. We model such an environment in our "rural" scenario, shown in Figure 8. Internally, this is structured as two distribution points each with four stations surrounding each distribution point. The distribution points operate on different channels and the backbone connection connects to a "gateway" that represents a connection to the internet. In practice, the stations surround the distribution points uses unidirectional antenna to talk to the distribution services. To achieve this, we extended Ns2 to support unidirectional antennas. These antenna emulate sectored antennas.

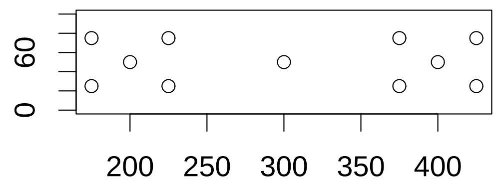


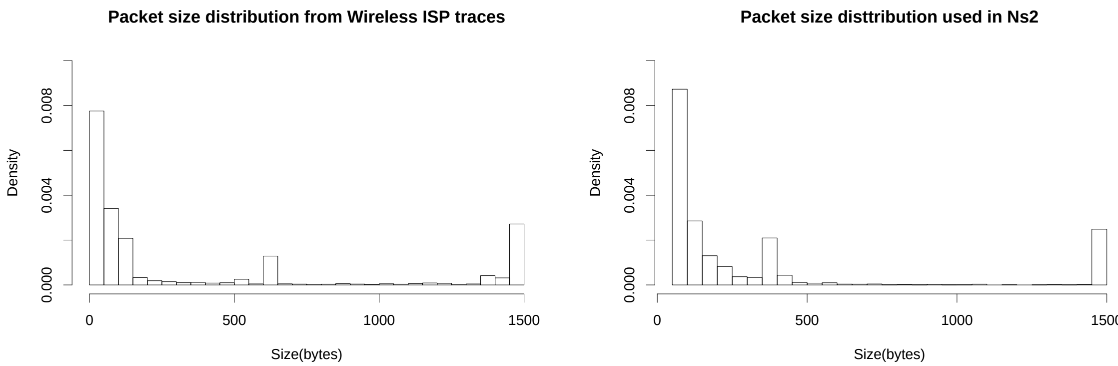
Figure 8: Simplified layout (in meters) for "rural" scenario

In order to estimate gains achieved by aggregation, we needed realistic traffic load. We achieved this by two means. First, we used a trace driven traffic generator which randomly sampled data from actual traces logged by a local wireless ISP. Second, we built a model based on these traffic to generate traffic inside *ns-2*. It is widely believed that network traffic is self-similar [14] [15]. Such network traffic can be modeled by multiplexing a large number of ON/OFF sources that have ON and OFF period lengths that are heavy tailed. Such a source send data at constant rate during the ON period and stay silent doing the OFF period. The heavy tailed nature is emulated by the Pareto distribution. The WWW traffic is also self-similar and the size distribution of the packets are also self-similar [16]. We used the *ns-2* Pareto traffic generator to emulate the inter-packet time distribution. However, this generator only sends packets of constant size. So, we further extended it to send variable sized data packets following Pareto distribution. However, we needed to clip this distribution beyond MTU (1500 bytes) and for packet sizes less than the sum of standard headers (IP, UDP and ethernet headers). The Pareto distribution is characterized by parameter α. We used the value of α for a inter-packet time distribution from the work cited in [16]. This was 1.21 for ON period. We also looked at the packet size distribution of actual traces logged by a local wireless ISP. Based on that we used a clipped Pareto distribution with mean packet size 450 bytes and the value of α as 1.09. The distribution of packet size is shown in figure 9.

4.2 Analysis of Noise Impact

It is conventional wisdom that packets in wireless networks should be shorter, rather than long, because errors are more likely to impact longer packets. We examined a number of wireless transmission error models to determine what impact such errors would have on packet aggregation.

The most commonly used error model for wireless networks is the *Gilbert-Elliot* model, which uses a Markov model that transitions between two states. In one state, there is a very low mean bit error rate and in the other state there is a higher mean bit error rate. Both states use a constant mean bit error rate. In such a model, let the probability of any bit being corrupted be p_e. Since all n bits of a message must be received correctly for the entire message to be received, the probability of correctly receiving a message is geometrically distributed, $P_m(n) = 1 - (1 - p_e)^n$. To a first approximation, the expected amount of data delivered by a message transmission is $E_m(n) = nP_m(n)$. Depending on the value of p_e, the value of E_m for larger packets



a) b)

Figure 9: Distribution of packet sizes a) Actual traces from a Wireless ISP b) Ns2 simulation

is less than that for smaller packets. Figure 10(a) plots $E_m(500)$ *v.s.* $E_m(1500)$ for p_e varying from 0.00001 to 0.01. For values greater than ≈ 0.001, the expected data delivered by the 1500-byte packet is less than that for the 500-byte packet. This difference is one reason that the 802.11b standard includes an automatic fragmentation mechanism; under conditions of high noise, the network interface may elect to fragment packets based on a fragmentation threshold.

This simple error model does not include the impact of overheads on packets. Overheads cause packets to be larger (decreasing P_m) but do not deliver additional data. Figure 10(b) shows the same model as Figure 10(a) with an overhead of 114 bytes, 11b standard, the approximate time overhead in the 802.11 packets [1]. In this case, $P_m, o(n) = (1 - p_e)^{(o+n)}$ and $E_m, o(n) = nP_m, o(n)$ where o is the overhead. The difference reduces the effectiveness of smaller message slightly; as the overhead increases, the difference between small and larger data payloads decreases.

The constant bit error model ignores the physical characteristics of transmission in the 802.11b model. The packet preamble is always transmitted at the base rate (1mb), and increases the relative overhead. Using our regression model from the maximum throughput measurements, a 1500-byte packet takes 1.88ms, of which 1.104ms is preamble overhead. Part of this time is actually "dead-time" to arbitrate media access and data is not actually being transmitted. We can use our 11mb/s regression model to compute the approximate maximum transmission rate for 1500 byte packets at 1mb/s – *i.e.* by taking $(1.88 - 1.104)/1500$ as the time-per-byte at 11mb/s and assuming that channel is 11-times more efficient, we arrive at an approximate transmission time of 9.7ms per packet. At that speed, the packet overhead is only 11% of the total packet time. If we assume that the errors are characterized by a *temporal error rate* rather than a *bit error rate* and we assume that the error rate is the same for the 1mb/s and 11mb/s portions of the message, then the characteristics of the expected data delivered by a packet changes considerably, as shown in Figure 10(c)[2]. In this case, we assume $P_m, o(n) = (1 - p_e)^o * (1 - p_e)^n$ and $E_m, o, r(n) = n * Pm(n/r)$ where n is message size, o is the amount of overhead (in bits) and r is the efficiency of the modulation technique relative to the base modulation rate (11 in our example).

[1]The mandatory ACK frame contains a 14 byte MSDU, 34 byte optional RTS and CTS (not included), two 18 byte PLCP headers, 38 byte MAC header and CRC, and 26 byte Ethernet MAC header and CRC.

[2]In this example, we use a error-rate-per-bit-transmission-time that is the same as bit-error-rate, and use an adjusted number of bits to provide an approximately comparable model. We do not claim this is an accurate error model or that these probabilities are realistic; we're only illustrating the difference between a *bit error model* and a *temporal error model*

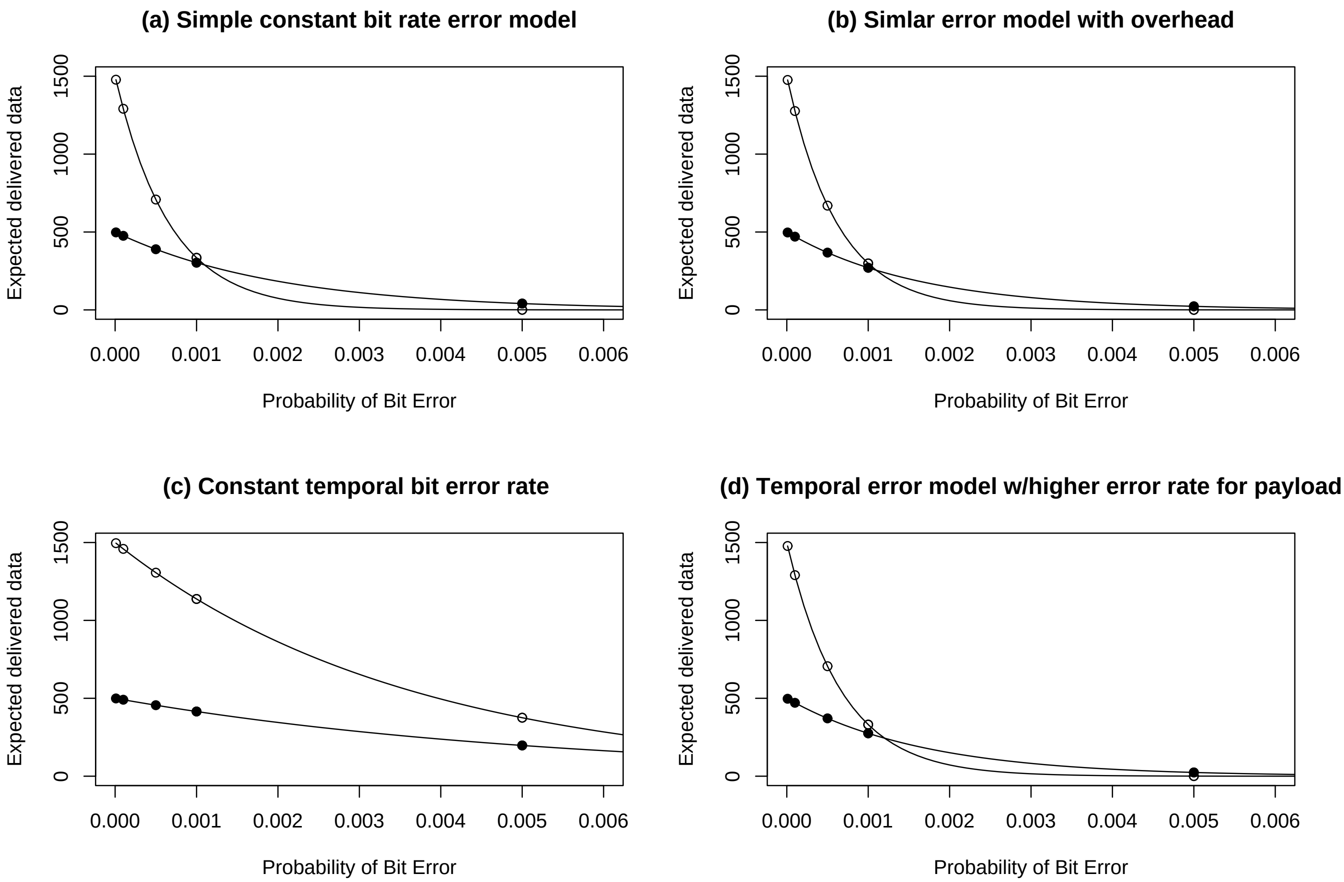


Figure 10: Expected delivered data *vs.* error rate for different error models

However, it's unlikely that the different modulation techniques have similar error rates. Indeed, Willig [17] measured error rates in an exceptionally noisy industrial environment and found that mean bit error rates were $2.5e^{-5}$ for the 1mb/s modulation and 0.0544 for the 11mb/s modulation. However, as the authors indicate, the 11mb/s modulation technique could not be used in this this exceptionally noisy environment, and they did not think these error rates were representative of most environments. If we assume a more conservative 10-fold increase in error rates for the 11mb/s portion of the message, the estimated throughput is shown in Figure 10(d); again, depending on the actual error probabilities used, there is a region where larger packets are beneficial and a region where smaller packets effectively deliver more data.

Most prior studies that indicated that larger packets significantly reduce the effective delivered data rate used the older 802.11 modulation techniques that offered 1mb/s or 2mb/s of bandwidth [18]. In these environments, the fixed preamble overhead was only 10-20% of the transmission time and even less of the actual "on air" data transmission time, and the model of Figure 10(a) might hold. However, other studies using similar equipment in a variety of common environments [19, 20] found few end-to-end packet errors. Moreover, even using the error rate for 1mb/s networks in a harsh industrial environment [17], the constant bit error model shown in Figure 10(a) should favor larger packets. In fact, that same study found that the source of corruption was usually biased to the early bytes of packets; if this was the case, than the effective delivered payload would be almost independent of packet size. Lastly, studies comparing interference between Bluetooth and IEEE 802.11b network streams [21]

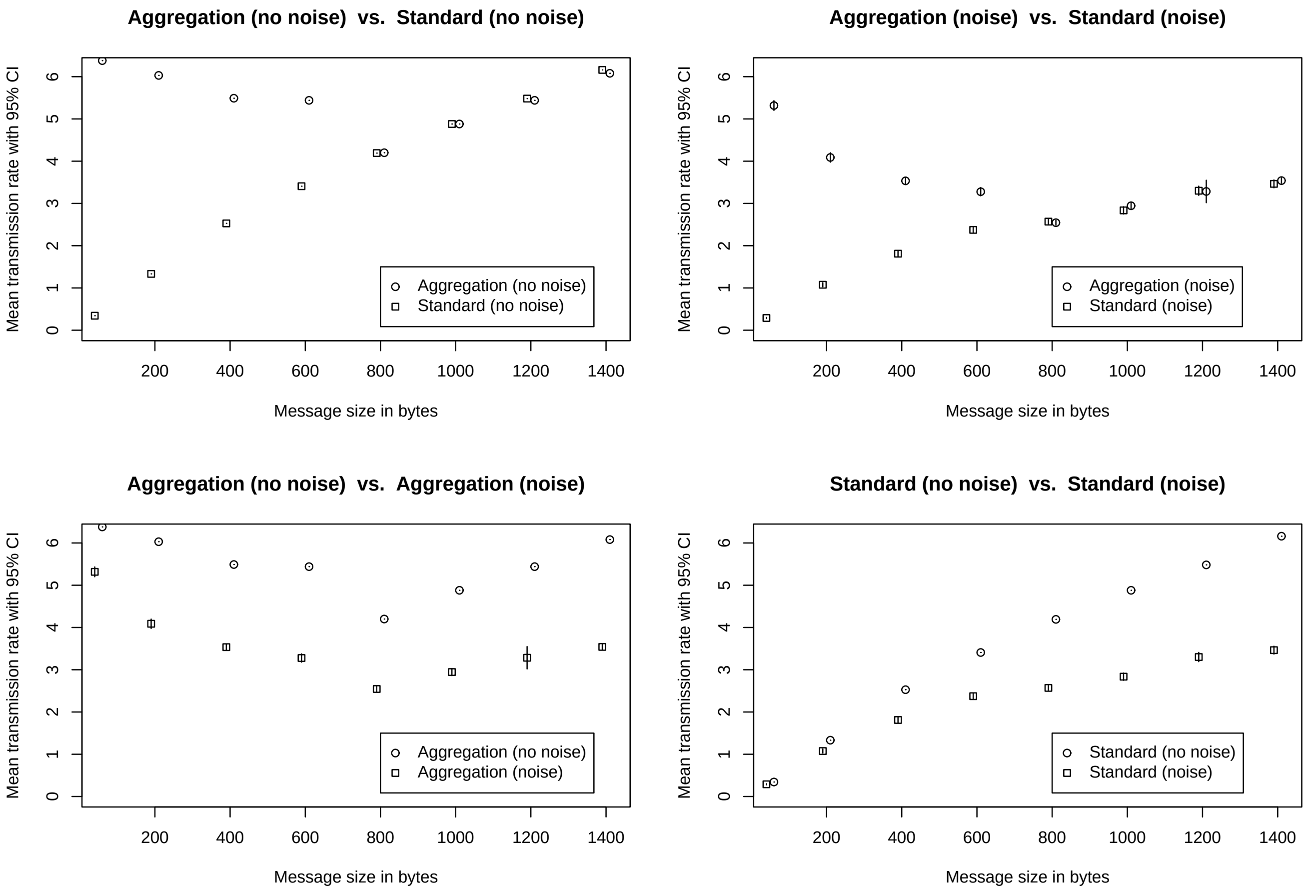


Figure 11: Throughput performance of aggregated and non-aggregated traffic, both in ambient conditions and the presence of induced noise

indicate that an 802.11b network running at 11mb/s experiences *a smaller* performance degradation than a 1mb/s network in the same environment and the shorter packet transmit time was given as the reason.

Thus, we encountered mixed results while searching for an explanatory model that would provide a solid basis for determining if larger packet sizes incur more packet loss. Clearly the larger per-packet overhead in the higher modulation 802.11b networks makes experience with the older 1mb/s networks less applicable.

Since our environment combines both simulation and a small experimental community network, we elected to use the existing 802.11b networking model in NS-2 for the simulation and use ambient and induced noise for the experimental network. We conducted a series of experiments using our deployment environment comparing the performance of standard (non-aggregated) and aggregated transmissions. We used an AT&T model 2455 phone, which operates in the 2.4GHz range, to cause interference via voice communication, similar to the method used in the Bluetooth study [21]. The measurements were conducted over the period of four hours with the placement of the phone remaining consistent during the experiments. The phone was left in "intercom" mode and placed within 15cm of the laptops used for the measurements. The results are shown in Figure 11. Figure 11(a) compares aggregation *vs.* standard network performance. The performance for aggregation approaches that of the standard network as packet sizes approach 800bytes because the MTU is 1500 bytes and can no longer hold multiple packets. Figure 11(b) shows that aggregation is still beneficial in the presence of noise. We are uncertain why the measurements for the aggregated 50 byte packets are higher than those for the 1400 byte packets; we believe this was due to the slightly larger mean packet size (1470 bytes) and changes in ambient noise. Figure 11(c) shows the impact of noise on the aggregated packets while Figure 11(d) shows the impact on non-aggregated packets. There is a significantly smaller impact on the non-aggregated packets, but the overall bandwidth is significantly lower.

The noise induced by our experiment caused an $\approx$ 15-44% bandwidth reduction depending on packet size, but overall, aggregation provides a higher delivered data rate even in the presence of this induced noise. For 50-byte packets, we achieve a 25-fold increase in packet size for a 2-fold increase in transmission time and 3-fold increase in packet loss; this still provides more than a 4-fold improvement in throughput.

5 Analysis

5.1 Validation of theoritical model for aggregation

We had earlier stated that aggregation should reduce contention for the wireless media (see section §2.1). In order to validate that result, we build a simple NS2 simulator scenario. The scenario consisted of four nodes placed such that each node can overhear any ongoing transmission (this would increase the contention for the media). Two oppositely placed nodes were paired up and they sent constant amount of data in each run. In each run, a different packet size was chosen which was usually a multiple of a base packet size. Each node transmitted the same amount of data via packets of this chosen size. The idea was to encapsulate the notion of packet aggregation by choosing a packet size which was multiple of a base size.

The result of the measurements are given in table 5.1. Packets of size 125, 250, 500, and 1000 bytes were chosen to send the same amount of data. In the table 5.1, the ratio of packets delivered through aggregation and without is denoted by η_{agg}/η_{base}. t_{con_agg} represents the contention time (in seconds) spent per packet by aggregation. $\sum_i^n t_{con} + (n-1).T_o$ denotes the contention time (along with overhead) spent when packets were sent without aggregation. Thus, in all cases where aggregation delivers more data (i.e., when $\eta_{agg}/\eta_{base} > 1$), we find that contention time for the media is reduced. This can be seen by inspecting the third and the fourth column in 5.1. Thus, this simplistic set of the *ns-2* simulations validate the fact that aggregation reduces the contention for the media.

Table 1: Validation of theoretical model for aggregation

Packets Aggregated	η_{agg}/η_{base}	t_{con_agg}	$\sum_i^n t_{con} + (n-1).T_o$
1	1	0.001341	0.001341
2	1.37	0.002310	0.003334
4	1.65	0.004341	0.007308
8	1.83	0.008330	0.015264

Figure 12 shows a boxplot of the packet distribution for N.Boulder neighborhood scenario. As can be seen from this figure, the packet latency distribution is very heavy tailed. This is due to the fact that the scenarioes used represented an open network, that is, the nodes keep sending UDP streams irrespective of the fact that next neighbour is unable to handle incoming traffic. Long latencies (>1s) were observed at the bottlenecks, e.g. at the servers. Because of the heavy tail nature of the packet delay distribution, the median is a better representative of system performance than the mean.

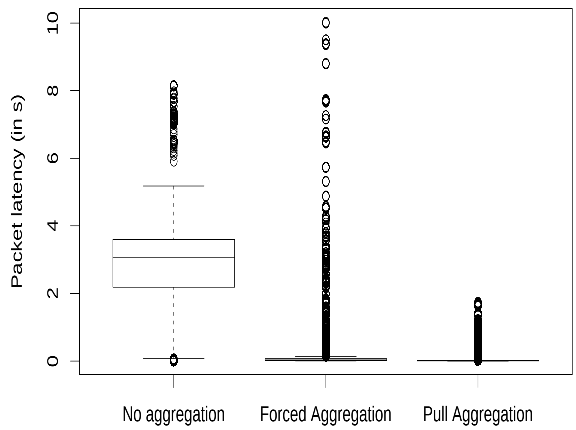
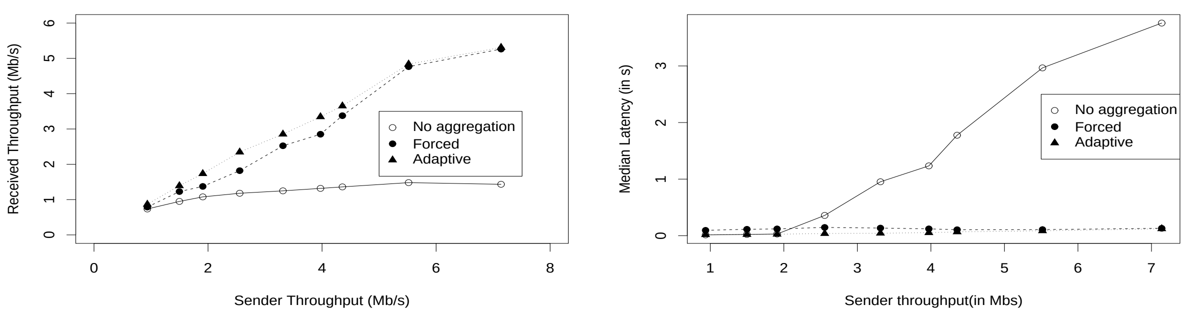
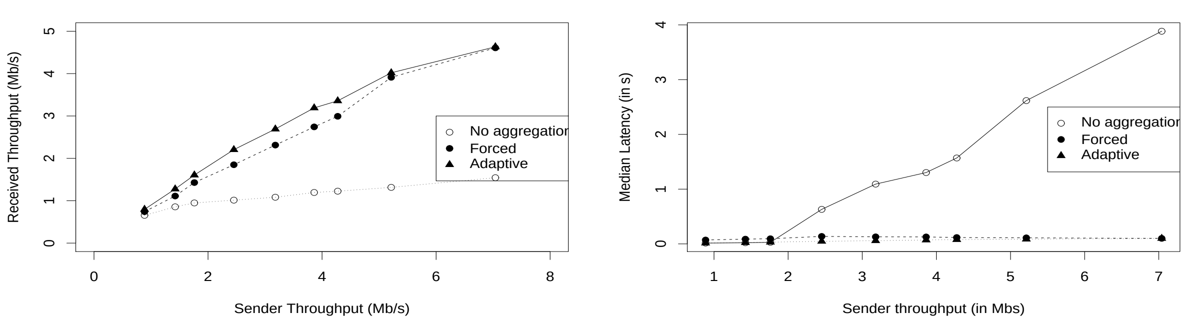


Figure 12: Paket latency distribution for one of the scenarios. The long tails emphazies the fact they are packet latency is heavy tailed distribution - hence median and not mean give better representation



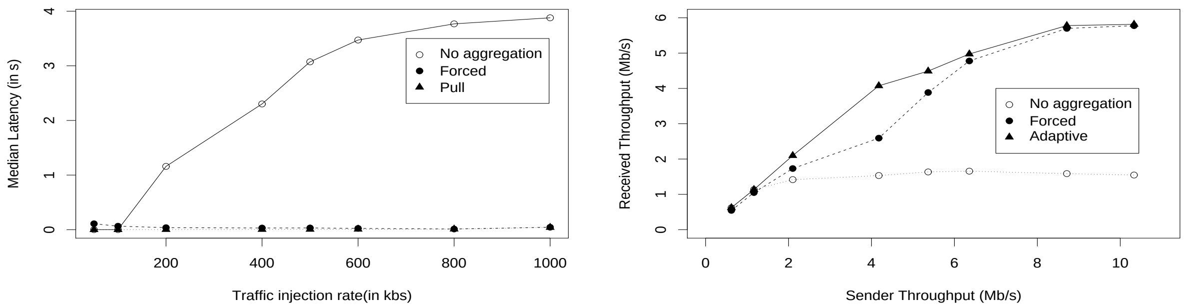
a) b)

Figure 13: Client-server scenario on N.Boulder topology a) Median Latency b) Throughput



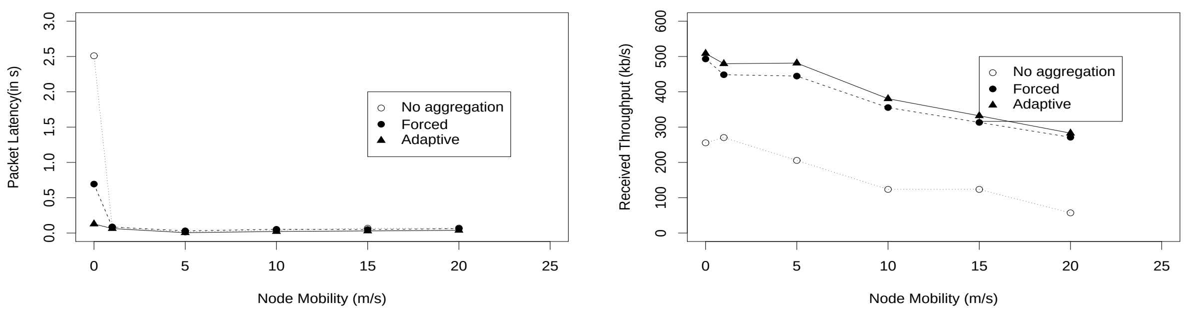
a) b)

Figure 14: Client-server-gateway scenario on N.Boulder topology a) Median Latency b) Throughput



a) b)

Figure 15: Rural topology a) Median Latency b) Throughput

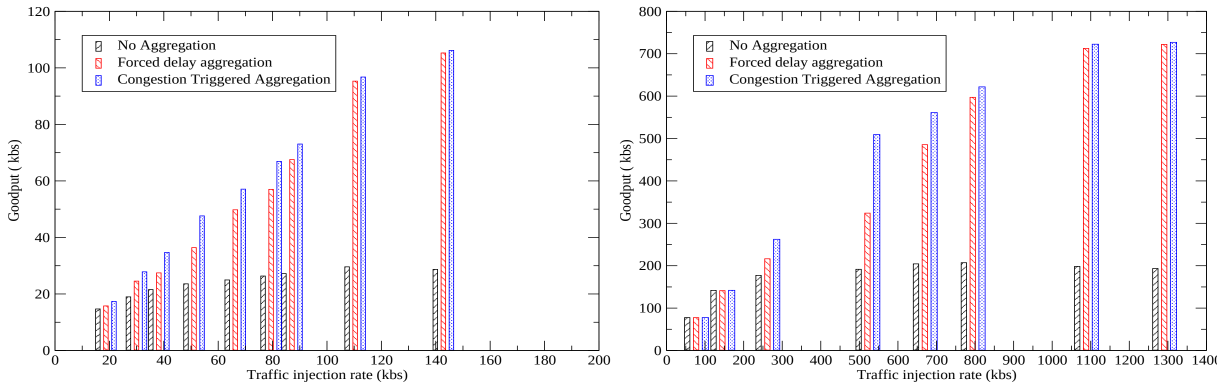


a) b)

Figure 16: Random pair communication model, random topology a) Median Latency b) Throughput

Looking at the figure 14(b) and 13(b), we see that forced aggregation increases the median packet delay relative to non-aggregated traffic for low traffic injection rate. As we increase the injection rate, more packets get aggregated and thus it reduces the mean packet delay suffered. The pull or congestion triggered aggregator performs best as it does not introduce extra delay as well as it reduces mean packet delay. Aggregation also reduces the variance in the delay (this can be deduced by looking at the third bar in the boxplot which signifies the third quartile in figure 12) Again this is expected since by aggregating more packets, it avoids or reduces the formation of long queues.

Figure 14(a) and 13(a) shows the *goodput*, or fraction of delivered packets in the suburban scenario. In all cases, forced delay aggregation increases the goodput by making more efficient use of the media compared to non-aggregated case. Congestion triggered aggregation performs better since it does not involve any dead period (while its waiting for fixed delay to aggregate packets). It is important to note that both forms of aggregation performs equally well under heavier traffic since long queues are formed at each node and thus enabling it to send MTU size packets. Under lower traffic rate, congestion triggered aggregation sends more aggregated packets. Under all traffic rate, bottleneck nodes performs equally well on both forms of aggregation since they have formed long queues and therefore always have packets to aggregate. Aggregation improved goodput from 60% to 83% (a a 30% improvement) for light traffic scenarios and by 270% for heavier traffic rates.



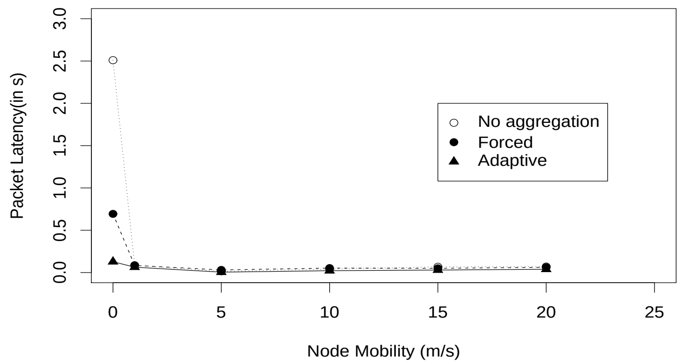
a) b)

Figure 17: Goodput reveived at each node a) N.Boulder topology. Even though delivered end-to-end throughput remains high, goodput received at each node is agonizingly low becuase of use of same channel and omni-directional antennas. b) Rural topology. Careful use of multi channels and uni-directional antennas substantially improve the goodput received at each node.

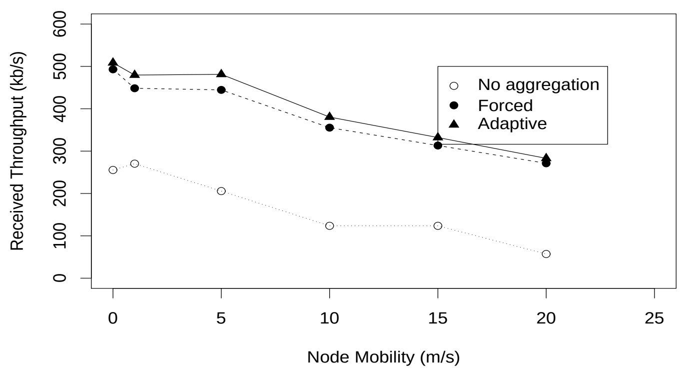
Overall, as seen from figure 17(a) the throughput rates are poor for individual nodes – this occurs because of the interference in large *ad hoc* networks without proper channel assignment and antenna placement.

Figure 15 shows the effect on latency and overall end-to-end goodput on the rural scenario. Use of unidirectional antennas and separate channels for each subnets and backbone improves the throughput substatially for aggregation. There was an improvement of 250% in end-to-end throughput. The latency, as expected, improved with traffic injection rate since more packets get aggregated. Further careful assignment of channels and directional antennas improves the goodput received at each individual node. For example, in figure 17(b), each node received up to 730kbs.

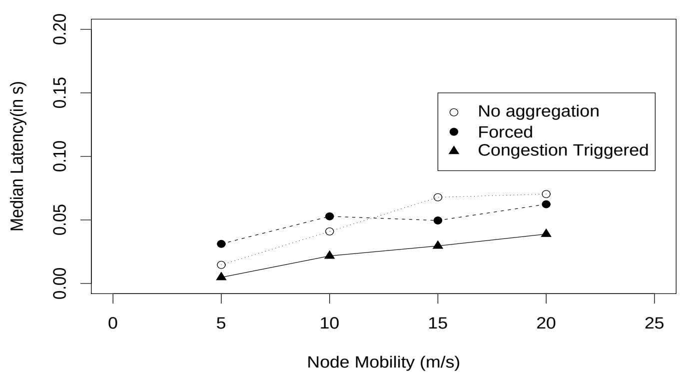
Figure 18 and 19 summarizes the results for the peer-to-peer communication model on a randomly selected topology. We used 7 and 10 communicating pairs respectively. The rest of the nodes simply acted as a route forwarder. As can be seen from the figure 18(b) and 19(b), throughput drops for all cases as we increase the node mobility. This is expected since links will break more frequently and thus requiring re-initiation of route discovery. It is interesting to analyze why the aggregation is able to outperform the non-aggregation case in this scenario besides the reasons discussed in earlier scenarios. The non-aggregation case is unable to send any packets except to its immediate neighbor, implying that it is unable to handle congestion caused by route discovery. On the other hand, aggregation performs better as it is still able to route packets by virtue of reduced media contention. This was verified by looking at the median hop count of packets in both cases. The median hop count for non-aggregation case was *one* while for aggregation was *two*. The latency distribution for delivered packets is summarized in graphs 18(a) and 19(a). Interestingly, the latency for all cases in the no motion scenario is high. This is often encountered when choosing random pairs on a randomly placed topology. When nodes are static, any misplaced nodes (for the purpose of routing) won't be able to deliver packets or take a long route. Mobility homogenizes node positions and hence does not suffer from such problems. The packet latency for non-aggregation case is fairly low for high mobility. This can again be explained once we realize that the packets are only being delivered to its immediate neighbor (see figure 20(a)). Further, the non-aggregation case sent the most number of route discovery packets, thus increasing congestion since they are broadcast messages. (see figure 20(b)). Congestion triggered aggregation on the other hand has better latency than the non-aggregation case under all mobility cases. Forced aggregation has worse latency compared to non-aggregation but it is still able to route packets.



a)

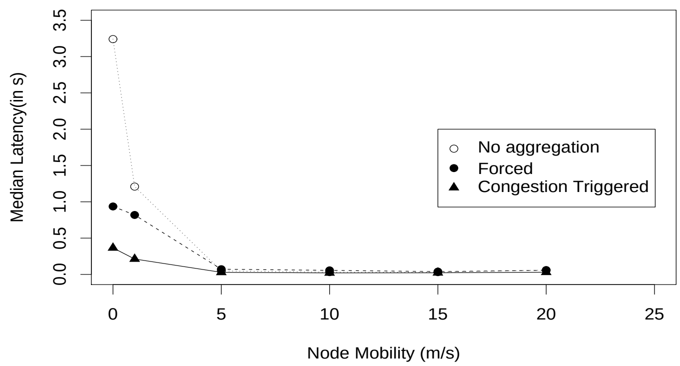


b)

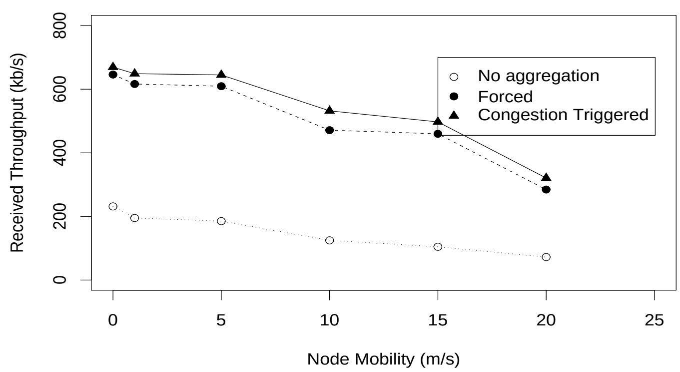


c)

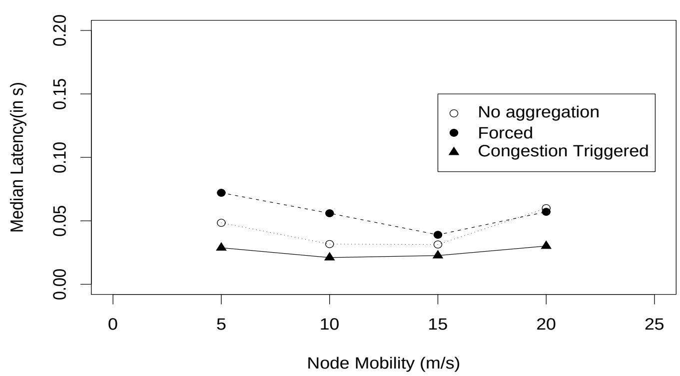
Figure 18: Peer-to-peer communication model on a randomly placed topology. The communicating nodes were also chosen randomly and paired up. This figure summarizes the results for 7 pairs or 14 communicating nodes amongst 50 nodes a) Median Latency for delivered packets b) Throughput c) a closer look at latency



a)

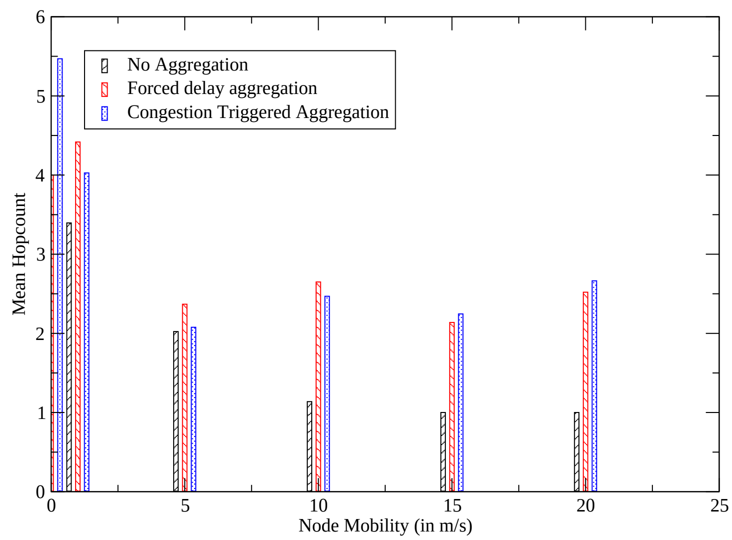


b)

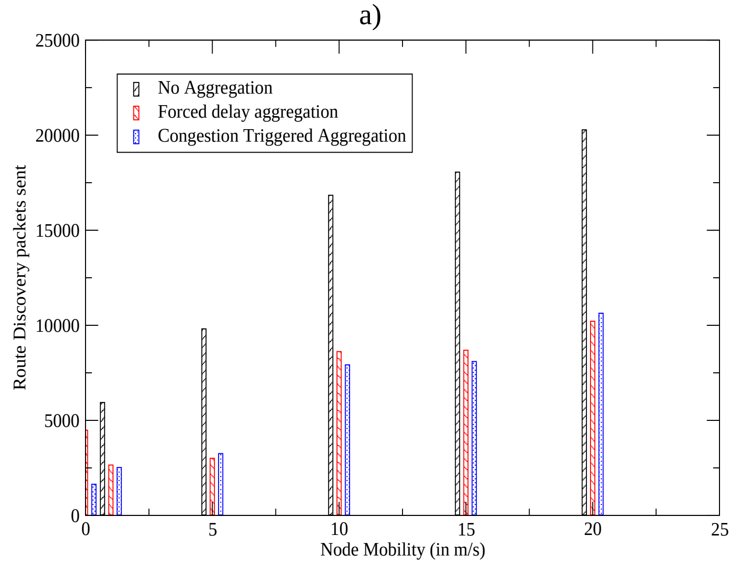


c)

Figure 19: Peer to peer communication model on a randomly placed topology. This figure summarizes the result for 20 communicating random-20. a) Median latency for delivered packets b) Throughput c) a closer look at the latency



a)



b)

Figure 20: With increasing mobility, non-aggregation case is unable to route packets. It can be seen from mean hopcount.a) Mean hop-count b) Number of Route Discovery messages sent

Scenario	Send rate	No aggregation	Forced	Congestion Triggered
Client-Server	795kb	360kb	510kb	626kb
Gateway	977kb	409kb	535kb	663kb
Rural scenario	406kb	202kb	326kb	393kb

Table 2: Summary of throughput achieved by aggregation when using trace driven traffi c generator on various *ns-2* scenarios

Scenario	No aggregation(in s)	Forced(in s)	Congestion Triggered(in s)
Client-Server	0.605	0.675	0.153
Gateway	0.985	1.120	0.078
Rural	0.894	0.673	0.165

Table 3: Summary of median latency observed when using trace driven traffi c generator on various *ns-2* scenarios

5.1.1 Results using trace driven traffi c generators

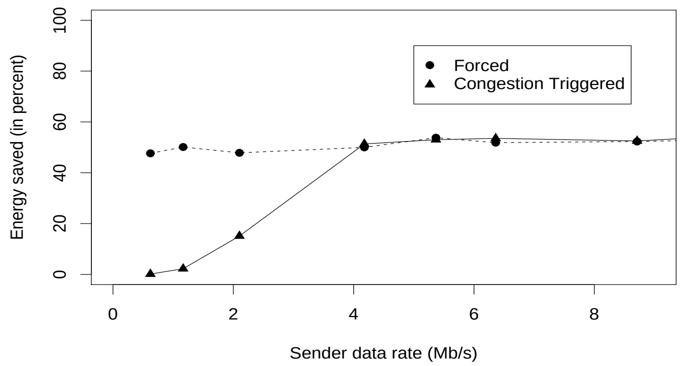
Table 2 and 3 summarizes the results for different *ns-2* scenario while using a trace driven traffic generator. The traces used were logged by a local wireless ISP. Looking at table 2, it is noticed that the gains in throughput for aggregation (both forced and congestion triggered) are smaller than those reported while using Pareto traffic model. The congestion triggered aggregation achieves 94% throughput for rural scenario and 73% throughput for N.Boulder scenario. This is due to the fact that sender traffic rate is fairly low, roughly around 50kbs. Again congestion triggered aggregation outperforms forced aggregation. Table 3 summarizes the median latency observed.

5.1.2 Effect of aggregation on energy savings

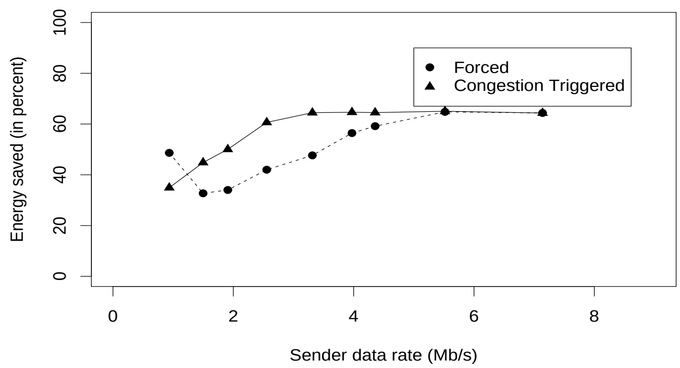
Energy has become a central design constraint in mobile computing. Aggregation also helps to reduce the active transmission power consumption. Transmission energy for packets can be approximated by a linear model consisting of a fixed energy overhead and another factor dependent of packet size. For small packets, the fixed overhead forms a significant portion of the transmission energy. Again, it is more efficient to send larger packets than small packets since both incur same fixed overhead. Details of this are given in [22]. The energy savings for different scenarios are given in figure 21. It is noticed that the forced aggregation gives better energy savings as compared to congestion triggered in an uncongested scenario. This is because congestion triggered aggregation aggregates fewer or none packets. This is seen in the initial portion of figure 21(a). However, with increasing congestion, the congestion triggered aggregation gives better energy savings since it is able to send more aggregated packets (This can be seen in the figure 21(b) and 21(c)). Once the network is saturated, both forms of aggregation give similar energy savings.

5.1.3 Effect of aggregation on TCP

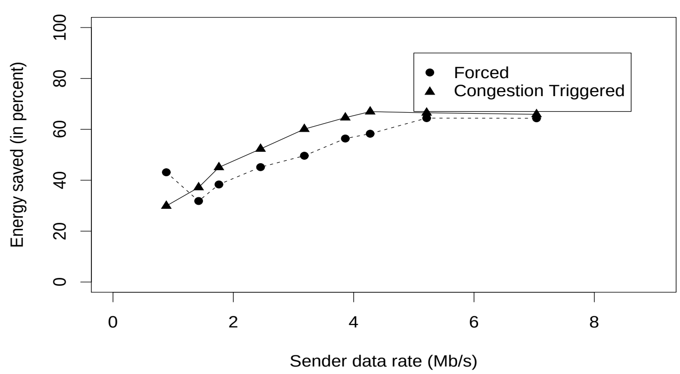
It is also interesting to evaluate the effect of aggregation on TCP streams. To do so, we used a peer-to-peer communication model on a randomly placed topology. All communicating nodes ran TCP Reno and used delayed acknowledgements. FTP traffic was chosen to emulate large file transfers. The results of the experiments are shown in figure 22. It shows number of packets received per second during the course of simulation. Both the base case(no aggregation) as well as aggregation deliver equal number of packets, showing that aggregation does not degrade large TCP file transfers.



a)



b)



c)

Figure 21: Transmission energy saved via aggregation in a) Rural scenario b) Client-server scenario c) Client-server-gateway scenario

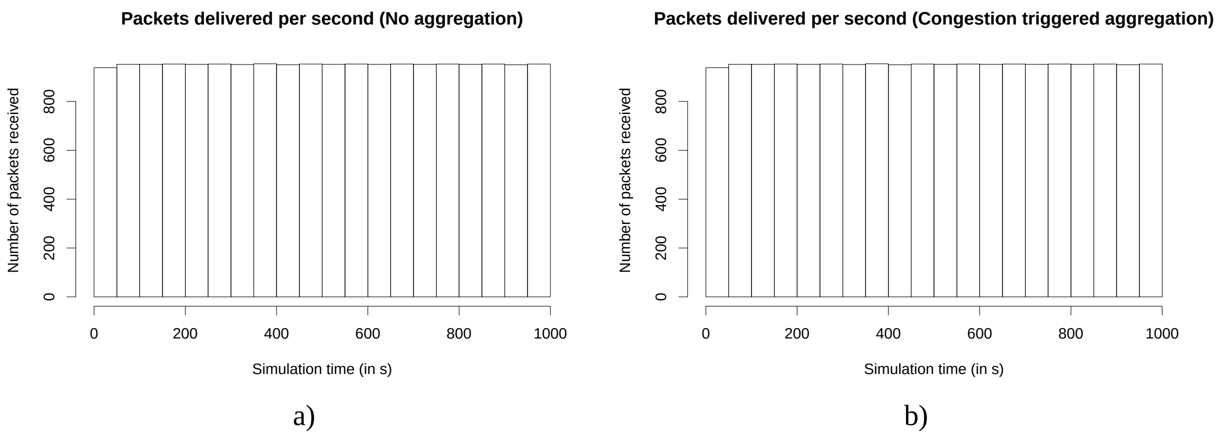


Figure 22: Number of packets sent in simulation involving TCP/IP a)No Aggregation b) Congestion triggered aggregation

6 Related Work

The most directly related work is by KarlNet Inc. [10], who use the *forced delay aggregation* model without classification for QoS to improve throughput. We have shown that our *congestion triggered aggregation* has better performance.

Several researchers have studied packet size adjustments to optimize throughput on a wireless channel. Specifically, Modiano [23] has designed an adaptive algorithm that optimizes packet size for wireless automatic repeat request protocols. The algorithm estimates the current bit-error-rate using the recent history of acknowledgments. The optimal packet size is derived based on a Gilbert–Eliott (Markovian) channel model. Ci and colleagues [24] adapted similar ideas to an IEEE 802.11b environment. They approach the prediction problem with a Kalman filter and present simulation results based on a channel with constant bit-error-rate. Ci and coworkers also designed an algorithm that adaptively controls the IEEE 802.11 packet fragmentation threshold to optimize throughput. This approach splits larger packets to obtain the optimal packet size; exactly opposite to our solution.

Zorzi and Rao [25] have surveyed error models for wireless networks; they emphasize the importance of higher-order statistics for performance evaluation of wireless protocols. A channel model with independent and identically distributed errors does not capture the correlation between errors found on a real channel. These error correlations can have significant impact on the performance of network protocols.

The described simulation results have been supplemented through the following multiple experimental studies on IEEE 802.11 networks. Nguyen and colleagues [18] collect traces from an older 2Mbps Lucent WaveLAN card to validate error models. Concerning packet size, they find an exponential relationship—that is, the “error rate doubles for every 300-byte increment of the packet size.” They also conclude that errors can be more accurately modeled through extensions to the Gilbert–Eliott that model the burstiness with Pareto and exponential distributions.

Eckhard and Steenkiste [19, 20] provide measurements that detail the cause of an error. Specifically, they distinguish truncated packets, corrupted packets (full packet received, but the packet contains errors), and lost packets (the receiver never reported the packet). With respect to packet size, they observe that the packet loss rate is independent of packet size, while the truncation rate increases with greater packet sizes. The results on corruption rate are not so clear: in one experiment with intermittent noise it does correlate with packet size, while in another experiment with constant noise it appears independent. In a noisy industrial environment, Willig and coworkers [17] found a high time variability in the mean bit error rate. Mean bit error rates seemed insensitive to packet size. Overall, however, the Gilbert–Eliott model proved useful for characterizing the noise patterns.

The evaluation of TCP performance over wireless networks also received a fair amount of attention. Balakrishnan [] found that performance is negatively affected by bursty noise on the wireless channel. TCP was designed with the assumption that packet loss is predominantly due to network congestion; thus, TCP flow control slows the sending rate upon packet loss and the slow start feature only gradually increases the rate afterwards. However, on the wireless channel packets are lost mainly through bursty noise—rendering a slow start highly inefficient.

7 Discussion & Conclusions

Our experiments have

- Successfully demonstrated the utility of packet aggregation in community networks using workloads representative of such networks. This includes conventional 'adhoc' scenarios that use omnidirectional antennas with isotropic propagation as well as wireless environments that use more directional antennas, although the "adhoc" environments have low overall throughput,
- Demonstrated a range of performance improvements, ranging from 40-250% improvement in "adhoc" environments to 10-fold improvement in simpler scenarios that allow better media utilization. These improvements should be robust across different network technologies because most such technologies have a) lead-in or carrier-sense periods or b) contention models; both of these factors contribute to overhead which increases the effectiveness of packet aggregation,
- Shown that aggregation need not cause additional packet delay in practice by using a naturally adaptive packet aggregation technique based on contention for the wireless media,
- Emphasized the fact that practical community networks either need to use more efficient PHY and MAC layers or must emphasize proper antenna design and placement,
- Shown empirically that aggregation appears to be robust in the presence of some common noise sources for 802.11b. While not definitive; our work, when combined with that of others [21] indicates that simple mean bit error rate models need to be be augmented with experimental validation

We intend to use the packet aggregation scheme in our community networking deployment. We are focusing on environments that emphasize a diversity of channels and sectored antennas to build scalable and secure community networks. In those environments, packet aggregation should improve overall media utilization.

References

[1] HPWREN. The high performance wireless research and education network (hpwren). http://hpwren.ucsd.edu, 2001.

[2] Locust World. The locust world network project. http://www.locustworld.com, 2002.

[3] Brisbane Mesh. The brisbane mesh network. http://www.itee.uq.edu.au/ mesh/, 2001.

[4] Seattle Wireless. The seattle wireless broadband coop. http:////www.seattlewireless.net/, 2001.

[5] J. Li, C. Blake, D. DeCouto, H.I. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2001)*, Rome, Italy, July 2001.

[6] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, and M. Frans Kaashoek. The click modular router. *ACM Transactions on Computer Systems*, 18(3):263–297, August 2000.

[7] Charles E. Perkins and Elizabeth M. Royer. Adhoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, Feb. 1999.

[8] Alex Rozin. Rstp 802.1w library. available from `http://rstplib.sourceforge.net`.

[9] Michael Neufeld, Ashish Jain, and Dirk Grunwald. Nsclick:: bridging network simulation and deployment. In *Proceedings of the 5th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems*, pages 74–81. ACM Press, 2002.

[10] KarlNet Inc. Turbocell white paper. Available for download from http://www.karlnet.inc/Documents/WhitePaper/TurboCellWhitePaper, 2003.

[11] Microsoft. Microsoft terraserver web service. http://terraserver.microsoft.com, 2001.

[12] SDTS. Spatial data transfer standard. http://mcmcweb.er.usgs.gov/sdts/, 1995.

[13] USGS. Digital line graph data. http://edc.usgs.gov/geodata/, 2002.

[14] W. Willinger D.V. Wilson W.E. Leland, M.S. Taqqu. On the self-simmilar nature of ethernet traffi c(extended version). In *IEEE/ACM Transactions on Networking*, 1994.

[15] Vern Paxson and Margaret M. Recker. The failure of poisson modeling. In *In Proceedings of SIGCOMM*, 1994.

[16] Mark Crovella and Azer Bestavros. Explaining world wide web traffi c self-similarity. Technical Report 1995-015, 29, 1995.

[17] A. Willig, M. Kubisch, C. Hoene, and A. Wolisz. Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer. *IEEE Transactions on Industrial Electronics*, 49(6):1265–1282, Dec 2002.

[18] Giao T. Nguyen, Randy H. Katz, Brian Noble, and Mahadev Satyanarayanan. A trace-based approach for modeling wireless channel behavior. In *Proceedings of the 28th conference on Winter simulation*, pages 597–604. ACM Press, 1996.

[19] David Eckhardt and Peter Steenkiste. Measurement and analysis of the error characteristics of an in-building wireless network. In *Conference proceedings on Applications, technologies, architectures, and protocols for computer communications*, pages 243–254. ACM Press, 1996.

[20] David A. Eckhardt and Peter Steenkiste. A trace-based evaluation of adaptive error correction for a wireless local area network. *Mobile Networks and Applications*, 4(4):273–287, 1999.

[21] N. Golmie, R. E. Van Dyck, and A. Soltanian. Interference of bluetooth and ieee 802.11: simulation modeling and performance evaluation. In *Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 11–18. ACM Press, 2001.

[22] Jing Deng Feng Zhao Marco Gruteser, Ashish Jain and Dirk Grunwald. Exploiting physical layer power control mechanisms in ieee 802.11b network interface. Technical Report CU-CS-924-01, 2001.

[23] Eytan Modiano. An adaptive algorithm for optimizing the packet size used in wireless arq protocols. *Wireless Networks*, (5), 1999.

[24] S. Ci and H. Sharif. Adaptive optimal frame length predictor for ieee 802.11 wireless lan. In *Proceedings of the 6th International Symposium on Digital Signal Processing for Communication Systems (IEE DSPCS'2002)*, 2002.

[25] M. Zorzi and R.R. Rao. Perspectives an the impact of error statistics on protocols for wireless networks. *IEEE Personal Communications*, 6(5):32–40, Oct 1999.