Universidade de Aveiro
2014

Departamento de Electrónica, Telecomunicações e Informática

Vahid Nazari Talooki

Encaminhamento Confiável e Energeticamente Eficiente para Redes Ad hoc

Reliable and Energy Efficient Routing for Ad hoc Networks

Programa de Doutoramento em Informática
das Universidades do Aveiro, Minho e Porto

universidade de aveiro

Universidade do Minho

U.PORTO

Universidade de Aveiro
2013
Departamento de Electrónica, Telecomunicações e Informática

Vahid Nazari Talooki

Encaminhamento Confiável e Energeticamente Efieiente para Redes Ad hoc

Reliable and Energy Efficient Routing for Ad hoc Networks

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Informática, realizada sob a orientação científica do Doutor Jonathan Rodriguez Gonzalez do Instituto de Telecomunicações.

Apoio financeiro da FCT com a referência SFRH / BD / 61375 / 2009.

To my mother and my father for their love.

o júri / the jury

presidente / president

Prof. Doutor Joaquim José Borges Gouveia
Professor Catedrático da Universidade de Aveiro

vogais / examiners committee

Prof. Doutor José Manuel Esgalhado Valença
Professor Catedrático da Universidade de Minho

Prof. Doutor Joaquim Arnaldo Carvalho Martins
Professor associado da da Universidade de Aveiro

Prof. Doutor Rui Luís Andrade Aguiar
Professor associado da da Universidade de Aveiro

Prof. Doutor Tasos Dagiuklas
professor Auxiliar da Universidade do Hellenic Open

Prof. Doutor Paulo Jorge Coelho Marques
professor Adjunto, escola Superior de Tecnologia, Instituto Politecnico de Castelo Branco

Prof. Doutor Jonathan Rodriguez Gonzalez
professor Auxiliar da Universidade de Aveiro

Prof. Doutor Klaus Moessner
Professional research Fellow, faculty of Engineering and Physical Sciences, University of Surrey, Reino Unido

agradecimentos / acknowledgments

Queria usar este pequeno espaço para deixar os meus mais sinceros agradecimentos aqueles que foram essenciais neste percurso.
Desde já agradeço ao meu orientador Dr. Jonathan Rodriguez Gonzalez pela orientação, ajuda e confiança demonstradas ao longo destes anos.
Durante estes anos tive a oportunidade de participar nos projetos europeus SMARTVISION e CODELANCE. Quero agradecer calorosamente à Fundação para a Ciência e a Tecnologia (FCT), SFRH / BD / 61375 / 2009.
Ao Instituto de Telecomunições pelos excelentes meios e recursos disponibilizados.
Aos colegas do grupo 4TELL pela entre ajuda e partilha de ideias.
Aos meus amigos pelos momentos de descontração passados juntos.
Aos meus pais pela inspiração que sempre foram para mim.

palavras-chave

Redes Ad hoc, Encaminhamento, equilibrio de carga, Energeticamente, Seguro, Redes sem Fios, Codificação de Rede, Random Codificação de Rede Linear.

resumo

Nas redes móveis *ad hoc* (MANETs), onde o comportamento cooperativo é obrigatório, existe uma elevada probabilidade de alguns nós ficarem sobrecarregados nas operações de encaminhamento de pacotes no apoio à troca de dados com nós vizinhos. Este comportamento altruísta leva a uma sobrecarga desequilibrada em termos de tráfego e de consumo de energia. Nestes cenários, os nós móveis poderão beneficiar do uso da eficiência energética e de protocolo de encaminhamento de tráfego que melhor se adapte à sua capacidade limitada da bateria e velocidade de processamento. Este trabalho de doutoramento centra-se em propor um uso eficiente da energia e protocolos de encaminhamento para balanceamento de carga nas redes ad hoc. Actualmente a maioria dos protocolos de encaminhamento existentes considera simplesmente a métrica da extensão do caminho, ou seja o número de nós, para a escolha da melhor rota entre fonte (S) e um nó de destino (D); no mecanismo aqui proposto os nós são capazes de encontrar várias rotas por cada par de nós de origem e destino e seleccionar o melhor caminho segundo a energia e parâmetros de tráfego, aumentando o tempo de vida útil da rede. Os nossos resultados mostram que pela aplicação deste novo mecanismo, os protocolos de encaminhamento *ad hoc* actuais podem alcançar uma maior eficiência energética e balanceamento de carga.

Para além disso, devido à natureza de difusão dos canais sem fio em redes ad-hoc, outras técnicas, tais como a Codificação de Rede (NC), parecem ser também promissoras para a eficiência energética. NC pode reduzir o número de transmissões, e número de retransmissões e aumentar a taxa de transferência de dados traduzindo-se directamente na melhoria da eficiência energética. No entanto, devido ao acesso dos nós intermediários aos pacotes em trânsito e sua codificação, NC necessita de uma técnica que limite as acessos não autorizados e a corrupção dos pacotes. Explorou-se o mecanismo de forma a oferecer um novo método de segurança que propõe um grau adicional de protecção contra ataques e invasões. Por conseguinte, os nós intermediários mal-intencionados irão encontrar pacotes em trânsito computacionalmente intratáveis em termos de descodificação. Adoptou-se também outro código que usa Luby Transform (LT) como um código de pré-codificação no NC. Projectado inicialmente para aplicações de segurança, este código permite que os nós de destino recuperem pacotes corrompidos mesmo em presença de ataques bizantinos

keywords Ad hoc networks, Routing, Load Balancing, Energy Efficiency, Security, Wireless Networks, Network Coding, Random linear Network Coding.

abstract In Mobile Ad hoc NETworks (MANETs), where cooperative behaviour is mandatory, there is a high probability for some nodes to become overloaded with packet forwarding operations in order to support neighbor data exchange. This altruistic behaviour leads to an unbalanced load in the network in terms of traffic and energy consumption. In such scenarios, mobile nodes can benefit from the use of energy efficient and traffic fitting routing protocol that better suits the limited battery capacity and throughput limitation of the network. This PhD work focuses on proposing energy efficient and load balanced routing protocols for ad hoc networks. Where most of the existing routing protocols simply consider the path length metric when choosing the best route between a source and a destination node, in our proposed mechanism, nodes are able to find several routes for each pair of source and destination nodes and select the best route according to energy and traffic parameters, effectively extending the lifespan of the network. Our results show that by applying this novel mechanism, current flat ad hoc routing protocols can achieve higher energy efficiency and load balancing. Also, due to the broadcast nature of the wireless channels in ad hoc networks, other technique such as Network Coding (NC) looks promising for energy efficiency. NC can reduce the number of transmissions, number of re-transmissions, and increase the data transfer rate that directly translates to energy efficiency. However, due to the need to access foreign nodes for coding and forwarding packets, NC needs a mitigation technique against unauthorized accesses and packet corruption. Therefore, we proposed different mechanisms for handling these security attacks by, in particular by serially concatenating codes to support reliability in ad hoc network. As a solution to this problem, we explored a new security framework that proposes an additional degree of protection against eavesdropping attackers based on using concatenated encoding. Therefore, malicious intermediate nodes will find it computationally intractable to decode the transitive packets. We also adopted another code that uses Luby Transform (LT) as a pre-coding code for NC. Primarily being designed for security applications, this code enables the sink nodes to recover corrupted packets even in the presence of byzantine attacks.

Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

AODV	Ad Hoc On-demand Distance Vector
DoS	Denial of Service
DREAM	Distance Routing Effect Algorithm
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing
DYMO	DYnamic MANET On-demand
E2AODVv2	Energy Efficient Ad Hoc On-demand Distance Vector version 2
GPS	Global Positioning System
GPSR	Greedy Perimeter Stateless Routing
HSR	Hierarchical State Routing
IMS	IP Multimedia Subsystem
LBDSR	Load balanced Dynamic Source Routing
MAC	Message Authentication Code
MANET	Mobile Ad hoc NETworks
NGN	Next-Generation Network
NRL	Normalized Routing Load
OLSR	Optimized Link State Routing Protocol
PDR	Packet Delivery Ratio
Qos	Quality of Services
RERR	Route ERRor
RREP	Route REPly
RREQ	Route REQuest
TORA	Temporary Ordered Routing Protocols
WPO	Weighted Path Optimality
WRP	Wireless Routing Protocol
WSN	Wireless Sensor Networks
ZRP	Zone routing protocol
NC	Network Coding

LNC	Linear Network Coding
RLNC	Random Linear Network Coding
LT	Luby Transform

CHAPTER 1

1 INTRODUCTION

Ad hoc networking paradigm is a key enabling technology for Next-Generation Network (NGN), having significant application in connecting a multitude of wireless nodes or being able to operate in a more advanced mode by locally connecting different "things" of different capabilities, such as mobiles, Personal Device Assistance (PDAs), and laptops, among others. However, a pivotal design requirement for these scenarios is energy efficiency; many of these devices will be battery powered placing a fundamental limit on network, and specifically node lifetime. In the introduction, we outline the importance and application of ad hoc networks in society and for NGNs. In particular Section 1.1 describes ad hoc networks usages for future networks, and provides an overview of energy efficiency and the load balancing problem as a key driver for the work in this thesis. Section 1.2 outlines the objectives of this PhD work while the focus of each chapter is presented in Section 1.3. Finally, the novel contributions of this thesis are summarized in Section 1.4.

1.1 Motivation

1.1.1 "Ad Hoc" as Key Enabler for Future Networks

Wireless networking technology has a profound effect on our everyday life due to its inherent features such as being mobile and ubiquitous in nature, and offering an inexpensive alternative to traditional wired networks. Wireless networking metaphorizes diverse networking topologies and capabilities: they may refer to a cordless alternative of the traditional fixed network, using a wireless infrastructure (e.g. a server, fixed routers, and access points) or may simply refer to an infrastructure-less solution that operates in "ad hoc" mode. In both cases, wireless networking technology can benefit from multi-hop communication in which a packet transverses multiple nodes to reach its destination. The

main difference between ad hoc networks and the traditional cellular networks is that no dedicated infrastructure exists for ad hoc networks. Therefore, in the latter, users have to collaborate in the routing process by storing and forwarding data packets on behalf of each other. Furthermore, the network should be designed for autonomous and distributed operation with capabilities for self-configuring, self-healing. These features of ad hoc networking technology make them attractive for NGNs.

1.1.1.1 Ad hoc networks fundamentals

An ad hoc network is a set of wireless mobile nodes forming a temporary network with a dynamic topology, without the aid of any established infrastructure or centralized administration. Each wireless mobile node operates not only as a host, but also as a router and forwards packets to other mobile nodes in the network that may not be within direct transmission range of each other. Therefore, by operating in "ad hoc" mode, a packet travelling from a source node toward a destination node may pass through multiple nodes to reach its destination.

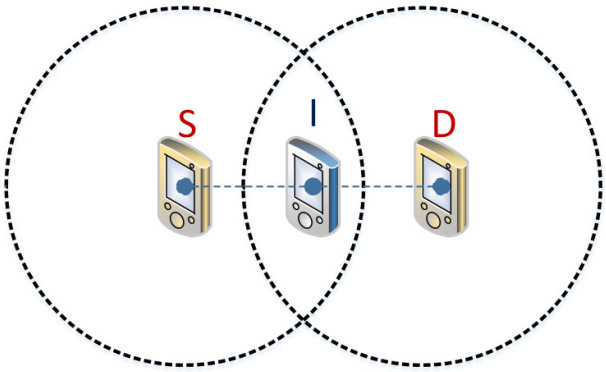

Figure 1.1: A simple scenario of ad hoc network

Figure 1.1 shows a simple ad hoc network where a source node (S) and a destination node (D) are not within direct transmission range of each other but still, via an intermediate node (I), can communicate with each other. Based on this approach, several nodes such as laptop, smart phone, cell phone, tablet, and so on, may be connected to each other via an ad hoc network, and can reach the internet through an access router, as illustrated by figure 1.2.

1.1.1.2 Characteristics

Ad hoc networks have several key characteristics, included but not limited to:

- Autonomous: Self-creation, self-organization, self-administration, self-healing, self-configuring, and distributed manner
- Infrastructure-less
- Multi-hop routing
- Dynamic network topology
- Device heterogeneity
- Nodes with limited resources: energy, bandwidth, memory, and processing power.

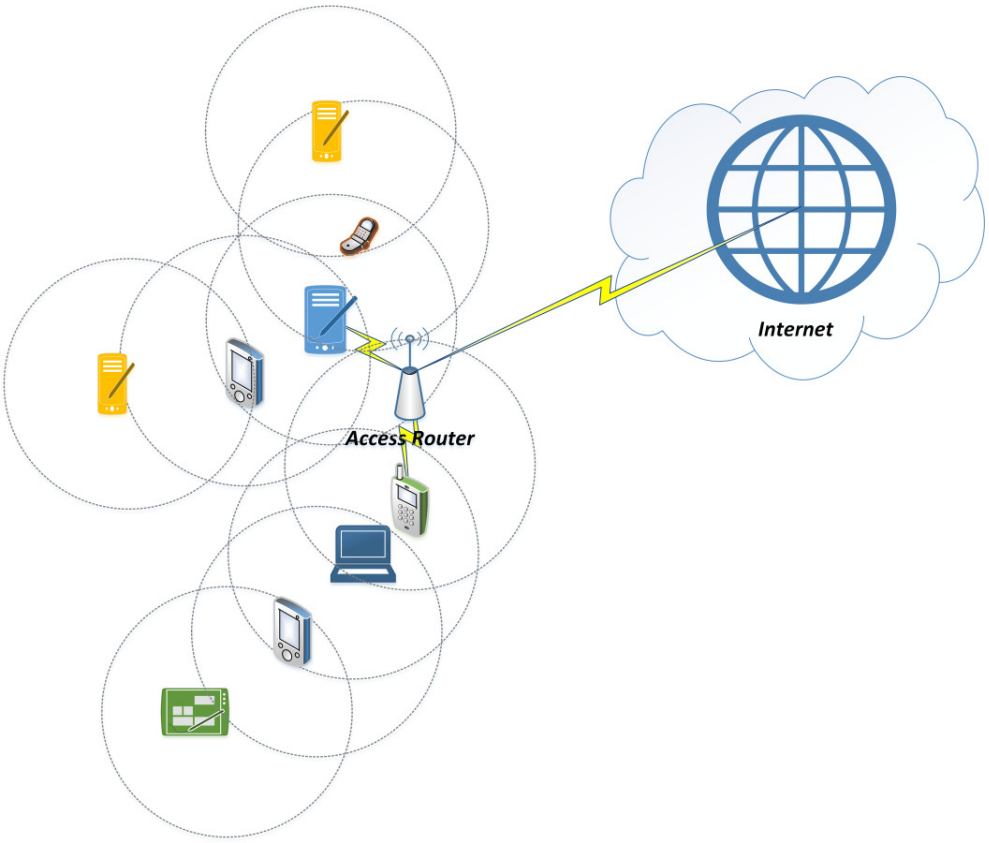

Figure 1.2: A simple scenario that shows ad hoc network nodes, via an access router, can reach to the internet

1.1.1.3 Applications

There are several applications for ad hoc networks, including but not limited to:

- Mobile communications: when there is little or no communication infrastructure or the existing infrastructure is expensive or inconvenient to use. By setting up an ad hoc network, wireless mobile users may still be able to communicate with each other.
- Military: for autonomous communications and operations in battlefields
- Emergency services: such as search and rescue operations, the recovery of networks after disasters, replacing current fixed infrastructure in case of earthquake, terrorist attack, and fire, ambulances communication, firefighting, and policing
- Commercial and civilian applications: such as e-commerce, vehicular services like road or accident guidance, weather conditions, and so on.
- Home and enterprise: such as wireless networking in house, conferences, and meeting rooms
- Personal Area Networks (PANs), Wireless Sensor Networks (WSNs), smart and wearable sensors in Body Area Networks (BANs)
- Other applications: virtual classrooms, entertainment, robotic pets, touristic information,…

1.1.1.4 Wide range of usage possibilities for future

In this context, ad hoc networks is foreseen to create new paradigms where mobile nodes can connect to form Mobile Ad hoc NETwork (MANET) [1], geared towards future emerging scenarios such as D2D communication, traffic loading, and energy saving among others. Mobile communications plays a pivotal role in our everyday life, with the number of mobile phone subscribers has already surpassing the 7 billion mark [2]. Ad hoc networking technology can connect a multitude of wireless tiny sensors or operate in a more advanced mode by locally connecting different “things” of different capabilities, such as Personal Device Assistance (PDAs), laptops, and so on, making then an ideal solution for the next-generation networks, envisaged for 2020. Apart from human communications, machines will also be connected and enabled to communicate either with humans or with

each other. Hence, these wireless nodes will be very diverse in terms of resources, such as memory, battery power, bandwidth, and processing power.

Also, one of the main benefits of ad hoc networks is related to extreme emergency cases like flood, earthquake, terrorist attacks, and disaster, among others. When the infrastructure goes down, ad hoc network can be a very efficient solution for setting up connectivity on demand.

1.1.2 Reliable Ad hoc Routing Protocols and Energy Efficiency

Pursuing high energy efficiency (EE) will be the trend for the design of future wireless networks. ICT sector including cellular networks, wired networks, and internet consumes more than 3% of the worldwide electric energy [1], which is expected to increase rapidly in the future. As an important part of ICT, wireless communications are also responsible for energy efficiency, considering exponential growth of mobile traffic in recent years. On the other hand, mobile terminals in wireless systems necessitate energy saving since the development of battery technology are much slower compared with the increase of energy consumption. Therefore, reducing energy consumption while meeting QoS requirements (such as data rate) in wireless networks and devices is an urgent task. Moreover, wireless networks and in particular ad hoc modes of operation have applications in extreme emergency scenarios, placing additional emphasis on EE design.

EE operation in ad hoc networks, not only suggests that we ought to take into account the energy profile of the routing nodes, but also the way we manage the network traffic. In legacy routing approaches, some nodes may become overloaded by packet forwarding, especially the ones located at the edge of the network. Therefore, EE and load balancing are crucial for ad hoc routing protocols to not only prolong the network lifetime but also to ensure its robust and continuous operation.

In legacy fixed networks, QoS has always been a predominant design. Metric, even though the internet could only deliver best effort. In wireless networks, QoS requirements are even more stringent, due to the variability of the channel and node mobility in MANETS. Therefore future ad hoc networks all need to be reliable, energy efficient whilst preserving QoS.

Traditionally, hard QoS routing guarantee in ad hoc networks is a Nondeterministic Polynomial time (NP)-Complete problem [3]. Therefore, the current works focus on providing a soft QoS routing guarantee for ad hoc networks as a more realistic solution. As a result, most of the existing routing protocols, proposed in MANET group of IETF [1], simply consider the path length metric when choosing the best route between a source (S) and a destination node (D). Although these approach can reduce the end-to-end delay in a communication session, they overlook the residual energy of the nodes as a decision criterion in the route selection process [4]. As such, they may not be efficient from energy perspective and lack a proper mechanism to handle and save critical nodes that are running out of power or being overloaded with traffic forwarding for other nodes. In this thesis, we consider energy efficiency and load balancing as the two main efficiency metrics.

1.1.3 Reliable Network Coding and Ad hoc Networks

In coding theory, there are three main coding families: source coding, channel coding and network coding (NC). The first, aims to compress information at the source, while the second introduces redundant bits at the link layer to guarantee reliable communications. On the other hand, the third consists in a coding process that takes place at the intermediate nodes in the network and at different layers of the network communication protocol stack.

In fact network coding (NC), first proposed by Ahlswede et al. [5], allows intermediate nodes not only to store and forward data packets, but also combine digital messages. Intelligent mixing of packets is achieved by performing algebraic operations on such packets. Therefore, NC generalizes the *store-and-forward* approach of ad hoc networks into the *store-code-forward* paradigm to achieve a higher throughput. As illustrated by figure 1.3, the traditional *store-and-forward* approach needs four transmissions in a simple unicast communication between three mobile nodes, while the *store-code-forward* paradigm of NC using the dimple XOR operation for encoding, needs only three transmissions.

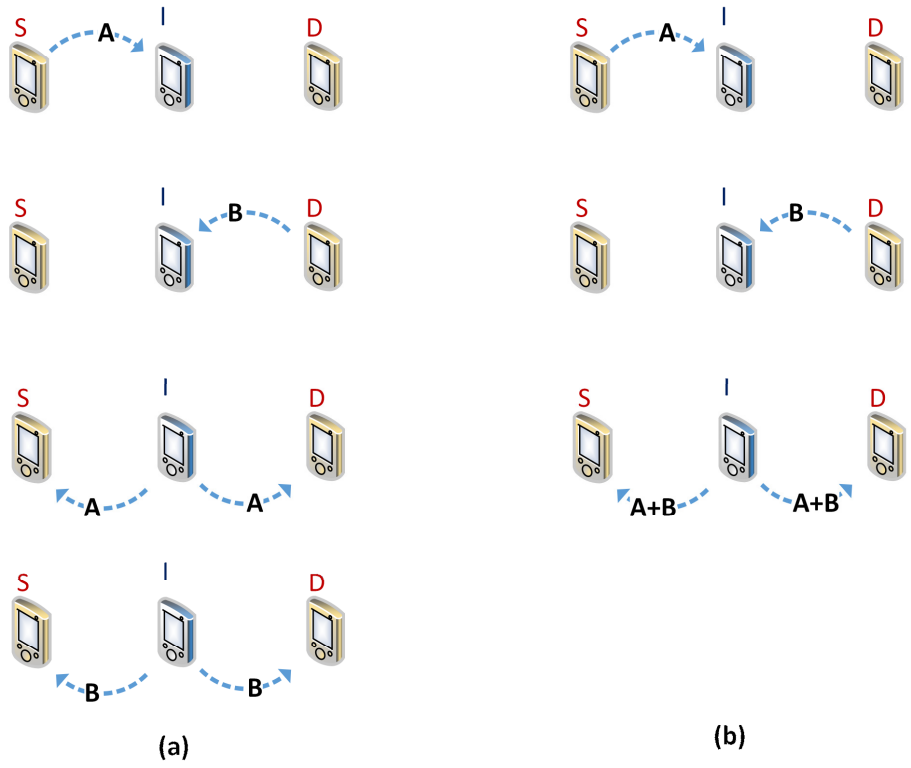

Figure 1.3: (a) the traditional *store-and-forward* versus (b) the *store-code-forward* paradigm of NC.

1.1.3.1 NC meets ad hoc networks requirements

Wireless multi-hop networks, e.g., ad hoc networks, have several certain features that provide a great opportunity for exploiting the benefits of NC in such networks. These features are : i) each transmission will be broadcasted in a certain zone via a shared wireless channel, ii) the transmission scheduling among nodes is conflict-free that means simultaneous single packet transmission or reception by any node of the network at any given time is not possible, and iii) instead of a continuous information flow for transmission, all communications are based on a multi-hop packet propagation in a store-and-forward transmission [6].

As an example to show the capability and the benefits of using NC in improving network throughput in a *multicast communication*, figure 1.4 shows a possible simple scenario for both the traditional store and forward mechanism and the elegant network coding *store-code-forward* paradigm. Here the source node S wants to multicast some

packets toward two sink nodes D_1 and D_2. Each packet like $p_{Time\ Stamp}^{Packet\ Number}$ has a packet number and time stamp that shows the packet number which was assigned by the source node and the time that packet was forwarded. For simplicity each packet (or symbol) is considered as one bit. A traditional store and forward mechanism, would achieve maximum throughput of 1.5 bits/s, but NC allows both D_1 and D_2 to achieve a rate of 2 bits/s at the same time which means more than 30% improvement in throughput for this scenario.

NC benefits are not limited only to multicast communications, as shown by figure 1.4, they may also decrease the number of transmissions in a unicast communication, as illustrated in figure 1.3.

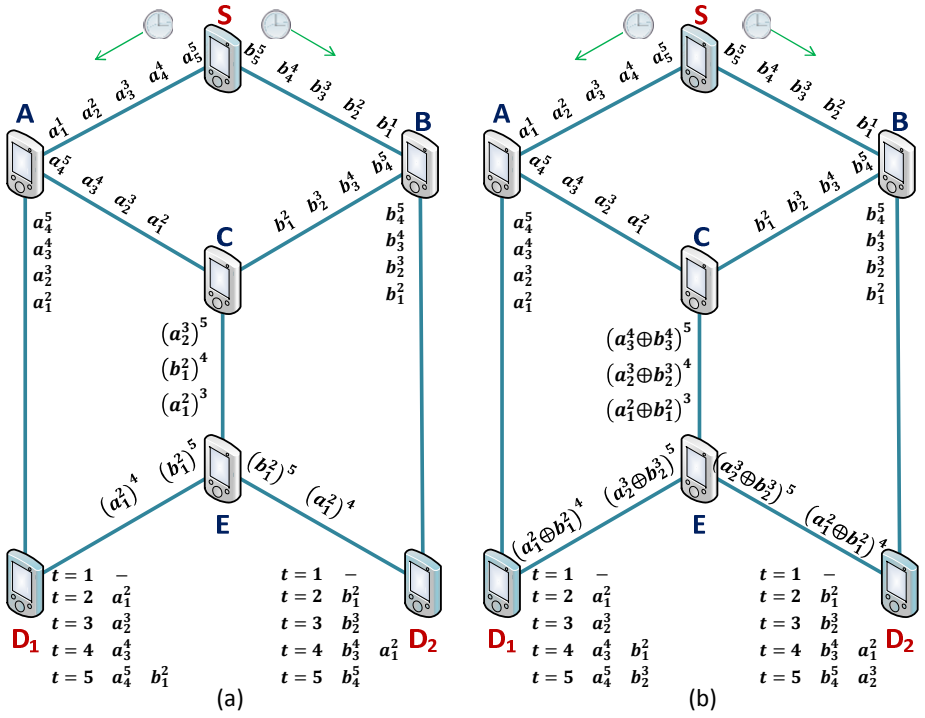

Figure 1.4: (a) Traditional store and forward mechanism in a multicast communication vs. (b) Network coding store-code-forward paradigm.

1.1.3.2 NC leads to energy efficiency in ad hoc networks

Wireless multi-hop networks can benefit from NC paradigm to reach higher energy efficiency due to several reasons:

i. *Reduced transmissions:* network coding in a multi-hop unicast and multicast transmission scenarios reduces the number of required transmissions, and the gain is reflected by the number of transmissions being proportional to energy efficiency [7]. Figure 1.3 (unicast transmission) and figure 1.4 (multicast transmission) show two of those scenarios.

ii. *Reduced re-transmissions*: network coding may decrease the number of re-transmissions in a network with noisy and erroneous channels. Consider a wireless ad hoc network consisting of an imperfect channel in which a source node S wants to communicate a message, e.g., a file, to a set of sinks. The sink nodes may fail to receive the sent file from the source node S due to erroneous links and therefore the whole file should be re-transmitted again. However, in the network coding paradigm, the source node can divide the file into h native packets and send out $h + r$ $(r \geq 0)$ linearly coded packets called *codewords*. All intermediate nodes send random linear combination of received codewords on the outgoing links. All sink nodes who receive any h linear combination of sent codewords are able to decode them and extract the native packets (i.e., the original file). In a network consisting of erroneous links, sink nodes still have a chance to receive these independent codewords from several different paths and there is no need for re-transmission of the whole file. Decreasing number of re-transmissions leads to higher energy efficiency.

iii. *Higher unit of data per unit of energy (bits/joule):* The main result of [5] was the enunciation of the max-flow min-cut theorem for information flow. In particular, the authors demonstrated that the multicast capacity is achieved by applying network coding. This achievement can bring a drastic change in improving the data throughput of networks. Achieving max-flow min-cut capacity means a higher data transfer rate and/or saving energy per each transferred bit of data (bits/ joule).

NC approach can be applied to several different layers of the network, protocol stack e.g., physical layer [8-12] and MAC layer [13-15]. As an example, NC is technology agnostic and can be applied to wireless multi-hop routing protocols. The coding operations of NC do not interfere with the routing protocols control packets and they effectively complement each other. NC applications are not limited to energy efficiency in wireless ad hoc networks [16-21] and include increasing network capacity, error detection and error correction in lossy networks, robustness, and security [5, 20, 22, 23].

1.1.3.3 Need for a reliable NC mechanism in erroneous networks

However, beside the aforementioned benefits of using NC for ad hoc networks especially in terms of energy efficiency, processing (recoding) the received data packets from the neighbors in the intermediate nodes and then forwarding them opens lots of challenging security issues in NC-based systems [24]. Codewords should pass through several relay terminals that give the opportunity to attackers and malicious nodes for unauthorized access, eavesdropping, and traffic analysis and monitoring. Consequently a security mechanism against eavesdropping attack is required for NC based networks.

Also in an NC based system, the sink node may receive *erroneous or corrupted* codewords due to lossy links in the network or malicious intermediate nodes; consequently, sink node may fail to successfully decode packets. Therefore a mitigation technique against packet corruption is needed too.

1.2 Objectives

Considering these aforementioned issues, this thesis will tackle the following objectives in the scope of this PhD work:

- ***Providing a concrete routing solution that is energy efficient:*** a key enabler of ad hoc networks is a routing approach that needs to be robust and lightweight (in terms of complexity) to support end-to-end connectivity in highly dynamic wireless environments. However, the effect of node/user mobility, dynamic topologies, frequent link breakages in the communication path, limitation on nodes resources such as battery energy, and lack of central point such as base stations or servers mean that routing in ad hoc networks can be a very challenging issue. In this context,

approaching a novel energy efficient routing protocol for ad hoc networks will be a key target.

- ***Routing approach for load balancing:*** load dispersion leads to traffic concentration on some nodes in ad hoc networks and may increase delay, packet loss, and node failure. Proving a traffic balanced routing mechanism that also be able to deliver adequate QoS is challenging, especially in a highly mobile environment. In this thesis, we propose a load balanced routing mechanism that is technology agnostic and can be applied to flat ad hoc routing scenarios.

- ***Proposing a mitigation technique against eavesdropping attacks in NC based systems:*** by using *store-code-forward* paradigm of NC, multi-hop networks can achieve a higher performance in terms of throughput and energy efficiency. However, wireless communications are susceptible to malicious relay nodes attempting to have unauthorized access to the transitive packets imposing several security threats and attacks such as eavesdropping, traffic analysis, and traffic monitoring. These issues serve as a springboard for providing reliable network codes that are robust against eavesdropping attacks.

- ***Novel mitigation technique against packet corruption in NC based systems:*** Using NC in multi-hop networks such as ad hoc networks is prone to packet corruption. The transitive packets can be corrupted due to erroneous channels in the network (such as erasure or noisy wireless links), or they can be corrupted by malicious byzantine relay nodes. These corrupted packets have a severe effect on the decoding process at the destination node. Therefore, an objective of this PhD work will be to increase the reliability of NC towards erroneous wireless channels (e.g., noise, fading, or erasure) and byzantine attackers.

1.3 Thesis Organization

- In Chapter 1, we outline the main motivation for investigating the problem of energy efficiency for wireless ad hoc networking technology. We presented the principal concepts and applications of ad hoc networking technology that may have the

potential to shape Next-Generation Networks. However, we argued that the pivotal design requirements for future networks and ad hoc networking technology is energy efficiency, since many of these devices will be battery powered, placing a fundamental limit on network, and specifically node lifetime. We also argued that in wireless networks that take advantage of multi-hop communications, e.g., ad hoc networks, the *store-code-forward* approach of network coding can be an effective technique to improve the energy efficiency. However, network coding mechanism is vulnerable to security attacks due to the role of intermediate nodes in packet processing. Therefore, to benefit from a NC technique for energy efficiency in ad hoc networks, we need to ensure to mould our NC design to make it reliable and secure against unauthorized accesses and packet corruption due to erroneous wireless channels and byzantine attackers.

- Chapter 2 provides the state of the art on wireless ad hoc networks. We categorized the current ad hoc routing protocols detailing their functionalities and applications. Also, the notion of providing an energy efficient and load balanced protocol for ad hoc networks was elaborated. We also compared several ad hoc routing protocols in terms of different QoS metrics. We discussed AODV, DSR, DSDV, and TORA routing protocols and compared them based on path optimality, delay, jitter, Normalize Routing Load, and Packet Delivery Ratio. We tested and analysed these protocols under several different situations categorized by changing the number of network's node, the speed of nodes, and pause time.

- In Chapter 3, we propose two new energy efficient and traffic balancing ad hoc routing protocols. The first routing protocol is based on the well-known Ad Hoc On-demand Distance Vector version 2 protocol (AODVv2). Our so called Energy Efficient AODVv2 ("E2AODVv2") has been designed towards energy efficiency and load balancing; this can help in detecting the nodes that reach a critical battery level in the network and so it switches the route in order to avoid network fragmentation and to achieve a higher network lifetime. E2AODVv2 achieves a higher performance with respect to energy consumption balancing, scalability, network lifetime, and the percentage of failed nodes in comparison to well-known baseline protocols such

AODV, DSR, DYMO, DSDV, and TORA. Also, the E2AODVv2 routing protocol outperforms specifically in scenarios where the load of the network is not balanced. We also propose a new load balanced and energy efficient ad hoc routing protocol, called Load Balanced Dynamic Source Routing (LBDSR). We defined the control packets, routing tables, and the route selection method, resulting in enhanced LBDSR performance with respect to traffic load balancing, energy consumption balancing, and route's reliability metrics.

- Chapter 4 starts by introducing the fundamentals of NC and presents the key security assumptions of NC systems as well as a detailed analysis of the security goals and threats. Presenting taxonomy of the existing NC security mechanisms and schemes reported in the literature, we propose new reliability and security mechanisms for NC based systems. In particular, we propose security mechanisms for handling eavesdropping attack and packet corruption.

- Chapter 5 includes the contributions of the thesis by presenting an overview of individual contributions per chapter and also our suggestions for future research.

1.4 Novel Contributions

The novel contributions of the thesis include the following:

- New energy efficient and traffic balancing ad hoc routing protocol based on the well-known IETF AODVv2 protocol. The new proposed routing protocol, E2AODVv2, achieves a higher performance with respect to energy consumption balancing, scalability, network lifetime, and the percentage of failed nodes in comparison to well-known baseline protocols.

- New energy efficient source routing protocol called Load Balanced Dynamic Source Routing (LBDSR). This version is more lightweight and can be customized to be more energy efficient or delay aware.

- Employing a serially concatenated NC based encoder in the source node, we propose a novel scheme, called Network Coding based Concatenated Code (NCCC), to make the underlying multi-hop path immune against any unauthorized accesses of an intermediate malicious node in NC based systems.

- By concatenating the NC encoder with another error correcting code, namely the Luby Transform (LT) codes [25], we generate a unique instanc of Batched Sparse (BATS) code that achieves higher error correction gains for NC based communications. BATS can be effectively employed here to mitigate against malicious nodes or erroneous channels corrupting the transitive packets.

The aforementioned contributions resulted in the following list of scientific publications:

Journal papers

1. V. Nazari Talooki, J. Rodriguez, and H. Marques, "**Energy Efficient and Load Balanced Routing for Wireless Multihop Network Applications**", International Journal of Distributed Sensor Networks, vol. 2014, p. 13, 2014.
2. T. A. Ramrekha, V. N. Talooki, J. Rodriguez, and C. Politis, "**Energy Efficient and Scalable Routing Protocol for Extreme Emergency Ad Hoc Communications**", Mob. Netw. Appl., vol. 17, pp. 312-324, 2012.

Book chapters

3. V. Nazari Talooki, H. Marques, J. Rodriguez, H. Água, N. Blanco, and L. Campos, "**E2DSR: Preliminary Implementation Results and Performance Evaluation of an Energy Efficient Routing Protocol for Wireless Ad Hoc networks**", in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering series. vol. 0077, ed, 2012.
4. T. A. Ramrekha, V. N. Talooki, J. Rodriguez, and C. Politis, "**Energy efficient and Scalable Routing Protocol for Extreme Emergency Ad Hoc Communications**", in Lecture Notes of the Institute for Computer Sciences,

Social Informatics and Telecommunications Engineering series. vol. 0077, ed, 2012.

Conference papers

5. Vahid N. Talooki, Riccardo Bassoli, Jonathan Rodriguez, Hugo Marques, Rahim Tafazolli ,"**Network Coding Based Surveillance Applications in Presence of Erroneous Channels**", Workshop co-located with the IEEE Symposium on Computers and Communications - ISCC 2014, 23-26 June 2014, Madeira Island, Portugal.
6. Riccardo Bassoli, Vahid N. Talooki, Hugo Marques, Jonathan Rodriguez, Seiamak Vahid, Rahim Tafazolli, "**LT Codes for Video Streaming in Burst Erasure Channels: An Energy Analysis**", Workshop co-located with the IEEE Symposium on Computers and Communications - ISCC 2014, 23-26 June 2014, Madeira Island, Portugal.
7. R. Bassoli, V. Talooki, H. Marques, J. Rodriguez, and R. Tafazolli, "**Energy Efficient Discovery of Neighbouring Nodes via Random Linear Network coding**", in IEEE 18th International Workshop on Computer Aided Modeling Analysis and Design of Communication Links and Networks (CAMAD), 2013, pp. 1-6.
8. V. Talooki, H. Marques, and J. Rodriguez, "**Energy Efficient Dynamic MANET On-demand (E2DYMO) Routing Protocol**", in IEEE World of Wireless-Mobile and Multimedia Networks - CONWIRE workshop, 2013, pp. 1-6.
9. V. Talooki, D. E. Lucani, H. Marques, and J. Rodriguez, "**Foreseen Risks for Network Coding based Surveillance Applications**", in 2nd ACM Workshop on "High Performance Mobile Opportunistic Systems", HP-MOSys, Barcelona, Spain, 2013, pp. 1-6.
10. D. Yang, J. Bachmatiuk, V. Talooki, and J. Rodriguez, "**Improved CodeCast For SVC Video Multicast**", in IEEE 18th International Workshop on Computer Aided Modeling Analysis and Design of Communication Links and Networks (CAMAD), 2013, pp. 1-6.
11. V. Nazari Talooki, H. Marques, J. Rodriguez, H. Água, N. Blanco, and L. Campos, "**E2DSR: Preliminary Implementation Results and Performance**

Evaluation of an Energy Efficient Routing Protocol for Wireless Ad Hoc networks", presented at the Proc. of the 6th International Mobile Multimedia Communications Conference (MOBIMEDIA), Lisbon, Portugal, September 2010.

12. T. A. Ramrekha, V. N. Talooki, J. Rodriguez, and C. Politis, "**Energy efficient and Scalable Routing Protocol for Extreme Emergency Ad Hoc Communications**", presented at the Proc. of the 6th International Mobile Multimedia Communications Conference (MOBIMEDIA), Lisbon, Portugal, September 2010.
13. V. Talooki, H. Marques, J. Rodriguez, H. Agua, N. Blanco, and L. Campos, "**An Energy Efficient Flat Routing Protocol for Wireless Ad Hoc Networks**", in Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on, 2010, pp. 1-6.
14. Panaousis., A. Ramrekha, K. Birkos, C. Papageorgiou, V. Talooki, G. Matthew, C. T. Nguyen, C. Politis, T. Dagiuklas, and J. Rodriguez, "**A Framework supporting Extreme Emergency Services**", presented at the ICT-MobileSummit 2009 Conference Proceedings.
15. V. N. Talooki and J. Rodriguez, "**Jitter based comparisons for routing protocols in mobile ad hoc networks**", in International Conference on Ultra-Modern Telecommunications, ICUMT '09, pp. 1-6.
16. V. N. Talooki, J. Rodriguez, and R. Sadeghi, "**A load balanced aware routing protocol for wireless ad hoc networks**", in Telecommunications, 2009. ICT '09. International Conference on, 2009, pp. 25-30.
17. V. Talooki and J. Rodriguez, "**Load Balanced DSR Protocol for Wireless Ad Hoc Networks**", presented at the 7th Conference on Telecommunications 2009 (Conftele 2009), Santa Maria de Feira, Portugal, May 2009.
18. V. Talooki and J. Rodriguez, " **Quality of Service for Flat Routing Protocols in Mobile Ad hoc Networks**", in 5th International Mobile Multimedia Communications Conference, London, UK, 7-9th of September 2009.

CHAPTER 2

2 FUNDAMENTALS ON AD HOC ROUTING PROTOCOLS

This chapter introduces the fundamental literature concepts of ad hoc networks that will act as a lunch pad for the proposed work in this PhD study. The chapter reviews the routing protocol concept for ad hoc networks and the motivation for energy efficient and load balanced protocol in ad hoc networks. In this chapter, we will also compare several ad hoc routing protocols in terms of different QoS metrics.

2.1 Ad hoc Routing Protocols

Ad hoc networks can appear in more complex scenarios which have a complicated and dynamic topology. In this case, any two nodes who want to start a communication may have several possible paths between each other via multi-hops routes, as illustrated by figure 2.1. Therefore, we need an *ad hoc routing protocol* for finding and maintaining a route between any desired source and destination nodes in the network.

Routing in ad hoc networks is a very challenging issue due to nodes mobility, dynamic topology, frequent link breakage, limitations of nodes (memory, battery, bandwidth, and processing power), and lack of central point like base stations or servers. On the other hand, there are several performance metrics and that should be satisfied in an ad hoc network such as end-to-end data delivery throughput, average end-to-end delay, and path optimality. Each protocol can satisfy some of these metrics, but with drawbacks towards others. Therefore, by comparing different ad hoc routing protocols we can extract important information surrounding the performance of these protocols in different scenarios.

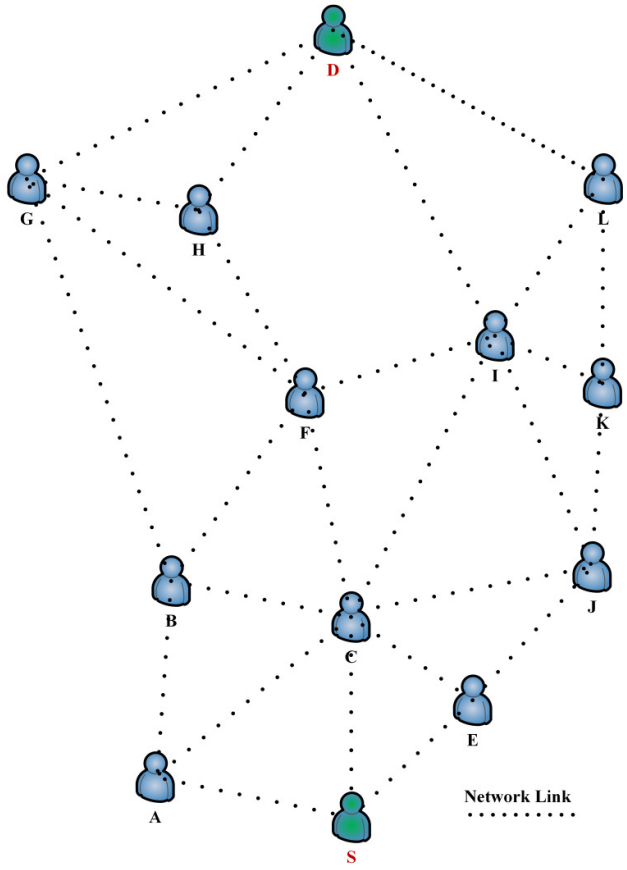

Figure 2.1: There are several possible multi-hop routes between S and D

Many routing algorithms have been proposed for ad hoc networks in the literature [26-33]. Survey paper [34] categorized and reviewed several of them. These routing protocols can be divided into several categories based on various criteria. Also, there is an active group, called Mobile Ad-hoc NETwork (MANET), in Internet Engineering Task Force (IETF) [1] regarding ad hoc networks. In this chapter, we briefly review the categories of ad hoc routing protocols.

2.1.1 Flat Routing Protocols

Flat routing protocols in ad hoc networks adopt a flat addressing scheme which means all nodes participating in the routing process play an equal role. Flat routing protocols may generally be classified into two main categories: i) proactive routing protocols and ii) reactive routing protocols.

2.1.1.1 Proactive protocols

This type of ad hoc routing protocols attempt to find and maintain consistent, up-to-date routes between all source-destination pairs regardless of the use or need of such routes. Therefore, each node maintains one or more tables to store routing information (table driven protocols). The proactive protocols require periodic control messages to maintain routes up to date for each node. Routing techniques for the proactive protocols are either i) *Distance Vector (DV)*, ii) *Link State(LS)*, or iii) a mixture of DV and LS [35, 36].

Destination Sequenced Distance Vector

DSDV was introduced in [26]; it is a proactive table-driven protocol based on the classical Bellman-Ford algorithm. In the DSDV, every node in the network maintains a routing table. A routing table has all of the possible destinations nodes and the number of hops to each destination. Each entry in the routing table has a sequence number assigned by the destination node that implies the freshness of that route; thereby, avoiding the formation of routing loops. By periodically update messages, routing tables maintain consistence. In order to avoid a growing network traffic, that can be caused by these update messages, route updates can employ two possible types of packets. The first is known as *full dump.* This type of update packet includes all available routing information. Smaller *incremental* packets include only that information which has changed since the last full dump [37]. New route broadcasts contain the address of destination, the number of hops to reach the destination, the sequence number of the information receipt regarding the destination, as well as a new unique sequence number for the new route broadcast. The route which is labelled with the most recent sequence number is always used [38, 39].

Optimized Link State Routing Protocol

OLSR [28] uses link state mechanism. It is an optimization of the classical link state algorithm that uses multipoint relays (called MPRs) for routing operations. The MPRs are selected nodes which forward broadcast messages during the flooding process. In OLSR, link state information is generated only by nodes selected as the MPRs. An MPR node reports only links between itself and its MPR selector; therefore, partial link state information is distributed in the network. This information is then used for route

calculation. The OLSR provides optimal routes in terms of number of hops and it uses two messages [40]: i) *Hello Messages:* are used by each node to find its one hop neighbours and its two hop neighbours through their responses. The sender can then select its MPRs based on the one hop node such that there exists a path to each of its two hop neighbours via its MPRs. ii) *TC (Topology Control) Messages:* are emitted by the MPR nodes for all other nodes and contain the MPR selectors (nodes have selected it as an MPR node).

Wireless Routing Protocol

WRP that applies to both distance vector and link state techniques was introduced in [33] for the first time. The WRP is a subset of a general class of Path-Finding Algorithms (PFA). The PFA defined as the set of distributed shortest-path algorithms that compute the paths using information concerning the length and second-to-last hop (predecessor) of the shortest path to each destination. The WRP also addresses the problem of avoiding short-term loops that can still be present in such algorithms for the paths specified by the predecessor node. The protocol uses the following four data structures: i) Distance table, ii) Routing table, iii) Link-cost table, and iv) Message Retransmission List (MRL) [36, 40].

2.1.1.2 Reactive protocols

In reactive approach, routes are created only when a source node requests them (also called *on-demand* approach). Forwarding process is accomplished according to two main techniques i) Source routing protocols and ii) Hop-by-hop routing.

Dynamic Source Routing

DSR [27] is a source routing protocol and requires the sender to know the complete route to the destination. It is based on two main processes: i) the route discovery process which is based on flooding and is used to dynamically discover new routes and maintain them in the nodes cache and; ii) the route maintenance process which periodically detects and notifies networks topology changes. The discovered routes will be cached in the relative nodes.

If the route to destination is not known, a route discovery process is initiated in order to find a valid route. The route discovery is based on flooding the network with route request (RREQ) packets. Every mobile host, that receives a RREQ packet, checks the

contents of its routing cache, and if it is the destination, it replies to the RREQ with a route reply (RREP) packet that is routed back to the original source. In case none of the above holds, the host that receives the RREQ re-broadcasts it to the neighborhood. In this way the RREQ is propagated till the destination. Note that both the RREQ and RREP packets are also source routed. The RREQ maintains the path navigated across the network; thus, the RREP can route itself back to the source by traversing the recorded path backwards. The route carried by the RREP packet is cached at the source for future use. If any link on a source route path is broken, the source host is notified with a special route error (RERR) packet from the immediate nodes. After receiving a RERR message, the node removes all the routes in its routing cache that use the 'broken' link [41].

Ad Hoc On-demand Distance Vector

AODV was introduced in [42] and it is a combination of the DSR and the DSDV protocols. It uses the on-demand mechanism of route discovery and route maintenance from DSR and the hop-by-hop routing and sequence number from DSDV. Per each destination, AODV creates a routing table like DSDV, while DSR uses node cache to maintain routing information [43].

In AODV, when a node needs a route to a destination node, it broadcasts a RREQ message to its neighbours. This message includes the last known sequence number for that destination. The RREQ is flooded in the network until it reaches a node that has a route to the destination. During this controlled flooding of the RREQ, each node, that forwards the RREQ, creates a reverse route for itself to source node and records it. If additional copies of the same RREQ are later received, they will be dropped. When the destination node receives the RREQ, it replies, with a RREP packet, back to the neighbour from which it first has received the RREQ. Intermediate nodes that forward this ROUTE REPLY back to the source, creates a forward route to the destination and updates their route tables. Each node periodically creates and transmits a HELLO message to maintain its route table up-to- date [44].

2.1.2 Hierarchical Routing Protocols

In Hierarchical routing protocols, the network's nodes are divided into clusters. Each cluster has an admin entity (head cluster) which is responsible for routing inside that

cluster; therefore, in this context nodes have different responsibilities and importance in routing algorithm. Also, the cluster head election can be a dynamical and distributed operation. *Zone Routing Protocol* (ZRP) [45] is a hybrid/hierarchical routing protocol that takes advantage of both proactive and reactive schemes by creating overlapped zones based on the separation distances between wireless nodes. The ZRP tries to limit the flooding area per each node by assigning a routing zone to that node [34].

2.1.3 Geographical Position Based

These types of protocols assume that each node in the network is aware of its own location and the status of its one-hop neighbors, e.g., via a Global Positioning System (GPS). In the *Greedy Perimeter Stateless Routing* (GPSR) [46] protocol, in any communication between a source-destination pair of nodes, the source node is aware of the destination node´s location. All one-hop neighbors exchange beacon control messages between each other to simultaneously update their routing tables and limit control message overhead [34].

2.1.4 Hybrid Protocols

Hybrid protocols, by mixing different routing algorithms, try to take advantage of the previously mentioned protocols to reach to the highest efficiency. *Temporally-Ordered Routing Algorithm* (TORA) was introduced in [31] and it is a highly adaptive loop-free distributed routing algorithm based on the concept of link reversal. TORA is proposed to operate in a highly dynamic mobile networking environment. It is source initiated and provides multiple routes for any desired source/destination pair. Nodes in TORA need to maintain routing information about their adjacents, i.e., one-hop neighbors. The protocol performs three basic functions: i) route creation, ii) route maintenance, and iii) route deletion. In TORA, the intermediate nodes maintain a height that is the number of hops between the intermediate and the destination node. To assign the height to each node, A Directed Acyclic Graph (DAG) that is rooted at the destination node is used. Due to dynamic topology of ad hoc networks, the DAG route may be disconnected when a node moves. Therefore route maintenance is necessary to re-establish a DAG rooted at the same destination. Also, the height parameter is determined based on a logical time of a link

failure. In the deletion phase, when a link breakage is detected by a node, it generates a CLEAR packet that deletes the invalid routes from the route cache of the node[47].

A category of current ad hoc routing protocols is illustrated by figure 2.2.

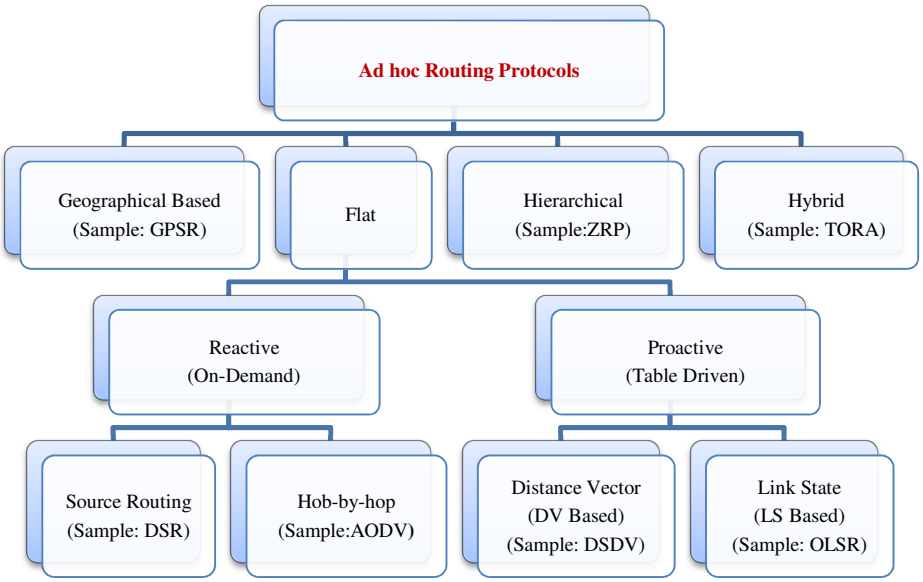

Figure 2.2: A structure of ad hoc routing protocols categories

2.2 Baseline for Legacy Ad hoc Routing Protocols

Routing, in ad hoc networks, is a very challenging issue due to nodes mobility, dynamic topology, frequent link breakage, limitation of nodes (memory, battery, bandwidth, and processing power), and lack of central point, such as base stations or servers. On the other hand, there are lots of performance metrics and quality services which should be satisfied in an ad hoc network, like end-to-end data throughput, average end-to-end delay, jitter, packet loss ratio, and so on. Each protocol can satisfy some of these metrics but with some drawbacks in terms of other metrics. Furthermore, due to the nature of ad hoc networks (distributed and cooperated routing), even for a fixed metric, each protocol can show a different performance for different networks features such as number of mobile nodes, mobility of nodes, and pause time. By comparing different ad hoc

routing protocols, we can extract very important information about the performance of these protocols in different situations.

In related works, the performance of some protocols like DSDV [37] and AODV [48] are analysed individually and some other multi hop protocols are compared [35, 49], but this chapter presents an extensive performance comparison between source routing Dynamic Source Routing (DSR), hob-by-hop routing Ad Hoc On-demand Distance Vector (AODV), distance vector based routing Destination Sequenced Distance Vector (DSDV), and a link state source routing Temporally Ordered Routing Algorithm (TORA). Furthermore, where most of the previous works focused on some well-known metrics (e.g., delay and throughput), the present study not only considers the average end to end delay, Packet Delivery Ratio (PDR), and Normalize Routing Load (NRL) but also introduces new weighted path optimality metric. Also, we present the performance comparisons of these protocols in terms of jitter metric that is an important QoS metric for the video and voice transmission networks. Additionally, most of the related works on performance comparisons of ad hoc routing protocols are limited to only number of the nodes in the network [50], pause time [51], or nodes´ speed; but, this study considers different movement and communication models.

In this section, we shortly review four well-known ad hoc routing protocols: i) DSDV, ii) DSR, iii) AODV, and iv) TORA and carry out a comparison study for different communication and movement models within a simulation environment.

i. Destination Sequenced Distance Vector (DSDV)

DSDV uses distance vector mechanism and it is a proactive table-driven protocol based on the classical Bellman-Ford algorithm. All nodes try to find all paths to the possible destination nodes in a network and to save them in their routing tables.

ii. Dynamic Source Routing (DSR)

DSR [27] is a reactive (On demand) source routing protocol. *Route Discovery* process is based on flooding the network with route request (RREQ) packets. Every mobile host that receives a RREQ packet checks the contents of its route cache, and if it is the

destination it replies to the RREQ with a route reply (RREP) packet that is routed back to the original source and the RREQ is propagated till the destination [27, 39].

iii. Ad Hoc On-demand Distance Vector (AODV)

AODV [42] uses the on-demand mechanism of route discovery and route maintenance from the DSR and also a mechanism for the hop-by-hop routing and sequence number. Per each destination, AODV creates a routing table, while DSR uses node cache to maintain routing information.

iv. Temporary Ordered Routing Protocols (TORA)

Temporally-Ordered Routing Algorithm (TORA) [31] is a highly adaptive loop-free distributed routing algorithm based on the concept of link reversal and it applies a reactive (on demand) source routing scheme. The key design concept of TORA is the reduction of control messages to a very small set of topological changes by performing three basic functions: Route creation, Route maintenance, and Route deletion. All nodes in TORA use a "height" metric to establish a direct acyclic graph (DAG) rooted at the destination [52].

2.3 Simulation Environment

We use Two-Ray Ground Reflection Model which considers both the direct path and a ground reflection path between two mobile nodes. Also, all nodes in the network move based on the Random Way Point Mobility Model. In this movement model, a mobile node starts its travel from a random location inside the topology area after pausing for a certain period of time (called "*pause time*"). After the initial pause time, the mobile node chooses another random location inside the topology area and moves toward this new location by a speed that is uniformly distributed between a predefined minimum and maximum speed. Upon arrival, the mobile node pauses again for the *pause time* and repeats the previous process again till the simulation time is expired [35, 53].

The simulation results presented in this chapter were obtained using the *ns-2* simulator [54]. This is a discrete event, object oriented, simulator from Lawrence Berkeley National laboratory (LBNL) with extensions from the MONARCH Project at Carnegie Mellon University [55]. The Distributed Coordination Function (DCF) of IEEE 802.11 for wireless LANs is used as the MAC layer. The RTS/CTS exchange precedes the data packet

transmission and implements a form of virtual carrier sensing and channel reservation to reduce the impact of the well-known hidden terminal problem.

Movement model I, characterized by pause time	
Topology area	800m x 800m
Maximum mobility of nodes	15 m/s
Pause time	0..100 s
Number of nodes	35
Simulation time	100s
Movement model II, characterized by number of mobile nodes	
Topology area	800m x 800m
Maximum mobility of nodes	20 m/s
Pause time	30s
Number of nodes	10…60
Simulation time	100s
Movement model III, characterized by node mobility	
Topology area	800m x 800m
Maximum mobility of nodes	0m/s…40m/s
Pause time	25s
Number of nodes	40
Simulation time	100s
Parameters of communication model	
Traffic sources	CBR
Data packets size	1024 bytes
Sending rate	8 packets/second
Maximum connection	4

Table 2.1: Simulation Parameters for Comparing the Protocols

Broadcasting data packets and the RTS control are sent using physical carrier sensing. An unslotted CSMA technique with collision avoidance is used to transmit these packets [35]. The radio model uses characteristics similar to commercial radio interface.

The movement scenario files we used for each simulation are categorized into three different groups based on the pause time, number of mobile nodes, and maximum node mobility.

Table 2.1 lists the parameters of the three movement models and the communication model. Each of the movement files, together with a communication scenario file, create the input to the simulation. In each of the two groups, each run is executed for 100 seconds of simulation time and models a network of ad hoc nodes in an 800m x 800m topology area.

2.4 Evaluation Metrics

There are several QoS metrics for comparing ad hoc (e.g., MANETs) routing protocols. In this section, we introduce some of these metrics that we used in this chapter for comparing different routing protocols.

i. Weighted path optimality

The difference between the number of hops a packet took to reach its destination, and the length of the shortest path that physically existed through the network when the packet was originated, is a metric that shows the optimality of paths selected by an ad hoc routing protocol [35]. Path optimality metric is as follows:

$$WPO = \frac{\sum_{i=1}^{N}(AL(i)-SL(i))}{N} \tag{2.1}$$

However, in this definition the importance or usage rate of paths selected by the protocols is not taken into account. Consequently, to reach to a more precise metric, we modify the traditional path optimality metric by considering the percentage of network traffic that is transferred through each path. Weighted Path Optimality (WPO) is as follows:

$$WPO = \frac{\sum_{i=1}^{N}((AL(i)-SL(i))\times SPS(i))}{N\times\sum_{i=1}^{N} SPS(i)} \tag{2.2}$$

Where WPO stands for *weighted path optimality*, AL(i) is the length of the path i that actually is taken by the protocol, $SL(i)$ is the length of the shortest path that existed physically when path i was selected by the protocol, $SPS(i)$ is the size of total packets that

had been transferred through path i, and N is the number of paths in the network (from the beginning of simulation to the end).

ii. Average end-to-end delay

End-to-end delay includes all possible delays caused by the buffering during route discovery latency, transmission delays at MAC layer, queuing at interface queue, and propagation and transfer time. During the entire simulation time, there are M data packets transmitted between the source and the sink nodes. Let us assume that at time S_i a source node sends a packet (P_i) and a sink node receives it successfully at time R_i, *i=[56]*. The end-to-end delay of packet P_i is calculated as follows:

$$d(P_i) = R_i - S_i \tag{2.3}$$

And the average end-to-end delay of successfully transmitted data packets in the network is calculated as follows:

$$d = \frac{\sum_{i=1}^{M} d(P_i)}{M} \tag{2.4}$$

iii. Jitter

In a data transmission between a pair of source and sink nodes, jitter is the variation in the time between packets arriving at the source node. Let us assume that at time S_i the source node sends packet P_i and the sink node receives it at time R_i. The jitter of packet P_i is calculated as follows:

$$Jitter\ P_i = |(R_{i+1} - R_i) - (S_{i+1} - S_i)| = |(R_{i+1} - S_{i+1}) - (R_i - S_i)| \tag{2.5}$$

During the entire simulation time, in a communication between a pair of source and sink nodes, there are M streams of packets, each stream consisting of P packets. We studied the average jitter of all streams of data in the network.

iv. Normalized routing load (NRL)

NRL is the number of routing packets transmitted per CBR data packet delivered successfully at the destination. That is:

$$NRL = \frac{\sum_{i=1}^{w} routing\_packets}{\sum_{i=1}^{M} CBR\_received} \quad (2.6)$$

Where w is the number of total routing packets generated by the routing protocol during the simulation time and M is the number of data packets delivered successfully to the destinations.

v. Packet delivery ratio (PDR)

PDR shows the ability of a routing protocol to successfully deliver the data packets from the source to the destination nodes. That is:

$$PDR = \frac{\sum_{i=1}^{v} CBR\_sent}{\sum_{i=1}^{M} CBR\_received} \quad (2.7)$$

Where v is the number of total CBR data packets sent by the source nodes during the simulation time and M is the number of data packets delivered successfully to the destinations.

2.5 Simulation Results

In this section, simulation results comparing DSDV, AODV, DSR, and TORA, in terms of weighted path optimality, average end-to-end delay, average jitter, NRL, and PDR are presented by changing the pause time, number of nodes, and nodes mobility. All times are in second.

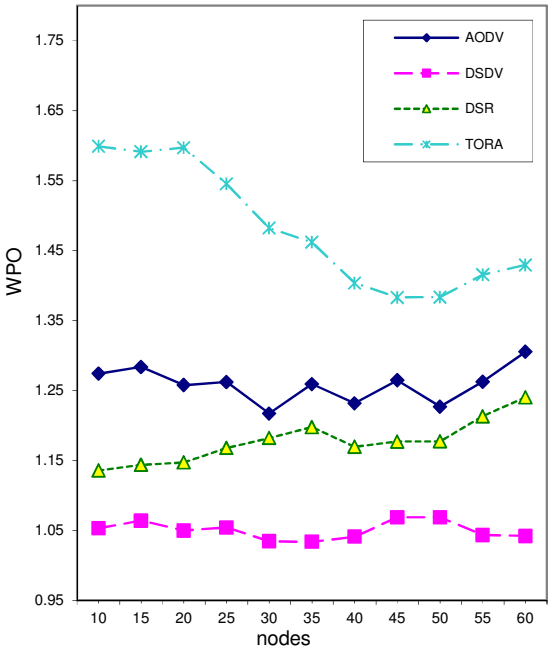

Figure 2.3: Weighted path optimality vs. number of nodes

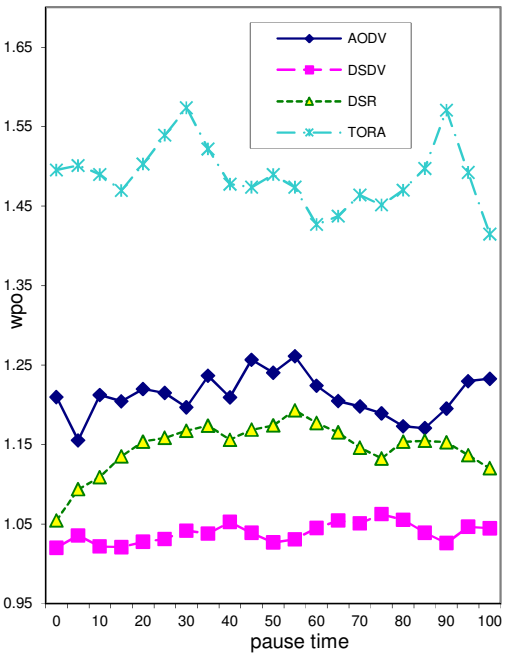

Figure 2.4: Weighted path optimality vs. pause time

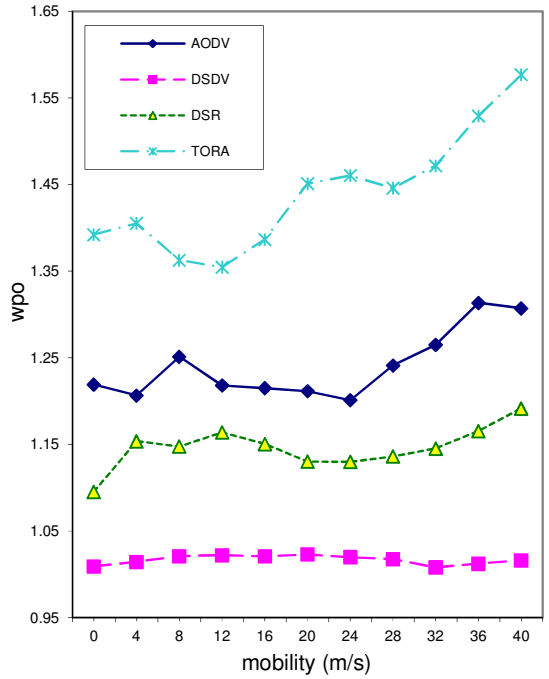

Figure 2.5: Weighted path optimality vs. mobility

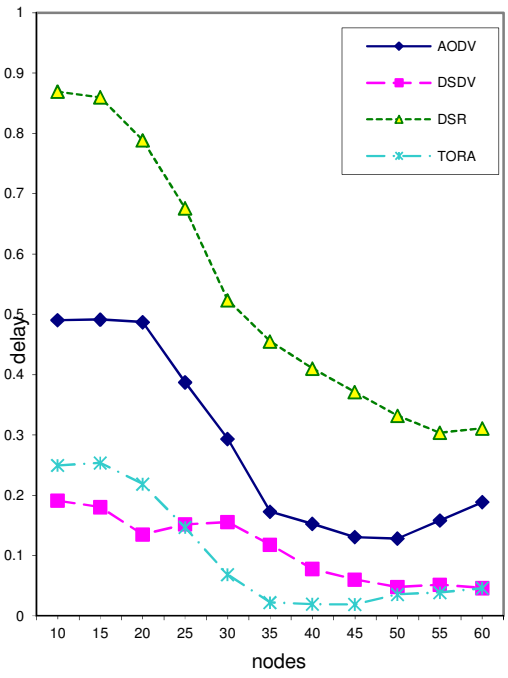

Figure 2.6: Average delay vs. number of nodes

i. Weighted path optimality

Figure 2.3-5 highlight the performance of four protocols with respect to weighted path optimality metric versus number of nodes in the network (i.e., network size), pause time, and the maximum speed of the nodes in the network (also called velocity or mobility of nodes). The closer the value is to zero, the better the weighted path optimality will be achieved. As mentioned before, DSDV employs Bellman-Ford algorithm to find the shortest path between a source node and a destination node. Hence, DSDV performs particularly well, regardless of changing in number of nodes, pause time, and mobility. DSR, AODV, and TORA are the next protocols in terms of performance, respectively.

ii. Average end-to-end delay

Figure 2.6-8 show the performance of DSR, AODV, DSR, and TORA in terms of average end-to-end delay versus number of nodes, pause time, and mobility, respectively. As mentioned before, end-to-end delay of data packets includes all possible delay caused by buffering during route discovery latency, transmission delays at the MAC, queuing at interface queue, and propagation and transfer time. Among the four protocols, DSDV has the shortest average end-to-end delay. DSDV is a proactive protocol and the advantage of these protocols is that a path to a destination is immediately available; therefore, no delay for route discovery is experienced when an application needs to send a packet. In addition, in reactive protocols, AODV is hop by hop initiated while the DSR is a source routing protocol.

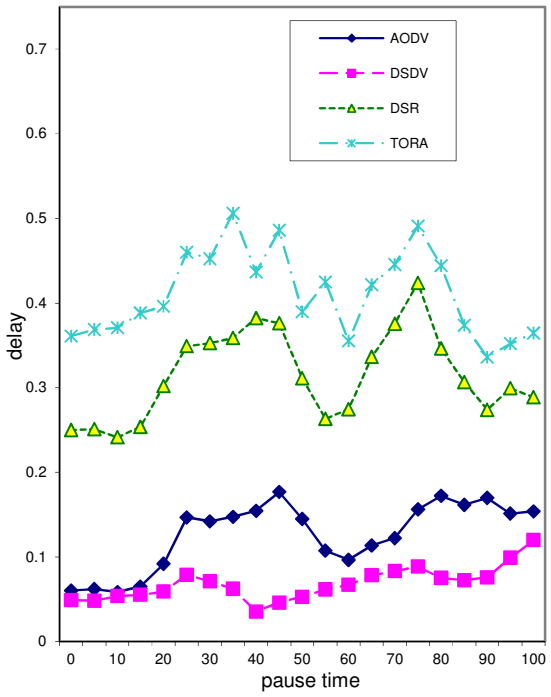

Figure 2.7: Average delay vs. pause time

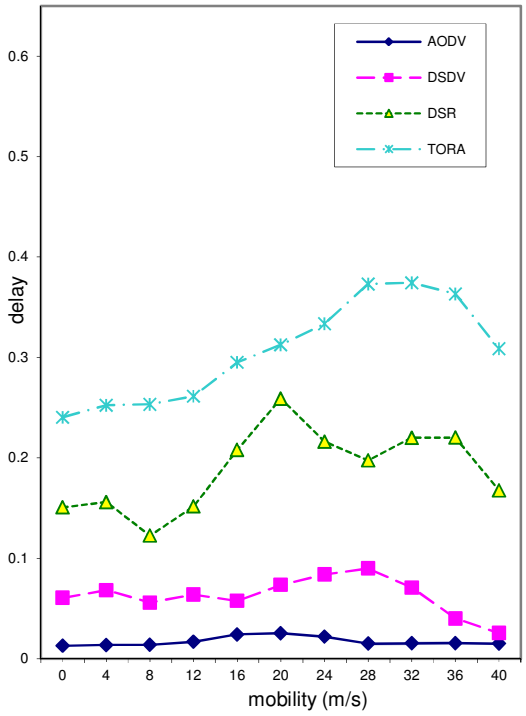

Figure 2.8: Average delay vs. mobility

The source routing protocols have longer delay because their route discovery takes more time as every intermediate node tries to extract information before forwarding the reply. And the same issue happens when a data packet is forwarded hop by hop through the path by the source routing method. Hence, while source routing makes route discovery more profitable, it slows down the transmission of packets. This is the reason why AODV performs better than the two other reactive protocols, in general. Simulation results demonstrate a good performance of DSDV and AODV, but a poor performance of DSR. TORA has the worst performance among the protocols. On the other hand, increasing the node number of networks does not greatly affect TORA with respect to average end-to-end delay.

iii. Jitter

Figure 2.9-11 show the performance of DSR, AODV, DSDV, and TORA, in terms of average jitter versus number of nodes, pause time, and mobility respectively. Here each protocol's curves fluctuate around some similar values, regardless of change in the number of nodes (figure 2.9), pause time (figure 2.10), and mobility (figure 2.11). All curves have

an ascending shape, but in a slight manner. The average jitter of DSDV is relatively low and within the expected range. DSR has the worst average jitter among all these routing protocols.

iv. Normalized routing load

Figure 2.12-14 show the performance of DSR, AODV, DSDV, and TORA, in terms of normalized routing load versus number of nodes, pause time, and mobility, respectively. A low pause time or a high mobility leads to a higher link breakage, less reliable routes, and very dynamic topology. Routing protocols may adapt themselves to a dynamic topology by maintaining the freshest routing information. DSR generates less control packets than the other protocols which make it a lightweight routing protocol that can be very suitable for limited devices, in terms of battery, memory, and processing power.

v. Packet delivery Ratio

On-demand routing protocols, such as AODV and DSR, have a high throughput in terms of successfully delivering the data packets to their destinations which lead to a high PDR, regardless of number of nodes, pause time, and mobility of nodes, as figure 2.15-17 show, respectively. On the other hand, DSDV is a reactive table driven protocol and is not very adaptive to route changes and link breakages due to node mobility, and low pause time leads to a lower PDR.

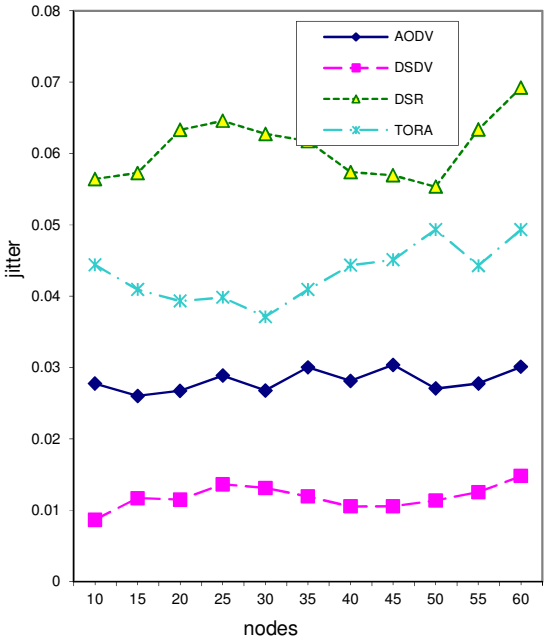

Figure 2.9: Average jitter vs. number of nodes

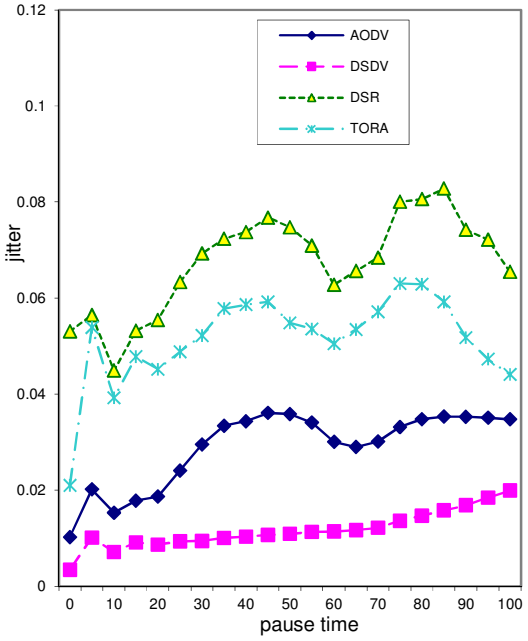

Figure 2.10: Average jitter vs. pause time

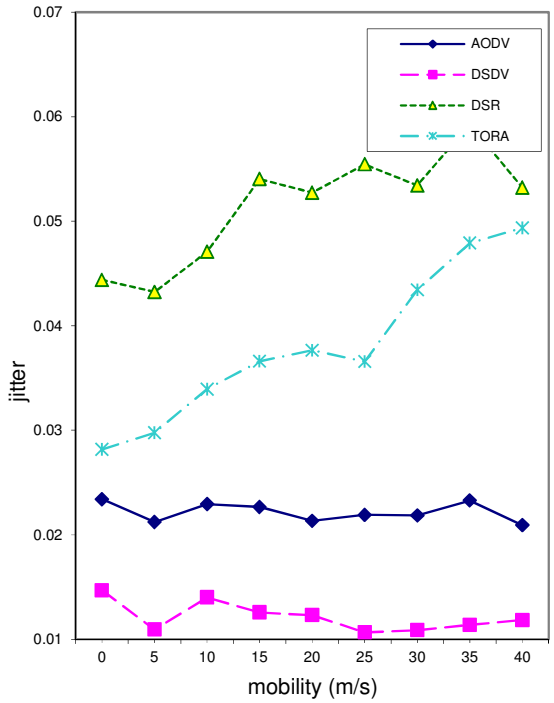

Figure 2.11: Average jitter vs. mobility

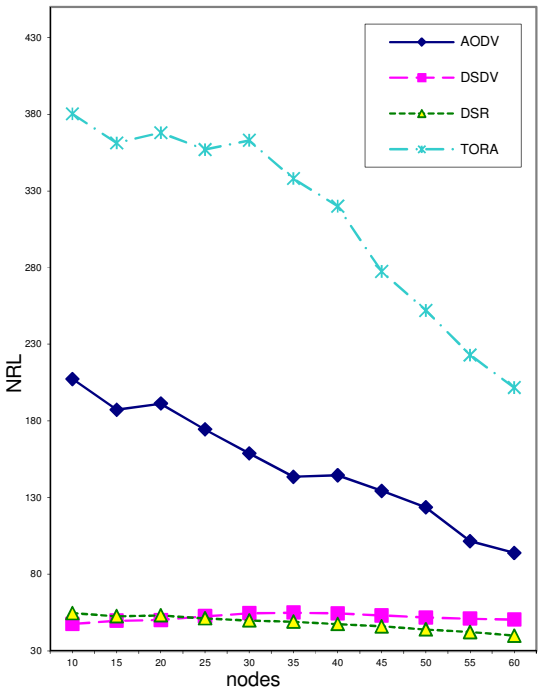

Figure 2.12: NRL vs. number of nodes

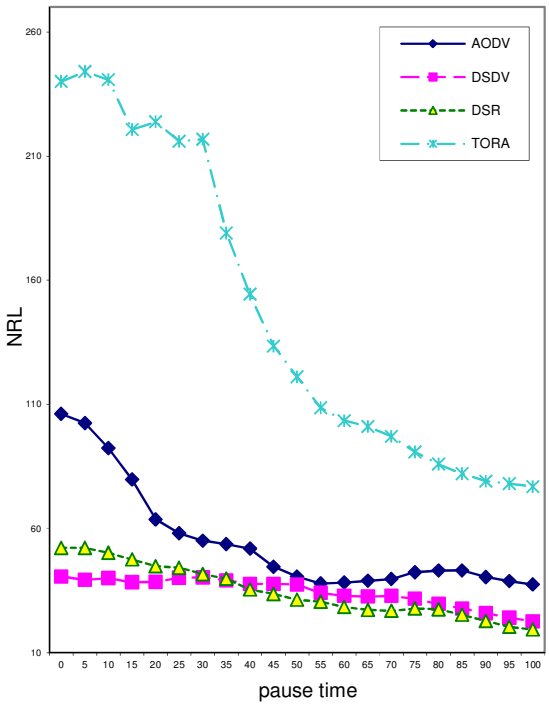

Figure 2.13: NRL vs. pause time

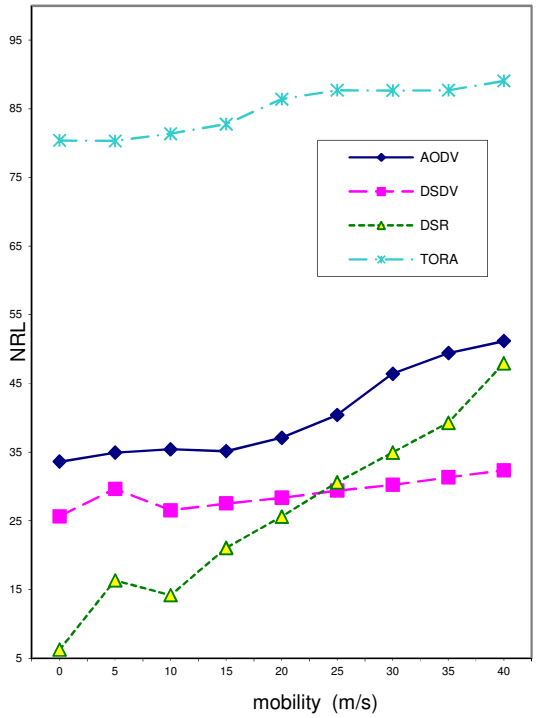

Figure 2.14: NRL vs. mobility

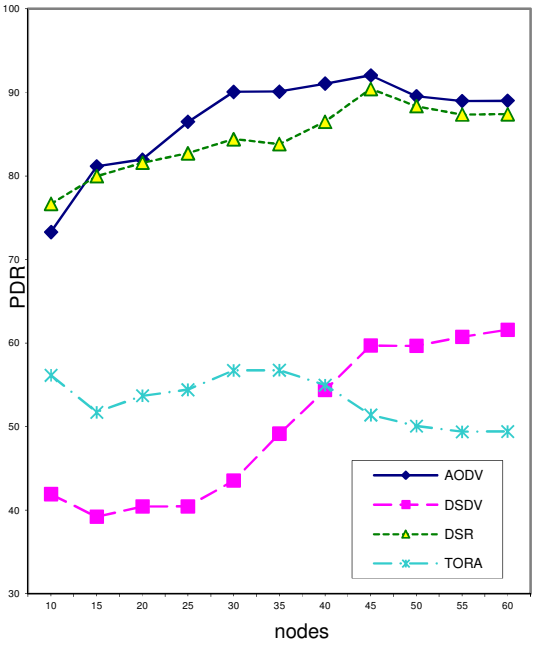

Figure 2.15: PDR vs. number of nodes

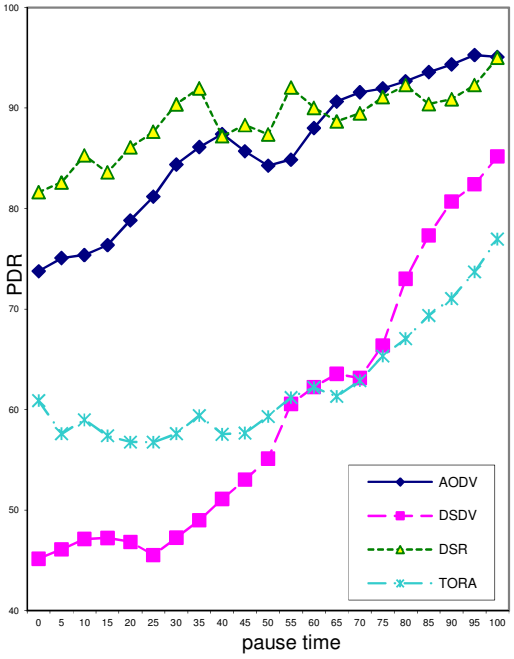

Figure 2.16: PDR vs. pause time

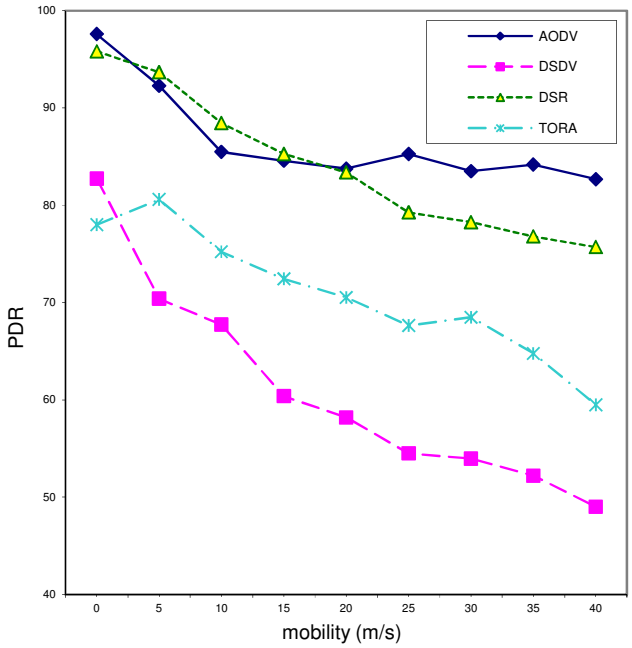

Figure 2.17: PDR vs. mobility

2.6 Conclusion

In this chapter, we described the basic principles of ad hoc networks. We presented a categorization of the current routing protocols for ad hoc networks, presented the performance study of several ad hoc routing protocols. We have compared four protocols, DSDV, AODV, DSR, and TORA, with respect to several metrics, such as path optimality, average end-to-end delay, jitter, Normalized Routing Load (NRL), and Packet Delivery Ratio (PDR). In comparison to the related previous works on this subject, here we used a wide range of different movement and communication scenarios which are characterized by the pause time, mobility, and the number of nodes to find the benefits and drawback of these protocols for different scenarios. Most of the studied protocols perform well in some cases, yet with some certain drawbacks in the others. DSDV performs very well in weighted path optimality while TORA has the worst performance. We observed that DSDV and AODV have the best performance in terms of average end to end delay. DSDV has the best average jitter and then AODV, TORA, and DSR in respective orders. Also, proactive on-demand routing protocols such as AODV and DSR show higher performance in terms of PDR and NRL.

CHAPTER 3

3 Energy Efficient Ad hoc Routing

In this chapter, we propose two new ad hoc routing protocols: Energy Efficient Ad hoc On-demand Distance Vector version 2 (E2AODVv2) and Load balanced Dynamic Source Routing (LBDSR) for achieving higher energy efficiency and load balancing capability. The new proposed routing protocols aim to achieve higher energy efficiency, enabling mechanisms to handle and save critical nodes, and to handle traffic balancing in the network by considering the energy, traffic, and length of the routes.

3.1 Energy Efficient Ad hoc On-demand Distance Vector version 2 (E2AODVv2)

As the first contribution of this thesis, we propose a novel energy efficient and load balanced ad hoc routing protocol, employing appropriate mechanisms to detect the critical nodes and adjust their load to avoid early node failure.

3.1.1 Introduction

In a dynamic topology, such as MANETs, nodes can freely move; this often leads to link breakage and effect the interlaying of the previously founded routes. It is a complex problem that any dynamic wireless routing protocol has to solve.

The dynamic topology in ad hoc networks implies that some nodes may relay more traffic than others, mainly because of their location in the network; these hot spots will consume their battery energy faster than the others. Unbalanced battery consumption in the network nodes can lead to several problems such as early node failure, network partitioning, and reduced route reliability. Traffic concentration on these nodes may increase delay and packet loss. Also, since the majority of the network traffic may potentially pass through these nodes, they can become an important target for attackers.

Power-aware routing algorithms aim to deliver new routing paths that take into account energy as a metric. For example, Gomez et al. [57] propose adding new intermediate nodes to a route from a source to a destination to reduce the overall required transmission power of the intermediate nodes. Lindgren and Schelen [58] improve the AODV routing protocol in terms of energy efficiency, by selecting paths through Power Base Stations (PBSs) instead of through normal nodes. Edwards et al. [59, 60] and Djenouri and Badache [59, 60] address two power-aware extensions of the original DSR ad hoc routing protocol that use energy aware metrics, such as the remaining battery power to decide which nodes should participate more often in packet forwarding. Load balancing and homogeneous distribution of battery energy consumptions among the nodes is another solution to achieve a power-aware ad hoc routing algorithm that is addressed in [61].

However, most of the existing power-aware approaches lead to some common drawbacks, such as requiring global topology information, increased delay, and increase in the number of required control messages that should be created by the routing protocol to deliver the user´s data packets (metric called Normalized Routing Load), and limited scalability [57-64].

Ad Hoc On-demand Distance Vector version 2 (AODVv2) protocol [65] is a well-known routing protocol presented by Mobile Ad hoc NETwork (MANET) group of Internet Engineering Task Force (IETF) [1]; it applies a uniformed message format and has inherent features to support many required QoS metrics; but it is limited in terms of energy and traffic management.

We exploit AODVv2 as a fundamental building block and go beyond this by designing and implementing energy and traffic aware capability by proposing an Energy Efficient Ad Hoc On-demand Distance Vector version 2 (E2AODVv2) that can provide a weighted and flexible trade-off between energy consumption and load dispersion for reactive flat routing protocols.

We implemented several simulation scenarios to test the capability of E2AODVv2, where simulation results show that the proposed protocol achieves high energy efficiency, decreases the percentage of failed nodes due to lack of battery power, and extends the lifespan of the network. The proposed approach can be applied to any flat ad hoc routing

protocols. The position of the new proposed routing protocol, E2AODVv2 is shown in figure 3.1.

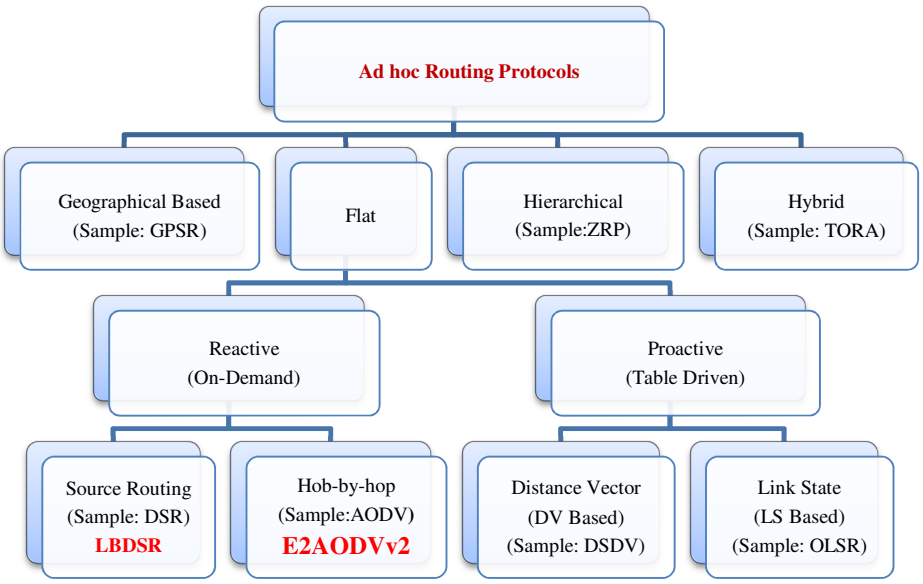

Figure 3.1: The position of E2AODVv2 and LBDSR in ad hoc routing protocols categories

3.1.2 Revised Ad Hoc On-demand Distance Vector (AODVv2)

AODVv2 is a successor to the AODV that is being developing by IETF MANET; prior to the 26th revision ([65]), AODVv2 was called DYnamic MANET On-demand (DYMO).

3.1.2.1 Protocol description of AODVv2

AODVv2 tries to simplify the current reactive protocols, such as DSR and AODV, and simultaneously conserves their two main well known routing operations: route discovery and route maintenance [66]. It has multipath capability (optional) and is also a hop-by-hop routing protocol. Therefore, intermediate nodes, located in a route between a source and a destination node, are able to extract additional information from the control packets. AODVv2 applies a standard generalized MANET Packet/Message format [67] for control packets: A uniformed message format, called Routing Message (RM), is used for all

control messages and routing packets. Another interesting feature of AODVv2 is the capability to support Internet Protocol (IP) [68] version 4 and 6 (IPv4 and IPv6 respectively) which can be considered an essential feature for next-generation networks.

Several works analysed the performance and benefits of AODVv2 [66, 69, 70]. [71] focused on reducing the delay and on increasing the packet delivery ratio, by finding routes with high probability to be used in future. Also, there are few works such as [69], that try to extend a multipath version of AODVv2 to switch between different paths from source to destination in order to balance the battery power consumption and the traffic of the nodes on the different routes. However, these works do not provide a mechanism for handling both load balancing and energy efficiency in highly dynamic topologies.

3.1.2.2 Routing operations of AODVv2

All routing operation in AODVv2 can be categorized into three phases:

i. Route Request Phase

In a communication between a source node (S) and a destination node (D), S originates a Route Message (RM), called ROUTE REQUEST (RREQ) message, and broadcasts this to all of its neighbours. These RREQs are then flooded in the network until they reach D. An intermediate node which does not know the route to D should forward the RREQ to its neighbours. The intermediate node will also drop the repeated request for the same destination and will not forward them anymore, as shown by figure 3.2. Routing Messages (RM), i.e., RREQ, RREP, or RERR, has several fields such as current node address, next node address, HopLimit (the default value is 10 based on [65]), Target (destination node address if it is an RREQ message), and the Origin (source node address if it is an RREQ message) [65]. By tracing the RREQ traversed path (called *accumulated path*), each node, such as the intermediate or destination node who receives an RREQ message, can extract routing information. Figure 3.2 shows the RREQ phase in AODVv2 routing protocol for a sample network.

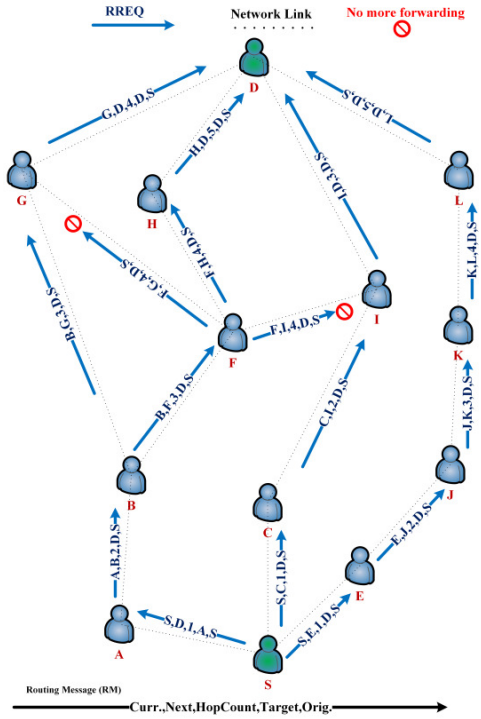

Figure 3.2: A sample of RREQ phase in AODVv2

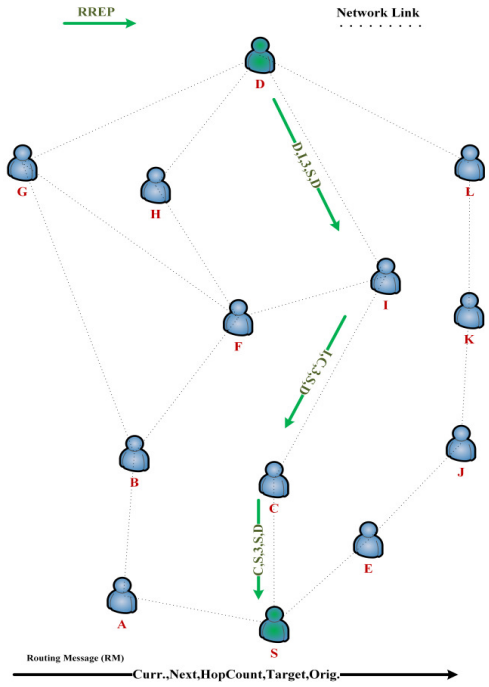

Figure 3.3: A sample of RREP phase in AODVv2

ii. Route Reply Phase

As shown by figure 3.3, when RREQs finally reach the destination node, another RM which is called ROUTE REPLY (RREP), will be originated by the destination node. The intermediate nodes who have, in their routing table, an entire route towards that destination node also can immediately reply to the RREQ originated by the source node (known as *gratitude reply* and it is an optional feature of AODVv2). Both the RREQ and the RREP have the same uniform structure. When a route is added to the route cache of a node, a $T_{timeout}$ field will be set for that route. This means that after $T_{timeout}$ this route will be deleted from the route cache.

iii. Route Error Phase

An intermediate node may originate an RM, called an ROUTE ERROR (RERR), in the two main scenarios. In the first case, the intermediate node does not have a valid route for the destination of a received data packet and consequently the packet is undeliverable. In the second case, the intermediate node detects a link breakage, as shown in figure 3.4.

3.1.3 Routing Mechanism of Energy Efficient AODVv2 (E2AODVv2)

The inherent benefits of AODVv2 due to standardized control packets following IETF uniform packet formats and the capability to support IPv6 suggest that this protocol will have a strong legacy in ad hoc networks. Therefore, if we are to pursue energy efficient operation for routing in ad hoc networks, then exploring AODVv2 as the fundamental building block can be potentially a springboard for promoting significant energy savings in the network. In this section, we describe our proposed approach to increase the energy efficiency of AODVv2.

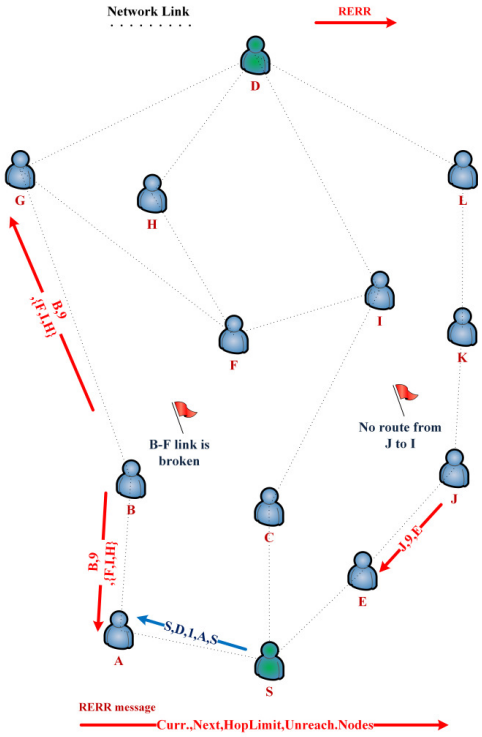

Figure 3.4: A sample of RERR phase in AODVv2

3.1.3.1 Routing messages in E2AODVv2

E2AODVv2 introduces two new fields for each routing message which are discussed in this section.

i. Energy Field

Suppose that for a communication between a source and a destination node there are N routes and the number of nodes in *i*th route, called $Route_i$, is M_i (as shown in figure 3.5). In E2AODVV2, each *kth* node in $Route_i$, has a battery power (BP_k) level quantified as 16 different values (from 0 to 15).

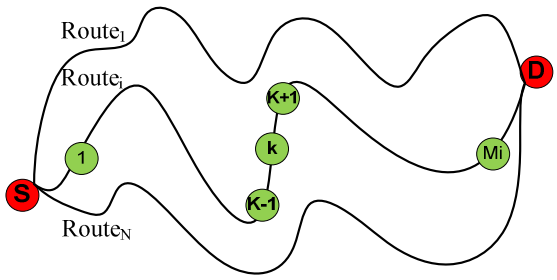

Figure 3.5: Multiple routes between source and destination

Moreover, there is a Critical Battery Power Level (CBPL) whose default value is 3 (CBPL=3). Therefore node node k in $Route_{i,}$, which has a BP of less than CBPL, is suffering from energy depletion and is considered as a critical node in terms of battery power (i.e., $BP_k < CBPL$). Critical nodes should not be selected for packet forwarding in E2AODVv2.

The *Energy* field of RREQ in E2AODVv2 has three cells: *TotBat*, *MinBat*, and *CritBat*. $TotBat_i$ is the summation of the total battery power level of all nodes of $Route_{i,}$ as foloows:

$$TotBat_i = \sum_{k=1}^{k=M_i} BP_k \tag{3.1}$$

The $MinBat_i$ cell in the energy field of $RREQ_i$ shows the minimum value of BP for all nodes in $Route_i$, whilst $CritBat_i$ shows the number of nodes which have a BP less than CBPL in $Route_i$.

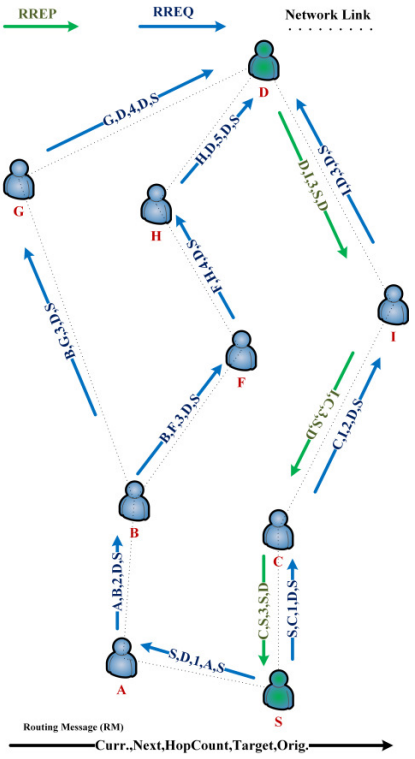

Figure 3.6: D originates route replies (RREP) toward S in AODVv2

Figure 3.6 shows a network based on the previous sample that has a simpler topology. In this example, there are 8 nodes. The battery power level and the traffic parameter of all these nodes are given in Table 3.1 (the traffic parameter will be introduced later in this section). Based on Table 3.1, the energy and the traffic parameters of all paths of the network in figure 3.6 are calculated and given by Table 3.2.

For example, as the first column of Table 3.2 shows, the first path (i=1) from S to D (i.e., S-**C**-**I**-D) has one node with a critical battery level (*CritBat*=1); the minimum battery power level of all nodes in this path is 2 which belongs to node C, as shown in Table 3.1 (*MinBat*=2*)*; the total battery power level of all nodes in this path is 12 (*TotBat*=12), and the number of intermediate nodes in this path is 2 (*M*=2).

Node	Battery Power (BP)	Traffic Parameter (TP)
A	6	0.7
B	8	0.5
C	2	0.8
F	12	0.3
I	10	0.6
G	5	0.9
H	14	0.4

Table 3.1: Battery power level and traffic parameter of all nodes of the network in Figure 3.6

For this network, the original AODVv2 chooses the shortest path (i.e., the first path S-C-I-D); however, the new proposed protocol, E2AODVv2, will choose the second path (i.e., S-A-B-F-H-D), which is a stronger alternative in terms of energy efficiency and even load balancing which is discussed later on.

ii. Traffic Field

In figure 3.5, if the queue size of the interface of node *k* located in $Route_i$ is AQ_k and the maximum queue size is MQ_k, then the traffic parameter of node *k* in E2AODVv2 (i.e., TP_k) will be calculated as follows:

$$TP_k = \frac{AQ_k}{MQ_{k_i}} \tag{3.2}$$

The *Traffic* field of RREQ has two cells: *TotTra* and *MaxTra*. The first one is the summation of all traffic parameter in $Route_{i,,}$ as defined by:

$$TotTra_i = \sum_{k=1}^{k=M_i} TP_k \tag{3.3}$$

The second one, i.e., $MaxTra_i$, is the maximum TP values of $Route_i$. The traffic parameters of all nodes of the network, which is presented in figure 3.6, is given by Table 3.1 whilst the traffic parameters of all paths is given in Table 3.2.

For example, as the first column of Table 3.2 shows, for the first path (i=1) from S to D (i.e., S-**C-I**-D), the total traffic parameter of all intermediate nodes is 1.4 (i.e., *TotTra* = 1.4) and the maximum traffic parameter is 0.4 (i.e., Max*Tra* = 0.4) which belongs to node C as shown in Table 3.1. Also, M is the number of intermediate (relay) nodes in a path, e.g., the first path S-**C-I**-D, has two intermediate nodes; therefore M=2.

Path	S-**C-I**-D	S-**A-B-F-H**-D	S-**A-B-G**-D
Path Number	i=1	i=2	i=3
CritBat	1	0	0
TotBat	12	40	19
MinBat	2	6	5
M	2	4	3
TotTra	1.4	1.9	1.8
MaxTra	0.8	0.7	0.7

Table 3.2: Energy and traffic features of all networks nodes in figure 3.6

Dest. Add.	Route Seq. Num.	Hop Count	Next Hop	Life Time (ms)	Last Used (ms)	Forward Flag	Broken Flag	…
D	112	3	I	30000	10000	T	F	…
E	36	2	S	20000	15000	T	F	…
K	76	4	I	50000	1000	T	F	…
…	…	…	…	…		…	…	…

Table 3.3: A sample routing table for E2AODV2

Therefore, in E2AODVv2, when the intermediate node *k* receives RREQ, it updates $TotTra_i$ and MaxTrai of the traffic field of the RREQ and forwards the RREQ toward the next node. A sample node routing table is shown in Table 3.3.

3.1.3.2 Route selection process

In E2AODVv2, when a destination node receives several RREQs from different routes, as in figure 3.5, it runs a route selection process to determine the best route in terms of *energy* and *traffic* parameters.

i. Calculating Energy Parameter of a route in E2AODVv2

The energy parameter of $Route_{i,}$ called *E(i),* indicates the priority of the $Route_{i,}$ in terms of the battery power level as presented by:

$$E_i = \frac{TotBat(i)}{M_i \times InitialBat} \tag{3.4}$$

Where M_i is the number of nodes in $Route_i$, *TotBat(i)* is the total battery power level of all nodes in $Route_{i,}$ as presented in equation 3.1, and *InitialBat* is the initial battery power level of a node (which is preset to the same value for all nodes). In some scenarios, a route may have a few nodes with a very low energy level; this route should be avoided due to these bottleneck nodes. Therefore, the negative impact of *MinBat(i)* (which is the minimum battery power level of $Route_i$) should be applied to *E(i),* as given by:

$$E_i = \frac{TotBat(i) \times MinBat(i)}{M_i \times InitialBat^2} \tag{3.5}$$

As mentioned before, *CritBat(i)* identifies the number of nodes in $Route_i$ which have reached a critical battery level and hence should be avoided from packet relaying functions (otherwise, the route is likely to break down). A large value for *CritBat(i)* triggers an alert that this route has several nodes with a critical battery level. Consequently, *E(i)* can be revised to reflect also the negative impact of these nodes in $Route_i$. The final *E(i)* of a route in E2AODVv2 will be calculated based on equation 3.6. When a destination node receives an RREQ, it can calculate E_i via the energy field of the RREQ.

$$E_i = \frac{TotBat(i) \times MinBat(i)}{M_i \times InitialBat^2 \times (CritBat(i)+1)} \tag{3.6}$$

ii. Calculating route Traffic parameter in E2AODVv2

A route with a lower *traffic* metric cost has a higher priority in the routing process of E2AODVv2. Nevertheless, a route may have a low overall *traffic* metric even if one of its

nodes is overloaded with traffic thus creating a bottleneck, and should be potentially avoided. The traffic parameter of $Route_i$ is shown by $T(i)$ and is calculated by:

$$T_i = \frac{TotTra(i) \times MaxTra(i)}{M_i} \quad (3.7)$$

Where M_i is the number of nodes in $Route_i$. When a destination node receives an RREQ, it can calculate T_i via the traffic field of the RREQ.

iii. Precedence Function

The function that determines the precedence of $Route_i$ is given by $RoutePrio(i)$. This function depends on the *energy* and the *traffic* parameters of $Route_i$ and is given by:

$$RoutePrio(i) = \frac{E_i}{T_i} \quad (3.8)$$

However, we can choose a trade-off between energy efficiency and traffic balancing, but this function is flexible and can be customized by giving a higher weight to the energy or traffic parameter.

3.1.3.3 Routing behaviour of nodes in E2AODVv2

In E2AODVv2, each node shows different routing behaviour that depends on the role of the node in the communication.

i. Source nodes routing behaviour

In E2AODVv2, when a source node S wants to start a communication with a destination node D, it searches its route cache. If there is a route in the cache toward D the source node starts the communication (note that the old route would have expire and deleted from the cache automatically). Otherwise, S generates a RREQ message, initializes the values of the energy and the traffic fields (they will be set to zero) and floods them into the network.

ii. Intermediate nodes routing behaviour

The intermediate nodes who receive the RREQ message, update the energy and traffic fields accordingly:

1. To update the energy field, the intermediate node (that is node k in $Route_i$) adds its own BP to the *TotBat* value. As a next step, the intermediate node, checks the *MinBat* field of the received RREQ message and if it has a BP_k less than *MinBat,* then *MinBat* will be updated by BP_k (i.e., if $BP_k < MinBat_i$ then $MinBat_i = BP_k$). Also if $BP_k < CBPL$, then *CritBat* will be increased by one.

2. To update the traffic field, the intermediate node, adds its TP_k number to the *TotTra* number. Also, if $TP_k > MaxTra$, then *MaxTra* will be updated by TP_k.

However, the intermediate node in traditional AODVv2 will drop the repeated request for the same destination and will not forward them anymore, as figure 3.2 shows; but, in our protocol the intermediate nodes will forward $K_{Inter.forward}$ of these RREQs as well, as shown in figure 3.7. Also intermediate node replies to the new coming RREQs only in $K_{Inter.timout}$ seconds. The received RREQ messages are saved in a RREQ table. E2AODVv2 uses the current tables and structures of AODVv2. $K_{Inter.forward}$ is set to 3 and $K_{Inter.timout}$ is set to 2 seconds in E2AODVv2.

iii. Destination nodes routing behaviour

When a destination node receives *n* RREQs from different *n* routes, by extracting the required information from RREQ, it can calculate the *RoutePriority* for all these RREQs and finally the route which has the best *RoutePriority* will be chosen.

Figure 3.8 shows the role of the source, the intermediate and the destination nodes in the communication. Each route request, such as $RREQ_{i,,}$ should be initialized by the source node. It will pass from one node to another within the route, and each node has the opportunity to update the energy and traffic fields accordingly (where each route will have M_i intermediate nodes). It finally reaches the destination node which is responsible for running the Precedence Function, finding the best route and initializing the RREP.

Therefore, via this distributed mechanism, the nodes in E2AODVv2 can cooperatively balance the load in the network, in terms of traffic and battery power consumption.

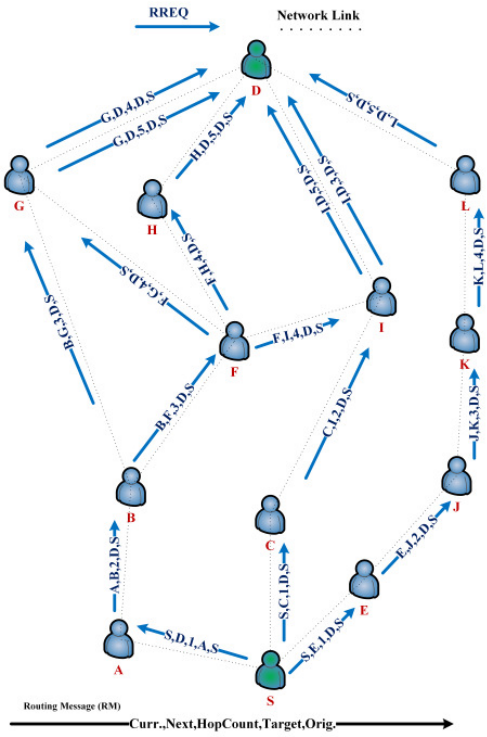

Figure 3.7: A sample of RREQ phase in E2AODVv2

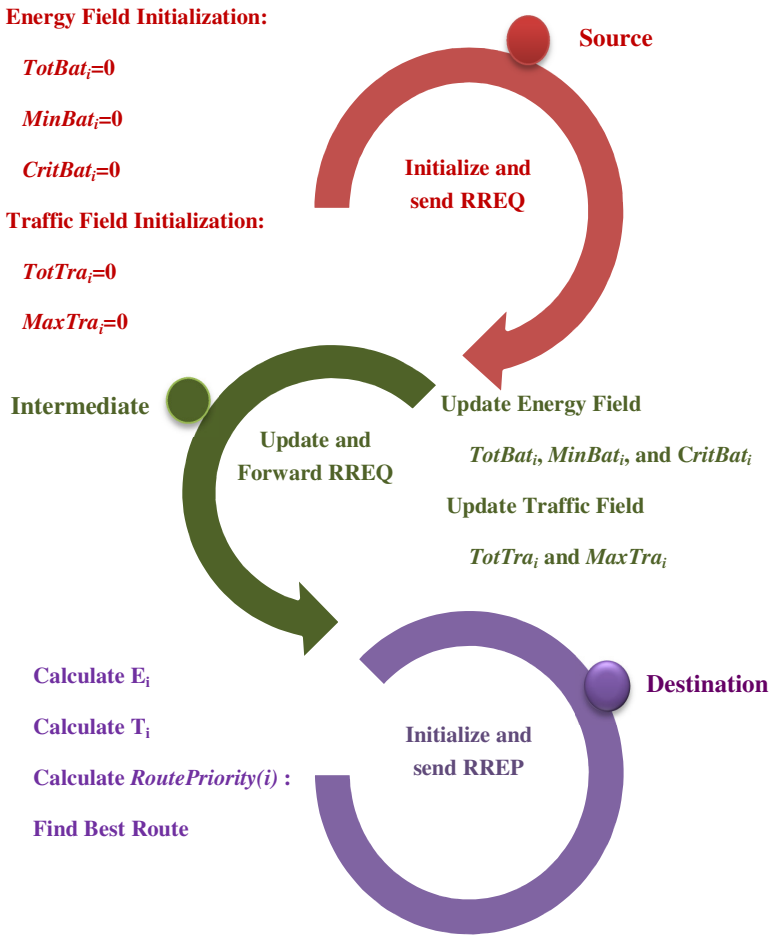

Figure 3.8: the routing behaviour of source, intermediate, and destination nodes in E2AODVv2.

3.1.4 Evaluating E2AODVv2

3.1.4.1 Analytical analysis of E2AODVv2

The traditional AODVv2 for the network in figure 3.6 chooses the first path (i=1) from S to D (i.e., S-C-I-D). The battery power level and the traffic parameter of all these nodes are given in Table 3.1. Also based on Table 3.1, the energy parameter, the traffic parameters, and the priority of all paths of the network in figure 3.6 are calculated and given in Table 3.4.

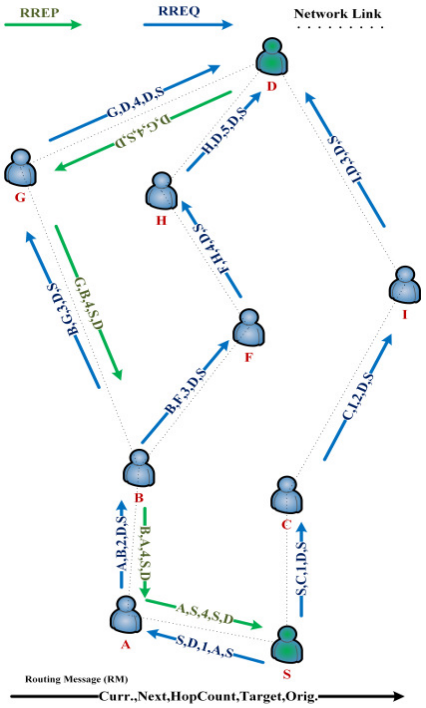

Figure 3.9: A sample of RREP phase in E2AODVv2

Path	S-**C**-**I**-D	S-**A**-**B**-**F**-**H**-D	S-**A**-**B**-**G**-D
Path Number	i=1	i=2	i=3
CritBat	1	0	0
TotBat	12	40	19
MinBat	2	6	5
M	2	4	3
E	0.026	0.26	0.14
TotTra	1.4	1.9	1.8
MaxTra	0.8	0.7	0.7
T	0.56	0.3325	0.63
RoutePrio (E/T)	0.046	0.781	0.22
Best Path	×	√	×

Table 3.4: The energy and traffic features of all paths in the network of figure 3.9

Our proposed E2AODVv2 protocol, based on Table 3.4, chooses the second path path (i.e., S-A-B-F-H-D), which provides equal priority towards reducing energy and balancing traffic load. Therefore, E2AODVv2 send RREP messages through this optimized path as shown by figure 3.9.

3.1.4.2 Performance metrics

We use Two-Ray Ground Reflection Model which considers both the direct path and a ground reflection path between two mobile nodes. Also all nodes in the network move based on the Random Way Point Mobility Model. In this movement model, a mobile node starts its travel from a random location inside the topology area after pausing for a certain period of time (called "*pause time*"). After the initial pause time, the mobile node chooses another random location inside the topology area and moves toward this new location by a speed that is uniformly distributed between a predefined minimum and maximum speed. Upon arrival, the mobile node pauses again for the *pause time* and repeats the previous process again till the simulation time has expired [35, 53].

The simulation results presented in this chapter were obtained using the *ns-2* simulator [54]. Traffic sources are CBR (Constant Bit Rate) and the packet sending rate at the source nodes is 8 packets per second. Table 3.5 presents a summary of the parameters for the simulated scenarios (movement and traffic files). We evaluated the proposed protocol using five metrics: i) balancing energy consumption, ii) scalability, iii) network lifetime, v) node's failure level, and iv) jitter.

i. Balancing Energy Consumption

This metric will allow measuring the effectiveness of the energy balancing algorithm used by E2AODVv2. Let us denote the energy load of all the nodes in the network consisting of N nodes, is a set $EL = \{el(i) \in EL: i = 1, \dots, N\}$.

Where element of this set, e.g., $el(i)$, is calculated as ratio of the consumed energy node *i* to the total aggregate energy consumed in all the nodes, including node i. That is,

$$el(i) = \frac{ConsumedEnergy(i)}{TotalConsumedEnergy} \quad (3.9)$$

Having calculated all N elements of the el, we can now calculate the standard deviation of el for a network consisting of N nodes (i.e., σ_{EL}^{N}). The value of σ_{EL}^{N} will be our energy balancing metric for comparing different protocols; the smaller the σ_{EL}^{N}, the more effective the energy balancing capability.

*Parameters of **movement model I**, characterized by **number of nodes***	
Topology area	500m × 500m
Maximum mobility of nodes	5 m/s
Number of nodes	**1…20**
Simulation time	300s
*Parameters of **movement model II**, characterized by **simulation time***	
Topology area	500m × 500m
Maximum mobility of nodes	5 m/s
Number of nodes	20
Simulation time	**200…1800s**
*Parameters of **traffic** model*	
Traffic sources	CBR
Data packets size	512 bytes
Sending rate	8 packets/second

Table 3.5: Parameters of movement models and communication model

ii. Scalability

Another interesting metric is the scalability of the proposed protocol in terms of load balancing; i.e., if the proposed protocol can balance the load of the network when we increase the number of network nodes (i.e., N in our settings). For this purpose we change the number of nodes N (the network size) to a maximum number (which is 20 in our setting, based on Table 3.5) and calculate the related σ_{EL}^{N} for each network size.

iii. Network lifetime

Network lifetime is another important metric that reflect the load balancing capability of the routing protocol; the bigger the lifetime, the more effective the energy balancing capability. To compare the lifetime of the proposed protocol with other protocols, we take two parameters into consideration: i) failure time FT_{first} (the time that the first node in the network ran out of battery power) and ii) last failure time FT_{last}.

iv. Node failure level

Finally the last performance metric that we use is the node failure level, which determines the capability of E2AODVv2 in keeping nodes alive for longer durations. The node failure level, for a time window *T*, is the percentage of nodes in the topology that have failed due to a depleted battery. This value is calculated as follows:

$$Node\ Failure\ Level = \frac{\#failed\_nodes\_in\_T}{\#nodes\_in\_topology} \tag{3.10}$$

v. Jitter

In a data transmission between a pair of source and sink nodes, jitter is the variation in the time between packets arriving at the source node. Let us assume that at time S_i the source node sends packet P_i and the sink node receives it at time R_i. The jitter of packet P_i is calculated as follows:

$$Jitter\ P_i = |(R_{i+1} - R_i) - (S_{i+1} - S_i)| = |(R_{i+1} - S_{i+1}) - (R_i - S_i)| \tag{3.11}$$

During the entire simulation time, in a communication between a pair of source and sink nodes, there are M streams of packets, each stream consisting of P packets. We studied the average jitter of all streams of data in the network.

3.1.4.3 Simulation results

This section presents the simulation results for the comparison between two reactive protocols AODV and DSR that are considered baselines in this work, against our proposed E2AODVv2 protocol.

i. Simulation results regarding balancing energy consumption and scalability

For this metric, simulation results are presented in terms of the capability of our approach to balance battery power (energy) consumption. The traffic model is the one presented in Table 3.5 with the following variants:

i. *Distributed traffic amongst nodes*: traffic sources and destinations will be chosen randomly in time. This Variant is shown by "*Distributed"* in Table 3.6.

ii. *Concentrated traffic with overloaded nodes*: traffic destinations will be predefined (all traffic in the network goes toward those sink nodes). This variant is shown by "*Concentrated"* in Table 3.6.

	σ_{EL}^{12} *Distributed*	σ_{EL}^{12} *Concentrated*	FT_{first} *Distributed* (second)	FT_{first} *Concentrated* (second)	FT_{last} *Distributed* (second)	FT_{last} *Concentrated* (second)
E2AODVv2	0.029	0.085	500	500	1700	1800
DSR	0.045	0.182	400	400	1700	1500
AODV	0.05	0.202	300	300	1400	1300

Table 3.6: Balancing Energy Consumption metric σ_{EL}^{N}, the first failure time FT_{first} (s), and the last failure time FT_{last} (s), for both *Distributed* and *Concentrated* traffic modes.

The metric for energy consumption balancing in a network consisting of N nodes is σ_{EL}^{N} as discussed earlier. As a sample, the value of σ_{EL}^{N} for the network consists of 12 nodes (i.e., σ_{EL}^{12}) is marked in figure 3.10; per different protocols for the *distributed traffic* mode and similarly in figure 3.11 for the *Concentrated traffic* mode. Also, Table 3.6 summarizes these sample values for σ_{EL}^{12}.

Furthermore, the value of σ_{EL}^{N} decreases as the network size N increases, demonstrating the scalability of the protocols as shown in figure 3.10 and figure 3.11 for *distributed* and *Concentrated* traffic modes, respectively.

For the first variant, *distributed traffic amongst nodes*, all protocols show a similar performance (figure 3.10): increasing the number of nodes leads to improved energy balancing (lower standard deviation). This can be explained due to the chosen traffic model since multiple traffic source/destinations are chosen randomly throughout the simulation, and thus more nodes will participate in the routing process.

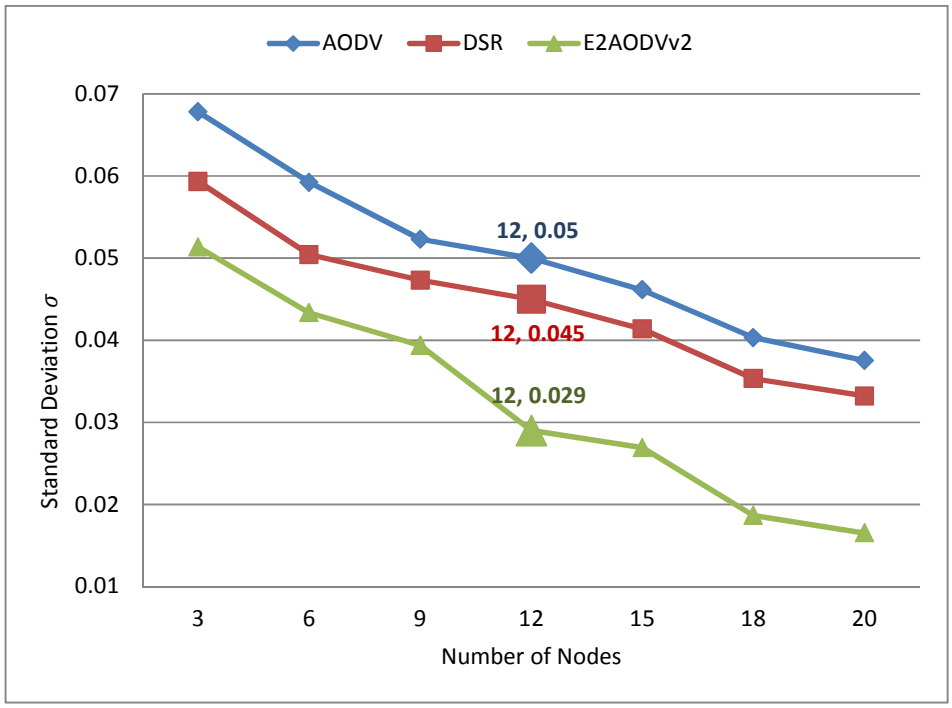

Figure 3.10: Balancing of energy consumption (σ_{EL}^{N}) and the scalability of protocols vs. number of nodes (when distributing traffic amongst nodes).

For the second variant, that targets concentrated *traffic with overloaded nodes*, we chose *n* nodes, amongst all network nodes (*N*), as destination nodes (sink nodes) for the generated traffic in the network. The number of *n* is set to 20% of *N*, which due to the many-to-one traffic patterns, creates overloaded conditions at the destination nodes. Simulation results show that in such scenarios, the standard deviation behaviour is similar to the previous variant, but with worst performance (figure 3.11). However, as in the previous scenario, all three protocols improve in terms of performance as the number of nodes in topology increases (figure 3.11). As both figure 3.10 and figure 3.11 show, the proposed protocol E2AODVv2 reaches a higher performance in terms of battery power efficiency, balancing and scalability in contrast to the baselines.

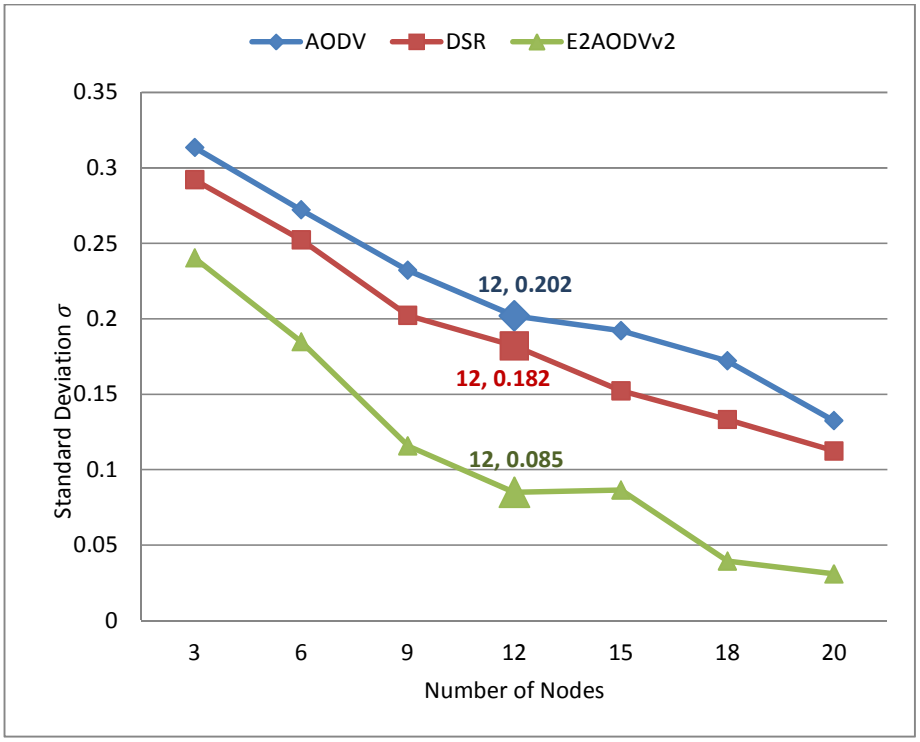

Figure 3.11: Balancing of energy consumption (σ_{EL}^{N}) and the scalability of protocols vs. number of nodes (when traffic is concentrated in few nodes)

ii. Simulation results regarding network lifetime and node failure level

This metric is measured using *movement model II* described in Table 3.5. Similar to the previous case, here again there are two different traffic models: *Distributed traffic amongst nodes* (figure 3.12) and *concentrated traffic with some overloaded nodes* (figure 3.13). We use the first failure time FT_{first} and the last failure time FT_{last} as the network lifetime metrics.

The values of FT_{first} and FT_{last} are marked in figure 3.12 and figure 3.13 and also summarized in Table 3.6. As these results show, E2AODVv2 has always a better value for both FT_{first} and FT_{last}.

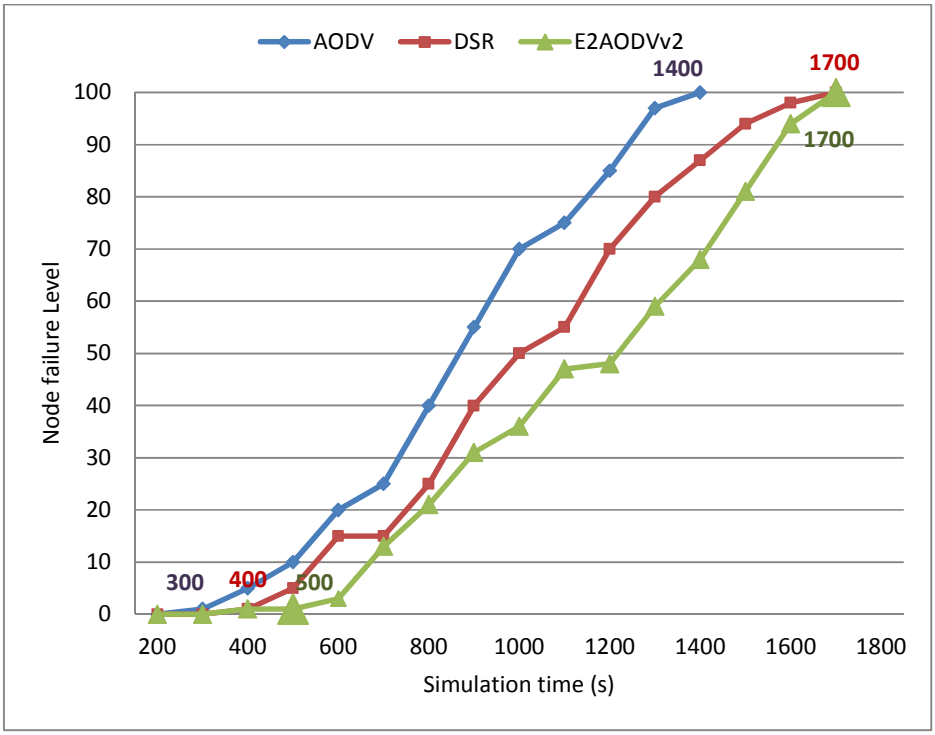

Figure 3.12: Percentage of failed nodes vs. simulation time (when distributing traffic amongst nodes)

Also, as expected, the number of failing nodes increases with simulation time. When nodes use E2AODVv2, their lifetime gets extended due to the energy balancing capabilities of E2AODVv2 and shows a lower *Node failure level* (figure 3.12). This feature is even more pronounced when traffic is concentrated in a few nodes (figure 3.13).

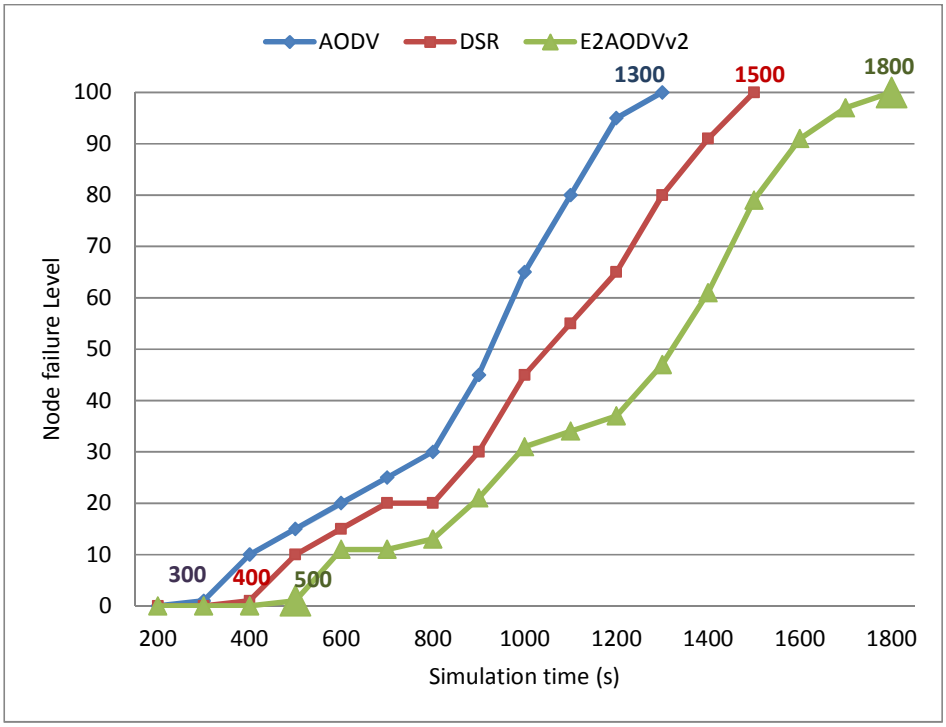

Figure 3.13: Percentage of failed nodes vs. simulation time (when traffic is concentrated in a few nodes)

iii. Simulation results regarding jitter

Jitter has several sources such as network congestion, route changes, and delaying packets in the buffer queues of the intermediate nodes in a source-sink communication session. Usually a jitter buffer is used at sink nodes to counteract the jitter.

Jitter is a measurement for the quality of the communication; a small jitter indicates a high quality communication due to a low latency. As figure 3.14 shows, the performance of the proposed protocol is better than DSR and very close to the original AODV. This can be explained due to the greater complexity of E2AODVv2 for choosing a route which leads to a higher delay.

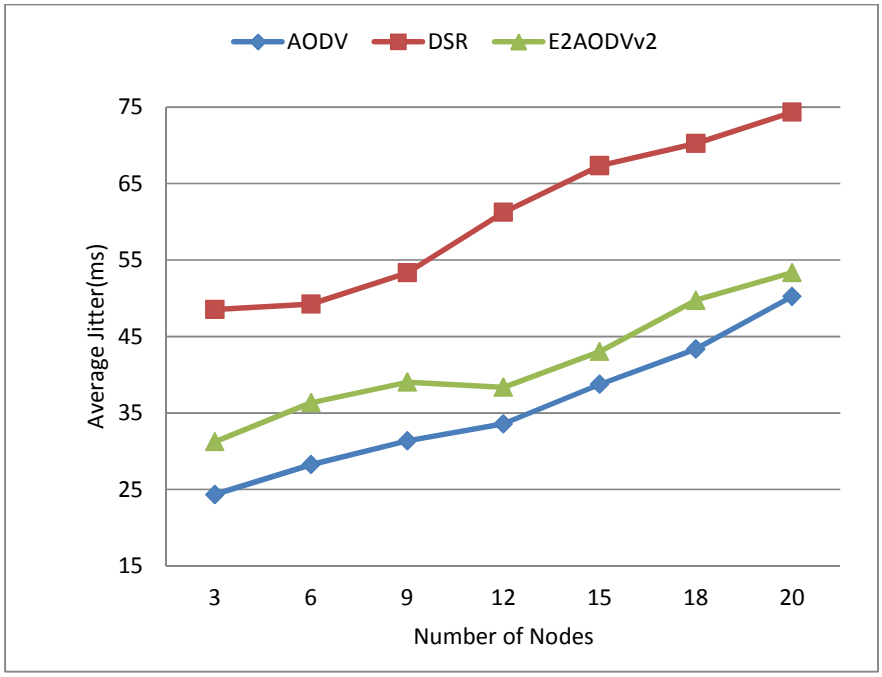

Figure 3.14: Jitter vs number of nodes

3.1.5 Conclusions Regarding E2AODVv2 Protocol

We proposed a new energy efficient and traffic balancing routing protocol based on the well-known IETF AODVv2 protocol, a popular approach for routing in ad hoc networks. The protocol uses a standard generalized MANET Packet/Message format. The E2AODVv2 has been enhanced with battery power efficiency and balancing capability, that can detect the nodes that reach a critical battery level in the network and switch the route in order to avoid network fragmentation and to achieve a higher network lifetime. The same behaviour is also fulfilled when bottlenecks are detected in a specific route in terms of traffic load. These additional functionalities are achieved without the need to create new disruptive approach in the protocol messages. E2AODVv2 achieves a higher performance with respect to energy consumption balancing, scalability, network lifetime, and the percentage of failed nodes in comparison to well-known baseline protocols such AODV and DSR and a jitter value very close to the AODV. As shown by the simulation results, the E2AODVv2 routing protocol specifically outperforms in scenarios where the load of the network is not balanced.

3.2 Load Balanced Dynamic Source Routing (LBDSR)

In previous section we proposed E2AODVv2, a new energy efficient and load balanced routing protocol for ad hoc networks. In this section, we propose a complimentary solution for load balancing and energy efficiency operation in ad hoc networks called Load balanced Dynamic Source Routing (LBDSR).

3.2.1 Introduction

DSR [27] is a reactive (On demand) source routing protocol that was proposed in the infancy of wireless ad hoc networks. Several current ad hoc routing protocols are based on DSR main approach[34]. Here, we apply the energy efficiency and load balancing approach that was proposed in the previous approach toward DSR, propose a new routing protocol called Load balanced Dynamic Source Routing (LBDSR); that is more energy efficient and load balanced. LBDSR aims to be usable not only for MANETs, but also for ad hoc sensor networks who have limited resources. Energy efficiency is of great importance in sensor ad hoc networks due to limitations in terms of battery, memory and processing power.

LBDSR has multipath capability and the intermediate nodes, located in a route between a source and a destination node, are able to extract additional information from the traversing control packets. It uses a new structure for control packets, changes the routing behavior in the nodes, and creates a whole new algorithm for route cache and selection. Although we opt to use DSR, the same principle is applicable for other current flat ad hoc routing protocols. The position of LBDSR in ad hoc routing protocols categories is illustrated by figure 3.1.

3.2.2 Revised Dynamic Source Routing (DSR)

LBDSR uses a modified version of dynamic source routing approach that was discussed in Section 2.1.1. All routing operations in this approach can be categorized into three phases as figure 3.15 shows:

i. Route Request Phase

In a communication between a source node (S) and a destination node (D), S originates a Route REQest (RREQ) message, and broadcasts this to all of its neighbours. These RREQs are then flooded in the network until they reach D. An intermediate node, which does not know the route to D, should forward the RREQ to its neighbours. The intermediate node will also drop the repeated request for the same destination and will not forward them anymore, as shown by figure 3.16.

ii. Route Reply Phase

As illustrated in figure 3.17, when the RREQs finally reach the destination node, another control message, which is called Route REPly (RREP), will be originated by the destination node. The intermediate nodes who have, in their route cache, an entire route towards that destination node also can immediately reply to the RREQ originated by the source node (known as optional *gratitude reply*). If the RREQ is repeated, it will be dropped. If none of the above cases occur, the intermediate node forwards the RREQ to its neighbours.

iii. Route Error Phase

An intermediate node may originate a control message, called a Route ERRor (RERR), in the two main scenarios. In the first case, the intermediate node does not have a valid route for the destination of a received data packet and consequently the packet is undeliverable. In the second case, the intermediate node detects a link breakage, as shown in figure 3.18. All the nodes, including the source host, that are notified with the RERR packet, will remove all routes in their route cache that use the 'broken' link.

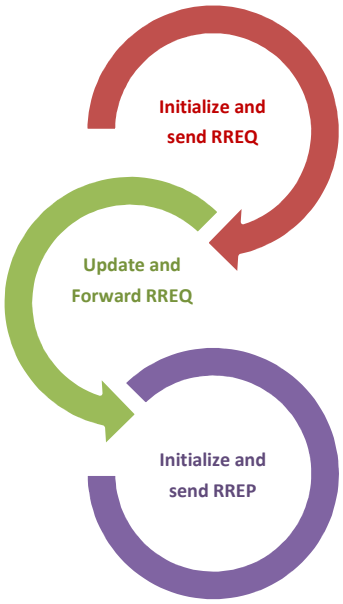

Figure 3.15: Three phases of source routing

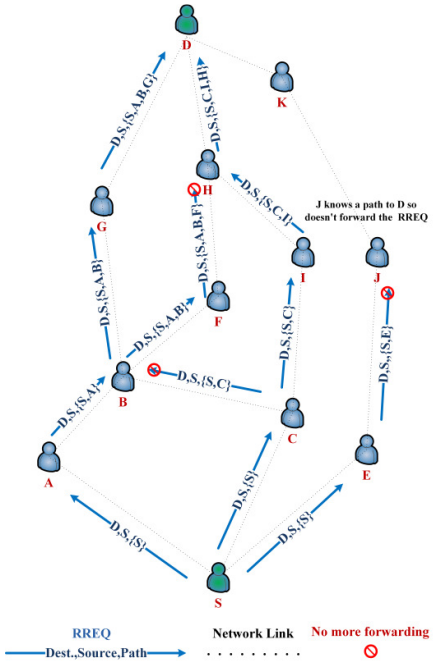

Figure 3.16: Route discovery phase by RREQ control message

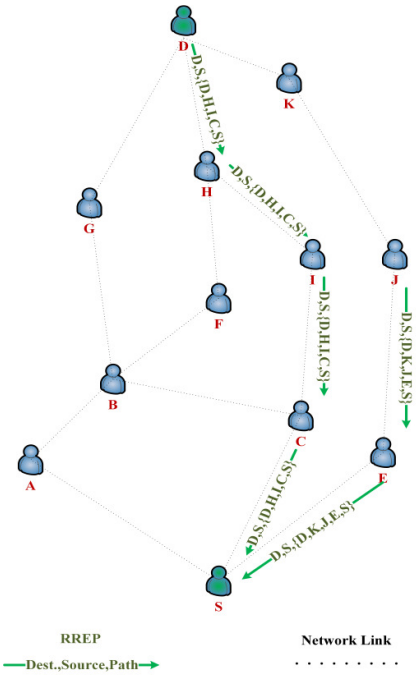

Figure 3.17: Route establishing by RREP control message

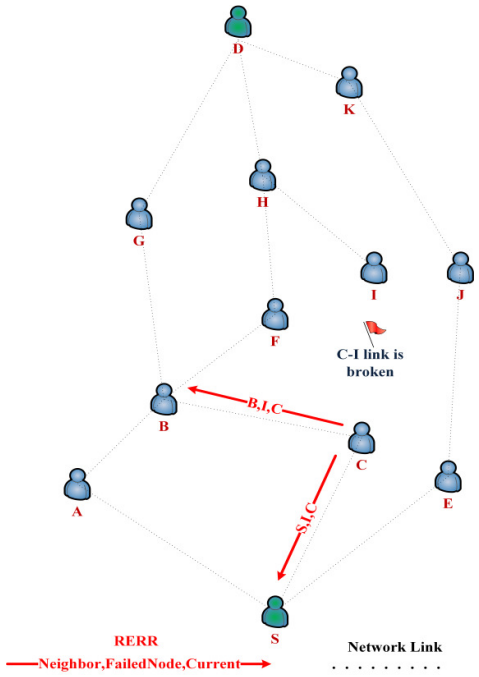

Figure 3.18: Route error message by RERR control message

3.2.3 Routing Mechanism of LBDSR

3.2.3.1 Structure of control packets

LBDSR protocol has three main control packets that are Route REQest (RREQ), Route REPly (RREP), and Route ERRor (RERR). This section describes these control messages in more details.

i. Structure of RREQ in LBDSR

The RREQ control message of LBDSR, like DSR, has several fields such as:

1. *pktType*: generally can be the RREQ or RREP or RERR.
2. *seqNumber*: unique identifier number of RREQ that is generated by the source node
3. D: destination address
4. S: source address
5. *Path*: nodes that the RREQ passs to reach the current node [21]. The path of current RREQ is the IPv4 address of all passed through nodes.
6. *HopCount*: number of intermediate nodes that RREQ passes through. The maximum number of intermediate nodes allowed to forward that copy of the RREQ is *HopLimit* [21].
7. Energy filed: discussed in Section 3.1.3.
8. Traffic filed: discussed in Section 3.1.3.

The structure of RREQ in LBDSR is shown in figure 3.19.

```
typedef struct
{
DSR_PacketType pktType; //will be RREQ
NODE_ADDR srcAddr;
NODE_ADDR targetAddr;
int seqNumber;
int hopCount;
NODE_ADDR path[HopLimit];
int TotBat, MinBat, and CritBat; //energy parameters
int TotTra and MaxTra;//traffic parameters
} LBDSR_RouteRequest;
```

Figure 3.19: Structure of RREQ control packet in LBDSR

i. Structure of RREP in LBDSR:

The RREP control message of LBDSR has a similar structure to RREQ, as illustrated by figure 3.20.

```
typedef struct
{
DSR_PacketType pktType; //will be RREP
NODE_ADDR srcAddr;
NODE_ADDR targetAddr;
int seqNumber;
int hopCount;
NODE_ADDR path[HopLimit];
int TotBat, MinBat, and CritBat; //energy parameters
int TotTra and MaxTra;//traffic parameters
} LBDSR_RouteReply;
```

Figure 3.20: Structure of RREP control packet in LBDSR

ii. Structure of RERR in LBDSR

The structure of RRER in LBDSR is the same as in DSR [21], as illustrated by figure 3.21.

```
typedef struct
{
DSR_PacketType pktType; //will be RERR
NODE_ADDR srcAddr; //*Originator of the Route Error*/
NODE_ADDR destAddr;//*Source of the broken route*/
NODE_ADDR unreachableAddr;//*Immediate downstream of broken link*/
int hopCount;
BOOL salvaged;
NODE_ADDR path[HopLimit];
} LBDSR_RouteError;
```

Figure 3.21: Structure of RERR control packet in LBDSR

3.2.3.2 Routing behavior of nodes

The routing behavior of the nodes in the LBDSR remains like E2AODVv2, as discussed in Section 3.1.3. Some of these parameters are summarized in Table 3.7. These parameters were discussed in Section3.1.3.

The route selection process of LBDSR is similar to E2AODVv2 that was discussed in Section 3.1.3. As a reminder, the route priority function that determines the priority of $Route_i$ between S and D in figure 3.5 in E2AODVv2 was calculated by the following equation:

$$RoutePrio(i) = \frac{E_i}{T_i} \quad (3.12)$$

Parameter	Definition	Value
$K_{Inter.forward}$	The maximum number of RREQs (generated by S toward D and belonging to the same flow) that can be forwarded by an intermediate node.	3
$T_{Inter.Wait}$	An intermediate node only forwards the RREQs (generated by S toward D and belonging to the same flow) before this timeout.	2s
CBPL	Critical Battery Power Level	3

Table 3.7: Some parameters which define the routing behavior of the nodes in LBDSR

Both E_i and T_i parameters were discussed previously in Section 3.1.3. The route priority function that determines the priority of $Route_i$ between S and D in figure 3.5 for LBDSR is:

$$RoutePrio(i) = \frac{K_E \times E_i}{K_L \times L_i + K_T \times T_i} \quad (3.13)$$

Where L_i is the length parameter of $Route_i$ and shows the priority of route i with respect to the length of the $Route_i$:

$$RoutePrio(i) = \frac{Actual\_Length\_Route_i}{\text{Max\_length}} \quad (3.14)$$

Where *Actual_Length_Route(i)* is the actual length of route $Route_i$ (i.e., the number of hops in route i) and *Max_Length* is the maximum length that a route can take in DSR routing protocols.

K_E, K_L, and K_T are the coefficients of energy, length, and traffic parameters, respectively. In our results, $LBDSR_L$ is a customized version of LBDSR for achieving better delay and jitter, and $LBDSR_E$ is a customized version of LBDSR for achieving better energy efficiency. Therefore, we can achieve higher LBDSR performance with regard to end-to-end delay and jitter ($LBDSR_L$), or Energy consumption ($LBDSR_E$) by adjusting the coefficients K_E, K_L, and K_T, as summarized by Table 3.8.

Protocol name	K_E	K_L	K_T
LBDSR	1	1	1
$LBDSR_L$	1	2	1
$LBDSR_E$	2	1	1

Table 3.8: Different customized versions of LBDSR by adjusting the coefficient K_E, K_L, and K_T

Other alternative functions can be also used without impacting the generality of the proposed approach. Based on a variety of workloads and scenarios for simulation, the coefficient of parameters can be determined to be a trade-off for improving end-to-end delay, traffic balancing, and power consumption balancing all together.

3.2.4 Evaluating LBDSR

This section presents ns2 simulation results for LBDSR protocol. The movement and communication models are summarized in Table 3.9. LBDSR is evaluated according to the following metrics: average jitter, balancing of energy consumption, and node's failure level.

i. Simulation results regarding jitter

Regarding Jitter, values of 20 msec to 50 msec can be acceptable for the average jitter value based on the type of the services (real time or stream) and QoS metrics [72]. All analysed protocols present an ascending profile when the number (figure 3.22) or the mobility of nodes (figure 3.23) are increased.

As expected, AODV performs better than DSR and LBDSR. Both are reactive protocols; the first uses a normal routing approach while the latter is a source routing protocol. Source routing protocols normally present longer delay and jitter because their route discovery takes more time, as every intermediate node tries to extract information before forwarding the reply, and because data packets are forwarded hop by hop through the path using the source routing method.

*Parameters of **movement model I**, characterized by **number of nodes***	
Topology area	500m × 500m
Maximum mobility of nodes	5 m/s
Number of nodes	**1…20**
Simulation time	300s
*Parameters of **movement model II**, characterized by maximum node **speed***	
Topology area	500m × 500m
Maximum mobility of nodes	**0m/s…5m/s**
Simulation time	300s
Number of nodes	20
*Parameters of **movement model III**, characterized by **simulation time***	
Topology area	500m × 500m
Maximum mobility of nodes	5 m/s
Number of nodes	20
Simulation time	**200…1300s**
*Parameters of **traffic** model*	
Traffic sources	CBR
Data packets size	512 bytes
Sending rate	8 packets/second

Table 3.9: Parameters of movement models and communication model

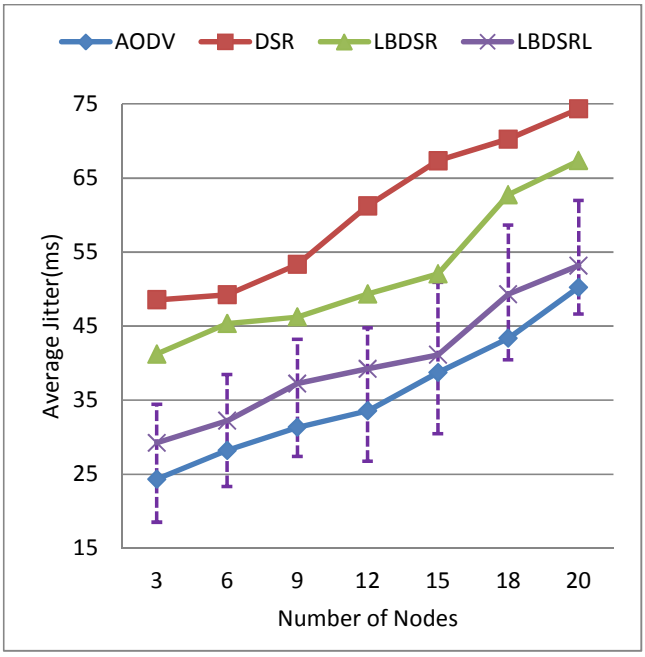

Figure 3.22: Jitter vs. number of nodes

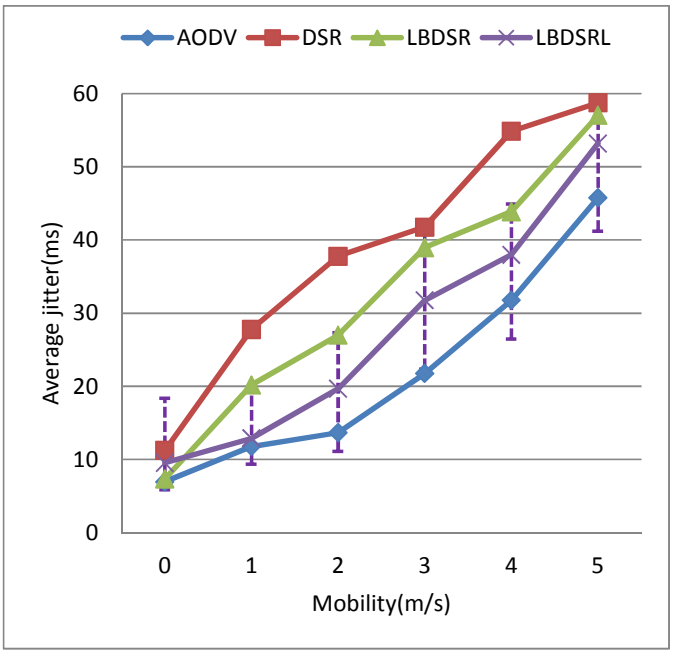

Figure 3.23: Jitter vs. mobility

As mentioned before, LBDSR *Route Priority* function can be customized to choose routes that minimize jitter and end-to-end delay by using a higher coefficient for the length parameter (K_L). The behaviour of this version of LBDSR is shown in both figure 3.22 and

figure 3.23 as $LBDSR_L$. Therefore, as the figures suggest, the $LBDSR_L$ protocol has a better average jitter in comparison to AODV, original DSR, and LBDSR. The maximum and minimum average of $LBDSR_L$ is shown by the purple dot which varies around the average jitter value of $LBDSR_L$.

ii. Simulation results regarding balancing energy consumption and scalability

For this metric, simulation results are presented in terms of battery power (energy) consumption balancing. Also, the energy parameter (K_E) of LBDSR *Route Priority* function can be customized to achieve higher energy efficiency; this version of LBDSR is called by $LBDSR_E$ in the simulation results that follow. The traffic model is the one presented in Table 3.9 with the following variants:

i. *Distributed traffic amongst nodes*: traffic sources and destinations will be chosen randomly in time.

ii. *Concentrated traffic with overloaded nodes*: traffic destinations will be predefined (all traffic in the network goes toward those sink nodes).

These two variants were discussed before in Section 3.1.4.

For the first variant, *distributed traffic amongst nodes*, all protocols achieve similar performance (figure 3.24), obtaining a higher balancing (low standard deviation) for energy consumption as the number of nodes increases. This can be explained due to the chosen traffic model; since multiple traffic source/destinations are chosen randomly throughout the simulation, more nodes will participate in the routing process.

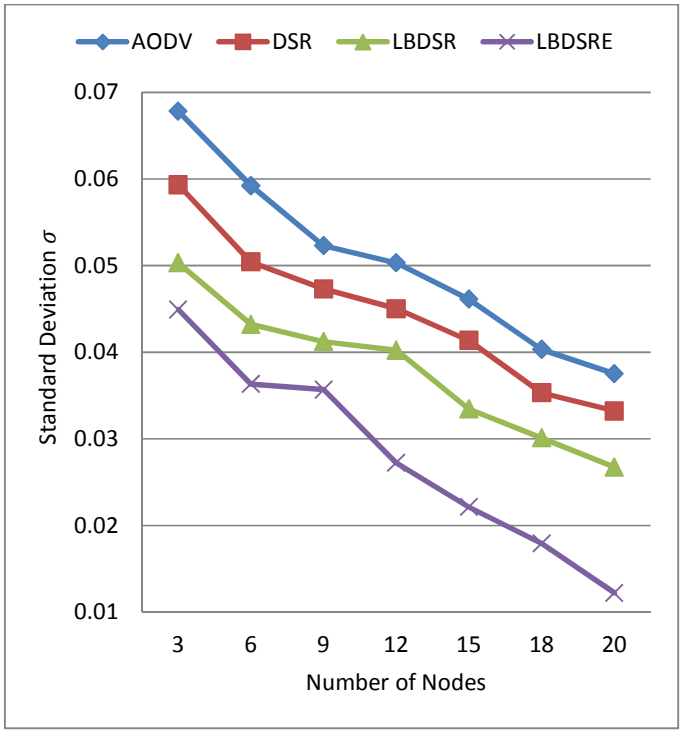

Figure 3.24: Balancing of energy consumption (σ_{EL}^{N}) vs. number of nodes (when distributing traffic amongst nodes).

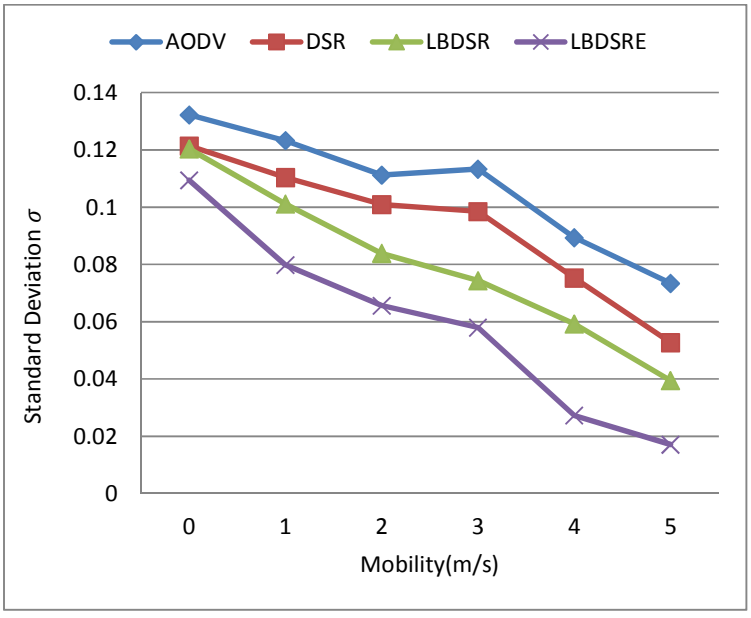

Figure 3.25: Balancing of energy consumption (σ_{EL}^{N}) vs. mobility of nodes (when distributing traffic amongst nodes).

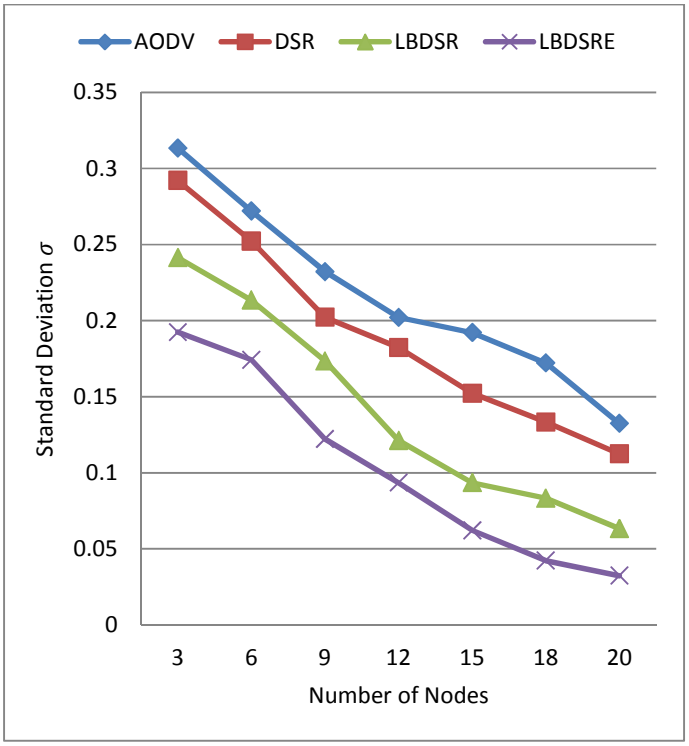

Figure 3.26: Balancing of energy consumption (σ_{EL}^{N}) and the scalability of protocols vs. number of nodes (when traffic is concentrated in few nodes)

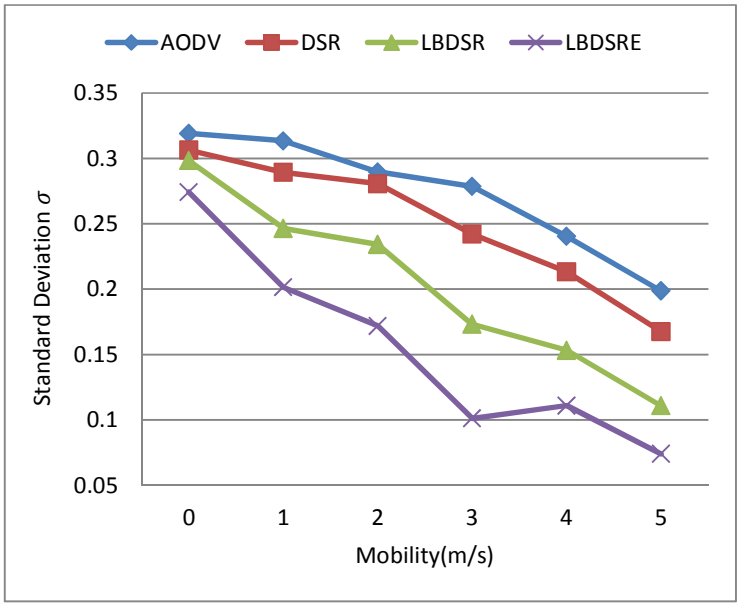

Figure 3.27: Balancing of energy consumption (σ_{EL}^{N}) vs. mobility of nodes (when traffic is concentrated in few nodes)

By increasing mobility, we also achieve better balancing in the energy consumption (figure 3.25). Increasing mobility implies a more dynamic topology, in which almost all nodes participate in the routing process. Having such condition leads to equality in energy consumption leading to a low standard deviation.

Comparing both figure 3.24 and figure 3.25, we can see that by increasing the mobility, all protocols perform a little worse; this can be explained by the fact that greater mobility leads to more broken links decreasing the stability of the routing paths. This behaviour results in a worse delivery ratio and, as a result, greater retransmissions and energy consumption.

For the second variant, *concentrated traffic with overloaded nodes*, we chose *n* nodes amongst all network nodes (*N*), as destination nodes (sink nodes) for the generated traffic in the network. The number of *n* is set to 20% of *N*, which due to the many-to-one traffic pattern creates an overload condition at the destination nodes.

Simulation results show that in such a scenario, the standard deviation behaviour is similar to the previous variant, but with worse performance (figure 3.26 and figure 3.27). As in the previous scenario and for the same reasons, the performance of all three protocols improves as the number of nodes in topology increases (figure 3.26). The same applies when mobility is increased (figure 3.27).

iii. Simulation results regarding network lifetime and node failure level

This metric is measured using movement model III, described in Table 3.9. Like in the previous case, again we have two different traffic models: Distributed traffic amongst nodes (figure 3.28) and concentrated traffic with overloaded nodes (figure 3.29). The percentage of failed $\text{nodes} = 0$ means none of the nodes have failed and the percentage of failed nodes=100 means all of them have failed. As expected, the number of failing nodes increases as simulation time increases.

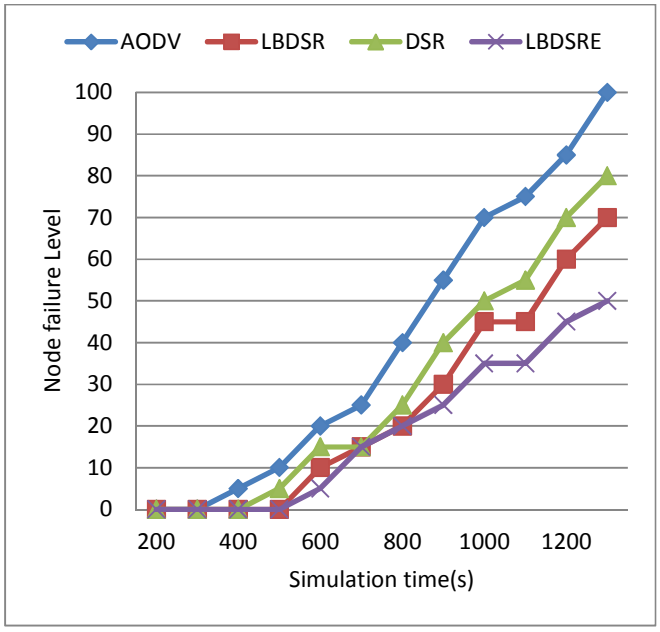

Figure 3.28: Percentage of failed nodes vs. simulation time (when distributing traffic amongst nodes)

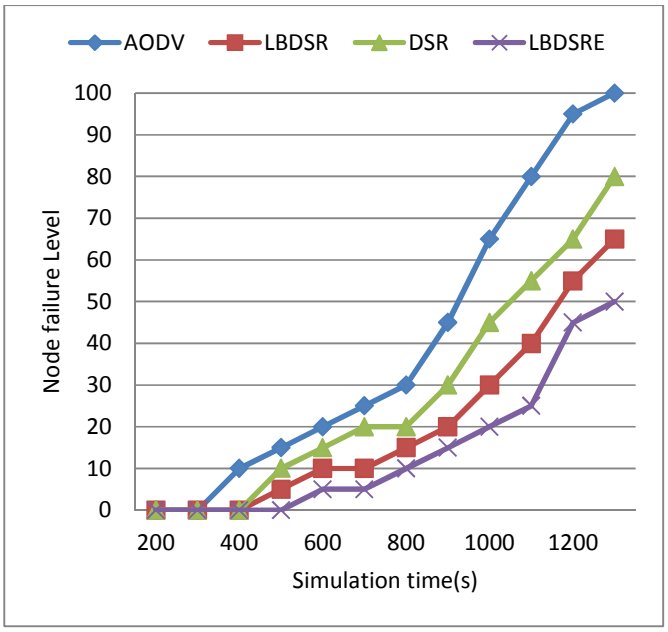

Figure 3.29: Percentage of failed nodes vs. simulation time (when traffic is concentrated in a few nodes)

When nodes use LBDSR, their lifetime gets extended due to the energy balancing capabilities of LBDSR. This is even more clear for the energy shaped $LBDSR_E$, especially when traffic is concentrated in a few nodes (figure 3.29).

3.2.5 Conclusions Regarding LBDSR Protocol

We modified the control packets, routing tables, and the route selection method resulting in the enhanced Load Balanced Dynamic Source Routing (LBDSR) approach. Performance statistics with respect to traffic load balancing, energy consumption balancing, average end-to-end delay, and route's reliability metrics were measured. As shown by the simulation results, the LBDSR protocol outcomes legacy approaches specially in scenarios where the load of network is not balanced. Also LBDSR can be customized to operate in different network scenarios to either to be more energy efficient, or to be jitter and delay aware. In this case, LBDSR shows an improvement of up to 30% with respect to these metrics. The proposed approach is technology agnostic and can apply to any legacy reactive routing protocols.

CHAPTER 4

4 Network Codes for Multi-hop Networking Technology

Network Coding is a promising technique that combines several input packets into several output packets, thus improving the throughput and energy consumption. Due to the multi-hop approach of ad hoc networks and the broadcast nature of the wireless channels, the store-code-forward approach of NC can complement the ad routing protocols in the network layer. NC can reduce the number of transmissions in unicast and multicast communications and the number of re-transmissions in lossy networks whilst increasing the data transfer rate. All these capabilities directly translate to greater energy saving per transferred bit of data. However, due to the role of intermediate nodes in processing (coding) of the transitive packets, network coding can have either positive or negative security aspects. Eavesdropping attack and packet corruption (by erroneous channels or byzantine intermediate attacker nodes) have a severe effect on NC based systems performance and its integrity. This chapter presents the state of the art in security mechanisms for NC based systems, the taxonomy of current mitigations techniques, and finally proposes two novel mitigation techniques against eavesdropping attack and packet corruption in NC based systems.

4.1 Introduction

Network coding (NC) [5] constitutes a capacity achieving solution to packet multicasting. In the *store-code-forward* paradigm of NC systems, the source messages are not only something to be merely routed, but are subject to algebraic operations resulting in encoded packets. This contrasts with current state-of-the-art routing solutions, where source messages are merely routed from source to destination.

In 2003, [73] demonstrated that linear operations at the nodes were sufficient to achieve the max-flow min-cut bound: linear network coding (LNC) was defined in directed

acyclic networks with single-source multicast. Side by side, [74] provided an algebraic formulation of linear network coding and demonstrated equivalent results in the new framework for both acyclic and cyclic networks. This algebraic formulation opened the way to Random Linear Network Coding (RLNC) [75], a type of network codes in which the coefficients of linear combinations are randomly chosen over a finite field.

4.1.1 Principles of Random Linear Network Coding

In the RLNC approach, the source generates a linear combination of its original packets using coefficients in a finite field chosen uniformly at random from the elements of the field. Intermediate nodes, on the other hand, perform this random linear combination using the coded packets previously received and that stored in the node's buffer [76] . RLNC enables distributed NC protocols in which a node without any network state information can randomly and independently choose a set of coefficients to perform a linear combination of two or more received data packets, and then sends the resulting coded packet(s) to its output links. This operation was proven to be asymptotically optimal with the field size [75].

However, the capability of designing network codes without knowledge about the network is paid in terms of successful decoding probability: in fact, in randomised scenarios, the capability of a receiver to completely decode the information depends on the number of sinks and the size of the finite field. This also introduces a trade-off between the decoding error probability and the complexity of the coding operations over the finite field. In case of an acyclic network, the lower bound on the successful decoding probability is $P_s = (1 - \frac{d}{q})^{|E|}$, where d is the number of destinations, q is the size of the finite field and E is the set of edges. The value of the probability approaches 1 when $q \to \infty$ [77]. Figure 4.1 shows P_s versus number of edges for different number of sink nodes in the network where $q = 256$.

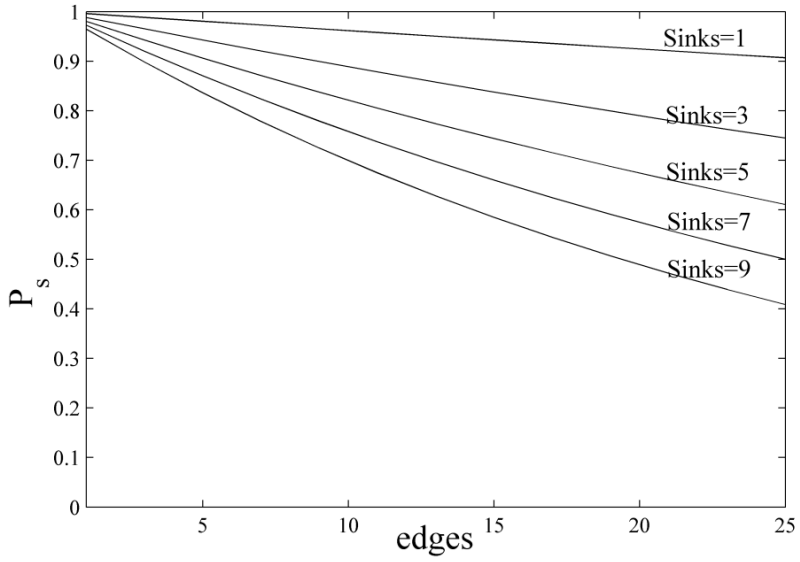

Figure 4.1: P_s vs. number of edges

Figure 4.2 shows a representation of a random linear network code in the butterfly network with correlated source processes. Either all the coefficients of the linear combinations or a part of them are randomly chosen over a finite field of size q. In particular, X denotes the source processes, Y the processes at the edges, and Z the ones collected by the sinks. Matrices A and B are random matrices that contains the coefficients, matrix F is the adjacency matrix of the directed labelled line graph [74]. The transfer matrix of the system is M.

The randomized approach of RLNC is distributed, easier to implement than LNC, and especially suitable for changing topologies, large networks or in the presence of dynamically varying connections, e.g., ad hoc networks. The key characteristic of RLNC is that coefficients of linear combinations are chosen randomly over a finite field. This implies the transmission of coding vectors to the receivers by appending them as an overhead to the header of the messages. This overhead is quantified as $h \log q$, where h is the number of information flows at the source and q is the size of the finite field [77].

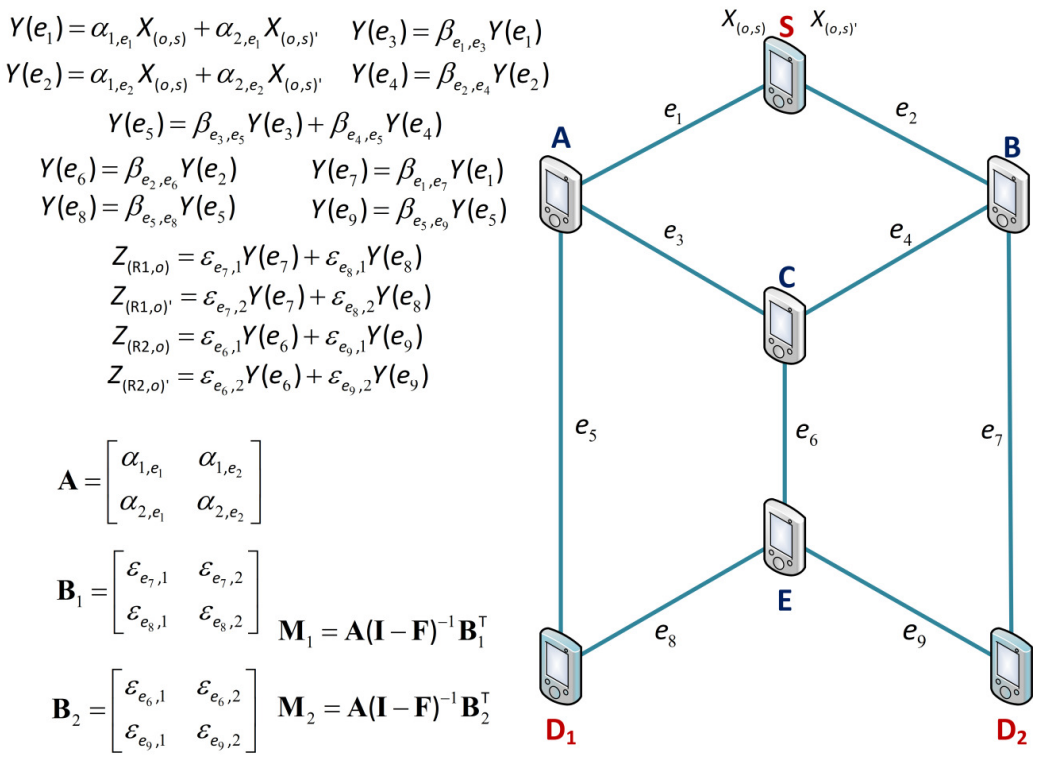

Figure 4.2: an example of random linear network code in a butterfly scenario

In general, RLNC can be designed in practice according to two main approaches, called respectively intra-session and inter-session. In the former [78-81], routers combine packets belonging to the same session. It is typically used in multicast application and in case of unpredictable topologies, and it has been demonstrated to improve reliability. The decoding operations are only performed at the destination. In the latter [82-86], packets from distinct flows are mixed when they pass through a common router. This approach is especially suitable for unicast applications and static topologies.

4.1.2 A General Model for RLNC

figure 4.3 shows a basic scenario based on a line network in which there is a RLNC based communication via three nodes: A source node *encodes* the native data packets and floods them into the network; an intermediate node *recodes* the incoming encoded packets and forwards them toward a sink node, and the sink node *decodes* the incoming coded packets to extract the native packets. Any generalization of NC systems will be based on the same three phases as illustrated in figure 4.3. Per each information flow, the source

node floods g native data packets by size s into the network. All nodes know the setup parameters g and s. The required steps for this RLNC based communication are discussed in the following:

i. Source node: encoding operations

The source generates g native data packets; each of them can be considered as a vector $\boldsymbol{p_i} = [\alpha_i^1\ \alpha_i^2\ \dots \alpha_i^s]$. These g packets could be represented by the following matrix:

$$\mathbf{P} = \begin{bmatrix} \alpha_1^1 & \cdots & \alpha_1^s \\ \vdots & \ddots & \vdots \\ \alpha_g^1 & \cdots & \alpha_g^s \end{bmatrix} \tag{4.1}$$

Where each element α_j^i, $i = 1,2, \dots, s$ and $j = 1,2, \dots, g$ is a symbol in a finite filed $\mathbb{F}_q$ by size q (where $q = 2^m$); consequently the size of matrix **P** is $g \times s \times m$ (bits). The size of filed, q is a design parameter for the RLNC based system. After generating matrix **P**, the source node randomly chooses $n \times g$ symbols from $\mathbb{F}_q$ (each symbol by probability $1/q$) and forms a coefficient matrix:

$$\mathbf{C} = \begin{bmatrix} c_1^1 & \cdots & c_1^g \\ \vdots & \ddots & \vdots \\ c_n^1 & \cdots & c_n^g \end{bmatrix} \tag{4.2}$$

Then the source node encodes g native packets into n encoded packets ($n \geq g$) in a form of an RLNC encoded message as follows:

$$\mathbf{M} = \mathbf{C} \times \mathbf{P} = \begin{bmatrix} m_1^1 & \cdots & m_1^s \\ \vdots & \ddots & \vdots \\ m_n^1 & \cdots & m_n^s \end{bmatrix} \tag{4.3}$$

Now, the source node concatenates the coefficient matrix **C** as packet header to the encoded message **M** and generates matrix **A**:

$$\mathbf{A} = [\mathbf{C}|\mathbf{M}] = \begin{bmatrix} \beta_1^1 & \cdots & \beta_1^{g+s} \\ \vdots & \ddots & \vdots \\ \beta_n^1 & \cdots & \beta_n^{g+s} \end{bmatrix} \tag{4.4}$$

Finally, the source node sends **A** to the intermediate node in the scenario illustrated by figure 4.3. All the matrices elements and the coding operations in RLNC will remain in $\mathbb{F}_q$.

Considerations related to the encoding phase: Matrix **A** is called *generation* (also called *chunk, batch*, or *class*). Each block of data by size B bits may be sent in h generations (information flows). Any row of generation matrix **A** (i.e., $\boldsymbol{\gamma_i} = [\beta_i^1 \dots \beta_i^{g+s}]$, $i = 1,2, \dots, n$) is called a *codeword*. The number of codewords (i.e., n) is another design parameter. The sink node finally should receive at least g linear independent codewords to be able to extract the g native data packets therefore $n \geq g$ in the source node side. $r = n - g$ determines the *redundancy* of the RLNC code and $\varepsilon = g / n$ shows the optimality of the code or *code rate* ($0 < \varepsilon \leq 1$). Therefore $\varepsilon = 0.5$ leads to 50% redundancy in the code, implying that the transmission rate of the native information is not more than 50%. When there are erasure channels in the network, the source node may need to produce a higher amount of codewords (i.e., $n \gg g$) to assure that the sink node will finally receive sufficient linear independent codewords. This can lead to a higher redundancy and lower code rate.

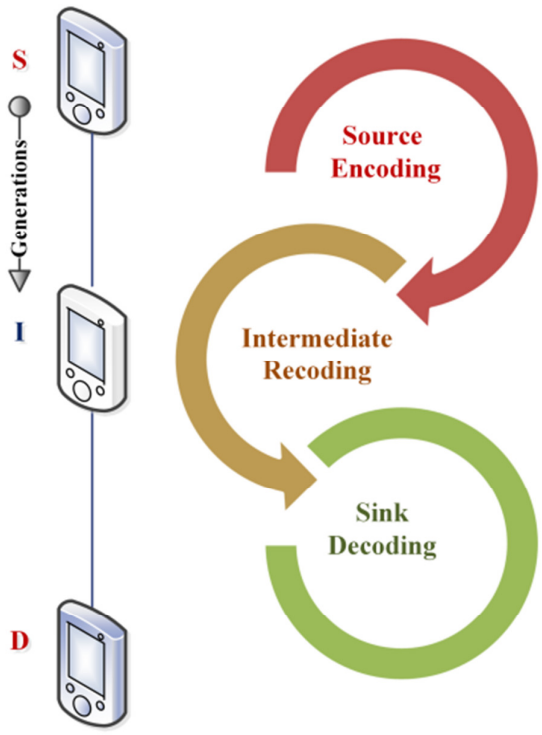

Figure 4.3: An example of RLNC based system with three mobile nodes: Any generalization of NC systems will be based on the same three phases.

The coefficients matrix $\mathbf{C}$, that should be created and interpolated in the header of each flow of information, can be considered as an *coefficients overhead* of RLNC algorithm. This overhead is quantified as $h \log q$, where h is the number of information flows (total generations) at the source and q is the size of the finite field. Unlike RLNC, in Linear Network Coding (LNC) mechanism, instead of randomly choosing the coefficients (elements of matrix $\mathbf{C}$) from $\mathbb{F}_q$, each node in the network uses its own fixed and pre-defined coefficients for all the streams of data.

ii. Intermediate node: recoding operations

As a possible design for the basic scenario illustrated by figure 4.3, when intermediate node receives all n codewords from the source node, it starts the *recoding* operations. As the first step, the intermediate node randomly chooses $l \times n$ symbols from $\mathbb{F}_q$ and forms a coefficient matrix:

$$\mathbf{D} = \begin{bmatrix} d_1^1 & \cdots & d_1^n \\ \vdots & \ddots & \vdots \\ d_l^1 & \cdots & d_l^n \end{bmatrix} \tag{4.5}$$

As the next step, the intermediate node recodes the receiving n codewords into l new recoded codewords by the following operations:

$$\mathbf{A}' = \mathbf{D} \times \mathbf{A} = \begin{bmatrix} b_1^1 & \cdots & b_1^{g+s} \\ \vdots & \ddots & \vdots \\ b_l^1 & \cdots & b_l^{g+s} \end{bmatrix} \tag{4.6}$$

At the end of this phase, the intermediate node forwards these l new recoded codewords (i.e., matrix $\mathbf{A}'$) to the sink node. Again, like the encoding phase in the source node, the number of new codewords generated by the intermediate nodes in the recoding phase (i.e., l) is another design parameter.

Considerations related to the recoding phase: The intermediate node can apply different policies for the recoding operations. Each intermediate node, that has e input and u output links, buffers the input coming codewords. Then it may start the recoding process in each time interval like τ or by having at least ω codewords in its input buffer. Both τ and ω in these different policies are design parameters and should be determined carefully

to minimize delay. Each intermediate node may choose τ and ω values independent from the other nodes. Also, the network may be based on an adhoc networking technology in which the topology of the network could be dynamic and the value of e and u can vary during the network lifetime. In a general scenario, the intermediate node may receive the codewords from different paths. Also, the intermediate node may mix codewords from different *flows* of data (called *inter-flow* or *inter-session* network coding), or it can recode and mix only the codewords of the same flow (called *intra-flow* or *intra-session* network coding).

iii. Sink node: decoding operations

Each receiving codeword in the sink node has two parts: 1) *header*: the first g symbols of the codeword which are the coefficient symbols. 2) t*railer*: the last s symbols of each codeword which is the real coded packet. As mentioned before, the sink node knows the setup parameters g and s. The sink node keep checking the rank of the total received codewords belonging to the same generation. When the rank of these receiving codewords reaches to g; it means that the sink node has received g linear independent codewords (called also *innovative* codewords) and it can immediately start the decoding process. Suppose that the sink node forms a matrix by the trailer of the g linear independent codewords represented by the following matrix:

$$\mathbf{T} = \begin{bmatrix} t_1^1 & \cdots & t_1^s \\ \vdots & \ddots & \vdots \\ t_g^1 & \cdots & t_g^s \end{bmatrix} \tag{4.7}$$

Also, the sink node forms a coefficient matrix by the header of these codewords:

$$\mathbf{H} = \begin{bmatrix} h_1^1 & \cdots & h_1^g \\ \vdots & \ddots & \vdots \\ h_g^1 & \cdots & h_g^g \end{bmatrix} \tag{4.8}$$

The matrix of coded packets, coefficients, and native packets (i.e., $\mathbf{T}_{g\times s}$, $\mathbf{H}_{g\times g}$, and $\mathbf{P}_{g\times s}$, respectively) builds a systems of linear equations like $\mathbf{T} = \mathbf{H} \times \mathbf{P}$. Now, the sink node, by having $\mathbf{T}$ and $\mathbf{H}$, can extract the g native data packets by solving this system via Gaussian elimination method. Matrix $\mathbf{P}$ also can be represented as follows:

$$\mathbf{P} = \mathbf{H^{-1}} \times \mathbf{T} \tag{4.9}$$

Considerations related to the decoding phase: After successful decoding operations, the sink node *may* send an *Ack message* back to the source node as a confirmation for receiving a generation and requesting the next generation. Beside this *optional Ack* mechanism, another policy could be waiting for a certain amount of time between the transmission of the two generations in the source node.

4.1.3 NC Protocols Categorization

Different categorizations of NC aware routing protocols for wireless networks, namely centralized [87], source routing [88-90], hop-by-hop [78, 82, 91-95], and active [96, 97], are presented and reviewed in [6]. In general scene, NC protocols can be classified in two main groups based on their use of network state information:

*i) Stateless NC protocols***:** The stateless NC protocols do not rely on network state information such as topology, link cost, and node location, to perform coding operations like the mixing of data packets. In stateless NC protocols, nodes do not rely on any assumptions about network topology, accordingly coding operations in a communication (coding at source node(s), recoding in intermediate node(s), and decoding in sink node(s)) are independent from dynamically changing topologies. RLNC is a stateless NC approach in which only the sink nodes who have access to sufficient decoding vectors can recover native packets [7, 98].

ii) State-aware NC Protocols: The state-aware NC protocols rely on partial or full network state information to optimize the coding operations carried out by each node. Here, nodes have some information about network topology; hence, they can use local information to achieve the most optimized encryption codes. Also, they can exchange the required coding vectors at the beginning of communication. The optimization process in the state-aware NC protocols may target the throughput or the delay. The optimization process in a node can be based on exchanging information only with close neighbours (local optimization) or it can address the end-to-end communication across the entire network (global optimization). The COPE [15] protocol, which runs between the IP and MAC layers, is a state-aware NC protocol that uses local information and network topology information.

Table 4.1 lists the benefits and drawbacks of the state-aware and the stateless NC protocols especially in terms of security issues.

Protocol Type	Key Features	Benefits	Drawbacks
Stateless NC	Do not rely on network state information	Do not use control packets for updating their knowledge about topology state	Need more sophisticated coding operations
	Coding operations can work properly even under a dynamic network topology	More prone against wrong network state information and invalid control traffic packets that leads to be more immunity against some types of fabrication, modification, and impersonation attacks.	To guarantee a successful decoding, coefficients should be selected from a sufficiently large field which increases the overhead.
	Nodes chose encoding coefficients randomly and independently	Extracting information from received packets in receivers is independent from the identity of sender nodes	The required decoding coefficients vector should be included in the header of each data packets.
State-aware NC	Rely on local (one or two levels of neighbors) or global network state information	Instead of including encoding coefficient at the header of all packets, the fixed optimized coefficients codes can be exchanged at the beginning of communication.	Rely on network state information, routing tables, control packets, and so on.
	Nodes can use network state information to achieve the most optimized encoding codes.	Local or global optimization schemes	Rely on vulnerable control packets which lead to more threats and attacks

Table 4.1: Benefits and drawbacks of state-aware and stateless NC protocols

4.2 Overview of Security Attacks Against NC Systems

Beside general features such as capability for sending and receiving packets, and being equipped with a normal processing power and storage device, such as CPU and memory, we expect several rules that should be followed by a well behaved, benign node in a NC based system [99, 100]:

i. *Coding:* Performing valid coding operations such as mixing, encoding different data packets, and contributing actively and correctly in the overall *store-code-forward* mechanism.
ii. *Recoding:* Recoding the data packets that are merely intended for it, therefore satisfying the basic confidentiality requirements and enabling the sink nodes to correctly decode data the packets and extract the information. Then the recoded packets should be forwarded correctly and validly.
iii. *State Dissemination:* Participating in the timely dissemination of correct state information (applicable in the state-aware NC protocols) [11].

When a node violates one or more of these so called good behaviours rules, the NC system will be vulnerable to several attacks. There is a wide range of security threats and attacks in NC based systems. Furthermore, even in a network composed of benign nodes, lossy or erroneous channels may lead to corrupted codewords at the sink nodes.

We shortly review general security threats that are vital to be managed by a NC system. Not addressing these threats may nullify the performance gain of coded networks or even worse they can completely disrupt the whole network operation. Security goals and requirements for an NC network are not limited to this short list and there are also other requirements, such as ordering, timestamp, location privacy, and self-organization [36, 101, 102].

4.2.1 Passive Threats and Attacks

Passive threats and attacks do not disturb the normal operation of the network. Malicious nodes that perform a passive attack only snoop (or read) the exchanged information without modifying it. The detection of this type of attacks is difficult because it does not compromise the operation of the network. Passive attacks mainly violate

confidentiality or are used to reveal information on the network topology or capturing sensitive information such as passwords [103].

Eavesdropping

An eavesdropper attacker reads data traffic to obtain sensitive information (e.g., native data, secret keys, and location) about the other nodes. In NC stateless protocols, a malicious intermediate node can act as an eavesdropping attack if it has access to a sufficient number of linearly independent combinations of packets. In this case, the malicious node can easily decode the packets and can have access to all transmitted information.

Figure 4.4 shows several possible scenarios for security attacks in a network coding system such as eavesdropping, selectively dropping, Byzantine modifying, and Pollution attacks (they will be discussed later in this section). Unlike figure 1.4, the time stamp of packets is disregarded in figure 4.4, for simplicity. Suppose that node E in figure 4.4.a only should send packet $a \oplus b$, where $\oplus$ represents the bit by bit XOR of the two packets, and node E is not authorized to have access to the native packets a and b. Here if node E, as an adversary internal eavesdropper, success to overhear at least one of two links AD_1 and BD_2, eventually it can decode XOR packets and obtain fully access to the both native packets a and b. Beside possible internal eavesdroppers, the network could be threatened by an external eavesdropper such as node F in figure 4.4.a; it tries to overhear AD_1 and ED_1 links to successfully decode the XOR packets and obtain unauthorized native packets a and b.

There are several solutions [104-116] for handling an Eavesdropping attack, as a main and the most important passive attack in the NC systems. A solution to the wiretap channels is using *homomorphic hash functions* [117] in which all the intermediate nodes can verify the validity of the encoded packets *on-the-fly* prior to recoding them without knowing the content of native data packets. Also, random linear network coding has been shown to be a cipher, more specifically a Hill cipher, protecting the data if either (i) the attacker does not capture enough coded packets, even if the coding coefficients are sent unprotected, or (ii) if the sender encrypts only the coefficients [118].

Traffic analysis and monitoring

An attacker may monitor and analyse packet transmission in order to extract information about the source and the destination as well as the network topology. Generally, traffic analysis and monitoring threat is due to violating the privacy of nodes by an adversary node. Handling these threats could be more challenging because of intermediate nodes authorization for processing the packets in the NC systems. However, in the other hand, due to the nature of coded networks in using coded packets in intermediate nodes, NC has a potential to thwart these threats if a proper coding mechanism is applied. Both the state-aware and stateless NC protocols can be jeopardized by this threat. There are several works in the literature that have focused on traffic analysis threats and attacks [119-123].

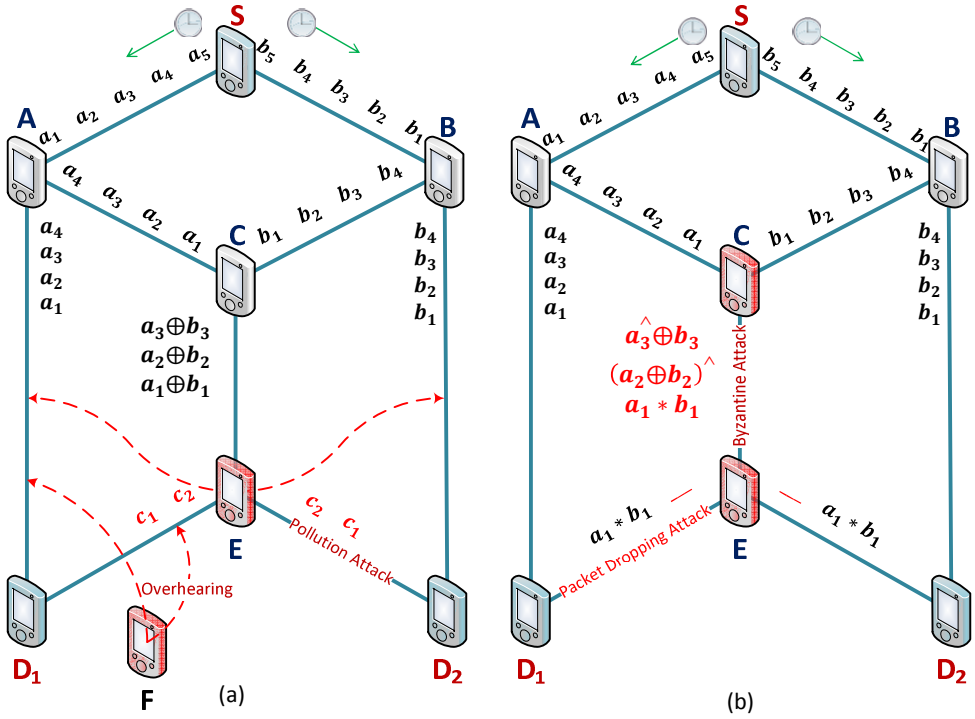

Figure 4.4: (a) node E bogus packets to carry out a pollution attack. Here also node E and F are internal and external eavesdropper respectively that try to overhear some unauthorized packets to obtain required information for decoding XOR messages. (b) Shows the Byzantine modifier node C that modifies received packets and performs invalid operation on them. Also the attacker node E selectively forwards some packets and drops the others.

4.2.2 Active Threats and Attacks

Active attacks, contrary to passive attacks, try to disrupt the normal network operation and may also alter, corrupt, or delete the data packets being exchanged in the network. Active attacks can be caused by an external (outside the network) or an internal (belonging to the network) attacker. As an example of an active attack in NC systems, a malicious node can store and pollute (*corrupt*) data packets by using malware processing functions and then forward (inject) the polluted packets into the network. This attack can extend (advance) rapidly, leading to a network full of *polluted packets* [103].

Denial of Service (DoS)

In NC based systems, when a victim node receives lots of requests (such as packets processing and forwarding) from either reliable neighbors or adversary nodes, that may lead to lack of sufficient resources e.g., bandwidth, CPU power, memory, and battery level in the victim. A malicious node can easily perform a DoS attack by flooding its neighbors injecting lots of junction or corrupted block of coded packets. Therefore, due to the DoS attack, the victim nodes in NC systems should perform frequently the recoding, forwarding, and decoding functions on the corrupted or non-innovative packets: this leads to high traffic and rapid battery exhaustion in victims.

Also, DoS may have an unintentional origin when some particular nodes may become unfairly burdened to support many packets mixing, recoding, and forwarding functions due to their location in the topology of the network. This also can lead to more load these hot spots in terms of radio jamming and energy exhaustion. The DoS attacks can happen in different forms on several layers of the network protocol stack, like jamming and tampering at physical layer, collision, exhaustion and unbalanced loads at link layer, sleep deprivation, blackhole, routing table overflow at network layer, malicious flooding and de-synchronization at transport layer, and finally failure in the remote services, e-banking, and web servers at the application layer [124]. Also, if attackers have enough resources like computing power and bandwidth they can perform a more severe distributed denial of service (DDoS) attack. DoS can have a variety of origins and it affects different layers of network [36].

Due to the role of the intermediate nodes in packet mixing in distributed mechanism of NC systems, DoS is more challenging to handle**.** The mitigation techniques are *statistical analysis* of network traffic for detecting the suspicious nodes that send lots of packets, control messages, and requests to their neighbours, or applying an *authentication and verification* of traffic flows. Also *packet leashes* such as adding a field to the header of the packet in order to limit the maximum allowed hops for the packet can be another possible mitigation technique [125, 126].

Blackhole

A blackhole attacker may exploit the routing protocols to advertise itself as a valid - and usually the shortest- path to a destination [124]. This leads to place itself in the path of data packets toward that destination. Then, the attacker can intercept/eavesdrop the data traffic, or as a blackhole attack can simply deny the routing operations like packet forwarding. In more subtle forms of blackhole attack, called selective forwarding or dropping attack (e.g., node E in figure 4.4.b), the attacker selectively forwards some packets and drops the others. These type of misbehaviours can degrade the performance of the surveillance applications and also violate the privacy of legitimate nodes. Also, refusing to forward the packets and dropping them in the intermediate nodes may result in lack of sufficient coefficients for decoding in sink nodes and render the native packets non-decodable in NC based surveillance applications.

The source nodes in the NC aware systems may generate more coded packets to increase the chance of delivery. The state-aware routing protocol may also find several paths for each source-destination pairs to alleviate disruptive consequences of the selective forwarding and dropping attacks. However, all these mechanisms may lead to more overhead and thus degrade the throughput gain of the NC. Finding an efficient, scalable, and extremely lightweight solution for blackhole attack in the NC aware protocols is an ongoing research.

Byzantine fabrication

A Byzantine fabrication attacker generates messages containing false information. Specially, it can disrupt the routing operation of network in different ways including, but not limited to, forwarding data packets through non-optimal or even invalid routes, generating routing loops, modifying or altering packet headers, routing table overflow, route poisoning, and ACK pollution. Some of the most important Byzantine fabrication attacks include:

i. *Routing table overflow:* Similar to the traditional MANET protocols, NC aware routings can apply proactive (table driven) and reactive (on-demand) routing approaches [6]. A malicious node, especially in a proactive approach, can advertise

lots of new routes to the nonexistent nodes in the networks to overflow other node's routing tables.

ii. *Route poisoning:* Malicious nodes can continuously flood fake or invalid control packets (such as routing requests, replies, errors, and hello packets) into the network to perform a routing table poisoning attack. Also, leading network layer misbehaviors like dropping routing control packets, creating routing loops, extending or shortening service routes, mis-reporting in packet reception, increasing end-to-end delay, and link quality falsification or modification, the attackers can decrease quality of services or worse yet they can lead to network partitioning and DoS attack through mixing all these behaviors [101].

iii. *ACK pollution*: In an NC aware system, for each flow of data packets between a source-sink pair of nodes, the source node continuously flood the coded packets toward the sink node for the current generation of data packets until an acknowledgment (ACK) is received from the sink [127]. The attacker can deceive the source node by creating and injecting fake ACKs or modifying them and leading the source node to move onto the next generation. Therefore, the sink may not receive all the required coded packets in a generation for decoding the whole generation. This leads the source to keep sending coded packets from the current generation forever.

General mitigation techniques can be the *algebraic watchdog*, *packet authentication*, and *cryptography mechanisms* [128].

Byzantine modification and pollution attack

An adversary node may perform invalid coding operations on the transit packets and modify them incorrectly to perform a Byzantine modification attack. Many of the previously discussed attacks like wormhole, blackhole, selective forwarding and dropping attack, and routing attacks can be considered as a special type of Byzantine attacks.

Figure 4.4.b shows a possible scenario for a Byzantine attacker in which the adversary node C, is supposed to perform valid XOR operation on the received packets, create correct packets like $a \oplus b$, and eventually forward them toward downstream nodes. But it wrongly modifies the native packets, modifies the XOR packets, and instead of valid XOR operation, performs another invalid process on them. Figure 4.4.a shows a possible

scenario for packet corruption attack in which the adversary node E, receives some XOR packets like $a \oplus b$ from upstream node C that should be forwarded toward sink nodes D_1 and D_2, but it bogus corrupted packets like c and forwards them toward sink nodes.

Byzantine modification [104, 105, 126, 129-145] and pollution attacks [146-158] are the most important active attacks and probably, beside eavesdropping attacks, are three most studied security attacks in NC systems. Augmenting native data packets via additional or redundant information (e.g., parity checks) enables intermediate and destination nodes to check the validity of packets, drop the corrupted packets, and correct them when it is possible (*error detecting* and *error correcting* mechanisms). Monitoring neighbours via *intrusion detection and prevention mechanisms* are some mitigations techniques for handling these attacks.

Entropy

While eavesdropping or pollution attacks have received a great deal of consideration, there is another attack which has not been studied extensively enough: entropy attack. It happens when an adversary intermediate node generates *valid but non-innovative* packets that are trivial linear combinations of the stale packets, stored or overheard at an earlier time by the attacker. These valid but non-innovative packets decrease the decoding opportunities at sinks, waste network resources, and eventually degrade the overall throughput rate [159-161].

4.3 Current Security Mechanisms Taxonomy

This section presents some mechanisms that can be used by NC to handle some security attacks that have been reviewed in the previous sections.

4.3.1 Security via Network Codes

In a system that errors occur frequently or there are several malicious nodes that inject a large number of polluted packets into the network, the capability of error correction in NC system can be degraded and overwhelmed [127]. Although NC mechanism for using intermediate nodes in packet coding may lead to several security issues; in the other hand, NC scheme itself has a security aware nature that can be useful for handling these attacks.

The inherent security mechanism provided by some NC schemes like RLNC can be easily combined (strengthened) with the application of other security techniques.

For example, in RLNC, by applying a proper coding algorithm on the data packets, we can limit the capability of attackers to perform eavesdropping, since only the destination nodes who have access to sufficient decoding vectors can recover native packets [7, 98]. An adversary intermediate node can perform an eavesdropping attack on the transit packets for extracting some unauthorized information. The problem of eavesdropping attack can be modeled as follows [7, 162]:

i. Suppose Alice is the source node and she wants to send the original coded packet *x* composed of *N* symbols to the sink node.
ii. Bob is the sink node and he receives a coded data packet $\boldsymbol{y}$ composed of *N* symbols.
iii. Calvin, who is the malicious node, eavesdrops the coded packet $\boldsymbol{z}$ composed of *R* symbols.

The coded packets $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ can be represented as $\boldsymbol{x} = (x_1\ x_2\ \ldots\ x_N)$, $\boldsymbol{y} = (y_1\ y_2\ \ldots\ y_N)$, and $\boldsymbol{z} = (z_1\ z_2\ \ldots\ z_R)$.

Several solutions [104-116] have been proposed to handle Eavesdropping attacks in NC systems especially in a wiretap channel [106, 110, 112-116]. NC, itself, due to using packet coding, is one of these solutions [114, 163-166].

Additionally, by using appropriate network codes, it is possible not only to detect corrupted packets, but also to correct them, mitigating a significant part of the most well-known active attacks. However, error detection and error correction (erasure) mechanisms for NC systems, such as [98, 152, 167-173], may lead to some undesired problems. An error detection scheme creates monitoring overhead and error correction is possible only after occurring pollution attacks, which may bring about epidemic disruptive problems for an NC system.

4.3.2 Security via Cooperative Mechanisms

In information theoretic approaches [129, 130], intermediate nodes may simply insert some redundant decoding information into packets, recode packets, and forward them

toward sink nodes. Then, by means of this redundant information, the sink node, the only one being responsible to verify the received packets, can verify the received packets, detect corrupted packets, recover or correct them if possible, and drop the unrecoverable or uncorrectable packets [127]. Therefore, in information theoretic mechanisms, one corrupted packet epidemically may lead to several corrupted packets and can severely degrade the throughput of the network.

Traditional watchdog [174] for MANETs [1] is based on a distributed monitoring, detecting, and isolating malicious nodes. All nodes in the network, by some mechanisms, like overhearing and acknowledgment control packets, are able to monitor their neighbours and detect malicious behaviors in them, like packet dropping, false routing information, and packet modification. After the monitoring and detection phase, nodes will inform each other about these malicious nodes and finally run an isolating phase.

In an NC system, each intermediate node in a data flow, belonging to an arbitrary source-sink communication, may have a simple *store-forward* role, like nodes in traditional MANETs, in which the *non-recoding* intermediate node simply forwards it, without any packet processing. On the other hand, the intermediate node, depending on its position in the network topology or NC mechanism, can also be a *recoding* node that performs *store-code-forward* paradigm: it receives several packets from upstream nodes and, via different paths, mixes and recodes them into one packet, and forward it to downstream nodes.

Like traditional MANET, an upstream node in an NC aware protocol can run Watchdog mechanism and overhear its *non-recoding* downstream neighbours. Hence, the upstream node is able to detect if the *non-recoding* downstream node is forwarding the exact packets properly or it is manipulating the packets and showing misbehaviours. However, traditional Watchdog fails in detecting misbehaviours of *recoding* malicious nodes. When benign upstream nodes forward packets toward a downstream attacker for more recoding operations, it can perform invalid recoding operations. In this scenario, the benign upstream nodes may have no chance to detect these misbehaviors via overhearing, due to the lack of sufficient information for decoding the downstream packets flooded by downstream attackers [127]. As a result, it paves the way for malicious node to creating a variety of misbehaviours.

In 2009, MinJi et. al [175] customized traditional Watchdog for MANETs and proposed the first version of *Algebraic Watchdog* for NC networks. In Algebraic watchdog, nodes can verify their neighbours probabilistically and, by means of overheard messages, can police them locally [128, 142, 175, 176].

4.3.3 Cryptographic and Key Management Based Schemes

The inherent security, provided by some NC protocols, like RLNC, can be easily combined (strengthen) with the application of other security techniques, such as the use of digital signatures, and symmetric or public key encryption. Cryptographic based schemes include a wide range of solutions for pollution attacks. A key management mechanism can be used to exchange shared keys with the sink nodes, which are used for the encryption of the coding coefficients [117, 133, 141, 146, 147, 156, 161, 177-186].

As mentioned before, NC systems are very vulnerable due to the coding role of intermediate nodes. Prolonging the verification process of the encoded packets to the extent that they reach the sink nodes can be potentially, the main reason for intensifying the disruptive epidemic damage of pollution attacks in the NC systems. This problem can be mitigated if intermediate nodes are able to verify polluted coded packets without knowing the native packets. Homomorphic Hash Functions (HFF) [117, 133, 156, 161, 186] have this helpful property for NC systems.

Any hash function like $h(.)$ can map a normally large message like m into a typically small size output $h(m)$. It is computationally very hard to reach m by having $h(m)$, that means, finding m' in a scene that $h(m) = h(m')$ is very difficult. Beside these strong properties, Homomorphic Hash Functions have an additional property called *homomorphism,* in which hash of some native messages (like hash value of a linear combination of some messages in a RLNC system) is equal to combinations of the hashes of those messages [161].

Therefore, because of the *homomorphism* property of these hash functions, all intermediate nodes in the pollution attack model, presented in the previous section, are able to verify the validity of encoded packets *on-the-fly* prior to mixing them algebraically [100]. Now that the intermediate nodes can collaborate in verifying the transit packets, polluted packets will be dropped very soon and malicious nodes can be detected and

isolated. Also, because of *homomorphism* property, the intermediate nodes can combine and encode the incoming hash packets and forward them without knowing the content of native packets or private key of the source node that prevents them from performing an eavesdropping attack. However, this solution has several drawbacks. It is computationally expensive and several works reported the poor performance of the scheme even by applying powerful CPUs [161, 187]. Also, the hash function parameters should be distributed among the network nodes before the communication and, in some implementation, a secure channel is needed for this purpose [188]. Figure 4.5 represents the security taxonomy in network coding systems.

Figure 4.5: Security Taxonomy in Network Coding Systems

4.4 Analysis of RLNC

4.4.1 Error Decoding Probability

There are several cases in which the sink node in the scenario illustrated by figure 4.3 and discussed in Section 4.1.2, may fail to fully recover the native packets even in a network composed of error-free channels and benign nodes. These cases lead to an increase in the *decoding error probability.* Some of these possible cases are:

i. Insufficient receiving codewords: The sink node may not receive g codewords. In this case, the source nodes and the intermediate nodes (as a factory of codewords) may generate more codewords to be sure that the sink node finally receives sufficient innovative codewords. However, by using this mechanism, the redundancy of the code will be increased.

ii. Insufficient rank: The sink node receives more than g codewords but the rank of the received codewords is not sufficient; it means that the receiving codewords are not innovative. This problem also can happen due to erroneous channels in the network. In a network with erasure channels and noisy links, the *decoding error probability* may severely increase. Beside generating more codewords in the source and intermediate nodes, this error can be decreased by using a large enough q for the finite filed size. However, large q will increase the complexity and the cost of encoding, recoding, and decoding operations.

iii. Corrupted symbols: Even when sink node receives sufficient innovative codewords, it may fail to decode correctly due to receiving *polluted* codewords. These codewords have got polluted either by a malicious intermediate node (s) or by noisy links in the network. Several possible mitigation techniques have been proposed in the literature for handling the security attacks in a network coding based network.

In our results in this section, where filed size is $q = 2^m$, the decoding error probability or failure probability of the decoding phase at the sink node is:

$$p_e = \frac{k-k'}{k} \tag{4.10}$$

where the source node floods totally k data symbols into the network per generation however finally only k' of them are decoded successfully at the sink node ($k' \leq k$).

4.4.2 Design Parameters

As discussed in Section 4.1.2, there are several key features for an RLNC based system such as:

i. g : Number of the native data packets in each generation (input native packets)
ii. s : Size of each native data packet (number of the symbols)
iii. n: Number of the output codewords per generation generated by the source node(s)
iv. l: Number of the output codewords per generation generated by the intermediate node(s)
v. q: Size of finite filed $\mathbb{F}_q$ whose binary representation length is m (where $q = 2^m$)

These parameters can influence many properties of the constructed RLNC, including but not limited to: delay, code rate, code redundancy, coefficients overhead, decoding error probability, and security of the code. Also these parameters determine the complexity of encoding operation in source node(s), recoding operations in intermediate node(s), and decoding operations in sink node(s) as shown in Table 4.2 [189]. The number of symbols in data packet will be influenced by the field size. Also, the size of generated codewords in one second in the source node during the encoding phase could be considered as *encoding throughput* of an RLNC system (kB/s). The same concepts, i.e., *recoding throughput* and *decoding throughput*, could be considered for the recoding and decoding phase in the intermediate and sink node as well. All these throughput rates are influenced by both the field size and generation size.

	Encoding	Recoding	Decoding
Complexity	$O(ns)$	$O(ls)$	$O(g^2 + gs)$

Table 4.2: Complexity comparisons of three phases of RLNC in terms of per packet

4.4.3 Simulation Analysis

In this section, we present the simulation results related to the performance of the RLNC mechanism in two different scenarios. In all simulations, each packet size is *1024 bits*.

i. First scenario: network composed of error-free channels and benign nodes

The first scenario is an ideal network consisting of error-free channels and benign nodes. In this case, the results for the line network illustrated in figure 4.3, are categorized into two classes based on the number of intermediate nodes: i) there is no intermediate node (distinguished by the "SD" label in figures) and ii) there are five intermediate nodes (distinguished by "SI5D" label). We set the redundancy (i.e., $r = n - g$ as discussed in Section 4.1.2) to zero for first scenario.

Results regarding encoding and decoding throughput: figure 4.6 shows the encoding and decoding throughput of the RLNC mechanism versus m where the finite filed $\mathbb{F}_q$ size is given by $q = 2^m$. The generation size is 32 packets (i.e., each generation is 32Kb). Increasing m leads to a higher complexity for decoding phase; however, a small m can increase the decoding error probability and decrease the throughput of decoding. Therefore, $m = 8$ could be a trade-off for this case, as illustrated by figure 4.6. Also, the complexity of decoding is higher which leads to lower the throughput of decoding in comparison of encoding. Figure 4.7 shows the encoding and decoding throughput versus generation size where $m = 8$ and $m = 16$. The effect of increasing the generation size on the complexity of decoding phase is higher than the encoding phase as shown in Table 4.2. Therefore decoding throughput is lower than the encoding throughput as illustrated by figure 4.7.

Results regarding decoding error probability: figure 4.8 shows the decoding error probability p_e versus m. The lower p_e could be achieved by a larger filed size, as

illustrated by figure 4.8. However, a large filed leads to a higher complexity, delay, and memory consumption in the encoding, recoding, and decoding phase. Changing the generation size doesn't affect the decoding error probability.

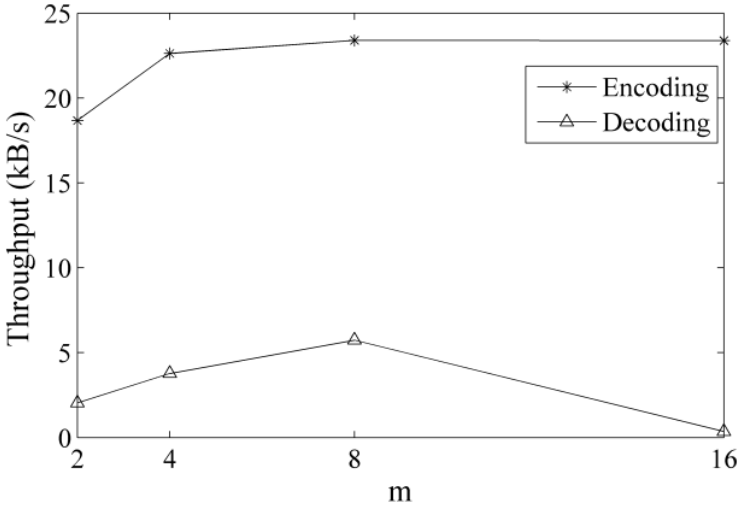

Figure 4.6: Throughput of encoding and decoding versus m (filed size $q = 2^m$)

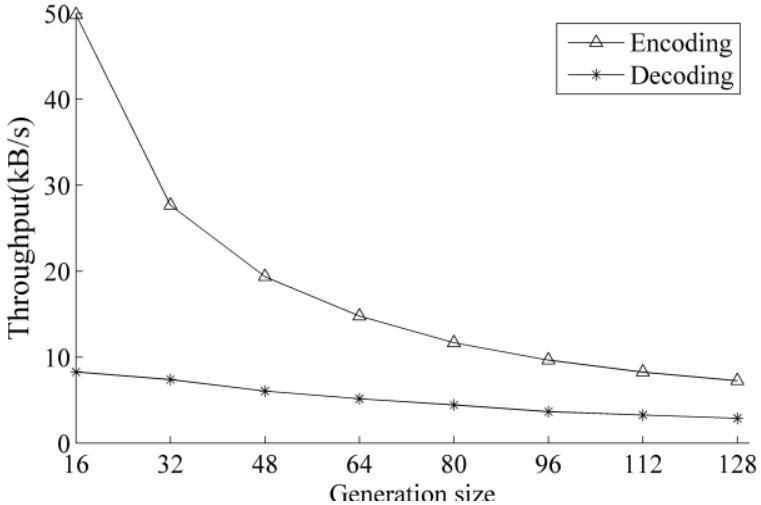

Figure 4.7: Throughput of encoding and decoding versus generation size

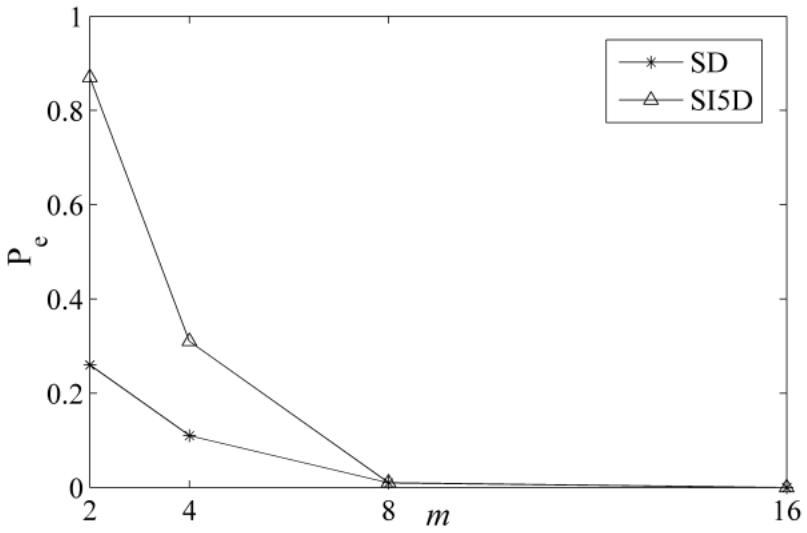

Figure 4.8: Error decoding probability (p_e) versus m

ii. Second scenario: corrupted symbols by erroneous channels or attackers

In the second scenario, the erroneous channel or malicious intermediate node (as an attacker) may corrupt the transitive symbols (codewords) in several forms. Two of which, the most important ones, were investigated:

i. Random corrupted symbols: several distributed random symbols inside each generation will become corrupted by the channel or attacker.

ii. Burst corrupted symbols: a burst of transitive symbols (i.e., several sequential symbols) inside each generation will become corrupted.

The generation size is 32 packets and redundancy is zero for both cases. Figure 4.9 and figure 4.10 show p_e versus the percentage of symbols that randomly and sequentially (as a burst) will be corrupted by the attackers (or channels), respectively. As illustrated by figure 4.9 and figure 4.10, p_e grows very fastly in both mentioned cases.

As the results show, the original mechanism of the RLNC is vulnerable against erroneous channels or intermediate attacker nodes. These type of channels and attackers corrupt the transitive symbols (codewords). To achieve a mitigation technique against packet corruption, in Section 4.5 and Section 4.6, we propose two novel mitigation

techniques for recovering native data packets in a network consisting of erroneous channels and attackers.

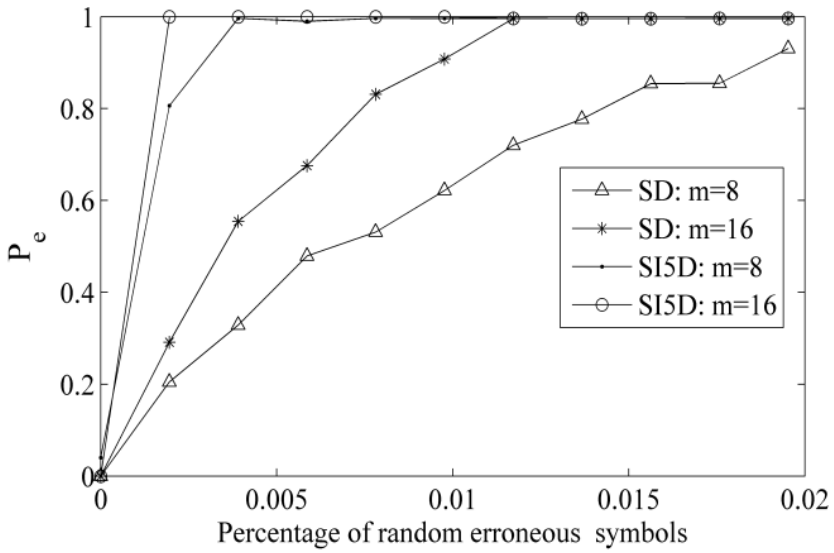

Figure 4.9: p_e versus percentage of random erroneous symbols

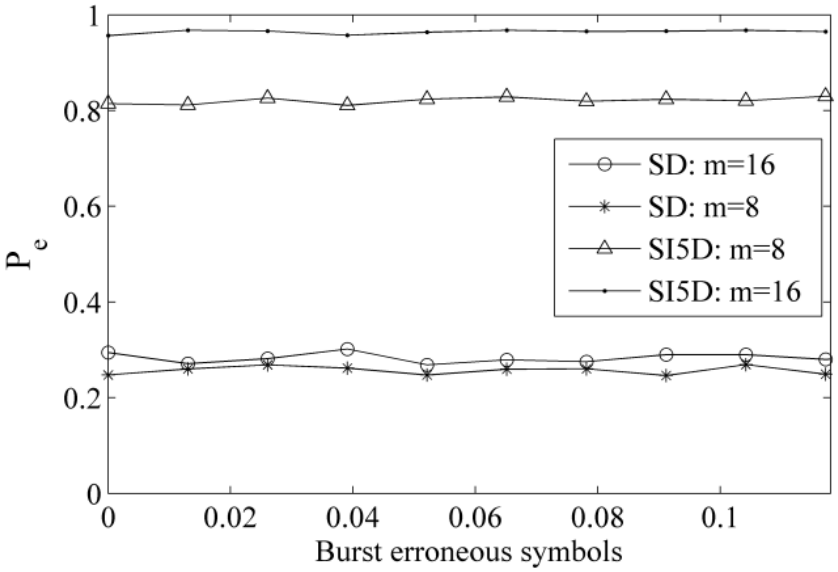

Figure 4.10: p_e versus percentage of burst erroneous symbols

4.5 Secure Network Coding for Eavesdropping

As discussed in Section 4.3.1 ,linear combination of packets and protection of the coefficients from unauthorized access at intermediate nodes are useful for handling security attacks. Therefore one of the existing frameworks for handling security attacks in NC based systems is to modify the three phases of RLNC (i.e. encoding, recoding, and decoding) to achieve a network code that can provide a higher level of security, as illustrated by figure 4.5. In this context, we explored RLNC to handle eavesdropping attacks in NC based systems.

In this section, we present serially concatenated network codes, constituted by the concatenation of LNC and RLNC. The encoding phase of these codes is depicted in figure 4.3. The proposed network coding based concatenated code (NCCC) mechanism, is a mitigation technique both against unauthorized accesses and against eavesdropping attacks due to malicious intermediate nodes.

4.5.1 Three Phases of NCCC

As illustrated by figure 4.11, NCCC mechanism has three phases:

i. Encoding phase

The encoding phase has two steps: i) the source node (e.g., node S in figure 4.3) encodes the native data packets via LNC encoder; and ii) the source node encodes the output of LNC encoder via RLNC and floods the generated encoded packets into the network. These two steps both have two main features:

i. Generation size (g): The number of packets that are linearly combined. The generation size of LNC (first step of encoding) and RLNC (second step of encoding) will be shown by g_1 and g_2, respectively.
ii. The number of bits per symbol (m) that are chosen from a finite filed $\mathbb{F}_q$ by size q (where $q = 2^m$). This value for LNC and RLNC are m_1 and m_2, respectively.

The features of RLNC, i.e., (g_2 , m_2), can be trivial for all intermediate nodes, but the features of LNC, i.e., (g_1 , m_1) remains a secret between the source and the sink nodes and

need to be securely transmitted to the decoder(s). There are several possible solutions for transmitting (g_1, m_1). Some of them can be:

i. The result of the hash function of the two values will be appended to the header of the first packet of each generation.
ii. They can be transmitted via a secure channel.

ii. Recoding phase

An intermediate node (e.g., node I in figure 4.3) *recodes* the incoming encoded packets and forwards them toward the sink node (e.g., node D in figure 4.3). Note that the intermediate node(s) already knows g_2 and m_2 but it does not know g_1 and m_1, which remain secret between the source and the sink.

In a general scenario, (g_1, m_1) and (g_2, m_2) can be the same; however, we choose them differently as a security mechanism of NCCC. Therefore, a malicious unauthorized intermediate node that intends to make an eavesdropping attack, by decoding the incoming coded packets, should examine many possibilities to guess the correct (g_1, m_1) combination. The complexity of this operation is high and, consequently, prepares a level of security for the NCCC against eavesdropping attacks. Moreover, the intermediate nodes can still recode the incoming packets. Nevertheless, since they are unaware of (g_1, m_1), they cannot reach native data packets without paying a cost due to high computational complexity.

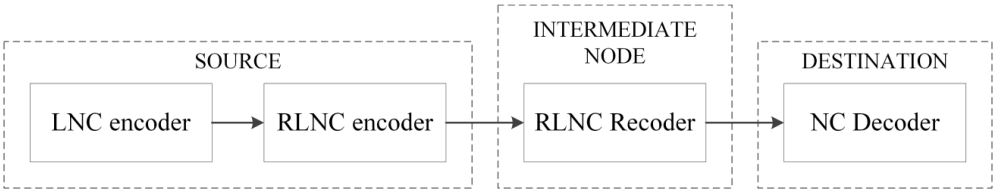

Figure 4.11: A scenario which illustrates the three phases of NCCC mechanism

iii. Decoding phase

The sink node(s) should *decode* the incoming coded packets to extract the native information. Therefore, in NCCC based approach, sink node iteratively decodes the incoming generation, first with RLNC decoder and next with LNC decoder by having (g_2, m_2) and (g_1, m_1), respectively.

4.5.2 Simulation Results

In this section, we present the simulation results related to the performance of NCCC when the network is composed of error-free channels. As mentioned earlier, in NCCC mechanism, the first step of encoding phase via LNC uses m_1 and the second step of encoding phase via RLNC uses m_2 (filed size $q = 2^m$). We examined several (m_1,m_2) combinations which are summarized in Table 4.3. Other simulation features are the same as those in Section 4.4.

case	LNC: m_1	RLNC: m_2
1	2	2
2	2	4
3	**2**	**8**
4	2	16
5	4	2
6	4	4
7	4	8
8	4	16
9	8	2
10	8	4
11	8	8
12	16	2
13	16	4

Table 4.3: different samples of (m_1,m_2)

Figure 4.12 and figure 4.13 show the *decoding error probability* and *encoding and decoding throughput* of NCCC mechanism versus (m_1, m_2), respectively.The simulation results for RLNC, presented in Section 4.4, proved that the minimum error decoding probability, i.e., $p_e \approx 0$, can be achieved for $m \geq 8$. By applying NCCC mechanism, as illustrated by figure 4.13, the minimum error decoding probability, i.e., $p_e \approx 0$, can be achieved for NCCC by choosing ($m_1 = 2$, $m_2 = 8$). Note that $m_1 = 2$, which yields a very small filed size $q = 2^{m_1} = 4$, for LNC has a very low complexity. Also, as illustrated by figure 4.12, using $m_1 = 2$ for LNC encoder, as the first step of encoding phase of NCCC, does not decrease the decoding throughput.

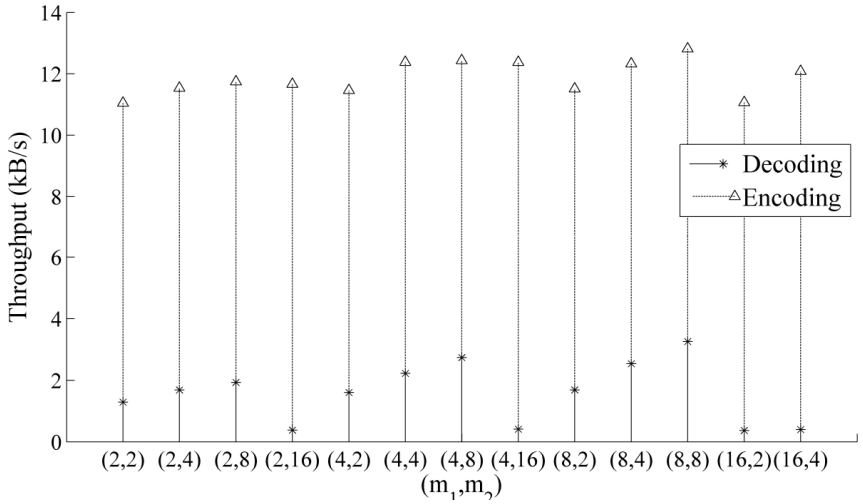

Figure 4.12: Throughput of encoding and decoding versus (m_1,m_2)

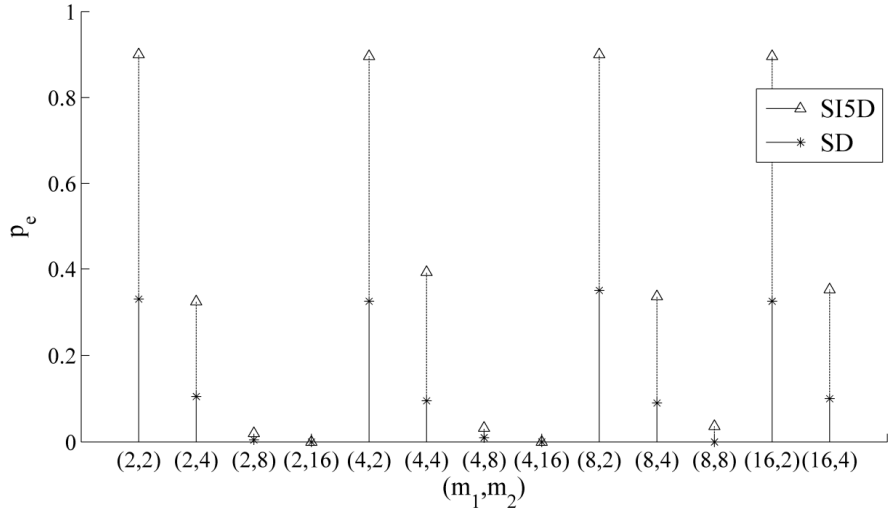

Figure 4.13: Error decoding probability (p_e) versus (m_1, m_2)

4.6 Deploying Pre-coded Network Codes: LT and RLNC

Let us consider an RLNC based line network, in which the source and destination communicate with each other via N intermediate nodes, as illustrated in figure 4.14. RLNC codewords have to pass through channels and relay terminals. However, as shown by the simulation results in Section 4.4, the erroneous channels or byzantine attackers that corrupt the transitive codewords can severely decrease the performance of RLNC.

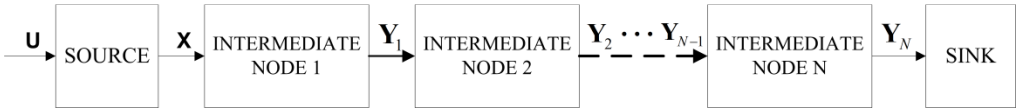

Figure 4.14: A RLNC based line network with N relays

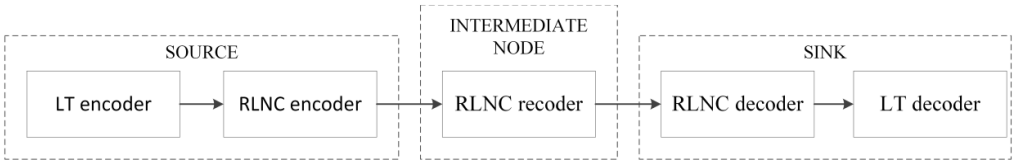

Figure 4.15: BATS mechanism for recovering corrupted packets

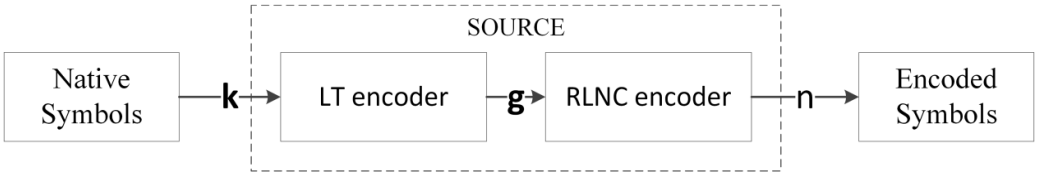

Figure 4.16: The BATS code generate n encoded packets by having *k* native input packets: code rate

A well-known approach for empowering RLNC based system to handle packet corruption is to use *pre-coding* techniques. Current network coding literature particularly focuses on Batched Sparse (BATS) code, a coding scheme that combines RLNC with Luby Transform (LT) codes [25] as a pre-coding scheme. LT codes are rateless error correcting code of the family of fountain codes, which can generate a variable quantity of encoded information according to the needs. Using an LT code as a pre-coding code for RLNC is theoretically described in literature such as [189, 190].

An LT code is a code defined on bipartite graph. A bipartite graph can be algebraically represented by a binary matrix G. In particular, the rows and the columns of this matrix are respectively the nodes on the left and on the right of the graph, and each entry different from zero identify the presence of a connection between two nodes. In the special case of LT codes, the G matrix is randomly defined according to a robust Soliton distribution [25].

However in this thesis, we explore LT as a pre-coding mechanism for RLNC encoder towards security applications specifically against byzantine attackers. Different phases of this technique of concatenating LT codes and RLNC are illustrated in figure 4.15. In this case, we adopt LT for pre-coding to provide a new security framework that proposes an additional degree of protection against third party attackers since malicious users will find it challenging to decode the LT packets since the decoding parameters will remain a secret between the source and sink nodes, whereas in previous applications the encoding parameter of LT approach were easily accessible.

The features of BATS code, e.g., generation size and filed size of RLNC, can be trivial for all the intermediate nodes but G matrix of LT remains as secret information between source and sink nodes. Therefore, the intermediate nodes can still recode the incoming packets but they cannot reach to the native data packets. The source node(s) has several possible solutions for transmitting G matrix to sink node(s). Two of them are:

i. The result of the hash function of G matrix will be appended to the header of the first packet of each generation.

ii. They can be transmitted via a secure channel between source and sink nodes.

The theoretical model that is presented in figure 4.15 has been implemented in MATLAB. The source is generating information composed by k packets of the same size (1600 bits). Since the symbols are over a finite field of size $q = 2^8$, each packet is constituted of 200 symbols. Next, the k packets are encoded into g packets and passed to RLNC encoder which generates n output encoded packets. In order to obtain reliable results, the simulations were performed with the iteration equal to 200.

As we discussed in Section 4.1.2, the code rate shows the performance of the code. As figure 4.16 shows, the code rate of BATS is $\varepsilon = n/k$. By varying code rate, which determines the number of generated output encoding symbols by the source node, the probability of successful decoding will change. Therefore by considering different rates, i.e., $\varepsilon = 2,3,4$, we calculate the error decoding probability p_e of BATS codes.

Assume the scenario network that is illustrated by figure 4.14 without any intermediate node. Figure 4.17 shows the error decoding probability of BATS codes versus

percentage of random corrupted symbols. The X axis shows the percentage of random corrupted symbols by either the channel or attacker nodes. By decreasing the code rate (i.e., by increasing code redundancy), we can decrease the error decoding probability. Also, we assume the scenario network, illustrated by figure 4.14, when there are five intermediate nodes in the network. Figure 4.18 shows the error decoding probability of BATS versus the percentage of random corrupted symbols.

As the simulation results presented in Section 4.4.3 showed, p_e of RLNC, for the case where only less than 0.02 percent of transitive symbols became corrupted, is almost one. However, by using BATS coding scheme we can achieve a much lower p_e.

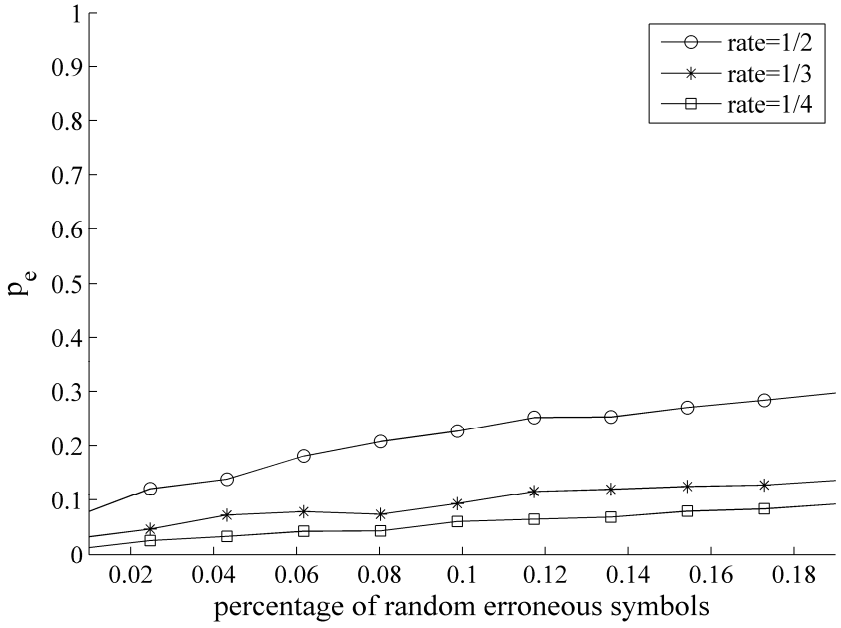

Figure 4.17: p_e versus percentage of random corrupted symbols: there is no intermediate node

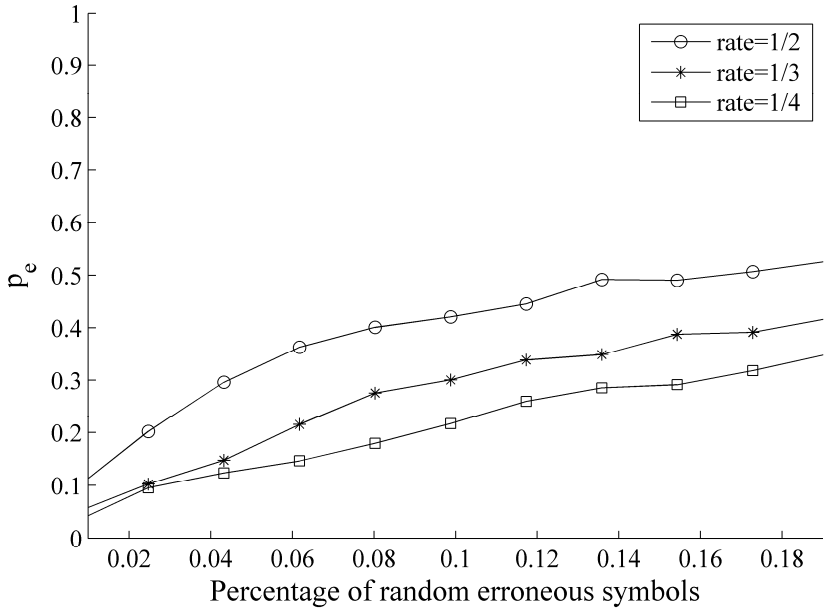

Figure 4.18: p_e versus percentage of random corrupted symbols: there are five intermediate nodes

4.7 Conclusion

NC mechanism can reduce both the number of transmissions and the number of re-transmissions, as well as increase the data transfer rate. These benefits directly translate to energy efficiency. In other hand, the paradigm of RLNC is highly well-matched with the dynamic trend of ad hoc network: any node can join to the network at a given time, start a communication by choosing the coefficients of linear combinations randomly over a finite field, and finally leave the network at any desired time. Also, due to the broadcast nature of the wireless channels in ad hoc networks, the store-code-forward mechanism of NC is very compatible with ad hoc network. However, due to the access of intermediate nodes to the transitive codewords, an attacker can make eavesdropping attacks as well as packet corruption.

As a mitigation technique against eavesdropping attacks, we proposed Network Coding based Concatenated Code (NCCC) that uses Linear Network Coding (LNC) and Random Linear Network Coding (RLNC) in the encoding phase. In NCCC, the intermediate nodes can recode like traditional RLNC. Our results show that although the

complexity of the proposed approach is close to RLNC, it is not computationally feasible for an intermediate attacker node to obtain the native packets. In fact, the required computation for an attacker is so costly in terms of energy consumption, that it will run out of power before completing the decoding process. Also, we applied Luby Transform (LT) code as a pre-coding mechanism for RLNC encoding phase in source node to achieve Batched Sparse (BATS) code. Our results show the high capability of BATS against packet corruption, as illustrated by figure 4.17 and figure 4.18. Original paradigm of RLNC, as simulation results presented in Section 4.4.3 showed, is very vulnerable against packet corruption and less than 0.02 percent of corrupted transitive symbols will lead to almost 100% percent corruption (i.e., decoding error probability or $p_e = 1$) at the sink node. However, using the proposed technique, when symbol error rate is less than 0.02 percent, p_e is decreased to around 0.05.

CHAPTER 5

5 Conclusion and Future Work

In this chapter, we summarize the obtained results and provide some conclusive statements towards the key approaches investigated in this thesis, with some directions for future work.

5.1 Conclusion

The main conclusions of this thesis are summarized as follows:

- Energy efficiency and load balancing are critical design requirement for ad hoc networking technology. Therefore, we proposed a new energy efficient and traffic balancing ad hoc routing protocol based on the well-known IETF AODVv2 protocol in Chapter 3. The new proposed routing protocol, E2AODVv2, achieves high performance with respect to energy consumption, balancing, scalability, network lifetime, and the percentage of failed nodes in comparison to well-known baseline protocols, as illustrated by figure 3.10-13.

- We developed a new load balanced and energy efficient ad hoc routing protocol, called Load Balanced Dynamic Source Routing (LBDSR) in Section 3.2. Having modified the control packets, the routing tables, and the route selection procedure of DSR algorithm, LBDSR achieves higher energy efficiency and better load balancing. LBDSR can also be customized for either energy efficiency or load balancing, leading to a considerable improvement of up to 30% with respect to these metrics. This version is more lightweight and can be customized to be more energy efficient or delay aware.

- The approach for route selection that we developed for E2AODVv2 and LBDSR is technology agnostic. We take into account context parameters such as the energy of routes, length, and traffic to choose the best route. E2AODVv2 and LBDSR showed an improvement of up to 35% and 30% with respect to energy efficiency, respectively. The technology agnostic proposed approach can be applied to any legacy reactive routing protocols. Also it can be customized for energy efficiency or load balancing.

- In wireless networks that take advantage of multi-hop communications, e.g., ad hoc networks, the *store-code-forward* approach of network coding can be an effective tool to improve energy efficiency. In these networks, network coding exploits the broadcast nature of the wireless channel to combine several input packets into several output packets to improve throughput and energy consumption. We proposed Network Coding based Concatenated Code (NCCC) that achieves tighter security against eavesdropping attack, unauthorized access, traffic analysis, and traffic monitoring by malicious intermediate nodes in NC systems. In this novel approach, the intermediate nodes, like traditional RLNC, are still able to recode the received codewords; however, the decoding features will remain a secret between the source and the sink nodes. Therefore, computationally, it is extremely hard for an intermediate node to obtain the native packets.

- We analyzed Random Linear Network Coding (RLNC) and showed that NC is vulnerable to packet corruption due to erroneous channels or the security attacks. As simulation results (presented in Section 4.4.3) showed, even 0.02 percent of corrupted transitive symbols leads to almost 100% percent corruption at the sink node. Therefore, we proposed a mitigation technique for recovering the corrupted native packets by using BATS code. BATS, as an error correction code, uses Luby Transform (LT) codes [25] as a pre-coding technique for NC. However, we explored BATS code for security applications in NC based system. In the novel proposed paradigm, the decoding features of LT will remain a secret between the source and the sink nodes, and we can simultaneously recover the corrupted native packets to provide more reliable network codes.

5.2 Future works

There are number of possibilities for future research. In the following, some research directions are mentioned:

- One of the parameter for determining the quality of a link between two nodes is Signal-to-noise Ratio (SNR) which can determine the required amount of transmits energy per bit. The route priority functions for E2AODVv2 and LBDSR could be modified to take SNR of the links in a route as a metric that determines the priority of a route.

- Until now, we have designed our routing protocols independently of the underlying wireless technology. However, the modulation and access scheme can create additional transmission delay reducing the performance of the overlay routing approaches. Therefore an interlayer collaboration (for example between Data and MAC) may empower the proposed protocols to provide more intelligent routing decisions. The lower layer can inform the upper layer about the quality of the link, thus customize its dynamic route priority function accordingly.

- Wireless Sensor networks (WSNs) can also benefit from ad hoc networking and the proposed ad hoc routing protocols E2AODVv2 and LBDSR. WSN is a key enabling technology for next generation networks, having significant application in connecting a multitude of wireless tiny sensors or being able to operate in a more advanced mode by locally connecting different “things” of different capabilities, such as Personal Device Assistance (PDAs), Mobiles, laptops, and so on. WSNs technology has a profound effect on our everyday life due to its inherent features such as being ubiquitous in nature and offering an inexpensive alternative to traditional networks. However, nodes in WSNs are very limited in terms of battery power, CPU, and memory. Therefore, as a future work, E2AODVv2 and LBDSR can be customized to reflect the requirements of WSNs, as potential applications for NGNs.

- Designing an efficient network code in the presence of all kind of adversaries and erroneous channels in the network still needs further studies. Mixing RLNC and other

error correcting codes is another green research field in the scope of network coding. We proposed two serially concatenated codes, i.e., i) LNC and RLNC and ii) LT and RLNC; however, other error correcting codes, such as Reed-Solomon code, can be also studied for empowering NC mechanism against security attacks and erroneous channels.

Bibliography

[1] MANET, "Working group in IETF, http://datatracker.ietf.org/wg/manet," ed, 2013.

[2] I. T. U. (ITU). Available: http://www.cisco.com/assets/sol/sp/vni/forecast_highlights_mobile/index.html

[3] C. Lei and W. B. Heinzelman, "QoS-aware routing based on bandwidth estimation for mobile ad hoc networks," *Selected Areas in Communications, IEEE Journal on,* vol. 23, pp. 561-572, 2005.

[4] T. A. Ramrekha, V. N. Talooki, J. Rodriguez, and C. Politis, "Energy Efficient and Scalable Routing Protocol for Extreme Emergency Ad Hoc Communications," *Mob. Netw. Appl.,* vol. 17, pp. 312-324, 2012.

[5] R. Ahlswede, C. Ning, S. Y. R. Li, and R. W. Yeung, "Network information flow," *Information Theory, IEEE Transactions on,* vol. 46, pp. 1204-1216, 2000.

[6] M. A. Iqbal, B. Dai, B. Huang, A. Hassan, and S. Yu, "Survey of network coding-aware routing protocols in wireless networks," *Journal of Network and Computer Applications,* vol. 34, pp. 1956-1970, 2011.

[7] C. Fragouli and E. Soljanin, "Network coding applications," *found and trends netw,* vol. 2, pp. 135-269, 2007.

[8] W. Weichao, P. Di, and A. Wyglinski, "Detecting Sybil nodes in wireless networks with physical layer network coding," in *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*, 2010, pp. 21-30.

[9] Z. Li, D. Pu, W. Wang, and A. Wyglinski, "Forced collision: Detecting wormhole attacks with physical layer network coding," *Tsinghua Science and Technology,* vol. 16, pp. 505-519, 2011.

[10] D. Zhiguo, I. Krikidis, J. Thompson, and K. K. Leung, "Physical Layer Network Coding and Precoding for the Two-Way Relay Channel in Cellular Systems," *Signal Processing, IEEE Transactions on,* vol. 59, pp. 696-712, 2011.

[11] M. Bloch and J. Barros, *Physical-Layer Security: From Information Theory to Security Engineering*: Cambridge University Press, 2011.

[12] B. Nazer and M. Gastpar, "Reliable Physical Layer Network Coding," *Proceedings of the IEEE,* vol. 99, pp. 438-460, 2011.

[13] J. K. Sundararajan, D. Shah, M. Médard, S. Jakubczak, M. Mitzenmacher, and J. Barros, "Network Coding Meets TCP: Theory and Implementation," *Proceedings of the IEEE,* vol. 99, pp. 490 – 512, 2011.

[14] L. Scalia, F. Soldo, and M. Gerla, "PiggyCode: A MAC Layer Network Coding Scheme to Improve TCP Performance Over Wireless Networks," in *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, 2007, pp. 3672-3677.

[15] S. Katti, D. Katabi, W. Hu, H. S. Rahul, and M. M´edard, "The importance of being opportunistic: Practical network coding for wireless environments," presented at the In Proc. 43rd Annual Allerton Conf. Commun. Control Comput., 2006.

[16] M. Xufei, T. Shaojie, X. Xiahua, L. Xiang-Yang, and M. Huadong, "Energy-Efficient Opportunistic Routing in Wireless Sensor Networks," *Parallel and Distributed Systems, IEEE Transactions on,* vol. 22, pp. 1934-1942, 2011.

[17] W. Suyu, G. Xuejuan, and Z. Li, "Survey of network coding and its benefits in energy saving over wireless sensor networks," in *Information, Communications and Signal Processing, 2009. ICICS 2009. 7th International Conference on*, 2009, pp. 1-5.

[18] K. Sukwon, H. Tracey, and M. Effros, "Network coding with periodic recomputation for minimum energy multicasting in mobile ad-hoc networks," in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, 2008, pp. 154-161.

[19] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Comput. Netw.,* vol. 47, pp. 445-487, 2005.

[20] I. Curran and S. Pluta, "Overview of machine to machine and telematics," in *Water Event, 2008 6th Institution of Engineering and Technology*, 2008, pp. 1-33.

[21] IETF. *The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4*. Available: http://tools.ietf.org/html/rfc4728

[22] L. Lima, S. Gheorghiu, J. Barros, M. Médard, and A. L. Toledo, "Secure Network Coding for Multi-Resolution Wireless Video Streaming," *Jou. of Sel. Areas in Comm.,* vol. 28, pp. 377-388, 2010.

[23] Y. Fan, *Network Coding based Information Security in Multi-hop Wireless Networks, Ph.D Thesis, page 24*. Ontario, Canad: Electrical and Computer Engineering, University of Waterloo, 2010.

[24] C. Fragouli and J. W. Jean-Yves Le Boudec, "Network coding: an instant primer," *ACM SIGCOMM Computer Communication,* vol. 36, pp. 63 - 68, 2006.

[25] M. Luby, "LT codes," in *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, 2002, pp. 271-280.

[26] C. E. Perkins and P. Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers," *SIGCOMM Comput. Commun. Rev.,* vol. 24, pp. 234-244, 1994.

[27] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: the dynamic source routing protocol for multihop wireless ad hoc networks," in *Ad hoc networking*, ed: Addison-Wesley Longman Publishing Co., Inc., 2001, pp. 139-172.

[28] A. Laouiti, P. Jacquet, P. Minet, L. Viennot, T. Clausen, and C. Adjih, *Optimized Link State Routing Protocol (OLSR)*, 2003.

[29] A. Iwata, C. Ching-Chuan, P. Guangyu, M. Gerla, and C. Tsu-Wei, "Scalable routing strategies for ad hoc wireless networks," *Selected Areas in Communications, IEEE Journal on,* vol. 17, pp. 1369-1379, 1999.

[30] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A distance routing effect algorithm for mobility (DREAM)," presented at the Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking, Dallas, Texas, USA, 1998.

[31] V. D. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," presented at the Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution, 1997.

[32] P. Guangyu, M. Gerla, and C. Tsu-Wei, "Fisheye state routing: a routing scheme for ad hoc wireless networks," in *Communications, 2000. ICC 2000. 2000 IEEE International Conference on*, 2000, pp. 70-74 vol.1.

[33] S. Murthy and J. J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," *Mob. Netw. Appl.,* vol. 1, pp. 183-197, 1996.

[34] E. Alotaibi and B. Mukherjee, "A survey on routing algorithms for wireless Ad-Hoc and mesh networks," *presented at Computer Networks,* pp. 940-965, 2012.

[35] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," presented at

the Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking, Dallas, Texas, USA, 1998.

[36] M. e. Illyas, *The handbook of ad hoc wireless networks, chapter 30, security in wireless ad hoc networks*: CRC press Boca Raton, 2003.

[37] V. Garousi, " Simulating Network traffic in Multi-hop Wireless Ad Hoc Networks based on DSDV protocol using NS (Network Simulator) Package," University of Waterloo, Fall 2001.

[38] V. Talooki and J. Rodriguez, " Quality of Service for Flat Routing Protocols in Mobile Ad hoc Networks

" in *5th International Mobile Multimedia Communications Conference,*, London, UK, 7-9th of September 2009.

[39] V. N. Talooki and J. Rodriguez, "Jitter based comparisons for routing protocols in mobile ad hoc networks," in *Ultra Modern Telecommunications & Workshops, 2009. ICUMT '09. International Conference on*, 2009, pp. 1-6.

[40] PEACE, "iP-based Emergency Application and serviCes for nExt generation networks (PEACE)," project that is supported by the European Commission in the framework of FP7 ICT-SEC-2007 with contract number 225654.2008.

[41] A. Argyriou and V. Madisetti, "Realizing load-balancing in ad-hoc networks with a transport layer protocol," in *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, 2004, pp. 1897-1902 Vol.3.

[42] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, 1999, pp. 90-100.

[43] V. N. Talooki, J. Rodriguez, and R. Sadeghi, "A load balanced aware routing protocol for wireless ad hoc networks," in *Telecommunications, 2009. ICT '09. International Conference on*, 2009, pp. 25-30.

[44] S. R. Das, C. E. Perkins, and E. M. Royer, "Performance comparison of two on-demand routing protocols for ad hoc networks," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2000, pp. 3-12 vol.1.

[45] Z. J. Haas and M. Pearlman, " The Zone Routing Protocol (ZRP) for Ad-Hoc networks, IETF Internet draft," July 2002.

[46] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," presented at the Proceedings of the 6th annual international conference on Mobile computing and networking, Boston, Massachusetts, USA, 2000.

[47] E. M. Royer and T. Chai-Keong, "A review of current routing protocols for ad hoc mobile wireless networks," *Personal Communications, IEEE,* vol. 6, pp. 46-55, 1999.

[48] E. M. Royer and C. E. Perkins, "An implementation study of the AODV routing protocol," in *Wireless Communications and Networking Confernce, 2000. WCNC. 2000 IEEE*, 2000, pp. 1003-1008 vol.3.

[49] H. M. D. Sun, "“TCP Flow-based Performance Analysis of Two On-Demand Routing Protocols for Mobile Ad Hoc Networks”," presented at the In Vehicular Technology Conference VTC, IEEE VTS 54th, 2001, Fall.

[50] A. K. B. R. L. C. R. P. S. Hiremath, "Performance Comparison of Wireless Mobile Ad-Hoc Network Routing Protocols," presented at the IJCSNS International Journal of Computer Science and Network Security, 2008.

[51] Karavetsios and Economides, "Performance comparison of distributed routing algorithms in ad hoc mobile networks," *WSEAS Transactions on Communications,* vol. 3, pp. 317-321, 2004.
[52] C. K. Toh, *Ad Hoc Mobile Wireless Networks: Protocols and Systems*: Prentice Hall, 2002.
[53] D. Johnson and D. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Computing*. vol. 353, T. Imielinski and H. Korth, Eds., ed: Springer US, 1996, pp. 153-181.
[54] n. The Network Simulator, http://www.isi.edu/nsnam/ns/.
[55] http://www.monarch.cs.cmu.edu. (2009). *The cmu monarch project*.
[56] S. N. t. i. M.-h. W. A. H. N. b. o. D. p. u. N. N. S. P. V. Garousi, University of Waterloo, Fall 2001.
[57] J. Gomez, A. Campbell, M. Naghshineh, and C. Bisdikian, "PARO: Supporting Dynamic Power Controlled Routing in Wireless Ad Hoc Networks," *Wireless Networks,* vol. 9, pp. 443-460, 2003/09/01 2003.
[58] A. Lindgren and O. Schelen, "Infrastructured Ad-Hoc networks," in *International Conference on Parallel Processing (International Workshop on Ad-Hoc Networking (IWAHN 2002))*, 2002, pp. 64-70.
[59] J. J. Edwards, J. David Brown, and P. C. Mason, "Using covert timing channels for attack detection in MANETs," in *MILITARY COMMUNICATIONS CONFERENCE, 2012 - MILCOM 2012*, 2012, pp. 1-7.
[60] D. Djenouri and N. Badache, "Dynamic source routing power-aware," *International Journal of Ad-Hoc and Ubiquitous Computing (IJAHUC'06),* vol. 1, pp. 126–136, 2006.
[61] G. Chakrabarti and S. Kulkarni, "Load balancing and resource reservation in mobile ad hoc networks," *Ad Hoc Netw.,* vol. 4, pp. 186-203, 2006.
[62] S. Singh, M. Woo, and C. S. Raghavendra, "Power-aware routing in mobile ad hoc networks," presented at the Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking, Dallas, Texas, USA, 1998.
[63] C. Jae-Hwan and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2000, pp. 22-31 vol.1.
[64] A. Zhou and H. Hassanein, "Load-Balanced Wireless Ad Hoc Routing," in *Proceedings of Canadian Conference on Electrical and Computer Engineering*, pp. 1157– 1161.
[65] IETF. (2013). *Dynamic MANET On-demand (AODVv2) Routing, draft-ietf-manet-dymo-26*. Available: http://tools.ietf.org/html/draft-ietf-manet-dymo-26
[66] S. K. Bisoyi and S. Sahu, "Performance analysis of Dynamic MANET Ondemand (DYMO) Routing protocol," *Special Issue of IJCCT* vol. 1, 2012.
[67] IETF. *Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format*. Available: http://tools.ietf.org/html/rfc5444
[68] IETF. *Internet Protocol (IP), RFC 791*. Available: http://tools.ietf.org/html/rfc791
[69] G. Koltsidas, F. N. Pavlidou, K. Kuladinithi, A. Timm-Giel, and C. Gorg, "Investigating the Performance of a Multipath DYMO Protocol for Ad-Hoc Networks," in *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, 2007, pp. 1-5.

[70] D. Singh, A. K. Maurya, and A. K. Sarje, "Comparative performance analysis of LANMAR, LAR1, DYMO and ZRP routing protocols in MANET using Random Waypoint Mobility Model," in *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*, 2011, pp. 62-66.

[71] C. Kretschmer, S. Ruhrup, and C. Schindelhauer, "DT-DYMO: Delay-Tolerant Dynamic MANET On-demand Routing," in *Distributed Computing Systems Workshops, 2009. ICDCS Workshops '09. 29th IEEE International Conference on*, 2009, pp. 493-498.

[72] N. Anastacio, F. Merca, O. Cabral, and F. J. Velez, "QoS Metrics for Cross-Layer Design and Network Planning for B3G Systems," in *Wireless Communication Systems, 2006. ISWCS '06. 3rd International Symposium on*, 2006, pp. 592-596.

[73] S. Y. R. Li, R. W. Yeung, and C. Ning, "Linear network coding," *Information Theory, IEEE Transactions on,* vol. 49, pp. 371-381, 2003.

[74] R. Koetter and M. Medard, "An algebraic approach to network coding," *Networking, IEEE/ACM Transactions on,* vol. 11, pp. 782-795, 2003.

[75] T. HO, M. Medard, R. Koetter, D. Karger, M. Effros, S. Jun, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory 52,* pp. 4413–4430, 2004.

[76] D. E. Lucani, M. Medard, and M. Stojanovic, "On Coding for Delay - Network Coding for Time Division Duplexing," *Information Theory, IEEE Transactions on,* vol. 58, pp. 2330-2348, 2012.

[77] H. M. Riccardo Bassoli, Jonathan Rodriguez, Kenneth W. Shum and Rahim Tafazolli, "Network Coding Theory: A Survey," *ieee,* 2013.

[78] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," *SIGCOMM Comput. Commun. Rev.,* vol. 37, pp. 169-180, 2007.

[79] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard, "Symbol-level network coding for wireless mesh networks," presented at the Proceedings of the ACM SIGCOMM 2008 conference on Data communication, Seattle, WA, USA, 2008.

[80] H. Tracey and H. Viswanathan, "Dynamic Algorithms for Multicast With Intra-Session Network Coding," *IEEE transactions on INFORMATION FORENSICS AND SECURITY,* vol. 55, pp. 797-815, 2009.

[81] B. Radunovic, C. Gkantsidis, P. Key, and P. Rodriguez, "An Optimization Framework for Opportunistic Multipath Routing in Wireless Mesh Networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, 2008, pp. 2252-2260.

[82] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: practical wireless network coding," *SIGCOMM Comput. Commun. Rev.,* vol. 36, pp. 243-254, 2006.

[83] B. Scheuermann, W. Hu, and J. Crowcroft, "Near-optimal co-ordinated coding in wireless multihop networks," presented at the Proceedings of the 2007 ACM CoNEXT conference, New York, New York, 2007.

[84] S. SENGUPTA, S. RAYANCHU, and BANERJEE, "An Analysis of Wireless Network Coding for Unicast Sessions: The Case for Coding-Aware Routing," in *In Proceedings of the 26th Annual IEEE Conference on Computer Communications (INFOCOM 2007)*, 2007.

[85] E. Rozner, A. P. Iyer, Y. Mehta, L. Qiu, and M. Jafry, "ER: efficient retransmission scheme for wireless LANs," presented at the Proceedings of the 2007 ACM CoNEXT conference, New York, New York, 2007.

[86] P. Chaporkar and A. Proutiere, "Adaptive network coding and scheduling for maximizing throughput in wireless networks," presented at the Proceedings of the 13th annual ACM international conference on Mobile computing and networking, Montréal, Québec, Canada, 2007.

[87] B. Ni, N. Santhapuri, Z. Zhong, and S. Nelakuditi, "Routing with opportunistically coded exchanges in wireless mesh networks," in *Proceedings of the 2nd IEEE Workshop on Wireless Mesh Networks (WiMesh 2006)*, 2006, pp. 157-159.

[88] W. Xin, Z. Li, X. Ji, and W. Qingyun, "Network Coding Aware Routing Protocol for Lossy Wireless Networks," in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, 2009, pp. 1-4.

[89] Y. Yan, Z. Zhuang, Z. Baoxian, H. T. Mouftah, and M. Jian, "Rate-Adaptive Coding-Aware Multiple Path Routing for Wireless Mesh Networks," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, 2008, pp. 1-5.

[90] J.-z. Sun, Y.-a. Liu, H.-f. Hu, and D.-m. Yuan, "On-demand coding-aware routing in wireless Mesh networks," *The Journal of China Universities of Posts and Telecommunications,* vol. 17, pp. 80-92, 2010.

[91] W. Yunnan, S. M. Das, and R. Chandra, "Routing with a Markovian Metric to Promote Local Mixing," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, 2007, pp. 2381-2385.

[92] Y. Yan, Z. Baoxian, H. T. Mouftah, and M. Jian, "Practical Coding-Aware Mechanism for Opportunistic Routing in Wireless Mesh Networks," in *Communications, 2008. ICC '08. IEEE International Conference on*, 2008, pp. 2871-2876.

[93] L. Jilin, J. C. S. Lui, and C. Dah-Ming, "DCAR: Distributed Coding-Aware Routing in Wireless Networks," *Mobile Computing, IEEE Transactions on,* vol. 9, pp. 596-608, 2010.

[94] Y. Lu, C. Shen, Q. Xia, and J. Tao, "ICM: A Novel Coding-Aware Metric for Multi-Hop Wireless Routing," in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, 2009, pp. 1-4.

[95] G. Hui, Q. Yi, L. Kejie, and N. Moayeri, "Backbone Routing over Multihop Wireless Networks: Increased Network Coding Opportunity," in *IEEE International Conference on Communications (ICC), 2010*, 2010, pp. 1-5.

[96] J. Xianlong, W. Xiaodong, and Z. Xingming, "Active Network Coding Based High-Throughput Optimizing Routing for Wireless Ad Hoc Networks," in *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, 2008, pp. 1-5.

[97] H. Song, Z. Zifei, L. Hongxing, C. Guihai, E. Chan, and A. K. Mok, "Coding-Aware Multi-path Routing in Multi-Hop Wireless Networks," in *Performance, Computing and Communications Conference, 2008. IPCCC 2008. IEEE International*, 2008, pp. 93-100.

[98] R. Koetter and F. R. Kschischang, "Coding for Errors and Erasures in Random Network Coding," *IEEE Transactions on Information Theory,* vol. 54, pp. 3579-3591, 2008.
[99] L. Lima, J. P. Vilela, P. F. Oliveira, and J. Barros, "Network Coding Security: Attacks and Countermeasures," *IEEE,* 2008.
[100] L. Lima, "Network Coding Security: Algebraic Properties and Lightweight Solutions," PhD, Faculty of Science, Universidade of Porto, 2010.
[101] G. Padmavathi and S. D., "A Survey of Attacks, Security Mechanisms and Challenges in Wireless Sensor Networks," *(IJCSIS) International Journal of Computer Science and Information Security,* vol. 4, 2009.
[102] N. Cai and T. Chan, "Theory of secure network coding," *IEEE,* vol. 99, 2011.
[103] J. Barros, "Network Coding Security, Seminar on Security in the Information Society," University of Porto, Department of Computer Science, Porto2008.
[104] S. Jaggi and M. Langberg, "Resilient network codes in the presence of eavesdropping Byzantine adversaries," in *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, 2007, pp. 541-545.
[105] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Medard, and M. Effros, "Resilient Network Coding in the Presence of Byzantine Adversaries," *Information Theory, IEEE Transactions on,* vol. 54, pp. 2596-2603, 2008.
[106] A. R. Hammons, Z. Qinqing, and B. Haberman, "On the Eavesdrop Vulnerability of Random Network Coding over Wireless Networks," in *Distributed Computing Systems Workshops, 2009. ICDCS Workshops '09. 29th IEEE International Conference on*, 2009, pp. 201-207.
[107] Y. Hongyi, D. Silva, S. Jaggi, and M. Langberg, "Network Codes Resilient to Jamming and Eavesdropping," in *Network Coding (NetCod), 2010 IEEE International Symposium on*, 2010, pp. 1-6.
[108] A. R. Hammons Jr, Q. Zhang, and B. Haberman, "An Eavesdrop Vulnerability Analysis of Random Network Coding over Wireless Ad-Hoc Networks," in *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*, 2010, pp. 1-6.
[109] G. Zhenzhen, Y. Yu-Han, and K. J. R. Liu, "Anti-Eavesdropping Space-Time Network Coding for Cooperative Communications," *Wireless Communications, IEEE Transactions on,* vol. 10, pp. 3898-3908, 2011.
[110] N. Cai and R. W. Yeung, "Secure network coding," in *Information Theory, 2002. Proceedings. 2002 IEEE International Symposium on*, 2002, p. 323.
[111] K. Jain, "Security based on network topology against the wiretapping attack," *Wireless Communications, IEEE,* vol. 11, pp. 68-71, 2004.
[112] C. Ning and R. W. Yeung, "Secure Network Coding on a Wiretap Network," *Information Theory, IEEE Transactions on,* vol. 57, pp. 424-435, 2011.
[113] N. Cai and R. W. Yeung, "Secure Network Coding on a Wiretap Network," *IEEE TRANSACTIONS ON INFORMATION THEORY,* vol. 57, 2011.
[114] N. Cai and R. W. Yeung, "Secure Network Coding on a Wiretap," *IEEE TRANSACTIONS ON INFORMATION THEORY 1,* 2002.
[115] S. El Rouayheb, E. Soljanin, and A. Sprintson, "Secure Network Coding for Wiretap Networks of Type II," *Information Theory, IEEE Transactions on,* vol. 58, pp. 1361-1371, 2012.
[116] C. Xiangmao, W. Jin, W. Jianping, V. Lee, L. Kejie, and Y. Yixian, "On Achieving Maximum Secure Throughput Using Network Coding against Wiretap Attack," in

Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on, 2010, pp. 526-535.

[117] M. N. Krohn, M. J. Freedman, and D. Mazieres, "On-the-fly verification of rateless erasure codes for efficient content distribution," in *IEEE Symp. on Security and Privacy, pp. 226-240, 2004*, 2004, pp. 226-240.

[118] C. Ning and T. Chan, "Theory of Secure Network Coding," *Proceedings of the IEEE,* vol. 99, pp. 421-437, 2011.

[119] F. Yanfei, J. Yixin, Z. Haojin, C. Jiming, and X. S. Shen, "Network Coding Based Privacy Preservation against Traffic Analysis in Multi-Hop Wireless Networks," *Wireless Communications, IEEE Transactions on,* vol. 10, pp. 834-843, 2011.

[120] L. Buttyan and T. Holczer, "Traffic analysis attacks and countermeasures in wireless body area sensor networks," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a*, 2012, pp. 1-6.

[121] W. Zhiguo, X. Kai, and L. Yunhao, "Priv-Code: Preserving privacy against traffic analysis through network coding for multihop wireless networks," in *INFOCOM, 2012 Proceedings IEEE*, 2012, pp. 73-81.

[122] F. Yanfei, J. Yixin, Z. Haojin, and S. Xuemin, "An Efficient Privacy-Preserving Scheme against Traffic Analysis Attacks in Network Coding," in *INFOCOM 2009, IEEE*, 2009, pp. 2213-2221.

[123] H. Sousa-Pinto, D. E. Lucani, and J. Barros, "Hide and code: Session anonymity in wireless line networks with coded packets," in *Information Theory and Applications Workshop (ITA), 2012*, 2012, pp. 262-268.

[124] P. Jawandhiya, M. Ghonge, M. S. Ali, and J. S. Deshpande, "A survey of Mobile ad hoc network attacks," *Internationnal Journal of Engineering science and Technology,* vol. 2, pp. 4036-4071, 2010.

[125] R. H. Jhaveri, S. J. Patel, and D. C. Jinwala, "DoS Attacks in Mobile Ad Hoc Networks: A Survey," in *Second International Conference on Advanced Computing & Communication Technologies (ACCT)*, 2012, pp. 535-541.

[126] L. Lima, J. Barros, and R. Koetter, "Byzantine attacks against network coding in peer to peer distributed storage," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, 2009, pp. 1164-1168.

[127] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Secure network coding for wireless mesh networks: Threats, challenges, and directions," *Comput. Commun.,* vol. 32, pp. 1790-1801, 2009.

[128] M. Kim, M. Medard, and J. Barros, "Algebraic Watchdog: Mitigating Misbehavior in Wireless Network Coding," *Jou. of Sel. Areas in Comm.,* vol. 29, pp. 1916-1925, 2011.

[129] T. Ho, B. Leong, R. Koetter, M. Medard, M. Effros, and D. R. Karger, "Byzantine modification detection in multicast networks using randomized network coding," in *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*, 2004, p. 144.

[130] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Médard, and M. Effros, "Resilient Network Coding in the Presence of Byzantine Adversaries," *IEEE TRANSACTIONS ON INFORMATION THEORY,* vol. 54, 2008.

[131] B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru, and H. Rubens, "ODSBR: An on-demand secure Byzantine resilient routing protocol for wireless ad hoc networks," *ACM Trans. Inf. Syst. Secur.,* vol. 10, pp. 1-35, 2008.

[132] M. J. Siavoshani, C. Fragouli, and S. Diggavi, "On Locating Byzantine Attackers," in *Network Coding, Theory and Applications, 2008. NetCod 2008. Fourth Workshop on*, 2008, pp. 1-6.

[133] T. Ho, L. Ben, R. Koetter, M. Medard, M. Effros, and D. R. Karger, "Byzantine Modification Detection in Multicast Networks With Random Network Coding," *Information Theory, IEEE Transactions on,* vol. 54, pp. 2798-2803, 2008.

[134] L. Nutman and M. Langberg, "Adversarial models and resilient schemes for network coding," in *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, 2008, pp. 171-175.

[135] G. Sharma, S. Jaggi, and B. K. Dey, "Network tomography via network coding," in *Information Theory and Applications Workshop, 2008*, 2008, pp. 151-157.

[136] M. Kim, M. Medard, and J. Barros, "Counteracting Byzantine adversaries with network coding: An overhead analysis," in *Military Communications Conference, 2008. MILCOM 2008. IEEE*, 2008, pp. 1-7.

[137] T. Ho, L. Ben, K. Ralf, M. Médard, Michelle, Effros., and K. David, "Byzantine Modification Detection in Multicast Networks With Random Network Coding," *IEEE Transactions on Information Theory,* vol. 54, pp. 2798-2803, 2008.

[138] T. Ho, L. Ben, K. Ralf, Médard, and Karger, "Byzantine Modification Detection in Multicast Networks With Random Network Coding," *IEEE Transactions on Information Theory,* vol. 54, pp. 2798-2803, 2008.

[139] R. Curtmola and C. Nita-Rotaru, "BSMR: Byzantine-Resilient Secure Multicast Routing in Multihop Wireless Networks," *IEEE Transactions on Mobile Computing,* vol. 8, pp. 445-459, 2009.

[140] O. Kosut, T. Lang, and D. Tse, "Nonlinear network coding is necessary to combat general Byzantine attacks," in *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*, 2009, pp. 593-599.

[141] M. Kim, L. Lima, Z. Fang, J. Barros, M. Medard, R. Koetter, T. Kalker, and K. J. Han, "On counteracting Byzantine attacks in network coded peer-to-peer networks," *Selected Areas in Communications, IEEE Journal on,* vol. 28, pp. 692-702, 2010.

[142] M. Kim, Me, x, M. dard, and J. Barros, "A multi-hop multi-source Algebraic Watchdog," in *Information Theory Workshop (ITW), 2010 IEEE*, 2010, pp. 1-5.

[143] L. Anh and A. Markopoulou, "Locating Byzantine Attackers in Intra-Session Network Coding Using SpaceMac," in *Network Coding (NetCod), 2010 IEEE International Symposium on*, 2010, pp. 1-6.

[144] K. Sukwon, T. Ho, M. Effros, and A. S. Avestimehr, "Network Error Correction With Unequal Link Capacities," *Information Theory, IEEE Transactions on,* vol. 57, pp. 1144-1164, 2011.

[145] M. J. Siavoshani, C. Fragouli, and S. N. Diggavi, "Subspace Properties of Network Coding and Their Applications," *Information Theory, IEEE Transactions on,* vol. 58, pp. 2599-2619, 2012.

[146] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An Efficient Signature-based Scheme for Securing Network Coding against Pollution Attacks," in *IEEE INFOCOM*, 2008.

[147] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Practical defenses against pollution attacks in intra-flow network coding for wireless mesh networks," presented at the Proceedings of the second ACM conference on Wireless network security, Zurich, Switzerland, 2009.

[148] y. Zhen, W. Yawen, B. Ramkumar, and G. Yong, "An Efficient Scheme for Securing XOR Network Coding against Pollution Attacks," in *INFOCOM 2009, IEEE*, 2009, pp. 406-414.

[149] S. Vyetrenko, A. Khosla, and T. Ho, "On combining information-theoretic and cryptographic approaches to network coding security against the pollution attack," in *Signals, Systems and Computers, 2009 Conference Record of the Forty-Third Asilomar Conference on*, 2009, pp. 788-792.

[150] J. Dong, R. Curtmola, C. Nita-Rotaru, and D. Yau, "Pollution Attacks and Defenses in Wireless Inter-Flow Network Coding Systems," in *Wireless Network Coding Conference (WiNC), 2010 IEEE*, 2010, pp. 1-6.

[151] L. Buttyan, L. Czap, and I. Vajda, "Detection and Recovery from Pollution Attacks in Coding-Based Distributed Storage Schemes," *Dependable and Secure Computing, IEEE Transactions on,* vol. 8, pp. 824-838, 2011.

[152] Q. Wenbo, L. Jian, and R. Jian, "An Efficient Error-Detection and Error-Correction (EDEC) Scheme for Network Coding," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, 2011, pp. 1-5.

[153] L. Yongkun and J. C. S. Lui, "Identifying Pollution Attackers in Network-Coding Enabled Wireless Mesh Networks," in *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, 2011, pp. 1-6.

[154] L. Anh and A. Markopoulou, "TESLA-Based Defense against Pollution Attacks in P2P Systems with Network Coding," in *Network Coding (NetCod), 2011 International Symposium on*, 2011, pp. 1-7.

[155] F. Oggier and H. Fathi, "An Authentication Code Against Pollution Attacks in Network Coding," *Networking, IEEE/ACM Transactions on,* vol. 19, pp. 1587-1596, 2011.

[156] L. Anh and A. Markopoulou, "On detecting pollution attacks in inter-session network coding," in *INFOCOM, 2012 Proceedings IEEE*, 2012, pp. 343-351.

[157] D. Jing, R. Curtmola, C. Nita-Rotaru, and D. K. Y. Yau, "Pollution Attacks and Defenses in Wireless Interflow Network Coding Systems," *Dependable and Secure Computing, IEEE Transactions on,* vol. 9, pp. 741-755, 2012.

[158] A. Newell and C. Nita-Rotaru, "Split Null Keys: A null space based defense for pollution attacks in wireless network coding," in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2012 9th Annual IEEE Communications Society Conference on*, 2012, pp. 479-487.

[159] Y. Jiang, Y. Fan, X. Shen, and C. Lin, "A self-adaptive probabilistic packet filtering scheme against entropy attacks in network coding," *Comput. Netw.,* vol. 53, pp. 3089-3101, 2009.

[160] A. J. Newell, R. Curtmola, and C. Nita-Rotaru, "Entropy attacks and countermeasures in wireless network coding," presented at the Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks, Tucson, Arizona, USA, 2012.

[161] C. Gkantsidis and P. R. Rodriguez, "Cooperative Security for Network Coding File Distribution," in *INFOCOM, 25th IEEE International Conference on Computer Communications*, 2006.

[162] T. MATSUDA, T. NOGUCHI, and T. TAKINE, "Survey of Network Coding and Its Applications," *EICE TRANS. COMMUN,* vol. E94–B, 2011.

[163] k. Harada and H. Yamamoto, "Strongly secure linear network coding," *IEICE Trans. Fundamentals,* vol. E91-A, pp. 2720–2728, 2008.

[164] J. Feldman, T. M. Servedio, R. A., and C. Stein, "On the Capacity of Secure Network Coding," in *42nd Allerton Conf. Commun.,Control, and comput.*, Monticello, 2004.
[165] K. Bhattad and K. R. Narayanan, "Weakly Secure Network Coding," 2005.
[166] D. Silva and F. R. Kschischang, "Universal Weakly Secure Network Coding," in *ITW 2009, Volos, Greece*, 2009.
[167] D. Silva, F. R. Kschischang, and R. Koetter, "A Rank-Metric Approach to Error Control in Random Network Coding," *Information Theory, IEEE Transactions on,* vol. 54, pp. 3951-3967, 2008.
[168] S. Hong, X. Xiao, W. Weiping, and Y. Luming, "DENNC: A Wireless Malicious Detection Approach Based on Network Coding," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on*, 2011, pp. 160-165.
[169] E. Kehdi and L. Baochun, "Null Keys: Limiting Malicious Attacks Via Null Space Properties of Network Coding," in *INFOCOM 2009, IEEE*, 2009, pp. 1224-1232.
[170] L. Shizheng and A. Ramamoorthy, "Improved Compression of Network Coding Vectors Using Erasure Decoding and List Decoding," *Communications Letters, IEEE,* vol. 14, pp. 749-751, 2010.
[171] S. Vladimirov, "New algorithm for message restoring with errors detection and correction using binary LDPC-codes and network coding," in *Computational Technologies in Electrical and Electronics Engineering (SIBIRCON), 2010 IEEE Region 8 International Conference on*, 2010, pp. 40-43.
[172] N. Cai and R. W. Yeung, "Network Error Correction, II: Lower Bounds," *Commun. Inf. Syst.,* vol. 6, pp. 37-54, 2006.
[173] R. W. Yeung and N. Cai, "Network Error Correction, I: Basic Concepts and Upper Bounds," *Commun. Inf. Syst.,* vol. 6, pp. 19-36, 2006.
[174] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," presented at the Proceedings of the 6th annual international conference on Mobile computing and networking, Boston, Massachusetts, United States, 2000.
[175] M. Kim, M. Medard, J. Barros, and R. Kotter, "An algebraic watchdog for wireless network coding," presented at the Proceedings of the 2009 IEEE international conference on Symposium on Information Theory - Volume 2, Coex, Seoul, Korea, 2009.
[176] J. Hortelano, J. C. Ruiz, and P. Manzoni, "Evaluating the Usefulness of Watchdogs for Intrusion Detection in VANETs," in *Communications Workshops (ICC), 2010 IEEE International Conference on*, 2010, pp. 1-5.
[177] D. Boneh, D. Freeman, J. Katz, and B. Waters, "Signing a Linear Subspace: Signature Schemes for Network Coding," presented at the Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography: PKC '09, CA, 2009.
[178] D. Charles, K. Jain, and K. Lauter, "Signatures for network coding," *International Journal of Information and Coding Theory,* vol. 1, pp. 3-14, 2009.
[179] E. Kehdi and B. Li, "Null Keys: Limiting Malicious Attacks Via Null Space Properties of Network Coding," in *INFOCOM 2009*, 2009.
[180] Q. Li, D.-m. Chiu, and J. C. S. Lui, "On the Practical and Security Issues of Batch Content Distribution Via Network Coding," presented at the Proceedings of the

Proceedings of the 2006 IEEE International Conference on Network Protocols, 2006.

[181] Y. Li, H. Yao, M. Chen, S. Jaggi, and A. Rosen, "RIPPLE authentication for network coding," presented at the Proceedings of the 29th conference on Information communications, San Diego, California, USA, 2010.

[182] F. Zhao, T. Kalker, M. Médard, and K. J. Han, "Signatures for content distribution with network coding," in *In Proc. of International Symposium on Information Theory (ISIT)*, 2007.

[183] E. Porat and E. Waisbard, "Efficient signature scheme for network coding," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, 2012, pp. 1987-1991.

[184] S. Zhong, L. E. Li, Y. G. Liu, and Y. R. Yang, "On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks: an integrated approach using game theoretic and cryptographic techniques," *Wirel. Netw.,* vol. 13, pp. 799-816, 2007.

[185] S. Agrawal and D. Boneh, "Homomorphic MACs: MAC-Based Integrity for Network Coding," presented at the Proceedings of the 7th International Conference on Applied Cryptography and Network Security, Paris-Rocquencourt, France, 2009.

[186] M. Adeli and L. Huaping, "Secure network coding with minimum overhead based on hash functions," *Communications Letters, IEEE,* vol. 13, pp. 956-958, 2009.

[187] Z. Kaiyong, C. Xiaowen, W. Mea, and J. Yixin, "Speeding Up Homomorpic Hashing Using GPUs," in *Communications, 2009. ICC '09. IEEE International Conference on*, 2009, pp. 1-5.

[188] K. Han, T. Ho, R. Koetter, M. Medard, and F. Zhao, "On network coding for security," in *Military Communications Conference, 2007. MILCOM 2007. IEEE*, 2007, pp. 1-6.

[189] S. Yang and R. W. Yeung, "Large file transmission in network-coded networks with packet loss: a performance perspective," presented at the Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies, Barcelona, Spain, 2011.

[190] Y. Shenghao and R. W. Yeung, "Coding for a network coded fountain," in *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, 2011, pp. 2647-2651.