Offered Load and Stability Controls in Multi-hop Wireless Networks

NG Ping-chung

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Master of Philosophy

in

Information Engineering

May 2005

摘要

在無線多跳(multi-hop) ad-hoc 網絡中, 工作站可能會將過量的數據傳輸至網絡中而引致高掉包率, 重新路由不穩(re-routing instability), 頻寬不公的問題。這份論文提出兩個解決以上問題的方法: 1) 提供負載控制(offered load control)及 2) 改良 ad-hoc 路由協定的穩定控制(stability control)。準確地說, 方法 1) 能解決高掉包率和頻寬不公的問題而方法 2) 則能解決重新路由不穩的問題。

在方法 1) 方面, 我們建立了一個由六台工作站所組成的多跳網絡,實驗證明了最理想提供負載的確存在於實際應用上。再者, 我們提出了一個數學分析方法以計算最理想提供負載的數值。我們相信這份論文是第一份企圖用數學分析去進行計算的論文, 這有別於先前單純是電腦模擬的研究成果。從數學分析中, 我們發現無線 ad-hoc 網絡的效能表現取決於兩個主要因素: 1) 隱蔽工作站(hidden-node) 及 2) 載波偵聽機制(carrier-sensing mechanism)。

在方法 2) 方面, 我們重新定義爲人熟識的頻寬不穩問題爲重新路由不穩問題, 因爲此問題是由 ad-hoc 路由協定中的重新路由程序所引起,而非媒體接取控制協定(MAC)的問題。爲此, 我們提出了一個改良 ad-hoc 路由協定的方法 – "保留原有路徑作傳輸直至發現新路徑 "。這個方法不需修改媒體接取控制協定, 故能容易地應用在現有的網絡結構及硬件配置下。再者, 這個方法能有效地減少百分之五十至七十的頻寬不穩問題並提升多流情況下平均頻寬高達十倍。

Abstract

In wireless multi-hop ad-hoc networks, stations may pump more traffic into the networks than can be supported, resulting in high packet-loss rate, re-routing instability and unfairness problems. In this thesis, we propose two solutions to eliminate these problems: 1) Offered load control and 2) Stability control by modifying ad-hoc routing protocols. Specifically, offered load control can be adopted to eliminate high packet loss rate and unfairness problems while our proposed modification on ad-hoc routing protocols (i.e., the "don't-break-before-you-can-make" strategy) can be used to eliminate the re-routing instability problem.

For 1), this thesis shows that controlling the offered load at the sources can eliminate high packet-loss rate. To verify the simulation results, we set up a real 6-node multi-hop network. The experimental measurements confirm the existence of the optimal offered load. In addition, we provide an analysis to estimate the optimal offered load that maximizes the throughput of a multi-hop traffic flow. We use this result to devise schemes that can achieve fairness when there are multiple flows from different sources to different destinations. We believe this is the first successful attempt to provide a quantitative analysis (as opposed to simulation) for the impact of hidden nodes, exposed nodes, and signal capture on sustainable throughput. The analysis is based on the observation that a large-scale 802.11 network with hidden nodes is a network in which the carrier-sensing capability breaks down partially. Its performance is therefore somewhere between a carrier-sensing network and an Aloha network. Indeed, our analytical

closed-form solution has the appearance of the throughput equation of the Aloha network. Our approach allows one to identify whether the performance of an 802.11 network is hidden-node limited or spatial-reuse limited.

For 2), we find that the well-known throughput instability problem is not restricted to TCP traffic only, but also occurs in UDP traffic. The associated throughput oscillations are not acceptable for real-time applications such as video conferencing and voice over IP. This thesis re-defines this throughput fluctuation as a "re-routing instability problem" since it is caused by the triggering of the re-routing function. In particular, we show that the throughput instability is mainly induced by re-routing, not the binary exponential back-off of the IEEE 802.11 MAC protocol. Turning off the re-routing function, for example, eliminates the problem. We believe that this is the first successful attempt to study this phenomenon in the context of re-routing instability. We propose to modify the ad-hoc routing protocols with a "don't-break-before-you-can-make" strategy. The scheme does not require modifications of the IEEE 802.11 standard, making it readily deployable using existing commercial Wireless LAN (WLAN) products. Simulations show that the proposed scheme can significantly reduce the throughput variation by 50-70% in the single-traffic flow case and improve the average throughput by up to ten times in multiple flow cases.

Acknowledgement

I would like to express a few words of gratitude. First, my sincere thanks to my supervisor Professor Liew Soung Chang for guiding me throughout my postgraduate studies. He taught me the first step into research – to be inquisitively motivated, to analyze intellectually and to diligently explore various approaches to a problem. I have learnt a lot from his enthusiasm in research and invaluable advices. Special appreciation to my father Po, mother Celina, aunt Sanie, sister Joyce, grandmother MaMa, uncle Victor, uncle Tony, grandparents and to the rest of my family for their support and concern. Finally, thanks also to my colleagues Mr. Wang Wei and Mr. Patrick Lam for their encouragements, especially at the beginning of my studies.

External Publications

Parts of the work in this thesis were published in 1) the paper "Offered Load Control in IEEE 802.11 Multi-hop Ad-hoc Networks" in the proceeding of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor System (MASS'04) and 2) the paper "Re-routing Instability in IEEE 802.11 Multi-hop Ad-hoc Networks" in the proceeding of the 4th IEEE International Workshop on Wireless Local Network (WLN'04 in LCN'04). Both papers won the Best Paper Award of the conferences. The extension of 2) was submitted as an invited paper to OCP Ad-hoc & Sensor Wireless Networks, An International Journal. The extension of 1) was submitted to IEEE/ACM Transactions on Networking.

Contents

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Overview and Motivation

A wireless multi-hop ad-hoc network provides quick and easy networking in circumstances that require temporary network services or when cabling is difficult - for example, in open-area conversations, environmental information gathering, disaster rescues and military usages. The IEEE 802.11 Distributed Co-ordination Function (DCF), based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), is the most popular MAC protocol used in wireless ad-hoc networks.

In wireless networks, interferences are location-dependent. For a traffic flow from a source node to a destination node in a multi-hop network, the nodes in the middle of the path have to contend with more nodes when forwarding the traffic of the flow. Experiencing lighter contention, the source node may inject more traffic into the path than can be forwarded by the later nodes. This may result in excessive packet losses and re-routing instability. When there are multiple flows, unfairness may also arise when some flows experience higher contention than other flows. To eliminate these problems, we propose two solutions: 1) Offered load control and 2) Stability control by modifying ad-hoc routing protocols.

1.2 Background of Offered Load Control

The capacity of wireless networks has been studied extensively. Much of the previous work focused on computing theoretical throughput bounds (e.g. [GK][LB]). Some of these throughput limits are obtained under the assumption of global scheduling [KJ][KN]. The popular IEEE 802.11 wireless networks in use today are not amenable to such global scheduling.

This thesis primarily focuses on 802.11 and 802.11-like networks. Although there were also prior investigations [XG][SA] on how to modify the 802.11 protocol to solve performance problems, we try not to perturb the protocol too drastically so that the same standard-based equipment can be used without major redesign.

To devise schemes to achieve high throughput and fairness in multi-hop networks, it is important to be able to analyze the *contention experienced by a node* as a function of the network topology and traffic flows in a quantitative manner. Such an analysis is currently lacking in the literature, possibly due to the fact that the analysis is complicated by the existence of hidden-node, exposed-node and signal-capturing effects. This thesis is a first attempt toward such a quantitative analysis. The analysis yields insight into the impact of different network parameters and properties on performance. As an example, we use our analysis to establish the optimal offered load for a traffic flow in this thesis. We also show that the analytical approach can be used to achieve fairness when there are multiple flows in the network.

Most previous studies of the hidden-node problem of 802.11 were conducted by simulations [LB][HG]. References [SK] [SK2] extended the hearing graph framework in [TK] to model hidden nodes and node mobility using a Markov chain. They established a relationship between the average number of stations hidden from each other and the likelihood of a station remaining in its Basic Service Area. Their results

on the effect of hidden nodes on throughput, however, were obtained from simulations, not analysis. In addition, the signal capture property that allows a packet to be received successfully despite transmissions by hidden nodes was ignored.

1.3 Background of Stability Control

The performance of wireless ad-hoc networks based on IEEE 802.11 has been extensively studied. Much of the previous work attempts to solve the one-hop performance problems [BW][BS]. In the multi-hop scenario, most of investigations focused on TCP performance [HV][JG]. Besides traditional TCP applications like file transfer and e-mail, the demands for real-time applications like multi-media streaming and voice services are also increasing. These real-time services are usually transported on UDP rather than TCP. In this thesis, we investigate a common phenomenon that leads to throughput degradations and oscillations for both TCP and UDP traffic in multi-hop networks: the re-routing instability problem.

Previous studies [XS][XS2][SA] showed that the so-called "TCP instability problem" exists in a multi-hop flow. References [XS][XS2] provided a solution to solve TCP instability by limiting the traffic at the transport layer. The solution assumes TCP Vegas and limits the TCP window size to at most 4. This limit bounds the number of packets in the path to prevent individual nodes from capturing the channel for a sustained period of time. Two observations are as follows. First, it is not clear that the solution is effective when there are multiple TCP flows along the same path, or when TCP flows on adjacent paths may interfere with the flow. Second, perhaps more importantly, the instability problem is caused by false declaration of link failures which is rooted at the link layer. In other words, this problem is not a phenomenon for TCP traffic only, but also for other types of traffic. The declaration of link failures in turn triggers the re-routing function, which exacerbates the situation. We believe that the problem should be properly defined as a "re-routing instability problem", and a

more general approach should be used to solve the problem by eliminating its root cause directly.

Reference [SA] reconfirmed the TCP throughput instability and proposed a modification of the IEEE 802.11 back-off algorithm such that only two back-off window sizes could be used. The main idea is to adopt the larger window for the next packet after a successful transmission. This allows other nodes using the smaller window to transmit with less chance of collisions. However, the decision for the choice of the value of these two back-off window sizes is based on the assumption that the packet payload is fixed at 1460bytes. We believe this assumption is not valid in real wireless LAN networks. When packets could be of different size, this scheme may fail to work properly.

1.4 Organization of the Thesis

The rest of this thesis is organized as follows. In Chapter 2, we review the major performance problems in multi-hop ad-hoc networks and suggest possible solutions to them. Our real-network experiments confirm the offered load control solution. Chapter 3 analyzes factors which degrade the throughput, and formulate a method to estimate the optimal offered load in a single multi-hop traffic flow. In particular, we present the derivation of the throughput limits imposed by (i) carrier sensing and (ii) hidden nodes. For simplicity, the analysis in Chapter 3 is based on a specific inter-node distance in the multi-hop flow. The analysis is extended to the general case in the Appendix. We show that in general, the throughput of a single multi-hop flow is hidden-node limited and not carrier-sensing limited. Chapter 4 gives an example where two opposite directional multi-hop flows may cause the throughput to be carrier-sensing limited instead. In Chapter 5, we show that an offered-load control scheme can achieve fairness of channel bandwidth usage among multiple flows.

In Chapter 6, we suggest a solution to deal with re-routing instability and show how our solution can be applied to the AODV routing protocol to eliminate instability. Chapter 7 analyzes factors that cause the triggering of re-routing. Finally, in Chapter 8, we investigate the link-layer penalty in a scenario with multiple flows interfering with each other and identify the factors that affect the impact of hidden-terminal flows. Chapter 9 concludes this thesis and suggests possible directions for future research.

Chapter 2 Performance Problems and Solutions

In a multi-hop ad-hoc network, sources may inject more traffic into the network than can be supported. This may result in two problems: 1) high packet loss rate, and 2) re-routing instability. In this chapter, we use an 8-node string multi-hop network as an example to illustrate these problems. In Fig. 2.1, node 1 sends a UDP traffic stream to node 8. The traffic is generated at node 1 in a saturated manner in which as soon as a packet is transmitted to node 2, another is waiting in line. The traffic at later nodes all originates from node 1 and is not saturated.

2.1 Simulation Set-up

The simulations in this thesis were conducted using NS2.1b9 [SK]. All nodes communicate using identical, half-duplex wireless radio based on the IEEE 802.11 Distributed Coordination Function (DCF), with data and basic rates set at 11Mbps. The RTS/CTS mechanism is turned off. Nodes are stationary. The transmission range is 250m, the carrier-sensing range is 550m, and the capture threshold, *CPThreshold*, is set to 10dB. The Ad-hoc On-Demand Distance Vector (AODV) routing protocol and the two-ray propagation model are used. Unless otherwise indicated, all traffic streams

use fixed packet size of 1460bytes. The TCP Reno algorithm is used since it is the most widely deployed TCP version. The advertised window (window_) of TCP is set to a large value to prevent the TCP traffic from being limited by the receiver.

2.2 High Packet-Drop Rate

Figure 2.2 shows the per-hop throughput of an 8-node flow obtained from simulations. The throughputs plotted are obtained by averaging over one-second intervals.

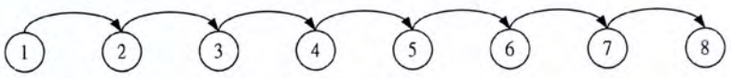


Figure 2.1. UDP traffic flow with node 1 as the source and node 8 as the destination in an 8-node multi-hop traffic flow

In Fig. 2.1, node 1 can sense the transmissions from nodes 2 and 3. This means node 1 must share the channel capacity with them. As a result, the throughput of the first hop is approximately 1/3 of the total channel capacity. Node 2, on the other hand, can be interfered by nodes 1, 3 and 4. This results in approximately 1/4 of the total channel capacity for the second hop. After that, each node must compete with four other nodes. The per-hop throughput stabilizes from the third hop to the last hop with approximately 1/5 of the total channel capacity. The first and the second nodes pump more packets to the following nodes than they can forward. This results in excessive packet drops at the second and the third node.

As shown in Fig. 2.2, the average throughput drops from 1.86Mbps at the first hop to 1.13Mbps at the last hop. In other words, about 40% of packets are lost in transit. This high packet-loss rate is undesirable, especially for real-time traffic without a retransmission mechanism at the upper protocol layer.

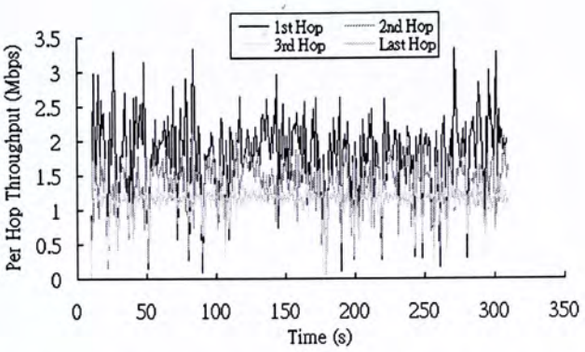


Figure 2.2. Per-hop throughputs of an 8-node flow

2.3 Re-routing Instability

Figure 2.2 also shows that the throughputs tend to oscillate widely over time. The throughput oscillations are caused by triggering of the re-routing function. In the multi-hop path, nodes 1 and 2 sense fewer interfering stations than later nodes. As a result, they pump more traffic into the network than it can support. This results in a high contention rate at the later nodes. When one of the later nodes fails to transmit a packet after a number of retries, it declares the link as being broken. The routing agent is then invoked to look for a new route. Before a new route is discovered, no packet can be transmitted, causing the throughput to drop drastically. In the string network topology under study, there is only one route from node 1 to node 8, so the routing agent will eventually "re-discover" the same route again. The breaking and rediscovery of the path results in the drastic throughput oscillations observed. For a general network with multiple paths from source to destination, the same throughput oscillations will still be expected. This is because the declaration of the link failure is caused by self-interference of traffic of the same flow at adjacent nodes.

2.3.1 Hidden-Node Problem

Besides the collisions of packets among nodes inside a carrier sensing range, the hidden-node problem further increases the chance of link-failure declarations.

Consider Fig. 2.3. When node 4 sends a packet to node 5, node 2 senses the channel to be busy while node 1 senses the channel to be idle, since node 4 is inside the carrier-sensing range of node 2 but outside that of node 1. Once node 1 senses the channel as idle, it may count down its back-off contention window until zero and transmit a packet to node 2.

If the transmission from node 4 is still in progress, node 2 will continue to sense the channel as busy, and it will not receive the packet from node 1. As a result, node 2 will not return an ACK to node 1. Node 1 may then time out and double the contention window size for retransmission later.

Meanwhile, node 4 transmits the packet successfully and is not aware of the collision at node 2. When transmitting the next packet, node 4 will use the minimum contention window size. The hidden-node scenario favors node 4, and the chance of collision at node 2 can not be reduced even though node 1 backs off before the next retry. The hidden-node problem increases the chance of multiple retries by node 1, making the wrong declaration of link failures and therefore re-routing instability more likely.

Note that the negative effect of a hidden node is much more than that of a contending node within the carrier-sensing range. This is because the carrier-sensing capability in the CSMA protocol breaks down with respect to the hidden node, making collisions much more likely.

2.3.2 Ineffectiveness of Solving Hidden-Node Problem with RTS/CTS

The RTS/CTS mechanism in 802.11 is designed to solve the hidden node problem. However, using RTS/CTS in multi-hop networks does not eliminate the hidden node problem. The effectiveness of RTS/CTS mechanism is based on the assumption that

transmissions by mutually hidden nodes are to a common receiver. Before the transmission of a hidden node begins, the receiver will forewarn other hidden nodes to prevent them from transmitting. This assumption may not hold in a multi-hop network.

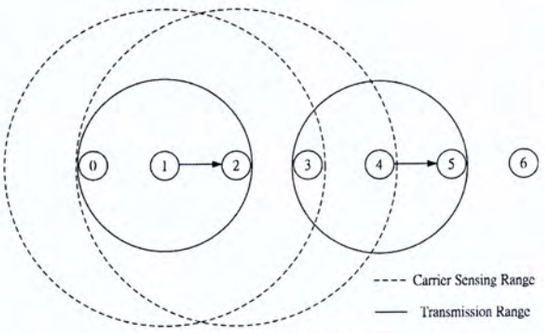


Figure 2.3. Node 4 as a hidden node to node 1

Consider the scenario in Fig. 2.3 again. The RTS transmitted by node 4 will cause a CTS to be returned by node 5. However, this CTS cannot be received by node 1. Therefore, node 1 may still transmit a packet to node 2 while the transmission of node 4 is in progress. The hidden-node effect as described in the previous chapter cannot be eliminated. For more details, the interested reader is referred to [XG], in which it was argued that when the carrier-sensing range is larger than two times of the transmission range, RTS/CTS is no longer needed. In this thesis, we assume the use of the basic access mode without RTS/CTS.

2.4 Solutions to High-Packet Loss Rate and Re-routing Instability

Reference [XS] demonstrated the existence of an instability problem for a TCP traffic flow in a multi-hop network. It provided a solution to solve TCP instability by limiting the traffic at the transport layer. The solution assumes TCP Vegas and limits the TCP window size to at most 4. As a result, only a maximum of four packets can be in transit

in the path at any one time. This prevents a node from hogging the channel for a long period of time.

Two observations are as follows. First, it is not clear that the solution is effective when there are multiple TCP flows along the same path, or when TCP flows on adjacent paths may interfere with the flow on the path. Second, the instability problem is caused by false declaration of link failures which is rooted at the link layer. This problem is not a phenomenon for TCP traffic only, but also for other types of traffic. Therefore, we believe a more general approach should attempt to solve this problem at the link layer.

There are two possible link-layer solutions: 1) do not declare link failures before a new path can be discovered; or 2) control the offered load at the source to reduce contention rate.

2.4.1 Link-Failure Re-routing

Strictly speaking, in the above scenario the link has not failed, although it is congested and the attempt to look for a new path is definitely warranted. However, before a new route can be discovered, one should continue to use the old route. That is, a "don't-break-before-you-can-make" strategy should be adopted.

To show that the throughput oscillations are in fact due to triggering of re-routing, we disabled the link-failure triggered re-routing function in one of our simulations. Figure 2.4 shows the result. The throughput becomes more stable and the drastic drops in throughput are eliminated.

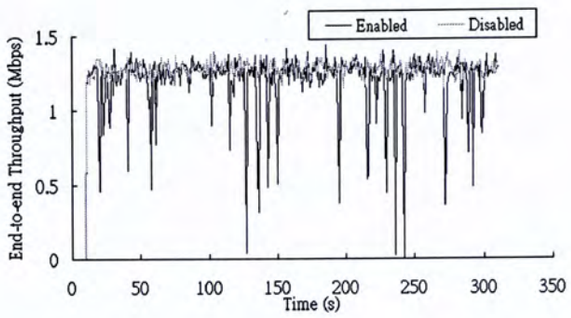


Figure 2.4. End-to-end throughputs with link-failure declarations enabled/disabled

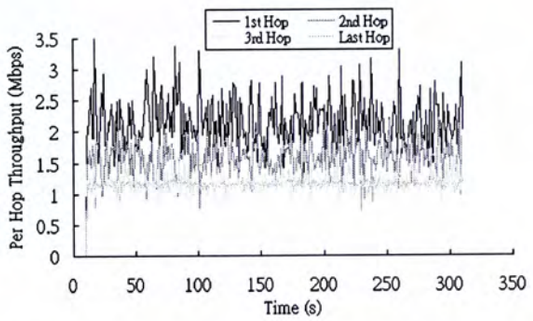


Figure 2.5. Per-hop throughputs of an 8-node flow after disabling link-failure re-routing

The study of multi-hop routing will be presented in Chapters 6 to 8. Here, we just want to point out that false triggers of re-routing should be studied as a separate problem. It could be more effectively dealt with directly rather than indirectly through higher-layer mechanisms. In Chapter 6, the "don't- break-before-you-can-make" strategy is implemented. Simulation results show that the strategy can prevent the re-routing instability problem and reduce the throughput variations in multi-hop ad-hoc networks drastically.

Figure 2.5, however, shows that the average throughput still drops from 2.14Mbps in the first hop to 1.15Mbps in the last hop even when re-routing is disabled. The high packet-loss rate remains.

2.4.2 Controlling Offered Load

To prevent high packet loss rate for a flow, the offered load must be controlled. Figure 2.6 plots the end-to-end throughput of a 12-node multi-hop path versus offered load. The peak throughput is obtained at offered load of 1.18Mbps. Offered load beyond this is unsustainable and high loss rate results because Throughput < Offered Load. This existence of an optimal offered load for a multi-hop path was also pointed out in [LB]. In this thesis, we provide an analysis to estimate the maximum sustainable throughput, and in doing so, reveal the factors that govern it.

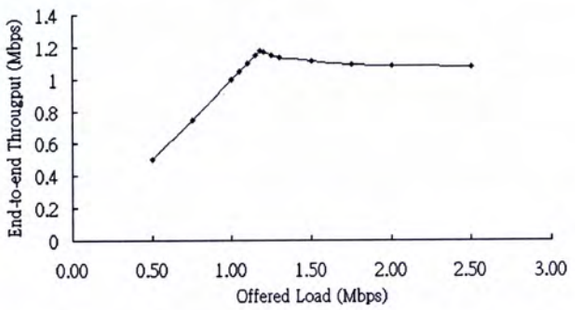


Figure 2.6. End-to-end throughput versus offered load in a 12-node flow

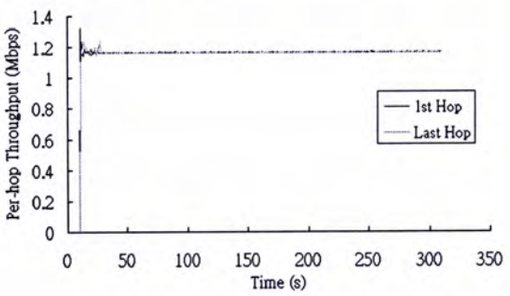


Figure 2.7. Per-hop throughputs with offered load control (at 1.18Mbps).

Controlling offered load also prevents the instability problem even when the link-failure-triggered re-routing in the routing agent is enabled. Figure 2.7 shows that the instability problem is eliminated by setting the offered load at the optimal sending rate (1.18Mbps). However, the instability problem is solved by avoiding congestion condition rather than the removal of the problematic strategy of suspending the link

usage before a new route can be discovered. A temporary external interference source (e.g., a nearby microwave oven) can easily cause the condition to arise again. We believe that even when offered-load control is exercised, a mechanism to deal with re-routing instability, such as our proposed "don't- break-before-you-can-make" strategy, is sill needed.

2.5 Verification of Simulation Results with Real-life Experimental Measurements

To verify the simulation results, we set up a real 6-node multi-hop network with six symmetric DELL Latitude D505 laptop PCs with 1.5GHz Celeron Mobile CPU and 512MB RAM. Each node has a Buffalo WLI2-CF-S11 IEEE 802.11b Wireless LAN card (as shown in Fig. 2.8). All nodes run RedHat Linux 9 with HostAP [HA] driver. To facilitate experimentation, we fixed the transmission power of each WLAN card to a small value (-38dBm), with basic and data rates set at 11Mbps. We obtained the transmission range of TxRange $\approx$ 2m and the carrier-sensing range of CSRange$\approx$5m=2.5*TxRange by following similar approaches as mentioned in [AB]. We fixed the routing table of each node and set the distance between successive nodes to 2m. The data sources are UDP traffic streams with fixed packet size of 1460bytes. Figure 2.9 shows that the simulation throughputs match closely with the experimental measurements, indicating that our simulations do not contain major deficiencies. We adjusted the offered load at the source in the 6-node network. Figure 2.10 shows the existence of the optimal offered load (1.25Mbps). This confirms our simulation results.

Figure 2.8. A 6-node multi-hop wireless network

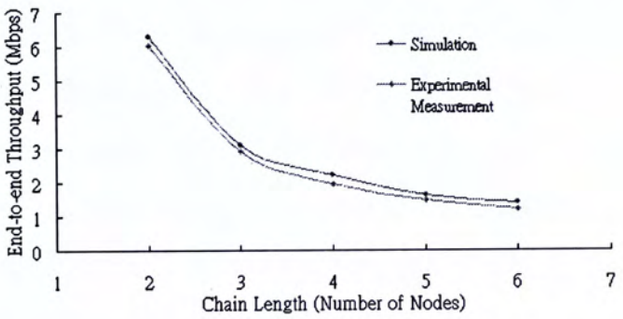


Figure 2.9. End-to-end throughput versus number of nodes in a string multi-hop network with saturated traffic source

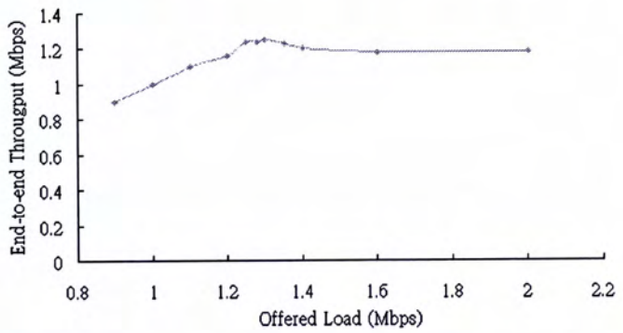


Figure 2.10. Experimental Measurements of end-to-end throughput versus offered load in a 6-node flow

Chapter 3 Offered Load Control

We now consider the problem of determining the optimal offered load (i.e., the maximum sustainable throughput) for a single flow in a multi-hop network. The throughput is limited by two factors: 1) the hidden-node and exposed-node problems; and 2) the carrier sensing mechanism. We first analyze the impact of these two factors. After that, we present numerical results showing that the analytical results match the simulation results closely. Our analysis yields a closed-form solution, which we believe provides the insight and foundation for the study of more complex situations involving multiple flows in future work.

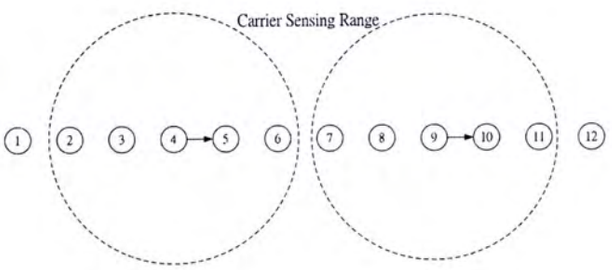


Figure 3.1. A 12-node string multi-hop network

3.1 Capacity Limited by the Hidden-node and Exposed-node Problems

We will express the throughput of a single flow in terms of the airtime used by a node.

Figure 3.1 shows a chain of 12 nodes. The traffic flows from left to right. Imagine that this is a longer chain with more nodes extending to the left of node 1 and the right of node 12. By the time the traffic reaches node 1, a "steady state in space" has been reached in which all nodes experience the same situation without the boundary effects. The question we ask is "What is the maximum throughput that can flow through this chain?"

Consider a long stretch of time in the interval [0, *Time*]. Let Si be the airtime within this interval that a "steady-state" node i transmits. This airtime includes the transmission times of the data packets (PACKET), the transmission times of the acknowledgements (ACK) from node (i+1), the durations of the distributed interframe space (DIFS) and the durations of the short interframe space (SIFS). Also, included in Si are the times used up for retransmissions in case of collisions. However, Si does not include the count-down of the idle slots of the contention window, since adjacent nodes can count down together and these count-down times are not unshared resources used up exclusively by node i.

Let $x = |Si| / Time$, T = traffic throughput (in Mbps) flowing through the a "steady-state" node (and therefore also the end-to-end throughput), and ρ = the collision probability for a transmission. Then, we have.

$$T = x \cdot (1 - \rho) \cdot d \cdot data\_rate \qquad (1)$$

where $d = DATA/(DIFS + PACKET + SIFS + ACK)$ which is the proportion of time within x that is used to transmit the data payload; and $data\_rate$ is the data transmission rate. Note that *DATA* is the pure payload transmission time of a packet, while *PACKET* includes transmission times of the physical preamble, MAC header, and other higher-layer headers.

For simplicity, we assume that the carrier-sensing mechanism eliminates collisions to the extent that they are negligible, and that collisions are predominantly caused by hidden and exposed nodes. Consider node 4 in Fig. 3.1. Our assumption means that the transmission of node 4 will not collide with the transmissions of nodes 2, 3, 5, and 6; but node 1 and node 7 may cause collisions at node 4 due to the exposed and hidden-node effects, respectively.

To derive ρ, we consider the "vulnerable period" induced by the hidden and exposed nodes. During a vulnerable period, a node may suffer a collision if it transmits a packet. ρ can be decomposed into two factors: 1) the collision probability due to a hidden node (ρ_{HT}) ,and 2) the collision probability due to an exposed node (ρ_{ET}). They are related as follows:

$$\rho = 1-(1-\rho_{HT})(1-\rho_{ET}) \qquad (2)$$

In the following chapters, we first explain the effect of the packet arrival order on signal capture. Then, we derive ρ_{HT} and ρ_{ET}. We show that the later is relatively small and can be ignored.

Our analysis is based on the following assumptions:
(A.1) The transmission of a node is independent of the transmissions of nodes outside its carrier sensing range.
(A.2) The packet collision probability of a node with nodes inside its carrier sensing range is negligible, thanks to the carrier-sensing property of CSMA.

3.1.1 Signal Capture

In Fig. 3.2, both nodes 4 and 7 have a packet to transmit. This may cause the

aforementioned hidden-node collision. However, the signal capturing property may still allow a packet from node 4 to be received successfully, provided it transmits before node 7.

More specifically, suppose that node 4 transmits first and the signal power of the transmission received at node 5 is P_4. Node 7 then transmits a packet with power of P_7 at node 5. If $P_4 > P_7 + CPThreshold$, where *CPThreshold* is the capture threshold, then no collision occurs, and node 5 can still receive the packet from node 4 successfully.

However, if node 7 transmits first, node 5 senses the signal from node 7 and declares the channel to be busy. In that case, a newly arriving packet from node 4 can not be received even if $P_4 > P_7 + CPThreshold$. Effectively, the packet from node 4 to node 5 experiences a collision.

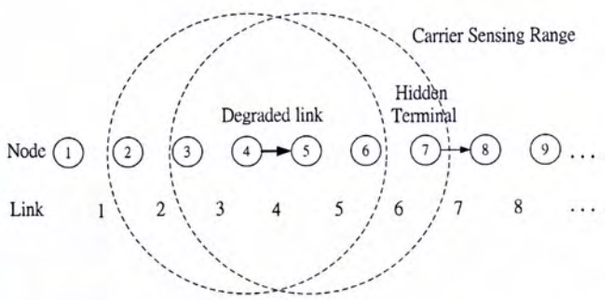


Figure 3.2. Node 7 as a hidden-node to node 4

For the sake of argument, suppose that *CPThreshold* is set to be 10dB. Let d be the fixed distance between nodes. In this case, node 4 and node 7 are separated by a distance larger than the carrier sensing range. Thus, node 4 and node 7 can send packets at the same time. From [TR], in a two ray propagation model, the signal-to-noise ratio at node 5 is

$$SNR = P_4 / P_7 = (2d/d)^4 = 2^4 = 16 > CPThreshold$$

This means that the power level of the packet transmitted by node 4 and received at

node 5 is always more than *CPThreshold* higher than the power level of the received signal from node 7.

3.1.2 Analysis of Vulnerable Period induced by Hidden Nodes

In the analysis of the hidden-node problem, the key is to identify the vulnerable period during which the transmission of a node will collide with the transmission of a hidden node. This is illustrated in Fig. 3.3. Note that a hidden-node collision only occurs if the transmissions of nodes 4 and 7 overlap and that the transmission of node 7 precedes that of node 4. More specifically, after receiving the PHY header from node 7, node 5 will declare the channel as busy and will not receive the data from node 4 for the duration of the transmission time of the MAC header and DATA.

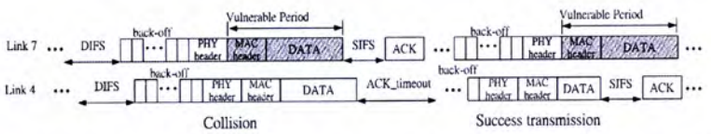


Figure 3.3. Collision occurs when the transmission of node 4 begins inside the vulnerable period.

If this were an Aloha network, nodes 4 and 7 could collide at anytime during the interval [0, *Time*]. However, in a carrier-sense network, some of the times during this interval must be removed from the "sample space" in the analysis of collision probability.

Consider Fig. 3.1. When node 5 or 6 transmits, node 4 and node 7 will not by assumption (A.2). This means that *S4*, *S5*, and *S6* are non-overlapping; and *S5*, *S6*, and *S7* are non-overlapping. In particular, node 7 cannot cause collision on node 4 during *S5* and *S6*. Now, nodes 5 and 6 use up $2 \cdot x$ fraction of the airtime during [0, *Time*]. The remaining fraction of airtime where node 4 and node 7 may collide is $(1 - 2 \cdot x)$. Since node 7 uses x fraction of remaining airtime for transmissions, the vulnerable period

induced by node 7 on node 4 is

$$\rho_{HT} = \frac{x}{1-2x} \cdot a \qquad (3)$$

by assumption (A.1), where

$$a = \frac{MAC\_Header + DATA}{DIFS + PACKET + SIFS + ACK}$$

is fraction of time used for transmitting the MAC header and data.

3.1.3 Analysis of Vulnerable Period induced by Exposed Nodes

In Fig. 3.4, nodes 1 and 4 are outside the carrier-sensing range of each other. At a given time, both nodes 1 and 4 attempt to send a packet to nodes 2 and 5, respectively.

Node 1 is outside the carrier-sensing range of node 4, so the transmission of node 1 does not affect the transmission of node 4. However, node 2 is inside the carrier-sensing range of node 4. Node 4 can sense the ACK returned from node 2 to node 1. When the ACK from node 5 overlaps with the ACK from node 2 at node 4 and the ACK from node 5 reaches node 4 later than that of node 2 as shown in Fig. 3.5, a collision occurs.

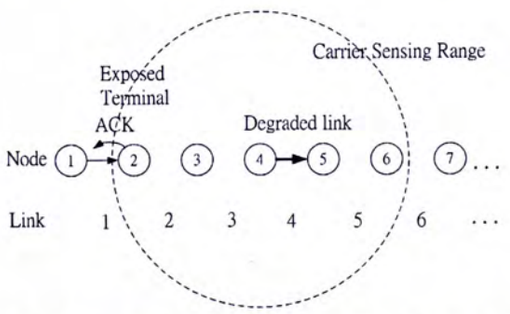


Figure 3.4. Node 2 as an exposed-node to node 4

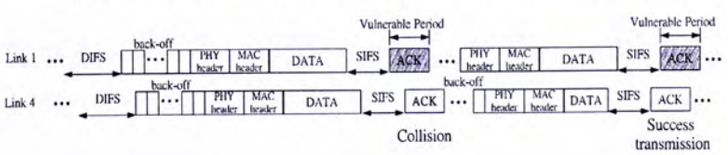


Figure 3.5. Collision occurs when the ACK from node 5 begins inside the vulnerable period.

However, this ACK-ACK collision can only occur if the transmission of node 4 begins at time $t <$ SIFS later than the transmission of node 1. When $t >$ SIFS, the transmission of node 4 is still in progress and node 4 is not aware of the transmission of ACK from node 2: that is, node 4 will not be able to read the physical preamble in ACK from node 2 and initiate the physical carrier-sensing mechanism that prevents node 4 from receiving the ACK from node 5 later. Therefore, no collisions can occur if $t >$ SIFS. Under the randomization assumption of (A.1), the chance for $t <$ SIFS equals: $SIFS/(DIFS + PACKET + SIFS + ACK) = 0.0064$ under the settings in Table 3.1. Therefore, the ACK-ACK collision rarely happens. This has been borne out by our simulations, in which we could not detect collisions due to the exposed-node problem. We will therefore assume that the degradation caused by exposed nodes is negligible in our analysis henceforth. That is, equation (2) becomes

$$\rho \approx \rho_{HT} \qquad (4)$$

3.1.4 Sustainable Throughput

Substituting equations (3) and (4) in (1), we have

$$T = x \cdot (1 - a \cdot \frac{x}{1-2x}) \cdot d \cdot data\_rate \qquad (5)$$

Physically, there are two factors affecting T in the opposing directions. As x increases, more airtime is used by a node and there is less idling, and this should push T up. However, larger x also leads to a larger vulnerable period, pulling T down.

Differentiating (5) with respect to x and setting $dT/dx = 0$, the optimal value of x that

maximizes the throughput is given by

$$x^* = \frac{(2+a) - \sqrt{a^2 + 2a}}{4 + 2a} \qquad (6)$$

Substituting equation (6) in (1) yields the maximum sustainable throughput $T(x^*)$. The offered load should be set to a value smaller than $T(x^*)$ to prevent excessive packet loss.

3.2 Capacity Limited by Carrier Sensing Property

Carrier sensing prevents simultaneous transmissions of nodes within the carrier-sensing range of a node. This imposes a limit on channel spatial-reuse. Potentially, the throughput could be limited by carrier sensing rather than hidden nodes. The maximum throughput derived above is due to hidden nodes. We now consider whether carrier sensing further reduces the sustainable throughput. We focus on the local observation of a particular node.

Let C_i be the airtime used for counting down the contention window of node *i*. Consider node 4 as the *local observer*. Within the time window [0, *Time*], it can only observe the airtimes used by the nodes within its carrier-sensing range, as illustrated in Fig. 3.1. So, as far as node 4 is concerned, it only observes $C4$, $S2$, $S3$, $S4$, $S5$ and $S6$. Note that it does not observe the countdowns of nodes 2, 3, 5, and 6. In particular, *C2*, *C3*, *C5*, and *C6* may overlap with *C4*. From node 4's point of view, the total airtimes used up by these nodes cannot exceed *Time*. Thus, $|C4 \cup S2 \cup S3 \cup S4 \cup S5 \cup S6| \leq$ *Time*.

Define $y = |C4 \cup S2 \cup S3 \cup S4 \cup S5 \cup S6| / Time$, to be the fraction of airtime used up

by these nodes within the interval [0, *Time*]. Now, $|C4 \cup S2 \cup S3 \cup S4 \cup S5 \cup S6|$ can be decomposed using the inclusion-exclusion principle:

$$|C4 \cup S2 \cup S3 \cup S4 \cup S5 \cup S6| = |C4| + |S2| + |S3| + ... + |S6| - |C4 \cap S2| - |S2 \cap S3| - |S2 \cap S4| - ... + |C4 \cap S2 \cap S3| + |S2 \cap S3 \cap S4| + ...$$

However, we note that the intersection of the airtimes used by any three nodes or above is null, thanks to carrier sensing. Also, node 4 can count down only if nodes 2, 3, 5 and 6 are not transmitting, thus $C4 \cap Si$ for i = 2, 3, 5, 6 is null. In addition, the intersections of airtimes used by two nodes are non-null only for $S2 \cap S5$, $S3 \cap S6$, and $S2 \cap S6$. We therefore have

$$y \cdot Time = |C4| + \sum_{i=2}^{6} |Si| - |S2 \cap S5| - |S3 \cap S6| - |S2 \cap S6| \qquad (7)$$

Let $z = |Ci| / Time$. By assumption (A.2), the packet collision probability is negligible. Before the transmission of a data packet, the node randomly chooses a contention window size between [0, $CW_{\min} - 1$] for countdown. The average time for counting down the contention window becomes $(CW_{\min} - 1) \cdot \sigma / 2 = 15.5 \cdot \sigma$ where σ is the mini slot time. We can express z in term of x,

$$z = x \cdot c$$

where $c = \dfrac{(CW_{\min} - 1) \cdot \sigma / 2}{DIFS + PACKET + SIFS + ACK}$

Consider the overlapped airtimes of node 2 and node 5. When node 3 or 4 transmits or when node 4 is counting down, node 2 and 5 do not transmit, by virtue of carrier sensing. The remaining fraction of airtime where *S*2 and *S*5 may overlap is (1-2*x*-*cx*). In particular, we have

$$|S2 \cap S5| = |S3 \cap S6| = \frac{x^2}{1-(2+c)x} \cdot Time \qquad (8)$$

Nodes 3 and 6 face the same situation. Hence, $|S2 \cap S5| = |S3 \cap S6|$ in (8).

For $|S2 \cap S6|$, the amount of airtime of node 2 that may overlap with that of node 6 is ($|S2|$-$|S2 \cap S5|$), and the amount of airtime of node 6 that may overlap with that of node 2 is ($|S6|$-$|S3 \cap S6|$). The "sample space" within which $S2$ and $S6$ may overlap is [0, *Time*] – $S3$ – $S4$ – $S5$ – $C4$. As a result, we have

$$|S2 \cap S6| = \frac{(|S2| - |S2 \cap S5|) \cdot (|S6| - |S3 \cap S6|)}{Time - |S3| - |S4| - |S5| - |C4|}$$

The above gives

$$|S2 \cap S6| = \frac{(x - x^2/(1-(2+c)x))^2}{1-(3+c)x} \cdot Time \qquad (9)$$

Substituting equations (8) and (9) into (7), we have

$$y = (5+c)x - \frac{2x^2}{1-(2+c)x} - \frac{x^2(1-(3+c)x)}{(1-(2+c)x)^2} \qquad (10)$$

The value of x for $y > 1$ is an "infeasible region". Let the x at which $y(x) = 1$ be x'. This corresponds to a saturated case where the node always has packets to send, so either it is counting down, transmitting a packet itself, or sensing the transmission by a neighbor. The saturated case may not occur if the system is hidden-node limited because packets from upstream fail to arrive fast enough to keep the node busy all the time.

If the throughput obtained from x' is greater than the throughput obtained from x^* of equation (6), then the system throughput is limited by hidden nodes. However, if the throughput obtained from x' is smaller than that from x^*, the system is limited by the spatial-reuse restriction caused by the carrier-sensing mechanism. The optimal

throughput of the hidden-node limited analysis can be obtained by substituting x^* into equation (5) while that of the carrier-sensing limited analysis can be acquired by substituting x' into equation (1) with the collision probability caused by hidden-terminal (ρ) set to zero. In the next chapter, we show that for the case under study, the system throughput is hidden-node limited.

3.3 Numerical Results

In Chapters 3.1 and 3.2, we have provided the analysis on the capacity limited by 1) hidden nodes and exposed nodes and 2) the carrier sensing mechanism. We now examine the numerical results. Table 3.1 shows the system parameters assumed, and the associated analytical T and y.

For 1), Figure 3.6 shows the simulation results, which indicate that the optimal offered load (or sustainable throughout) decreases as the number of nodes increases in a string multi-hop topology. For chains with more than 20 nodes, the optimal offered load stabilizes at 1.16Mbps. Our analytical result yields 1.218Mbps, a close match.

Table 3.1. System parameters and Max Throughput.

Packet payload (*DATA*)	1460 bytes
UDP/IP header	20 bytes
MAC header	28 bytes
PHY header	24 bytes
ACK size	14 bytes
Channel bit rate	11 Mbps
PHY header bit rate	1 Mbps
Slot time σ	20 *us*
SIFS	10 *us*
DIFS	50 *us*
$CW_{\min}$	32

CW_{max}	1024
Retransmission limit	7
x^*	0.24445
$T(x^*)$	1.2183Mbps
$y(x^*)$	0.95166
x'	0.3110
$T(x')$	2.3421Mbps
$y(x')$	1

Table 3.2. A summary of variables used in the analytical model.

ρ	collision probability for a transmission
ρ_{HT}	collision probability due to a hidden node
ρ_{ET}	collision probability due to an exposed node
T	traffic throughput
a	fraction of time used for transmitting the MAC header and data
d	proportion of time within x that is used to transmit the data payload
k	number of nodes within a carrier-sensing range
l	uniform distance between two successive nodes

For the analytical results, Fig. 3.7 plots network throughput T (left y-axis) versus x as limited by the hidden-node effect, and y (right y-axis) versus x as limited by carrier sensing. The maximum $T(x^*)$=1.218Mbps is achieved with x^*=0.245. For x^*, y = 0.952 < 1. This means that the capacity of the network is limited by hidden nodes rather than carrier sensing. Note that when the number of nodes within a carrier-sensing region is large and the number of hidden nodes is small, the capacity could in principle be limited by carrier sensing instead. This could be the case, for example, when the carrier sensing range is much larger than that of the transmission range.

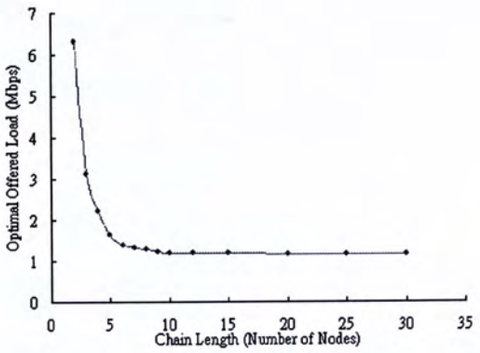


Figure 3.6. Optimal offered load versus number of nodes in a string multi-hop network.

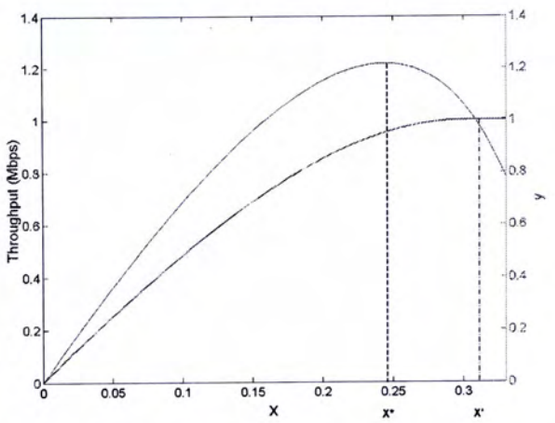


Figure 3.7. The flow throughput T in Mbps (left y-axis) and the fraction of airtime y used by all nodes within a carrier sensing range (right y-axis) versus the airtime x used by a node.

For the interested reader, reference [NL2] showed that the carrier-sensing mechanism of 802.11 may impose a constraint on channel spatial-reuse that is overly restrictive, making the network performance non-scalable. The same paper also provides a scheme that modifies 802.11 slightly to achieve scalable performance. We believe the scheme may relieve both the carrier-sensing and hidden-node effects being investigated here, although further study will be needed to validate this conjecture.

3.4 General Throughput Analysis of a Single Multi-hop Traffic Flow

In the previous chapters, we have shown that the capacity of a single string multi-hop network is hidden-node limited when the distance between two successive wireless nodes is set to the maximum transmission range (i.e., 250m). In this chapter, we discuss the capacities of other string network topologies. In particular, we show that our analytical results, again, match simulation results closely when we reduce the distance between two successive nodes to 170m and 130m. We study the link distance up to 130m because some intermediate nodes may be skipped if the node-to-node distance is less than 125m. Since this general analysis is similar to the analysis in Chapters 3.1 and 3.2, we refer interested readers to the Appendix for details.

Let k be the number of nodes within a carrier-sensing range (CSRange, i.e., 550m) and let l be the uniform distance between two successive nodes. For example, $k = 2$ if $l = 250m$ (the minimum value of k since nodes are separated by maximum transmission range), $k = 3$ if $l = 170m$ and $k = 4$ if $l = 130m$ (this is the largest value of k, since closer packing with larger k allows data signal to jump over successive nodes).

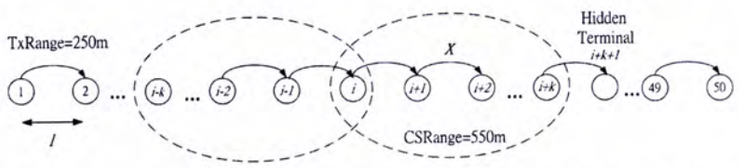


Figure 3.8. A 50-node string multi-hop network with variables k and l.

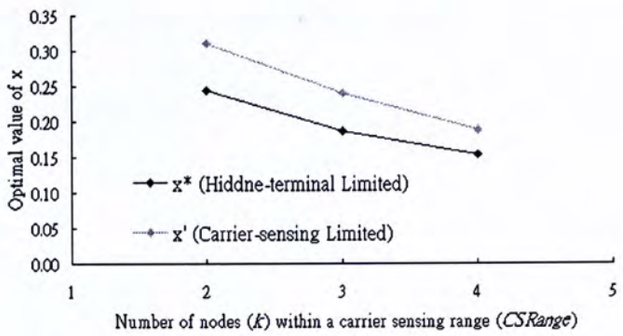


Figure 3.9. Optimal values of x versus number of nodes within a carrier sensing range

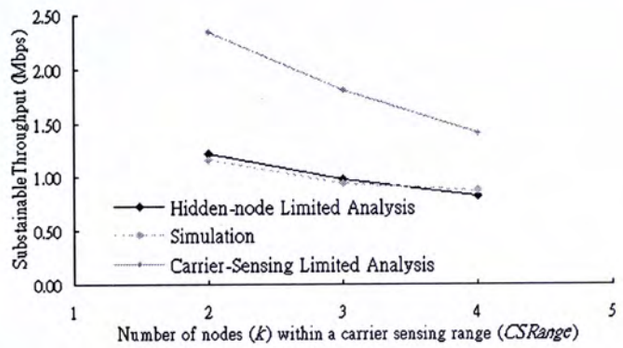


Figure 3.10. Sustainable throughput versus number of nodes within a carrier sensing range

We now examine the numerical results when the distance between two successive nodes is set to 170m (k=3) and 130m (k=4). Figure 3.9 plots the optimal values of x by 1) hidden nodes and exposed nodes and 2) the carrier sensing property when k=2 to 4. In these three cases, x^* is less than x' which means the capacities of these string network topologies are still hidden-node limited rather than carrier-sensing limited. As a side note, the graph also implies that if a strategy could be devised to remove the hidden-node effect, considerable throughput improvement could be obtained.

Figure 3.10 shows the simulation results for chains with 50 nodes. Our hidden-node analytical results yield close matches with simulation results.

3.5 Throughput Analysis on Topologies with Variable Distances between Successive Nodes

In our previous analysis, we assume the distances between successive nodes are constant such that all nodes experience the same situation. However, this assumption may be invalid when distances between successive nodes vary. Figure 3.11 shows an example. The link between node 17 and node 18 suffers from five hidden-nodes (i.e., nodes 20 to 24). Node 17 can sense four nearby nodes (i.e., nodes 15, 16, 18, 19). The link between node 20 and node 21 suffers from one hidden-node (i.e., node 24). Node 20 has to share the channel capacity with five other nodes (i.e., nodes 18, 19, 21, 22, 23).

Simulation shows that the maximum throughput of the flow in Fig. 3.11 is 0.70Mbps, a 40% reduction compared with the maximum throughput (1.16Mbps) of a linear flow with nodes separated by 250m. This throughput is even smaller than that of a linear flow with nodes separated by 130m (0.88Mbps). This means the capacity is not limited by the closer packing at the end of the flow (node 20 to 25), but limited by the larger vulnerable period induced by the multiple hidden nodes.

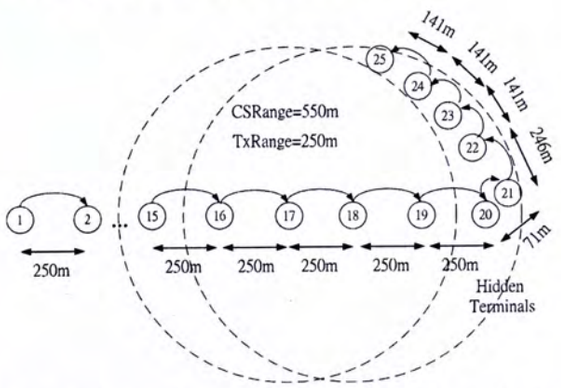


Figure 3.11. A 25-node multi-hop network with multiple hidden-nodes

The different numbers of hidden nodes and carrier-sensed nodes complicate the

analysis. Because of the asymmetry, the airtimes used by different nodes are different, complicating the analysis. A possible analytical method is to use an iterative approach: First, we obtain the airtime used by the last node (e.g., node 24 in Fig. 3.11), x_n, in terms of the throughput T. Then, T as a function of x_{n-1}, x_n is computed. From this, we obtain x_{n-1} in terms of T. This is repeated until we have x_1 in terms of T. Then, we compute the maximum T. This iterative approach, however, does not yield a nice closed-form solution.

Chapter 4 Discussions of Other Special Cases

In Chapter 3, we have shown that the capacities of string network topologies are hidden-node limited. In this chapter, we will demonstrate a carrier-sensing limited scenario. In addition, we will show a practical solution by which the hidden-node problem can be eliminated and the sustainable throughput can be boosted.

4.1 A Carrier-sensing Limited Example

Figure 4.1 shows two flows with opposite directions in an 11-node multi-hop network. Two UDP traffic sources at node 6 and node 7 transmit data to each end (node 1 and node 11) through the 5-hop (to the left) and 4-hop (to the right) networks respectively. In this scenario, there is no hidden node since the sender of each link can carrier-sense other transmitters that can be sensed by the receiver of the link.

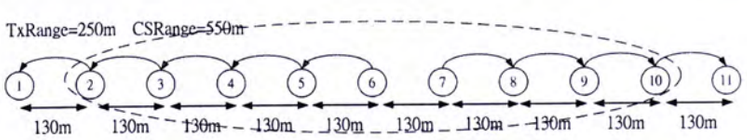


Figure 4.1. An 11-node multi-hop network with two opposite directional flows.

Consider node 6 as the local observer and nodes within its carrier-sensing range in Fig. 4.1. The total airtimes used up by these nodes cannot exceed *Time*. That is, $|C6 \cup S2 \cup S3 \cup \ldots \cup S9 \cup S10| \leq Time$.

Simulation shows that the optimal sustainable throughput for each flow is obtained at 0.920Mbps which is higher than the simulation throughput (0.870Mbps) obtained in a single flow multi-hop case as shown in Fig. 3.10. This means the throughput is boosted by releasing the bundle of hidden-node as there is no hidden-node problem in this specific topology.

4.2 A Practical Solution to Improve Throughput

In Sub-section 3.3, we have shown that the optimal value of x obtained by hidden-node analysis (x^*) is less than that of the carrier-sensing analysis (x'). This means the network throughput is limited by hidden nodes rather than the carrier-sensing mechanism. If the hidden-node problem can be eliminated, we can increase the sustainable throughput.

To do this, node 5 as shown in Fig. 3.2 must be able to receive the signal from node 4 successfully even though node 5 can sense the signal from node 7. In some commercial 802.11 chips, there is a so-called "re-start mode" in the receiver design. If the receiver is in the midst of receiving a signal, another signal with sufficiently large power margin arrives, the receiver will switch to receive the new signal. This feature can be used to lift the hidden-node problem in multi-hop networks.

With the two-ray ground propagation model, when nodes 4 and 7 transmit at the same time (as shown in Fig. 3.2), the signal to noise ratio (*SNR*) at node 5 is 16 (as shown in Sub-section 3.1.1) which is sufficiently larger than the capture threshold (*CPThresh*=10dB). With the re-start mode, node 5 can switch to receive the stronger

signal from node 4 even if the signal from node 7 reaches node 5 before that of node 4. In this way, the vulnerable period induced by the hidden-node (node 7) can be eliminated.

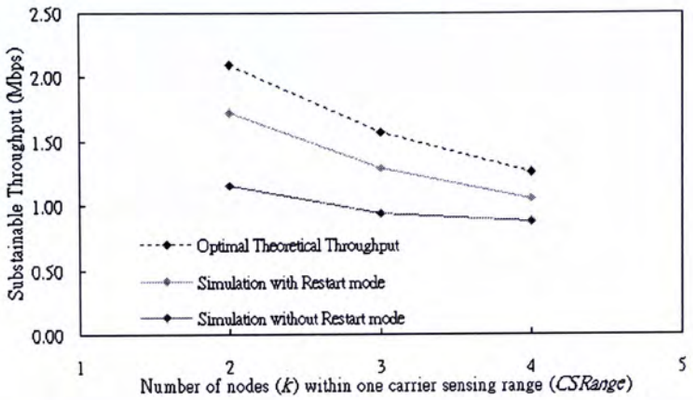


Figure 4.2. Sustainable throughput with restart mode versus number of nodes within a carrier sensing range

We implemented the re-start mode in NS2. Figure 4.2 shows the simulation results. The sustainable throughput can be boosted up to 50% with the use of the re-start mode. In Fig. 4.2, the optimal theoretical throughputs can be used as benchmarks for comparisons and are obtained under the assumption of perfect scheduling. For example, as shown in Fig. 4.3, nodes 1, 4, 7, 10 … are scheduled to transmit simultaneously when k =2 and this yields 1/3 of the total channel capacity (1/3*6.3=2.1Mbps).

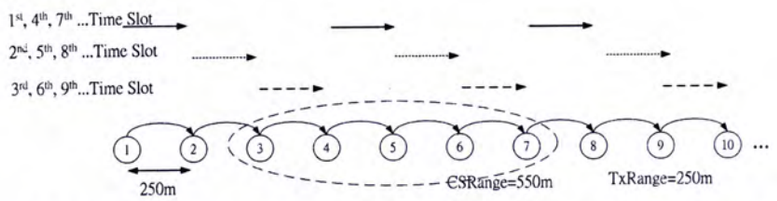


Figure 4.3. A single string multi-hop network with transmissions of prefect scheduling

Chapter 5 Achieving Fairness in Other Network Topologies

We have shown that controlling the offered load at the source node of a single-flow path eliminates high packet-loss rate. In this chapter, we will show that controlling the offered load can achieve fairness of channel bandwidth usage among multiple flows.

5.1 Lattice Topology

To study the interactions among multiple flows, we consider an N x M lattice network as shown in Fig. 5.1. All nodes are separated by 200m. The nodes in the first column are the source nodes, and each of them injects traffic into the networks destined for nodes in the last column. In our simulation, we set M=N for convenience sake.

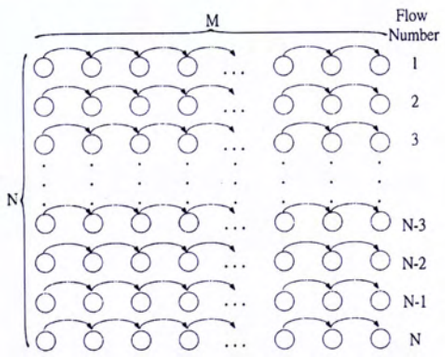


Figure 5.1. An N x M lattice topology with N traffic flows from left to right

Figure 5.2 shows that the average end-to-end throughput of all flows decreases as the size of the lattice increases. Reference [LB] reported a similar trend in the lattice topology. In addition, we observe an unfairness problem between flows. Figure 5.3 shows the per-flow end-to-end throughput of a 4x4 lattice network. The flows on two sides (flow 1 and 4) have fewer interfering stations than the middle flows (flow 2 and 3). This causes the flows on two sides to pump more traffic into the network than the middle flows. In the 4x4 lattice network, flow 2 and flow 3 have to compete with the aggressive transmissions of flow 1 and flow 4, resulting in severe throughput degradations.

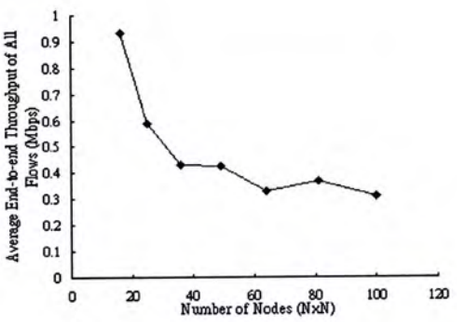


Figure 5.2. Average end-to-end throughput of all flows versus number of nodes in an N x N lattice network when the source nodes inject traffic into the network in a saturated manner

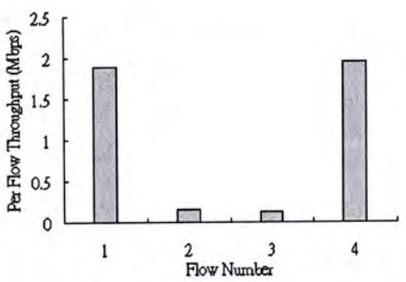


Figure 5.3. Per-flow end-to-end throughput of a 4x4 lattice network with saturated traffic sources

The uneven numbers of competing stations in the lattice structure severely degrades the performances of flows in the middle. Controlling the offered load in lattice

networks prevents aggressive transmissions from two sides to give more chances for nodes in the middle to transmit.

Figure 5.4 shows that a fair share of the channel throughput among the flows in an 8x8 lattice can be achieved when the offered loads at the sources are limited to 0.256Mbps. This sustainable offered load is obtained by extending the single-flow analysis given in the preceding chapters. Although the average end-to-end throughput is slightly lower than that of using saturated traffic sources, controlling the offered load can prevent unacceptable per-flow throughput performance and achieve fair bandwidth allocation.

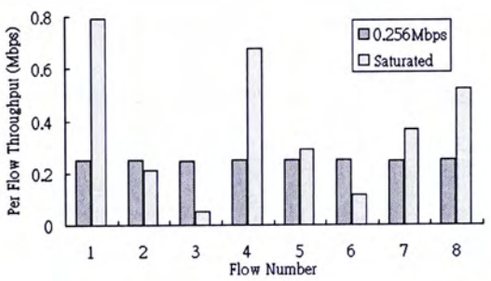
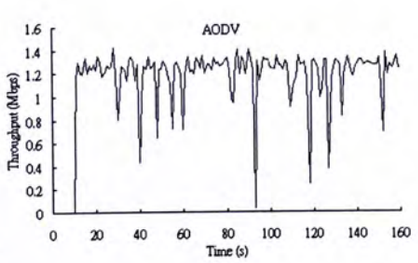


Figure 5.4. Per-flow throughput of an 8x8 lattice network with the offered load of 0.256Mbps and saturated traffic sources

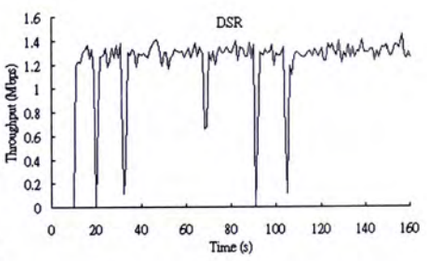
Chapter 6 Stability Control

6.1 Ad-hoc routing protocols

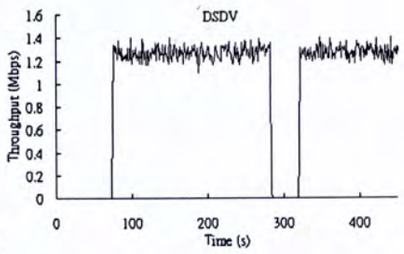
Numerous ad-hoc routing protocols have been proposed in the literature. They can be categorized into two approaches: 1) proactive / table-driven; or 2) reactive / on-demand-driven [CT]. The proactive approach protocols (e.g., Destination Sequenced Distance Vector (DSDV)), attempt to preserve consistent and up-to-date routing information from each node to every other node in the entire network. Each node maintains its own routing table and propagates route updates throughout the network to notify other nodes of changes in the network topology. In reactive approach protocols (e.g. Ad-hoc On-demand Distance Vector (AODV) and Dynamic Source Routing (DSR)), route discoveries are initiated only when desired by the source nodes. A node keeps using the created route until that route becomes inaccessible or the route is no longer needed.



a)



b)



c)

Figure 6.1. UDP end-to-end throughput in a 7-node flow using a) AODV, b) DSR and c) DSDV

The "re-routing instability problem", as mentioned in Chapter 2.4.1, is a common performance problem suffered by various ad-hoc routing protocols. Figure 6.1 show that AODV, DSR (reactive) and DSDV (proactive) all experience throughput oscillations. Although the severity of the oscillations may vary, they are caused by the same reason, the triggering of the re-routing function. These routing protocols treat the link-failure notification as an indication of the loss of the link to next hop. In IEEE 802.11, this link-failure notification can be induced by the hidden-terminal problem as well as the real-break case. Obviously, simply discarding the route after receiving a link-failure notification is not appropriate for IEEE 802.11 multi-hop networks.

6.2 Proposed scheme

A possible solution is to modify the routing algorithm so that the routing agent continues to use the previous route for transmissions before a new route can be found. In practice, this means computers equipped with wireless LAN devices only need to install slightly modified routing agent software. In this thesis, we choose the AODV routing protocol for implementation of this "don't-break-before-you-can-make" strategy, mainly because details of AODV have been published in an IETF RFC [RFC]. There is no reason why this approach can not be applied in other ad-hoc routing protocols.

6.2.1 Original AODV

We quote the following excerpt from the IETF RCF 3561 on AODV [RFC]: "Any suitable link layer notification, such as those provided by IEEE 802.11, can be used to determine connectivity, each time a packet is transmitted to an active next hop. For example, absence of a link layer ACK or failure to get a CTS after sending RTS, even after the maximum number of retransmission attempts, indicates loss of the link to this active next hop."

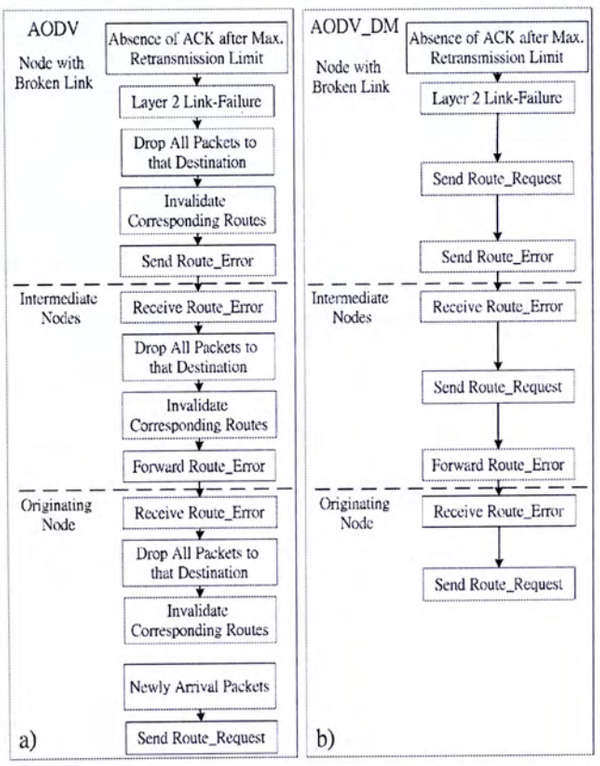


Figure 6.2. Procedures in handling link-failure in a) original AODV and b) our proposed scheme (AODV_DM)

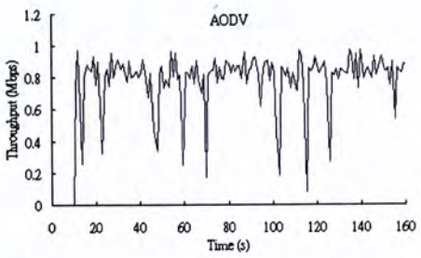


Figure 6.3. TCP end-to-end throughput in a 7-node flow using original AODV

Figure 6.2a shows the procedures for handling link-failure in the original AODV. When a node fails to receive the link-layer ACK from the next hop after the retransmission limit, its link layer reports the link failure to the routing agent. The AODV protocol then generates a list of unreachable destinations that use the unreachable neighbor as the next hop. It drops all packets destined to that hop and invalidates the corresponding routes in its routing table. Then the node with the broken link propagates the route error (RERR) message to its upstream neighbors until the source node is reached. When the source and intermediate nodes receive the RERR message, they also drop all packets that utilize the broken route for forwarding and are destined to the nodes in the unreachable destination list attached with the RERR message. The nodes then remove the corresponding routes form their routing tables. After that, a newly arrival packet targeted for these unreachable destinations will trigger the route discovery process, and the transmissions of packets to that destination will be resumed after the new route is generated.

6.2.2 AODV with Proposed Scheme

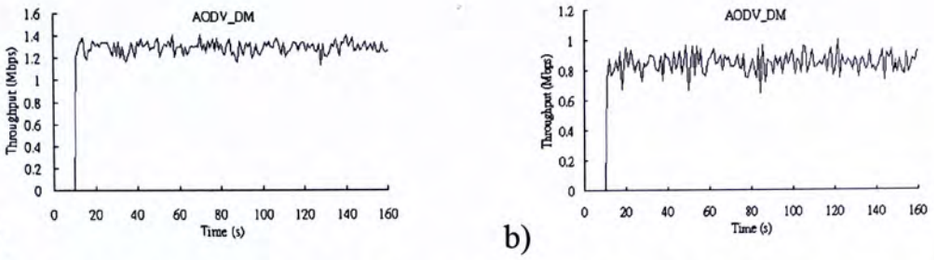
In our proposed solution as shown in Fig. 6.2b, the link layer notifies the routing agent of the "link failure" after the maximum retransmission attempts. The AODV routing agent then broadcasts a route request (RREQ) message immediately. Unlike the original AODV, our routing agent does not drop packets and invalidate the corresponding routes. However, it continues to propagate the RERR message to its

upstream neighbors. When an intermediate node receives the RERR message, it broadcasts another RREQ message and forwards the RERR message to upstream nodes until the source node is reached. During this process, no packets will be dropped and all nodes continue to use the previous routes. After sending RREQ messages, the nodes wait for the route reply (RREP) message returned by the destination node or an intermediate node with an up-to-date route (i.e., the destination sequence number stored in the node's routing table is greater than that in the RREQ message [RFC]). After a new route is created, all nodes discard the previous route and switch to the new one for transmissions.

In the following chapters, we will show simulation results of AODV modified with "don't-break-before-you-can- make" strategy (AODV_DM) in two scenarios: 1) a single flow in a single chain of nodes; and 2) a real-break case.

6.2.2.1 A Single Flow in a Single Chain of Nodes

Figures 6.1a and 6.3 show the existence of "re-routing instability" of UDP and TCP traffic in a 7-node chain using the original AODV. As shown in Fig. 6.4, the AODV_DM scheme eliminates these oscillations. With the AODV_DM scheme, no packets are dropped and nodes continue to use the old route, while the new route discovery process is ongoing. For our scenario of a single-chain network, when the node with the broken link receives the responded RREP message or the Hello message broadcasted periodically by the next hop, it notices that the next hop is still active and the routing agent will re-discover the same route for transmissions.



a) b)

Figure 6.4. a) UDP and b) TCP end-to-end throughput in a 7-node flow using AODV_DM

6.2.2.2 Real-break Case

Figure 6.5 shows a scenario with two alternative routes from node 1 to node 7. Both of them are accessible in the first 70 seconds. At the 70th second, node 4 is switched off and this breaks the upper route. Figures 6.6 and 6.7 show the simulation results. In the first 70 seconds, both the original AODV and AODV_DM choose the upper route since this path requires fewer number of hops. After the 70th second, they switch to the lower route for transmissions. Since the number of hops in the lower route is more than that of the upper route, the average throughputs are slightly reduced. Our proposed scheme keeps the route discovery property of original AODV and switch to a new route if the existing one is broken. At the same time, AODV_DM eliminates the "re-routing instability problem" experienced by the original AODV.

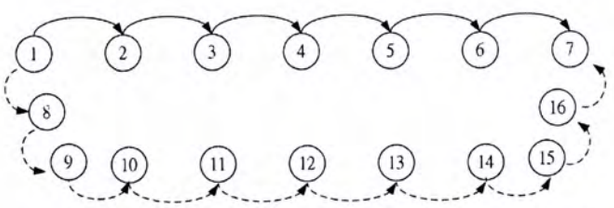
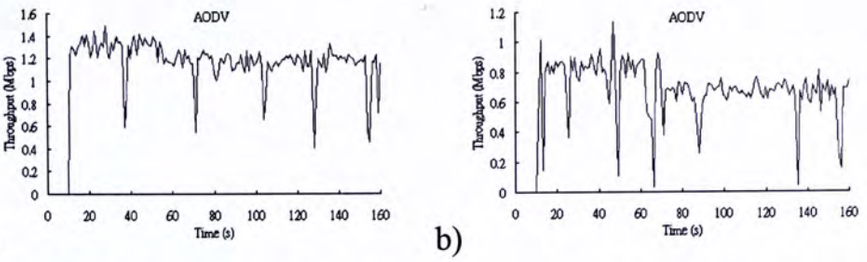
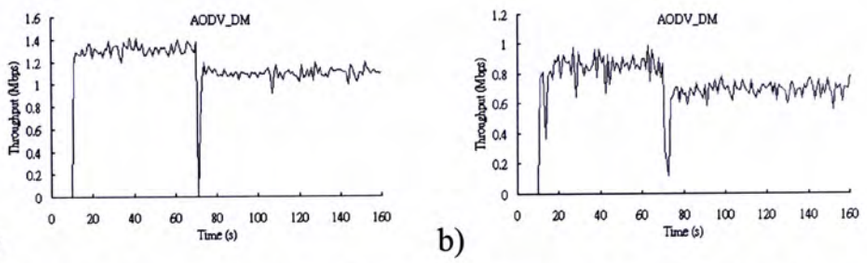


Figure 6.5. Two alternative routes for UDP/TCP traffic flow with node 1 as the source and node 7 as the destination in a multi-hop network



a) b)

Figure 6.6. a) UDP and b) TCP end-to-end throughput in a real-break case using original AODV



a) b)

Figure 6.7. a) UDP and b) TCP end-to-end throughput in a real-break case using AODV_DM

6.3 Improvements

Simulations show that whenever re-routing occurs, the throughput drops severely for the duration of 1 to 3 seconds. For real-time applications like video conferencing or voice over IP (VoIP), this may not be acceptable. Compared with the original AODV, our proposed solution reduces the throughput variations by 70% for UDP and 50% for TCP as shown in Fig. 6.8. Also, from Table 6.1, the minimum throughputs of the original AODV are near zero when there are more than five nodes in the UDP flow; and when there are more than three nodes in the TCP flow. Using AODV_DM, the minimum throughputs are only slightly less than the average values. As shown in Fig. 6.9, another improvement of our proposed scheme is to boost the average throughput up to 11% for both TCP and UDP in a long chain of nodes (i.e., more than 12 nodes).

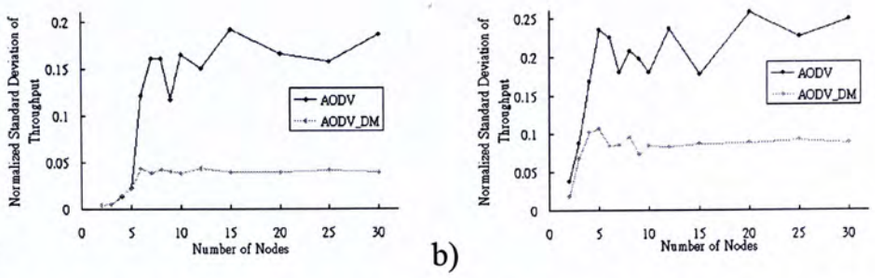


Figure 6.8. Normalized standard deviation of a) UDP and b) TCP end-to-end throughput versus the number of nodes in a string multi-hop network

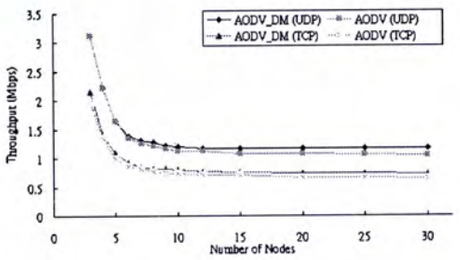


Figure 6.9. UDP and TCP end-to-end throughput versus number of nodes in a string topology

Table 6.1. a) UDP and b) TCP throughput result (Mbps) with various number of nodes in a string multi-hop network using AODV and AODV_DM in a 500-second simulation run

a)

Num. of Nodes	AODV			AODV_DM		
	Mean	Max	Min	Mean	Max	Min
2	6.304	6.389	6.237	6.303	6.366	6.225
3	3.120	3.165	3.084	3.118	3.154	3.084
4	2.213	2.301	2.114	2.213	2.336	2.102
5	1.646	1.775	1.565	1.646	1.764	1.553
6	1.354	1.542	*0.350*	1.391	1.530	1.226
8	1.211	1.448	*0.245*	1.276	1.448	1.110
10	1.131	1.320	*0.199*	1.197	1.320	1.040
15	1.074	1.261	*0.070*	1.170	1.332	1.016
20	1.080	1.261	*0.070*	1.166	1.285	0.958
30	1.049	1.238	*0.093*	1.171	1.296	0.993

b)

Num. of Nodes	AODV			AODV_DM		
	Mean	Max	Min	Mean	Max	Min
2	4.231	4.659	3.746	4.341	4.560	4.117
3	1.969	2.405	1.521	2.155	2.571	1.758
4	1.359	1.946	*0.194*	1.403	1.994	0.999
5	1.002	1.457	*0.000*	1.087	1.525	0.712
6	0.867	1.101	*0.000*	0.933	1.151	0.652
8	0.766	1.098	*0.000*	0.819	1.030	0.486
10	0.742	1.029	*0.000*	0.799	1.012	0.578
15	0.710	0.976	*0.025*	0.762	0.968	0.544
20	0.671	0.952	*0.000*	0.742	0.931	0.539
30	0.649	0.811	*0.000*	0.720	0.989	0.534

Chapter 7 Impacts of Data Transmission Rate and Payload Size

This chapter shows the effects of the data transmission rate and payload size on the re-routing instability problem. We first show the condition for the occurrence of hidden-terminal collisions. Then we introduce a quantitative approach to analyze the impact of various data transmission rates and payload sizes.

7.1 Signal Capture

The treatment in this chapter is similar to that of Chapter 3.1.1. Instead of referring the reader back to Chapter 3.1.1 for the needed materials, we choose to present a self-contained treatment here for ease of reading, at the expense of repeating some materials in Chapter 3.1.1. Consider Fig. 7.1 again, both nodes 3 and 6 have a packet to transmit. This may cause the aforementioned hidden-terminal collision. However, the signal capturing property may still allow a packet from node 3 to be received successfully, provided it transmits before node 6.

More specifically, suppose that node 3 transmits first and the signal power of the transmission received at node 4 is P_3 . Node 6 then transmits a packet with power P_6 received at node 4. If $P_3 > P_6 + CPThreshold$, where *CPThreshold* is the capture threshold,

then no collision occurs, and node 4 can still receive the packet from node 3 successfully.

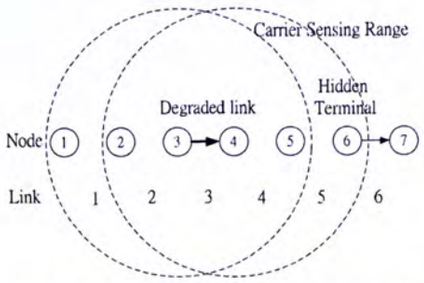


Figure 7.1. Node 6 as a hidden terminal to node 3

On the other hand, if node 6 transmits first, node 4 senses the signal from node 6 and declares the channel to be busy. In that case, a newly arriving packet from node 3 can not be received even if $P_3 > P_6 + CPThreshold$. Effectively, the packet from node 3 to node 4 experiences a collision.

In our simulation, *CPThreshold* is set to be 10dB. Let d be the fixed distance between nodes. In this scenario, node 3 and node 6 are separated by a distance larger than the carrier sensing range. Thus, node 3 and node 6 can send packets at the same time. From [TR], in a two ray propagation model, the signal-to-noise ratio at node 4 is

$$SNR = P_3 / P_6 = (2d/d)^4 = 2^4 = 16 > CPThreshold \quad (11)$$

This means that the power level of the packet transmitted by node 3 and received at node 4 is always more than *CPThreshold* higher than the power level of the received signal from node 6.

7.2 Vulnerable region

In the analysis of the effect of the hidden-terminal problem, the key is to identify the vulnerable region during which if the node transmits, it may collide with the transmission of a hidden node. This is illustrated in Fig. 7.2. Note that a hidden-node collision only occurs if the transmissions of nodes 3 and 6 overlap and that the transmission of node 6 precedes that of node 3. Let $PACKET_i$ be the time to transmit packet *i*.

$$PACKET_i = PHY + (MAC + Payload)/TxRate \quad (12)$$

where *PHY* is the time to transmit the physical header, *MAC* is the size of the MAC header, *Payload* is the size of the packet payload, and *TxRate* is the data transmission rate. Let T_i be the time of the transmission cycle of packet i at node 6. As illustrated in Fig. 7.2, T_i includes the back-off period, the packet transmission time, the idle period, I_i,when node 6 does not have a packet to transmit, and the busy periods used by other nodes within its carrier sensing range for their transmissions, B_i. We have

$$T_i = I_i + DIFS + W_{avg} + PACKET_i + SIFS + ACK + B_i \quad (13)$$

Let ρ be the fraction of the time corresponding to the vulnerable region induced by node 6. We have

$$\rho = \lim_{K\to\infty} \frac{\sum_{i=1}^{K} PACKET_i}{\sum_{i=1}^{K} T_i} \quad (14)$$

where *ACK* is the transmission time for an acknowledgement, *SIFS* is the time duration of short interframe space, *DIFS* is the time duration of distributed interframe space, and W_{avg} is the average contention window size. Thus, ρ varies with different data transmission rates and payload sizes. With lower data transmission rate or larger payload size, the fraction of the time that belongs to vulnerable region in each transmission cycle becomes larger. As a result, a higher chance of hidden-terminal collisions is expected. In other words, the link-failure re-routing occurs more frequently which further deteriorates the instability problem. As shown in Fig. 7.3 and 7.4, using lower data transmission rate or larger payload size increases the number of severe drops of throughputs.

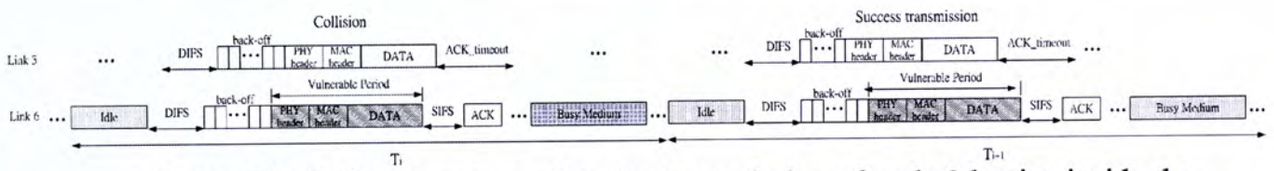


Figure 7.2. Collision occurs when the transmission of node 3 begins inside the vulnerable period

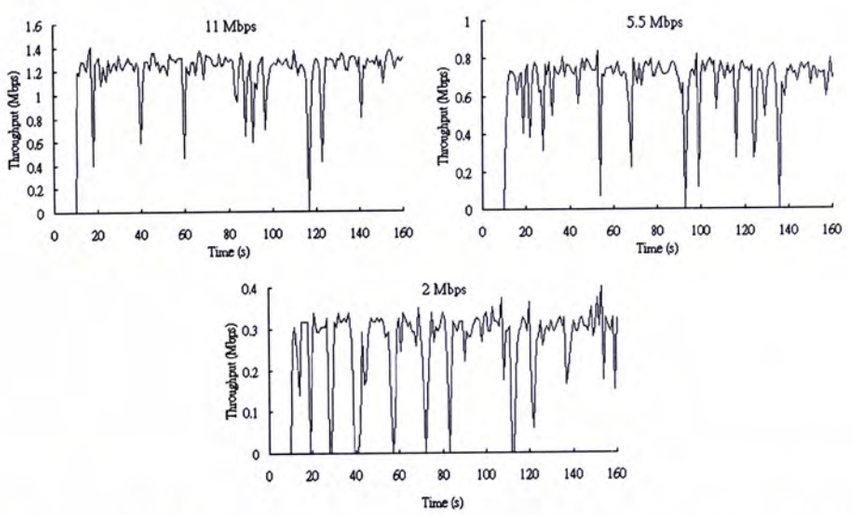


Figure 7.3. UDP end-to-end throughput in a 7-node flow using original AODV with various data transmission rates

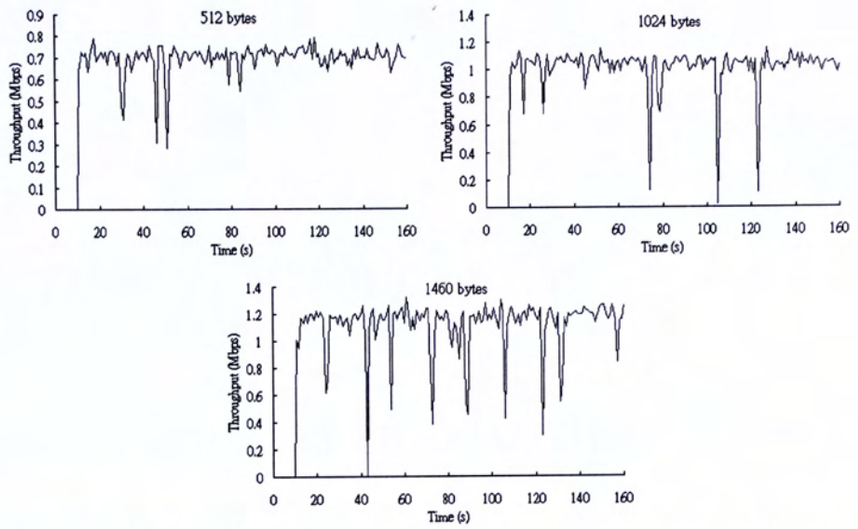


Figure 7.4. UDP end-to-end throughput in a 7-node flow using original AODV with various payload sizes

Chapter 8 Performance Enhancements in Multiple Flows

In previous chapters, we have focused on the performance degradations induced by self interference of single-flow traffic. In this chapter, we consider the interferences between multiple flows. First, we use a two flow scenario to demonstrate that the severe throughput degradation due to the hidden-terminal problem is mainly caused by the "re-routing instability problem" rather than the "binary exponential back-off". Then we consider more complicated scenarios with multiple hidden-terminal flows. We identify the factors that affect the impact of hidden-terminal flows. Most importantly, we show that our proposed scheme can substantially increase the average throughput of a flow suffering from the hidden terminal problem in all scenarios.

Figure 8.1. Two 1-hop saturated UDP flows

8.1 Impacts of Re-routing Instability in Two Flow Topology

Figure 8.1 shows a scenario with five nodes and two 1-hop saturated UDP traffic flows.

As mentioned in Chapter 2.3.1, the transmissions of flow 1 may collide with the transmissions of flow 2 at node 2 due to the hidden-terminal problem. This severely deteriorates the throughput of flow 1, while flow 2 continues to achieve a much higher throughput as demonstrated in Fig. 8.2a. In addition, the throughput of flow 1 drops to zero from 70^{th} to 110^{th} second due to the successive collisions of RREQ sent out by node 1 with the transmissions of flow 2. Node 4 does not notice that node 2 is suffering from hidden-terminal collisions and attempts to transmit at the maximum sustainable rate. Once the link at node 1 is declared as failure, node 1 sends out RREQ and waits for RREP. However, this RREQ message easily collides with the aggressive transmissions of flow 2. In this way, no RREP is responded by node 2, and node 1 times out and retransmit another RREQ. No packet can be transmitted for a long period of time after a number of failed RREQ transmissions.

Previous work in the literature [HB] also reported that the throughput can degrade severely in similar scenarios. They attribute this degradation to the binary exponential back-off for retransmissions caused by hidden nodes. However, we believe it is only part of the cause. Once a node fails to receive the link-layer ACK after the retry limit, it triggers the re-routing function of the routing agent. Before a new route or the previous route is discovered, no packets can be transmitted. This "re-routing instability problem" and the "binary exponential back-off" should be treated and solved separately.

Our proposed scheme addresses the first issue. The average throughput of flow 1 is doubled as show in Fig. 8.2b. The "binary exponential back-off" does degrade the throughput, resulting in average throughput of flow 1 slightly less than that of flow 2. However, its influence is much smaller than that of "re-routing instability problem". To limit the scope of this thesis, we refer interested readers to [HB], in which MAC layer solutions were proposed to address the degradations caused binary exponential back-off.

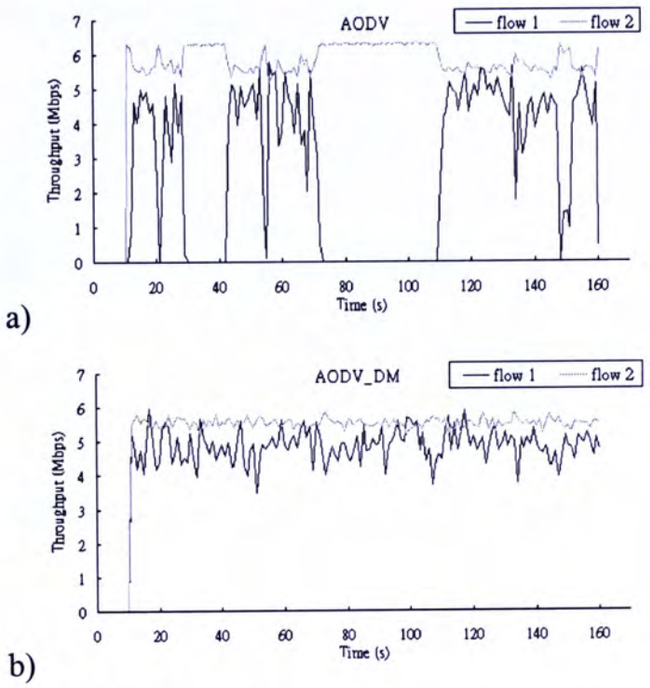


Figure 8.2. UDP throughputs of two 1-hop flows using a) original AODV and b) AODV_DM

8.2 Impacts of Vulnerable Periods in Multiple Flow Topologies

In Chapter 7, we have shown that a hidden-terminal flow with lower data transmission rate and larger payload size can induce larger vulnerable period, which in turn increases the chance of hidden-terminal collisions. To investigate the impact of multiple hidden-terminal flows, we have to consider the overall vulnerable period induced on the suffering flow.

Consider a long stretch of time in the interval [0, *Time*]. Let S_i be the airtime within this interval that hidden-node i transmits. This airtime includes the transmission times of the data packets (PACKET), the transmission times of the acknowledgements (ACK)

from node (i+1), the durations of the distributed interframe space (DIFS) and the durations of the short interframe space (SIFS). Also, included in Si are the times used up for retransmissions in case of collisions. However, Si does not include the count-down of the idle slots of the contention window, since adjacent nodes can count down together and these count-down times are not unshared resources used up exclusively by node i. Let

$$x = \frac{|S_i \cup S_{i+1} \cup S_{i+2} \cup \ldots|}{Time} \qquad (15)$$

The fraction of time when the suffering flow is vulnerable to hidden-terminal collisions is then

$$\rho = x \cdot a \qquad (16)$$

where $a = (PACKET)/(DIFS + PACKET + SIFS + ACK)$ is fraction of time used for transmitting the data packet during the airtime used by the hidden terminals.

The size of the overall vulnerable period can be determined by three factors: 1) the vulnerable periods induced by individual hidden-terminal flows; 2) the number of hidden-terminal flows; 3) the correlations between hidden-terminal flows. In the following chapters, we express the impacts of these three factors in term of an overall vulnerable period and demonstrate that our proposed scheme can obtain significant improvements even in multiple hidden-terminal scenarios.

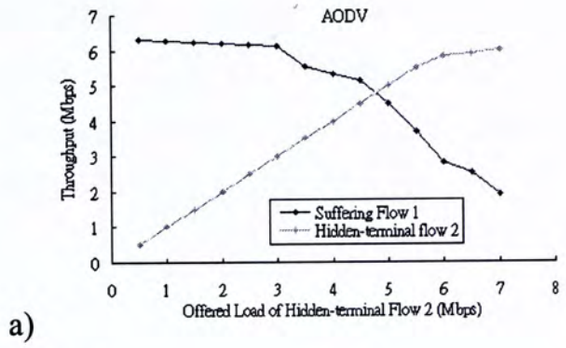
Figure 8.3. Two 1-hop UDP flows

8.2.1 The Vulnerable Period induced by Individual Hidden-terminal Flow

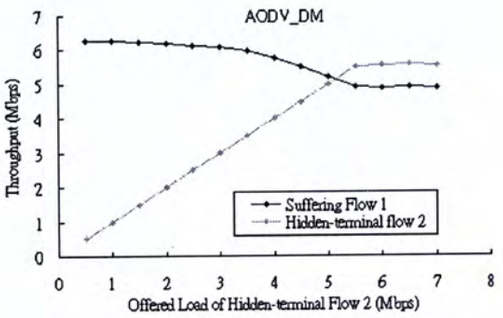
The vulnerable period induced by a hidden-terminal flow depends on the throughput (l_i) of that flow. Thus,

$$x = \frac{|S_i(l_i) \cup S_{i+1}(l_{i+1}) \cup S_{i+2}(l_{i+2}) \cup \ldots|}{Time} \qquad (17)$$

A hidden-terminal flow with higher throughput utilizes a larger fraction of airtime for transmitting packets, and this leads to larger vulnerable period. Figure 8.3 shows a scenario with the suffering flow (flow 1) associated with a saturated traffic source, and the hidden-terminal flow (flow 2) associated with a traffic source with a variable offered load. Figure 8.4a and 8.4b plot the throughputs of flows 1 and 2 against the offered load of flow 2 when the original AODV and AODV_DM are used respectively. As shown in both figures, the throughput of flow 2 increases with the offered load until the maximum network capacity is reached. Meanwhile, the throughput of flow 1 decreases. This is because a larger throughput of flow 2 leads to a larger vulnerable period to flow 1 which makes hidden-terminal collisions more likely. In other words, the link-failure re-routing happens more frequently and this deteriorates the throughputs of flow 1. However, our proposed scheme can prevent the link-breakage triggered by the re-routing function and thus the throughput of flow 1 can still be maintained at a comparatively high level (see Fig. 8.4b) even under the influence of a hidden-terminal flow transmitting at maximum network capacity.



a)



b)

Figure 8.4. UDP throughputs of the suffering flow and the hidden-terminal flow using a) original AODV and b) AODV_DM

8.2.2 The Number of Hidden-terminal Flows

Figure 8.5 shows a scenario with multiple hidden-terminal flows with saturated traffic within the carrier-sensing range of each other. In this case, the hidden-terminal flows have to take turn to transmit and thus they must share the network capacity. Figure 8.6 shows the throughputs of the suffering flow (flow 1) decreases with the number of hidden-terminal flows. Flows within the same carrier-sensing range can share the time for contention window countdown. This increases x and the fraction of time that contributes to vulnerable periods. As a result, more hidden-terminal collisions are expected.

Besides the above scenario, it is also possible for some hidden-terminal flows to be outside the carrier-sensing range of each other. In that case, the throughputs of

hidden-terminal flows will depend on the network topology. The analysis of the throughputs of flows under various network topologies is outside the scope of this thesis. A possible analytical method is to use the quantitative analysis for the sustainable throughput of a string topology as shown in Chapters 3 to 5. That work may act as a building block for determining throughputs in more complicated scenarios.

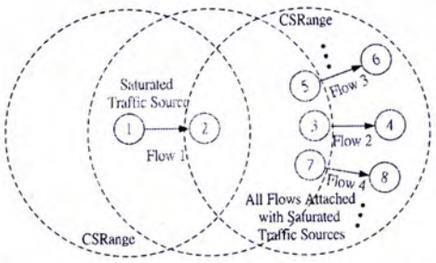


Figure 8.5. Multiple hidden-terminal flows and a suffering flow with saturated UDP traffic sources.

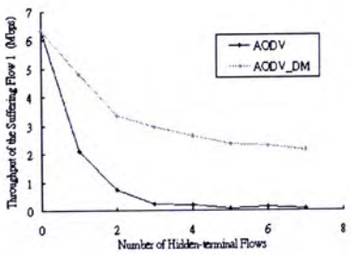


Figure 8.6. UDP throughputs of the suffering flow using original AODV and AODV_DM

For simplicity, we only consider the impact of the number hidden-terminal flows within the same carrier-sensing range in this thesis. The main observation is that our proposed scheme can maintain the throughput of the suffering flow (flow 1) at 2.1Mbps under the influence of seven hidden-terminal flows. On the other hand, using the original AODV, the throughput drops nearly to zero when there are only three hidden-terminal flows.

8.2.3 Correlation between Hidden-terminal Flows

In the scenario with all hidden-terminal flows inside a carrier-sensing range, all of them must take turn to transmit and thus the airtimes used by them are exclusive to each other. We call them correlated flows. When some of the hidden-terminal flows are outside the carrier-sensing range of others, they can transmit simultaneously and thus their airtimes can overlap with each other. This reduces the overall vulnerable period. We call them independent flows. With the same number of hidden-terminal flows and all flows induce the same size of vulnerable period. Exclusive correlated hidden-terminal flows are expected to induce a larger overall vulnerable period than independent hidden-terminal flows and thus results in a lower throughput of the suffering flow.

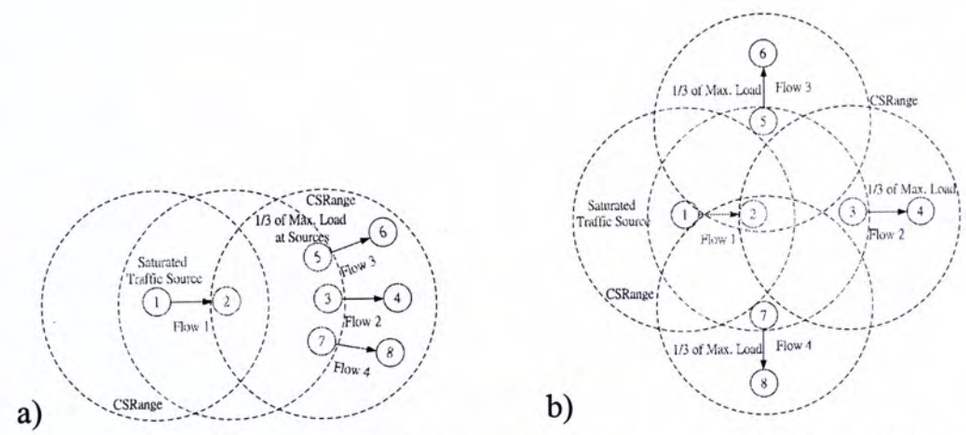


Figure 8.7. Three a) correlated or b) independent hidden-terminal flows with one-third of maximum offered load and a saturated suffering UDP flow

For example, Figures 8.7a and 8.7b show examples for three correlated and independent hidden-terminal flows respectively. For fair comparisons, all hidden-terminal flows have the same offered load (one-third of the network capacity) which induce the same size of vulnerable region, $\frac{|S|}{Time} \approx \frac{1}{3}$. Then, for correlated flows,

$$x = \frac{|S_1 \cup S_2 \cup S_3|}{Time} = \frac{3|S|}{Time} \approx 1 \qquad (18)$$

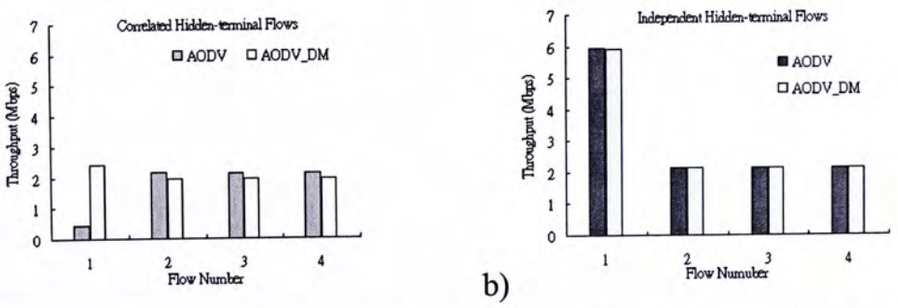
For independent flows, using the inclusion-exclusion principle,

$$x = \frac{|S_1 \cup S_2 \cup S_3|}{Time}$$

$$= \frac{|S_1| + |S_2| + |S_3| - |S_1 \cap S_2| - |S_1 \cap S_3| - |S_2 \cap S_3| + |S_1 \cap S_2 \cap S_3|}{Time}$$

$$= \frac{3|S| - 3|S|^2 + |S|^3}{Time} \approx \frac{19}{27} < (15) \qquad (19)$$

From equation (19), correlated hidden-terminal flows in Fig. 8.7a induce a larger overall vulnerable period than independent hidden-terminal flows in Fig. 8.7b. The throughput of flow 1 in Fig. 8.8b is thus higher than that in Fig. 8.8a because independent hidden-terminal flows allow overlapping of vulnerable periods and thus reduces the size of the overall vulnerable period. This reduces the chance of hidden-terminal collisions and the triggering of the re-routing function. As a result, flow 1 in Fig. 8.7b can achieve a much higher throughput than that in Fig. 8.7a. On the other hand, the larger overall vulnerable period induced by flows in Fig. 8.7a leads to a higher collision probability and more frequent re-routing instability. This degrades the throughput of flow 1 to 0.4Mbps. However, our proposed DM scheme can prevent re-routing instability and boosts the throughput of flow 1 to 2.4Mbps as shown in Fig. 8.8a.



a) b)

Figure 8.8. Throughputs of the suffering flow and three a) correlated or b) independent

hidden-terminal flows using original AODV and AODV_DM

Tables 8.1 and 8.2 summarize the performance improvements obtained by our proposed scheme under various network topologies considered in this thesis. In all cases, our proposed scheme (AODV_DM) improves the average throughput up to ten times and reduces the throughput variation by more than 60%.

Scenarios		Throughput of Suffering Flow (Mbps)		Throughput Improvement
		AODV	_DM	
Single Flow	30 nodes with TCP source	0.65	0.72	4.3%
	30 nodes with UDP source	1.05	1.17	16 %
Multiple Flow	2 flows with saturated sources (Fig. 8.3)	1.94	4.89	1.5 times
	3 correlated saturated hidden-terminals (Fig. 8.5)	0.42	2.40	5 times
	7 saturated hidden-terminal flows (Fig. 8.5)	0.08	2.13	26 times

Table 8.1. A summary of throughput improvements achieved by AODV_DM in various network topologies

Scenarios		Normalized Standard Deviation of Throughput		Reduction
		AODV	_DM	
Single Flow	30 nodes with TCP source	0.25	0.09	64.8 %
	30 nodes with UDP source	0.19	0.04	78.7 %
Multiple Flow	2 flows with saturated sources (Fig. 8.3)	1.11	0.09	92.4 %
	3 correlated saturated hidden-terminals (Fig. 8.5)	2.47	0.16	93.6 %
	7 saturated hidden-terminal flows (Fig. 8.5)	2.28	0.21	92.0 %

Table 8.2. A summary of throughput variation reductions achieved by AODV_DM in various network topologies

Chapter 9 Conclusion

This thesis attempts to 1) identify the maximum sustainable throughput and 2) solve the throughput instability problem in IEEE 802.11 multi-hop network.

For 1), we believe that this is a first paper in the literature to provide a *quantitative analysis* on the fundamental impact of hidden nodes and carrier sensing on system throughput. Our contributions are three-folds:

a. We have shown that uncontrolled, greedy sources can cause unacceptably high packet-loss rate, large throughput oscillations, and unfair bandwidth allocations among traffic flows. Judicious offered load control at the sources, however, can eliminate these problems effectively without modification of the 802.11 multi-access protocol. Our simulations and real-network experiments have confirmed the existence of this optimal offered load in a 6-node multi-hop network.

b. We have established an analytical framework for the study of the effects of hidden nodes and carrier-sensing operation. This analysis allows one to determine whether the system throughput is hidden-node limited or spatial-reuse limited. In particular, we have shown that the maximum sustainable throughput is limited by two factors: (i) the vulnerable periods which depend on the numbers of hidden nodes and the fraction of airtime in the time horizon when hidden-node collisions

may occur; (ii) the number of nodes within a carrier-sensing region and the total airtime used up by them.

c. We have studied the single-flow case in detail. The throughput limitation of a single multi-hop flow is typically dominated by the hidden-node effect of (i). However, a modification on the receiver design can eliminate the hidden-node effect so that the throughput is limited by (ii) instead. Throughput improvement as high as 50% is possible.

The single-flow analysis in this thesis serves as a "building block" for the study of the multiple-flow case, in which besides self-interference induced by traffic of the same flow, there are also mutual interferences among traffic of different flows. By way of an example, we have shown how to apply the single-flow result to control the offered loads of multiple non-overlapping flows in a lattice network. More complicated situations with overlapping multiple flows remain to be further investigated. We believe the approach in this thesis provides a good foundation for such an extension.

For 2), existing ad-hoc routing protocols simply inherit the method for link-failure handling from the routing protocols used in wired networks, and treat the link-failure notification as an indication of the loss of the link to the next hop. This is not appropriate for wireless networks with hidden-terminal problems such as IEEE 802.11. The triggering of the re-routing function may be induced by consecutive hidden-terminal collisions rather than real link failures. This thesis has four major contributions:

a. We have argued that the throughput instability problem should properly be re-defined as a "re-routing instability problem", since it is caused by the triggering of the re-routing function and is not specific to TCP traffic alone.

b. We have proposed to adopt a "don't-break-before-you-can-make" modification to the existing ad-hoc routing protocols. In this strategy, the old route will continue to be used until a new one can be established. We have implemented this scheme with AODV as an example, and have shown that the instability problem can be eliminated. The modified routing agent can still switch to a new route successfully in a real-break case.

c. We have analyzed the hidden-terminal problem by considering the "vulnerable regions: the time windows during which transmissions may collide with transmission of hidden node". We have established the impact of data transmission rate and payload size on the severity of hidden-node collisions. In particular, we have shown that lower data transmission rates and/or larger payload sizes will incur more frequent throughput oscillations. In multiple hidden-terminal cases, we have shown that higher individual throughputs, larger number of hidden-terminals and/or correlated flows will induce larger vulnerable regions which further degrade the throughput of the suffering flow. Most importantly, our proposed scheme can significantly reduce throughput variations and increase average throughputs in all kinds of scenarios.

d. This thesis has also investigated a multiple-flow scenario. The throughput degradation induced by "re-routing instability" is much larger than that induced by "binary exponential back-off", as has been demonstrated by the restoration of UDP throughput when our "don't-break- before-you-can-make' ad-hoc routing protocol is used. We believe that this is the first paper in the literature to report this phenomenon.

Finally, we believe the offered load control and our proposed modifications on ad-hoc routing protocol (i.e., the "don't-break-before-you-can-make" strategy) are attractive solutions to eliminate high packet-loss rate, re-routing instability and unfairness

problems in IEEE 802.11 Multi-hop Ad-hoc Networks.

Appendix A: General Throughput Analysis of a Single Multi-hop Traffic Flow

Let k be the number of nodes within a carrier-sensing range (CSRange, i.e., 550m) and let l be the uniform distance between two successive nodes. Figure 3.8 illustrates a string network topology with variables k and l.

A.1 Capacity Limited by Hidden-node and Exposed-Node

Following similar approaches in deriving the vulnerable period induced by hidden-node as shown in Chapter 3.1.2, we can express ρ_{HT} in term of x. In Fig. 3.8, when node $i+1$ to $i+k$ transmit, node i and node $i+k+1$ will not. This means that S_i to S_{i+k} are non-overlapping; and S_{i+1} to S_{i+k+1} are non-overlapping. In particular, node $i+k+1$ cannot cause collision on node i during S_{i+1} to S_{i+k}. Now, nodes $i+1$ to $i+k$ use up $k \cdot x$ fraction of the airtime during [0, *Time*]. The remaining fraction of airtime where node i and node $i+k+1$ may collide is (1- $k \cdot x$). Since node $i+k+1$ uses x fraction of remaining airtime for transmissions, the vulnerable period induced by node $i+k+1$ on node i is

$$\rho_{HT} = \frac{x}{1-k\cdot x}\cdot a \qquad (20)$$

Again, as explained in Chapter 3.1.3, the ACK-ACK collision can only occur if the transmission of node i begins at time $t <$ SIFS later than the transmission of node $i-k-1$. Therefore, the ACK-ACK collision rarely happens. Thus we assume that the degradation caused by exposed nodes is negligible in our analysis.

A.1.1 Sustainable Throughput

Substituting equations (20) and (4) in (1), we have

$$T = x\cdot(1-a\cdot\frac{x}{1-k\cdot x})\cdot d\cdot data\_rate \qquad (21)$$

Differentiating (21) with respect to x and setting $dT/dx = 0$, the optimal value of x that maximizes the throughput is given by

$$x^* = \frac{(k+a)-\sqrt{a^2+ka}}{k^2+ka} \qquad (22)$$

Substituting equation (22) in (21) yields the maximum sustainable throughput $T(x^*)$.

A.2 Capacity Limited by Carrier Sensing Property

Carrier sensing prevents simultaneous transmissions of nodes within the carrier-sensing range of a node. Consider node i as the local observer and nodes within its carrier-sensing range in Fig. 3.8. The total airtimes used up by these nodes cannot exceed *Time*. That is,

$$|C_i \cup S_{i-k} \cup S_{i-k+1} \cup ...S_i \cup ...S_{i+k}| \leq Time$$

Define $y = |C_i \cup S_{i-k} \cup S_{i-k+1} \cup ...S_i \cup ...S_{i+k}| / Time$, to be the fraction of airtime used up by these nodes within the interval [0, *Time*]. Now, $|C_i \cup S_{i-k} \cup S_{i-k+1} \cup ...S_i \cup ...S_{i+k}|$ can be

decomposed using the inclusion-exclusion principle:

$$|C_i \cup S_{i-k} \cup S_{i-k+1} \cup ... S_i \cup ... S_{i+k}| = |C_i| + |S_{i-k}| + |S_{i-k+1}| + ... |S_{i+k}|$$
$$- |C_i \cap S_{i-k}| - |S_{i-k} \cap S_{i-k+1}| - |S_{i-k} \cap S_{i-k+2}| - ...$$
$$... + |C_i \cap S_{i-k} \cap S_{i-k+1}| + |S_{i-k} \cap S_{i-k+1} \cap S_{i-k+2}| + ... \quad (23)$$

However, we note that the intersection of the airtimes used by any three nodes or above is null, thanks to carrier sensing. Also, node *i* can count down only if nodes *i-k*, *i-k+1* ..., *i+k-1* and *i+k* are not transmitting, thus $C4 \cap Si$ is null. In addition, the intersections of airtimes used by two nodes are non-null only for $|S_j \cap S_{j+m}|$ for any node *j* where $m \geq k+1$.

We therefore have

$$y \cdot Time = |C_i| + \sum_{j=i-k}^{i+k} |S_j| - \sum_{j=i-k}^{i-1} |S_j \cap S_{j+k+1}| - \sum_{j=i-k}^{i-2} |S_j \cap S_{j+k+2}| - - \sum_{j=i-k}^{i-3} |S_j \cap S_{j+k+3}| - ... \quad (24)$$

Consider the overlapped airtimes of node $i-k$ and node $i+1$. When node $i-k+1$ to i transmits, node $i-k$ and $i+1$ do not, by virtue of carrier sensing. The remaining fraction of airtime where S_{i-k} and S_{i+1} may overlap is $(1-k \cdot x - c \cdot x)$. In particular, we have

$$|S_{i-k} \cap S_{i+1}| = |S_{i-k+1} \cap S_{i+2}| = \frac{x^2}{1-(k+c) \cdot x} \cdot Time \quad (25)$$

Nodes $i-k+1$ and $i+2$ face the same situation. Hence, $|S_{i-k} \cap S_{i+1}| = |S_{i-k+1} \cap S_{i+2}|$ in (25).

For $|S_{i-k} \cap S_{i+2}|$, the amount of airtime of node $i-k$ that may overlap with that of node $i+2$ is $(|S_{i-k}| - |S_{i-k} \cap S_{i+1}|)$, and the amount of airtime of node $i+2$ that may overlap with that of node $i-k$ is $(|S_{i+2}| - |S_{i-k+1} \cap S_{i+2}|)$. The "sample space" within which S_{i-k} and S_{i+2} may overlap is [0, *Time*] $-S_{i-k+1} - S_{i-k+2}$ -...$-S_{i+1} - C_i$. As a result, we have

$$|S_{i-k} \cap S_{i+2}| = \frac{(|S_{i-k}| - |S_{i-k} \cap S_{i+1}|) \cdot (|S_{i+2}| - |S_{i-k+1} \cap S_{i+2}|)}{Time - |S_{i-k+1}| - |S_{i-k+2}| - ... - |S_{i+1}| - |C_i|} = \frac{(x - x^2/(1-k \cdot x))^2}{1-(k+1+c)x} \cdot Time \quad (26)$$

Let $D_m \cdot Time = |S_j \cap S_{j+m}|$. If node $j+m$ is within the carrier-sensing range of node j or vice versa, their airtime cannot overlap due to the carrier-sensing mechanism. Thus,

$D_m \cdot Time = |S_j \cap S_{j+m}| = 0$ if $m \le k$

For $m > k$, following similar approaches as with equations (25) and (26), we have,

$$D_{k+1} = |S_j \cap S_{j+k+1}| / Time = \frac{x^2}{1-(k+c)x} \qquad D_{k+2} = |S_j \cap S_{j+k+2}| / Time = \frac{(x-D_{k+1})^2}{1-(k+1+c)x}$$

$$D_{k+3} = |S_j \cap S_{j+k+3}| / Time = \frac{(x-D_{k+1}-D_{k+2})^2}{1-(k+2+c)x+D_{k+1}}$$

$$D_{k+4} = |S_j \cap S_{j+k+4}| / Time = \frac{(x-D_{k+1}-D_{k+2}-D_{k+3})^2}{1-(k+3+c)x+2D_{k+1}+D_{k+2}}$$

$$D_{k+n} = |S_j \cap S_{j+k+n}| / Time$$

$$= \frac{(x-D_{k+1}-D_{k+2}-D_{k+3}\cdots-D_{k+n-1})^2}{1-(k+n-1+c)x+(n-2)D_{k+1}+(n-3)D_{k+2}\cdots+D_{k+n-2}}$$

where $k+n \le 2k$, thus $n \le k$

So, in general,

$$D_{k+n} = |S_j \cap S_{j+k+n}| / Time$$

$$= \frac{(x-\sum_{m=1}^{n-1} D_{k+m})^2}{1-(k+n-1+c)x+\sum_{m=2}^{n-2} mD_{k+n-m-1}} \qquad (27)$$

Substituting into (27) into (24),

$$y = (2k+1+c)x - k \cdot D_{k+1} - (k-1) \cdot D_{k+2} - (k-2) \cdot D_{k+3} - \cdots$$

$$y = (2k+1+c)x - \sum_{i=1}^{k} (k-i+1) \cdot D_{k+i} \qquad (28)$$

The value of x for $y > 1$ is an "infeasible region". Again, let the x at which $y(x) = 1$ be x'. If the throughput obtained from x' is greater than the throughput obtained from x^* in equation (22), then the system throughput is limited by hidden nodes. However, if the other way round, the system is limited by the carrier-sensing mechanism.

Bibliography

[GK] P. Gupta, P. R. Kumar, "The Capacity of Wireless Networks", *IEEE Trans. Inform. Theory*, Vol.46, No.2, pp.388-404, Mar. 2000.

[LB] J. Li, C. Blake et al., "Capacity of Ad Hoc Wireless Networks", *ACM MobiCom'01*, Rome, Italy, July 2001.

[KJ] K. Jain et al. "Impact of Interference on Multi-hop Wireless Network Performance", *ACM MobiCom'03*, San Diego, USA, Sept. 2003.

[KN] M. Kodialam, T. Nandagopal, "Characterizing Achievable Rates in Multi-hop Wireless Networks: The Joint Routing and Scheduling Problem", *ACM MobiCom'03*, San Diego, USA, Sept. 2003

[XG] K. Xu, M. Gerla, S. Bae, "How Effective is the IEEE 802.11 RTS/CTS Handshake in Ad Hoc Networks?", *IEEE GLOBECOM '02*, Vol. 1 , pp. 17-21, Nov. 2002.

[SA] S. Ansari et al. "Performance Enhancement of TCP on Multihop Ad hoc Wireless Networks", *IEEE ICPWC'02*, pp. 90-94, Dec. 2002.

[HG] Z. Hadzi-Velkov, L. Gavrilovska, "Performance of the IEEE 802.11 Wireless LANs under Influence of Hidden", *IEEE PWCS'99*, pp. 221-225, Feb. 1999.

[SK] S. Khurana et al., "Effect of Hidden Terminals on the Performance of IEEE 802.11 MAC Protocol", *IEEE LCN'98*, pp. 12-20, Oct. 1998.

[SK2] S. Khurana et al., "Performance Evaluation of Distributed Co-Ordination Function for IEEE 802.11Wireless LAN Protocol in Presence of Mobile and Hidden Terminals", *IEEE MASCOTS'99*, pp.40-47, Oct. 1999.

[TK] F. A. Tobagi, L. Kleinrock, "Packet switching in radio channels: Part ii - the hidden terminal problem in carrier sense multiple-access and the busy-tone solution", *IEEE Trans. on Commun.*, pp.1417–1433, December 1975.

[NS]"The Network Simulator–ns2",http://www.isi.edu/nsnam/ns

[NL] P. C. Ng, S. C. Liew, "Re-routing Instability in IEEE 802.11 Multi-hop Ad-hoc

Networks", *IEEE WLN'04*, Nov. 2004, Tampa, USA.

[XS] S. Xu, T. Saadawi, "On TCP over Wireless Multi-hop Networks", *IEEE MILCOM 2001*, Vol.1, pp.282-288, Oct. 2001.

[HA] "HostAP" driver, http://hostap.epitest.fi/

[AB] G. Anastasi, E. Borgia et al., "Wi-Fi in Ad Hoc Mode: A Measurement Study", *IEEE PERCOM'04*, March 2004.

[TR] T. Rappaport, "Wireless Communications: Principles and Practice", Prentice Hall, New Jersey, 2002.

[NL2] P.C. Ng, S. C. Liew, L. B. Jiang, "A Performance Evaluation Framework for IEEE 802.11 Ad-hoc Networks", *ACM PE-WASUN'04*, Venice, Italy, Oct. 2004.

[BW] B. Bensaou, Y. Wang, C. C. Ko, "Fair media access in 802.11 based wireless ad-hoc networks", *ACM MobiHoc'00*, pp. 99 – 106, 2000.

[BS] K. Brown, S. Singh, "M-TCP: TCP for mobile cellular networks", *ACM Computer Communication Review*, 27(5), Oct. 1997.

[HV] G. Holland, N. Vaidya, "Analysisi of TCP Performance over Mobile Ad Hoc Networks", *ACM MobiCom'02*, pp.219-230, Seatle, USA, 2002

[JG] R. Jiang, V. Gupta, C. V. Ravishankar, "Interactions between TCP and the IEEE 802.11 MAC protocol", *IEEE DISCEX'03*, Vol. 1, pp.273 – 282, April 2003.

[XS2] S. Xu, T. Saadawi, "Revealing and solving the TCP instability problem in 802.11 based multi-hop mobile ad hoc networks", *IEEE VTC'01 Fall*, Vol. 1, pp.257-261, 2001

[CT] C. K. Toh, "Ad hoc mobile wireless networks: protocols and systems", Prentice Hall, New Jersey, 2002.

[RFC] "IETF RFC 3561 AODV Routing", http://www.ietf.org/rfc/rfc3561.txt

[HB] X. Huang, B. Bensaou, "On Max-min Fairness and Scheduling in Wireless Ad-Hoc Networks: Analytical Framework and Implementation", *ACM MobiHoc'02*, Long Beach, USA, Oct. 2001.

[NL3] P. C. Ng, S. C. Liew, "Offered Load control in IEEE 802.11 Multi-hop Ad-hoc Networks," *IEEE MASS'04*, Fort Lauderdale, Florida, U.S.A, Oct 2004.