FACTA UNIVERSITATIS (NIŠ)
SER.: ELEC. ENERG. vol. 17, April 2004, 53-68

Performance Evaluation of an IEEE 802.11b Ad-hoc Network

Srdjan Krčo, Marina Dupčinov, and Seán Murphy

Abstract: The performance of an IEEE 802.11b ad-hoc network that uses the AODV (Ad hoc On-demand Distance Vector) routing protocol is evaluated. One significant issue relating to the behavior of WLAN cards that has considerable impact on AODV performance was observed during the initial testing of the system and it is discussed and a solution proposed. Some aspects of the network performance are then assessed for several scenarios with low mobility. Route discovery latency results indicate that it is possible for mobile applications to operate reasonably well over ad-hoc networks in light to moderate traffic. UDP throughput results indicate that such networks could support tens of users using low-bit rate applications or possibly higher bit rates if applications generate data in bursts. Finally, some problems with TCP operating in this context were observed.

Keywords: Ad hoc networking, routing, IEEE 802.11b

1 Introduction

Ad-hoc networks are formed by users or devices wishing to communicate, without the necessity for help or existence of any infrastructure or centralised administration. These networks can function as standalone networks meeting direct communication needs of their users or as an addition to infrastructure based networks to extend or enhance their coverage. Applications of ad-hoc communication include emergency/disaster situations, military communication, sensor networks as well as commercial scenarios like sharing files at a meeting.

Manuscript received November 21, 2003.

S. Krčo and M. Dupčinov are with Ericsson Expertise Ireland, Radio House, Beech Hill, Clonskeagh, Dublin 4, Ireland (emails: [srdjan.krco, marina.dupcinov] @ericsson.com). S. Murphy is with the Department of Computer Science, University College Dublin, Belfield, Dublin 4, Ireland (email: sean.murphy@iname.com)

A common characteristic of all ad-hoc networks is the use of a short-range wireless technology (Bluetooth, WLAN, UWB, etc.) for communication between the network nodes. Each node in an ad-hoc network has at least one wireless access interface, is free to join or leave the network at any time and is usually mobile. Due to the limited range of the wireless interfaces used, multiple hops may be needed for communication. Various routing protocols have been proposed to facilitate route establishment and maintenance in such scenarios. As nodes in the network can move freely and randomly, frequent topology changes occur and routing protocols have to be able to cope with such changes efficiently. In addition, susceptibility of wireless links to interference can lead to sporadic connectivity patterns that can cause various problems for the route establishment and maintenance algorithms.

In this paper we present results of network performance experiments from an IEEE 802.11b based network that uses the *Ad-hoc On-demand Distance Vector* (AODV) routing protocol [1]. The contributions in this paper are twofold:

- A performance evaluation of an IEEE 802.11b based ad-hoc network which uses AODV is described;
- A description, analysis and solution of an important problem with the use of AODV due to the different range of broadcast and unicast packets are presented.

The remainder of this paper is organised as follows. The next section gives a high-level overview of the AODV routing protocol operation. Section 3 explains an important issue observed during the implementation and testing of this protocol. A solution to the observed problem is given in section 4. Section 5 presents some of the results obtained from testing live, WLAN 802.11b based, ad-hoc network. Section 6 concludes the paper.

2 Overview of the AODV Routing Protocol

There are number of routing protocols proposed for IP based ad-hoc networks [1], [2]. They have adopted different approaches in an effort to optimise various routing parameters. However, none of the chosen routing strategies have proved to be the best in all scenarios, but depending on the scenario (mobility, size of the network or traffic patterns and applications used), various protocols may prove to have more or less advantages/disadvantages.

IETF's Manet working group is focused on standardization of IP routing protocols suitable for wireless environment. Currently, there are 4 routing protocols, AODV (Ad-hoc On demand Distance Vector), DSR (Dynamic Source Routing), OLSR and TBRPF (Topology Dissemination Based on Reverse-Path Forwarding), proposed by this group [2]. AODV has experimental RFC status, while others are expected to gain the status soon. In order to evaluate the performance of a real ad-hoc network, we have implemented the AODV routing protocol.

AODV is a reactive routing protocol. It is developed for use by mobile nodes in an ad-hoc network and enables dynamic, self-starting, multi-hop routing between participating nodes.

2.1 Route discovery

When a route is needed, the source node broadcasts a *Route Request* message (RREQ). Any node with a route to the destination responds to a RREQ by sending a *Route Reply* message (RREP) to the source node. If a node does not have a route to the requested destination, it rebroadcasts the RREQ and propagates it to its neighbours. If the source node receives a RREP, the route is established and the source can start using it.

In order to avoid unnecessary flooding of the network and thus reduce the routing overhead, the expanding ring search technique is used, i.e. the TTL (Time To Live) value of the RREQ message is increased in steps from the minimum to the maximum value. Initially, the TTL is set to the minimum value (TTL=1) and a timer is started. If no RREP message is received before the timer expires (2$*$10ms$*$TTL), a RREQ message is broadcast again, but with increased TTL value (TTL=3). The source node again waits until either the timer expires or a RREP message is received. Each time a RREQ message is broadcast the TTL value is increased up to the maximum value and the timeout value for the timer is increased accordingly. If no RREP is received after the RREQ is broadcast twice with the maximum TTL value (TTL_threshold = 7), the whole network is flooded. If the route is still not found, operating system informs the upper layers that it is not possible to establish the connection.

2.2 Route maintenance

The route is maintained as long as it is used. Due to the dynamic topology of ad-hoc networks and the wireless environment, routes can break quite frequently. When a link break occurs, the node upstream of the link break notifies neighbours about all destinations that became unreachable using a *Route Error* message (RERR). Notification of the link break is propagated to the source node; it can re-establish a route using the route discovery mechanisms described above.

2.3 Route expiry

If a route is not used for a certain time, a timer will expire and the route will be removed from the list of active routes (active routes are routes currently in use) in the routing table. However, it will not be deleted from the routing table for an additional time period. If the same route is needed in that period, the existing parameters will be used for a faster route establishment. When this additional time expires, the route is deleted from the routing table.

2.4 Local connectivity

A node may offer connectivity information by broadcasting `Hello` messages periodically. When a `Hello` message is received, a route to the neighbour should be added to the routing table if it does not already exist. If the route exists, its lifetime is increased. When the topology of the ad-hoc network changes and `Hello` messages are not received for a short period of time, the route expires.

3 AODV Performance Problems Arising from 802.11b Implementation

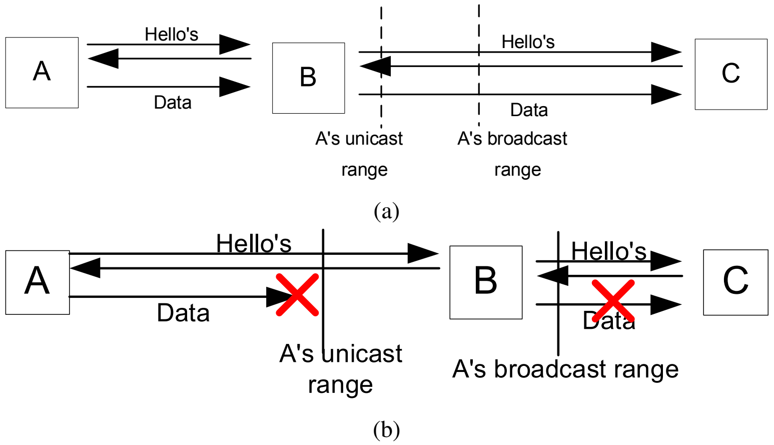
An important issue was identified during the initial system test which had a very significant impact on the performance of the system and hence the system usability. It pertains to a difference in the transmission range of different types of packets [3, 4, 5].

All experiments were based on a linear chain topology of nodes. Five nodes were used, four fixed and one mobile. The communication was established between the mobile node and the first node in the chain. The mobile node had different positions in experiments, which resulted in different path lengths (1 to 4 hops).

Frequently a very poor network performance was observed, i.e. only a very small fraction of user traffic was successfully transmitted from a source to a destination, although the nodes' routing tables were as expected: routes from the source to the destination nodes existed, which meant that all routing protocol messages were transmitted and received correctly.

In order to locate the problem we performed some tests on the MAC layer. The Orinoco driver, which we were using with ELSA Airlancer MC-11 PCMCIA cards, provided functionality to verify that the receiver acknowledged transmitted packets. Initially, we placed nodes as in Figure 1a so that node B had good communication links with other nodes. `Hello` messages were exchanged correctly between nodes A-B and B-C and each node added the other to its routing table. Then, node A requested a route to C by broadcasting a RREQ message. Node B received this

RREQ and since it had a route to C in the routing table, it sent a RREP message back to node A.



(a)

(b)

Fig. 1. Difference in transmission ranges test. The initial topology is illustrated in (a) and the topological configuration which can be used to demonstrate the problem is depicted in (b).

After receiving the RREP, node A added a 2-hops route to node C via node B to its routing table and we were able to exchange data between nodes A and C successfully. Then, node B gradually moved away from node A towards node C (Figure 1b). As the quality of the A-B link decreased, less and less of the user data was received successfully by node C (number of unacknowledged messages was increasing) up to the point when almost no unicast packets (user data) were received successfully.

However, because the routing tables were as expected at that moment, we inferred that the control traffic was transmitted correctly. Thus, there was a noticeable difference between the behaviour of routing protocol messages and user data traffic. On further investigation, this was found to be attributable to the fact that the 802.11b MAC mechanism differentiates between the so-called unicast and broadcast packets: user data was sent in unicast MAC frames, while the most routing protocol messages were sent in MAC broadcast frames.

We observed that the rate at which broadcast packets were sent was set to 2Mbit/s while we were using 11Mbit/s for user data.

Since transmission power was constant, packets transmitted using lower bit rates have more energy per transmitted bit and hence can be sent further than their higher bit rate counterparts. Range differences of approximately 10% were observed, i.e. broadcast packets transmitted at 2Mbit/s could be sent approximately 10% further than unicast messages transmitted at 11Mbit/s. We also changed user transmission rate to other standardised 802.11b rates: 1Mbit/s, 2Mbit/s and 5.5Mbit/s, but the broadcast packets were consistently transmitted at 2Mbit/s.

To be sure that this behaviour was not specific to the ELSA cards, we performed some basic tests on other WLAN 802.11b cards. Lucent Orinoco cards exhibited the same behaviour but interestingly, both Netgear and Cisco Aeronet differed in that they transmitted broadcast messages at the same rate as user data. The different behaviour of the network cards can be explained by different interpretations of the IEEE 802.11 specification. This has considerable implications for running AODV over an 802.11b network because it is difficult to predict how the network will perform due to the way the standard is implemented in different network cards.

The described difference in ranges seriously affects AODV performance. From the routing point of view everything looks normal, i.e. all control messages are exchanged properly and as long as node A has data to send to node C that route will not expire. However, since user data is sent in unicast packets, it will suffer from a considerable loss. The two main drawbacks of the existing algorithm are waste of wireless resources (sending data that cannot be received) and delaying of route re-establishment (route A-C will be broken up or established in a different way only when node B gets out of the broadcast range of node A), which eventually deteriorate network performance.

A similar situation can occur when nodes A and C have direct communication link initially and if node C moves away towards node B. Then, when C comes to the ”problem” zone, instead of changing to 2-hops route via B, the 1-hop A-C route will be maintained as long as the routing protocol control messages are exchanged properly, although user data sent from node A never reaches its destination.

4 Improved Neighbourhood Detection Algorithm

There are several solutions to the above problem. Our approach relies on differentiation between so-called ’good’ neighbours and ’bad’ neighbours. Classification is done dynamically whenever a packet is received over a 1-hop route. Neighbours are classified as ’bad’ if the quality of the interconnecting channel is poor. AODV control messages (including `Hello` messages) are accepted only from ’good’ neighbours.

In order to differentiate between “good” and “bad” neighbours, the SNR (Signal to Noise Ratio) is measured each time a packet that contains an AODV control message is received. Based on the SNR value, it is possible to estimate if the wireless channel between two nodes is good enough to carry broadcast and unicast messages with sufficient quality regardless of the transmission rate used. The following rules are applied to accept AODV messages from a neighbour node:

- If an AODV-message is received from a node that has no entry in the routing table, the SNR of the packet must be higher than a certain threshold (Threshold_High), otherwise the AODV message is ignored;
- If an `Hello` message is received from a node, which is a neighbour in the routing table, the SNR of the packet must be higher than *Threshold_Low = Threshold_High - Delta_SNR*, otherwise the AODV message is ignored.

The first rule defines required quality of the link and effectively differentiates between "good" and "bad" neighbours, i.e. between good and bad quality links. If the quality of the link to a potential neighbour (previously unknown to the node) is above the defined threshold, that node becomes a new neighbour.

Due to wireless nature of the link, the SNR can vary even if nodes are not moving. The second rule defines a margin for SNR that allows SNR variations to a certain extent. This increases the stability of the network. This means that once a route is established, it is not lost because of one weak signal observation.

We did several tests in an office environment to determine the optimum threshold values. We found that good performance is obtained when Threshold_High and Threshold_Low are set to 15dB and 5dB respectively.

Our experiments have proved that when this algorithm is implemented, the problem described above does not occur any more and this effectively improves data throughput, decrease delays and improve overall user experience. The proposed algorithm was used to obtain performance evaluation results presented in the following sections.

5 Experimental Evaluation

Results obtained from measurements on a network containing 5 laptops, 4 statically positioned and 1 mobile, in a linear chain are presented here. The distance between the nodes was chosen so that direct connection is possible only between the neighbouring nodes. The SNR between the neighbouring nodes was in the range of 10-25 dB. The laptops used in the experiments were Pentium II based, running Red Hat Linux version 7.2 with kernel 2.4.9 and were equipped with ELSA Airlancer MC-11 WLAN cards. In order to evaluate network performance we measured *r*oute discovery latency and *o*ffered load-received throughput ratio for UDP and TCP traffic.

*R*oute discovery latency is the time to discover a new route and it is a key performance metric for on-demand routing protocols. In networks in which the topology changes rapidly this can have a profound impact on the performance of the network. In more static networks, this can affect the application level performance:

if this is high, the application response time may be slow; if this is very high there may be some data loss due to buffer overflow.

The *o*ffered load-received throughput ratio is the ratio between the load offered to the network (on one end of a single connection) and the throughput received at the other end of the connection. The goal was to analyse throughput of a multi-hop connection in a WLAN 802.11b based ad-hoc network. Studies [6, 7] have shown that the 802.11 MAC layer is not optimal for multi-hop connections and that throughput decreases rapidly as the number of hops increases. Hence, it is interesting to obtain some experimental results to see if this is really the case; these results could then be useful in determining a MAC mechanism that is more suitable for this context.

5.1 Route discovery latency results

The route discovery latency is measured as the time between a transmission of a RREQ message and the reception of the corresponding RREP at the source node. In order to make these measurements, it was necessary to transmit timestamp information in the packets: this involved modifying the format of the RREQ and RREP messages. Thus, the RREQ messages were modified such that the time the message was transmitted was included in the message. When a node that has a route to the destination received the RREQ message, it copied the time stamp from RREQ message to the RREP message and sent it back to the source node. The route discovery latency was then measured as the difference between the time when the RREP was received and the timestamp information included in the RREP message.

As noted previously, the route discovery process operates using an expanding ring search technique. For the purposes of determining the route discovery latency, each time the RREQ was transmitted from the source, the timestamp included in the message was that of the transmission of the first RREQ message.

In our experiments, we measured the route discovery latency for two, three, and four hop routes. We made measurements in scenarios in which there was no traffic load on the network and we made measurements in scenarios in which there was a significant traffic load on the network. To obtain a single route discovery latency measurement, three hundred route establishments were initiated by sending ICMP Echo Request packets to the destination (using ping).

In tests with no user data traffic, only routing messages were exchanged between the nodes. Since this is not such a realistic scenario, we introduced some background traffic. UDP packets were transmitted bi-directionally at a rate of 250 kbit/s between each two neighbouring nodes of the chain, modelling light to moderate traffic between the neighbouring nodes. This traffic does not influence the routing tables; hence the same route discovery messages have to be sent as in the unloaded case.

The route latency measurements obtained in our experiments are shown in Tables 1 and 2.

For each route latency data set, the mean and standard deviation of the route latencies were calculated. We observed that the data set is quite skewed: the vast majority of samples are quite close to the mean but some deviate very significantly. To indicate how skewed the data set is then, two further parameters are shown: the maximum route latency observed and the 90% quantile.

Two different measurements are presented for 3 and 4 hop experiments:

- *R*oute discovery latency for initial route setup: In this test the routing table does not contain an entry for the designated route. This is denoted 'initial setup' in the tables below.

- *R*oute discovery latency when route refreshment is used: In this case, there is an entry in the routing table for the required route but it is marked as expired and cannot be used, i.e. a new route discovery procedure has to be started, but the existing routing information, number of hops in particular, can be used to reduce the time required for route establishment. Hence, the first RREQ sent to re-establish the route can be sent with the TTL equal to the previously known number of hops and can reach the destination in the first attempt. This is denoted by 'refresh' in the tables below.

Table 1. Route Discovery Latency (no user traffic)

	Average latency (ms)	Max latency (ms)	Standard deviation (ms)	90% quantile (ms)
2 hops	4.15	88	6.8	3
3 hops (refresh)	8.21	106	11.23	6
3 hops (initial set up)	26.02	85	3.86	28
4 hops (refresh)	23.38	295	43.41	10
4 hops (initial set up)	36.65	327	33.30	30

The average route latency for 2 hops in the scenario without user traffic in the network is rather small (just over 4ms), but the maximum route latency is very high (Table 1). Such high route latencies are rare events and can be attributed to the loss of control messages and resulting timeouts. For the most part, however, the route latencies are considerably smaller with 90% of the latencies below 3ms. The average route latency for 3 and 4 hops is much bigger, especially for the initial route set up case (26.02ms and 36.65ms for 3 and 4 hops respectively).

Table 2. Route Discovery Latency (with user traffic)

	Average latency (ms)	Max latency (ms)	Standard deviation (ms)	90% quantile (ms)
2 hops	12.61	203	25.99	77
3 hops (refresh)	28.44	240	38.53	102
3 hops (initial set up)	39.33	260	35.10	85
4 hops (refresh)	59.63	996	135.83	136
4 hops (initial set up)	68.69	1048	98.29	187

One cause of these large values is the expanding ring search mechanism of AODV. With that mechanism it is not possible to discover 3 and 4 hop routes with the initial RREQ broadcast and hence a delay equal to the RREP timeout value is introduced. The initial timeout in our implementation was set to 20ms. The recommended value of this parameter is based on a conservative estimate of the average one hop traversal time for packets and should include queuing delays, interrupt processing times and transfer times. We measured this value to be less than 5ms, so we conservatively set the timeout to the above value. Note that a trade-off arises when choosing the timer value. If the chosen value is too small a new RREQ will be sent before the RREP to the previous one is received, creating unnecessary load on the network. On the other hand, if the value is too high, unnecessary delays will be introduced and route discovery latency will be high.

The other causes of long route establishment times are packet collisions and packet losses due to the radio environment. Packet collisions mainly occur at the intermediate nodes due to the concurrent reception of AODV messages from several nodes. Note that such so-called hidden terminal problems can be largely solved using the RTS/CTS mechanisms defined in the IEEE 802.11 standard. However, since it is most likely that in the current generation of WLAN equipment, the majority of such contention problems will occur around an access point, equipment manufacturers have focused on implementing such intelligence for the infrastructure mode only. A minimal amount of this RTS/CTS functionality is implemented for ad-hoc mode thus preventing a satisfactory resolution of the hidden terminal problem in that mode.

When AODV can use prior knowledge (route refresh case) to guess the number of hops to the destination, route discovery latency is significantly smaller (average of 8.21ms and 23.38ms for 3 and 4 hops respectively) because the TTL is initially set to the appropriate value. Then, it is possible to discover the route using just one RREQ and the expanding ring search is not used. When background traffic is introduced in the network, route latency increases as expected. User packets collide with AODV messages and that primarily causes the longer route establishment times. The expanding search ring technique has similar impact as in the test without user traffic.

In [8], the average latency for 2 hop routes without user traffic is 6.63ms and approximately 4ms are added per each additional hop. Our refresh route latency results are consistent with these. The initial route set-up results are much higher due to the constant delay introduced by expanding ring search technique that was not used in [8].

When there is some user traffic in the network, results differ significantly (see table 2). It is not straightforward to make a valid comparison between our results and those published in [8] due to a different amount of user traffic in the network and different available throughput. The ad-hoc environment is a dynamic environment and frequent route breakages can be expected. This could affect applications and have a detrimental impact on their performance. However, the observed results for route discovery latency show that it is possible to (re)-establish a route quickly in the light to moderate traffic conditions. Hence, it should not cause serious difficulties for applications operating in this environment, particularly if the applications are aware that they are operating in an environment that can deliver a very variable performance and is sometimes not available.

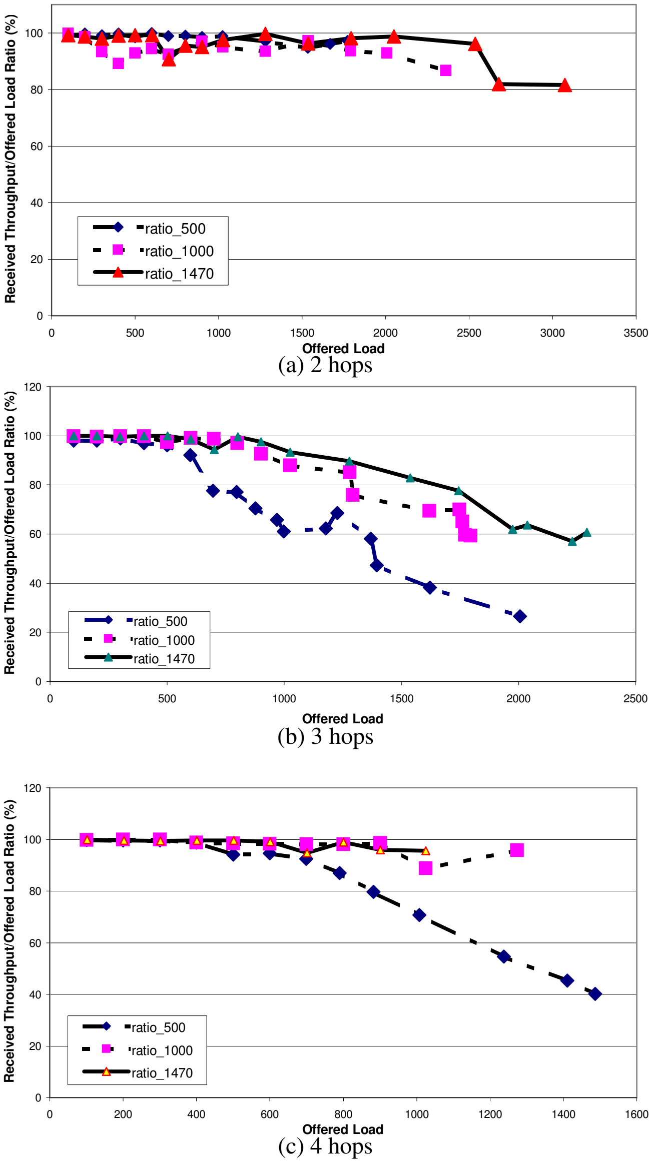
5.2 UDP experiment results

TCP/UDP throughput measurements were done using `iperf` a network performance measurement tool. `iperf` can perform both TCP-based and UDP-based measurements. In the former case, it can measure throughput and in the latter case, it can measure throughput, delays, jitter and packet loss. `iperf` is flexible: different bit rates and packet sizes can be specified for UDP-based measurements and different transmission times or data transfer sizes can be specified for TCP- based measurements.

UDP packets of chosen length were sent at the following data rates: 100, 200, 300, 400, 500, 600, 700, 800, 900 Kbits/s, 1, 1.25, 1.5, 1.75, 2, 2.5 and 3 Mbits/s. Received throughput and error rate were measured.

The measurements were done for three different packet sizes: 500 bytes, 1000 bytes and 1470 bytes and the average of 30 measurements for each data rate were calculated. The same experiment was performed for paths with two, three and four hops.

In Figure 2, the received throughput as a percentage of the average offered load is presented for paths of different lengths. A value of 0% means that there is no throughput and all transmitted traffic is lost in the network and a value of 100% means that all the traffic is successfully transmitted from source to destination.



(a) 2 hops

(b) 3 hops

(c) 4 hops

Fig. 2. Offered load-received throughput ratio for 2, 3 and 4 hop scenarios.

For 2 hops the observed ratio is close to 1, i.e. error rate is very small up to the maximum throughput regardless of the packet size.

More hops introduce more errors as expected. Error free behaviour for 3 hops is observed for offered load of up to 600 kbit/s for 500-byte packet size and up to 1Mbit/s for larger packet sizes. Above these values, the network becomes gradu-

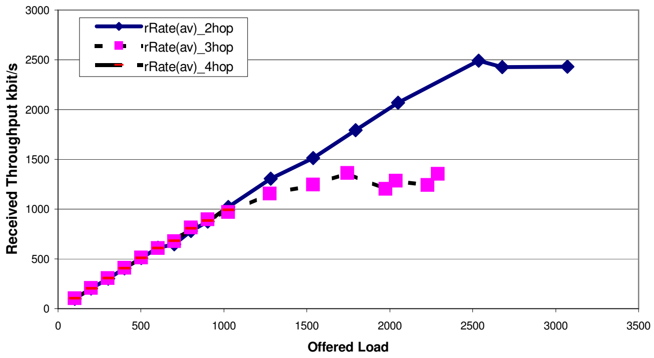


Fig. 3. Received throughput for paths of different lengths with 1470 byte packets.

ally overloaded, more collisions occur and received throughput decreases. Packet collisions, as stated before, mostly occur at intermediate nodes. It is clear from Figure 2 that there is a lower throughput for packets of smaller size. Smaller packets have greater overhead, so it is not so surprising that they result in a lower throughput. Note that in this case the overhead manifests itself both in the network layer overhead, but also, importantly, in the link layer overhead due to the 802.11b MAC mechanism.

In Figure 3 the received throughput for 1470-byte packet size, the largest packets used in experiments, for different number of hops is presented. This is interesting because it enables us to see what the highest possible throughput is with AODV. Thus, for 2 hops, the maximum possible throughput is 2.5Mbit/s, for 3 hops, 1.3Mbit/s and for 4 hops 1Mbit/s.

In [8] similar results are presented for 2Mbit/s WLAN cards. Absolute throughput values were higher in our experiments, but that is due to the higher interface speeds (802.11b network interface cards can operate at up to 11Mbit/s). Observed throughputs in both our experiments and those published in [9, 8] are similar relative to the maximum possible single hop throughput.

5.3 TCP experiment results

We performed similar experiments to obtain performance data for TCP. The experiments were performed using a fixed packet size of 1470 bytes. In each case we attempted to transmit as much data as possible for a fixed duration of 30 seconds and measured the resulting throughput.

There was a very great variation in the results we obtained even greater than that observed in our UDP results. Our results are summarised in the histogram in Figure 4. The results clearly show that as with the UDP case, the throughput depends on the number of hops, i.e. as the length of the route increases, the throughput decreases. This decrease in throughput can be very significant: in routes containing more than two nodes, the throughput can decrease to almost 0 kbits/s.

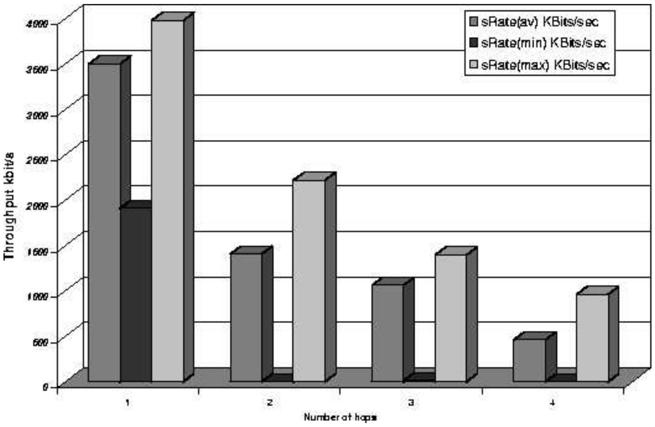


Fig. 4. TCP throughput.

Problems with operating TCP in a wireless environment have been previously reported [10]. These problems have been identified in contexts in which the end terminal uses a wireless link for network access. TCP has largely been designed for scenarios in which there are reliable links and hence it concludes that the network is congested at some point when it infers a packet loss. However, in wireless contexts, the packet loss may not necessarily be due to network congestion, but may be attributed to poor channel conditions. Such problems can be largely solved by adding functionality to the link layer such that the higher layers perceive it as a reliable link connection: ARQ mechanisms are typically used to realise this. Delay issues can also be a problem in this context, but they are largely solved by some small modifications to TCP to add more sophisticated timestamp handling [11].

The environment used in our experiments differed from those assumed in the work mentioned above. Specifically, in this case, there are multiple wireless links that are coupled, i.e. packet transmission on one link can interfere with packet transmission on another resulting in both packets being corrupted.

Using such coupled links resulted in significant performance problems. A considerable amount of user data traffic was lost due to such collisions which clearly resulted in poor network performance. Note also that control traffic could be lost due to collisions: this can also have an impact on the performance of the network as it can result in incorrect routing tables (loss of routes in routing tables, maintenance of routes when they should be lost, etc.).

This link-interference problem is the so-called hidden terminal problem. As previously noted, the solution to this is well-known, but it was not possible to use it in our experiments since the network interface cards did not have the required functionality. Performing such experiments with cards that did support the RTS/CTS mechanism in ad hoc mode would produce considerably different - and better - results.

Many cards in the current generation of WLAN network interface cards do not have support for RTS/CTS in ad hoc mode. To realise an ad hoc network then using these cards, for which performance is an important factor, it is probably necessary to develop some alternative congestion control algorithm that is aware of the fact that there can be significant loss at the link layer that may not necessarily be due to congestion.

6 Conclusions

We performed an experimental performance evaluation of a network that uses the AODV routing protocol operating over an IEEE 802.11b network. We identified an important issue with the WLAN cards that had a detrimental impact on network performance and we proposed a solution to the problem.

We found that the performance of the network can vary significantly due to radio conditions. Hence there was a considerable variation in the obtained results. Nevertheless, we can still make some meaningful observations and conclusions from our results.

The results showed that for light-moderate network loads, AODV typically finds new routes quickly (usually less than 100ms for up to 4 hop routes). This means that the impact on applications should not be so great. There are times, however, when route discovery takes considerably longer and hence applications do need to be somewhat flexible in this environment.

We observed that reasonable UDP throughput could be obtained. Throughputs of 1Mbit/s can be obtained on 5 node linear chain networks. Hence the network could certainly support some tens of users running applications requiring constant bit rates of some tens of kbit/s or perhaps more if their behaviour was of the burst nature.

In our TCP experiments, the performance observed was very variable. This could be attributed to the fact that the packet loss that occurred when transmitting data, largely due to collisions rather than radio conditions, was interpreted as a congestion signal by TCP. Thus, the TCP sender reduced its transmission rate substantially. There are problems using TCP connections in this environment, which can be largely solved by using network cards that support the RTS/CTS mechanism.

We have successfully streamed audio and video across 4 hop routes. That clearly demonstrates that the network can be used to support useful applications. Future work involves using different variants of TCP, integrating mobility into the experiments and performing experiments with more users on the network.

References

[1] C. Perkins, *Ad-Hoc Networking*. Addison-Wesley, 2000.

[2] IETF MANET Working Group, http://www.ietf.org/html.charters/manet-charter.html. [Online]. Available: http://www.ietf.org/html.charters/manet-charter.html

[3] M. Dupcinov, M. Jakob, S. Murphy, and S. Krco, "Experimental performance evaluation of an ad-hoc network that uses aodv routing," in *Proc. of Medhoc '02*, Sept. 2002.

[4] H. Lundgren, E. Nordstrom, and C. Tschudin, "Coping with communication gray zones in ieee 802.11b based ad hoc networks," in *Proceedings of WoWMoM '02*, Sept. 2002.

[5] S. Krco and M. Dupcinov, "Improved neighbor detection algorithm for aodv routing protocol," *IEEE Communication Letters*, Dec. 2003.

[6] P. Gupta, R. Gray, and P. R. Kumar, "An experimental scaling law for ad hoc networks," May 2001, technical Report.

[7] J. Li, C. Blake, D. D. Cuoto, H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *Proceedings of ACM Mobicom 2001*, 2001.

[8] S. Desilva and S. R. Das, "Experimental evaluation of a wireless ad hoc network," in *Proc. of the 9th International Conference on Computer Communications and Networks (IC3N)*, 2000.

[9] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of Mobicom '98*, Oct. 1998.

[10] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving tcp performance over wireless links," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, 1997.

[11] R. Ludwig and R. H. Katz, "The Eifel algorithm: Making TCP robust against spurious retransmission," *ACM Computer Communication Review*, Jan. 2000.