ON SECURE MEDIA STREAMING WITH PATH DIVERSITY IN MANETS

Except where reference is made to the work of others, the work described in this dissertation is my own or was done in collaboration with my advisory committee. This dissertation does not include proprietary or classified information.

Lei Chen

Certificate of Approval:

Kai H. Chang Professor Computer Science and Software Engineering	Chung-wei Lee, Chair Assistant Professor Computer Science and Software Engineering
Yu Wang Assistant Professor Computer Science and Software Engineering	Joe F. Pittman Interim Dean Graduate School

ON SECURE MEDIA STREAMING WITH PATH DIVERSITY IN MANETS

Lei Chen

A Dissertation

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Doctor of Philosophy

Auburn, Alabama
August 4, 2007

ON SECURE MEDIA STREAMING WITH PATH DIVERSITY IN MANETS

Lei Chen

____

Signature of Author

____

Date of Graduation

VITA

Lei Chen, son of Qingzhan Chen and Guiqin Xu, was born on July 28, 1978, in Xi’an, Shaanxi, China. After graduated from Shenzhen Hongling High School in 1996, he attended Nanjing University of Technology and graduated with a Bachelor’s Degree of Engineering in Computer Science and Applications in June, 2000. He began graduate study in Computer Science at Auburn University in August, 2001 toward a Doctor of Philosophy degree.

DISSERTATION ABSTRACT

ON SECURE MEDIA STREAMING WITH PATH DIVERSITY IN MANETS

Lei Chen

Doctor of Philosophy, August 4, 2007
(B.Eng., Nanjing University of Technology, China, 2000)

116 Typed Pages

Directed by Chung-wei Lee

To provide reliable and secure media streaming is quite challenging in the environment of Mobile Ad Hoc Networks (MANETs) in which wireless signals are exposed in air and the quality of media streaming is degraded by signal loss and interference. The mobility of wireless devices makes it even more difficult for such applications as wireless links are often broken when the devices move out of the communication range of their neighbors. In this research, we first study the work that has been done in the above issues, then we propose a serial of methods, techniques and mechanisms of stable multi-path routing, selective encryption and redundancy allocation, and data distribution in MANETs for media streaming. The proposed Multi-path Neighbor Stability Routing algorism finds more stable and long lasting paths than other routing algorithms in simulations. The selective encryption method that we proposed provides a set of four different balance points between the intensity of computation involved and the energy resource of mobile devices. We also proposed a smart data distribution method which takes the consideration of historical data distribution and maximally distributes data in a wide range so that malicious nodes could hardly collect adequate data to reconstruct the original media.

ACKNOWLEDGEMENTS

My graduate study and research at Auburn University has been the most exciting and valuable experience I have ever had. I sincerely appreciate all the professors and my family from whom I gained instructions, advices and help.

First, I would like to thank Dr. Chung-wei Lee, my major professor, as he has been very patient in guiding and helping me through my research. He not only directed me in the research details but also helped develop my serious research discipline.

I thank Dr. Kai Chang for being a great mentor in both my research and life. He helped me build the correct attitude toward academia – an eager heart to learn and keen curiosity to study others' work.

I would like to thank Dr. Yu Wang for giving me excellent advice on my dissertation. She is a very kind and nice person to work with.

Last but not least, I would like to give thanks to my wife Bo Dai, my parents Qingzhan Chen and Guiqin Xu, and my mother-in-law Qinhe Mao, all of whom have been very supportive throughout my work and life.

Style manual or journal used: IEEE Transactions

Computer Software used: Microsoft Word 2003 (text), Microsoft Excel 2003 and 2007 (statistical graphs)

Compiler used: gcc-g++ 3.4.4 (C++ compiler), J2SE 1.4.2 SDK (java compiler)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

As wireless technology is applied more and more intensively and extensively, research on wireless communication has become more important and practical than ever. Conventional wireless infrastructure may be effective in most scenarios; however, in places where no fixed wireless devices, such as base stations or designated routers, exist, new technology tailored for infrastructure-less wireless networks, or Mobile Ad Hoc Networks (MANETs), need to be developed. Section 1.1 describes three real-time video applications utilizing such technology in daily life as well as military and extreme cases such as disaster rescue. Introduction to MANETs and the application model is presented in Section 1.2.

1.1 Applications

Imagine a hiking team is on their way to reach the top of Mount Everest in the Himalayas where no wireless base station can be built. Mobile devices equipped with satellite transceivers may be capable to handle wireless communication at very low data rate, but they fall short in real-time video and/or audio peer communication which can be extremely helpful and even vital in such applications. Each climber can carry a camera and a wireless mobile device so that the devices from all the hikers form a MANET enabling the team to communicate with each other and observe the climbing situation at different angles and altitudes.

Soldiers at a battle field can be equipped with similar devices in order to obtain a panorama of the whole battle field. Real-time video streams from different soldiers can be merged together to plot such virtual reality.

In extreme cases such as disaster rescue, real-time video streaming among mobile devices in MANETs can be very helpful as it enables the rescue team members to collect information from others, or even from mobile robots. Conventional wireless communication can hardly be applied here because most of the fixed wireless communication infrastructure has already been destroyed in cases such as earthquakes or fire.

The above three applications tell us that the new technology of MANETs not only makes our life more convenient and versatile, but also plays a key role in providing public safety and national defense.

1.2 Mobile Ad Hoc Networks (MANETs)

In a Mobile Ad Hoc Network (MANET), each node (black dots in Figure 1.1) is a mobile wireless device (e.g., notebook computer, pocket pc or 3G cell pone etc.) which can communicate with its neighbors within its limited communication range. A pair of two communication end nodes (S and D) far away from each other can utilize nodes between them to establish path(s) (the line segments in the figure) as data delivery channels. Thus, each node can act not only as a transmitter and a receiver, but also as a router which helps find path(s) and forward data for other nodes. MANETs are characterized by wireless connectivity through multi-hops and frequently changing network topology among wireless mobile devices. These characteristics require routing algorithms, methods finding the path(s) from source node to destination node, to be dynamic and adaptive to the constantly changing network structure. The ever changing topology, open signal transmission and low computation power of mobile devices make it very difficult for high data volume applications such as real-time video streaming.

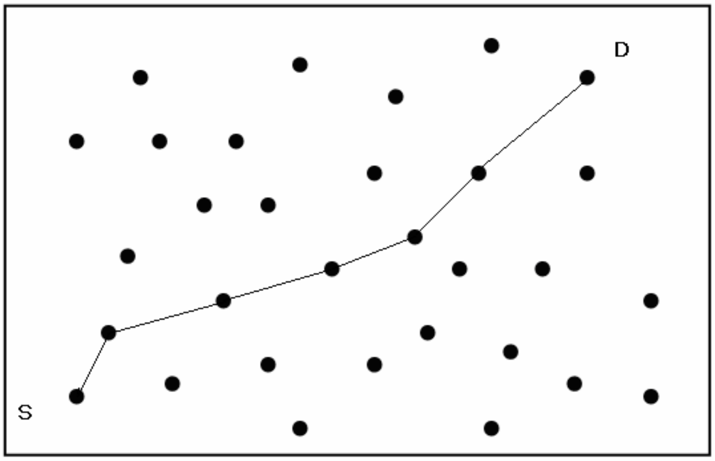


Figure 1.1: A typical Mobile Ad Hoc Network

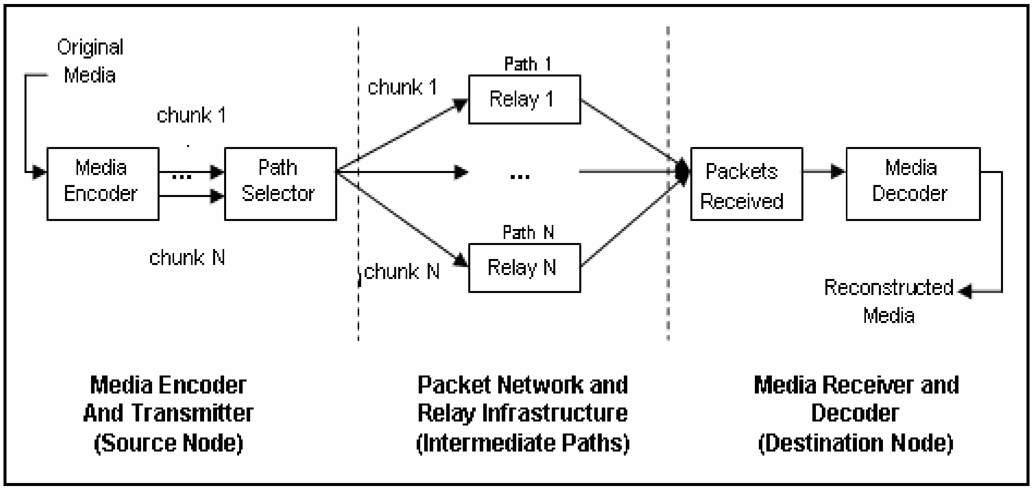

Original Media
chunk 1
Media Encoder
Path Selector
chunk N
chunk 1
Path 1
Relay 1
...
Path N
Relay N
chunk N
Packets Received
Media Decoder
Reconstructed Media
Media Encoder And Transmitter (Source Node)
Packet Network and Relay Infrastructure (Intermediate Paths)
Media Receiver and Decoder (Destination Node)

Figure 1.2: A multi-path media streaming application model over MANETs

Real-time video applications over MANETs can be modeled as Figure 1.2 (a modified and extended model from [Apostolopoulos and Trott 2004]) in which the source node (on the left) takes charge in the operations of routing initiation, media encoding and path selection for data transmission. The paths (in the middle), formed by all the selected intermediate nodes help forward all chunks of data to the destination node (on the right) which then collects data and reconstructs the original video using the media decoder.

1.3 Research Issues

Several issues in the above model need to be solved in order to support real-time video applications.

The first issue to deal with is routing, or to find the path(s) from the source node to the destination node. In order to deliver the video smoothly, stable paths are preferred. Routing for multiple paths is basically an iteration and extension to routing for a single path. In Chapter 3, we first introduce the single path Neighbor Stability Routing (NSR), an ad hoc routing algorithm finding the most stable or long-lasting path between the source and the destination nodes. This single path routing algorithm is then extended to equip multi-path capability (Multi-path Neighbor Stability Routing or MNSR) and Quality of Service (QoS) features.

Before the source node can send out media data through paths to the destination, it needs to prepare the media data suitable to be distributed over the multiple paths in between. When security is taken into consideration in cases such as battle fields, this data preparation process becomes more complex. In Chapter 4, we introduce a set of secure selective encryption solutions which sets four balance points between intensity of encryption and computation power. In other words, when the computation power allows, mobile devices can choose a higher encryption scheme whereas for low computation power units, a lower encryption scheme would fit.

In the environment of MANETs where wireless signals can be degraded seriously by many factors and thus loss of packets could happen at any nodes on the paths, redundancy for the original data is needed in order to provide better video playback quality at the receiver. Since video data is content sensitive, the same amount of redundancy on different segments of source data may lead to various playback qualities even at the same data loss rate. In Chapter 5, we conduct research on how various amount of redundancy at different context affects the playback quality. On the other hand, the mobility of wireless devices is a factor that degrades the performance of MANETs. A study of the effects of mobility on wireless video streaming is presented in Chapter 5 as well.

In Chapter 6, we discuss how paths can be evaluated according to the local neighborhood condition and previous data distribution. This evaluation contributes in making the decision on which path the current chunk of data should go through. Quantitative security analysis shows that the proposed distribution algorithm has advantages on both providing better security and more redundancy.

We draw conclusions in the last chapter, Chapter 7. Future work is also proposed in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Routing over MANETs

MANETs consist of mobile wireless devices which communicate with one another through relatively unreliable wireless connections compared to those in wired networks. This unreliability is mainly caused by two factors: device mobility and signal interference. Mobility can greatly reduce the performance of MANETs, since greater mobility leads to a higher incidence of broken links and changes of signal strength. When each wireless mobile device is moving at a unique speed and direction, the topology of the network keeps changing and it is almost impossible to predict the accurate network structure. A mobile node in a path which is performing well at one moment might be unavailable in the next time interval. Thus, a stable or long lasting path routing algorithm needs to be developed to achieve better performance.

2.1.1 Single-path Routing

In order to achieve better reliability in MANETs, in [Ye et al. 2003], ZhengQiang Ye and his partners introduced a reliable routing framework in which some reliable R-nodes are manually inserted into the MANETs and play a role of supporting the network as backbones. This idea could be practical in some metropolitan areas where advanced network infrastructures are available, but it is not suitable for other improvisational situations such as disaster rescue.

Noticing that information exchange is important among neighbor nodes, in [Joe and Batsell 2002] Joe and Batsell introduced MPR-based hybrid routing which makes use of the multipoint relaying based on the information exchange among neighbor nodes. Nevertheless, this routing algorithm does not involve any memory of the relationship among neighbor nodes. Thus all nodes are treated the same no matter they are stable in terms of mobility or not.

Some other researchers have suggested that routing can be done by collecting and aggregating relative information among neighbor nodes from the source to the destination. For example, in [Toh 2002] and [Toh et al. 2002], Chai-Keong Toh and his co-workers introduced ABR (Associativity-Based Routing), a routing algorithm that makes use of the Associativity Ticks among neighbor nodes. The Associativity Ticks show a mobile node’s dormant time, in which the node is in a stable status. However, these Associativity Ticks are not able to show the long term accumulated relationship among neighbors. For example, a node will treat its old, stable neighbor nodes the same as newcomer neighbor nodes, which might be just passing by the neighborhood. Also, in ABR only the source node maintains the routing information, which means that when a node is temporarily unavailable (for example, when a device is restarting), ABR will redo routing and might choose a worse node and not be able to switch back.

2.1.2 Multi-path Routing

In order to fight against the insecure nature of wireless communications, researchers tried to find more secure paths between the source and destination node [Hu, Y.-C. et al. 2002a], [Hu, Y.-C. et al. 2002b], [Papadimitratos and Haas 2002] and [Yang et al. 2002]. Routing is the process of finding paths before the actual data transmission starts, thus security routing can do very limited help to secure the actual data transmission, especially considering malicious nodes can move around without being caught by others which is totally out of the control of secure routing protocols.

Although routing cannot contribute much to communication security, finding more paths for data delivery can provide better application performance, e.g., smoother video streaming and better video quality due to less packet loss, than only finding and using one single path. Multi-path routing became a research topic as early as in the mid 1970s [Maxemchuk 1975]. Slowly this technique was applied to the most dominant ATM networks in the early 1990s [Dejean et al. 1991], [Lee and Liew 1993] and [Plotkin and Varaiva 1993]. Around the same period of time Multi-path routing appeared in wireless networks [Shacham and King 1987] and [Hu, L. 1993]. Research on Multi-path routing over Local Area Networks and even the Internet were presented in [Tsirigos and Haas 2001], [Lee and Gerla 2001], [Pearlman et al. 2000], [Wu and Harms 2001], [Apostolopoulos 2001], [Liang, Y. J. et al. 2001], [Gogate et al. 2002], [Liang, Y. et al. 2002], [Miu et al. 2003] and [Begen et al.2003]. These researchers suggested that Multi-path routing can be achieved by applying iteration or changes to the corresponding Single-path routing. Finding multiple paths is proven to be very helpful for more bandwidth, better load balancing, less packet loss and less latency in Mobile Ad Hoc Networks in which single path may suffer greatly from the above problems.

2.1.3 QoS Routing

Quality of Service, or QoS, routing has drawn attention in the recent years. QoS routing can be categorized into two paradigms: source QoS routing and hop-by-hop QoS routing [Zhang and Mouftah 2005].

In source QoS routing, the source node locally maintains the global state information and thus is able to locally compute the entire constrained path to the destination node. This approach is easy and straightforward in execution, but it introduces excessive overhead in both gathering and maintaining the global state information and calculating the constrained paths. Examples in this category include the predictive location-based QoS routing designed by Shah et al. in [Shah and Nahrstedt 2002]. In this protocol, instead of disseminating the status of all links throughout the whole network, each node broadcasts its node status, including position, velocity, direction, available resources, etc., periodically or upon significant changes of the network. This protocol only fits in small or medium-sized networks and mobile devices are quipped with Global Positioning Systems (GPS) so that their mobility is predictable. In summary, source QoS routing is not a good option when limiting computation and saving battery power are taken into consideration.

In hop-by-hop QoS routing, on the other hand, no centralized node is available to maintain and compute the global state information. Thus, routing is done via the propogation of the information of the link status between neighbor nodes. There are two routing strategies in this category: shortest path routing and flooding. The shortest path routing simply returns the shortest path when it meets all QoS requirements. An example of such strategy can be found in [Lin and Liu 1999]. The advantages of their work include simplicity, fast route acquisistion and low control overhead. It works well in the case that traffic demand remains low so that the shortest path can meet the QoS requirements. However, it does not fit into real-time video applications which require large volume of data. Flooding strategy on the other hand disseminates a route request message across the whole network for a QoS route on demand. In [Zhu and Corson 2002] and [Kim et al. 2004] this strategy was applied to find the bandwidth-constrained paths in MANETs. Their work can discover the competent paths; however, it also introduces a route discovery communication overhead of O(V) in which V is the set of all nodes in the MANETs. This overhead needs to be cut down when this strategy is applied to a larger sized MANET with moderate or high traffic demand.

Our research in routing over MANETs includes finding the stable or long lasting paths between the source and destination nodes and equipping this routing algorithm with QoS features at a low or moderate overhead. This work is described in Chapter 3.

2.2 Secure Video Streaming

Streaming is the technical term for transferring data as in a flow pipe such that it can be processed as a steady and continuous stream. Streaming technologies are becoming increasingly important and pervasive with the growth of the Internet since most users do not have fast enough access to download large multimedia files quickly. With streaming, the client browser or plug-in can start playing the media before the entire file has been transmitted. In our research, we focus on video streaming since handling video is much more complicated and versatile than handling audio. There are multiple ways to secure media data from being hijacked at different stages of communication. For example, data can be encrypted at the source node before it is transmitted; on the other hand, different chunks of data can be smartly distributed onto different paths so that collecting all chunks in order to restore the original video becomes a non-easy task for the attackers.

2.2.1 Video Formats and Encoding

Currently several video codec (coder-decoder) standards are widely used. Motion Joint Photographic Experts Group (MJPEG or Motion JPEG) [Wiki-Mjpeg] and Moving Picture Experts Group (MPEG) [Wiki-MPEG] are among the popular ones. They are also two of the most popular video streaming formats for almost all different network types. They each have advantages and disadvantages. For example, MPEG applies the concept of Group of Pictures (GOP, or GOV, Group of Video object plain, in MPEG-4; we use GOP and GOV interchangeably in this research) in which a series of P-frames (and B frames if needed) follow a leading I-frame. As the I-frame contains most of the information and P-frames only record the differences between the I-frame and themselves, the video data volume can be reduced and the loss of a few P-frames does not dramatically degrade the visual quality. Different from MPEG, each frame in Motion JPEG video is independent from others, which makes it well suited in video containing lots of scenery changes and movements. Motion JPEG is an ideal codec for emergent or unprepared situations such as battle fields or disaster rescue and recovery where camera is moving fast and successive images in the video vary significantly. Other codec such as H.263 [ACM-H.263] may not be efficient in these situations because their frames are organized in groups of images with only minor changes among the images thus video quality decreases when footage contains lots of movement.

In terms of how different segments of the source data are related to one another, there are two coding schemes, Scalable [Wee and Apostolopoulos 2003] and Multiple Description (MD) [Apostolopoulos 2001], [Apostolopoulos and Trott 2001], [Apostolopoulos et al. 2002], [Apostolopoulos 2004], [Apostolopoulos and Trott 2004], [Gogate et al. 2002], [Heng et al. 2005], [Liang, Y. J. et al. 2001], [Orchard et al. 1997], [Ozarow 1980], [Pearlman et al. 2000], [Tsirigos and Haas 2001], [Wang, Y. and Chung 1996], [Wang, Y. et al. 1997] and [Wolf et al. 1980]. For the purpose of having more bandwidth, less packet loss and less end to end delay, both Scalable coding and Multiple Description coding need to cooperate with Multi-path routing. In Scalable, or layered coding, video is encoded into multiple bitstreams among which a base layer provides low yet usable quality and one or more enhancement layers improve quality. In order to recover the original video, the destination must receive the base layer; in other words, the enhancement layers are useless without the delivery of the base layer. Multiple Description (MD) coding differs Scalable coding in that, each description in MD is equally important and each description can be independently decoded for a usable reproduction of the original video; video quality can be improved on receiving more descriptions. These different coding schemes fit well in the Multi-path environment between source and destination nodes. Nevertheless, they also bring more security issues into the research since securing one single bitstream over one single path is much easier than securing multiple bitstreams over multiple paths. Analyzing source data dependency among the layers or descriptions and mapping this information onto a more secure data distribution method (e.g., splitting the base layer of Scalable Coding into segments and sending them out via different paths can help fight against single attack as no single node can collect all pieces needed for the based layer) is the challenge here.

The relationship between the coding schemes (Scalable and MD) and video coder-decoders (MPEG and MJPEG) is an interesting issue. At different coding levels of a certain coder-decoder, say MPEG, different coding schemes are applied according to the relationship between data segments. For example, at Group of Picture coding level, the relationship among different GOPs is considered as Multiple Description because the decoding of one certain GOP does not depend on other GOPs at all. More details of research on this issue are presented in Section 5.1.

An intensive study of MJPEG codec is included in Chapter 4 followed by a set of proposed selective encryption solutions based on the study.

2.2.2 Security

Security became a major research topic in recent years as more and more data requiring high security is sent through the Internet or open MANETs. A common way to achieve the security goal is to encrypt data at the source and decrypt data at the destination. Encryption and decryption algorithms such as DES (Data Encryption

Standard) and AES (Advanced Encryption Standard) [Daemen and Rijmen 2002] are designed to process data such that encrypted data provides little information to attackers. Some other research focuses on key management [Zhou and Haas 1999], [Kong et al. 2001], [Hubaux et al. 2001], which probably is the most fundamental security issue in MANETs. However, these algorithms do not show any smartness in selecting data to be encrypted, as algorithms themselves treat every bit of data in the same way.

Researchers have proposed the Secure Real-time Transport Protocol (SRTP) [Schulzrinne et al. 1998] which adds security concerns to Real-time Transport Protocol (RTP) [Baugher et al. 2003] and confidential multimedia communications in IP networks [Iacono and Ruland 2002]. However, very little research has been done on mechanisms of selecting or preparing multimedia data before encryption. Adaptively selecting data for encryption at the media source is a non-neglectable issue in our research. This issue is interesting and challenging since in order to select and filter data, it is a must to analyze how media data is formed and organized at the source before encryption. There is always a trade off between encryption and computation overhead that is worth consideration. Uniformly strong encryption involves more computation thus it is not suitable for real-time video streaming, because many mobile devices have very limited computation power.

Using Multi-path routing can broadcast data to a wider range and increase the probability of being overheard or compromised. In [Lou, W. et al. 2004], the source data is split into shares (or chunks), then scrambled and encrypted. Additional redundant shares are added to achieve better reliability. However, the source data mentioned in their work is non-real-time data. Applying similar splitting and scrambling techniques for real-time video streams would not be as effective because video data is more content sensitive. Thus, if shares are not organized in a specific format and sequence, they are meaningless. This would affect the preprocessing methods used for the source data to address security concerns when the shares are sent along multiple paths. Quantitative security analysis is presented in their work for a scenario in which an attacker can eavesdrop on or compromise a node in order to hijack the data transmitted over that specific path. However, no discussion of the case where more than one path is compromised is presented. We consider this is a critical omission because it is very likely that an attacker sits between two close paths and is able to eavesdrop on both paths. A challenging question would be: "What if the shares traveling on these two paths that the attacker has collected are sufficient to recover the original media?"

Research on MANET security has been going on for some years, from general security challenges [Lou and Fang 2003], to detecting malicious behavior [Buchegger and Le Boudec 2002], then to detecting and fighting against intrusion attacks [Zhang et al. 2003] and [Borisov et al. 2001]. We consider that, although detecting malicious behavior and attacks could be helpful against attacks, it is a passive way to provide security as data might have already been hijacked and the original video has been reconstructed by attackers by the time the attacks are detected. A proactive way, e.g., distributing data segments in a pattern unrecognized by attackers via multiple paths, could be more helpful even before the attacks ever start.

Our research on security includes the selective data encryption solutions in Section 4.2 and the smart secure data distribution methods explained in Chapter 6. For security purpose, selecting the path for the current chunk of data to be sent should be based on previous data distribution and current path security status.

2.3 Data redundancy and Mobility

Unlike generic data traffic which is mainly used for static content delivery or file downloads, video streaming is characterized by delay constraints and tolerance to loss. These characteristics indicate that generally it is very difficult to provide guaranteed video streaming service and retransmission of packets is not helpful. Under such condition, data redundancy can help achieve better playback quality at the receiver because it just needs one copy of either the original data chunk or its redundant counterparts. With the help of multi-path routing, bandwidth has been dramatically increased and consequently adding redundancy becomes practical.

Wireless techniques extend the mobility and range of accessible communications. On the other hand, the mobility of the mobile devices possibly results in serious broken links and signal fading. It becomes an interesting research issue to investigate how data redundancy and node mobility effects the performance of video streaming over MANETs as one factor can help achieve better data reliability while the other is on the contrary.

2.3.1 Data Redundancy

Redundancy [Reza 1994] [Schneier 1996] [Venken et al. 2002] can be used to achieve better performance when data gets lost between the source and destination. Redundancy techniques such as Reed-Solomon coding [Koohi et al. 2003] [Lee and Park

2004] can efficiently handle conventional data but are not applicable to most real-time video streaming since video data is encoded in a hierarchy where some portions depend on others. In this case, various amounts of redundancy can be added at different levels according to the coding scheme being used in order to obtain better playback quality at the receiver.

Our research on redundancy focuses on how redundancy affects the performance of video streaming when various amount of redundancy is applied to different levels of the video encoding. This work is presented in Chapter 5.

2.3.2 Mobility

Dealing with real-time video streaming in the environment of MANETs is challenging due to the combination of a lack of infrastructure and high node mobility, which differentiates MANETs from conventional wireless networks. Each mobile node can simultaneously operate as a source node, a router or forwarding node and a receiver node, while at the same time continues moving at a variable speed. This high mobility greatly increases the likelihood of link disconnections, which lead to high data loss rate and thus degrades the video playback quality at the receiver.

Some researchers developed group mobility models according to the movement patterns of mobile nodes in certain applications. For example, in [Hong et al. 1999] and [Wang, K. and Li 2002], the Reference Point Group Mobility (RPGM) model and some extensions were introduced. In RPGM, mobile hosts are organized in groups according to their relative mobility relationships and each group has a logical center whose motion defines the entire group’s motion behavior (location, speed, direction & acceleration). The motion of a single node is the vector sum of the motion of the reference point and the independent random movement. Reference Velocity Group Model (RVGM) extends RPGM by adding velocity vector of the group and each node which gives a more accurate depict of how nodes move. Ravikiran & Singh [Ravikiran and Singh 2004] carried out a simulation and performance analysis on the above models over routing protocols such as Dynamic Source routing (DSR), Ad Hoc On-demand Distance Vector (AODV) routing and Destination-Sequenced Distance-Vector (DSDV) routing. Cano and Manzoni [Cano and Manzoni 2004] also reported the simulation and performance of several different group mobility models. The above mobility models may work well in predicting the location of mobile nodes, however, they fail to record the real-time relationship among neighbor nodes, which is critical in routing optimization (e.g. the most stable or reliable path routing).

Our research on mobility is to find out how the intensity of mobility, in terms of moving speed, affects the performance of video streaming in MANETs. Chapter 5 shows the experimental results and two practical scenarios in which users can choose different amount and level of redundancy in data transmission with the concerns of devices’ moving speed in order to achieve acceptable video streaming performance.

CHAPTER 3

NEIGHBOR STABILITY ROUTING

Routing is the first step to start a network application. In source routing, it is initiated by the source node and looks for all possible paths between the source and destination nodes. Multi-path routing is basically an iteration or extension to single path routing. Finding stable or long lasting paths is essential to providing smooth media streaming over MANETs. In order to achieve this goal, it takes more factors into consideration during the routing process. In this chapter, we present the Neighbor Stability Routing (NSR) algorithm which utilizes the Neighbor Stability Factor (NSF), defined in 3.2.2, between neighbor nodes to construct more stable paths for media streaming. The Multi-path and QoS extensions to NSR are described in Section 3.4.

3.1 Motivation

With the unsolved issues of MANET routing in Section 2.1, we try to map the relationship among neighbor nodes into variable factors, or Neighbor Stability Factors (NSF). At the first peek, this idea looks similar to the ABR algorithm [Toh 2002] [Toh et al. 2002] using Associativity Ticks; however, a very distinct difference is that NSF is a cumulative factor which shows the historical relationship among mobile neighbor nodes over a period of time and this exactly reflects the relationships of the members of a mobile node group, unlike the ABR in which a previous neighbor node becomes totally unknown as soon as the link between them breaks. The actual routing is implemented based on the propagation of NSFs. Compared to AODV and DSR implemented in [Chakeres and Belding-Royer 2004] and [Das et al. 2001] (shown in the simulation section), not only is the NSR algorithm well suited in typical MANET conditions but also is more adaptive to an unstable MANET in which mobile nodes could become temporarily down or fluctuate over a certain range.

The underlying theory of routing algorithms in MANETs should reflect real application scenarios. In a typical scenario, wireless mobile devices will not move in a totally random way and their movement will accord with the intentions of the people who carry these mobile devices. The motion of a group of people who are doing the same work is very likely to be the same, or at least similar, and these people are very likely to stay relatively close to one another. Thus, it is quite reasonable and necessary to distinguish between those nodes which have strong relationships and those with no relationship at all.

A practical solution is to use the concept of “Credit”, or historical behavior, since it is based on the individual’s habits and historical data. For example, a person who has a good credit history and has never paid his or her monthly balance late is much more likely to pay full in the next billing cycle than a person with a bad credit history. In MANETs, a group of nodes, who are conducting the same mission staying close to one another over a period of time, is very likely to remain neighbors compared to those nodes that are just passing through or temporarily visiting the neighborhood.

We can use the Neighbor Stability Factor to show the “credit” described above. A node (as a credit card company) periodically checks (as monthly bills) its neighbors (credit card holders) to see if they are still valid neighbors (if they pay their balances on time). If a neighbor is still valid (pays balance on time) then increase the NSF (or credit line) of that neighbor (card holder); on the other hand, if the neighbor is no longer valid (did not pay on time), decrease the NSF (record the missing or late payment) till it drops to 0 (the credit history becomes really bad), at which time the node will delete that neighbor from its neighbor list (close the credit account or start a law case).

3.2 Neighbor and Stability Routing

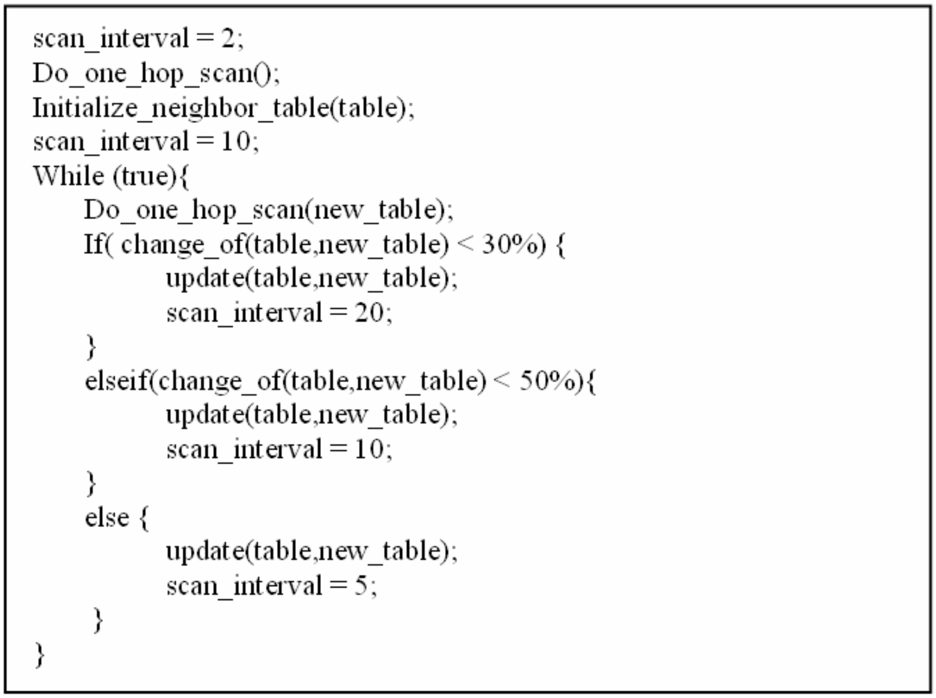
In this section, we describe a new routing algorithm based on the accumulative relative stability among neighbor mobile nodes. The Neighbor Stability Routing algorithm selects historically and accumulatively the most stable mobile nodes to form a path between the source node and destination node. The relative stability is then propagated from the collective data by all the nodes along a path. The cumulative collective data, or Neighbor Stability Factor, reflects the historical neighborhood stability among neighbors. When a node or segment on the path is down, NSR dynamically starts to find an alternative most stable path. In simulation, the NSR algorithm outperforms some major ad hoc routing protocols such as AODV and DSR in packet delivery ratio and number of paths rerouted. NSR also well handles issues such as group node mobility and temporary node unavailability.

In NSR, every node periodically scans its one hop range, within which it can directly sense its neighbors and receive replies from all of them. Armed with such information, each node then sets up a table of its neighbors' stability. At the time of routing, the NSF will be propagated along the path so that the destination node will be able to figure out the path information based on the NSFs and return it back to the source node.

3.2.1 One-hop scan

The way that each node collects its neighbor information is to scan within a one hop range periodically. There are two different types of scan. Initially, when a mobile device is powered on, it performs the first scan. The first scan is different from future scans in that it builds the neighbor list and quickly gets ready for participation in real network activities. This should happen in a relatively short period of time. In our simulation, it is set to two seconds.

After the initial scan, the node enters a normal working status and it needs to keep on collecting its neighbors' information periodically. This time interval should be dynamic, since it depends on how fast the nodes are moving. In an area where most of the nodes are moving at relatively slow speeds the routing will not change frequently and the scan interval can thus be set to a longer time in order to reduce the overhead. However, in an area where most of the nodes are running fast, the scan interval should be set to a shorter time in order to be more adaptive and quickly respond to network changes. Considering the one hop range and the possible values of device movement speeds, the default scan interval in the simulation is set to 10 seconds. In the next scan the node compares the new neighbor list with the old one, and if it discovers that less than 30% of its neighbors have changed (this can be caused by either new neighbors coming in or old neighbors going away), it will set the scan interval to 20 seconds; if more than 30% but less than 50% of its neighbors change, it will keep the time interval as 10 seconds. In the case that more than 50% of its neighbors have been changed, it will then set the scan interval to 5 seconds in order to gather its neighbor information more frequently. By this means, each node can dynamically collect information from its neighbors in order to calculate the NSFs (shown in the next section). Figure 3.1 lists the pseudo code of a one-hop scan at an individual node.

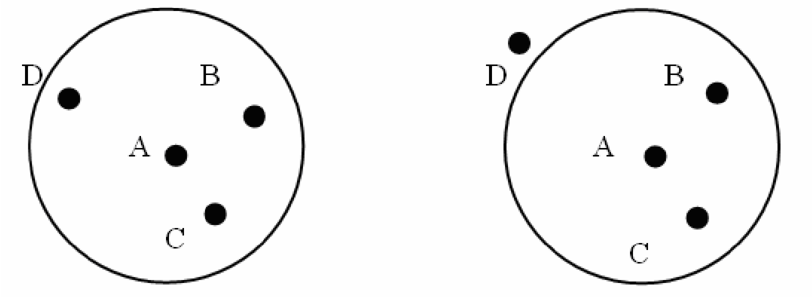


```
scan_interval = 2;
Do_one_hop_scan();
Initialize_neighbor_table(table);
scan_interval = 10;
While (true){
    Do_one_hop_scan(new_table);
    If( change_of(table,new_table) < 30%) {
            update(table,new_table);
            scan_interval = 20;
    }
    elseif(change_of(table,new_table) < 50%){
            update(table,new_table);
            scan_interval = 10;
    }
    else {
            update(table,new_table);
            scan_interval = 5;
     }
}
```

Figure 3.1: Pseudo code of one-hop scan at an individual node

3.2.2 Neighbor Stability Factor

The NSF (Neighbor Stability factor) indicates the cumulative relative stability among neighbors. Each node stores a neighbor list which not only shows the neighbor members of the node, but also records the NSFs of the neighbors and whether those neighbors are still valid. A neighbor node is considered valid if it is still within the one-hop range and thus useable for routing; on the other hand, a neighbor node is not valid if it is out of the one hope range at this time spot. In the NSR algorithm, for the latter case, the neighbor entry will not be deleted if its NSF is still a positive value. Figure 3.2 shows node A's neighborhood in two continuous time intervals. Within this period of time, node D, a neighbor node of A at time T, moves out of the communication range of node A and thus becomes an invalid node for A at time T+1. Table 3.1 records the neighbor lists and the Neighbor Stability Factors at the same two time intervals. Please note that node D's record will remain in the list until its NSF drops to 0.

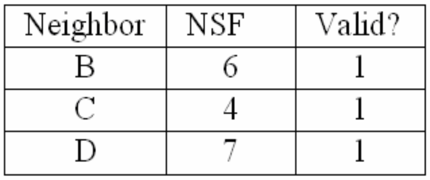


Time T Time T+1

Figure 3.2: Node A's neighborhood at time T (left) and T+1 (right)

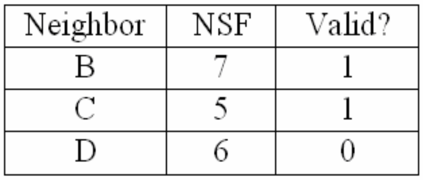
Table 3.1

Neighbor lists of node A at time T (left) and T+1 (right)



Neighbor	NSF	Valid?
B	6	1
C	4	1
D	7	1

Time T



Neighbor	NSF	Valid?
B	7	1
C	5	1
D	6	0

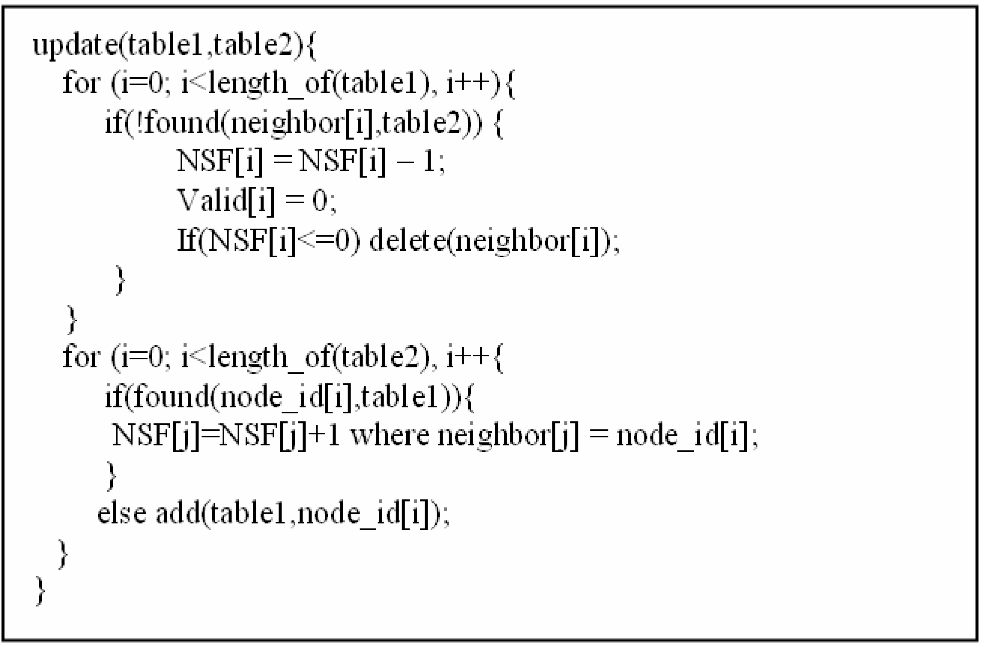
Time T+1

The update of the Neighbor Stability Factors follows these rules:

1. Increase the NSF by 1 when the node remains in the one-hop range in the new time interval.
2. Decrease the NSF by 1 when the node is out of the one-hop range in the new time interval. If the NSF drops to 0, delete the neighbor entry from the list; otherwise, set the item “Valid” to zero indicating it is not available for routing.

The value of NSF indicates the difference of the intervals in which a certain node is considered a neighbor and the others in which it is not considered a neighbor of Node A. For example, until time T, node B has been node A’s neighbor for 6 more intervals than it is not a neighbor of A; and at time T+1 after A’s next scan, the NSF of B has been increased by 1 since it is still in the range.

Although at time T+1 node D is no longer a neighbor of A and has become invalid, node A still keeps D’s record in its neighbor list table since D’s NSF is still positive. Node A will set the third column “Valid?” of node D to zero. This will exclude D from neighbor selection if routing is required. The reason why D is retained in A’s neighbor list is because node D might be just moving out of A’s one-hop range temporarily, or it might just restart the device. If D returns to be a neighbor of A at time T+2, it will be treated as an “old neighbor” and resume its last NSF. Consequently, D is still considered more stable relative to A than other nodes that just pass by the area randomly and have relatively small NSF values.



```
update(table1,table2){
   for (i=0; i<length_of(table1), i++){
       if(!found(neighbor[i],table2)) {
              NSF[i] = NSF[i] – 1;
              Valid[i] = 0;
              If(NSF[i]<=0) delete(neighbor[i]);
        }
   }
   for (i=0; i<length_of(table2), i++{
       if(found(node_id[i],table1)){
        NSF[j]=NSF[j]+1 where neighbor[j] = node_id[i];
       }
      else add(table1,node_id[i]);
  }
}
```

Figure 3.3: Pseudo code of Neighbor List Table update method

3.2.3 NSF Propagation and Routing

The operations of NSF and the Neighbor List Table explained in the last section are prerequisites of the Neighbor Stability Routing (NSR) algorithm. When a node starts routing operation, as node A initiates routing to node E shown in Figure 3.4, A will broadcast route request RREQ packets to all its neighbors. All the neighbors who receive such packets will then forward them to all their neighbors. By this means, E will finally receive the RREQ. The intermediate node only responses the first RREQ it receives in order to save bandwidth and energy. During this process, every intermediate node includes the current hop count and the NSF of the neighbor node from which it receives the RREQ. For example, B will include the NSF of A, which is stored in B's Neighbor List Table in the RREQ packet header, and forward it to C. At the end of the route, E will calculate the sum, average and standard deviation of the NSFs along this path.

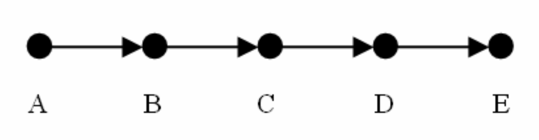


Figure 3.4: Direction of source routing

The sum of the NSFs is related to both the length of the path (number of hops) and the average value of the NSFs along this path. The average NSF thus shows the mean stability among neighbors along this path. However it does not show whether every segment on this path is stable, or if some segments are stable and others are not. Therefore, it is necessary to calculate the standard deviation of NSFs. If the average NSF is large while the standard deviation is small, it means that every segment on this path is stable; on the other hand, if the average NSF is large and the standard deviation is also large, it means that some segments on this path are in very good condition while some others are in bad condition. The method of NSF calculation is shown in Figure 3.5.

$$SUM_{A\text{->}E}=NSF_{A\text{->}B}+NSF_{B\text{->}C}+NSF_{C\text{->}D}+NSF_{D\text{->}E}$$
$$AVG_{A\text{->}E}=SUM_{A\text{->}E}/\text{Hop-Count}$$
$$STDDEV_{A\text{->}E}=\text{sqrt}(\ \Sigma\,\text{sqr}(NSF_i-AVG_{A\text{->}E})/\text{Hop-Count}\ )\ (\text{i is from A to E})$$

Figure 3.5: Method of NSF calculations

Two network scenarios are discussed below to further explain the purpose and advantages of the NSF calculation methods. Both of them demonstrate the advantage of the NSR algorithm over others and why the above calculation is necessary for routing optimization. They are related to a group of nodes passing through a region of some other nodes.

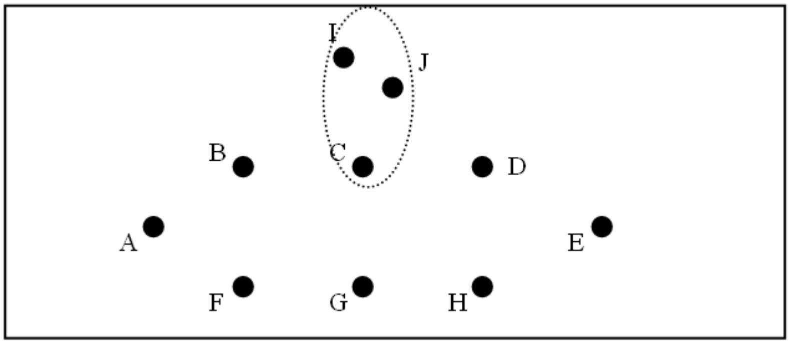


Figure 3.6: Group movement in a MANET -- Scenario 1

Consider a MANET (shown in Figure 3.6) with two paths from source A to destination E. One is the upper path A→B→C→D→E and the other is the lower path A→F→G→H→E. Notice that node C is a member of a group containing three nodes (I, J & C). Suppose this group of nodes is moving downward and C has just entered this network region. At this time, because C is a new neighbor to both B and D, by calculating the average of NSFs it is known that this path is not as stable as the lower path because the upper one has a smaller average of NSFs.

Simply calculating the average of NSFs is enough to handle scenario 1 but not for scenario 2, shown in Figure 3.7. C & J are both in the same group and on the same path from A to E. Because they are in the same group, the NSFs between C and J are quite large as they are the neighbors to each other in the same mobility group for a relatively long time. This consequently will increase the average of the NSFs in the upper path considerably. Even though it is not as stable as the lower path, the algorithm might still choose the upper path if it only calculates the average of the NSFs. In order to solve this problem, the calculation of standard deviation of NSFs is required. Because node pairs (B, C) and (J, D) are both new neighbors to each other, and (C, J) are old and stable neighbors, the deviation of NSFs of the upper path will be larger than the lower one. Thus the lower path will be chosen. If there are more nodes from other groups in a path, the same approach can be applied.

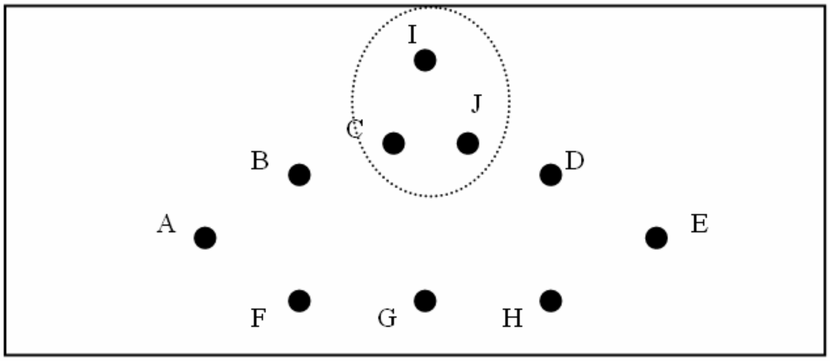
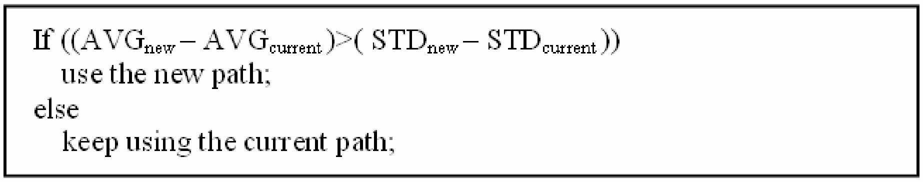


Figure 3.7: Group movement in a MANET -- Scenario 2

After the destination node receives the first RREQ packet, it will send the route reply RREP packet along the same path back to the source node. Every intermediate node along this path will confirm its routing table entry when it receives the RREP packet and then forward it to the next node on the path closer to the source. As the source receives the RREP packet, the path is established and data packets are ready to be sent.

3.2.4 Path Updates

When the destination receives other RREQ packets from other paths, it will compare the new NSFs to the NSFs of the current path according to the path update method in Figure 3.8. According to the method, in order to permit a new path to replace the current path, the new path must have an advantage, due to the average NSFs over the current path, larger than the advantage due to the deviation of NSFs of the current path over the new path. For example, a new path with NSF mean and standard deviation values of 8 and 4, respectively, will be considered a less stable path compared to the current path with NSF mean and standard deviations values of 7 and 2, even though the new path has larger average NSFs. In other words, even thought the link segments of the new path are more stable on average than the current path, they also have a larger variation (caused by some bad condition segments) than the segments on the current path. In the above scenario, this variation of the new path is so large (4-2=2) that it overshadows the advantage it has on average of NSFs (8-7=1) over the current path. Thus, the new path will not be used to replace the current one.



```
If (($AVG_{new} - AVG_{current}$)>( $STD_{new} - STD_{current}$))
    use the new path;
else
    keep using the current path;
```

Figure 3.8: Path update method

3.3 Simulation and Results

The simulations were based on NSR using the NS-2 network simulator. We compared the performance of NSR to AODV and DSR with respect to Packet Delivery Fraction, number of paths rerouted and average data packet delay. For the first two metrics, NSR outperforms AODV and DSR; and for the last metric, NSR performs between AODV and DSR.

3.3.1 Environmental Setup

The simulation scenario is based on an area of 1500m by 500m with a reflecting boundary as shown in Figure 3.9, in which 18 nodes are moving in random directions. Other 12 nodes are organized in two groups of 6 nodes (with a dashed circular boundary shown) are moving in random direction and keeping the intra-group relative member locations unchanged. All 30 nodes have a maximum moving speed of 10m/s. The radio transmission radius is 150m. In this simulation, we use CBR (Constant Bit-Rate) traffic sources. NS-2 generates data packets at an average interval of 50ms and the packet size is 512 bytes. The simulation time was set to 1000 seconds.

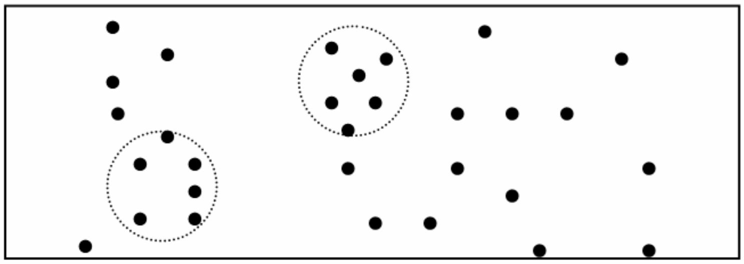


Figure 3.9: NSR simulation environment

3.3.2 Simulation Results

As shown in Figure 3.10, the advantage of PDF (Packet Delivery Fraction) for NSR over that of AODV and DSR is between 7% and 15% when the pause time of nodes is between 0 and 900 seconds. At pause time of 150 seconds, the difference reaches the peak of 15%. When the mobility is very low or very high, the difference becomes smaller.

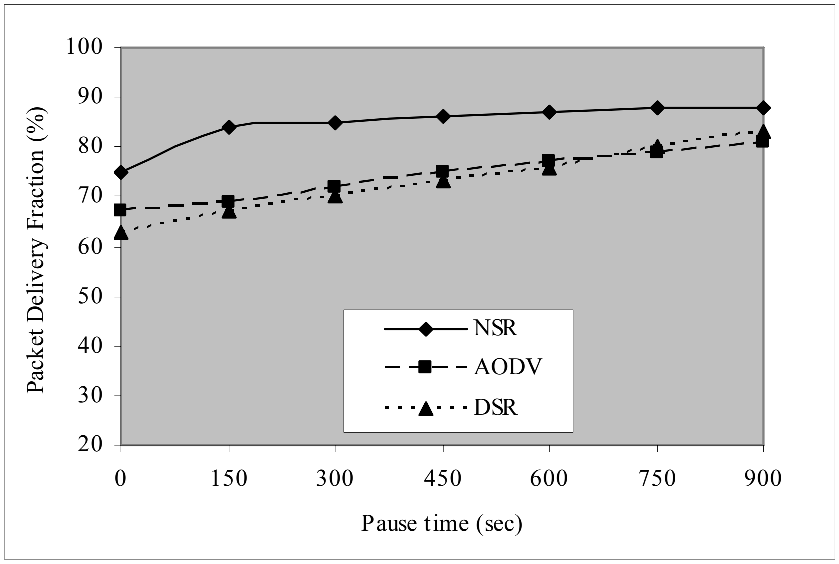


Figure 3.10: Comparison of Packet Delivery Fraction among NSR, AODV and DSR

With very low mobility (long pause time), network topology does not change frequently, thus AODV and DSR can both achieve a relatively high PDF; with very high mobility (short pause time), the topology changes rapidly and even with dynamic path update method, the PDF of NSR is still dragged down close to that of AODV and NSR.

The total number of paths rerouted is the second performance metric in the simulation. This metric reflects how stable the paths are. As shown in Figure 3.11, NSR averages 200 plus path reroutes fewer than AODV and DSR due to the group mobility along with the dynamic scans. If more groups of nodes are involved, the performance of NSR would be even better.

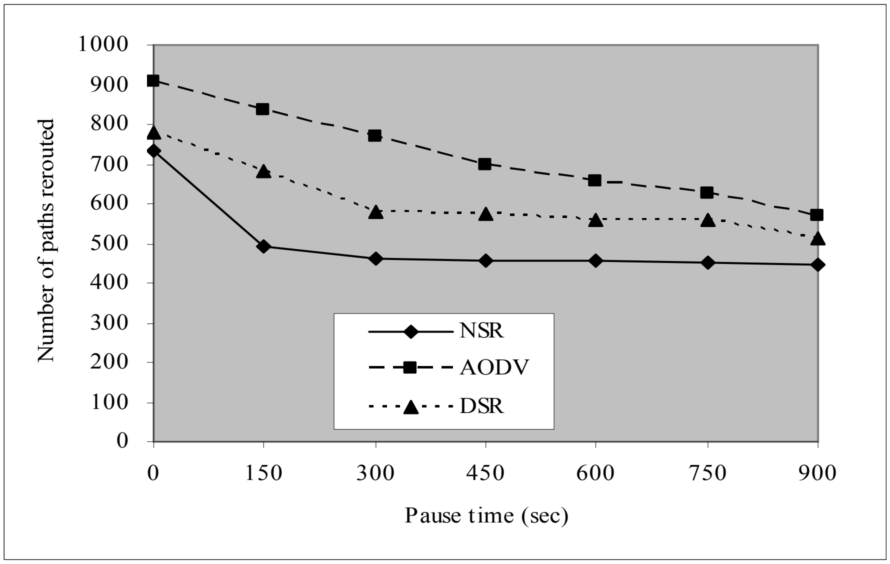


Figure 3.11: Comparison of number of paths rerouted among NSR, AODV and DSR

The third metric we compared is the average of data packet delay as shown in Figure 3.12. Since NSR has more control packet overhead than AODV, it has a longer average packet delay than AODV and slightly lower than DSR. The average delay is mainly caused by queuing delay. In the whole spectrum from 0 to 900 seconds, the curves show a decreasing trend, it may not be the same in a small range such as from 450 to 750 seconds. This is because the average delay is highly related to the traffic balance and topology of the network. Neither of these three protocols has any mechanism for load balancing which very likely leads to a result that long queues are generated at certain nodes (for example, the only node that connecting two groups of nodes).

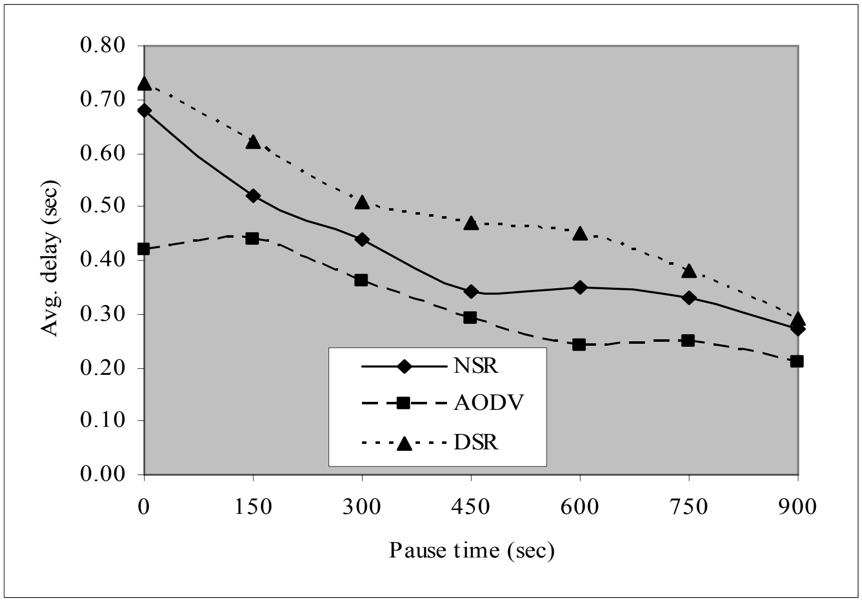


Figure 3.12: Comparison of average data packet delay among NSR, AODV and DSR

3.4 Extensions

3.4.1 Multi-path Extension

The original NSR is a node-disjoint single path routing algorithm in which each node periodically sends out scan packets to update its neighbor list. In the long term, a larger value of NSF indicates a more stable relation between a specific pair of neighbors. When the NSFs at each node are aggregated from source to destination, a more stable path can be found to support real time traffic such as video streaming. In order to achieve better video streaming performance, routing for more paths is preferred. The following section discusses how the original NSR can be revised and extended to search for multiple paths.

A. Revision 1

Using multiple paths can be achieved by making changes at the destination node. The destination node replies to each Routing Request packet (RREQ) and sends back Route Reply packet (RREP) more than once. This does not cause excessive overhead since in normal cases, the number of available paths is between 6 and 12 [Apostolopoulos 2001]. The Path Updates portion of NSR is removed during Multi-path routing and the destination should first provide the source with the information of all possible paths, leaving it to the source node to compare the security level and stability level of all the paths and decide which paths to use and how traffic should be sent over these paths. Based on the RREPs sent by the destination node, the source node is aware of the average and standard deviation of the Neighbor Stability Factors along each available path. With security as the first consideration, stability becomes the second consideration and will be chosen as high as possible without compromising security.

B. Revision 2

The source node collects local and environmental density information from all the intermediate nodes. This is very important information, as denser areas make it much easier for an attacker to eavesdrop and collect more data. We assume no topology control is applied and all nodes move freely. Therefore, the source node needs to find a way to avoid sending too many packets through the denser areas. This can be easily achieved because in NSR each node periodically polls its neighbors and thus knows the density in its area. Each intermediate node needs to report to the source what neighbors it has.

Each intermediate node only needs to report to the source the list of neighbor nodes around him. In figure 3.13, suppose A is the source, E is the destination, and B, D and G are C’s neighbors. C only needs to report to A that B, D & G are its neighbors. Upon receiving this information, and since A knows there is another path from A to E (A→F→G→H→E), A knows that G is a neighbor of C and at the same time it is on another path. This means that the area around C is more vulnerable to an attacker, such as X, as X can capture traffic traveling along both paths.

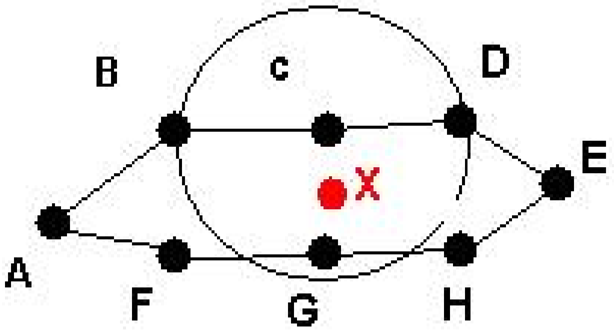


Figure 3.13: A simple MANET with single attacker

In order to be aware of the density distribution and thus make a better decision when sending out traffic through paths consisting of different density areas, the source must have a way to evaluate these paths. This is discussed in Section 6.1.

3.4.2 Quality of Service Extension

In general, QoS routing aims at finding paths that meet certain application requirements such as bandwidth and delay constrains. The conventional way to achieve this goal is to include QoS metrics during the process of routing. The difficulty in this process is, however, how to decide which path(s) to take when multiple metrics are considered; for example, which path to choose when a path has the most available bandwidth while another has the least delay. Theoretically this is an NP-complete problem. Another issue is how to reduce the overhead of routing as these routing packets could possibly consume quite some bandwidth of the network, energy and time of mobile nodes when they respond to such routing information.

In our research, we focus on two QoS metrics: bandwidth and delay constraints. Multi-path NSR is modified as follows in order to meet the above requirements.

A. Modification to One-hop Scan

In order to consider QoS, each node needs to obtain information of bandwidth and delay about the links between itself and all its neighbor nodes. Each neighbor node examines its current available bandwidth, in kbps (kilo-bit-per-second), or the bandwidth it wants to serve and includes this information in the reply packet. The local delay can be estimated from the queue length and marked in milliseconds. On receiving the above information, a node consequently updates its Neighbor List Table which extends Table 3.1 with two more columns: bandwidth and delay. An example of this extended table is shown in Table 3.2:

Table 3.2

An example of Extended Neighbor List Table

Neighbor	NSF	Valid?	Bandwidth	Delay
B	7	1	300	100
C	5	1	400	120
D	6	0	--	--

B. Modification to routing propagation

During the routing propagation process, each intermediate node updates the propagated information of bandwidth and delay, as shown in Figure 3.14. This propagation extends the methods of NSF calculation in Figure 3.5. In Figure 3.14, BW stands for bandwidth and DL stands for delay. The network is based on the scenario in Figure 3.4.

$$BW_{A->E} = MIN(BW_{A->B}, BW_{B->C}, BW_{C->D}, BW_{D->E})$$

$$DL_{A->E} = SUM(DL_{A->B}, DL_{B->C}, DL_{C->D}, DL_{D->E})$$

Figure 3.14: Propagation of bandwidth and delay information

The destination node returns this information along with the information of NSFs and hop count back the source node.

C. Modification to Path Selection

The source node takes charge in deciding which paths to take. It takes the following steps for it to finalize the list of paths:

Step 1. Delay Examination

This step examines two things – both the absolute value of the delay and the difference of the delay from different paths. The latter determines the delay jitter and consequently decides the required video buffer size at the destination node.

The delay of a path is the accumulation of the delays happen at each node of the path. The source node first sorts the path candidates according to the estimated delay in an ascending order. Then it calculates the differences between the delay of least-delay path and that of other paths. Finally, the paths with a delay difference larger than the preset delay jitter throttle (this can be set in an network application) will be removed from the list, because traffic goes through these paths may arrives at the destination node much later than the rest of the traffic which degrades or even pauses the playback.

Step 2. Bandwidth Examination

The available bandwidth of a path is decided by the bottle neck of the path, or the link providing the least bandwidth on a path. Basically, the consideration on bandwidth follows the rule "the more the better" and thus the more paths selected, the better the performance. However, the paths with apparently very little bandwidth (e.g., less than 10kbps) should be removed from the list because these paths often cause problems when slight traffic disturbance occurs.

3.5 Chapter Summary

In this chapter, we present the Neighbor Stability Routing and its multi-path and QoS extensions. The routing algorithm is based on the calculation and propagation of the Neighbor Stability Factors along a path. This factor ties closely to the historical and cumulative relationship among neighbor nodes and is obtained via the dynamic one hop scans performed at each node. NSR can not only be applied to the common MANETs with only randomly moving nodes, but also fit into scenarios that include groups of nodes in which all the members have same or similar mobility patterns. Even when a group of nodes are scattered in an area (yet still in communication range) with random nodes moving in and out of range, NSR will still find the most stable and reliable group members for routing consideration. Because the NSFs used in NSR are obtained in a dynamic way, it can also be applied to scenarios which may involve nodes moving at high speeds and/or with little pause time. The simulation results showed that NSR is superior to AODV and DSR in terms of performance in Packet Delivery Fraction and number of paths rerouted.

Our future work includes the collaboration of NSR with load balancing mechanisms (for better media streaming purpose) which will decrease queuing delay caused by traffic congestions at certain key link nodes.

CHAPTER 4

SECURE MEDIA STREAMING

Routing between the source and the destination nodes is just the first step in a typical application shown in Figure 1.2. In this chapter, we propose a set of solutions to data selection for encrypting Motion JPEG video frames over MANETs. These solutions are based on our intensive analysis of the JPEG image compression procedure. The solutions select the most important data which contains most of the information of an image for encryption. Consequently this mechanism provides a set of balances between security and computation for different hardware environments. Extensive experiments are conducted and performance data has been analyzed to evaluate the proposed solutions. Results show that these solutions are well suited in different scenarios with various computation power and security requirements.

4.1 Motion JPEG and JPEG Encoding

In this section, we choose a typical video format Motion JPEG as an example for media streaming over MANETs. Other video formats might have difference, but the concept is basically the same.

4.1.1 Motion JPEG

Motion JPEG differs from other video types such as MPEG or H263 in that Motion JPEG consists of individual frames or images instead of images groups within which frames are highly correlated. This unique characteristic of Motion JPEG makes it more suited for video taken in emergency or unprepared scenes such as battle fields or disaster rescue where cameras move frequently and two adjacent video frames may be quite different from each other. Motion JPEG video consists of a series of independent JPEG images. An analysis on JPEG compression is presented in this section.

4.1.2 JPEG image preparation

JPEG, or Joint Photographic Experts Group, is a standardized image compression mechanism. It is designed for compressing full-color or gray-scale continuous still images of natural, real-world scenes. A JPEG image has a certain resolution which is presented as the number of pixels in X and Y directions. Before a source image is sent to be further processed and compressed, it has to be divided into small blocks of 8*8 pixels each. The reason of dividing an image into small blocks is because generally within such a small block, luminance and chrominance values do not change much and thus a unit of block is good for differential compression.

4.1.3 Discrete Cosine Transform (DCT)

The Discrete Cosine Transform (DCT) helps sort the pixel values within a block according to their appearance frequency (with respect to the image's visual quality). DCT is similar to discrete Fourier transform: it transforms a signal or image from the spatial domain to the frequency domain. According to [Wiki-DCT], the formula for the standard DCT-II, the most popular DCT used in JPEG encoding, is as Formula 4.1.

$$X_k = \sum_{n=0}^{N-1} x_n \cos[\frac{\pi}{N}(n+\frac{1}{2})k] \qquad k = 0, ..., N\text{-}1$$

Formula 4.1: DCT-II

In the above formula, X_k is the value of the coefficient after being transformed and x_n is the original value of the pixels either in a row or a column of the block. This formula will be applied to each row and column of the 8*8 block, thus $N=8$ for a single row or column. After this formula is applied, most of the information of a block moves to the upper left corner of the block. The element at the upper left corner is called the DC coefficient which sets the basic tone of the whole block. The other 63 values in the block are called AC coefficients.

This process prepares data in an importance-based way that makes encoding and compression easier. Our selective data encryption method in the next section is based on this preparation step. No loss or quality reduction has been introduced up to this step.

4.1.4 Quantization

Quantization is the step that brings loss to a JPEG image. The main purpose of quantization is to greatly reduce the amount of the information in a block, especially among the high frequency components which gather in the lower right area of the block. In frequency domain, this can be done by dividing the components in the block by a matrix of constants and then rounding the results to the nearest integers. This matrix of constants is called the Quantization Table. The values of these constants can be predefined for different image qualities. For example, if the Quantization table is set to use relatively small values, the quantized values will be relatively large so that more information is maintained and better image quality is reserved. However, this will increase the amount of data for encoding.

After the step of Quantization, most values in the quantized matrix become very small with many of them having value of zero which greatly reduces the amount of data to be further encoded.

4.1.5 Encoding

Up to this step, entries in a matrix are logically separated. Encoding methods for DC coefficients (the upper left corner component for each block) and AC coefficients (the rest components) are different.

A. DC coefficients encoding

Since each block only has one DC coefficient and it reflects the average intensity of that specific block, encoding of DC coefficients is done using the Differential Pulse Code Modulation (DPCM) which records the first DC coefficient in the first block of the image and the differences between every two DC coefficients in two adjacent blocks. By only recording the differences starting from the second block, the amount of data for encoding has been reduced considerably. This process is shown below in Figure 4.1 and Figure 4.2.

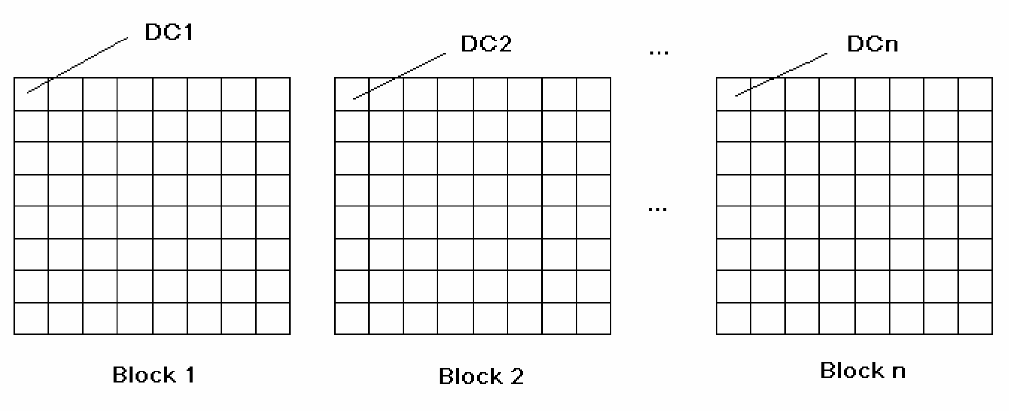


Figure 4.1: DC coefficients of blocks in an image

$$DC_1, (DC_2\text{-}DC_1), (DC_3\text{-}DC_2), \ldots, (DC_n\text{-}DC_{n\text{-}1})$$

Figure 4.2: DC encoding sequence and values

The entropy coding for DC coefficients finally formats DC coefficients into pairs of symbols (Size, Amplitude) in which Size is the number of bits representing the DC coefficient and Amplitude is the actual bits of the DC coefficient.

B.	AC coefficients encoding

AC coefficients are handled in a different way as the ones close to the DC coefficient at the upper-left area of the matrix are more important than the others. A Zig-Zag [Wiki-JPEG] scan, grouping components with similar frequencies, is used to guarantee that the more important data is kept together.

After the Zig-Zag scan and Run-length encoding, the AC coefficients in a block look something like (42 –3 2 1 –1 1 0 0 0 0 0 –1 EOB) and this sequence is then further formatted as pairs of symbols.

4.1.6 JPEG bit stream

The well prepared data (containing all DC and AC coefficients) is then sent as bit stream in the order and layers of blocks to segments, segments to different scans (e.g., Y C_b & C_r), and finally scans to an individual frame.

Quantization tables, entropy encoding tables, and other head information such as the resolution of the image and the number of components in scan are also inserted in the bit Stream.

4.1.7 Decoding

Decoding is a reverse process of encoding. A compressed data block delivered at the receiver first goes through the Lossless Decoder using the Decoding Table; then it is de-quantized by the Dequantizer using the same Quantization Table. Finally, after going through the Inverse DCT, it is reconstructed to the original 8*8 block which functions as a small portion of the image.

In this section, we have analyzed the JPEG compression and encoding process in details. This analysis plays a crucial role in the following encryption selection in the next section since it clearly describes which parts of data are the most important.

4.2 Selective Data Encryption

This section explains how the data for encryption is selected and why it is selected. The data selection process is performed after the DC and AC coefficient encoding has finished. We propose four different data selection solutions to balance the computation and security in order to make the solutions fit in various situations.

4.2.1 Full encryption

Full Encryption includes all data for encryption. It simply forwards every bit of data to the process of encryption. This solution fits for communications among machines with high computation power where the computation of encryption and decryption does not seriously delay the playback of the video.

4.2.2 Half encryption

When only less powerful machines are available and in order to achieve smooth playback of the video, the amount of data needs to be reduces for encryption and decryption. This solution involves selecting half of the most important data for processing. As mentioned earlier, each image contains luminance (Y) and chrominance (C_b & C_r). These three components have independent scans during the compression process of JPEG. An interesting and important observation is that our human vision system is sensitive to lightness and brightness (Y component) while relatively insensitive to chrominance (C_b & C_r) [Borisov et al. 2001]. In other words, data of Y component in JPEG should be considered as more important data than C_b & C_r components. There are several combinations of Y and (C_b & C_r) with different ratios of components and the most commonly used one is the "4:2:2" format in which the ratio of the number of information bits for Y component to the number of bits for C_b and C_r is 4:2:2. The amount of data obtained in scan of Y component takes about half of the amount of data in all scans of Y, C_b and C_r. Thus in this half encryption solution, only the Y component is encrypted. Although C_b and C_r components are not encrypted, it is considered very difficult for an attacker to restore the original image without the plaintext of Y, the luminance or the intensity, component information.

4.2.3 Quarter encryption

When video streaming devices are not powerful enough to handle Motion JPEG video at half encryption mode, the solution needs to further reduce the amount of computation. For this Quarter Encryption solution, we analyze the data of the Y component in the Half Encryption solution. Since the DC coefficients in the scan of Y component decide the average intensity of an image, they are the first part of data for encryption.

Besides these DC coefficients, the AC coefficients, especially those located close to the upper-left corner of the matrix are more important than the others. Thus part of the AC coefficients close to the DC coefficient need to be included for encryption too. After Zig-Zag scans, AC coefficients are represented as pairs of symbols. For example, the Zig-Zag scan result in Section 4.1.5 (42 –3 2 1 –1 1 0 0 0 0 0 –1 EOB) will be represented as (0,-3)(0,2)(0,1)(0,-1)(0,1)(5,-1) in which the first value in a pair is the number of zeroes and the second value is the value of next-nonzero digit [Chen and Lee 2005a] and the first value 42 is skipped since it is the DC coefficient.

This Quarter Encryption is not fixed exactly to 1/4 of the amount of data since for different image quality requirements the number of non-zero AC coefficients in a block would be different. For instance, a high quality JPEG image would have a Quantization table with small values mentioned in Section 4.1.4. In this case, more non-zero AC coefficients appear in the block and thus more AC coefficients need to be sent for encryption in order to achieve the same security level as a lower quality image. However, in an average quality image with only a few pairs of symbols for AC coefficients in a block, having the first 2 pairs of AC symbols for encryption will make the whole encryption percentage around 25%.

4.2.4 Minimum encryption

With mobile devices more and more popular in nowadays, video streaming on mobile devices needs to be taken into consideration. Mobile devices commonly have very limited battery power and computation power. However, security issue is more important in wireless communication than in its wired counterparts. In order to obtain a certain level of security while still minimize the consumption of battery power and computation, we propose this Minimum Encryption solution.

In the Minimum Encryption solution, only the DC coefficient of the first block of the image and the differences of DC coefficients of the rest of the blocks are selected for encryption. This solution provides an acceptable security level as the most important information of an image (the base tone or average intensity) is hidden and at the same time, it minimizes the amount of data for processing, making it good for video streaming over mobile devices.

In the above section, we analyze the importance of the different data segments in a JPEG image and accordingly propose four solutions suited for diverse hardware environments.

4.3 Cryptography and Encryption over SRTP (Secure Real-Time Transport Protocol)

After the most important data has been selected, it is sent to the process of encryption. We use the Advanced Encryption Standard (AES) to encrypt the selected data and the unselected data will just skip this part.

4.3.1 Advanced Encryption over SRTP

The Advanced Encryption Standard (AES) [Bacard 1995] or Rijndael is a symmetric iterated block cipher [Hu, L. 1993] [Apostolopoulos 2001] cryptographic algorithm designed to only use simple whole-byte operations. AES fixes the block length to 128 bits and supports key lengths of 128, 192 or 256 bits. AES offers a large key size and ensures that the only known approach to decrypt a message is the Brute Force searching which tries all possible candidates for the solution.

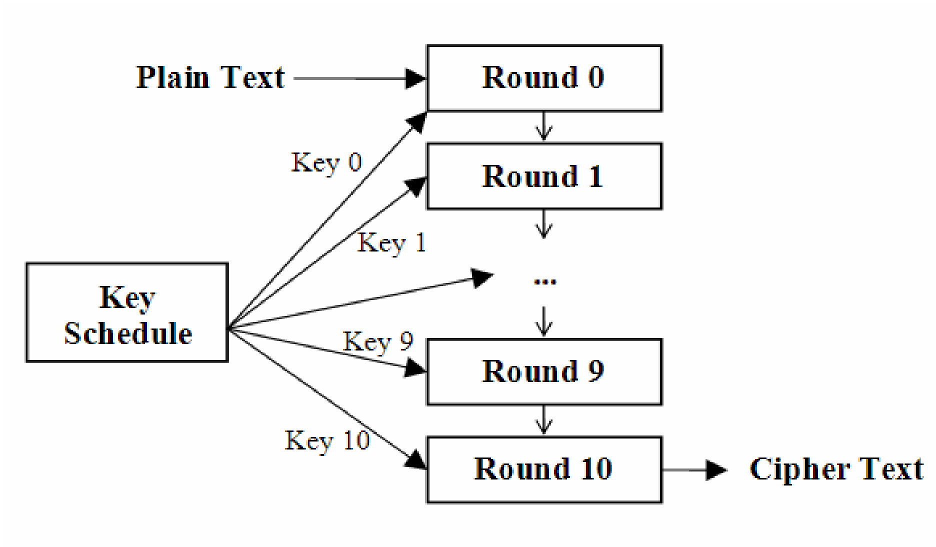


Figure 4.3: Diagram of Rijndael Algorithm with 128-bit key

Figure 4.3 shows the block diagram of Rijndael Algorithm when the key length is set to 128 bits. The number of rounds will increase to 12 and 14 when 192-bit and 256-bit keys are used. A data block in the above transformations is a four-column rectangular array of 4-byte vectors. In order to match the format of data blocks, keys are also stored in a rectangular array of 4-byte vectors. However, the number of columns is dependent on key length.

4.3.2 AES vs. Rivest Cipher 4 (RC4)

Introduced by Ron Rivest, RC4 is a symmetric stream cipher with an arbitrary key size. Stream ciphers such as RC4 encrypt plaintext one byte or one bit at a time and can be considered as a block cipher with very small block size. RC4 has a higher speed than AES (about 1.77 times) as shown in the benchmarks comparison [Baugher et al. 2003]. However, unlike AES, RC4 can only take the key as an argument [Buchegger and Le Boudec 2002]; while AES can take both key and blocks of plaintext as arguments. This makes AES flexibly suited in the different data selection algorithms that we have introduced. RC4 also suffers significant attacks making it not viable for long term standardization while AES has no known shortcut attacks. Based on the above, we choose AES for the cipher.

4.3.3 Secure Real-Time Transport Protocol (SRTP)

Currently, RTP (Real-time Transport Protocol) [Apostolopoulos and Trott 2001] is the only Transport Protocol for real time media. Therefore, adding security features on

RTP, as in SRTP, is necessary for real time applications. SRTP intercepts RTP packets; then after cipher encryption, forwards SRTP packets to the receiver. During this process, only the payload of the RTP packet will be encrypted.

In the above section, we describe the encryption method, simple comparison between AES and RC4, and protocols we use to ensure that the data we selected in Section 4.2 can be encrypted and transmitted in a secure and proper way.

4.4 Simulation and Performance Analysis

4.4.1 Environmental Setup

The simulation was carried out between two workstations. The server runs on a desktop workstation with Pentium 4 2.53GHz CPU and 768MB of memory and the client runs on a Compaq R3000T notebook computer with Athlon64 3000+ CPU running at 1.8GHz and 512MB of memory. They are both connected to a router through CAT5 cables. No other machines are connected to the router.

4.4.2 Implementation

As shown in Figure 4.4, the server processes the source frame using JPEG encoding and selects data for encryption using the four different solutions mentioned in Section 4.2; then it passes the selected data to java encryption programs which implement different key length of AES ciphers and finally joins with the unencrypted data. All data will be sent over SRTP to the client.

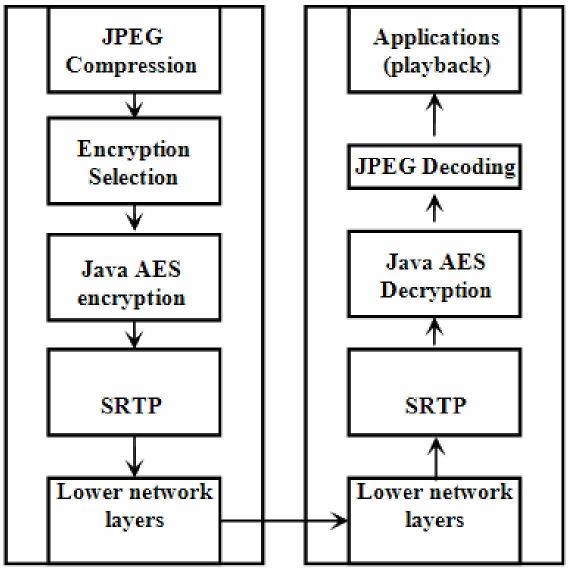


Figure 4.4: Flow chart of the secure communication in simulation

The desktop workstation sends Motion JPEG video stream at a constant rate of 30 frames per second. The video we used contains 500 frames and each frame has the resolution of 320x240. Each frame of the video is packed as the payload of a RTP packet and sent independently. The goal is to see at how many frames per second the client can playback the video received. A larger frame rate indicates less computation is involved since the CPU frequency is fixed. All numbers are based on an average of 10 runs.

4.4.3 Performance Analysis

4.4.3.1 Frame Rate Comparison

Figure 4.5 shows the real time playback frame rate at the client side using different data selection algorithms at the server side and various key lengths. The best frame rate (20.8 frames / second) with encryption and decryption is achieved when we use the Minimum Encryption with 128 bits of key. While with full encryption and 256 key lengths, the frame rate drops to 11.5 frames / second.

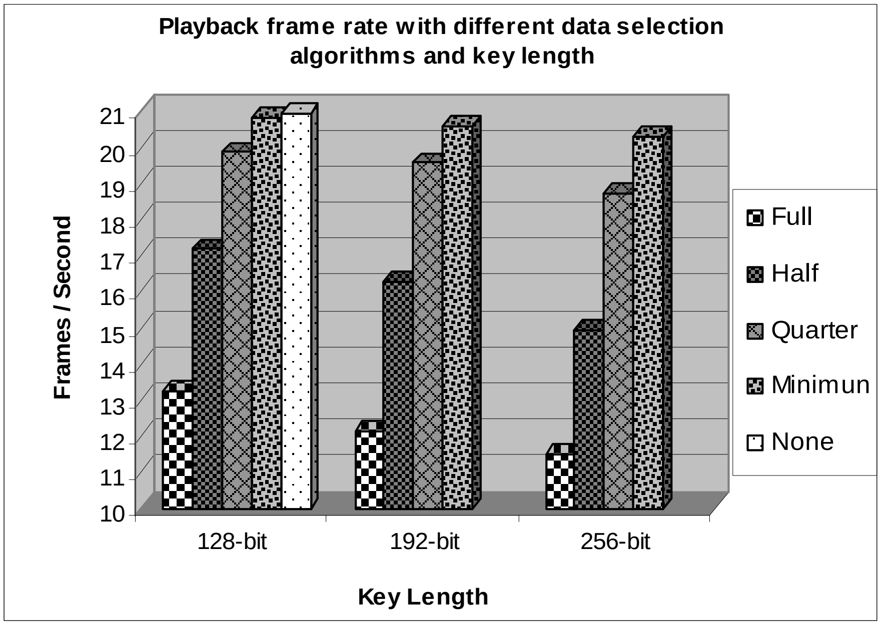


Figure 4.5: Frame rates comparison of selective data encryption

Performance of none encryption/decryption is plotted to the right of 128 bit key length encryption/decryption for easy comparison. As we can see, frame rates with Quarter and Minimum Encryption are close to that of none encryption/decryption while using short lengths of keys does not lift up performance much. This indicates that choosing different data selection algorithms for encryption can provide a more adaptive solution than changing the key length for different hardware environment. For example, choosing Quarter or Minimum Encryption will provide smooth enough (about 20 frames / second) playback on a client with about the same computation power as the client. Half and even Full Encryption can be applied to a more powerful desktop PC or workstation.

4.4.3.2 Computation Load Comparison

The Average CPU Load (ACL) is chosen to represent the computation load involved. As on the server side, most of the ACL is caused by video encryption alone, in all of experiments, ACL was always below 35%. However, on the client side, both decryption and JPEG decoding for playback need to be handled; thus the ACL on client side is the main concern.

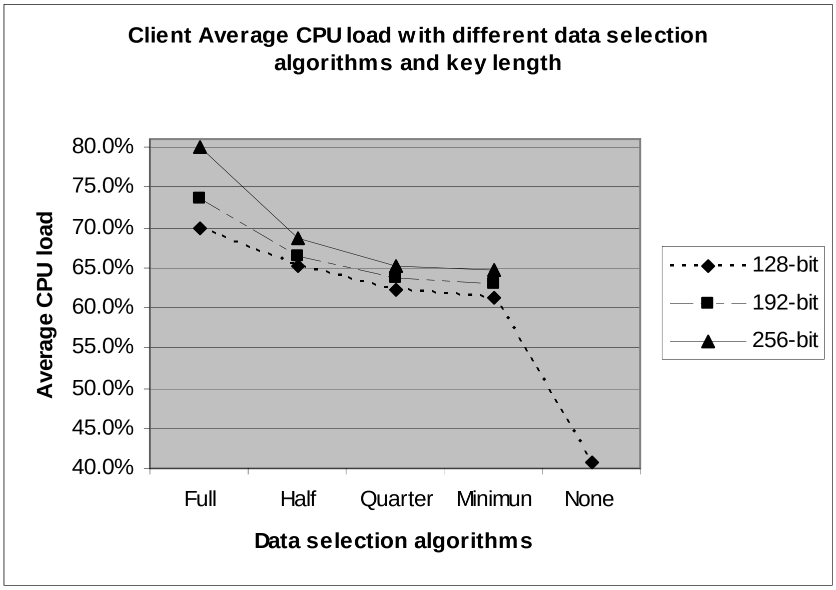


Figure 4.6: Client Average CPU Load comparison with selective data encryption

In Figure 4.6, the ACL of None encryption is plotted on the same curve with 128 bit encryption for easy comparison. For mobile devices such as a notebook computer, when ACL remains at a high value, battery will run out off power quickly. From Figure 4.6 we can see that the ACL drops relatively quickly from Full to Half encryption while slowly to other less encryption algorithms. If we also take Figure 4.5 into consideration, which shows the playback frame rate drops relatively less from Minimum to Quarter encryption than from Quarter to other heavy encryption algorithms, we can conclude that, for a mobile client, in order to obtain a certain level of security it is a good idea to choose either Minimum or Quarter encryption algorithm which cuts down the power consumption in a certain range while at the same time, achieves a relatively good playback frame rate.

4.5 Chapter Summary

In Chapter 4, we analyze the JPEG compression and encoding process in details in order to achieve multi-level security for encrypting bit stream of Motion JPEG video, which is a typical example of media streaming. We propose a set of four different solutions, each of which gives a reasonable balance and tradeoff between computation and security for different hardware environments. We further implement simulation and analyze the performance results. The results indicate that the four data selection algorithms for encryption can be adaptively applied to various computer systems with different computation power and power supply.

As for future work, further analysis of data selection for encryption can also be considered. Different data selection solutions for encryption could be used to obtain different image quality since an image of low quality will have fewer and smaller non-zero AC coefficients than an image of high quality. Also, similar idea could be used on MPEG video with further investigation and modification as frames in MPEG video are organized as Group of Pictures (GOP) which is different from independent frames in Motion JPEG. Different media coding methods, such as Multiple Description (MD) Code and Scalable Coding, can directly affect the amount and way of encryption. In MD, data segment are relatively independent to one another thus data selection and encryption could be relatively random; however, in Scalable Coding, some data segments are based on others and this fact should be mapped into data selection and encryption for better security and load balance concerns.

CHAPTER 5

DATA REDUNDANCY AND NODE MOBILITY

This chapter examines the extended use of different coding schemes (Scalable vs. Multiple Description or MD) in different video formats (MPEG vs. Motion JPEG or MJPEG) in a Mobile Ad Hoc Network (MANET) environment and proposes a set of redundancy solutions at different coding levels. Simulations and analysis demonstrate and explain the relationship among the coding schemes, video formats, level and amount of redundancy, mobility of mobile devices, CPU load and battery life and frame loss rate at the receiver side. Finally, two practical scenarios are discussed to illustrate these findings and proposed solutions.

Applications of real-time video streaming in Mobile Ad Hoc Networks are very complicated because they involve complex and intricate relationships among the different input and output values. Coding schemes, video formats, mobility of mobile devices and redundancy are considered to be input values of the applications because they can be changed directly to affect performance; while values such as average frame loss rate and CPU load are treated as output values that a user can monitor and perceive.

Under certain network conditions, different input values will lead to different output performance, but there is no fixed formula to directly express the relationships among them. This chapter explores how some of these values can affect others using both theoretical and experimental analysis and discusses how they can be applied to practical examples.

5.1 Video Formats and Coding Schemes

A. Video Formats

MPEG is currently the most popular video format. Its three levels of coding structure make it highly efficient in terms of video quality vs. data size. At the top level, or Level I, video frames are organized in Group of Pictures (GOP) in which an I-frame leads several P-frames (and a few B-frames if needed). An I-frame does not depend on the data in the preceding or following frames as opposed to P-frames which rely on the preceding I-frame in the same GOP. At this level, different GOPs are independent from one another. The Level II coding structure describes how frames within the same GOP are formed. As described above, the receiver is not able to reconstruct the P-frames unless the I-frame in the same GOP has been received. Level III of the coding structure determines how a single frame is composed of different components. For example, in a single I-frame all the other components depend on the DC coefficients in order to be decoded.

The lack of inter-frame predication in Motion JPEG (MJPEG) results in a loss of compression capability, but the independent frames may be preferred for video editing. Level I or GOP level is not available for MJPEG since it lacks the concept of GOP as in MPEG. In Level II coding, it is as simple as a sequence of individual frames, while Level III coding is similar to that of MPEG as they both deal with a single frame.

Table 5.1

Video formats and coding schemes

	Level I	Level II	Level III
MPEG	MD	Scalable	Scalable
MJPEG	N/A	MD	Scalable

B. Coding Schemes

Coding schemes describe how contents are encoded or decoded, emphasizing the relationship among different contents. Multiple Description or MD coding refers to the cases where the contents are complementary to one another and losing one does not lead to an inability to reconstruct others; while for Scalable coding, some of the content must be received first in order to reconstruct the rest. These coding schemes are logically mapped into different coding levels of MPEG and MJPEG video formats, as illustrated in Table 5.1. As shown in Table 5.1, MD appears at both coding Level I of MPEG and Level II of MJPEG. For Level I of MPEG, each individual coding description of MD is within a single GOP and thus losing one GOP does not affect reconstructing others (the scope an I-frame can affect is within the same GOP). In Level II coding of MJPEG, a single coding description is for an independent frame. Since MJPEG does not have the concept of GOP or I-frame, losing any frame does not cause an inability to reconstruct other individual frames.

5.2 Redundancy

Redundancy in video streaming refers to the content that is duplicated to ensure better playback performance when part of the original content is lost en route. Redundancy at different coding levels in different video formats will lead to various results, particularly with regard to the playback quality. Theoretical redundancy solutions are presented in this section and their performance is analyzed in the following simulation section.

A. Level I Redundancy

Level I redundancy can only be applied to the MPEG format and adds a certain percentage of duplicate GOPs. For example, 20% Level I redundancy adds 1 duplicated GOP from every 5 GOPs. Since the GOP is the largest coding structure unit in MPEG and the size of a duplicated GOP is larger than the units in Level II and III, so frequent link disconnections (due to the high mobility of devices) are likely to significantly degrade the performance of Level I redundancy when the amount of redundancy has not reached to a point to overcome the loss.

B. Level II Redundancy

For MPEG, Level II redundancy adds duplicates at the frame level within a GOP. As duplicating I-frames has a very similar effect to Level I redundancy due to the fact that the I-frame takes up more than half of the size of a GOP, here the redundancy is limited to only duplicating P-frames. Many current applications do not include B-frames, or bi-directional frames, and thus duplicating B-frames is not part of this research either. A 20% of Level II redundancy for MPEG indicates that one P-frame is copied from every five. It is relatively efficient to duplicate P-frames because of their slim size compared to I-frames.

In the case of MJPEG, 20% Level II redundancy duplicates one out of every five individual frames. This is likely to produce a performance that is close to that achieved using Level I redundancy for MPEG due to the similar coding structure at the relevant levels.

C. Level III Redundancy

Since a P-frame only records the differences between itself and the preceding I-frame in the same GOP, it is not relevant for Level III redundancy. Knowing that an individual frame in MJPEG is very similar to an I-frame in MPEG, redundancy at this level refers to duplicating the DC coefficients and a few AC coefficients (the most important content inside a single frame) depending on the amount of redundancy. 20% redundancy would refer to duplicating the DC coefficient and the first two pairs of AC coefficient symbols after the Zig-Zag scan. Redundancy duplicates at this level are so fine grained that are expected to survive even in high mobility MANETs though more computation is involved.

In this section, possible redundancy solutions at different levels were proposed for both MPEG and MJPEG and their theoretical performance was briefly analyzed. The next section reports on experiments conducted to test their performance.

5.3 Simulation

This section verifies the above expectations and analyzes the experimental results.

5.3.1 Metrics

The input metrics included:

- Mobility as V_{max} – each node, representing a mobile device, has a random velocity within the range [0, V_{max}].
- Redundancy – tested at three different levels ranging from 0% to 50%, in 10% increments.

The output metrics included:

- Avg_Frame_Loss_Rate – the number of frames unreconstructed at playback per second (unit frames-per-second or fps).
- Avg_CPU_Load – the average percentage of CPU load; the higher the load, the faster the battery is drained.

5.3.2 Tools and Settings

The hardware utilized for the experiment included:

- Desktop PC – Intel Core 2 Duo E6300 @ 1.83GHz/ 2GB DDR2 RAM/ 1TB Hard Drive/ Gbit/s NIC. This computer operated as a network simulation host running NS2 (Network Simulator 2), and the video stream was sent to the client notebook PC via a router.
- Router – D-Link DI-524. The router was set in an outside-traffic-free environment reserved for exchanging data between the desktop PC and the notebook PC via CAT5 network cables.
- Notebook PC – HP Pavilion DV6205US/ Core Duo T2250 @ 1.73GHz/ 1GB DDR2 RAM/ 80GB Hard Drive/ Intel Graphics media Accelerator 950/ 6 cell battery. This notebook PC represents average mobile computer hardware and was used here to receive video stream data from the desktop PC via the router and decode the video stream for playback. It ran solely on a 6 cell battery.

The software used for the experiment included:

- NS2 [NS2] – network simulator ns-allinone-2.30. The "benchmark" mobility Random Waypoint model was used. The setdest tool from the CMU Monarch group was used to generate the node traces.
- Simulation Scenario – this was set to an area of 1500m by 500m, with 30 mobile nodes moving under the Random Waypoint model each with a transmission range of 150m. AODV was the routing protocol used here. V_{max} was set to a set of

different values ranging from 2 to 10 m/s and the node pause time was set to a fixed 20 seconds. A source mobile node generated Constant Bit Rate traffic of 1Mbps or a packet of 512bytes every 4ms. When redundancy was applied, the traffic rate was set to a higher value according to the value of the redundancy.

5.3.3 Results and Analysis

A. MPEG Redundancy Performance

Simulation results in Figure 5.1 shows that Level I redundancy can help recover 1.65 (2.57-0.92) ~ 3.96 (5.46-1.50) frames per second when the max node velocity varies from 2m/s to 10m/s. This implies that redundancy is more necessary when mobile devices are moving at a high speed. For little or no Level I redundancy, the node speed can seriously affect the average frame loss; for example, increasing V_{max} from 2m/s to 10m/s would introduce a loss of 3 additional frames per second.

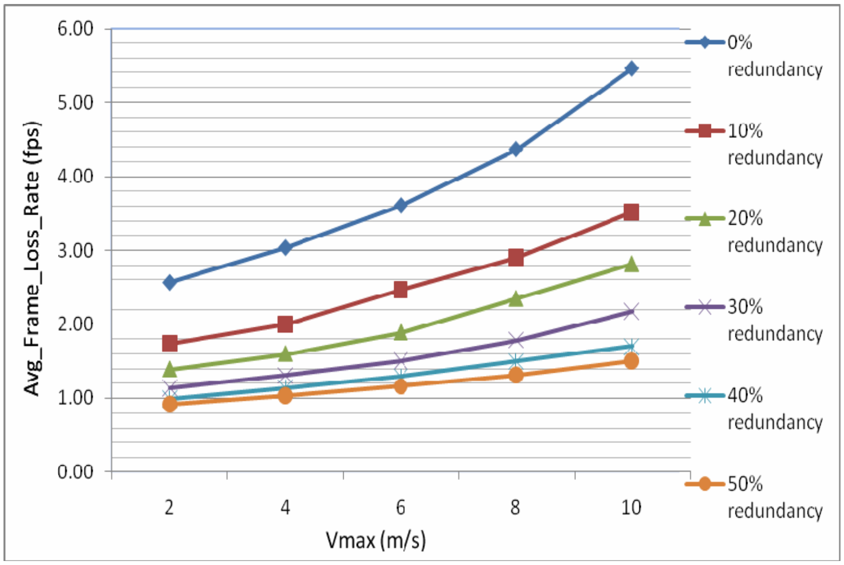


Figure 5.1: MPEG video performance with Level I redundancy

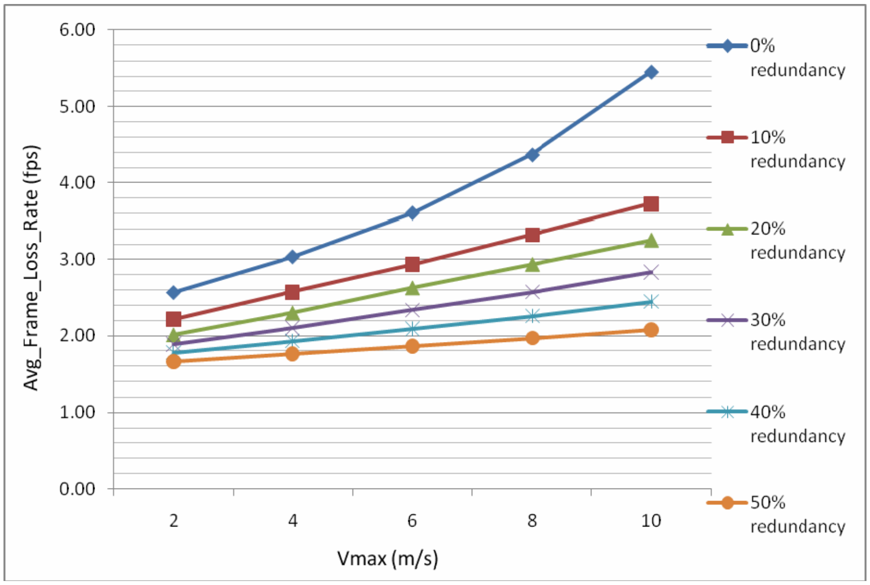


Figure 5.2: MPEG video performance with Level II redundancy

At first glance, the Level II redundancy in Figure 5.2 does not appear to perform as well as Level I redundancy at the same redundancy percentages, and all the curves are above 1.5fps in terms of Avg_Frame_Loss_Rate. This is due to the fact that adding redundancy to P-frames will not help when the leading I-frame in the same GOP is lost. However, this redundancy level is still valuable because it cuts down the total redundancy volume by duplicating P-frames instead of I-frames. Another useful finding is that the curves of Level II redundancy look more like straight lines than curves. This indicates that incrementing the node speed will have less impact on the Average Frame Loss Rate with Level II redundancy than with Level I. This is caused by the fine grained frame level redundancy control in Level II, which avoids the loss of large chunks of data when a link disconnection occurs.

Compared to the other two levels of redundancy, Level III, shown in Figure 5.3, can be more effective at reducing the frame loss rate. Here, the lines are more parallel than the other redundancy levels and this indicates that firstly changes in the node velocity have the least impact on Level III and secondly the frame loss rate is more linearly related to the amount of redundancy. Both results are due to the fact that Level III has the finest grained redundancy, making it more dynamic and adaptive to changing conditions such as high speed and link failures.

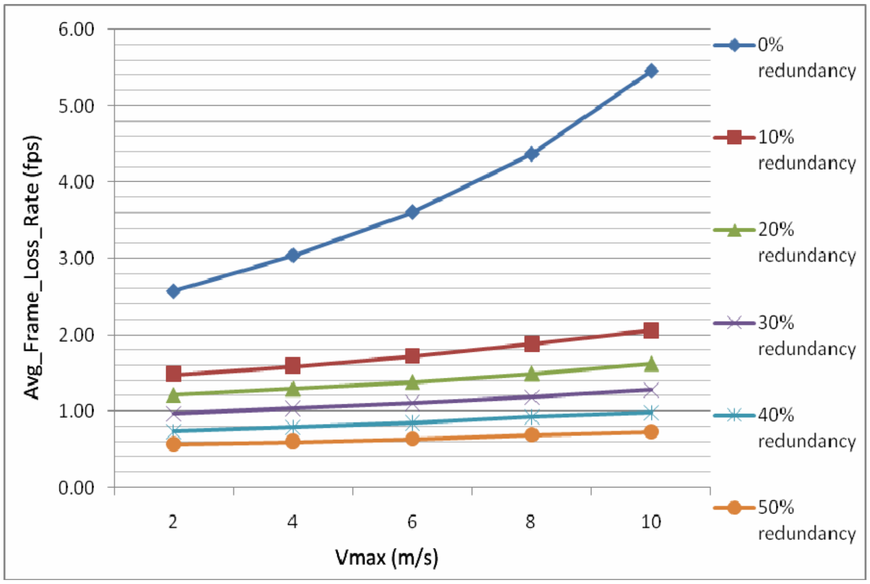


Figure 5.3. MPEG video performance with Level III redundancy

B. MJPEG redundancy performance

Compared to Level II redundancy on MPEG, MJPEG (Figure 5.4) achieves better results, with average frame loss rates that are lower than those of MPEG at all node velocities with the same amount of redundancy, which becomes particularly obvious as the node velocity increases. This is due to the fact that in the same MPEG GOP, all P-frames are based on the I-frame ahead of them. Thus, the loss of an I-frame affects the reconstruction of all the P-frames in the same GOP. In contrast, for MJPEG each frame is independent and is thus more able to adapt to the high link disconnection rate at high node speeds. However, this does not necessarily indicate that MJPEG is superior to MPEG, because losing a few P-frames may not degrade the playback as seriously as losing a single I-frame.

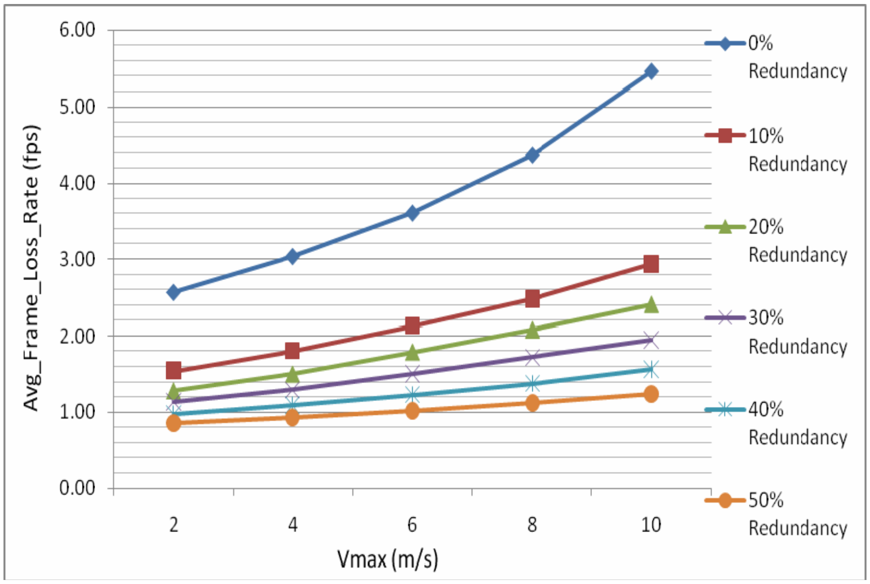


Figure 5.4. MJPEG video performance with Level II redundancy

Level III redundancy for MJPEG (Figure 5.5) generated the best performance among all the experiments; for all 5 different node velocities and 5 different amounts of redundancy, the average frame loss rate remained below 2 fps. This performance is due to the fine grain of redundancy level III and the use of independent frames in MJPEG video.

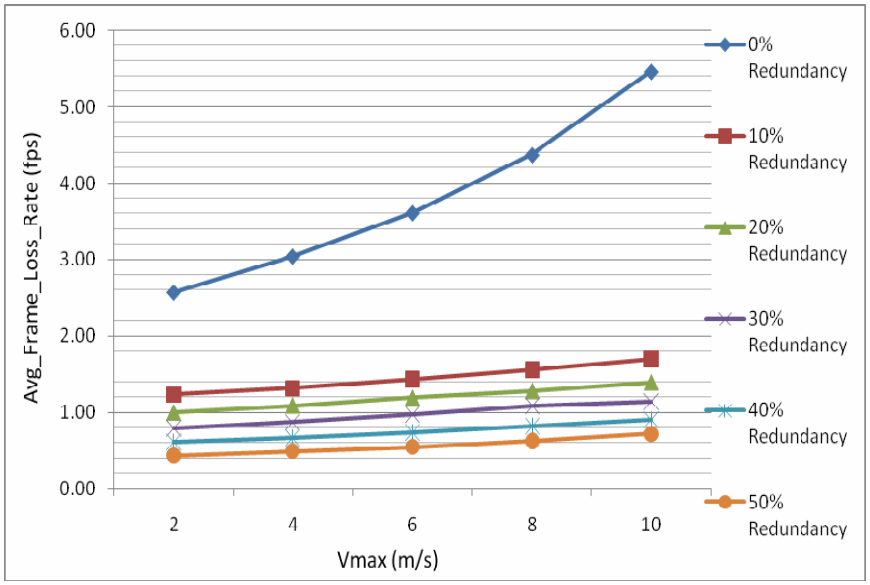


Figure 5.5. MJPEG video performance with Level III redundancy

C. CPU load and battery life performance

Comparing Figures 5.6, 5.7 and 5.8 (the verticle scale indicates the number of minutes for Battery Life results and the usage percentage for Avg_CPU_Load results), it is not surprising to note that as the redundancy level became more fine grained, more cpu computation was needed during the playback frame reconstruction and thus the battery drained faster, especially with Level III redundancy. The quantitative results of these experiments will be used to provide solutions to the two practical cases in the next section. The results obtained for MJPEG were very close to those of MPEG and thus are not presented.

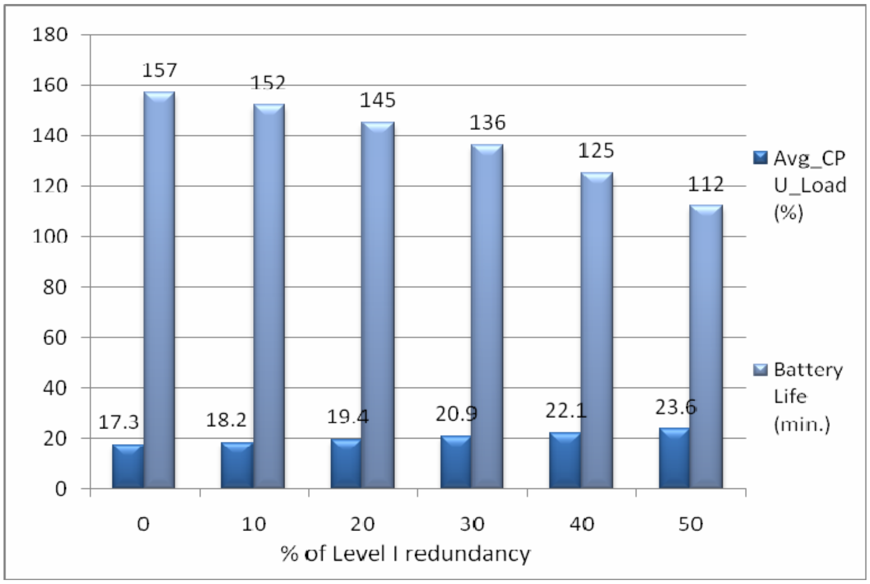


Figure 5.6. MPEG CPU and battery performance with Level I redundancy

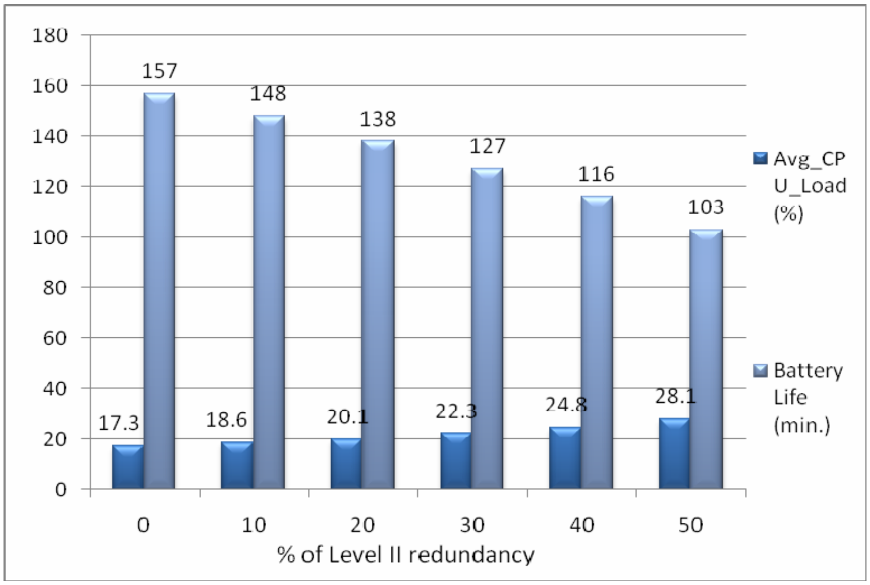


Figure 5.7. MPEG CPU and battery performance with Level II redundancy

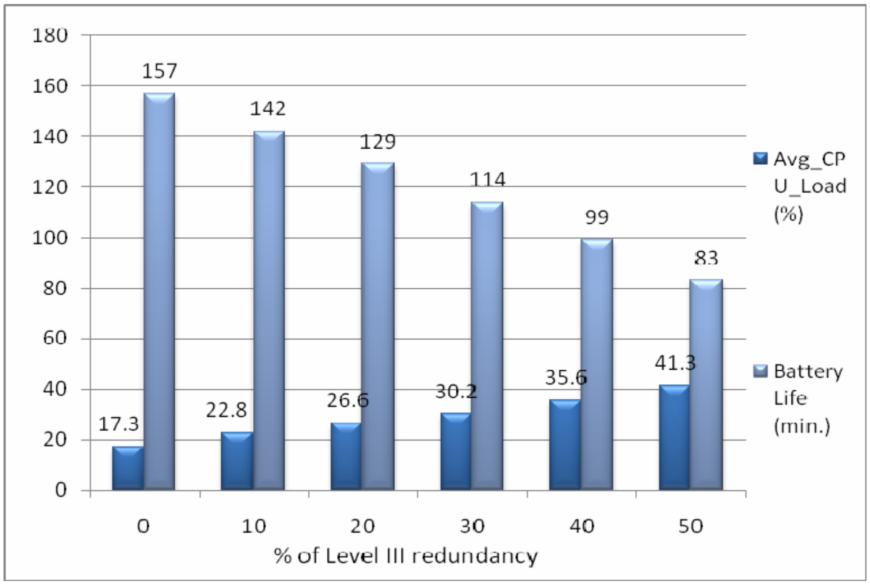


Figure 5.8. MPEG CPU and battery performance with Level III redundancy

5.3.4 Practical Scenarios

We use two practical scenarios to depict how our findings in this chapter can be applied to real-life situations.

Case 1: A pedestrian walking at less than 2m/s receives MPEG video stream using his notebook PC which runs on a battery and thus prefers to keep the CPU load below 20%.

To find out which redundancy level to use, Table 5.2 is constructed based on the data from Figures 5.1, 5.2, 5.3, 5.6, 5.7 and 5.8. Each pair in the table represents the values of (Avg_Frame_Loss_Rate, Avg_CPU_Load). All three of the highlighted pairs meet the requirement of a CPU load lower than 20%, so this pedestrian can then select

20% of Level I redundancy (the green pair), which generates the best performance at 1.39 frames lost per second.

Table 5.2

Avg_Frame_Loss_Rate and Avg_CPU_Load pairs for various redundancies at the device moving velocity of 2m/s

	10%	20%	30%	40%	50%
Level I	1.73, 18.2	1.39, 19.4	1.14, 20.9	0.99, 22.1	0.92, 23.6
Level II	2.22, 18.6	2.01, 20.1	1.89, 22.3	1.78, 24.8	1.66, 28.1
Level III	1.48, 22.8	1.22, 26.6	0.97, 30.2	0.74, 35.6	0.57, 41.3

Case 2: A car passenger who has his mobile device powered by his vehicle running at 22.5 miles/h (about 10m/s) in a downtown area wishes to have the best video streaming performance possible.

Similarly, Table 5.3 can be constructed to figure out the answer. All pairs in the table meet his requirements since there is no battery issue, but among them the green pair of 50% Level III redundancy gives the best performance of just 0.73 frames lost per second.

Table 5.3

Avg_Frame_Loss_Rate and Avg_CPU_Load pairs for various redundancies at the device moving velocity of 10m/s

	10%	20%	30%	40%	50%
Level I	3.52, 18.2	2.82, 19.4	2.18, 20.9	1.70, 22.1	1.50, 23.6
Level II	3.73, 18.6	3.26, 20.1	2.84, 22.3	2.45, 24.8	2.08, 28.1
Level III	2.06, 22.8	1.63, 26.6	1.28, 30.2	0.98, 35.6	0.73, 41.3

5.4 Chapter Summary

Based on the above theoretical and experimental analysis, we summarize our findings as follows:

1. Experiments show that regardless the level of redundancy, redundancy becomes a more important factor for improving performance when the nodes move at a higher speed.

2. A deeper level of redundancy is more effective than a shallower one, yet at the same time involves more computation and drains the battery faster.

3. For the two most popular video formats, redundancy tends to have more impact on MJPEG than on MPEG, especially when nodes are moving at a higher speed, because the structure of MPEG video is more inter-dependent (Scalable) and less adaptive than MJPEG independent frames (MD) when more link disconnections occur. However, this

does not imply that MPEG is less useful than MJPEG, because losing a few P-frames in MPEG may be less damaging than losing an individual frame in MJPEG.

The two practical cases, based on realistic daily life scenarios, demonstrate and explain how the above findings and experiments can actually be applied to the real world. Future work includes smart redundancy generation and distribution which controls the amount and forwarding of redundancy according to the content of the video and real-time network conditions.

CHAPTER 6

SECURE DATA DISTRIBUTION

Security is a critical issue in mobile ad hoc networks (MANETs), where mobile nodes communicate with each other over relatively unreliable wireless links with no fixed infrastructure. One example of this occurs on battle fields, where soldiers can wear cameras and wireless mobile devices to send or receive images/video from different angles or locations in order to obtain better observation of the area. Such an environment could be highly insecure since an enemy could place devices to eavesdrop on or compromise the soldier nodes. We consider the case that time sensitive multimedia is sent using multiple paths which can make better use of load balancing strategies, increase bandwidth and save node energy. However, an enemy node could listen in on more than one path at the same time, allowing it to collect sufficient shares of an image or video frame to recover the image. Our proposed algorithm is based on a quantitative analysis of security to protect against such multi-path eavesdropping. We prove that our algorithm can achieve better level of security by detouring traffic via paths without or with less vulnerable areas at the same amount of redundancy. On the other hand, we also prove, using simulation results and mathematical derivation, our algorithm can provide much more redundancy for better multimedia quality purpose while at the same time guaranteeing the same level of security.

In most cases, there is a trade off between security and reliability. Applying redundancy is a common way to increase reliability, e.g., by increasing the number of paths from the source to the destination or increasing the number of redundant pieces of data. However, this exposes data over a wider range and thus provides more opportunities for attackers. Finding the optimal balance between security and reliability is a critical issue.

In [Apostolopoulos 2001], the author explains his approach to this problem. He first divides the original data into T parts (or shares in his paper). These T shares then go through a (T, N) Secret Sharing process to generate (N-T) redundant shares and at the same time encrypt (N-T+1) shares, which finally produce a total of N shares that are to be sent via multiple paths. He gives the share allocation condition as below, where n_i is the number of shares allocated on the i_{th} path; m is the number of paths selected; T is the number of original shares of the message, and N is the total number of shares (or chunks) of data after redundancy is inserted:

$$\begin{cases} N - n_i < T, \forall i \in (1,2,...,m) \\ n_1 + n_2 + ... + n_m = N \end{cases} \qquad (1)$$

We are interested in these conditions and would like to see if they can provide a reasonable balance between redundancy and security.

$$r < 1/m \tag{2}$$

From (1) the author derives (2) (please refer to [Apostolopoulos 2001]) in which r is the allowed maximum redundancy. (2) tells us that the conditions proposed in (1) actually impose a boundary for redundancy. Whether a certain level of redundancy is proper or not depends on the application and Quality of Service (QoS) provided. In Figure 6 of [Apostolopoulos 2001], with an average node transmission range of 250m, in most cases the number of paths found is between 6 and 12. This means that the above conditions can provide 1/12 to 1/6 redundancies. We consider this boundary too stringent and does not significantly improve the reliability for real-time video streaming in an ad hoc wireless environment. For non-real-time traffic, this low level of redundancy might be enough since retransmission can help recover those shares that are lost. However, for real-time streaming such as video or audio, retransmission does not help achieve better Quality of Service (QoS). Thus, we need to find another balance point between security and reliability.

The above discussion shows us two limitations of what was proposed in [Apostolopoulos 2001]. First, it puts an unreasonable (at least for video streaming) value boundary on redundancy. The redundancy is set to a fixed value that is smaller than the boundary. This boundary is determined only by the number of paths used for traffic and has nothing to do with the ever changing network topology and historical traffic that was loaded onto the paths. Second, how the shares of traffic are loaded onto the selected paths is not clear. We consider this an important issue from the aspect of security, since the result of an attacker listening to nodes that are heavily loaded is totally different from listening to nodes that are lightly loaded.

The two issues above are highly related. A proper redundancy should dynamically change according to how secure the whole environment is. If the environment is denser, the probability that an attacker can hear more traffic will be higher, and thus the redundancy should be set to a lower value. This approach thus looks at the whole environment and needs the density feedback of all the paths involved. The solution to the second issue, how to allocate traffic (including the redundant traffic), depends partially on the regional density. If a certain area becomes dense and more nodes on the selected paths are close to one another, the chance of a nearby attacker hearing more traffic increases. Besides the regional density, traffic allocation is also related to historical traffic allocations. Paths or nodes that have not been loaded or only lightly loaded should have higher priority than those previously loaded or heavily loaded. In this way, traffic is reasonably distributed so that even if an attacker stays at a place where he is able to hear traffic from several nodes, he would only receive a limited portion of the traffic.

In order for the source node to obtain density information (both environmental and regional), nodes on the selected path need to report to the source the status of their local situation, that is, how many other selected nodes for traffic are close by. By collecting these regional density reports, the source node can calculate the environmental density in order to select a reasonable redundancy level. By combining the historical traffic allocation information and the regional density reports, the source is thus able to balance the traffic, routing more traffic onto the less used paths or those in a relatively sparse area, where there is less probability of a major breach of security.

Our goal is for the source node to send out data in a way such that data will be maximally distributed, detouring to avoid dense areas which are more vulnerable to hackers. Not only the current data distribution needs to be considered by the source node, but also how the historical data was distributed. The source node should avoid continuing sending data along the same "safe" paths without using other paths. Otherwise, an attacker can still increase the chance of catching more data by lurking around these "safe" paths. Therefore, the source node first needs to evaluate the currently available paths, then apply a historical traffic distribution algorithm to send out the traffic based on the path evaluation.

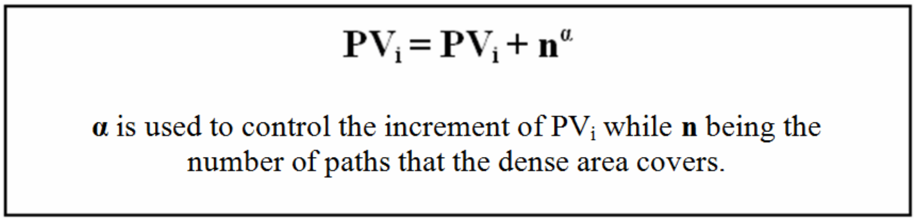
6.1 Path Evaluation

We now introduce a new metric: Path Vulnerability or PV. The PV of each path is originally set to an initial value PV_{in}. PV is then modified and maintained by the source node according to the path evaluation and historical data distribution information. A larger PV value for a path indicates that it has a higher probability of attack, where an eavesdropper could collect more data by staying around that specific path. PV_i of path i is initiated at source node as follows:

$$\boxed{\mathbf{PV_i = PV_{in}}}$$

Figure 6.1: Path Evaluation – Initiation

The source node needs to update the PV_i of path i after a share of data has been sent to path i. The update goes as follows:



$$\mathbf{PV_i = PV_i + n^{\alpha}}$$

α is used to control the increment of PV_i while **n** being the number of paths that the dense area covers.

Figure 6.2: Path Evaluation – Updates

If we take a look at the simple example in Figure 6.3, node C has a neighborhood that covers both the upper and bottom paths, thus n for the upper path where C is located will be 2. If on a single path there is more than one dense area, n should be set to the number of paths that the densest area covers among all the several dense areas along this path. α is set to 1 by default, but can be set to different values for different purposes. Please refer to Section 6.3 for more discussions about α.

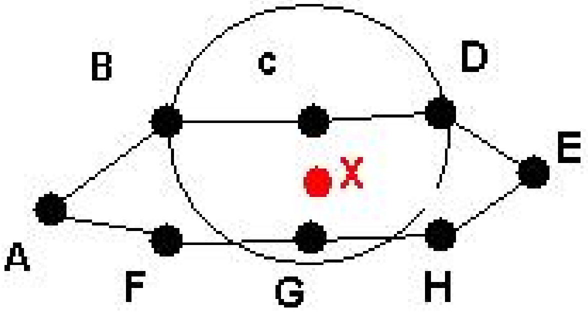


Figure 6.3: A simple example of MANET with single attacker

6.2 Data Distribution

The above evaluation of Path Vulnerability should be taken into account for traffic distribution. Therefore, the source node keeps records of the PVs of all the paths and sends out the next share of data according to the PV rankings, always sending it to the path that currently has the lowest value of PV in the PV list maintained at the source. After this share is sent, the PV of this specific path is increased by n^{α} according to the algorithm and thus the direction of the next share will be based on the updated PV rankings.

What if a certain node tries to receive more than its share of data and lies to the source node by saying it has very few neighbors yet in fact it is in a dense area? Since the source node receives local density information not just from one or two nodes, but from all the nodes from the selected paths, the neighborhood among nodes is self-proven. The source will be aware of the neighbor relationship between two nodes X and Y from Y even if X is trying to lie to the source node.

In such way, data shares are maximally scattered according to the real network topology and density and historical share allocation.

6.3 Quantitative Security Analysis

In order to compare the security level that our algorithm can achieve, we use a simple yet typical example below. The goal is to see the maximum number of data shares an attacker can hijack by eavesdropping on these paths. Figure 6.4 below shows 5 paths from source to destination. The circled area is the vulnerable dense area where an attacker can eavesdrop on data from 3 paths at the same time. n, the number of paths that the dense area covers is 3. The effects of different values of α on the Path Evaluation Updates (Figure 6.2) are discussed as follows.

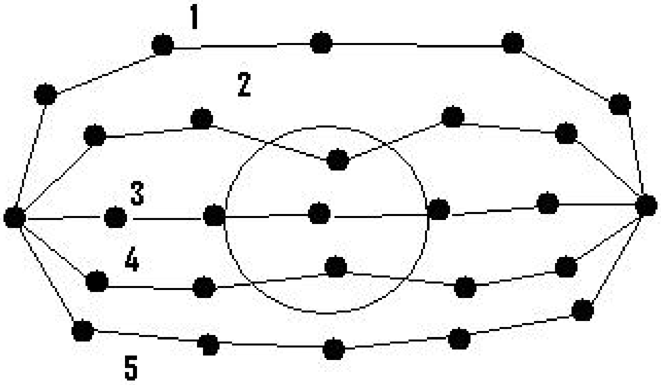


Figure 6.4: A typical ad hoc network with one dense area

1) $\alpha = 0$

In this case, the update equation becomes $PV_i = PV_i + 1$. This is the same way shares are sent in [Apostolopoulos 2001], where each path is treated the same even if any of the paths includes any vulnerable security areas. Each path transfers approximately N/5 shares of data, where N is the totally number of data shares. So in this case, the maximum number of shares an attacker can hijack is 3N/5 by staying in the dense area that covers 3 different paths in Figure 6.4.

2) $0 < \alpha < 1$

The vulnerable areas are taken into consideration in this case. However, the PV increment (n^{α}) of path 2, 3 or 4 is less than 3. Thus, each of the paths 2, 3 & 4 transfers fewer shares than a normal path such as 1 or 5 because of the vulnerable security area. In this case, the maximum number of shares an attacker can get is less than 3N/5 and the actual value depends on the value of α.

3) $\alpha = 1$

This is our default setting for α. The PV increment (n^{α}) of path 2, 3 or 4 is 3. This means the total number of shares for paths 2, 3, and 4 together is about the same as those on a normal path 1 or 5. So the maximum number of shares an attacker can hijack here is N/3.

4) $\alpha > 1$

This setting actually magnifies the effect of the vulnerable area by moving more shares to paths 1 and 5 from paths 2, 3 & 4. An attacker can thus only hijack less than N/3 shares in the vulnerable area, but more shares (although less than N/2) could be hijacked if the attacker moves close to either path 1 or 5. When $\alpha \rightarrow \infty$, no shares are sent through paths 2, 3 & 4 after the increment, and paths 1 and 5 each takes care of half of the shares.

Table 6.1

Max number of shares possibly eavesdropped according to different values of α

α	Max. # of shares eavesdropped
$\alpha = 0$	$n \times N/P$
$0 < \alpha < 1$	between $N/(P-n+1)$ and $n \times N/P$
$\alpha = 1$	$N/(P-n+1)$
$\alpha > 1$	between $N/(P-n+1)$ and $N/(P-n)$
$\alpha \to \infty$	$N/(P-n)$

A summary of the maximum number of shares an attacker is given in Table 6.1 above. In this table, n is the number of paths that the dense area covers; N is the total number of data shares, including the redundant shares; and P is the total number of paths from the source to destination nodes.

It is desirable to determine if the new algorithm can always guarantee that a lower maximum number of shares could be compromised. The derivation starts with inequation (3) which gives a condition when our algorithm (at $\alpha = 1$) has a smaller maximum number of shares than the algorithm in [Apostolopoulos 2001]. In other words, when (3) is valid, using our algorithm can achieve better security with the same amount of redundancy.

$$N/(P-n+1) \leq n \times N/P \qquad (3)$$

$$\Leftrightarrow \qquad 1/(P-n+1) \leq n/P \qquad (4)$$

$$\Leftrightarrow \quad P \leq P\times n - n^2 + n \qquad (5)$$

$$\Leftrightarrow \quad n^2 - n \leq P\times n - P \qquad (6)$$

$$\Leftrightarrow \quad n\times(n\text{-}1) \leq P\times(n-1) \qquad (7)$$

$$\Leftrightarrow \quad \text{as long as } (n-1) \geq 0, \quad n \leq P \quad (8)$$

Since the number of paths (n) that a dense area covers is always equal to or smaller than the total number of paths (P), (8) tells us that as long as $n \geq 1$, using our algorithm can achieve better security level with the same amount of redundancy. What if $n = 0$? If $n = 0$, it means the source node has not received any intermediate node reports of a dense area that covers one or more paths. In this case, each of the paths is totally independent and they do not affect each other. Thus, the source node can simply use the original share distribution in [Apostolopoulos 2001].

On the other hand, it is also an interesting issue to investigate how much more redundancy our new algorithm can offer than the one in [Apostolopoulos 2001] when both achieve the same level of security. We start with equation (9) below. The left part of the equation is the maximum number of shares eavesdropped using the original algorithm in which N' is the total number of shares, including the redundancy in the original algorithm; the right part of the equation is the maximum number of shares eavesdropped using our new algorithm in which N'' is the total number of shares, including the redundancy our algorithm provides.

$$n \times N'/P = N''/(P-n+1) \qquad (9)$$

$$\Leftrightarrow \qquad N''/N' = n \times (P-n+1)/P \qquad (10)$$

Since the total number of shares equals to the number of original data shares plus the redundant shares, we have

$$N' = T+r' \text{ and } N'' = T+r'' \qquad (11)$$

In (11), r' is the redundancy provided by original algorithm and r'' is the redundancy provided by our algorithm achieving the same level of security.

Take (11) into (10),

$$(T+r'')/(T+r')= n \times (P-n+1)/P \quad (12)$$

(12) shows the relationship between the amount of redundancy two algorithms can offer with the same level of security. Simulation results are provided in the next section to work with (12) together for proving the benefit of our algorithm.

We include a simple simulation in this section. The purpose of simulation is to know how much more redundancy our algorithm can provide with the same level of security in order to achieve better video/audio quality at the receiver side.

The simulation was implemented using the NS-2 network simulator. Simulation scenario is based on an area of 1000m by 1000m with a reflecting boundary in which 50 mobile nodes are moving around. All nodes have a maximum moving speed of 20m/s and radio transmission radius is 250m. We generate Constant Bit Rate traffic of 2mbps which simulates video streaming from source to destination nodes. All packets having more than 200ms delay are dropped and considered as lost.

Figure 6.5 shows that in 20 simulations, the number of paths found (P in equation (12)) from source to destination nodes is between 7 and 11 with the average of 8.85; the number of paths covered (n in equation (12)) by the dense areas is between 3 and 5 with the average of 3.75.

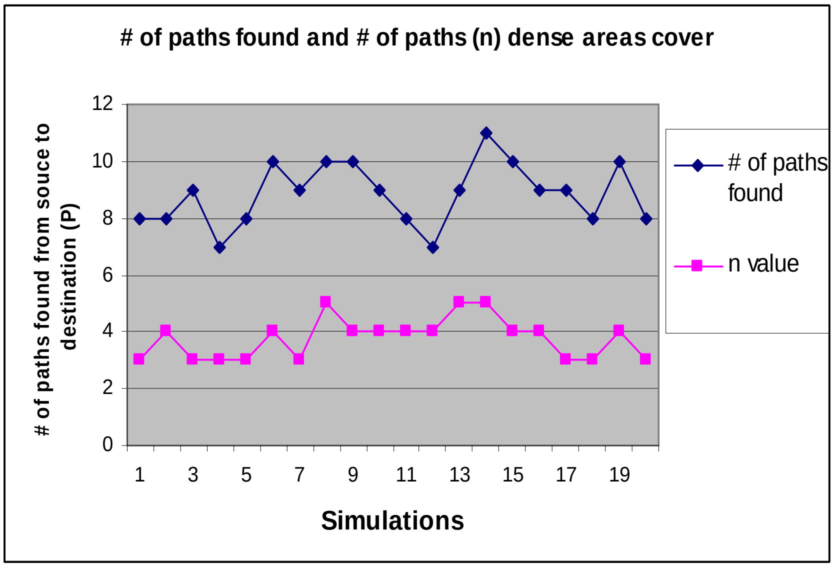


Figure 6.5: Number of paths found (P) and number of paths (n) dense areas cover in 20 simulations

If we take the simulation results of n and P into (12), we have (T+r'')/(T+r')=2.58 , or

$$r'' = 1.58\times T + 2.58\times r' \qquad (13)$$

T is the number of original shares of data. (13) tells us that our algorithm can achieve much more redundancy (r''), even more than the number of original shares, than the original algorithm while achieving the same level of security.

6.4 Chapter Summary

In this chapter, we first concluded that the limit of redundancy proposed in [Apostolopoulos 2001] in order to achieve a certain level of security is not reasonable for multimedia traffic, especially in an ad hoc network environment where packet loss can be of a high percentage. An algorithm that the source node can use to collect local density information from intermediate nodes is proposed. The source node can thus manage the data distribution by considering both vulnerable security areas and the historical data shares that were sent through each path. We proved mathematically that our new algorithm can achieve better security for video streaming applications. We also proved using simulation results and mathematical derivation that our algorithm can provide much more redundancy in order to achieve better performance at destination while guaranteeing the same level of security at the same time.

In the above discussion, we consider the impact of data redundancy on security with the respect of path security evaluation. Our future research will cover MANET attacks, dynamic path security and traffic load updates during data transmission.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

In the environment of Mobile Ad Hoc Networks, wireless communication signals among mobile nodes are exposed over the air. Although encryption and decryption techniques can help achieve better data confidentiality, extreme or full encryption and decryption does not fit in MANETs because of the very limited computation power and battery life of mobile devices. Further more, with wireless signal fading and interference, to provide secure and smooth media streaming such as video conferencing is very challenging in MANETs.

Our research focuses on the issues involved in the above scenarios. The Multi-path Neighbor Stability Routing in Chapter 3 finds the most stable or long lasting paths, for the purpose of better Quality of Service for media streaming, between a pair of source and destination nodes. These selected paths provide a relatively reliable and secure foundation for the data communication above them.

The data selection schemes for light-weight encryption and decryption in Chapter 4 well balance the trade-of between data confidentiality and limited computation power.

A set of four solutions works for different scenarios with different mobile computation and power conditions.

The effects of redundancy and mobility over video streaming in MANETs are explored in Chapter 5. Two practical examples are given in order to explain how much and at which encoding level, under certain mobility condition, redundancy should be applied in order to achieve better streaming quality.

To provide even higher level of security, in Chapter 6 we proposed a solution of secure data distribution schemes which ties closely to the dynamic path condition between the source and destination nodes. Redundant media data is added in at the source for the purpose of better streaming performance. The smart redundancy solution finds the right part of the original source data for redundancy and the amount of redundancy which makes a fit balance between data confidentiality and reliability at the source node.

As wireless applications become more pervasive, we believe that our work can help provide both more secure and smoother video media streaming in the environment of MANETs. The research outcome especially fits in Ad Hoc wireless media streaming applications such as higher level secure video conferencing between CEOs and managers in companies, or soldiers at battle fields, both of which require data confidentiality (sometimes against collude attacks) and smooth video streaming.

7.2 Future Work

Our future work includes the following research issues:

- In MANET routing, we are interested in routing considering load balance of the network. Load balancing refers to the algorithms that select paths or send out data packets in a way that no or few paths or mobile devices will be overloaded or exhausted due too excessive traffic and computation.
- For selective encryption, we would like to discover how smart selective encryption can be applied to MPEG video according to the priority of the importance of different data segments in the encoding process.
- Anti-attack is also one of the topics we are interested in. We will especially pay attention to conspiring attacks in which two or more attackers work together in a MANET at different locations. To defense against such attacks is much more challenging because if multiple attackers work in collaboration to find out the weak points of the network and lurk at those places, they can possibly collect data from multiple paths in order to reconstruct the original message.

REFERENCES

[ACM-H.263] ACM SIGGRAPH Education Committee, http://www.siggraph.org/education/materials/HyperGraph/video/codecs/H263.html

[Apostolopoulos 2001] John. G. Apostolopoulos, “Reliable Video Communication over Lossy Packet Networks Using Multiple State Encoding and Path Diversity”, VCIP, Jan. 2001, pp. 392-409

[Apostolopoulos and Trott 2001] John G. Apostolopoulos and Mitchell D. Trott, “Path Diversity for Enhanced Media Streaming”, IEEE Communications Magazine, August 2004, pp. 80-87

[Apostolopoulos et al. 2002] John Apostolopoulos, Wai-tian Tan, Susie Wee, "Modeling Path Diversity for Multiple Description Video Communication", ICASSP 2002, May 13-17, 2002

[Apostolopoulos 2004] John G. Apostolopoulos, "Secure Media Streaming & Secure Adaptation for Non-Scalable Video", IEEE International Conference on Image Processing (ICIP), 24-27 Oct. 2004, Singapore

[Apostolopoulos and Trott 2004] John G. Apostolopoulos, Mitchell D. Trott, "Path Diversity for Enhanced Media Streaming", IEEE Communication Magazine, Volume 42, Issue 8, 2004 Pages: 80-87

[Bacard 1995] Andre Bacard, “The Computer Privacy Handbook”, Peachpit Press, 1995

[Baugher et al. 2003] M. Baugher, D. MCgrew, M. Naslund, E. Carrara, K. Norrman http://www.ietf.org/rfc/rfc3711.txt

[Begen et al.2003] A. Begen, Y. Altunbasak, and O. Ergun, “Fast heuristics for multi-path selection for multiple description encoded video streaming,” IEEE ICME, pp. 517–520, July 2003.

[Borisov et al. 2001] N. Borisov, I. Goldberg, D. Wagner, “Intercepting mobile communications: the insecurity of 802.11”, MobiCom’01, Rome, Italy, July 2001, [Buchegger and Le Boudec 2002] S. Buchegger and J.-Y. Le Boudec, “Performance analysis of the CONFIDENT protocol”, MobiHOC'02, June 2002.

[Cano and Manzoni 2004] Juan-Carlos Cano and Pietro Manzoni, “Group mobility impact over TCP and CBR traffic in Mobile Ad Hoc Networks”, IEEE Infocom 2004.

[Chakeres and Belding-Royer 2004] Ian D. Chakeres and Elizabeth M. Belding-Royer, “AODV Routing Protocol Implementation Design”, Proceedings of the International Workshop on Wireless Ad Hoc Networking (WWAN), Tokyo, Japan, March 2004.

[Chen and Lee 2005a] Lei Chen & Chung-wei Lee, "Multi-level Secure Video Streaming over SRTP", Proceedings of the 43rd Annual ACM Southeast Conference (ACMSE), Kennesaw, Georgia, Mar. 2005

[Chen and Lee 2005b] Lei Chen and Chung-wei Lee, “Neighbor Stability Routing in MANETs”, IEEE Wireless Communications & Networking Conference (WCNC) 2005

[Chen and Lee 2006] Lei Chen, Chung-wei Lee & Jyh-haw Yeh, "Density-based Multi-path Secure Communication over Mobile Ad Hoc Networks", Proceedings of the 44th Annual ACM Southeast Conference (ACMSE), Melbourne, Florida, Mar. 2006

[Daemen and Rijmen 2002] John Daemen, Vincent Rijmen, “AES Proposal: Rijndael”, February 2002

[Dai 2004] Wei Dai, Cryptographic algorithms Benchmarks, www.eskimo.com/~weidai/benchmarks.html, 7/21/2004

[Das et al. 2001] Samir R. Das, Charles E. Perkins and Elizabeth M. Royer, “Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks”, IEEE Personal Communications Magazine Special Issue on Ad Hoc Networking, Feb 2001.

[Dejean et al. 1991] J. H. Dejean, L. Dittmann and C. N. Lorenzen, “String Mode-A New Concept for Performance improvement of ATM Networks," IEEE Journal on Selected Areas in Communications, Vol. SAC-9, pp. 1452-1460, No. 9, December 1991.

[Gogate et al. 2002] N. Gogate, D. Chung, S.S. Panwar, and Y. Wang, “Supporting image/video applications in a mobile multihop radio environment using route diversity and multiple description coding,” IEEE Trans. CSVT, pp. 777–792, Sept 2002.

[Heng et al. 2005] Brian A. Heng, John G. Apostolopoulos and Jae S. Lim, "End-to-End Rate-Distortion Optimized Mode Selection for Multiple Description Video Coding", IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP ‘05

[Hong et al. 1999] Xiaoyan Hong, Mario Gerla, Guangyu Pei and Ching-Chuan Chiang, "A Group Mobility Model for Ad Hoc Wireless Networks", In Proceeding of ACM/IEEE MSWiM'99, Seattle, WA, Aug. 1999, pp.53-60.

[Hu, L. 1993] L. Hu, "Distributed Code Assignments for CDMA Packet Radio Networks," IEEE/ACM Transactions on Networking, Vol. 1, no. 6, pp. 668-677, December 1993.

[Hu, Y.-C. et al. 2002a] Y.-C. Hu, D. B. Johnson and A. Perrig, "SEAD: secure efficient distance vector routing for mobile wireless ad hoc networks," WMCSA'02

[Hu, Y.-C. et al. 2002b] Y.-C. Hu, A. Perrig and D. B. Johnson, "Ariadne : a secure on-demand routing protocol for ad hoc networks," MobiCom 2002, September 2002.

[Hubaux et al. 2001] J-P. Hubaux, L. Buttyan and S. Capkun, "The quest for security in mobile ad hoc networks", MobiHOC'01, 2001.

[Iacono and Ruland 2002] Luigi Lo Iacono, Christoph Ruland "Confidential Multimedia Communication in IP Networks", IEEE International Conference on Communication Systems, Singapore, 2002

[Joe and Batsell 2002] Inwhee Joe and Stephen G. Batsell, "MPR-based Hybrid Routing for Mobile Ad Hoc Networks", Proceedings of the 27th Annual IEEE Conference on Local Computer Networks (LCN'02).

[Kim et al. 2004] D. Kim, C.-H. Min, and S. Kim, "On-demand SIR and Bandwidth-guaranteed Routing with Transmit Power Assignment in Ad Hoc Mobile Networks", IEEE Trans. Vehic. Tech., vol. 53, no. 4, July 2004, pp. 233-52

[Kong et al. 2001] J. Kong, P. Zerfos, H. Luo, S. Lu and L. Zhang, "Providing robust and ubiquitous security support for manet," ICNP 2001

[Koohi et al. 2003] A. Koohi, Nader Bagherzadeh and Chengzi Pan, "A fast parallel Reed-Solomon decoder on a reconfigurable architecture", 1st International Conference on Hardware/Software Codesign and System Synthesis, 2003

[Lee and Gerla 2001] S-J. Lee, M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks", ICC'01

[Lee and Liew 1993] T. T. Lee and S. C. Liew, "Parallel Communications for ATM Network Control and Management," Proc. of GLOBECOM'93, pp. 442-446, November 1993.

[Lee and Park 2004] Hyun-Yong Lee and In-Cheol Park, "A fast Reed-Solomon Product-Code decoder without redundant computations", Proceedings of the 2004 International Symposium on Circuits and systems, ISCAS '04

[Li and Drew 2003] Ze-Nian Li & Mark S. Drew, "Fundamentals of Multimedia", ISBN 0-13-061872-1, p. 253 ~ 265

[Liang, Y. et al. 2002] Y. Liang, E. Setton, and B. Girod, "Channeladaptive video streaming using packet path diversity and rate-distortion optimized reference picture selection," IEEE FifthWorkshop on Multimedia Signal Processing, pp. 420–423, Dec 2002.

[Liang, Y. J. et al. 2001] Y. J. Liang, E. G. Stainbach, and B. Girod, "Real-time Voice Communication over the Internet Using Packet Path Diversity", Proc. ACM Multimedia, Sept./Oct. 2001, pp. 431-40

[Lin and Liu 1999] C. R. Lin and J.-S. Liu, "QoS Routing in Ad Hoc Wireless Networks", IEEE JSAC, vol. 17, no. 8, Aug. 1999, pp. 1426-38

[Lou and Fang 2003] W. Lou, Y. Fang, "A survey on wireless security in mobile ad hoc networks: challenges and available solutions," book chapter in Ad Hoc Wireless Networking, Kluwer, May 2003

[Lou, W. et al. 2004] Wenjing Lou, Wei Liu and Yuguang Fang, "SPREAD: Enhancing Data Confidentiality in Mobile Ad Hoc Networks", IEEE Infocom 2004

[Maxemchuk 1975] N. F. Maxemchuk, Dispersity Routing in Store and Forward Networks," Ph.D. Dissertation, University of Pennsylvania, May 1975.

[Miu et al. 2003] A. Miu, J. Apostolopoulos, W. Tan, and M. Trott, "Low-latency wireless video over 802.11 networks using path diversity," IEEE ICME, pp. 441-444, July 2003

[NS2] http://www.isi.edu/nsnam/ns/

[Orchard et al. 1997] M. T. Orchard, Y. Wang, V. Vaishampanyan, and A. R. Reibman, Redundancy Rate Distortion analysis of Multiple Description Coding Using Pairwise Correlating Transforms," in Proc. IEEE Int'l Conf. Image Process (ICIP97), Santa Barbara, CA, Oct. 1997.

[Ozarow 1980] L. Ozarow, "On a source coding problem with two channels and three receivers," The Bell System Technical Journal, vol. 59, pp. 1909-1921, Dec. 1980.

[Papadimitratos and Haas 2002] P. Papadimitratos and Z. J. Haas, "Secure routing for mobile ad hoc networks," CNDS 2002, San Antonio, TX, January 2002

[Patel 2004] Sarvar Patel, Functional Requirements for 802.20 Security, Contributions to IEEE 802.20 Mobile Broadband Wireless Access (MBWA), 06/28/2004

[Pearlman et al. 2000] M.R. Pearlman, Z.J. Haas, P. Sholander, S. S. Tabrizi, “On the impact of alternate path routing for load balancing in mobile ad hoc networks”, MobiHOC, 2000

[Plotkin and Varaiva 1993] N. T. Plotkin and P. P. Varaiya, “Performance Analysis of Parallel ATM Connections for Gigabit Speed Applications", Proc. of IEEE INFOCOM'93, pp. 1186-1193, March 1993.

[Ravikiran and Singh 2004] Ghanta Ravikiran and Suresh Singh, “Influence of Mobility Models on the Performance of Routing Protocols in Ad-Hoc Wireless Networks”, IEEE VTC’ 04 (spring), Milan, Italy, May 17-19, 2004.

[Reza 1994] Fazlollah M. Reza., “An Introduction to Information Theory”, New York: McGraw-Hill 1961. New York: Dover 1994. ISBN 0-486-68210-2

[Schneier 1996] B. Schneier, “Applied Cryptography: Protocols, Algorithms, and Source Code in C”, New York: John Wiley & Sons, Inc. 1996. ISBN 0-471-12845-7

[Schulzrinne et al.1998] H. Schulzrinne, A. Rao, and R. Lanphier, “Real Time Streaming Protocol (RTSP),” IETF RFC 2326 (proposed standard), Apr. 1998 http://www.ietf.org/rfc/rfc2326.txt

[Shacham and King 1987] N. Shacham and P. King, “Architectures and Performance of Multichannel Multihop Packet Radio Networks," IEEE JSAC, vol. 5, no. 6, July 1987.
[Shah and Nahrstedt 2002] S. H. Shah and K. Nahrstedt, “Predictive Location-based QoS Routing in Mobile Ad Hoc Networks”, Proc. IEEE ICC’02, Apr. 2002, pp. 1022-27

[Sinha et al. 1999] P. Sinha, R. Sivakumar, and V. Bharghavan, “CEDAR: A Core-extraction Distributed Ad Hoc Routing Algorithm”, IEEE JSAC, vol. 17, no. 8, Aug. 1999, pp. 1454-65

[Smith 1997] Richard E. Smith, “INTERNET Cryptography”, Addison-Wesley, 1997

[Stalling 2002] William Stalling, “High-Speed Networks and Internets Performance and Quality of Service” second Edition, 2002

[Toh 2002] Ghai-Keong Toh, “Associativity-Based Routing for Ad-Hoc Mobile Networks”, Wireless Personal Communications 4: 103-139, 1997.

[Toh et al. 2002] Chai-Keong Toh, Minar Delwar, and Donald Allen, “Evaluating the Communication Performance of an Ad Hoc Wireless Network”, IEEE Transactions on Wireless Commnunications, Vol 1, No. 3, July 2002.

[Tsirigos and Haas 2001] A. Tsirigos, Z.J. Haas, “Multi-path routing in the presence of frequent topological changes”, IEEE Communication Magazine, Nov 2001

[Venken et al. 2002] K. Venken, I.G. Vinagre, R. Sigle and J.D. Cervera, “Enabling network redundancy in the radio access network”, 3rd International conference on 3G Mobile Communication Technologies, 2002

[Wang, K. and Li 2002] Karen H. Wang and Baochun Li, “Group Mobility and Partition Prediction in Wireless Ad-Hoc Networks”, IEEE Infocom 2002.

[Wang, Y. and Chung 1996] Y. Wang and D.-M. Chung, “Robust image coding and transport in wireless networks using non-hierarchical decomposition," presented in 3rd Int.Workshop Mobil Multimedia Communications, New Brunswick, NJ, Sep. 1996

[Wang, Y. et al. 1997] Y. Wang, M. T. Orchard, and A. R. Reibman, “Multiple description image coding for noisy channels by pairing transform coefficients," in Proc. IEEE 1997 First Workshop on Multimedia Signal Processing (MMSP97), Princeton, NJ, June, 1997.

[Wee and Apostolopoulos 2003] Susie Wee and John Apostolopoulos, "Secure Scalable Streaming and Secure Transcoding with JPEG-2000", International Conference on Image Processing, ICIP 2003

[Wiki-DCT] Wikipedia-DCT http://en.wikipedia.org/wiki/Discrete_cosine_transform

[Wiki-JPEG] Wikipedia-JPEG http://en.wikipedia.org/wiki/JPEG

[Wiki-Mjpeg] Wikipedia-Mjpeg, http://en.wikipedia.org/wiki/Mjpeg

[Wiki-MPEG] Wikipedia-MPEG, http://en.wikipedia.org/wiki/MPEG

[Wolf et al. 1980]] J. K.Wolf, A. Wyner, and J. Ziv, “Source coding for multiple descriptions”, the Bell System Technical Journal, vol. 59, pp. 1417{1426, Oct. 1980.

[Wu and Harms 2001] K. Wu, J. Harms, “Performance study of a multipath routing method for wireless mobile ad hoc networks”, 9th international symposium on modeling, analysis and simulation of computer and telecommunication system, 2001

[Yang et al. 2002] H. Yang, X. Meng and S. Lu, “Self-organized network-layer security in mobile ad hoc networks,” ACM WiSe'02, September 2002.

[Ye et al. 2003] Zhengqiang Ye, Srikanth V. Krishnamurthy and Satish K. Tripathi, “A Framework for Reliable Routing in Mobile Ad Hoc Networks”, IEEE Infocom2003

[Zhang and Mouftah 2005] Baoxian Zhang and Hussein T. Mouftah, “QoS Routing for Wireless Ad Hoc Networks: Problems, Algorithms, and Protocols”, IEEE Communications Magazine, October 2005

[Zhang et al. 2003] Y. Zhang, W. Lee and Y. Huang, “Intrusion detection techniques for mobile wireless networks,” ACM Wireless Networks Journal, Vol. 9, No. 5, Sep 2003

[Zhou and Haas 1999] L. Zhou and Z. J. Haas, “Securing ad hoc networks,” IEEE Network Magazine, vol. 13, no. 6, November/December 1999

[Zhu and Corson 2002] C. Zhu and M. S. Corson, “QoS Routing for Mobile Ad Hoc Networks,” Proc. IEEE INFOCOM ’02, June 2002, pp. 958-67