RTSS/CTSS: Mitigation of Exposed Terminals in Static 802.11-Based Mesh Networks

Kimaya Mittal and Elizabeth M. Belding
Department of Computer Science
University of California, Santa Barbara
{kimaya, ebelding}@cs.ucsb.edu

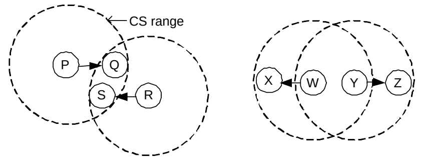
Abstract

Efficient usage of available capacity in wireless mesh networks is critical. Capacity is wasted due to the exposed terminal problem. In this paper, we propose a solution to mitigate the exposed terminal problem in static IEEE 802.11-based mesh networks, thereby improving the spatial reuse of the medium and increasing network throughput. Our solution is complementary to previously-proposed solutions that adjust the carrier-sense range for improved spatial reuse. The proposed solution consists of two phases. In the first phase, exposed links in the mesh topology are detected through an offline training process. Coordination of simultaneous transmissions over exposed links is then done in the second phase through the use of Request-To-Send-Simultaneously (RTSS) *and* Clear-To-Send-Simultaneously (CTSS) *messages, which are added to the MAC protocol. Our solution preserves the distributed nature of the MAC protocol and does not require time synchronization between nodes. We present a simulation-based evaluation that demonstrates that the proposed solution effectively improves capacity usage and network throughput in representative topologies and traffic scenarios.*

1. Introduction

The capacity of wireless mesh networks is severely constrained as compared to wireline networks due to the limited wireless bandwidth, the shared nature of the medium, and the contention among nodes located along multihop paths [6]. Efficient usage of available capacity is therefore critical, which in turn requires efficient spatial reuse of the wireless medium. For better spatial reuse, as many nodes as possible should obtain simultaneous medium access, provided their transmissions do not mutually interfere.

Access to the medium is regulated by the Medium Access Control (MAC) protocol. IEEE 802.11 [7], which is currently the most popular MAC protocol for wireless mesh networks, employs the Carrier Sense Multiple Access (CSMA) strategy for medium access control. CSMA permits a node to transmit if and only if the detected carrier signal is lower than the carrier-sense (CS) threshold; this occurs when no other node within CS range is currently transmitting.



(a) Hidden terminals (b) Exposed terminals

Figure 1. Hidden and exposed terminals.

Two well-known artifacts of the CSMA strategy are the *hidden terminal* and *exposed terminal* problems, both of which result in wasted capacity. The hidden terminal problem occurs when the simultaneous transmissions of two transmitters that lie outside CS range cause interference at one or both receivers and prevent successful reception. An example is shown in Figure 1(a)[1], where nodes P and R are hidden terminals. The exposed terminal problem, on the other hand, occurs when two transmitters lie within CS range and are prevented from transmitting simultaneously, even though their transmissions do not mutually interfere. This is illustrated in Figure 1(b), where nodes W and Y are mutually exposed terminals.

Hidden and exposed terminals waste capacity through failed transmissions and missed transmission opportunities, respectively. Addressing these problems is therefore critical to improve capacity utilization. The prevalence of hidden and exposed terminals in a given network depends on the topology and on the CS range. A large CS range is likely to reduce the number of hidden terminals, but creates more exposed terminals, while a smaller CS range results in fewer exposed terminals but more hidden terminals. There is thus

[1] In Figure 1, the CS range is represented as a circle for simplicity. In reality, it is not a perfect circle; wireless signal propagation is influenced by many factors, including multipath interference, obstacles, and environmental effects.

an inherent trade-off; hidden and exposed terminals cannot both be eliminated by adjusting the CS range alone.

Several solutions have previously been proposed that adjust the CS range to either eliminate hidden terminals [4, 5, 13, 17] or achieve a balance between hidden and exposed terminals [16, 18]. However, none of these solutions eliminates exposed terminals from the network. In this paper, we propose a solution that mitigates the exposed terminal problem. Our solution is complementary to the previously-proposed solutions that adjust the CS range and is intended for static network topologies. It consists of two phases. The first phase is an offline training phase that empirically detects exposed link pairs in a given network.

The second phase of our solution consists of coordination of simultaneous transmissions over exposed links. By definition, the exposed transmitters lie within CS range and are therefore prevented from transmitting simultaneously by the MAC protocol. To enable simultaneous transmissions, we introduce a *Clear-To-Send-Simultaneously (CTSS)* message to the MAC protocol. Consider the example in Figure 1(b), where nodes W and Y are mutually exposed. When node W obtains access to the medium and is about to transmit a packet to node X, it signals node Y to transmit simultaneously to node Z by sending node Y a CTSS message. On receiving the CTSS message, node Y immediately transmits a packet to node Z if one is available, over-riding the carrier-sense mechanism. In this manner, the CTSS message enables simultaneous transmissions over exposed links.

To reduce overhead, it is desirable to invoke this mechanism only when necessary. For this purpose, we introduce the *Request-To-Send-Simultaneously (RTSS)* message. The RTSS message is broadcast by a node when it identifies a need for additional opportunities to transmit to one or more neighbors based on the experienced traffic load. Thereafter, nodes that receive the RTSS message and are exposed to the specified links with respect to their own receivers send a CTSS message to the requester node at every transmission opportunity for some duration of time. RTSS messages are periodically broadcast by the requester node until it no longer requires additional transmission opportunities.

To avoid the overhead of an additional control message preceding every data transmission, we implement the CTSS as a header on the data packet, which is placed between the Physical Layer Convergence Protocol (PLCP) and MAC headers. This significantly reduces the overhead associated with CTSS messages. As a result, our solution provides significant throughput benefits when topology and traffic permits, but negligible overhead otherwise. The drawback of this design is that it requires modification to the Physical (PHY) layer, and therefore cannot be implemented on currently available wireless hardware. We believe that the benefits provided by this solution motivate the inclusion of this mechanism in next-generation wireless devices.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents simulation results indicating the prevalence of hidden and exposed links in a representative topology. Our proposed solution is described in Section 4. Section 5 presents our simulation-based evaluation and Section 6 concludes the paper.

2. Related Work

A common approach to improve spatial reuse in wireless networks is to tune the CS threshold [4, 5, 13, 16, 17, 18]. Most of the proposed solutions rely on overly simplistic assumptions relating inter-node distance to signal strength. Further, the resulting CS ranges are asymmetric in some cases, which can lead to unfairness. Another approach is to control the transmission power for better spatial reuse [3, 10]. However, hidden and exposed terminals cannot both be eliminated by either adjusting the CS threshold or controlling the transmission power. Note that our solution is complementary to these approaches.

The RTS/CTS mechanism, which is part of the IEEE 802.11 standard [7], attempts to address the hidden terminal problem, but has shortcomings when applied to multihop networks [15]. Further, it does not address the exposed terminal problem. MACA-P [2] enhances the RTS/CTS mechanism to increase concurrency. The RTS/CTS exchange between a pair of nodes is followed by a control gap during which another pair of nodes can also exchange RTS/CTS messages and synchronize its data transmission with the first node pair. Although this solution is conceptually similar to the one proposed in this paper, it suffers from all the weaknesses of the RTS/CTS mechanism [15]. Further, the RTS/CTS messages and control gap significantly increase the overhead per data transmission. Our solution, on the other hand, has negligible overhead.

Other researchers have proposed TDMA-style MAC protocols to maximize spatial reuse in static multihop wireless networks [12]. Unlike this approach, our solution preserves the distributed nature of the MAC protocol and does not require time synchronization between nodes.

3. Prevalence of Hidden and Exposed Terminals

The prevalence of hidden and exposed terminals in a given network depends on the network topology, the CS range, and the *capture* capability, i.e. the capability to successfully receive the stronger transmission in a collision [9], of the wireless hardware. The capture capability depends on the data rate; the lower the data rate, the lower the minimum Signal-to-Interference-and-Noise-Ratio (SINR) required for capture [8].

In this section, we use simulation to examine the number of hidden and exposed terminals that occur in a representa-

CS threshold (dBm)	-99	-97	-95	-93	-91	-89
CS range (m)	712	634	565	504	449	400

Table 1. CS threshold to CS range mapping.

tive topology at various CS ranges and data rates. The objectives of this study are (1) to get a sense of the prevalence of these effects in a representative topology; (2) to demonstrate how these effects are traded at different CS ranges and data rates; and (3) to motivate that there is significant scope to improve spatial reuse, and thereby increase throughput, in typical mesh networks through mitigation of the exposed terminal problem.

The QualNet simulator version 3.9 [1] is used for our simulations. We extend the simulator to implement the capture effect [9]. Our representative topology consists of 25 nodes arranged in a 5x5 grid with a distance of 150m between adjacent nodes. We use the IEEE 802.11b protocol with the data rate fixed at 2 Mbps and 11 Mbps in different tests. Retransmissions and the RTS/CTS handshake are disabled. The transmit power is set to the default value of 15 dBm, which results in a transmission range of 283m and 370m for the 11 Mbps and 2 Mbps data rates, respectively. Note that transmissions can still be received beyond the transmission range; however, the probability of successful reception decreases with increasing distance. We vary the CS threshold from -99 dBm to -89 dBm in different tests. The resulting CS ranges are indicated in Table 1. The default CS threshold value in QualNet is -93 dBm. All other simulation parameters are set to their default values.

The procedure for determining hidden and exposed terminals is as follows. For each data rate, we first identify all the links that have a delivery probability of 95% or higher in the absence of interference. These are the *strong* links that are likely to be used by a routing protocol. (The threshold value of 95% is picked arbitrarily; a different value results in different absolute numbers but similar trends.) Next, the strong links are considered pairwise. Each pair transmits a set of packets simultaneously. Note that, in order to correctly detect exposed terminals, *it is critical that the carrier-sense function be disabled* during the simultaneous transmissions; otherwise, carrier-sense prevents exposed nodes from transmitting simultaneously.

The delivery probabilities when transmitting simultaneously without carrier-sense are used to determine whether the links are mutually hidden or exposed. If each link achieves at least 95% of its original delivery probability and the two transmitters lie within CS range, the links are mutually exposed. On the other hand, if the delivery probability of either link falls below 5% and the two transmitters do not lie within CS range, the links are hidden. Note that, in the simulator, the distance between the two transmitters can be used to determine whether they lie within CS range. The link pairs that achieve between 5% and 95% of their original delivery probabilities are not considered as either exposed or hidden by our conservative definition.

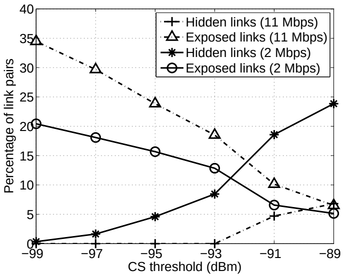


Figure 2. Prevalence of hidden and exposed links at different CS thresholds.

Figure 2 shows the results of our simulations. In the figure, we plot the percentage of link pairs that are mutually hidden or exposed in the representative topology at each data rate for different CS thresholds. At the lower data rate, the larger transmission range results in a greater number of strong links, which in turn leads to a significantly greater number of link pairs tested. Specifically, 8,688 link pairs were tested at 11 Mbps, while 37,476 link pairs were tested at 2 Mbps. Since the transmitter and receiver are likely to be separated by a greater distance at the lower data rate, the received signal strength is likely to be lower, resulting in a lower tolerance for interference. Therefore, the percentage of hidden links is higher and the percentage of exposed links is lower at 2 Mbps for all CS thresholds. Note that, if a fixed set of links is considered at both data rates, more exposed terminals and fewer hidden terminals occur at the lower data rate since the minimum SINR required for capture is lower [8].

Figure 2 confirms our expectation that as the CS threshold increases, the number of hidden link pairs increases while the number of exposed link pairs decreases. We observe that, at most values of the CS threshold, there exists a significant number of exposed link pairs in the network. This indicates that there is significant scope to improve capacity usage and increase throughput in the network by mitigating the exposed terminal problem.

It would be interesting to study the prevalence of hidden and exposed terminals in a wireless mesh testbed. However, such a study is extremely challenging due to the unavailability of a straightforward way to over-ride the carrier-sense function on currently available 802.11 hardware[2].

[2] Other researchers have achieved the effect of disabling carrier-sense by disabling random backoff [8]. We were unable to duplicate this effect on our testbed due to hardware differences.

4. Mitigation of Exposed Terminals

In this section, we present our solution for the mitigation of exposed terminals in static mesh topologies. Our solution consists of two phases. The first phase, described in Section 4.1, empirically detects exposed link pairs in a given network topology. Coordination of simultaneous transmissions over exposed links is then done in the second phase, as described in Section 4.2.

4.1. Detection of Exposed Terminals

Detection of exposed links in a given network topology is a non-trivial task. Approaches that rely on assumptions related to inter-node distance and signal strength [4, 5, 13, 17] are overly simplistic and ignore several important factors, such as the specific characteristics of the environment and the wireless hardware. Accurate modeling of all relevant factors is complex and challenging.

An alternative approach that automatically takes all relevant factors into consideration is empirical testing. We select this approach for our solution. Exhaustive testing of all pairs of links in the network, however, is inefficient and time-consuming (*O*(n^4) where n is the number of nodes in the network). This has previously been pointed out by Padhye et al., who propose an efficient estimation technique that uses broadcast transmissions to predict pairwise unicast link interference [11]. In their proposed technique, nodes in the network broadcast a set of packets, first individually and then pairwise. The received throughput in each case is recorded by all nodes. The interference ratio for a link pair (A, B), (C, D) is then calculated as the ratio of the aggregate throughput of the links when nodes A and C transmit simultaneously to the aggregate throughput when nodes A and C transmit individually. The value of the interference ratio lies between zero and one. If the links do not mutually interfere, the interference ratio is close to one. Further, a comparison of the aggregate *send rate* of nodes A and C when transmitting individually and simultaneously indicates whether they are within CS range of each other; if the nodes are within CS range, their aggregate send rate is approximately halved during simultaneous transmission.

We extend this interference estimation technique for our solution with a few significant modifications. First, to detect exposed terminals, simultaneous transmissions must be repeated with the carrier-sense function *disabled* for all pairs of transmitters that lie within CS range. Then, if the interference ratio of a link pair with carrier-sense disabled is close to one, the links are mutually exposed.

Second, since broadcast transmissions are not acknowledged, the interference estimation technique does not account for collisions involving ACK frames. Padhye et al. argue that ACK frames are small (14 bytes) and therefore unlikely to collide. This may be true in the case of randomized medium access by the interfering links. However, in our proposed solution, exposed links closely synchronize their transmissions using CTSS messages, as described in Section 4.2. Collisions involving ACK frames are thus guaranteed to occur and must be accounted for by the exposed link detection procedure. In other words, a link pair may be identified as mutually exposed only if both the data frame and ACK frame can be captured successfully by each link. Let $BIR_{(A,B),(C,D)}$ denote the broadcast interference ratio of links (A, B), (C, D) with carrier-sense disabled. The links are identified as mutually exposed if and only if $BIR_{(A,B),(C,D)}$, $BIR_{(A,B),(D,C)}$, $BIR_{(B,A),(C,D)}$, and $BIR_{(B,A),(D,C)}$ are each greater than a threshold. This ensures a high probability of successful capture in data-data, data-ACK, ACK-data, and ACK-ACK collisions. After the detection of exposed terminals is completed, the information is distributed to all mesh nodes.

Since the capture capability of wireless hardware is different at different data rates [8], if multiple data rates are used in the network, the detection of exposed links must be repeated for each data rate. This implies that the broadcast transmissions used for the BIR calculation must use the specified data rate, and not resort to the lowest data rate as is done by some wireless hardware. Note that testing all combinations of data rates between a pair of links is unnecessary. The ability of link (A, B) to capture a transmission in the presence of interference from link (C, D) depends only on the data rate of link (A, B) and the transmit power of node C; the data rate of link (C, D) is immaterial.

Although our procedure for detecting exposed links is more efficient than the exhaustive testing of all link pairs (*O*(n^2) as opposed to *O*(n^4)), it is still significantly time-consuming. Further, no other traffic can be permitted on the mesh network while the testing is in progress. Due to these reasons, it may not be possible or desirable to execute the exposed terminal detection phase frequently. Since the mesh topology is static, interference patterns are likely to remain fairly constant (interference patterns in a mesh testbed have been found to not vary significantly over a period of several days [11]). However, this may not be the case in all mesh deployments. To handle variations in interference patterns, it is necessary to extend our solution to dynamically learn and adapt to the current set of exposed links in the network, preferably without requiring the halt of ongoing traffic. This is interesting future work.

4.2. The RTSS/CTSS Mechanism

Coordination of simultaneous transmissions over exposed links is accomplished in the second phase of our solution. For this purpose, two new control messages, the *Request-To-Send-Simultaneously (RTSS)* and the *Clear-To-Send-Simultaneously (CTSS)*, are introduced to the MAC protocol.

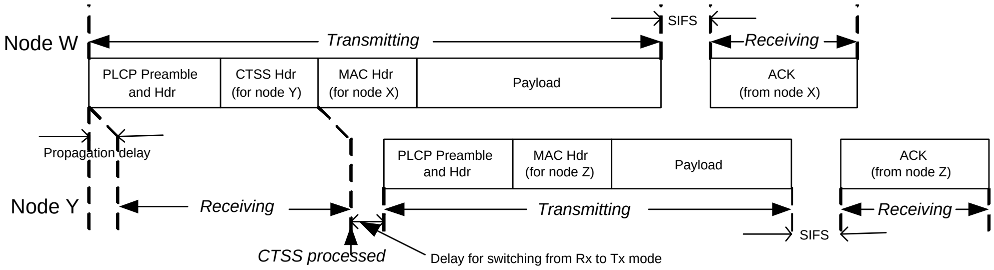


Figure 3. Coordination of simultaneous transmissions.

The CTSS message: We first describe the use of the CTSS message. Consider the example network from Figure 1(b), where links (W, X) and (Y, Z) are mutually exposed. When node W obtains access to the medium and is about to transmit a packet to node X, it signals node Y to simultaneously transmit to node Z by sending node Y a CTSS message. On receiving the CTSS, node Y immediately transmits a packet to node Z by over-riding the carrier-sense mechanism, provided a packet is available and certain criteria described further are met. In this manner, the two exposed links are able to transmit simultaneously.

The CTSS message must identify the link for which it is intended. Links may be identified by their source and destination addresses (6 bytes each). Alternatively, identifiers may be assigned to exposed links in the first phase of the solution and distributed to the mesh nodes along with the exposed link pair information. In this case, a 2-byte identifier is sufficient for a network consisting of up to 256 nodes. The CTSS message also contains a 2-byte Frame Control field, which is common to all 802.11 messages and identifies the type of message, and a 2-byte CRC for detection of transmission errors. The total CTSS size is thus 6 bytes.

Even though the size of the CTSS message is small, preceding every data packet with a separate CTSS packet generates significant overhead (note that every 802.11 packet carries the PLCP preamble and header, which are transmitted at the lowest data rate and consume 192 microseconds of medium time in 802.11b). To reduce this overhead, we implement the CTSS message as an additional header on the data packet. This header is placed between the PLCP and MAC headers. This implementation results in negligible overhead generated by the use of CTSS messages. Note that, for the remainder of the paper, the terms CTSS, CTSS message, and CTSS header are used interchangeably.

The CTSS header contains its own CRC and can therefore be processed independently as soon as it is received, without waiting for the following MAC header and payload. In our example, when node Y processes the CTSS from node W and decides to transmit simultaneously, it immediately switches its wireless interface to transmit mode and sends a packet to node Z. Figure 3 illustrates this process in detail. Note that the time to switch from the receive to the transmit mode is of the order of 10 microseconds for DSSS-based radios [14] and therefore negligible compared to the transmission duration of the packet.

Depending on the size of the data packet transmitted by each node, the data transmissions by nodes W and Y may overlap in time with the ACK transmissions by nodes Z and X, respectively. The two ACK transmissions may also overlap with each other. Therefore, when identifying exposed links in phase one, it is important to ensure successful capture of both data and ACK frames in data-data, data-ACK, ACK-data, and ACK-ACK collisions between the two links.

Causes of lost/unused CTSS messages: Some CTSS messages may be lost due to collisions, others due to transmission errors. Further, from among the CTSS messages that are received correctly, not all can be followed by a simultaneous data transmission. It is possible that when a node receives a CTSS message for a particular link, it has no data packet to transmit on that link at that instant.

In the example from Figure 1(b), when node W sends a CTSS to node Y, it is possible that there is another transmission in progress by a node P within node Y's CS range, which node W cannot sense. Node Y may still be able to successfully capture the CTSS. However, if it follows with a transmission to node Z, node Z may not be able to capture the transmission in the presence of the cumulative interference from nodes W and P. Moreover, the cumulative interference may also corrupt the ongoing transmissions of nodes W and P. To avoid this situation, a node is permitted to transmit immediately after receiving a CTSS only if the sensed interference just prior to receiving the CTSS was lower than a specified threshold.

To summarize, CTSS messages may be lost or wasted due to collisions, transmission errors, unavailability of packets to transmit, and sensed interference exceeding a threshold. However, due to our implementation of the CTSS as a header on the data packet, the overhead of sending CTSS messages is negligible, and so even if a fraction of the CTSS messages successfully result in a simultaneous transmission, an overall improvement in throughput is obtained. We validate this expectation in Section 5.

The RTSS message: A link may be exposed to multiple other links in the network. Among these, some links may be heavily loaded, while others have no ongoing data traffic. To aid in selection of an appropriate link as the CTSS destination in such situations, we introduce the RTSS message. The RTSS message is broadcast by a node when it identifies a need for additional opportunities to transmit to one or more neighbors based on the observed traffic. A good metric for this is the size of the interface queue at a node; a queue that is almost full indicates backed-up traffic and the need for more transmission opportunities. The contents of the RTSS message include the Frame Control field (2 bytes), the destination address (Broadcast, 6 bytes), a field indicating the number of links for which the node requires additional transmission opportunities (2 bytes) and a list of the corresponding link identifiers (2 bytes each).

A node may send CTSS messages to a neighbor only if it has previously received an RTSS from that neighbor and is exposed to one or more of the specified links with respect to its own receiver. The received RTSS remains valid for a certain duration of time, beyond which CTSS messages to the neighbor are ceased unless another RTSS is received. A node periodically broadcasts RTSS messages until its interface queue is no longer backed-up.

The RTSS message enables CTSS messages to be sent only when necessary, avoiding the additional overhead otherwise. It also helps nodes to select the appropriate CTSS destination when a link is exposed to multiple other links.

CTSS destination selection: A node may receive RTSS messages requesting additional transmission opportunities for multiple exposed links. All such links are candidates for a CTSS message, from among which the node must select a CTSS destination when it obtains access to the medium. The policy for selection of the CTSS destination is a key component of our solution, which can significantly affect the fraction of CTSS messages successfully used and thereby influence the throughput gain obtained.

With the goal of reducing the number of CTSS messages wasted due to collisions, transmission errors, and sensed interference, we use the following policy to select a CTSS destination. When a node receives an RTSS message from a neighbor, it records the received signal strength of the message. The candidate whose received signal strength is the highest is then selected as the CTSS destination. This candidate is likely to be nearest to the node, thereby reducing the probability of the CTSS being lost due to a collision or transmission error. Further, the nearer the CTSS destination, the lower the likelihood that it senses interference that the CTSS sender cannot sense; this reduces the probability of the CTSS being wasted due to sensed interference.

Our policy is effective and simple to implement. We evaluate the effectiveness and analyze the drawbacks of this policy in Section 5.4. More sophisticated criteria may be added to this policy to further improve the performance of this solution. For example, preference may be given to links that lie closer to the mesh gateway, since these links are likely to be the bottleneck for the mesh traffic. Additional information may be included in RTSS messages, such as the size of the interface queue or the traffic priority, to aid in CTSS destination selection.

RTSS/CTSS data rate: RTSS and CTSS messages must be transmitted at the same data rate so that they reach the same range and can be successfully exchanged between a pair of nodes. There is a tradeoff involved in selecting the data rate at which the messages should be transmitted. At a lower data rate, the messages can be received at a greater distance and can therefore reach more exposed links. However, the transmission duration of the messages increases, thereby increasing the overhead relative to the data transmissions. We evaluate this tradeoff in Section 5.5.

If the CTSS header is sent at a data rate different from the one used for the MAC header and payload, the PHY layer must be informed of this data rate in order that it correctly interprets the incoming signal. One option is to fix the data rate for the RTSS/CTSS messages to a single value. Alternatively, the PLCP header, which currently contains information about the payload data rate, may be modified to include information about the CTSS data rate as well.

Effect of auto rate selection: Mesh nodes are commonly configured to automatically vary the data rate of a link based on the observed link characteristics. Since the capture capability is different at different data rates [8], a node W may send a CTSS to a node Y only if it is exposed to node Y at its current data rate (recall that exposed terminals at each data rate are detected in phase one). On receiving the CTSS, node Y determines the highest data rate at which it can successfully transmit in parallel with node W. It then uses the minimum of this data rate and its current link data rate for the simultaneous data transmission.

5. Evaluation

We evaluate our proposed solution using the QualNet simulator version 3.9 [1]. The objectives of our evaluation are (1) to demonstrate the effectiveness of the solution, gain insight into its operation, and identify the tradeoffs involved; (2) to analyze the effectiveness and drawbacks of the CTSS destination selection policy; and (3) to examine the effect of data rate on the performance of the solution. Note that a testbed evaluation of the solution is currently infeasible due to the changes required at the PHY layer, which are not supported by currently available 802.11 hardware.

5.1. Simulation Environment

We implement our solution by extending the IEEE 802.11 implementation in QualNet. In our experiments, the CS threshold is set to the default value of -93 dBm. The network topologies used are described in Sections 5.3 and 5.4. Since our solution is intended for static topologies, no mobility is simulated. Static routes are pre-configured; no

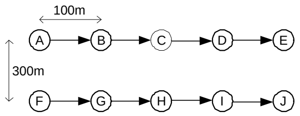


Figure 4. Parallel lines topology.

routing protocol is used. All other simulation parameters are configured as described in Section 3.

We use CBR traffic for our experiments. The number of CBR flows is varied in different tests. All flows begin simultaneously and are 10 seconds in duration. Each CBR source generates 512-byte packets at the rate of 1000 packets per second. This traffic configuration results in a heavily-loaded network, which is the target scenario for our solution.

Parameters specific to our solution are set as follows. A node broadcasts an RTSS message when its interface queue is over 10% utilized. RTSS messages are rebroadcast every second. The RTSS timeout parameter is set to 20 seconds, which exceeds the simulation duration. The threshold value for the sensed interference is set to -86 dBm. These parameter values were empirically found to be preferable for our simulated scenarios; the results of the corresponding experiments are omitted due to space constraints.

The preferable parameter values may vary for other topologies and traffic patterns. For example, if the traffic pattern is highly dynamic, it may be beneficial to use a higher value for the queue threshold that triggers RTSS messages and a lower RTSS timeout period in order to reduce wastage of CTSS messages due to data unavailability. Also, if the links in the topology are weaker, a lower threshold value for the sensed interference may be used to increase the probability of successful parallel transmissions.

5.2. Performance Metrics

The following performance metrics are used to evaluate our solution and gain insight into its performance:

Throughput improvement: This metric measures the percentage of aggregate throughput improvement obtained by our solution as compared to regular 802.11, and indicates how well our solution meets its primary objective. We measure both the **hop-by-hop** throughput improvement, which considers all transmissions successfully received by network nodes in the throughput calculation, and the **end-to-end** throughput improvement, which only considers the packets received by the CBR destinations. While hop-by-hop improvement indicates the raw benefit of our solution, its relationship with end-to-end improvement depends on the topology and the CTSS destination selection policy; a higher end-to-end improvement is obtained if throughput is improved at bottleneck links on paths. Our goal is to maximize the value of both metrics.

Percentage of data packets that carry a CTSS header: This metric indicates how frequently our solution comes into play in different topologies and traffic scenarios.

Percentage of CTSS messages received, used, and wasted: We measure the fraction of CTSS messages sent that are successfully received. The remaining messages are lost due to collisions. From among the received CTSS messages, some are used, i.e. followed by a simultaneous transmission, while the remaining are wasted due to either data unavailability, transmission errors, or sensed interference that exceeds the configured threshold. We measure each of these fractions in order to better understand the behavior of the solution and identify weaknesses that can be improved.

5.3. Simple Topologies

We first evaluate our solution in two simple network topologies to establish a baseline for our evaluation. The first topology is that represented in Figure 1(b). This topology helps us quantify the performance benefits obtained in the absence of interference and multiple hops. The distance between nodes W and X and nodes Y and Z is set to 100m, while the distance between nodes W and Y is set to 300m. CBR flows are set up from node W to node X, and from node Y to node Z.

Our second topology consists of 10 nodes placed in two parallel lines, as indicated in Figure 4. Adjacent nodes in each line are separated by a distance of 100m, while the distance between the two lines is 300m. In this topology, each link in a given line is exposed to all links in the other line. The topology thus offers significant scope for our solution to increase the network throughput. CBR flows are set up from node A to node E, and from node F to node J. In both topologies, the data rate on each link is set to 11 Mbps and the CTSS header is transmitted at 2 Mbps.

Table 2 shows the results, which are averaged over 10 runs with different seeds. As seen in the table, our solution results in a 59.7% throughput improvement in the two-links scenario. Ideally, if the transmissions of the two links were perfectly synchronized and every single-link transmission could be replaced by simultaneous transmissions, one might expect a 100% throughput improvement. However, in reality, this is not the case due to the following reasons.

First, the simultaneous transmissions are not perfectly overlapped; as shown in Figure 3, the second transmission starts later than the first, and so the total duration of the simultaneous transmissions is significantly larger than that of a single transmission. Second, following a successful simultaneous transmission, both nodes reinitialize their random backoff counters. While this is essential for fair medium access, it results in non-optimal medium usage when only two transmitters are present. Third, since the RTSS/CTSS mechanism does not come into play until the interface queue size

Metric (%)	Two-links topology	Parallel lines topology
End-to-end improvement	59.7	50.8
Hop-by-hop improvement	59.7	47.4
Data packets carrying CTSS	98.9	97.8
CTSS received	96.0	88.0
CTSS wasted (data unavailable)	0.0	3.9
CTSS wasted (transmission error)	0.0	0.1
CTSS wasted (sensed interference)	0.0	6.7
CTSS used	96.0	77.2

Table 2. Results from simple topologies.

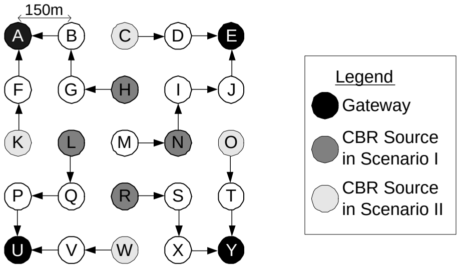


Figure 5. Grid topology.

exceeds the 10% threshold, the early data transmissions do not carry CTSS headers and therefore do not invite simultaneous transmissions. As seen in Table 2, 98.9% of data packets carry CTSS headers in this scenario. Finally, from among the CTSS messages sent, only 96% are received correctly in this scenario; the remaining are lost in collisions that occur when the backoff counters of the two nodes expire simultaneously.

The table shows that all received CTSS messages are used for simultaneous transmissions in the two-links scenario. No CTSS messages are wasted due to data unavailability (since both transmitters are sources of CBR flows that saturate the links), transmission errors (since the transmitters are well within transmission range at 2 Mbps), or sensed interference (since there are no other transmitters).

Table 2 also shows the results from the parallel lines topology. In this topology, the end-to-end and hop-by-hop throughput improvement obtained is 50.8% and 47.4%, respectively, which is less than the two-links topology, but still significant. Since this topology has more transmitters, several CTSS messages are lost due to collisions and only 88% are correctly received. From those received, 3.9% are wasted due to data unavailability; since CBR sources are located only on nodes A and F in this topology, data availability at all other transmitters depends on packets received from the previous link on the path. An additional 6.7% of CTSS messages are lost due to sensed interference that exceeds the configured threshold; this occurs when another nearby transmitter is accessing the medium. Simultaneous transmissions result from 77.2% of the CTSS messages sent. In this scenario, the end-to-end and hop-by-hop throughput improvements are of similar magnitude since all links on the multihop paths are equally likely to avail of simultaneous transmission opportunities.

5.4. Impact of CTSS Destination Selection Policy

We now examine the impact of our CTSS destination selection policy, which is based on received signal strength (RSS). For our experiments in this and subsequent sections, we use the grid topology from Section 3. As illustrated in Figure 5, the nodes at the corners of the grid (i.e. nodes A, E, U, and Y) are selected as the mesh gateways. Static routes are configured from each node to a mesh gateway, such that the number of hops is minimized and nodes are evenly distributed among gateways. Only those links that lie along the vertical and horizontal grid axes are used.

For this evaluation, we use two sample traffic scenarios. CBR sources are configured on nodes H, L, N, and R in scenario I, and on nodes C, K, O, and W in scenario II, as indicated in Figure 5. The destination for each flow is the corresponding gateway.

To demonstrate the effectiveness of our RSS-based CTSS destination selection policy, we compare its performance with a policy that randomly selects a CTSS destination from among valid candidates. Figure 6 shows the results of our simulations, averaged over 10 runs. In the figure, we plot the percentage of CTSS messages that are received, used, and wasted for each traffic scenario with both policies. Recall that the goal of the RSS-based policy is to reduce wastage of CTSS messages.

As seen in Figure 6, 78% and 62% of CTSS messages are received with the RSS-based and random policy, respectively, in Scenario I. The RSS-based policy reduces the loss of CTSS messages due to collisions since the greater RSS of the CTSS messages results in a higher capture probability. In scenario II, 83% and 76% CTSS messages are received with the RSS-based and random policy, respectively. Loss due to collisions is lower in this scenario since the CBR flows have a greater spatial separation, resulting in greater SINR values and a higher capture probability.

Figure 6 also indicates the composition of the received CTSS messages. In scenario I, fewer CTSS messages are wasted due to data unavailability with the RSS-based policy than the random policy. In scenario II, on the other hand, the reverse effect is observed. The RSS-based policy does not consider the traffic availability at links and always selects the nearest candidate link. In scenario I, this policy tends to select heavily-loaded first-hop links, while in scenario II, the second-hop links get selected more frequently. The loss of CTSS messages due to data unavailability could be further reduced by considering data availability as a metric in the CTSS destination selection policy.

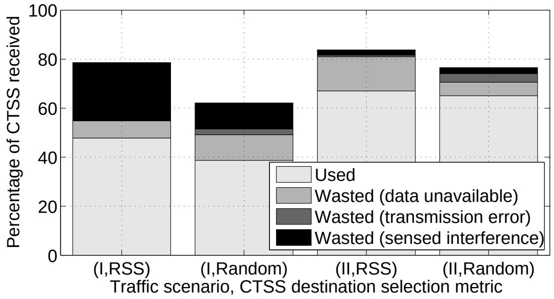


Figure 6. Fraction of CTSS received, used, and wasted in sample traffic scenarios with RSS-based and random policies.

	Scenario I		Scenario II	
	RSS	Random	RSS	Random
End-to-end impr (%)	36.8	27.0	8.0	6.0
Hop-by-hop impr (%)	39.6	27.5	16.7	18.0

Table 3. Throughput improvement in sample traffic scenarios with RSS-based and random policies.

Figure 6 shows that fewer CTSS messages are lost due to transmission errors with the RSS-based policy in both scenarios due to the higher RSS of the CTSS messages. The fraction of CTSS messages wasted due to sensed interference is significantly higher with the RSS-based policy in scenario I. With the random policy, most of these messages are not even captured due to lower SINR; correspondingly, the fraction lost due to collisions is high. In scenario II, sensed interference is generally weaker due to greater spatial separation, and so the fraction of CTSS messages lost due to sensed interference is low with both policies.

The throughput improvement obtained with the two policies in each scenario is shown in Table 3. Fewer wasted CTSS messages result in a significantly higher improvement (39.6%) with the RSS-based policy in scenario I. In scenario II, the fraction of CTSS messages used, and therefore the throughput improvement, is approximately the same with both policies. The significant difference between hop-by-hop and end-to-end throughput improvement in scenario II is due to unequal transmission opportunities for links along multihop paths. Thus, the magnitude of improvement and the relationship between end-to-end and hop-by-hop improvement are topology/traffic dependent.

5.5. Impact of CTSS Data Rate

As explained in Section 4.2, the CTSS data rate presents a tradeoff; the lower the data rate, the greater the number of exposed links that can be reached, but the higher the overhead. In this section, we validate this tradeoff through simulation and identify the optimal CTSS data rate for our representative topology and traffic scenarios.

The grid topology from Figure 5 is used. The number of CBR flows is varied from 2 to 8. CBR sources are selected randomly with the constraint that they are evenly distributed among the gateways. Results are averaged over 20 random traffic scenarios. The data rate is fixed at 11 Mbps, while the CTSS data rate is varied in different tests.

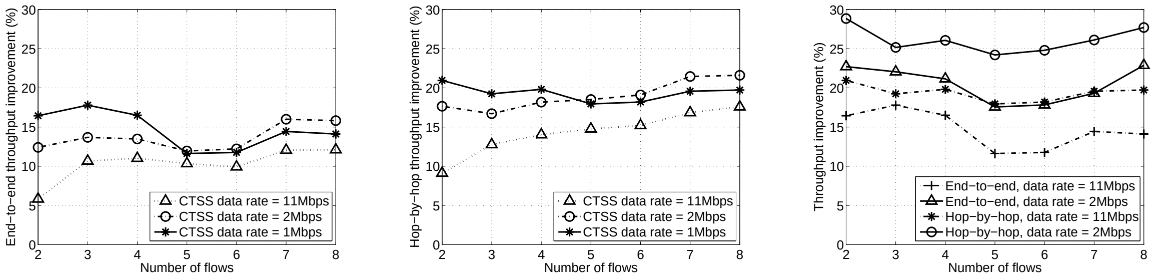
Figures 7(a) and 7(b) show the average end-to-end and hop-by-hop throughput improvements obtained with each CTSS data rate. When CTSS messages are sent at 11 Mbps, they reach a smaller range. Further, a significant fraction of messages is lost due to collisions and transmission errors. Therefore, the improvement is low at this CTSS data rate.

The throughput improvement increases when CTSS messages are transmitted at 2 or 1 Mbps. When there are fewer than 5 flows in the network, active links are generally separated by larger distances, and so sending the CTSS at 1 Mbps is beneficial since it can reach a larger range. On the other hand, when the number of flows is greater than 5, active links have less spatial separation, and so the 2 Mbps transmission range is sufficient to exploit all simultaneous transmission opportunities. Using the 1 Mbps data rate in this situation provides no additional benefit, but rather increases overhead due to longer transmission times. Hence, with less than 5 flows, a CTSS data rate of 1 Mbps is preferable, whereas with greater than 5 flows, a CTSS data rate of 2 Mbps produces the maximum benefit.

We observe in Figures 7(a) and 7(b) that the magnitude of improvement does not change significantly with an increase in the number of flows. When there are fewer flows, more CTSS messages are lost due to transmission errors, but less due to sensed interference and collisions. The effects reverse as the number of flows increases, and the overall improvement remains fairly unchanged. A comparison of the figures shows that the average end-to-end improvement is somewhat lower than the hop-by-hop improvement, since not all links along the multihop paths can avail of an equal number of simultaneous transmission opportunities.

5.6. Impact of Data Rate

In this section, we examine the impact of the data rate used for the data transmissions. The same topology and traffic pattern is used as in Section 5.5. The CTSS data rate is fixed at 1 Mbps. We vary the data rate used for the data transmissions in different tests. Figure 7(c) shows the results. As seen in the figure, the throughput improvement at 2 Mbps is significantly higher than at 11 Mbps. From among the fixed set of active links, more exposed link pairs exist at 2 Mbps, resulting in more simultaneous transmission opportunities that our solution can exploit.



(a) End-to-end improvement with different CTSS data rates.

(b) Hop-by-hop improvement with different CTSS data rates.

(c) Throughput improvement with different data rates.

Figure 7. Results for grid topology.

Summary: Our evaluation shows that our solution effectively improves the aggregate throughput in representative topologies and traffic scenarios. The magnitude of improvement varies in different scenarios from 12% to 60%. We analyze the impact of the CTSS destination selection policy and find that it effectively reduces the wastage of CTSS messages. Our examination of the CTSS data rate highlights the tradeoff between the range and reception probability of CTSS messages, and the overhead due to consumption of medium time. Finally, the lower the data rate of the links, the greater the benefit obtained through our solution.

6. Conclusion

In this paper, we proposed a solution that improves spatial reuse and increases throughput in static wireless mesh networks through mitigation of the exposed terminal problem. Exposed links in the network are detected through an offline training phase. Simultaneous transmissions over exposed links are then coordinated using RTSS and CTSS messages. Simulation results demonstrated the benefits of the proposed solution and highlighted various tradeoffs.

Our solution maintains the distributed contention-based nature of the MAC protocol and does not require complex time synchronization among nodes. The overhead of the solution is negligible. The drawback is the modification required to the PHY layer, which increases complexity and prevents implementation on currently available hardware. We believe that the benefits of the solution motivate its inclusion in next generation wireless devices.

The solution can be further improved along various directions. The CTSS destination selection policy can be modified to consider other criteria, such as traffic availability, in addition to received signal strength in order to further increase effectiveness. Variable interference patterns can be handled by designing a mechanism to dynamically learn and adapt to the current set of exposed links in the network.

References

[1] Scalable Network Technologies. The QualNet Network Simulator. http://www.scalable-networks.com/.

[2] A. Acharya, A. Misra, and S. Bansal. Design and Analysis of a Cooperative Medium Access Scheme for Wireless Mesh Networks. In *Broadnets Wireless Networking Symposium*, San Jose, CA, Oct 2004.

[3] M. Cesana, D. Maniezzo, P. Bergamo, and M. Gerla. Interference Aware (IA) MAC: an Enhancement to IEEE 802.11b DCF. In *Vehicular Technology Conference (VTC)*, Orlando, FL, Oct 2003.

[4] I. Chakeres and E. Belding-Royer. PAC: Perceptive Admission Control for Mobile Wireless Networks. In *QShine*, Dallas, TX, Oct 2004.

[5] J. Fuemmeler, N. Vaidya, and V. Veeravalli. Selecting Transmit Powers and Carrier Sense Thresholds for CSMA Protocols. Technical report, University of Illinois at Urbana-Champaign, Oct 2004.

[6] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 46(2), Mar 2000.

[7] IEEE Computer Society. IEEE 802.11 Standard, IEEE Standard For Information Technology, 1999.

[8] K. Jamieson, B. Hull, A. Miu, and H. Balakrishnan. Understanding the Real-World Performance of Carrier Sense. In *E-WIND*, Philadelphia, PA, Aug 2005.

[9] A. Kochut, A. Vasan, A. Udaya Shankar, and A. Agrawala. Sniffing out the correct Physical Layer Capture model in 802.11b. In *ICNP*, Berlin, Germany, Oct 2004.

[10] J. Monks, V. Bhargavan, and W. Hwu. A Power Controlled Multiple Access Protocol for Wireless Packet Networks. In *Infocom*, Anchorage, AK, Apr 2001.

[11] J. Padhye, S. Agarwal, V. Padmanabhan, L. Qiu, A. Rao, and B. Zill. Estimation of Link Interference in Static Multi-hop Wireless Networks. In *IMC*, Berkeley, CA, Oct 2005.

[12] N. Salem and J. Hubaux. A Fair Scheduling for Wireless Mesh Networks. In *WiMesh*, Santa Clara, CA, Sept 2005.

[13] A. Vasan, R. Ramjee, and T. Woo. ECHOS - Enhanced Capacity 802.11 Hotspots. In *Infocom*, Miami, FL, Mar 2005.

[14] J. Weinmiller, M. Schlager, A. Festag, and A. Wolisz. Performance Study of Access Control in Wireless LANs - IEEE 802.11 DFWMAC and ETSI RES 10 Hiperlan. *Mobile Networks and Applications*, 2(1), Jul 1997.

[15] K. Xu, M. Gerla, and S. Bae. How Effective is the IEEE 802.11 RTS/CTS Handshake in Ad Hoc Networks? In *GLOBECOM*, Taipei, Taiwan, Nov 2002.

[16] H. Zhai and Y. Wang. Physical Carrier Sensing and Spatial Reuse in Multirate and Multihop Wireless Ad Hoc Networks. In *Infocom*, Barcelona, Spain, Apr 2006.

[17] J. Zhu, X. Guo, L. Yang, W. S. Conner, S. Roy, and M. Hazra. Adapting Physical Carrier Sensing to Maximize Spatial Reuse in 802.11 Mesh Networks. *Wireless Communications and Mobile Computing Journal*, 4(8), Nov 2004.

[18] J. Zhu, B. Metzler, X. Guo, and Y. Liu. Adaptive CSMA for Scalable Network Capacity in High-Density WLAN: a Hardware Prototyping Approach. In *Infocom*, Barcelona, Spain, Apr 2006.