

## ABSTRACT

Title of Document:

COMPONENT BASED ROUTING  
PROTOCOL DESIGNING METHODOLOGY  
FOR MANET

He Huang

Directed By:

Professor John S.Baras

Mobile Ad Hoc Network is designed and deployed to achieve self-configuring and self-healing. MANET utilizes distributed wireless stations for relaying data packets. Every single station in the MANET can decide routing path for an incoming data packet. MANET has the most unfavorable conditions for routing path discovery due to node mobility and constant topology changes. Large variation of performance due to various environment inputs is a major impediment of implementing existing routing protocols for MANET in the battlefield. Therefore, it is a major challenge to design a routing protocol that can adapt its behavior to environment alteration.

In consideration of adaptability to the environment and flexibility in protocol construction, a novel component based routing protocol methodology is proposed in this paper. Distinguished from conventional investigation of routing protocols as individual entities, this paper will firstly generalize four fundamental components for MANET routing protocols. Then, a significant component diagnosis process is

proposed to detect significant component and enhance the overall performance. Finally, preliminary simulation results demonstrate the power of the component based methodology for improving overall performance and reducing performance variation. In conclusion, the evaluation and improvement at the component level is more insightful and effective than that at the protocol level.

The primary contribution of the work is proposing the Component Dependence Network the first time and innovative quantitative methods are proposed to learn the structure and significant component to analyze the impact of component on performance metrics.

Based on conditional independence test, hierarchical structure of Component Dependence Network can be discovered. An Inclusion and Exclusion algorithm is introduced to guarantee the minimal cut set returned for a pair of source and destination nodes. To determine the significant component, a significance indicator will be calculated based on comparing each component's impact by using a backward deriving method. Once the significant component being determined, the parameter of the significant component can be tuned to achieve the best performance. At the end, two real implementations are presented to show the achievement in performance improvement of the component dependence network, structure learning method and significant component indicator.

COMPONENT BASED ROUTING PROTOCOL DESIGNING METHODOLOGY  
FOR MANET

By

He Huang

Dissertation submitted to the Faculty of the Graduate School of the  
University of Maryland, College Park, in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
2010

Advisory Committee:

Professor John S. Baras, Chair

Professor Armand Makowski

Professor Richard J. La

Professor Mark Austin

Professor S. Raghuraghavan, Dean's Representative

© Copyright by  
He Huang  
2010

Dedication

To my grandparent

To my parent

## Acknowledgements

I would like to express my deep and sincere gratitude to my supervisor, Professor John S. Baras. His wide knowledge and logical way of thinking have a remarkable influence on my entire career of quantitative analysis. I am grateful to him for offering me an invaluable opportunity to work on challenging and extremely interesting projects. Thanks for broadening my scope and helping me achieving my goals. Dr. Baras is deeply appreciated for his guidance and full support during my difficult times in my research and life.

I would like to thank Professor Armand Makowski, Professor Richard J. La, Professor Mark Austin and Professor S. Raghu Raghavan for serving in my advisory committee and providing many insightful suggestions and comments.

I would like to express my gratitude to Kimberly Edwards for her full support for my work and life.

I would like to express my gratitude to my colleagues and friends at Maryland, who offered great help on my research and life. Thanks for sharing their knowledge and experience with me.

I would like to thank the ECE department for financial support for my graduate studies. I would like to thank the financial support for my research from U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. Research is also supported by the U.S. Army Research Office under grant No DAAD19-01-1-0494.

Finally, thanks to my parents for their unconditional love and support. And, thanks to my husband and daughter for accompanying me during ups and downs.

November 2010

He Huang

## Table of Contents

Dedication.....	ii
Acknowledgements.....	iii
Table of Contents.....	v
List of Tables.....	viii
List of Figures.....	xi
1. Introduction.....	1
1.1. Motivation for Component Based Routing Protocol Design.....	5
1.2. Contributions to Routing Protocol Design for MANET.....	8
1.3. Organization.....	11
2. Overview of MANET Routing Protocols.....	13
2.1. AODV.....	13
2.2. DSR.....	19
2.3. Other Protocols.....	24
2.3.1. Optimized Link State Routing.....	24
2.3.2. Probabilistic Emergent Routing.....	25
2.3.3. Zone Routing Protocol.....	25
2.4. Summary of Performance Analysis for Routing Protocols.....	26
3. Framework of Component Based Routing Protocol.....	30
3.1. Architecture of Component Based Routing Protocol.....	34
3.2. Component Framework.....	34
3.3. Specification of subcomponents.....	38



3.4. Identification of Information Carrier.....	42
3.5. Component Performance Metrics.....	45
3.6. Interpretation of Routing Performance by Component Metrics.....	51
4. Bayesian Induction of Components' Probabilistic Dependence.....	57
4.1. Differential Performance Projection.....	58
4.2. Detection of Significant Component by Bayesian Belief Network.....	61
4.3. Bayesian Network Structure for Component Dependence.....	71
4.4. Replacement of Significant Component.....	73
4.5. Simulation Results and Analysis.....	77
5. Structure Learning of Component Dependence Network.....	86
5.1 Introduction to Component Based Routing Design.....	89
5.2 Hierarchical Structure for Component Dependence Network.....	91
5.3 d-separations Theorem and Formal Logical Arithmetic.....	94
5.4 Include-Exclude Algorithm for Structure Induction .....	100
5.4.1 Introduction to Include-Exclude Algorithm.....	101
5.4.2 Proof of correctness of Include-Exclude algorithm.....	107
5.4.3 Parent Set Learning Algorithm.....	112
5.5 Statistical Methods Applied in Structure Learning and Significant Component Learning.....	113
5.5.1 Statistical Methods in Structure Learning.....	113
5.5.2 Statistical Methods in Significant Component Learning.....	114
5.6 Examples of application of Component Dependence Network.....	119
5.6.1 Application of component Dependence Network to detect	

significant component for Data Packet Loss Ratio.....	120
5.6.2 Application of component Dependence Network to detect	
significant component for End to End Delay.....	135
Bibliography.....	153

## List of Tables

5.1 Table of conditional probability of $X_{41}$ given $X_{31}$ and $X_{32}$ under data traffic and low mobility speed.....	123
5.2 Table of entropy of $X_{41}$ given $X_{31}$ and $X_{32}$ under data traffic and low mobility speed.....	123
5.3 Table of probability distribution of $X_{31}$ under data traffic and low mobility speed.....	124
5.4 Table of probability distribution of $X_{32}$ under data traffic and low mobility speed.....	124
5.5 Table of derivation of entropy of $X_{41}$ given $X_{32}$ under data traffic and low mobility speed.....	124
5.6 Table of derivation of entropy of $X_{41}$ given $X_{31}$ under data traffic and low mobility speed.....	125
5.7 Table of conditional probability of $X_{31}$ given $X_{21}$ , $X_{22}$ and $X_{24}$ under data traffic and low mobility speed.....	126
5.8 Table of entropy of $X_{31}$ given $X_{21}$ , $X_{22}$ and $X_{24}$ under data traffic and low mobility speed.....	127
5.9 Table of Joint probability distribution of $X_{22}$ and $X_{24}$ under data traffic and low mobility speed.....	128
5.10 Table of Joint probability distribution of $X_{21}$ and $X_{24}$ under data traffic and low mobility speed.....	128
5.11 Table of Joint probability distribution of $X_{21}$ and $X_{22}$ under data traffic and low mobility speed.....	128
5.12 Table of derivation of entropy of $X_{31}$ given $X_{22}$ and $X_{24}$	

under data traffic and low mobility speed.....	129
5.13 Table of derivation of entropy of $X_{31}$ given $X_{21}$ and $X_{24}$	
under data traffic and low mobility speed.....	129
5.14 Table of derivation of entropy of $X_{31}$ given $X_{21}$ and $X_{22}$	
under data traffic and low mobility speed.....	130
5.15 Table of conditional probability of $X_{24}$ given $X_{12}$ and $X_{14}$	
under data traffic and low mobility speed.....	132
5.16 Table of entropy of $X_{24}$ given $X_{12}$ and $X_{14}$	
under data traffic and low mobility speed.....	132
5.17 Table of Conditional Probability of $X_{41}$ given $X_{31}$ and $X_{32}$	
under Data Traffic and Low Mobility Speed.....	139
5.18 Table of Entropy of $X_{41}$ given $X_{31}$ and $X_{32}$	
under Data Traffic and Low Mobility Speed.....	139
5.19 Table of Probability Distribution of $X_{31}$	
under Data Traffic and Low Mobility Speed.....	140
5.20 Table of Probability Distribution of $X_{32}$	
under Data Traffic and Low Mobility Speed.....	140
5.21 Table of Deviation of Entropy of $X_{41}$ given $X_{32}$	
under Data Traffic and Low Mobility Speed.....	140
5.22 Table of Deviation of Entropy of $X_{41}$ given $X_{31}$	
under Data Traffic and Low Mobility Speed.....	141
5.23 Table of Conditional Probability of $X_{31}$ given $X_{21}$ , $X_{22}$ and $X_{23}$	
under Data Traffic and Low Mobility Speed.....	142

5.24 Table of Entropy of $X_{31}$ given $X_{21}$ , $X_{22}$ and $X_{23}$	
under Data Traffic and Low Mobility Speed.....	143
5.25 Table of Joint Probability Distribution of $X_{22}$ and $X_{24}$	
under Data Traffic and Low Mobility Speed.....	144
5.26 Table of Joint Probability Distribution of $X_{21}$ and $X_{24}$	
under Data Traffic and Low Mobility Speed.....	144
5.27 Table of Joint Probability Distribution of $X_{21}$ and $X_{22}$	
under Data Traffic and Low Mobility Speed.....	144
5.28 Table of Deviation of Entropy of $X_{31}$ given $X_{22}$ and $X_{23}$	
under Data Traffic and Low Mobility speed.....	145
5.29 Table of Deviation of Entropy of $X_{31}$ given $X_{21}$ and $X_{24}$	
under Data Traffic and Low Mobility Speed.....	145
5.30 Table of Deviation of Entropy of $X_{31}$ given $X_{21}$ and $X_{22}$	
under Data Traffic and Low Mobility Speed.....	146
5.31 Table of Conditional Probability of $X_{22}$ given $X_{11}$ and $X_{12}$	
under Data Traffic and Low Mobility Speed.....	148
5.32 Table of Entropy of $X_{24}$ given $X_{11}$ and $X_{12}$	
under Data Traffic and Low Mobility Speed.....	148

## List of Figures

1.1 Infrastructure-based wireless network.....	2
1.2 Wireless Ad-Hoc network.....	3
2.1 AODV activity flowchart for user: Data Packet Receiver.....	16
2.2 AODV activity flowchart for user: RREQ Packet Receiver.....	17
2.3 AODV activity flowchart for user: RREP Packet Receiver.....	18
2.4 DSR activity flowchart for user: Data Packet Receiver.....	21
2.5 DSR activity flowchart for user: RREQ Packet Receiver.....	22
2.6 DSR activity flowchart for user: RREP packet Receiver.....	23
3.1 Architecture of component based routing.....	34
3.2 Simulation result for path discovery overhead.....	53
3.3 Simulation result for percentage of cached RREP.....	54
3.4 Simulation result for percentage of path discovery success.....	54
3.5 Simulation result for percentage of cache hit for data packet.....	55
4.1 Simulation result for video traffic.....	59
4.2 Simulation result for mobility speed 15MPH.....	60
4.3 Outline of Bayesian network structure for component dependence.....	71
4.4 Simulation result for percentage of cache hit for data packet.....	74
4.5 Simulation result of data packet loss ratio.....	75
4.6 Activity diagram for topology database maintenance component.....	76
4.7 Simulation result for data packet received ratio.....	78
4.8 Simulation result for improving ratio for data packet received ratio.....	79

4.9 Simulation result for increment percentage of routing overhead.....	79
4.10 Simulation result for increment ratio for end-to-end delay.....	80
4.11 Simulation result for data packet received ratio for AODV.....	80
4.12 Simulation result for improving ratio for data packet received ratio.....	82
4.13 Simulation result for increment ratio for routing overhead.....	83
4.14 Simulation result for increment percentage for end-to-end delay.....	83
5.1 The component dependence network	
Ten variable nodes ( $X_{11}, X_{12}, \dots, X_{41}$ ) are contained.....	93
5.2 A sample CDN. Illustrate the scenario that requires	
Twice exclude phase to return minimal cut set.....	99
5.3 Flowchart for include-exclude algorithm.....	102
5.4 Step 1 in exclude phase for sample network.....	105
5.5 Step2 in exclude phase for sample network.....	105
5.6 Step3 in exclude phase for sample network.....	106
5.7 Four types of nodes on 2 <sup>nd</sup> layer.....	108
5.8 Flowchart for learning algorithm for 3 <sup>rd</sup> layer's parent.....	112
5.9 Layer structure of independence network for packet loss ratio learning.....	121
5.10 Layer structure obtained by Include-Exclude algorithm.....	122
5.11 Chart of data packet loss ratio and routing overhead	
against the frequency of hello message.....	134
5.12 Layer Structure of Independence Network for	
End to End Delay learning.....	136

5.13 Layer Structure Obtained by Include-Exclude Algorithm.....	138
5.14 Chart of End to End Delay against MANET Station Mobility Speed.....	149
5.15 Chart of Average Data Packet's End to End Delay and Routing Overhead against TTL Values.....	151



# **Chapter 1**

## **Introduction**

The initial development of ad-hoc networks was primarily driven by military applications, where self-configuring and self-healing in battlefield networks are key requirements. The battlefield network often yields unstable networks. In some situations, stations that are responsible for relaying data packets will be unavailable (destroyed or terminated). Under these harsh conditions, the networks are required to be able to search for new routing paths to deliver data packets to destinations successfully. The ability of self-configuring and self-healing are the most significant features for the mobile battlefield network.

As early as 1972, packet radio networks were developed by DARPA [9]. In the middle 1990s, research on wireless ad-hoc networks were accelerated by the popularity of cheaper radio equipment (e.g. 802.11 and Bluetooth). Examples of current wireless ad-hoc networks are Mobile Ad Hoc Networks (MANET) and Wireless Sensor Networks. We'll focus on MANET in this thesis. In terms of routing protocol, MANET have the most unfavorable conditions for routing path discovery due to node mobility and constant topology changes.

A wireless ad-hoc network is a computer network that deploys wireless communication links. Every node in the network is able to relay data packets on behalf of other nodes, and can determine dynamically which neighbor the data packet will be forwarded to. In traditional infrastructure-based wireless networks, the base station has to be installed before communication can start (Figure 1.1).

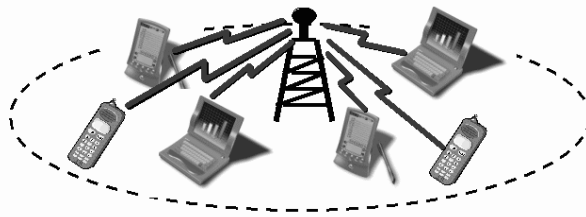


Figure 1.1 Infrastructure-based wireless network

In infrastructure-based wireless networks, every data packet will be first delivered to the nearest base station, and the base station will determine how to relay the data packet to the destination station (either determines by which neighboring station or by what station sequence). Actually, the base station is a central node that receives data packets from source nodes and delivers data packets to destination nodes.

If the base station is terminated in the battlefield, then the central stations relaying data packets will be unavailable. In this difficult situation, all data packets will be blocked and won't be delivered to destination stations. To avoid these centralized

style of transportation in wireless communication; a Mobile Ad Hoc Network is designed and deployed to achieve self-configuring and self-healing. MANET utilize distributed wireless stations for relaying data packets. Every single station in the MANET can decide the routing path for an incoming data packet.

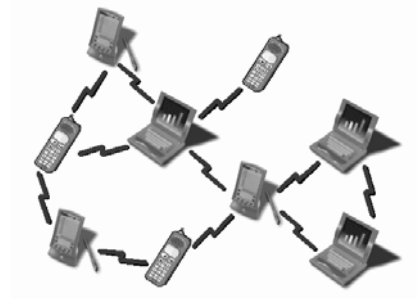


Figure 1.2 Wireless Ad-hoc network

As demonstrated in Figure 1.2, ad-hoc networks can be conveniently deployed without existing infrastructure or pre-configuration. Additionally, every node in ad-hoc networks maintains routing tables and updates routing tables dynamically by exchanging information of local link connections with other nodes. This distributed routing computation, performed dynamically in wireless ad-hoc networks, is self-healing. In the event that a node on a routing path is failed, or a link connection is going down, the routing computation will be performed immediately to recover new routing paths. Hence, unlike infrastructure-based wireless networks, in the situation

of failed relaying stations, the mobile Ad Hoc network is still able to deliver data packets to desired destination stations.

For the two major characteristics of self-configuring and self-healing, wireless ad-hoc networks are widely applied in emergency situations like disaster, military battlefields and urgent medical situations, etc.

In recent years, increasingly widespread application of Mobile Ad Hoc Networks accelerates development of mobile distributed computing. Many routing protocols have been designed for Ad Hoc networks to satisfy the needs for more actively distributed algorithms. Ad Hoc routing protocols can be categorized into two major prototypes: proactive (OLSR) and reactive (AODV, DSR). These two types of protocols are featured by the mechanism of path discovery. Proactive routing protocols constantly update the routing cache. One example of proactive routing is OLSR [10], which updates the routing cache by broadcasting Topology Control messages every fixed interval. The most up-to-date map of a network's topology is maintained in the route cache table regardless of the need for a route to a specified destination. Consequently, proactive routing protocols constantly maintain a set of available routes for all source and destination paired stations throughout the network. Reactive (on demand) routing protocols, on the other hand, only initiate the path discovery procedure in response to a route request to a specified destination. No route control message will be generated until a route is required for a destination (Some reactive routing protocols do employ regular interval routing messages to update neighborhood connectivity, such as AODV). Reactive routing protocols do not

propagate routing control message periodically, and only maintain routing entries for a subset of the entire network stations. Only in the face of need of routing path to a desired destination for a data packet, the route discovery procedure will be initiated, and a routing path will be returned to the source node. If an existing routing path can be located in the routing table, then this routing path will be used to forward the data packets, and no route discovery procedure will be initiated. AODV [11] and DSR [12] are the most popular examples of reactive routing protocols.

### **1.1. Motivation for Component Based Routing Protocol Design**

Up to now, routing protocols are always studied as individual entities. The complexity and ambiguous structures make protocols difficult to be understood and implemented in real applications. After thoroughly investigating existing routing protocols for MANET, we discovered that many common functions are shared among different routing protocols. For instances, both proactive routing protocol OLSR and reactive protocol AODV update their local topology information by broadcasting HELLO messages periodically. Furthermore, all routing protocols conduct four primary functions:

- Path Discovery
- Topology Database Maintenance
- Route Maintenance
- Data Packet Forwarding

A standardized framework for protocols' structures is desired for understanding protocols at the building-block level instead of the protocol level. Therefore, the first motivation for component based routing protocol design is to decouple complicated routing protocols into standardized component framework for easy understanding and implementation.

Secondly, extensive research has been done to compare and explain the differences of various routing protocols at the protocol level through simulation or intuitive analysis [13] [14] [15]. This enabled us to evaluate protocols according to their performance and distinguish the pros and cons of each protocol. But more insightful interpretation from the building-block level remains unavailable. The conventional metrics for routing protocols are end-to-end delay, packet loss ratio and routing traffic overhead. Statistics of these metrics can only demonstrate if one routing protocol performs well or not, without showing reasons of why the protocol performs well or not. Therefore, each building block needs to be traced separately to make conclusions of how each building block put an effect on the overall routing protocol's performance.

Thirdly, although many routing protocols have been designed for mobile ad hoc networks, none of them can maintain high performance under all situations. Each existing routing protocol can only outperform others for certain scenarios. In comparison of performance of various MANET routing protocols, a number of technical papers have been published. Some general conclusions have been obtained:

- a. Reactive routing protocols will perform better than proactive protocols in highly dynamic networks [15].
- b. Among reactive protocols, AODV outperforms DSR in more stressful situations (with higher mobility speed, with higher traffic density) [13].
- c. DSR consistently generates less routing traffic overhead than AODV [13, 14].

As stated in the literature, large variation of performance of routing protocols is exhibited in the above papers. A protocol outperforming in certain situations may not do so in other situations. None of the existing protocols can absolutely outperform all others in overall situations. Furthermore, none of the existing protocols can satisfy all applications since different users of routing protocols may have different performance expectations. For example, DSR is a better choice with expectation of low control overhead property, while AODV is a better choice with expectation of low packet loss ratio under situation of high mobility. By deeply investigating MANET routing protocols, we discovered that some key functions can influence overall performance significantly. In some cases, substitution of some functions instead of entire routing protocols can improve performance significantly. Hence, selecting suitable components from function repository and assembling them together into a well performed routing protocol is an efficient alternative other than rewriting the entire protocol to the end.

In conclusion, motivations for proposing component based routing protocol are listed below:

1. To abstract, decouple and organize routing protocols into a standardized framework for easier understanding and implementation.

2. To conduct performance analysis on building-block level for more insightful explanation on overall routing performance.

3. To construct highly adaptive routing protocols for all scenarios and QoS requirements by selecting suitable routing components.

## **1.2. Contributions to Routing Protocol Design for MANET**

In the past, all research on routing protocol design were only based on entire routing protocols. A lot of protocols are designed to satisfy a specified requirement of quality of service under different network conditions. However, as stated above, we don't need to replace an entire routing protocol under some situations. We only need to replace some routing functions to achieve a better routing performance (this will be proved in the following chapters). This thesis made the following contributions listed below:

1. This is the first time that the concept of component based routing protocol is proposed. Routing protocols are decoupled into a standardized block framework. This makes research and implementation of routing protocols much easier. In the future, the changing of routing protocols can be implemented only by selecting and assembling different components together. In real applications, this method of "plug and play" will make implementation much easier and more efficient. Under the



framework of component based routing protocol, a pool of available components will be available for configuration. When a new routing protocol is designed and implemented, it's convenient to select from available components and assembly them together. However, in the traditional routing design, the routing protocol has to be implemented from nothing. This is a difficult task that is very time consuming in research and field test.

2. Based on the concept of component based routing protocol, the idea is proposed to use component performance metrics to monitor routing performance. The component performance metrics are very beneficial in analyzing the routing component's performance. Component performance metrics are much more insightful than overall metrics. When we need an interpretation of bad routing performance, component performance can disclose more deeply why routing performance is deteriorated. Overall routing performance is closely related to component performance metrics. When overall routing performance is deteriorated, the change in quantitative measures can be detected from component performance. This quantitative change is critical in leading us to locate weak component, which is a bottleneck component that mostly negatively affect the overall performance. After locating the weak component, we can start to consider how to replace the weak component to improve the overall routing performance.

3. By analyzing component metrics, we can have a quantitative analysis of how each component affects overall routing performance. So, a component dependence network is proposed for the first time. The component dependence network is a

mathematical concept that can assist us to quantitatively analyze the internal relation between component performance and overall performance. A structure learning algorithm is proposed in the thesis to study the structure of the component dependence network. After the structure is determined, the affection of routing component can be learned by quantitative analysis. This is the first time that a quantitative method is proposed to learn the root source for a deteriorated routing performance. The structure learning method is inspired by Bayesian network. The following achievements are included in the thesis.

- a. Introduced a novel quantitative analysis model CDN for automatic learning of component's impact on performance metrics. CDN not only comprises all major achievements in our previous work for component based routing framework, but also develops a new idea of encoding the component's impact into a conditional probability distribution which can be visualized by a graphical topology.
- b. Develop a formal logical learning arithmetic for d-separation. This formal arithmetic can be directly implemented in application software to help identify d-separation, hence to determine local conditional independence.
- c. Design the Include-Exclude algorithm with the property of returning a minimal cut set for a source and destination pair.
- d. Based on an independence test, a simple and applicable structure learning algorithm for CDN is proposed.

### **1.3. Dissertation Organization**

In Chapter 2, several popular routing protocols are reviewed, and fundamental analysis of routing performance is proposed. Chapter 3 introduces the structure of a component based routing protocol. The framework of components of routing protocols is introduced. Subcomponents, information carrier and component performance metrics are presented and explained. Chapter 3 makes good preparation for the following chapters by providing a physical framework that will be used throughout the thesis.

Chapter 4 introduces the Bayesian network, and the theory behind the Bayesian network that will be applied in the component dependence network. And, several examples are provided to show how it's effective to replace some routing components to improve the overall routing performance. Chapter 4, by running some real simulation scenarios, can help us to better understand why component based routing protocols are proposed and how they outperform other protocols.

Chapter 5 formally proposes the component dependence network and introduces the Include-Exclude algorithm that is used to learn the structure of the component dependence network. A formal proof for the Include-Exclude algorithm is presented. And, two examples are provided to show it's simple and effective to apply the

Include-Exclude algorithm to learn the structure of the component dependence network.

## **Chapter 2**

### **Overview of MANET Routing Protocols and**

### **Performance Analysis**

In recent years, many routing protocols have been designed for providing dynamic routing computation for mobile ad hoc routing networks. Most of these routing protocols can be categorized into two types: proactive (table-driven) and reactive (on-demand) routing protocols. We'll firstly introduce two most popular MANET routing protocols: AODV and DSR. AODV and DSR are both reactive routing protocols.

#### **2.1. AODV**

Ad-hoc On-demand Distance Vector (AODV) builds routes using a route request / route reply query cycle [11]. When a source node desires a route to a destination for which it does not already have a route, it broadcasts a route request (RREQ) packet across the network. Nodes receiving this packet update their information for the source node and set up backwards pointers to the source node in the route tables. In addition to the source node's IP address, current sequence number, and broadcast ID, the RREQ also contains the most recent sequence number for the destination of which the source node is aware. A node receiving the RREQ may send a route reply (RREP) if it is either the destination or if it has a route to the destination with corresponding

sequence number greater than or equal to that contained in the RREQ. If this is the case, it unicasts a RREP back to the source. Otherwise, it rebroadcasts the RREQ. The RREP generated by destination is named as destination RREP, and RREP initiated by intermediate node is called cached RREP. Nodes keep track of the RREQ's source IP address and broadcast ID. If they receive a RREQ which they have already processed, they discard the RREQ and do not forward it. When RREP propagates back to the source, nodes on the returning path will set up forwarding pointers to the destination. Once the source node receives the RREP, it may begin to forward data packets to the destination. If the source later receives a RREP containing a greater sequence number or contains the same sequence number with a smaller hop count, it may update its routing information for that destination and begin using the better route.

As long as the route remains active, it will continue to be maintained. A route is considered active as long as there are data packets periodically traveling from the source to the destination along that path. Once the source stops sending data packets, the links will time out and eventually be deleted from the intermediate node routing tables. If a link break occurs while the route is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable destination(s). After receiving the RERR, if the source node still desires the route, it can reinitiate route discovery.

AODV remains several salient characteristics. The first is that data packet has to check up routing cache table at each intermediate node to learn the next routing hop. Data packet itself doesn't bring routing information. The second is that only one route is remained for each destination index in AODV's routing cache table. This unique cached route is always the latest route returned by RREP. The sequence number applied in RREQ and RREP enable AODV to compare route's timing order and decide which one is the latest route. The second point about AODV is that HELLO message will be broadcasted periodically to help detect failed link between neighbors. Each AODV node keeps a neighbor list, if one neighbor isn't heard for broadcasting HELLO message for an expiration period, the neighbor will be assumed to be unreachable and will be deleted from the neighbor list. The third point is that AODV has a recovering mechanism called local repair. Whenever a data packet can't find a cached route to its destination at a node, the node will initiate a route request process locally. In contrast, in some other routing protocols, the route discovery process can only be initiated by the source node.

The routing protocol is always activated by the arrival of various types of packet. When different packet is forwarded to routing layer, different protocols will react differently. Below is the activity flowchart for AODV. Figure 2.1 is the activity flowchart for user: data packet. Once data packet arrives, a flow of activity will be triggered by whether or not a routing neighbor can be located in routing table. Figure 2.2 shows that once RREQ packet is received, then if or not a RREP will be created

and sent back to source station. Figure 2.3 shows that what happens when RREP is received by a station.

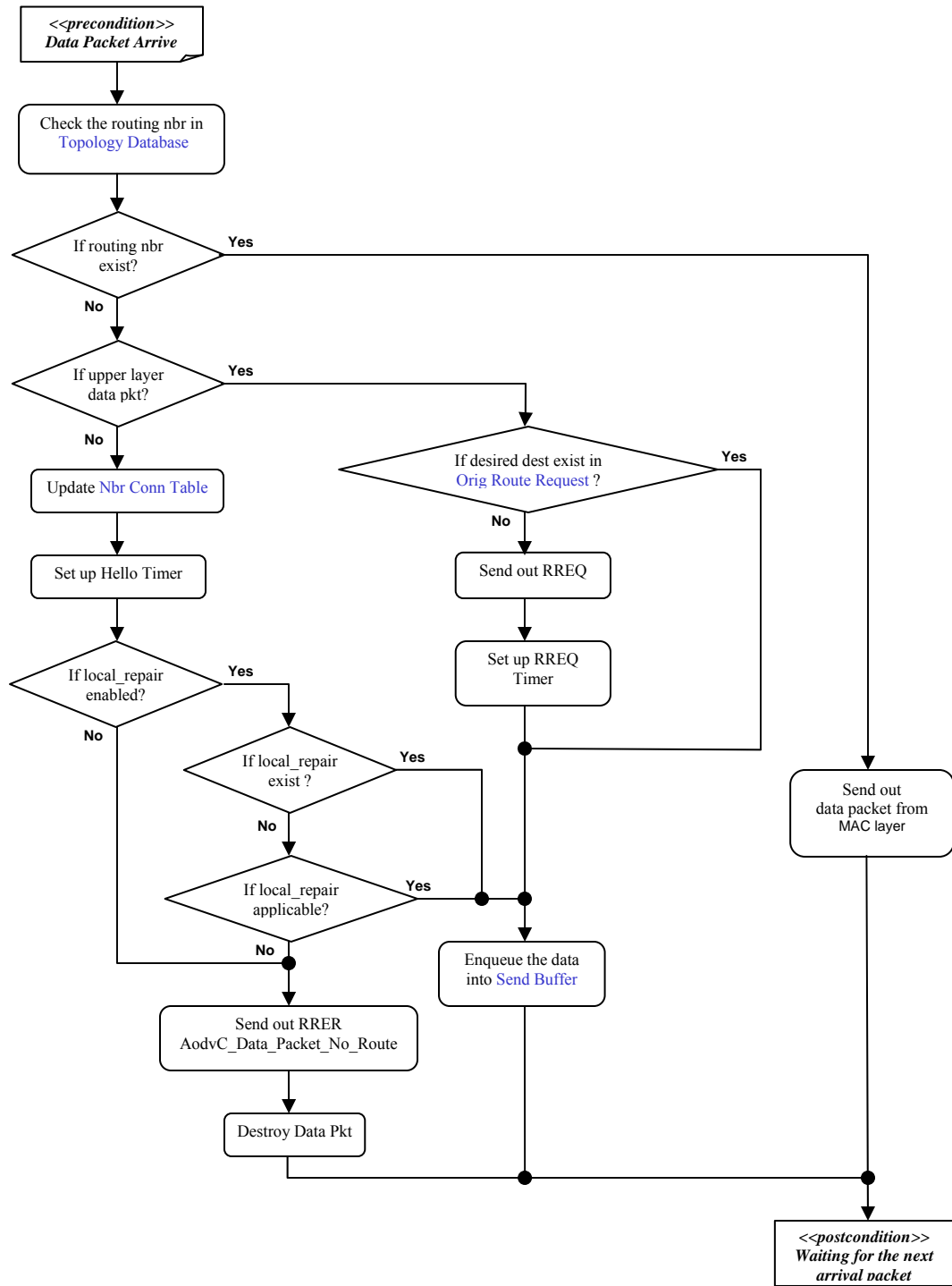


Figure 2.1 AODV Activity Flowchart for User: Data Pkt Receiver



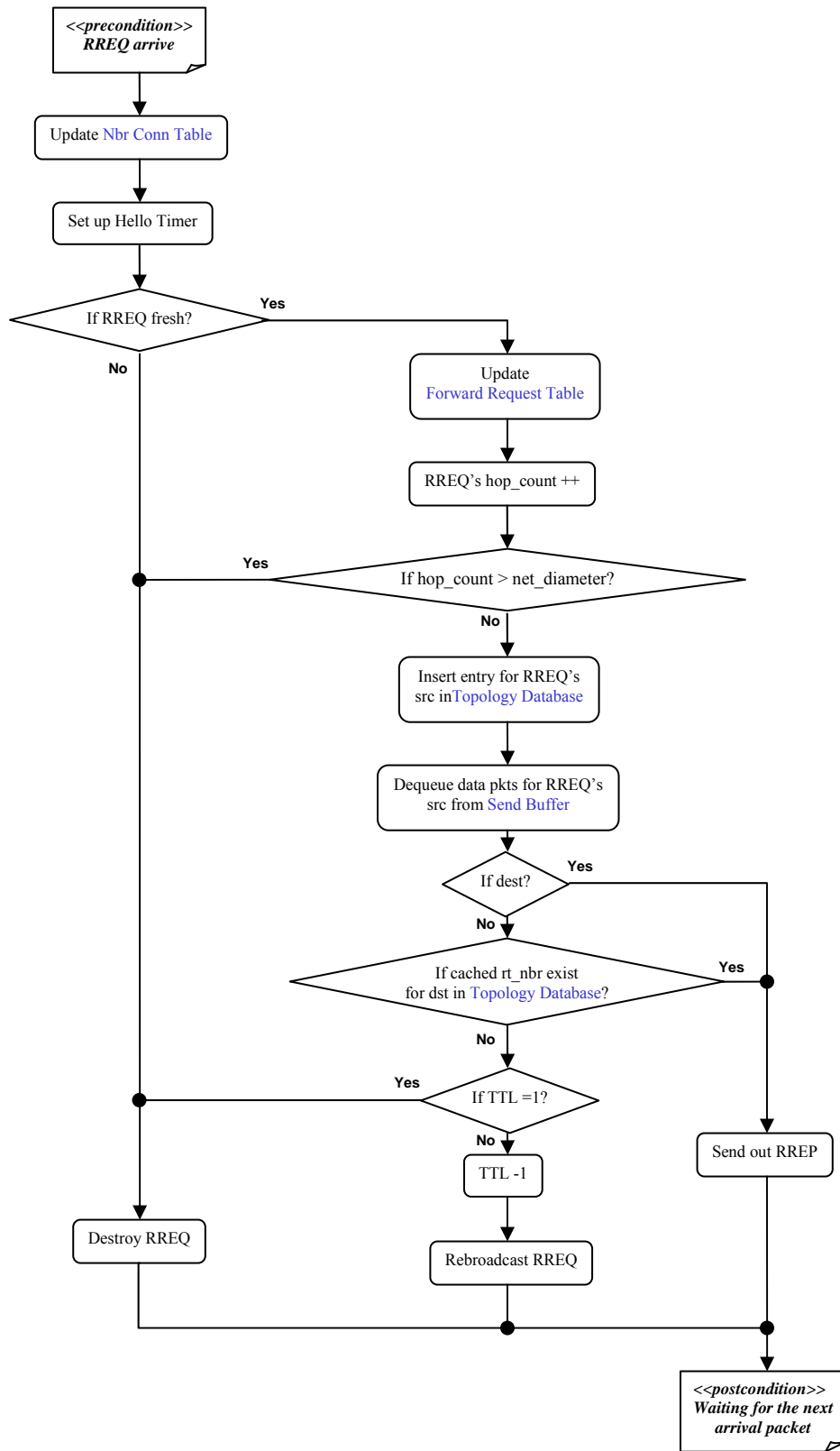


Figure 2.2 AODV Activity Flowchart for User: RREQ Pkt Receiver

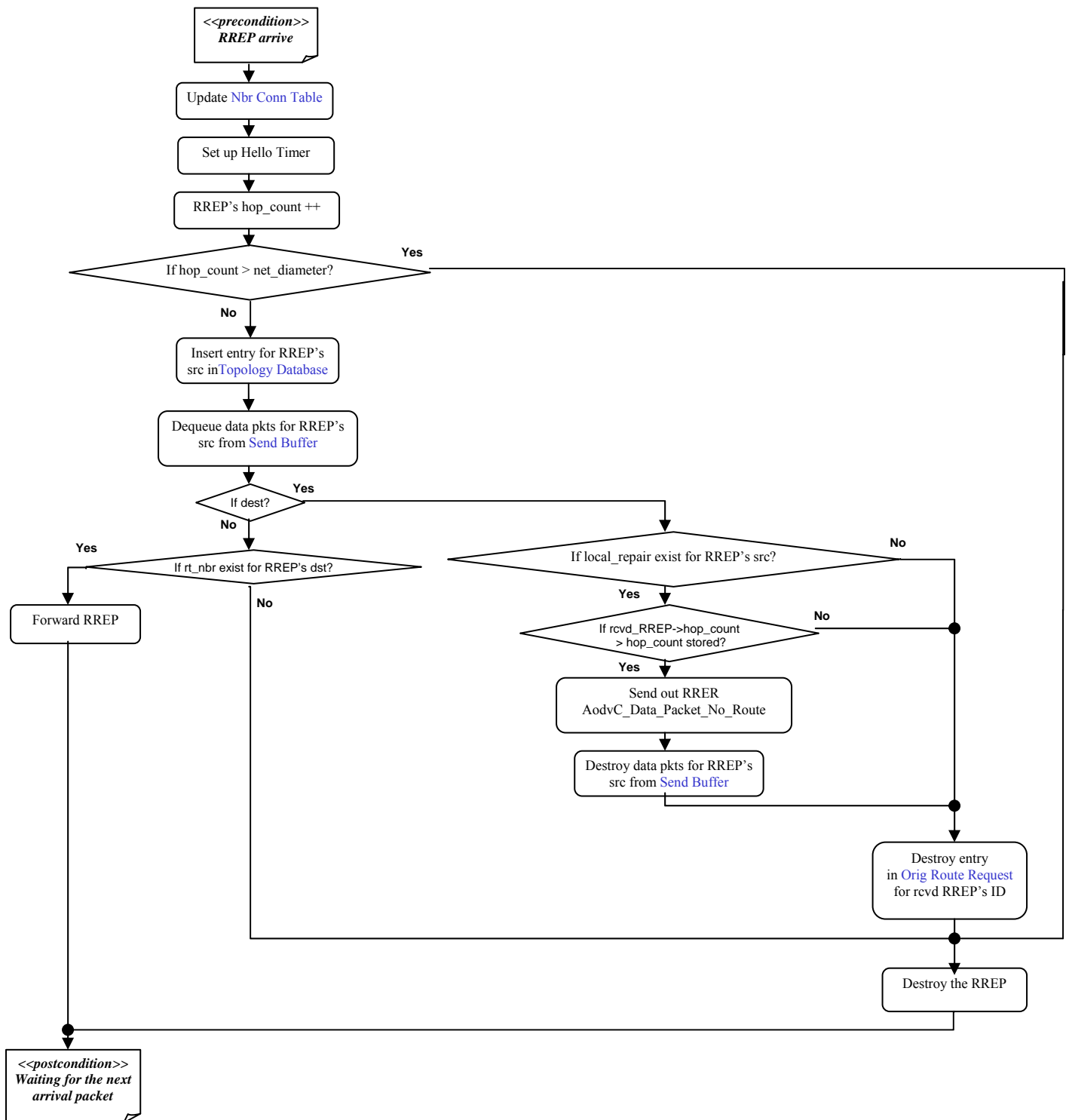


Figure 2.3 AODV Activity Flowchart for User: RREP Pkt Receiver

## 2.2. DSR

Dynamic Source Routing (DSR) is another leading routing protocol for mobile ad hoc networks [12]. It is similar to AODV in that it forms a route on-demand when a transmitting computer requests one. However, it uses source routing instead of relying on the routing table at each intermediate device.

Determining source routes requires accumulating the address of each node between the source and destination during route discovery. The source route is learned during route discovery process. RREQ packet is initiated at the source node, and is broadcasted to neighbors. The intermediate nodes passed by RREQ are recorded by RREQ packet. The accumulated path information is cached by nodes processing the route discovery packets. The learned paths are used to forward packets. To accomplish source routing, the routed packets contain the address of each node the packet will traverse.

This protocol is truly based on source routing whereby all the routing information is maintained (continually updated) at mobile nodes. It has only 2 major phases which are Route Discovery and Route Maintenance. Each data packet encapsulate entire routing path in packet header. Each intermediate node will search data packet's header to find out which is the next hop should forward the data packet to. Before forwarding data packet to next routing hop, the data packet will be copied into maintenance buffer, and wait for an acknowledgement returned from the next hop. If Ack can't be heard from the next hop within a timeout period, the data packet will be

forwarded again, until an Ack is received from the next routing hop. Otherwise, if Ack isn't received after the maximum number of retransmission is reached, data packet will be deleted from the maintenance buffer.

To return the Route Reply, the destination node must have a route to the source node. If the route is in the Destination Node's route cache, the route would be used. Otherwise, the node will reverse the route based on the route record in the Route Reply message header (symmetric links). In the event of fatal transmission, the Route Maintenance Phase is initiated whereby the Route Error packets are generated at a node. The erroneous hop will be removed from the node's route cache. All routes containing the hop are truncated at that point. Again, the Route Discovery Phase is initiated to determine the most viable route.

DSR has several characteristics. The first is that, in contrast to AODV using HELLO message, DSR doesn't use any periodic packet to detect link status. The second is that DSR can store multiple routing paths for a single destination. This is different from AODV, which only keep single cached route for each destination. The third one is that DSR implement Ack-request and Ack message to confirm the reachability of next routing hop.

Based on different routing mechanism, DSR has a different activity diagrams for routing functions. As shown in below activity flowchart, Figure 2.4 shows that what happens when data packet arrives. Figure 2.5 show that when RREQ is received, how the station react. And Figure 2.6 shows that when RREP is received by station, how stations operate.

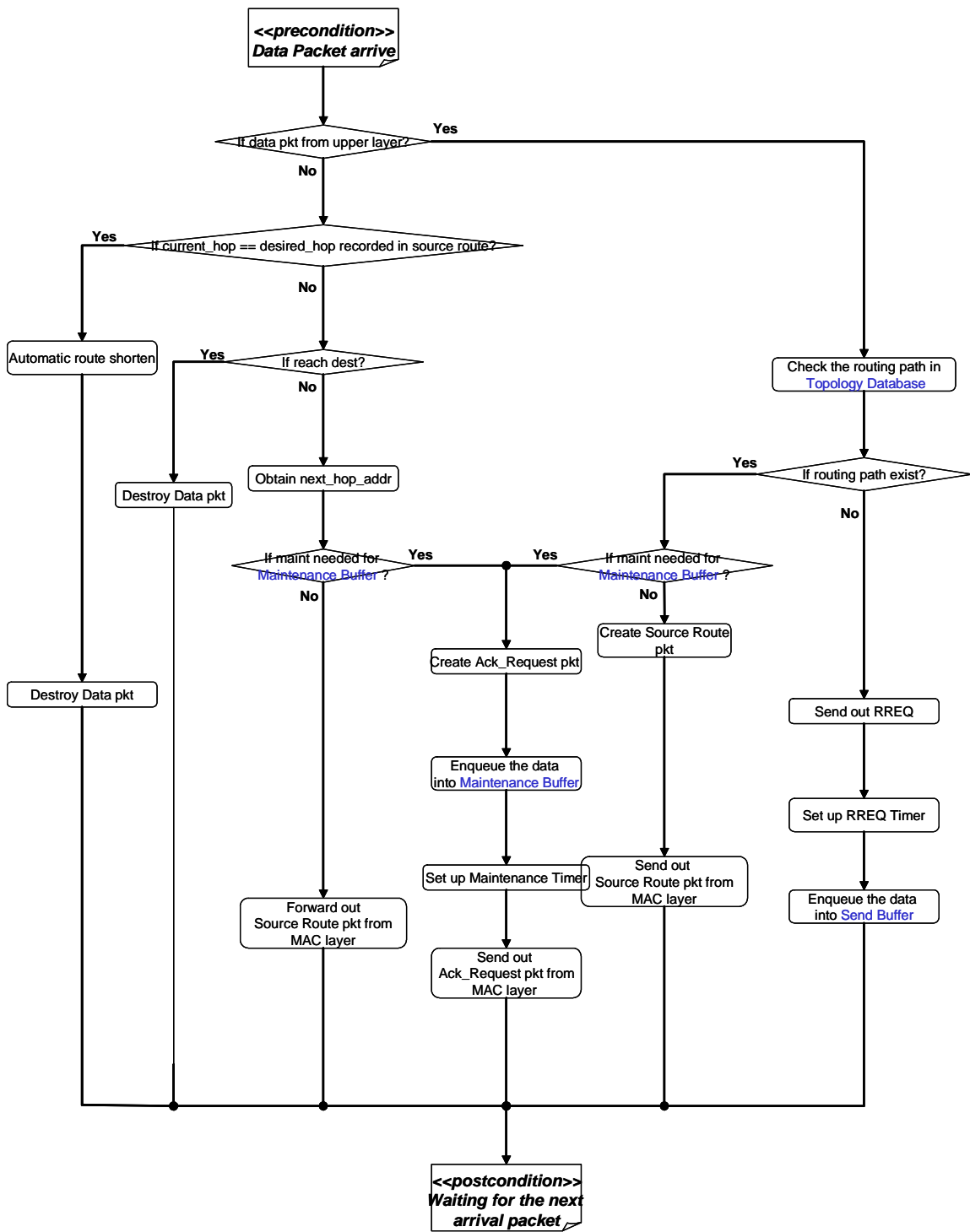


Figure 2.4 DSR Activity Flowchart for User: Data Pkt Receiver

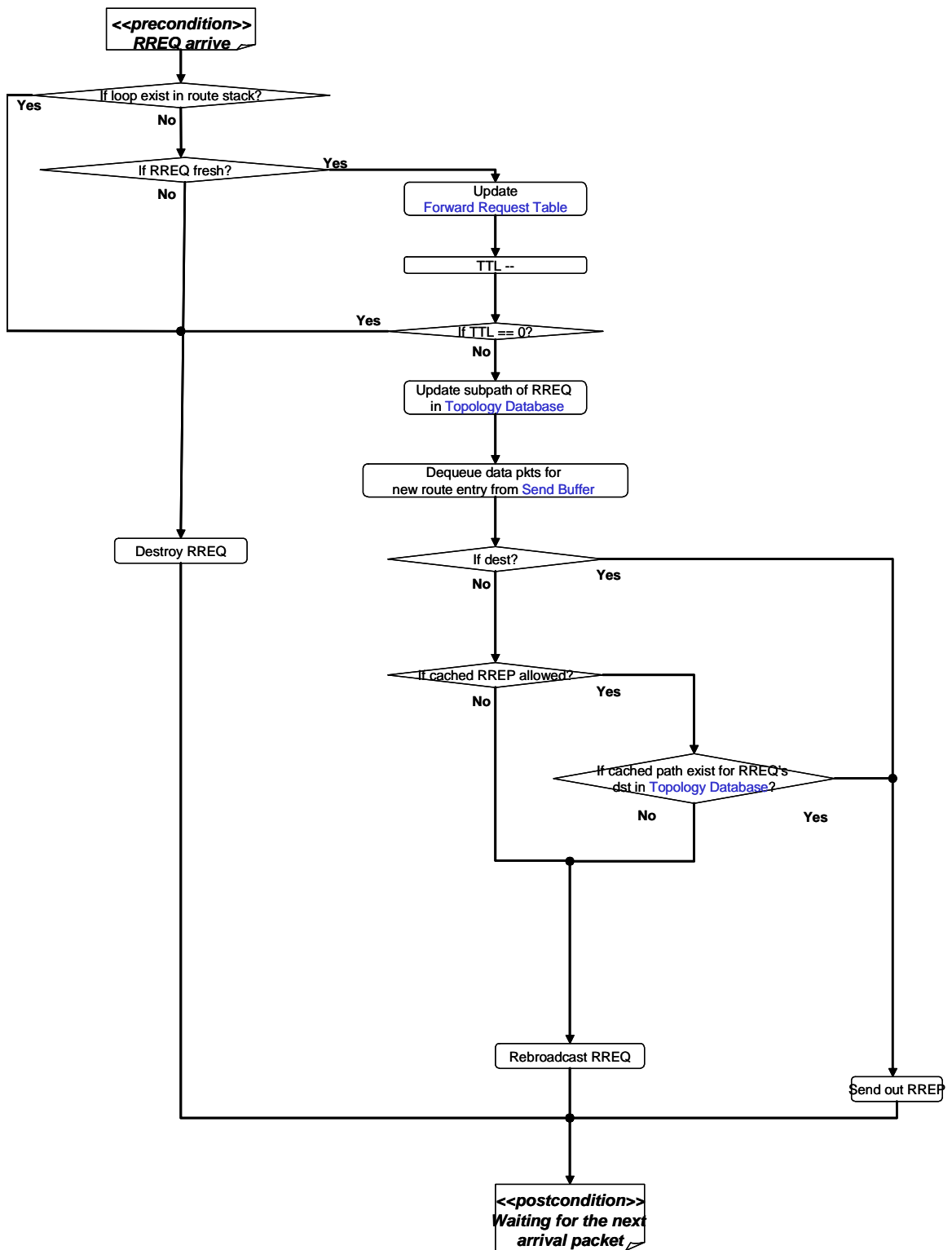


Figure 2.5 DSR Activity Flowchart for User: RREQ Pkt Receiver

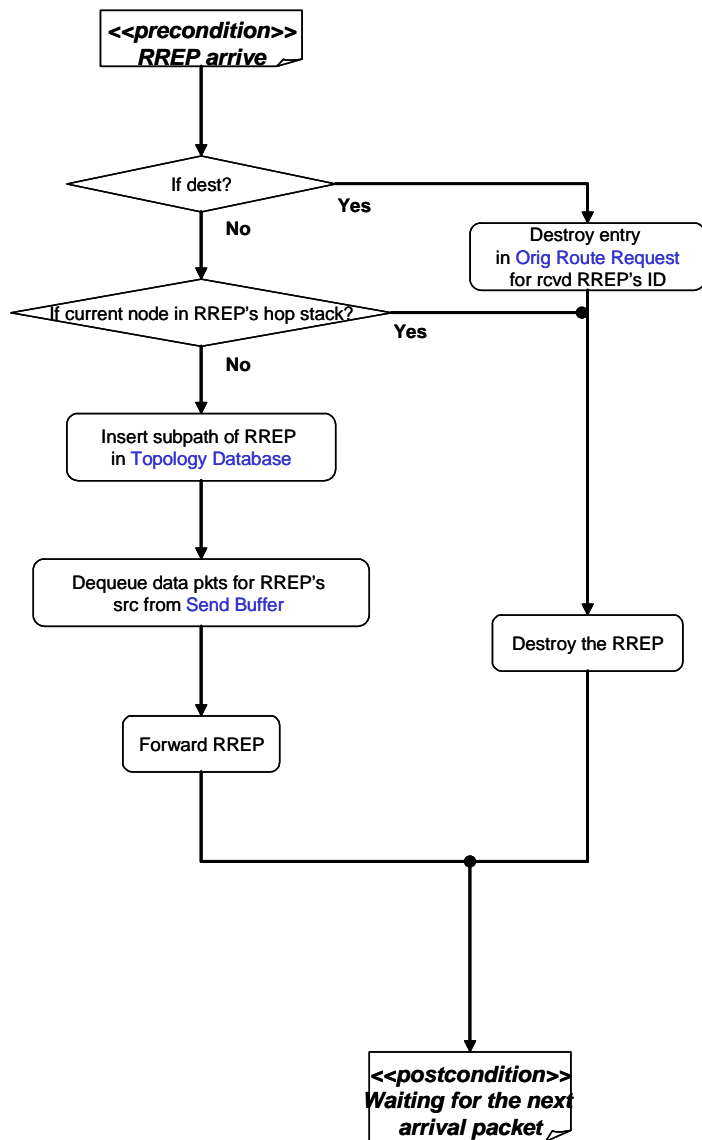


Figure 2.6 DSR Activity Flowchart for User: RREP Pkt Receiver

### **2.3. Other Protocols**

Besides AODV and DSR, Optimized Link Routing Protocol, Probabilistic Emergent Routing Protocol and Zone Routing Protocol are developed for different scenarios. A brief introduction is presented below.

#### **2.3.1. Optimized Link State Routing**

Optimized Link State Routing Protocol (OLSR) operates as a table driven, proactive protocol, i.e., exchanges topology information with other nodes of the network regularly [10]. Each node selects a subset of its neighbor nodes as “multipoint relays” (MPR). In OLSR, only nodes, which are selected as such MPRs, are responsible for forwarding control traffic, intended for diffusing information into the entire network. MPRs provide an efficient mechanism for flooding control traffic by reducing the number of routing overhead required. Each node will broadcast its neighbor list whenever its neighbors change. Through the network, each node will keep record of neighbor list of all other nodes. After collecting all link information throughout the network, each node will compute shortest path for all destination.

Compared to AODV and DSR, OLSR keeps cached routing path to all other nodes throughout the network, while AODV and DSR only maintenance cached routing path to subset of entire nodes. Additionally, OLSR not only implement periodical HELLO message but also broadcast neighbor list throughout network, this will lead to high routing overhead.



### **2.3.2. Probabilistic Emergent Routing**

This protocol is based on a type of learning algorithm, similar to AntNet [17], which provides natural resistance to the corruption of routing table entries. The algorithm uses Ant-like agents, *forward and backward ants*, as routing packets that explore available paths to a destination in the network through a number of nodes. These agents deposit pheromone or a “goodness value”, a probability distribution for taking a certain next hop to reach the destination. The agents reinforce good routes and reduce the goodness value of routes which do not perform well based on certain criteria; e.g. delay or number of hops. This is similar to the behavior of ants in nature when they forage for food as a colony. The communication between the ants is through the environment (pheromone, a chemical in this case). Global information, that is a complete route, emerges from local information, that is, the goodness value of a certain next hop for the packets destination.

### **2.3.3. Zone Routing Protocol**

This protocol divides the network into overlapping routing zones and runs independent protocols that work within and between the zones. The intra-zone protocol (IARP) operates within a zone and learns all the possible routes, proactively. So, all nodes within a zone know about their zone topology very well. The inter-zone protocol (IERP) is reactive, and a source node finds a destination node which is not located within the same zone by sending RREQ messages to all border nodes. This continues until the destination is found. ZPR is a hybrid routing protocol which runs

proactive protocol within a local range, and reactive routing protocols are implemented out of the local range.

## **2.4. Summary of Performance Analysis for Routing Protocols**

Along with the development of MANET routing protocols, the other fast developed topic is how to make performance analysis, and how to conduct performance comparison. Since it's difficult to give a mathematical description of routing protocol, most performance analysis is simulation based. Some general observations have been recognized:

- a. Reactive routing protocols will perform better than proactive protocols in highly dynamic network (Hsu, 2003).
- b. Among reactive protocols, AODV outperforms DSR in more stressful situations (with higher mobility speed, with higher traffic density) (Das, 2000).
- c. DSR consistently generates less routing traffic overhead than AODV (Choi 2004; Das, 2000).

Compared between reactive and proactive protocols, proactive usually generate more routing overhead than reactive routing protocols. Since proactive routing protocols have to send out routing controlling packets constantly to maintain an updated routing information all the time, the ratio of number of routing controlling packets to number of data packets will be much higher than reactive routing protocols. For reactive routing protocols, the routing controlling packets will be delayed until incoming data packets can't find routing information, then routing

controlling packets will be initiated and sent out. This delay in routing activities will lead to a less overhead for reactive routing protocols. However, since the routing topology in proactive routing protocols is updated constantly, data packets have a better chance to arrive the destination in proactive routing protocols. Therefore, the data packet loss ratio in proactive protocols will be less than reactive protocols.

However, reactive routing protocol is more adaptive to highly dynamic network. Since reactive routing protocol can update routing information faster than pro-active protocols. Especially, when communication stations move at high speed, the reactive routing protocols can quickly return routing path from source station to destination station. In pro-active routing protocols, each node will broadcast its neighbor list whenever its neighbors change. Through the network, each node will keep record of neighbor list of all other nodes. After collecting all link information throughout the network, each node will compute shortest path for all destination. It's time consuming to collect all link information throughout the network, and especially in highly dynamic network, the link information may be stale before routing path is calculated and become available for cache.

When compare between reactive routing protocols, AODV outperforms DSR in more stressful environments with higher mobility speed and higher traffic density. DSR has multiple routing paths cached in routing table for each destination, while AODV only use the earliest returned station sequence as routing path. Statistically, the earlier a routing path is returned, the less chance that the path will be broken when

data packet is passing through the path. Therefore, in AODV, less data packets will be lost than in DSR.

As mentioned in above, DSR caches multiple routing paths for a specified destination. Therefore, routing overhead will be much less in DSR than in AODV. Since a lot of route request packet can hit a cached path in intermediate stations. Concluded from above, usually the lower is the routing overhead, the higher is the data packet loss ratio. This is a rule of trade off that is recognized for MANET routing protocols. Furthermore, this rule of trade off between routing overhead and packet loss ratio is already approved by simulation tests.

When we introduce routing protocols above, route discovery and route maintenance is repeatedly mentioned. Actually, all routing protocols have four primary functions: path discovery, route maintenance, topology database maintenance and data packet forwarding. In order to organize all routing protocols by a generalized framework, we propose the concept of component, and will introduce a component based framework for decomposing various routing protocols in the next chapter.

As analyzed above, routing performance is closely affected by the four primary functions. So, it's extremely crucial to investigate four primary functions carefully.

Up to now, routing protocols are always studied as solid entities. The complexity and ambiguous structures make protocols difficult to be understood and implemented in real application. It's much easier to understand each routing protocol after defining them through a generic framework. In the next chapter, a framework for decomposing MANET routing protocol will be introduced. And, accordingly, component metrics

will be generated to facilitate studying of component performance and overall routing performance.

## **Chapter 3**

### **Framework of Component Based Routing Protocol**

#### **3.1. Architecture of Component Based Routing Protocol**

Component based routing protocol is a new paradigm for network modeling, designing and studying. In conventional research studies, routing protocols are always studied as a solid entity. However, for component based methods, we investigate component (unit of routing protocols) instead of entire routing protocols. The component based routing protocol ensures component functionality reusable in new and innovative contexts. If planned strategically, the component based routing protocol can be highly adaptive to various implementation scenarios. There are several challenges for the design of component based routing protocol. The first one is to define components and decompose routing protocols into systematic components. The second one is to evaluate performance level for each component in different contexts and make it a reference for component selection. The other challenges are how to translate performance requirement to selection of components and assembly isolated components together.

It's not straightforward to decouple routing protocols into well-defined components. This framework needs to be flexible enough to accommodate existing routing principles, functionalities and protocols, as well as new development in future. As such, we propose a framework for abstracting, identifying and organizing components

for ad hoc routing protocols. The hierarchical architecture of component based routing protocol is introduced below in three categories: definition of components, specification of subcomponents and information carriers:

*1. Definition of components:* Component is a fundamental abstraction that applies to many physical structures such as physical parts of machine, structures of a software objects. The concept of component proposed here is to define nonphysical objects: functionalities of routing protocols. No structures such as route cache table, neighbor list or packet format will be expressed by components.

Although different Ad Hoc routing protocol has different route discovery mechanism, some common constituent functionalities are shared among various Ad Hoc routing protocols. We synthesis the common functionalities and categorize them into four components:

*path discovery component (PD),*

*topology database maintenance component (TM),*

*route maintenance component (RM) and*

*data packet forwarding component (DF).*

Each component is well defined according to their functionality and cooperates with other components to achieve task of routing. In simple words, path discovery component can be applied to search for new routing path for data packets. The controlling packets such as RREQ and RREP will be implemented. Topology database maintenance component helps to update the connectivity information of neighboring stations. Controlling packets such as Hello and Hello\_Ack will be used.

The third component is Route maintenance component, when routing link or routing neighbor is lost, route maintenance component will help to delete stale routing link or routing neighbors. And, some route maintenance component will help to repair broken routing path. Route maintenance component will implement RERR packet. The fourth component is data packet forwarding component. This component determines how to forward data packet from source station to destination station. Under some situation, controlling packet of Ack-request and Ack will be implemented in data packet forwarding component.

2. *Specification of subcomponents:* subcomponents are component's methods, which are required to realize functional components of protocols. Subcomponents are components of components. The same components can be performed by different subcomponents. Such as to communicate a message, we can either email the message or just directly call someone. So, these are two different ways to achieve one target goal. Irreflexivity and antisymmetry are two primitive principles for definitions of subcomponents. Irreflexivity defines that a component can't be a subcomponent of itself. Antisymmetry states that two components can't be subcomponents of each other. A component can be executed by different methods. For instance, in Path Discovery Component, routing path can be searched on hop based method (AODV) or path based method (DSR). We define them as Hop\_based\_RREQ and Path\_based\_RREQ respectively.

3. *Information carrier: local database and packet formats.* Network topology information, local connectivity information and timing information are necessary for



execution of methods represented by subcomponents. All these information will be stored in some local database such as route cache table, local connectivity table and route request table. And the topology information is also exchanged between distributed hosts by disseminating some formats of packets. During operation, subcomponents will exchange information with local database and packet formats, and renew the information stored in these physical structures. For example, in reactive protocol AODV, the subcomponent Hello\_detection will update local connectivity list. If a neighbor can't be listened before an expiration of its lifetime, it will be assumed to be disappeared, and will be deleted from the local connectivity list. In addition, the last updating time will be recorded in order to count the lifetime of the neighbor. Hence, the physical structure is crucial to the execution of subcomponents. The detailed format and contents of routing protocol database will be decided by requirements of subcomponents in careful consideration of which information is needed by subcomponents, how to index the information and how to organize the information. Below local databases are associated with all reactive routing protocols: Route Cache Table, Route Request Table, and Send Buffer. The normal formats of packets implemented in MANET routing protocols are: RREQ, RREP and REER.

The hierarchical structure and their interaction are illustrated in the Figure 3.1. In order to illustrate the architecture of component based routing protocols, two protocols (DSR, AODV) will be translated into the architecture introduced in the above section.

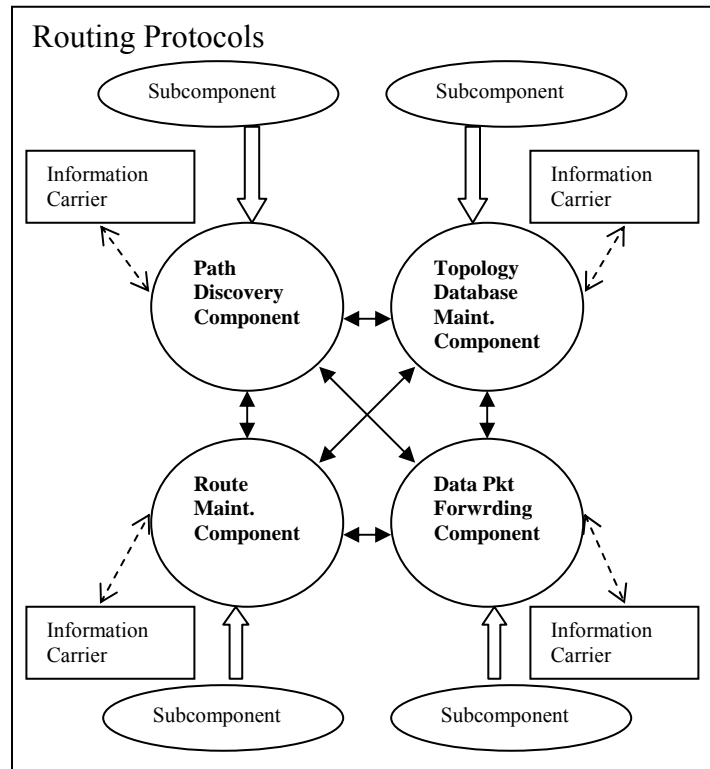


Figure 3.1: Architecture of Component Based Routing

## 3.2. Framework for Component Based Routing Protocol

### 3.2.1. Path Discovery Component

All functional methods related to searching path from a source to a desired destination by RREQ, RREP message or by link state advertisement are defined as path discovery component.

For reactive protocols such as DSR and AODV, a route discovery process will be initiated when a data packet can't obtain a route entry to a destination in the route

cache table. The source node will create RREQ (route request) packet, and broadcast it. All nodes received the RREQ will check if they have a route cached to the desired destination. If not, the RREQ will be rebroadcast; otherwise, the RREP will be created and sent back to the source node. Both DSR and AODV will create RREQ and RREP packets to disseminate routing information during path discovery process. However, they have some detailed discrepancies for path discovery component. The major differences are:

1. AODV has hop-based path discovery mechanism. This means that only next routing hop will be recorded in route cache table. While for DSR, a complete sequence of hops for the routing to a destination will be recorded in route cache table. Hence, compared to the hop-based routing protocol AODV, DSR is a path-based routing protocol.

2. AODV only record single routing entry (next routing hop) for a destination. In contrast, DSR can record multiple paths for a destination. Consequently, AODV only keep the routing information from the first RREQ packet that reaches target station, and destroy other late arriving RREQs. However, whenever a RREQ packet is received at a destination in DSR, it will initiate a RREP.

### **3.2.2. Topology Database Maintenance Component**

All functional methods determining the link state for local neighbors are defined as topology database maintenance component.

MANET is a mobile network with highly dynamic topology, and mobile station can only communicate to neighboring mobile stations within transmission range. A mechanism is required to detect local topology change to update a station's neighbor list. AODV can detect link state within 1 hop range by broadcasts Hello message with TTL = 1. If a mobile station can't hear Hello message from its neighbor before an expiring timeout, the neighbor will be deleted from neighbor list and route maintenance will be initiated as well. For DSR, no proactive control message like Hello will be adopted to detect link failure. A receiver on routing path will broadcast acknowledgment after receiving data packet. If no acknowledgement is received by sender of data packet before an expiring timeout, the link between sender and receiver is assumed to be broken, and route maintenance will be invoked. Hence, topology database maintenance is performed reactively in DSR to detect link state within 1 hop range.

Basically, topology database maintenance component is consists of mechanisms to detect link failure for neighboring stations. Being incurred by topology database maintenance component, route maintenance component is a consequent action of topology database maintenance component. For instance, in AODV, if Hello message isn't detected from a neighbor, the neighbor will be deleted from neighboring list, and RERR message will be created and broadcast, thus a route maintenance component will be put in action.

### **3.2.3. Route Maintenance Component**

Route maintenance component's main function is to propagate the information of a broken link once it's detected by Topology Database Maintenance Component, and to delete cached routing paths containing the broken links.

For reactive routing protocols, route maintenance component is performed independent of route discovery component. RERR message will be created after broken link is detected. The difference is that AODV will encapsulate list of unreachable destinations in RERR packet, while DSR will encapsulate the broken link presented by "Error Source Address" and "Error Destination Address". In addition, upon receiving RERR messages, AODV will delete all routing entries indexed by unreachable destinations recorded in RERR, and DSR will remove all cached paths containing the broken link.

### **3.2.4. Data Packet Forwarding Component**

Data packet forwarding component performs tasks of relaying data packets from source station to destination station by utilizing route cache table.

For hop based routing protocol, data packet doesn't record any routing information in it. Each intermediate node has to search its route cache table to determine to which neighboring station the data packet should be forwarded. For path based routing protocol, the complete sequence of nodes on routing path will be recorded in data packets, and the next routing hop can be decoded from data packet without searching intermediate stations' route cache table.

### **3.3. Specification of subcomponents**

As an object, component requires various methods to implement some functionality. To illustrate this, we'll take an example of one method from path discovery component. Path discovery component need to decide how many hops RREQ can be forwarded. This is performed differently in DSR and AODV. In DSR, TTL value is set before operation, and can't be changed during operation. However, expanding ring methods is executed in AODV. In AODV, after each expiring timeout, the source station will initiate a new RREQ to the target station. The TTL value will be incremented by a predefined incremental value in each retransmission. In this way, TTL value will be increased step by step during each retransmission. The DSR's static TTL value and AODV's dynamic TTL value are defined as Fixed\_ring and Expanding\_ring respectively.

Subcomponents are sub-functionalities of components that can facilitate components to accomplish one task. Subcomponents are components of components. Subcomponents cooperate closely to each other to implement a component. To instantiate subcomponents, we have abstracted and synthesized subcomponents for two routing protocols: AODV and DSR. Subcomponent's repository is defined in terms of methods implemented in each component.

#### **A. subcomponents for AODV:**

- For path discovery component, the functional subcomponents are:

1. Expanding\_ring: RREQ's TTL value will be incremented by a TTL\_increment in each retransmission of RREQ.
2. Hop\_based\_RREQ: Next routing hop will be recorded in route cache table. Intermediate nodes on path of RREQ won't be recorded in RREQ packet.
3. Cached\_RREP: Not only target station receiving RREQ can initiate RREP, but also intermediate node which has route cached to the target station can generate RREP packet.
4. Single\_RREP: Only the earliest RREQ arriving at target station will initiate RREP.
5. Single\_route\_cache: For each destination, only a routing neighbor is stored.

$sub_{PD}^{AODV} = \langle \text{Expanding\_ring}, \text{Single\_RREP} \rangle$

- For topology database maintenance component, subcomponents are:
  1. Hello\_detection: periodically broadcast beacon message to notify neighboring station the existence of the station.
  2. Local\_connectivity\_update: utilize overheard packet to update local connectivity table.

$sub_{TM}^{AODV} = \langle \text{Hello\_detection}, \text{Local\_connectivity\_update} \rangle$

- For route maintenance component, the subcomponents are:
  1. Local\_repair: Intermediate node on data packet's routing path can invoke route discovery procedure.

2. Route\_error\_disseminate: broadcast unreachable destinations throughout network, and remove routing entry indexed by the unreachable destinations.

$sub_{RM}^{AODV} = \langle Local\_repair, Route\_error\_disseminate \rangle$

- For data packet forwarding component, the subcomponents are:

1. Hop\_based: Look up next routing hop in each intermediate node's route cache table.

2. Unsolicited\_forwarding: forward data packet without waiting for acknowledgement from the receiver.

$sub_{DF}^{AODV} = \langle Hop\_based, Unsolicited\_forwarding \rangle$

## **B. subcomponents for DSR:**

- For path discovery component, the functional subcomponents are:

1. Fixed\_ring: TTL value keeps unchanged for each retransmission of RREQ.

2. Path\_based\_RREQ: hops traversed by RREQ are encapsulated into packet. Sequence of nodes on the path is stored in route cache table.

3. Cached\_RREP: Intermediate station on RREQ's path can initiate RREP packet if routing entry to target station can be obtained from its route cache table.

4. Multiple\_RREP: All RREQ reaching target stations will create RREP.

5. Multiple\_route\_cache: For each destination, multiple routing paths can be cached.

$sub_{PD}^{DSR} = \langle Fixed\_ring, Multiple\_RREP \rangle$



- For topology database maintenance component, subcomponents are:

Here is no method for topology database maintenance.

$$sub_{TM}^{DSR} = \langle \rangle$$

- For route maintenance component, the subcomponents are:

1. Packet\_slavage: In multipath route cache table, another valid path can substitute a broken path.
2. Route\_error\_disseminate: Propagate information of broken link throughout network, and delete routing entry containing the broken link.

$$sub_{RM}^{DSR} = \langle Packet\_salvage, Route\_error\_disseminate \rangle$$

- For data packet forwarding component, the subcomponents are:

1. SourcePath\_based: Data packet follows the sequence of hops duplicated from source node, and don't look up route cache tables at intermediate nodes.
2. Solicited\_forwarding: Each sender of data packet (source node and intermediate node) will wait for acknowledgement from receiving station, and make a copy of the data packet in maintenance buffer. The data packet will be retransmitted without receiving acknowledgement before an expiring timeout.

$$sub_{DF}^{DSR} = \langle SourcePath\_based, Solicited\_forwarding \rangle$$

### 3.4. Identification of Information Carrier

Physical structure of *information carrier* presented here is comprised of two types of entities: local database and packet formats. Network topology information, local connectivity information and timing information are necessary for execution of methods represented by subcomponents. All these information will be stored in some local database such as route cache table, local connectivity table and route request table. During operation, subcomponents will exchange information with local database and packet formats, and renew the information stored in these physical structures. For example, in reactive protocol AODV, the subcomponent Hello\_detection will update local connectivity list. If a neighbor can't be listened before an expiration of its lifetime, it will be assumed to be disappeared, and will be deleted from the local connectivity list. In addition, the last updating time will be recorded in order to count the lifetime of the neighbor. Hence, the physical structure is crucial to the execution of subcomponents. The detailed format and contents of local database will be decided by requirements of subcomponents in careful consideration of which information is needed by subcomponents, how to index the information and how to organize the information. Below local databases are associated with all reactive routing protocols.

*Route cache table* is required by all Ad Hoc routing protocols with different storage of sequence of nodes on path for path based routing (DSR) and storage of next routing hop for hop based routing (AODV). And the latest updating time of a route entry is recorded to calculate expiration time for the route entry.

*Route request table* is necessary to record RREQ packet initiated by a source station or bypassed RREQ packets. The RREQ packets will be indexed by 2-tuple <source address, sequence number>.

*Send buffer* is storage for data packets which are waiting for RREP packet returned from path discovery process. The waiting time of data packets are recorded and compared to expiring timeout of route discovery to decide if to initiate a new RREQ. Besides above local databases, different reactive routing protocol will have unique local database structures for special application. AODV employs *local connectivity table*, and DSR adopts *maintenance table*. For AODV's topology database maintenance component, the subcomponent of Hello\_detection is implemented to detect change of neighborhoods. Upon being detected by listening a HELLO, a new neighbor station will be recorded to a local connectivity table associated with its detection time. For DSR's topology database maintenance component, the subcomponent of Solicited\_connectivity\_checking is utilized to detect failure of link to a routing neighbor. Before a data packet is forwarded to its routing neighbor, a duplication of the data packet is stored in the maintenance table. If an acknowledgement from the routing neighbor can't be received before a timeout, the data packet will be retransmitted again until either the upper limitation of retransmission is reached or an acknowledgment is received, whichever happens earlier.

The other type of entity of physical structure is packet formats. MAENT routing protocols employ many types of routing control packets to disseminate and exchange

link status information among different stations. For instance of reactive routing protocols (AODV and DSR), during path discovery procedure, RREQ and RREP packets are required to carry information (address and sequence number) about source station and target station. Each intermediate station on routing path will check out RREQ packet's destination address and sequence number to verify if the RREQ packet was received before, and if the current RREQ packet is the latest request issued by the source station. The RREQ packet will be destroyed at the intermediate stations in case that either the RREQ is a duplication of an old RREQ received before or the RREQ is an obsolete request with a sequence number less than what recorded in the route request table. The above example illustrates that control packets are required to interact with local databases to exchange information and perform various subcomponents. The packet's formats are decided by tasks performed. In normal cases, RREQ and RREP packets are issued by stations that perform path discovery component, RREP packets are generated by stations that execute route maintenance component, and beacon packet (HELLO) are created by stations that implement topology database maintenance. Additionally, for DSR, ACK\_REQ and ACK packets are employed by topology database maintenance component. In DSR, before forwarding a data packet from a sending station to a routing neighbor, the data packet is first copied to a maintenance buffer, and ACK\_REQ message will be encapsulated into the data packet. Upon receiving the ACK\_REQ message, the routing neighbor will reply an ACK packet to the sender. After receiving ACK packet, the sender of the data packet will delete the copy of the data packet from the maintenance buffer.

Otherwise, if ACK packet can't be received by the sender before expiration, then the link will be assumed to be broken, and a new copy of the data packet will be delivered again to the routing neighbor, and ACK\_REQ will be encapsulated into the data packet. If ACK can't be received before the expiration, one more copy of the data packet and ACK\_REQ will be sent out again, until the maximum allowed number of repeated delivery is reached, then the copy of data packet will be deleted from the maintenance buffer, and RERR message will be initiated.

As shown in the above, the packet format and local database interact to each other to implement subcomponents. The information will be exchanged between packet format and local database, so that the decision can be made by node such as if the copy of the data packet should be deleted from maintenance buffer or not, or if the maximum number of re-delivery is reached. Along with making these decisions, the routing components accomplish its task.

### **3.5. Component Performance Metrics**

Performance metrics (latency, routing overhead and packet loss ratio) for overall assessment of routing protocols are not suitable to explain and compare components' performance. How to trace and compare each component's performance during execution of routing protocols? The components metrics are distinguished from overall metrics by its emphasis on evaluating merits of specified components instead of effectiveness of overall routing protocols. As results of whole routing protocols,

data packets are moved from source station to destination station. The overall performance metrics put emphasis on effectiveness of routing protocols. The popular overall performance metrics are latency, throughput and packet loss ratio. In respect to the effectiveness of routing protocols, overall metrics can only explain results of routing protocols. Overall metrics can only evaluate the effects of combined components' performance. Components are functional part of routing protocol, and more detailed component metrics should be designed to better assess execution of functional components. In addition, meaningful component metrics enables us to better understand the interaction between performance of components and overall performance. The following is a list of proposed meaningful metrics for each component.

### **3.5.1. Metrics for Path Discovery Component**

1.1. Percentage of Path Discovery Success:  $\#RREQ \text{ Replied} / \#RREQ \text{ Initialized}$ .

It's an assessment of effectiveness of path discovery component. Higher is the rate, more efficient is the activity of path discovery.

1.2. Path Discovery Inefficiency Factor:  $\#Total \text{ Path Discovery Traffic Rcvd} / \#RREQ \text{ generated}$ . The less is the value, more efficient is the path discovery.

1.3. Percentage of Route Cache Hit for Data Packet:  $\#Cache \text{ Hit Data Packet from High Layer} / \#Total \text{ Data Packet from High Layer}$ .

This is the measure of effectiveness of path discovery component with respect to its capability of obtaining bypass paths. Bypass paths are by results of path discovery

components, which are the paths captured during path discovery that lead to destination stations besides target stations.

1.4. Percentage of Cached RREP:  $\frac{\# \text{Cached RREP Generated}}{\# \text{Total RREP Generated}}$ .

It's an assessment for efficiency of subcomponent of Cached\_RREP.

1.5. Average Delay for Path Discovery:  $\frac{\text{Accumulated Path Discovery Delay}}{\# \text{RREQ Replied}}$ . It's a measure of latency of path discovery component. How fast can path discovery component be completed on average.

1.6. Path Discovery Control Overhead:  $\frac{\# \text{Total Path Discovery Traffic Rcvd}}{\# \text{Total Data Packet from High Layer}}$ .

This is a measure of controlling overhead for each data packet generated in path discovery component.

### **3.5.2. Metrics for Route Maintenance Component**

2.1. Percentage of Data Packet Reaching Destination Aided by Route Maintenance:  $\frac{\# \text{Data Packet Reaching Destination Aided by Route Maintenance}}{\# \text{Data Packet Reaching Destination}}$ .

It's a measure of importance of route maintenance during path repairing.

2.2. Average Overhead of Route Maintenance:

$\frac{\# \text{Total Control Traffic Introduced by Route Maintenance}}{\# \text{Data Packet Reaching Destination Aided by Route Maintenance}}$ . This assesses controlling overhead introduced by route maintenance.

2.3. Percentage of Route Maintenance Success:  $\frac{\text{\#Data Packet Reaching Destination Aided by Route Maintenance}}{\text{\#Data Packet Attempting Route Maintenance}}$ .

It measures efficiency of route maintenance.

### **3.5.3. Metrics for Data Packet Forwarding Component**

3.1. Percentage of Forwarding Failure:

$\frac{\text{\#Data Packet Forwarding Failure between Hops}}{\text{\#Data Packet Forwarding between Hops}}$ .

This measures efficiency of data packet forwarding. Lower is the rate, better is the performance of data packet forwarding.

3.2. Average End to End Delay for Packet Forwarding:

$\frac{\text{Accumulated End to End Delay}}{\text{\#End to End Data Packet Forwarding}}$ .

It's a measure of latency of data packet delivery.

### **3.5.4. Metrics for Topology Database Maintenance Component**

4.1. Overhead of Topology Database Maintenance:

$\frac{\text{\# Total Control Packet Traffic Introduced by Topology Database Maintenance}}{\text{\#Data Packet Reaching Destination}}$ . This measures controlling overhead introduced by topology database maintenance component.

As shown in the above section, the component performance metrics can be categorized into four parts according to component differentiation. This categorization is crucial in the process of investigating how each component works as an entity. In order to track and understand how each component perform, the



component performance metrics are extremely helpful, since the component performance metrics provide more insightful information for studying the component's performance than overall performance metrics. As stated previously, the overall performance metrics can only evaluate the overall performance, which is a result of routing protocols. Components are functional part of routing protocol, and more detailed component performance metrics can assist researchers to better assess execution of functional components.

In constructing the component dependence network, we need a hierarchical structure between overall performance metrics and component performance metrics. The selection of component performance metrics is crucial to structural learning of component dependence network. If too many component metrics exists in the dependence network, then too many calculations needed to be done in order to determine the structure of dependence network. However, if relevant components metrics is not included, the dependence network is not insightful enough to disclose the interaction between component and overall performance metrics. Hence, in the following, the suggested component metrics are listed for each overall performance metrics. The real structure learning for component dependence network will be benefited to this referred component selection. This component categorization is critical in helping to maintain balance between reasonable calculation work and an insightful component dependence network.

### **1. Component metrics relevant to Packet Loss Ratio:**

1.1 Percentage of Path Discovery Success: #RREQ Replied / #RREQ Initialized.

1.2 Percentage of Route Cache Hit for Data Packet:  $\frac{\#Cache\ Hit\ Data\ Packet\ from\ High\ Layer}{\#Total\ Data\ Packet\ from\ High\ Layer}$ .

1.3 Percentage of Cached RREP:  $\frac{\#Cached\ RREP\ Generated}{\#Total\ RREP\ Generated}$ .

1.4 Percentage of Data Packet Reaching Destination Aided by Route Maintenance:  $\frac{\#Data\ Packet\ Reaching\ Destination\ Aided\ by\ Route\ Maintenance}{\#Data\ Packet\ Reaching\ Destination}$ .

1.5 Percentage of Route Maintenance Success:  $\frac{\#Data\ Packet\ Reaching\ Destination\ Aided\ by\ Route\ Maintenance}{\#Data\ Packet\ Attempting\ Route\ Maintenance}$ .

1.6 Percentage of Forwarding Failure:

$\frac{\#Data\ Packet\ Forwarding\ Failure\ between\ Hops}{\#Data\ Packet\ Forwarding\ between\ Hops}$ .

## **2. Component metrics relevant to End to End Delay of Packet Delivery:**

2.1 Average Delay for Path Discovery:  $\frac{Accumulated\ Path\ Discovery\ Delay}{\#RREQ\ Replied}$ .

2.2 Average End to End Delay for Packet Forwarding:  $\frac{Accumulated\ End\ to\ End\ Delay}{\#End\ to\ End\ Data\ Packet\ Forwarding}$ .

2.3 Percentage of Route Cache Hit for Data Packet:  $\frac{\#Cache\ Hit\ Data\ Packet\ from\ High\ Layer}{\#Total\ Data\ Packet\ from\ High\ Layer}$ .

## **3. Component metrics relevant to Routing Overhead:**

3.1 Path Discovery Inefficiency Factor:  $\# \text{Total Path Discovery Traffic Rcvd} / \# \text{RREQ}$  generated

3.2 Path Discovery Control Overhead:  $\# \text{Total Path Discovery Traffic Rcvd} / \# \text{Total Data Packet from High Layer}$ .

3.3 Average Overhead of Route Maintenance:

$\# \text{Total Control Traffic Introduced by Route Maintenance} / \# \text{Data Packet Reaching Destination Aided by Route Maintenance}$ .

3.4 Overhead of Topology Database Maintenance:

$\# \text{Total Control Packet Traffic Introduced by Topology Database Maintenance} / \# \text{Data Packet Reaching Destination}$ .

In the following sequel, we will show one sampling component dependence network that is applied to study the packet loss ratio.

### **3.6. Interpretation of Routing Performance by Component Metrics**

Extensive research has been done to compare and explain the differences of various routing protocols at the protocol level through simulation or intuitive analysis [13] [14] [15]. This helped us to evaluate protocols according to their performance and distinguish the pros and cons of each protocol. But more insightful explanation from component level remains unavailable. The conventional metrics for routing protocols are end-to-end delay, packet loss ratio and routing traffic overhead. Statistics of these metrics can only demonstrate if one routing protocol performs well or not, without showing reasons of why the protocol performs well or not. Therefore, each building

block needs to be tracked separately to make conclusions of how each building block affect the overall routing protocol's performance.

Overall routing performance can be explained by component performance. In concern of overall performance metric of Goodput, the largest proportion of routing controlling overhead is usually generated by Path Discovery Component. Being illustrated by component metrics, higher is the "Percentage of Route Cache Hit for Data Packet" and "Percentage of Cached RREP", lower is the routing controlling overhead generated by path discovery component. Besides explained by simulation results, this can be explained intuitively by routing mechanisms. When data packet needs routing path or routing neighbor, it always first checks if any cached routing information is available. If no cached routing is available, then RREQ will be initiated, and sent out to target destinations. Furthermore, when broadcasted to destination, if a cached routing path or routing neighbor can be found on some intermediate stations, then RREP will be created and sent back to source station. Hence, more cached routing information is available in source nodes or intermediate nodes, less RREQ will be broadcasted to the destination, then less routing controlling overhead will be generated. Agreed with this intuitive analysis, below are some simulation results.

All simulations in this study are performed by OPNET. 20 mobile stations moved in a field of 2km x 2km. The mobility model is Random Way Point. Each node selects a moving direction randomly, and stops for 1 second after each moving step. The simulation will study performance at different mobility speed of 0, 10, 20, 30, 40

and 50 meters per second. The Traffic model implemented is the raw packet generation provided by OPNET. Each station chooses a random destination to send out data packet at a “Packet Inter-Arrival Time” which is an exponential probability function changes as exponential (1). The packet size is set to exponential (1024).

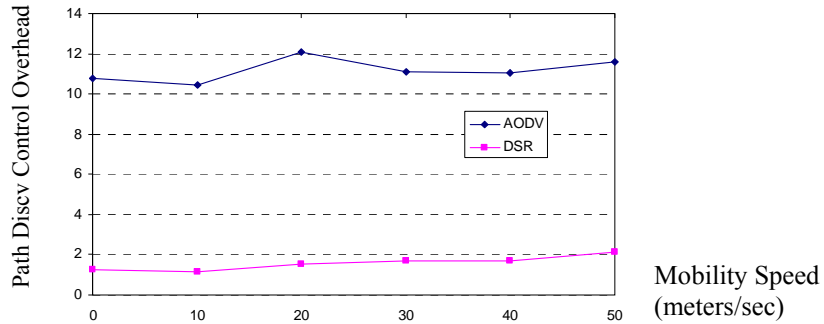


Figure 3.2 Simulation result for Path Discovery Overhead

From above simulation results, three observations can be achieved:

1. Percentage of Cached RREP:  $\frac{\# \text{Cached RREP Generated}}{\# \text{Total RREP}}$

Figure 3.3 shows that in DSR, a higher proportion of RREP is generated by cached routing information at intermediate nodes than in AODV. This property is decided by subcomponent of AODV and DSR. As described in chapter 3, for path discovery component, DSR has subcomponents of Multiple\_RREP and Multiple\_route\_cache. These two methods provide DSR more plentiful route cache than AODV, which has subcomponents of Single\_RREP and Single\_route\_cache. Thus, a RREQ in DSR has

higher chance than in AODV to find route cache to target stations at intermediate stations.

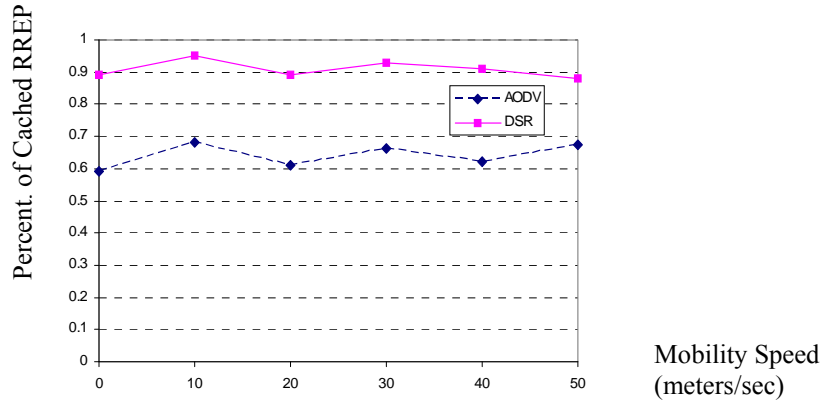


Figure 3.3 Simulation result for Percentage of Cached RREP

2. Percentage of Path Discovery Success: #RREQ Replied / #RREQ Initialized

Figure 3.4 shows that DSR has higher ratio of RREQ that is replied. After broadcasting RREQ, the initiator of RREQ will wait for RREP.

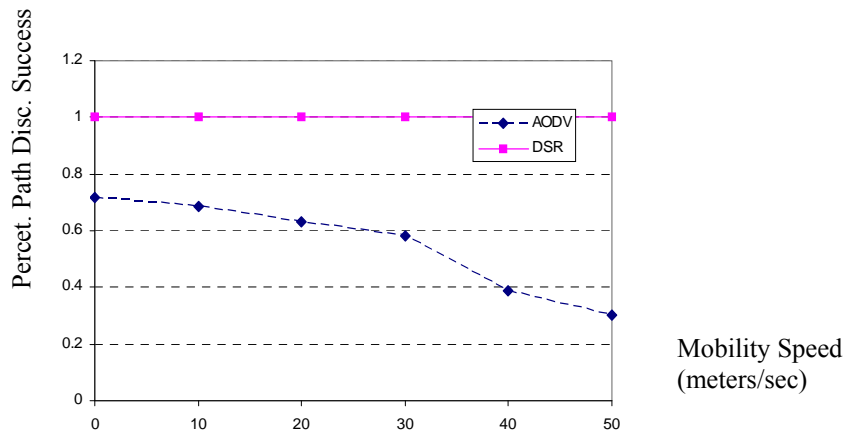


Figure 3.4 Simulation result for percentage of Path Discovery Success

In some situations, no reply will be returned to the initiator of RREQ. Normally, more route cache at intermediate nodes, higher probability of RREQ being replied.

3. Percentage of Route Cache Hit for Data Packet:  $\frac{\# \text{Cache Hit Data Packet from High Layer}}{\# \text{Total Data Packet from High Layer}}$

Figure 3.5 shows that data packet have higher cache hit ratio in DSR than that in AODV. And the difference between AODV and DSR will be greater in more stressful situation with larger mobility speed. This is also attributed to subcomponents of DSR: Multiple\_RREP and Multiple\_route\_cache. These subcomponents enrich route cache in DSR significantly. Therefore, higher chance for data packet to find an available route to its desired destination at route cache table at intermediate stations.

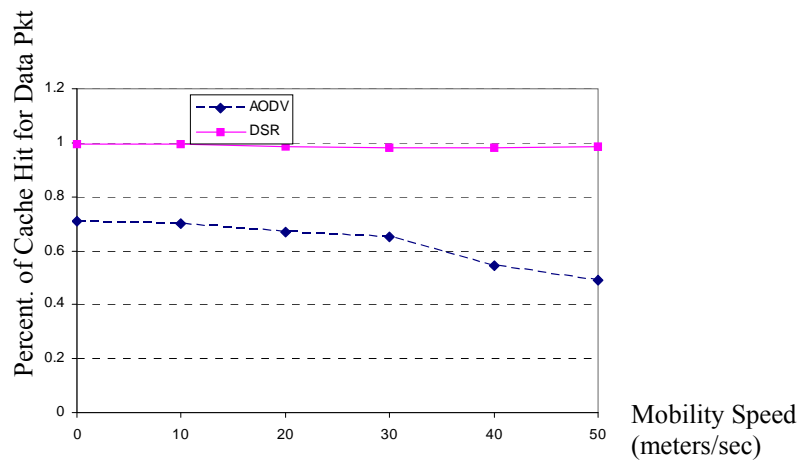


Figure 3.5 Simulation result for percentage of cache hit for data packet

This is a good example to show how the overall routing performance can be related to and interpreted by component performance metrics. By analyzing routing mechanism,

the overall routing performance can be explained well. However, we still need a quantitative explanation between overall routing performance and component performance. For a protocol engineer who is very familiar with the routing protocols and their performance, it's not difficult to name several impacts of components on overall routing performance. However, if for a person who has no experience in routing protocols, finding out root source for a deteriorated routing performance is a very tough job. The following chapters will introduce a mathematical tool that can facilitate the quantitative analysis for routing performance analysis. By statistically analyzing the simulation results, the mathematical model can automatically decoding the relationship between components and routing performance. And, the algorithm behind the mathematical model is easy for implementation.



## **Chapter 4**

### **Bayesian Induction of**

### **Components' Probabilistic Dependence**

Bottleneck components will be harmful to overall performance of routing protocols. We firstly present concept of “significant component” in the proposal. During previous analysis of simulation results for routing performance, three major metrics are applied to compare different routing protocols: end-to-end delay, packet loss ratio and routing control overhead. These metrics can't explain how this performance result is generated. Routing protocol can be understood as a system in which four components: path discovery component, topology database maintenance component, route maintenance component and data packet forwarding. We believe that component performance will have effects on overall routing protocol's performance. Additionally, different simulation inputs (network topology, mobility mode and traffic mode) will affect each component's performance. Our ultimate objective is to find out how each component performs under different scenarios and how each component affects overall routing performance? This can supply major reference for component selections.

## 4.1. Differential Performance Projection

The *significant component* is defined as the component which has the most contribution to the overall performance. Routing protocol is considered as a system of components and sub-components. The routing protocol's performance is an accumulated effect of components' performance. This accumulating effect can be assessed quantitatively by performance metrics. How to evaluate each component's contribution to the value of performance metrics? Performance projection is presented here to resolve the problem.

In component based routing protocol designing methodology, we will project each overall performance metric onto performance metrics of contributing components. For instance, the overall routing controlling overhead ( $O_{OVERALL}$ ) can be contributed to path discovery component ( $O_{PD}$ ), route maintenance component ( $O_{RM}$ ) and topology database maintenance component ( $O_{TM}$ ).

Thus:

$$O_{OVERALL} = O_{PD} + O_{RM} + O_{TM}$$

For overall data packet loss, it can be decomposed into data packet loss due to failure of path discovery component ( $L_{PD}$ ) and failure of data packet forwarding ( $L_{DF}$ ).

$$L_{OVERALL} = L_{PD} + L_{DF}$$

For end to end delay ( $D_{OVERALL}$ ), it comprises time consumed by path discovery component ( $D_{PD}$ ), data packet forwarding component ( $D_{DF}$ ), and possibly by route maintenance component ( $D_{RM}$ ).

$$D_{OVERALL} = D_{PD} + D_{DF} + D_{RM}$$

Some simulation results will demonstrate how each component's contribution to end-end delay is changed by simulation scenarios.

Below simulations are performed by OPNET. 20 mobile stations moved in a field of 2km x 2km. The mobility model is Random Way Point. Each node selects moving direction randomly at each step without stopping anytime. The simulation will study performance at different mobility speed of 0, 15, 30 meters per second. The Traffic model implemented is Data Traffic (12,000 bytes/sec), Voice Traffic (57,000 bytes) and Video Traffic (698,000 bytes/sec). Routing protocols is AODV.

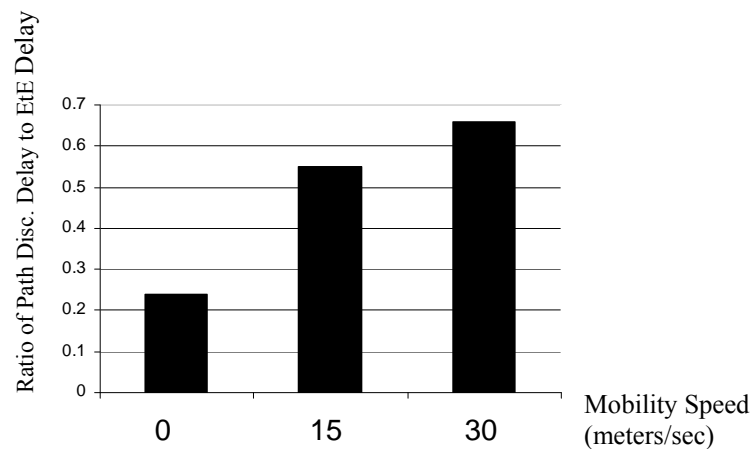


Figure4.1 Simulation Result for Video Traffic

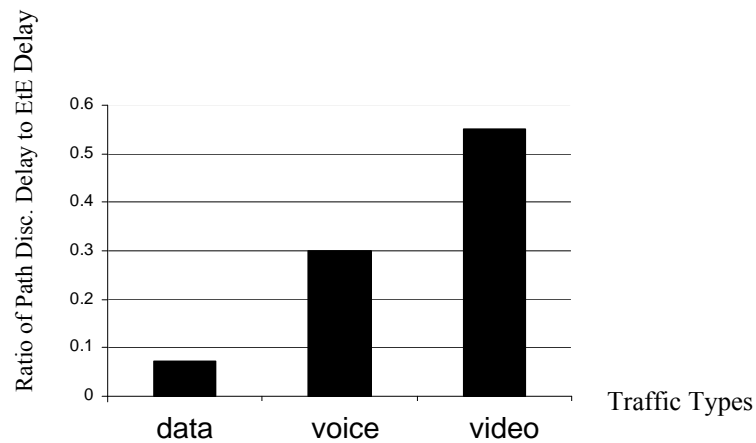


Figure4.2 Simulation Result for Mobility Speed 15

As shown in Figure 4.1, when mobility speed is increased, Path Discovery Delay has higher proportion of End to End Delay. In Figure 4.2, at the same mobility speed, higher is data traffic density, higher is the ratio of Path Discovery Delay to End to End Delay. Therefore, for high mobility speed network and high traffic density network, path discovery component is a significant component which needs to be investigated carefully to obtain possible improvement schemes for decreasing EtE delay, since path discovery delay has major portion of EtE delay for high mobility speed and high traffic density networks. From simulation results, performance projection is very effective for locating significant components. For different scenario inputs, different component have to be improved for overall performance's improvement.

## **4.2. Detection of Significant Component by Bayesian Belief Network**

As demonstrated in previous sections, components of routing blocks can be decomposed and evaluated separately. However, the most crucial problem to be resolved is how each component affects routing protocol's performance. For example, we hope to understand whether or not a path discovery component is a major reason for high data packet loss ratio for a specified scenario environment, and to what degree. Bayesian network is a good modeling tool with the ability to extract such casual relationships among different components. The component that has the most significant effect on a specified routing protocol's performance metric (end to end delay, routing overhead and data packet loss ratio) is defined as the significant component, and can be modified or replaced to achieve improvement for overall routing protocol's performance.

To discern inefficient routing component, component metrics have been defined, and will be applied during simulation to monitor component's performance. The performance metrics for overall routing protocols can be differentiated into component metrics. For Instance, the end-end delay comprises of path discovery delay, data packet forwarding delay and possible route maintenance delay. Each component has to be tracked to decide which component of routing protocols is the significant component that affects overall routing performance mostly. There are large interventions between components. For the person who is very familiar with

working mechanisms implemented in routing protocols, it's not difficult to name some of these relations between components intuitively. Such as, multiple routing caches in routing path will lead to less data packet loss ratio, compared to single cached routing path, and hello message dissemination will cause more routing overhead for overall routing system. Some of these intuitive recognitions can be wrong, as we demonstrated in sequel. Even if all these qualitative relations can be interpreted correctly, a quantitative analysis method is required to define the effects of components on system performance and dependence between component's performances. To acquire a good analytical model, we firstly consider the pure scoring method for components (e.g. comparative scoring function for component selection). However, this pure scoring method isn't good for differential analysis for component's effects on system performance. It's a good method to analyze combination of effects of constituent components on a system performance, but not applicable for differential analysis for a single component. To expect a quantitative description of single component's effect on a performance metric, we propose Bayesian belief network for induction of significance for each component's effects. After obtaining the probabilistic value of dependence of system performance on one or more components, we can easily detect significant component, who takes most responsibility for performance deterioration. So Bayesian network is a decent mathematical tool for constructing models of component dependence.

### 4.2.1. Introduction to Bayesian Belief Network

The Bayesian Belief Network (BBN) is a mathematical model suited for learning relationships among a large number of variables. BBN can be defined as  $B = (B_S, B_P)$  [22][23].  $B_S$  is a network structure that can be described as a directed acyclic graph (DAG). Nodes in networks represent variables  $U$  under studies, and arc directing from node  $i$  to node  $j$  represent probabilistic dependencies.  $B_P$  is a set of conditional probability tables associated with  $B_S$ . For each,  $x_i \in U$ , the set  $B_P$  contains a conditional probability table  $P(x_i | \pi_i)$ .  $\pi_i$  is combination of values of the parent variables for the variable  $x_i$ . Such a combination is called an instantiation of parent set of  $x_i$ .  $D$  is the set of training data. And,  $D = \{D_1, \dots, D_M\}$ .

The major feature of “conditional independence” is: each variable  $x_i$  is independent of its nondescendants given its parent  $\pi_i$ .

By factorization, we can denote joint probability distribution over  $U$  as:

$$P_B(x_1, \dots, x_N) = \prod_{i=1}^n P_B(x_i | \pi_i) \quad (4.1)$$

Thus, learning the joint probability distribution for BBN is equivalent to learn the conditional probability for each variable given its parent set. Here are two major problems to be resolved in Bayesian network. The first is how to determine network structure  $B_S$ , given training data set  $D$ . The common approach to this problem is to specify a scoring function that evaluate each network structure based on training data,

and determine the best structure in terms of the scoring function. Two popular scoring functions commonly implemented are Bayesian scoring function [22] and the function based on the principle of minimal description length (MDL) [25].

The second problem concerns about how to learn parameters (CPT) after obtaining Bayesian network's structure. We'll focus on resolving this problem throughout the paper. Since Bayesian network's structure is fixed in our research, we only address approaches to learn parameters based on fixed structures. In sequel, for CPT learning for fixed structure BBN, two common techniques will be introduced. They are Bayesian approach and maximum likelihood approach. However, taking advantages of its properties, only maximum likelihood approach will be implemented throughout the research to determine CPT based on training data  $D$  and Bayesian network structure  $B_S$ .

#### 4.2.2. Bayesian Approach to Parameter Learning

*Problem 1: How to determine Bayesian network structure ( $B_S$ ), given database of cases ( $D$ )?*

The basic model:

$$\frac{P(B_{S_i} | D)}{P(B_{S_j} | D)} = \frac{\frac{P(B_{S_i}, D)}{P(D)}}{\frac{P(B_{S_j}, D)}{P(D)}} = \frac{P(B_{S_i}, D)}{P(B_{S_j}, D)} \quad (4.2)$$



We always choose a structure ( $B_{S_i}$ ) with the maximum value of  $P(B_{S_i} | D)$ .

Here are some assumptions as specified in [22]:

*Assumption 1:* The database variables, which we denote as  $Z$ , are discrete.

Then:

$$P(B_S, D) = \int_{B_P} P(D | B_S, B_P) f(B_P | B_S) P(B_S) dB_P \quad (4.3)$$

*Assumption 2:* Cases ( $D_i$ ) occur independently, given a Bayesian network model.

Hence,

$$P(B_S, D) = \int_{B_P} \left[ \prod_{m=1}^M P(D_m | B_S, B_P) \right] f(B_P | B_S) P(B_S) dB_P \quad (4.4)$$

*Assumption 3:* There are no cases that have variables with missing values.

*Assumption 4:* The density function  $f(B_P | B_S)$  in equation (4.4) is uniform.

Together with above four assumptions, here is an important theorems developed for Bayesian by Cooper and Herskovits [22].

**Theorem 1.** Let  $Z$  be a set of  $n$  discrete variables, where a variable  $x_i$  in  $Z$  has  $r_i$  possible value assignments:  $(v_{i1}, v_{i2}, \dots, v_{ir_i})$ . Let  $D$  be a database of  $m$  cases, where each case contains a value assignment for each variable in  $Z$ . Let  $B_S$  denote a Bayesian network structure containing just the variables in  $Z$ . Each variable  $x_i$  in  $B_S$  has a set of parents, which we represent with a list of variables  $\pi_i$ . Let  $w_{ij}$  denote the  $j$ th unique instantiation of  $\pi_i$  relative to  $D$ . Suppose there are  $q_i$  such

unique instantiations of  $\pi_i$ . Define  $N_{ijk}$  to be the number of cases in D in which variable  $x_i$  has the value  $v_{ik}$  and  $\pi_i$  is instantiated as  $w_{ij}$ . Let

$$N_{ij} = \sum_{k=1}^{r_i} N_{ijk}, \quad (4.5)$$

Given assumptions 1 through 4, it follows that:

$$P(B_S, D) = P(B_S) \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \quad (4.6)$$

Hence, the most likely Bayesian network is the one which can maximize the equation (4.6).

*Problem 2: After the optimal Bayesian network structure is found, how to determine the parameters of Bayesian belief network, that is CPT ?*

**Theorem 2.** Let  $\theta_{ijk}$  denote the conditional probability  $P(x_i = v_{ik} \mid \pi_i = w_{ij})$ . The probability of variable  $x_i$  has value  $v_{ik}$ ,  $k$  is between 1 to  $r_i$ , given that the parents of  $x_i$ , represented by  $\pi_i$ , are instantiated as  $w_{ij}$ .  $\theta_{ijk}$  is named a network conditional probability. Background knowledge is denoted by  $\xi$ . Considering four assumptions introduced in step 1, the following results can be derived in [22]:

$$E[\theta_{ijk} \mid D, B_S, \xi] = \frac{N_{ijk} + 1}{N_{ij} + r_i} \quad (4.7)$$

This expectation value of conditional probability will be taken as the conditional probability value.

*Proof:*

$\theta_{ij} = [\theta_{ij1}, \dots, \theta_{ijr_i}]$ ,  $\theta_{ij}$  denotes distribution of values of variable  $x_i$ , given an instantiation  $w_{ij}$  of its parent set  $\pi_i$ .

Firstly, apply Bayes' rule to obtain the probability distribution for  $\theta_{ij}$ , given training data  $D$ , network structure  $B_S$ , and background knowledge  $\xi$ .

$$\begin{aligned} P(\theta_{ij1}, \dots, \theta_{ijr_i} | D, B_S, \xi) &= P(\theta_{ij} | D, B_S, \xi) \\ &= \frac{P(\theta_{ij} | B_S, \xi) \cdot P(D | \theta_{ij}, B_S, \xi)}{P(D | B_S, \xi)} \end{aligned} \quad (4.8)$$

In common,  $P(\theta_{ij} | D, B_S, \xi)$  and  $P(\theta_{ij} | B_S, \xi)$  are denoted as *posterior* and *prior* distribution respectively. Additionally,  $P(D | B_S, \xi)$  is referred as *marginal* distribution or *evidence* distribution.

$$P(D | B_S, \xi) = \int P(D | \theta_{ij}, B_S, \xi) \cdot P(\theta_{ij} | B_S, \xi) d\theta_{ij} \quad (4.9)$$

Apply the uniform Dirichlet prior distribution to  $\theta_{ij}$ , then background  $\xi$  can be replaced by hypo-parameter vector of Dirichlet distribution:

$$\vec{\alpha}_{ij} = (\alpha_{ij1}, \alpha_{ij2}, \dots, \alpha_{ijr_i})$$

Denote uniform Dirichlet prior below:

$$P(\theta_{ij} | B_S, \xi) = Dir(\theta_{ij} | \alpha_{ij1} = 1, \dots, \alpha_{ijr_i} = 1)$$

From the transformation and derivation in [22], the posterior distribution is:

$$P(\theta_{ij} | D, B_S, \bar{\alpha}_{ij}) = \text{Dir}(\theta_{ij} | \alpha_{ij1} = 1 + N_{ij1}, \dots, \alpha_{ijr_i} = 1 + N_{ijr_i}) \quad (4.10)$$

Hence,

$$\begin{aligned} E[\theta_{ijk} | D, B_S, \bar{\alpha}_{ij}] &= \int \theta_{ijk} \cdot P(\theta_{ij} | D, B_S, \bar{\alpha}_{ij}) d\theta_{ij} \\ &= \int \theta_{ijk} \cdot \text{Dir}(\theta_{ij} | \alpha_{ij1} = 1 + N_{ij1}, \dots, \alpha_{ijr_i} = 1 + N_{ijr_i}) d\theta_{ij} \\ &= \frac{N_{ijk} + 1}{N_{ij} + r_i} \end{aligned} \quad (4.11)$$

*Q.E.D*

### 4.2.3. Maximum Likelihood Approach to Parameter Learning

Suppose training data set to be  $D = \{D_1, \dots, D_M\}$ , each  $D_i = \{x_1[i], \dots, x_M[i]\}$  is generated independently. The problem is to find CPT to maximize probability of occurrence of training data set  $D$ .  $\bar{\theta}$  is referred to CPT parameters.  $\bar{\theta}$  is a 3-dimensional matrix, with each element  $\theta_{ijk}$  indicating the conditional probability  $P(x_i = v_{ik} | \pi_i = w_{ij})$ . Define log likelihood function of distribution probability of training set  $D$  as below:

$$LL(\bar{\theta}; D) = \log P_{\bar{\theta}}(D) \quad (4.12)$$

Then, the maximum log likelihood for Bayesian network can be formulated as below:

$$\left\{ \begin{array}{l} \text{Max}_{\bar{\theta}} LL(\bar{\theta}; D) \\ \text{s.t. } \sum_{k=1}^{r_i} \theta_{ijk} = 1, \\ \text{and } \theta_{ijk} \in [0,1], \forall i, j, k \end{array} \right. \quad (4.13)$$

With respect to the independence of elements in training data set  $D$  and formula (4.1), the log likelihood function can be transformed as below:

$$\begin{aligned}
LL(\vec{\theta}; D) &= \log\left(\prod_{m=1}^M P_{\vec{\theta}}(D_m)\right) \\
&= \sum_{m=1}^M \log(P_{\vec{\theta}}(D_m)) \\
&= \sum_{m=1}^M \log(P_{\vec{\theta}}(x_1[m], \dots, x_N[m])) \\
&= \sum_{m=1}^M \log\left(\prod_{i=1}^N P_{\vec{\theta}}(x_i[m] | \pi_i[m])\right) \\
&= \sum_{i=1}^N \sum_{m=1}^M \log(P_{\vec{\theta}}(x_i[m] | \pi_i[m])) \\
&= \sum_{i=1}^N LL_i(\theta_i; D)
\end{aligned} \tag{4.14}$$

The above deduction is based on decomposition of Bayesian network learning problem to summation of log likelihood of distribution probability of variable  $x_i$ . The maximization solution of (4.13) can be decomposed to summation of solutions of following independent maximization problems, for each  $x_i \in U$ :

$$\left\{ \begin{array}{l}
Max_{\theta_i} LL_i(\theta_i; D) \\
s.t. \sum_{j,k} \theta_{ijk} = 1, \\
and \theta_{ijk} \in [0,1], \forall j, k
\end{array} \right. \tag{4.15}$$

For each  $LL_i(\theta_i; D)$ , further decomposition can be performed as following. Denote  $N_{ijk}$  as the number of instantiation of  $(x_i = v_{ik} \mid \pi_i = w_{ij})$  in training data  $D$ .

$$\begin{aligned}
LL_i(\theta_i; D) &= \sum_{m=1}^M \log(P_{\theta}(x_i[m] \mid \pi_i[m])) \\
&= \sum_{\pi_i^j} \sum_{x_i^k} \log(P_{\theta}(x_i^k \mid \pi_i^j)) \\
&= \sum_{\pi_i^j} \sum_{x_i^k} \log \theta_{ijk} \\
&= \sum_{\pi_i^j} \sum_{x_i^k} \log \theta_{ijk}^{N_{ijk}} \\
&= \sum_{\pi_i^j} LL_{ij}(\theta_{ij}; D)
\end{aligned} \tag{4.16}$$

Then full maximum log likelihood for Bayesian network can be achieved by resolving following problem:

$$\left\{ \begin{array}{l}
Max_{\theta_i} LL_{ij}(\theta_{ij}; D) \\
s.t. \sum_{j,k} \theta_{ijk} = 1, \\
and \theta_{ijk} \in [0,1], \forall j, k
\end{array} \right. \tag{4.17}$$

**Theorem 3.** The solution for problem formatted in (4.17) is:

$$\theta_{ijk} = \frac{N_{ijk}}{\sum_{k=1}^K N_{ijk}} \tag{4.18}$$

The maximum likelihood estimator is minimum variance unbiased estimator (MVUE). More importantly, according to [25], for the *M-closed* case, i.e. the case in which the data generating distribution can be represented with the Bayesian network, the joint distribution of  $X$  given the maximum likelihood (ML) parameters converges (with growing sample size) to the generating distribution. Thus, also the conditional distributions of the class variable given the other variables and the maximum likelihood parameters converge to the true conditional distribution. Thus, for our Component Dependence Network, we'll choose Maximum likelihood approach to determine probability parameter CPT.

### 4.3. Bayesian Network Structure for Component Dependence

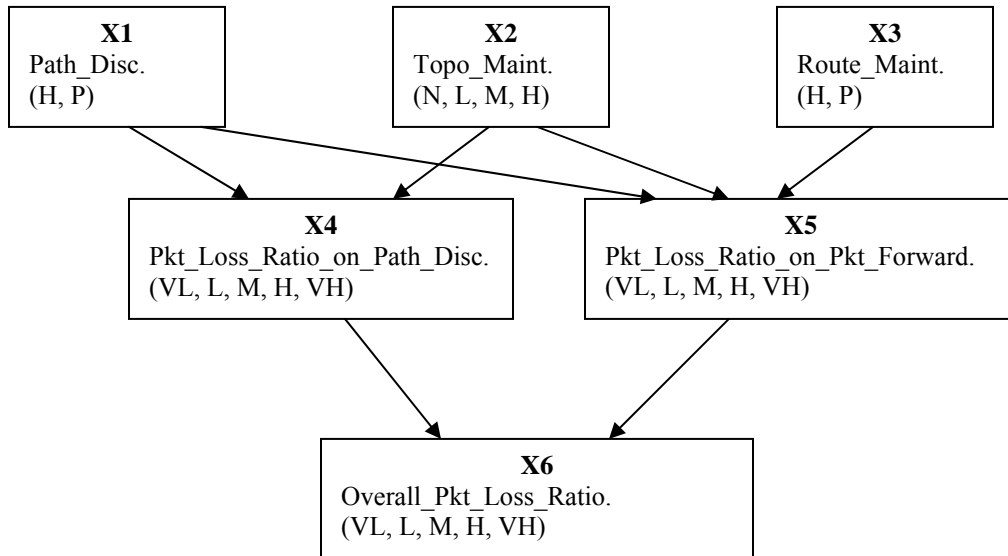


Figure 4.3 Outline of Bayesian Network Structure for Component dependence

As shown in Fig.4.3, to try to put overall routing performance metrics and component performance metrics into the framework of Bayesian network, we have five variables X1, X2, X3, X4 and X5. In the following chapters, the final and formal component dependence network will show a four layer structure, which includes one more layer of component metrics. X1 is path discovery component. Its value can be H (hop based) or P (path based). X2 is topology database maintenance component, whose value can be N (null), L (low frequency), M (medium frequency) and H (high frequency). X3 is route maintenance component, and has values of H (hop based) and P (path based). X4 represents packet loss ratio occurring in path discovery process, X5 represents packet loss incurred in packet forwarding process, and X6 is overall data packet loss ratio. X4, X5 and X6 all have values of (VL, L, M, H, VH). VL denotes very low. L denotes low. M denotes medium. H denotes high. VH denotes very high. All experimental data will be pre-discretized.

Each arc in Figure 4.3 represents a conditional probability for a specified value of a variable, given its parent variable set. We define a Bayesian network structure as in Figure 4.3. We need to firstly validate the structure by applying algorithm defined in Theorem 1. Based on algorithm (4.18), optimal network probability should be deduced from empirical data generated by simulations.

The ultimate purpose by applying component dependence network is to locate the significant component which can be replaced to improve the overall routing performance. The quantitative analysis based on dependence network is critical in



helping locate the significant component and thus to improve the overall routing performance.

#### **4.4. Replacement of Significant Component**

Below is an example to show the effectiveness of replacing weak component. Simulation results will show that changing or inserting a component (subcomponent) can improve overall performance.

In the below case, DSR will be performed for three scenarios: Data Traffic (640 bits/sec), Voice Traffic (2560 bits/sec) and Video Traffic (20480 bits/sec). Simulation is performed under mobility of 0, 10, 20, 30, 40 and 50 meters/second. We're trying to detect significant component for packet loss ratio and replace weak component to decrease packet loss ratio. By tracking each data packet and monitoring intra-component interface, the percentage of packet loss due to path discovery component is 0, and percentage of packet loss due to route maintenance is 1. Therefore, route maintenance component is the significant component which takes all responsibility for packet loss ratio. As a result, we'll focus on route maintenance component and try to figure out how to improve performance of route maintenance component and decrease packet loss ratio.

In DSR, after RREP is received, data packet will be copied to an intra-component interface: maintenance buffer. And sender of the data packet will send an Ack-request to next hop which will be used to relay the data packet. After receiving Ack-request, the next hop will send an Acknowledgement to the sender. If the sender can receive

the Ack within timeout, copy of the data packet will be destroyed. Otherwise, data packet will be resent with an Ack-request message until maximum number of retransmission is reached.

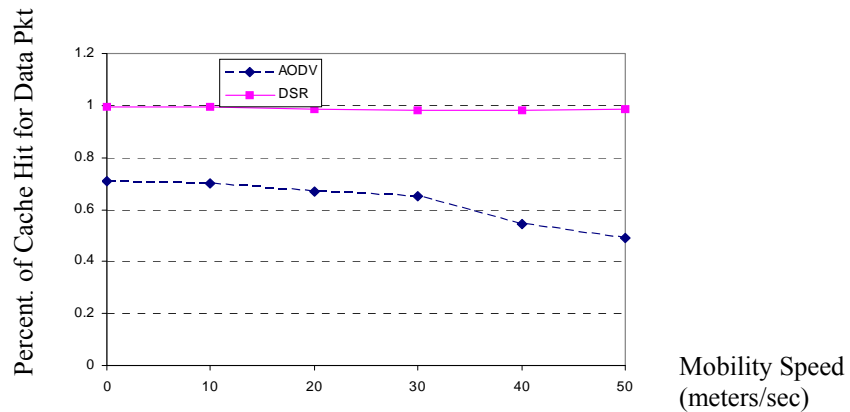


Figure 4.4 Simulation Result for Percentage of Cache Hit for Data Packet

When the maximum number of retransmission is reached, the data packet will be discarded. Since DSR has a pretty high cache hit ratio (shown in Figure 4.4), the controlling overhead remains low because path discovery process can be saved for cached route that can be found in route cache table. However, on the other side, this dependence on cached route is a key reason for high packet loss ratio. Especially for networks with fast topology changing, stale cached routes can't be deleted in time and pollute all other route cache tables throughout entire networks. When nodes receive stale path, and use it to forward data packets, it will be more difficult for data packet to hear Acknowledgement from next hop, since network topology is already changed

and next hop isn't reachable. After investigating mechanism of route maintenance component of DSR, we understand that some mechanism for deleting stale cached route is required to decrease DSR's packet loss ratio. In this case, the significant component is route maintenance component, and the weak component is topology database maintenance component.

As shown in Figure 4.5, data packet loss ratio is increased along with the mobility speed, and increased along with traffic density. So, video's packet loss ratio is the largest, voice's packet loss ratio is in middle, and data's packet loss ratio is the least. As shown in Figure 4.5, data packet loss ratio is larger than 0.3 mostly due to outdated cached routing paths.

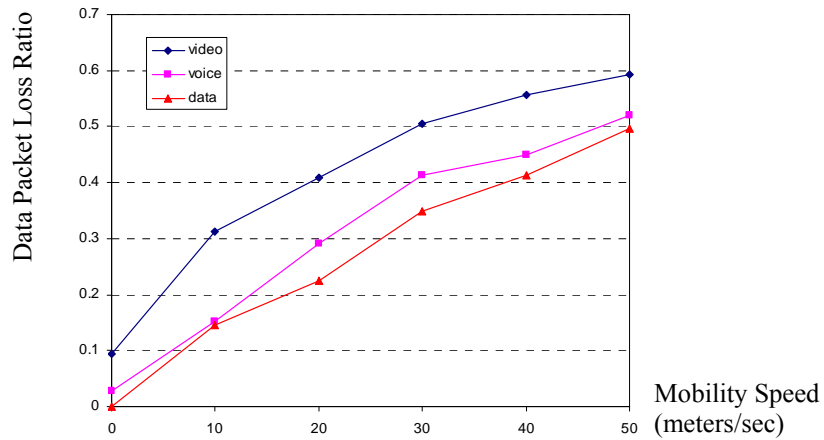


Figure 4.5 Simulation Result of Data Packet Loss Ratio

To reduce the data packet loss ratio, we adopt Topology Database Maintenance component to help detect stale neighbors, and delete cached path including the stale neighbors to keep cached path fresh enough. Each node will broadcast Hello message at an interval T (hello). And each node will maintain a neighbor list which record the latest update time for each neighbor. If a neighbor on the neighbor list can't be heard by the node before a timeout, the neighbor will be deleted from the neighbor list, and all cached routing path which include the stale neighbor will be removed from the route cache table. This Topology Database Maintenance component is borrowed from AODV. We borrowed the component from AODV to reduce the cache hit rate for routing links that contain stale routing neighbors. This introduction of the new component helps to reduce the packet loss ratio effectively. The activity diagram is shown in Figure 4.6. After inserting new component into DSR routing protocol, the data packet received ratio is incremented effectively as shown in Figure 4.7.

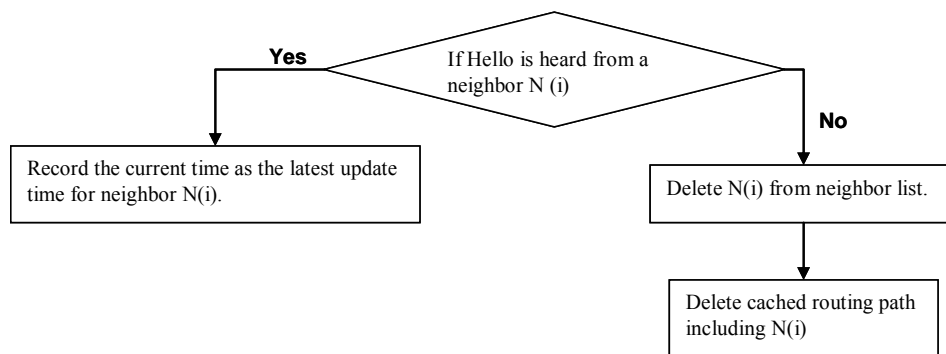


Figure 4.6: Activity Diagram for Topology Database Maintenance Component

## 4.5. Simulation Results and Analysis

By comparing improvement ratio for data packet received ratio at different traffic density, it's shown that video traffic has the largest improvement ratio at most of time.

$$\text{Improvement ratio} = \frac{\text{(packet received ratio with improved DSR)} - \text{packet received ratio with original DSR}}{\text{packet received ratio with original DSR}} \quad (4.19)$$

The improvement is larger for faster mobility and higher traffic density. This improvement is crucial for damping performance variance under different mobility speed and traffic density. As shown in Figure 4.7 and Figure 4.8, by only inserting an extra component, the data packet received ratio can be increased by an average of 43.67% for video traffic, 30.53% for voice traffic and 24.17% for data traffic. This is a good example to show that how to decide significant component and weak component and how to replace weak component to improve the overall performance.

After improving DSR by implementing Hello message, besides recording packet received ratio improvement percentage (Figure 4.8), we also make statistics for routing overhead increment ratio (Figure 4.9) and end to end delay increment ratio (Figure 4.10). As shown in these figures, all three traffic modes make improvement for data packet received ratio by sacrificing routing controlling overhead and end to end delay.

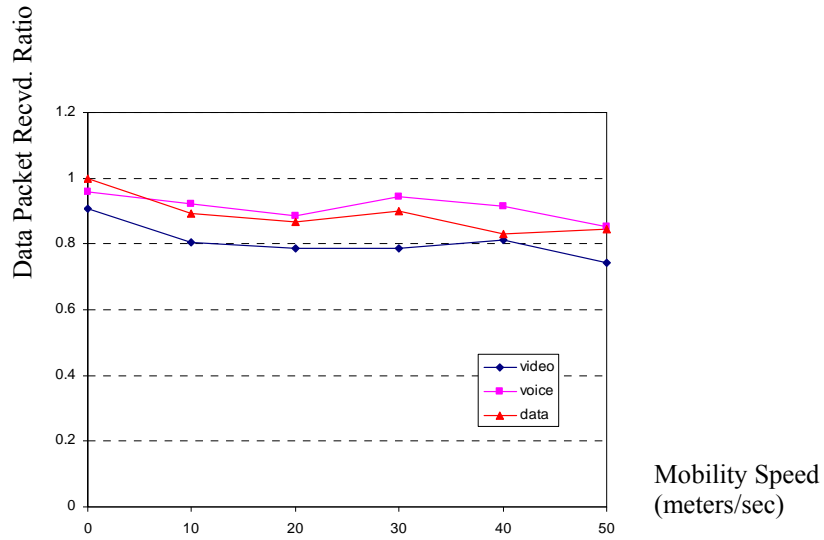


Figure 4.7 Simulation Result for Data Packet Received Ratio

However, this compromise in end-to-end delay and routing overhead is endurable under some scenarios. It's observed that video traffic has the least increment for controlling overhead and EtE delay under mobility scenarios higher than 20 meters/sec. Furthermore, video traffic has the least variance on changes of controlling overhead and EtE delay. Intuitively, we can conclude that for video traffic that expecting low data packet loss ratio, the improved DSR is preferred. And for mobility speed less than 20 meters/sec, voice traffic and data traffic both have endurable increment ratio for controlling overhead and EtE delay. Hence, voice and data traffic

under low speed (< 20 meters/sec) can generate better packet received ratio without deteriorating other two performance metrics.

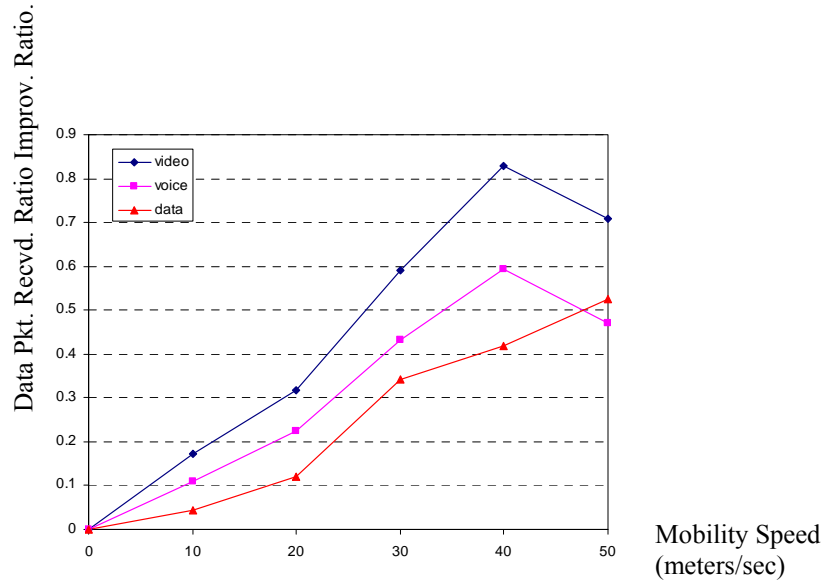


Figure 4.8 Simulation Result for Improving Ratio For Data Packet Received Ratio

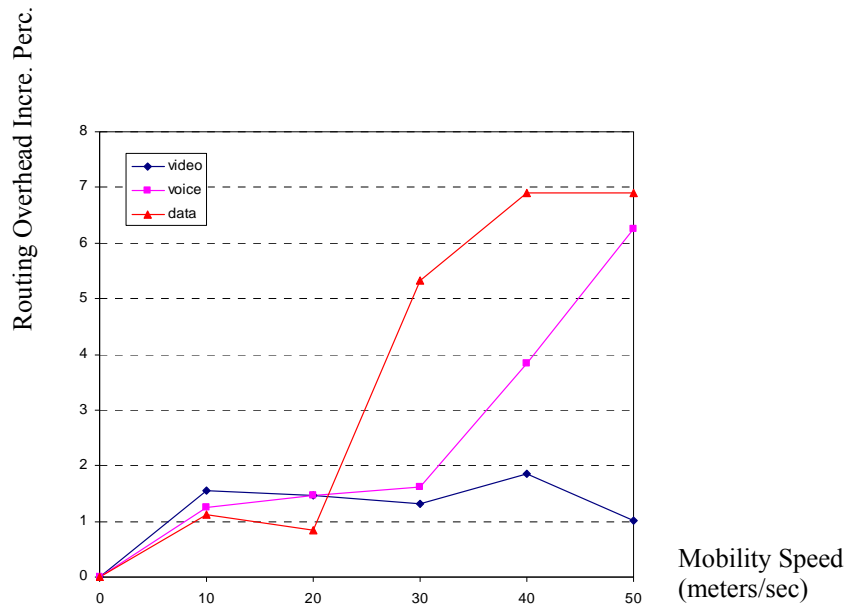


Figure 4.9 Simulation Result for Increment Percentage of Routing Overhead

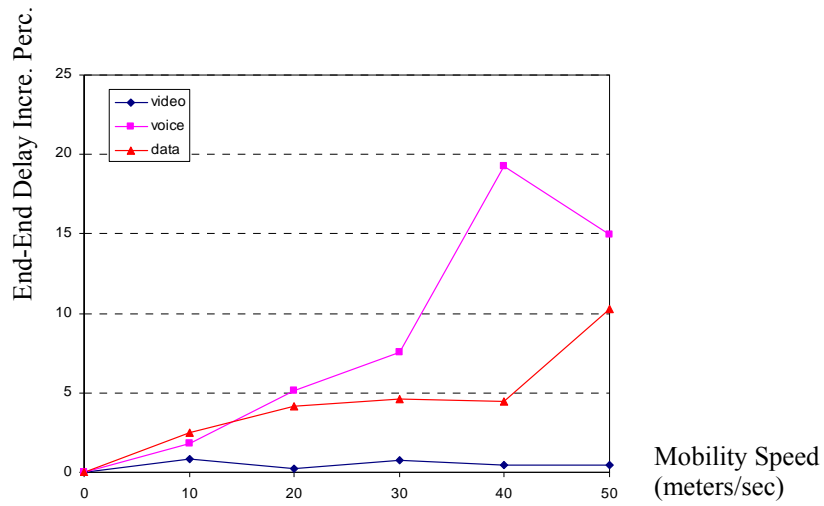


Figure 4.10 Simulation Result for Increment Ratio For End-to-End Delay

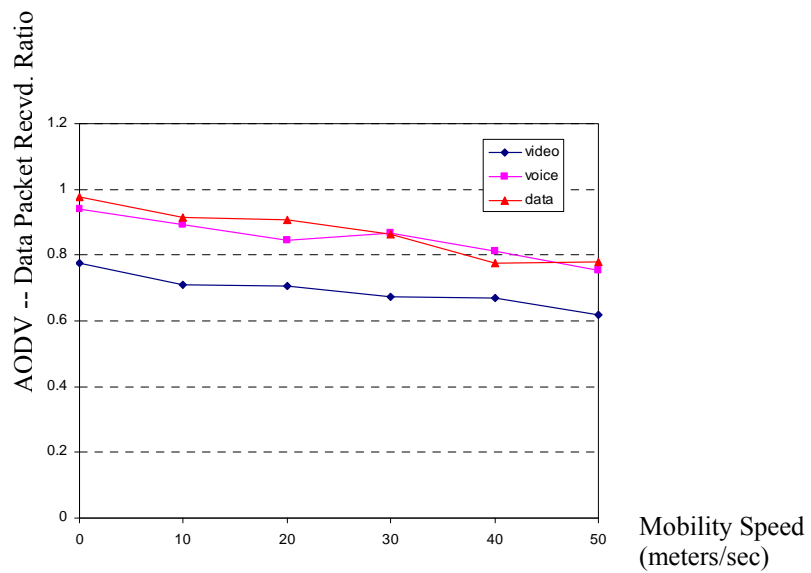


Figure 4.11 Simulation Result for Data Packet Received Ratio for AODV



Being compared to DSR in Figure 4.4, AODV's route cache hit rate is much lower than DSR's. Then, how is the data packet received ratio under AODV's working mechanism? As shown in Figure 4.11, the packet received ratio of AODV is much higher than DSR shown in Figure 4.5. It's instructive to compare AODV's packet received ratio to improved DSR's packet received ratio. The improvement ratio for packet received ratio is computed as below:

$$\text{Improvement ratio} = (\text{packet received ratio with improved DSR} - \text{packet received ratio with AODV}) / \text{packet received ratio with AODV} \quad (4.20)$$

As shown in Figure 4.12, the improvement is larger for faster mobility and higher traffic density. This improvement is crucial for damping performance variance under different mobility modes and traffic modes. As compared to AODV, the data packet received ratio can be increased by an average of 16.78% for video traffic, 7.3% for voice traffic and 2.5% for data traffic. Especially, for low speed scenarios (<20 meters/sec), the improvement ratio is negative for data traffic. It shows that packet received ratio under AODV is higher than the improved DSR. Intuitively, scenarios with low traffic density will prefer AODV to decrease data packet loss ratio for low

mobility speed scenarios. However, for high mobility scenarios, improved DSR is attractive for its higher improvement for data packet received ratio.

Being compared to AODV, the improved DSR scheme can have higher packet received ratio by sacrificing controlling overhead and EtE delay. As shown in Figure 4.9 and Figure 4.10, for improved DSR, video traffic has the least increment for controlling overhead and EtE delay under high mobility scenarios ( $> 20$  meters/second). However, voice traffic and data traffic both have large increment for controlling overhead and EtE delay under high mobility scenarios ( $> 20$  meters/second). Intuitively, high density traffic will prefer improved DSR for all mobility scenarios, and low density traffic with high mobility can select improved DSR with compromise for controlling overhead and EtE delay.

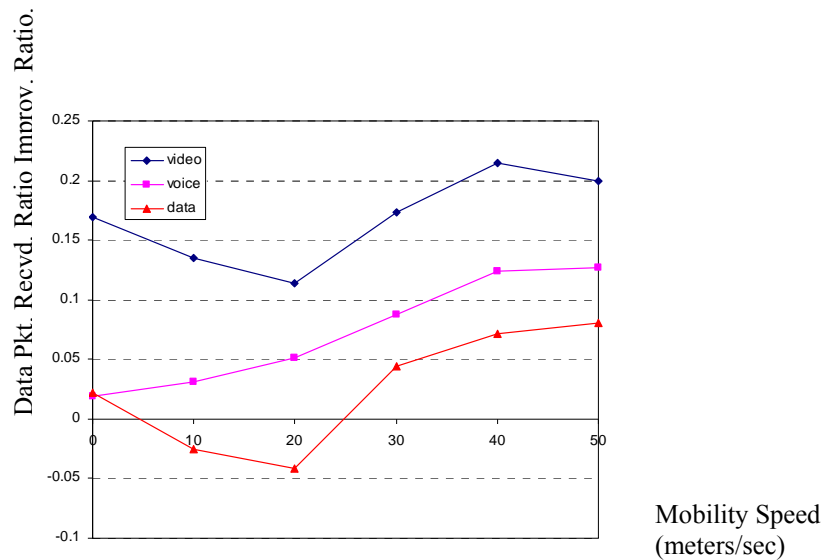


Figure 4.12 Simulation Result for Improving Ratio For Data Packet Received Ratio

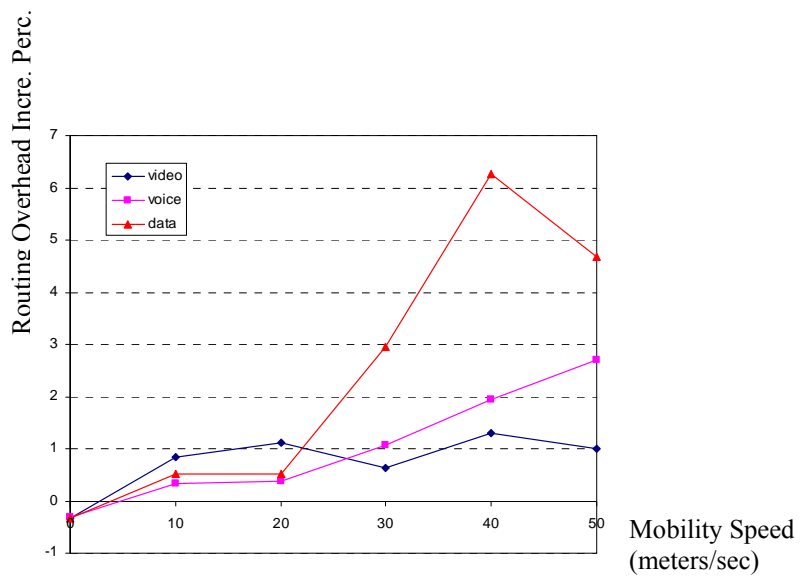


Figure 4.13 Simulation Result for Increment Ratio for Routing Overhead

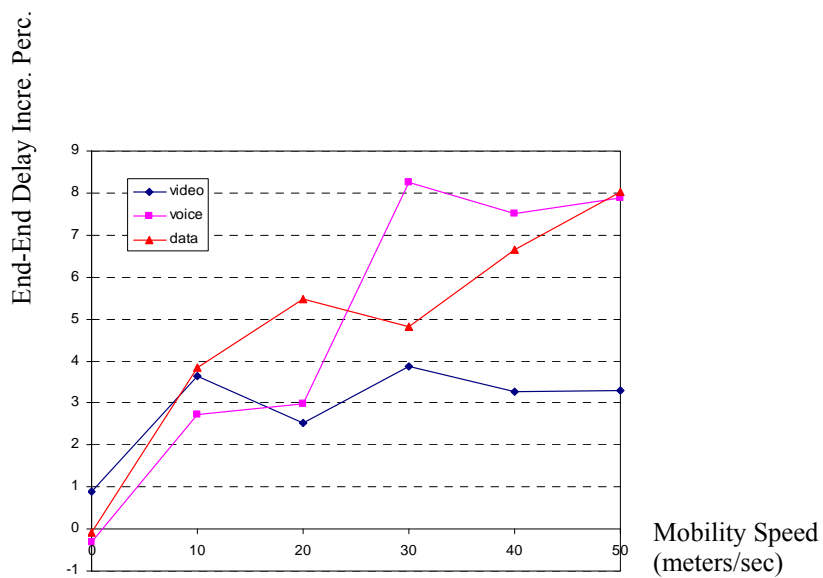


Figure 4.14 Simulation Result for Increment Percentage for End-to-End Delay

In conclusion, original DSR has the worst data packet received ratio for all scenarios, AODV is better than the original DSR, and improved DSR generate the best packet received ratio on average. However, the improved DSR has to sacrifice controlling overhead and EtE delay to exchange for higher packet received ratio. This compromise is endurable under some situations shown in above simulation results. Generally, for low density traffic with low mobility speed, AODV is preferred, and for high density traffic with high mobility speed, improved DSR is preferred.

In order to implement Bayesian network for analyzing dependence among various components, we at least have three problems to be resolved in the following chapters:

1. How to determine the structure of component dependence network?

A simple method is required to determine the structure of component dependence network fast and reliably. Once the structure is determined, the significance level can be calculated, and the significant component can be located and replaced to improve overall routing performance. In the following chapters, a conditional independence test based structure learning method will be presented. Based on the conditional independence test, an Include-Exclude algorithm will be presented to show the effectiveness in learning the structure of independence. The proof of the Include-Exclude Algorithm is provided in the following.

2. How to prove conditional independence within component dependence network?

It's intractable to build up analytical model for conditional probability and prove conditional independence. Hence, we need to collect simulation data and calculate statistics for correlation coefficient for a pair of conditional probabilities. If the correlation coefficient is very small, the two conditional probabilities can be assumed to be independent roughly.

3. What's a good scoring function for evaluating component's effects on overall performance?

After determining CPT for component dependence network, the significant component should be determined by calculating some scoring function. The component with max or min score is the significant component. The scoring function should have two properties below. Firstly, scoring function has low computation complexity, and is easy to be deduced from CPT. Secondly, the function should demonstrate the relationship between component performance and overall performance.

All above three problems will be resolved in the following chapter.

## **Chapter 5**

### **Structure Learning of Component's Dependence Network**

Since the development of the packet radio networks in the 1970s, Mobile Ad Hoc Network (MANET) has received significant research attention. The nature of easy deployment without pre-existing infrastructure makes ad hoc networks an attraction for dynamically distributed situations comprising mobile wireless stations. In recent years, increasingly widespread application of mobile Ad Hoc network accelerates development of mobile distributed computing. Many routing protocols have been designed for Ad Hoc network to satisfy needs for more actively distributed algorithms.

Ad Hoc routing protocols can be categorized to two major prototypes: proactive (OLSR) and reactive (AODV, DSR). These two types of protocols are featured by the different mechanism of path discovery. Proactive routing protocols constantly maintain a set of available routes for all stations throughout network. Reactive (on demand) routing protocols, on the other hand, only initiate path discovery procedure in desire of a route to a specified destination.

In comparison of performance of various MANET routing protocols, a number of technical papers have been published [3] [4] [5]. Large variation of performance of routing protocols is exhibited in above papers. A protocol outperforms in certain situation may not do so in other situations. None of existing protocols can absolutely

outperform all others in overall situations. By deeply investigating MANET routing protocols, we discovered that some key functions can influence overall performance significantly. In some cases, substitution of some functions instead of entire routing protocols can improve performance significantly. Hence, selecting components from function repository and assembling them together into a well performed routing protocol is an alternative other than rewriting entire protocol.

From above observations and considerations, Component Based Routing protocol is proposed in [1] [2]. Developed from this framework, a quantitative analysis model is introduced in this paper to facilitate learning of impact of each component on various performance metrics. Inspired by Bayesian network, Component Dependence Network (CDN) is proposed to represent complicated relationship between components and performance metrics. The impact of component to a performance metric is encoded into local conditional probability distribution function. CDN is a hierarchical network with four layers. And each intermediate  $n$ th layer will d-separate the  $(n-1)$ th layer and  $(n+1)$ th layer. This cross-layer d-separation property simplifies structure learning for CDN. Based on independence test, we develop Include-Exclude algorithm to learn parent set for CDN variables. And the ability of Include-Exclude algorithm to return the minimal cut set is investigated. Furthermore, formal logic arithmetic for determining d-separation is developed and implemented throughout the paper.

This paper is organized in the following way. The second section will briefly introduce architecture of component based routing protocol, and Component Dependence Network (CDN) will be presented. In the third section, d-separation and its formal logical arithmetic will be introduced to facilitate studies in the following sections. Then, section four will introduce Include-Exclude algorithm, and investigate its property. And the parent set learning algorithm will be presented. Contributions of this article are four folds:

- a. Introduce a novel quantitative analysis model CDN for automatic learning of component's impact on performance metrics. CDN not only comprises all major achievement in our previous work for component based routing framework, but also develop a new idea of encoding component's impact into conditional probability distribution which can be visualized by a graphical topology.
- b. Develop a formal logical learning arithmetic for d-separation. This formal arithmetic can be directly implemented in application software to help identify d-separation, hence to determine local conditional independence.
- c. Design Include-Exclude algorithm with the property of returning minimal cut set for a source and destination pair.
- d. Based on independence test, a simple and applicable structure learning algorithm for CDN is proposed.



## 5.1 Introduction to Component Based Routing Design

Component based routing protocol is a new paradigm for network modeling, design and study. In conventional research studies, routing protocols are always studied as a whole entity. However, for component based methods, we investigate component (unit of routing protocols) instead of entire routing protocols.

The hierarchical architecture of component based routing protocol is briefly listed below (details presented in[1][2]):

*1. Definition of components:* Component is a fundamental abstraction that applies to many physical structures such as physical parts of machine, structures of a software objects. The concept of component proposed here is to define nonphysical objects: behaviors of routing protocols. No structures such as route cache table, neighbor list or packet format will be expressed by components.

We synthesis the common functionalities and categorize them into four components: *path discovery component*, *topology database maintenance component*, *route maintenance component* and *data packet forwarding component*. *Subcomponents* are component's methods, which are required to realize functional components of protocols.

*2 .Derivative Component Metrics:* How to trace and compare each component's performance during execution of routing protocols? The derivative components

metrics are distinguished from overall metrics by its emphasis on evaluating merits of specified components instead of effectiveness of overall routing protocols. The conventional overall performance metrics are latency, throughput and packet loss ratio. In respect to the effectiveness of routing protocols, overall metrics can only explain results of routing protocols. More detailed derivative component metrics should be designed to better assess execution of functional components. In addition, meaningful component metrics enables us to better understand the interaction between performance of components and overall performance. An example of derivative component metrics: Percentage of Route Cache Hit for Data Packet =  $\frac{\# \text{Cache Hit Data Packet from High Layer}}{\# \text{Total Data Packet from High Layer}}$ . This is the measure of effectiveness of path discovery component with respect to its capability of yielding cached paths.

*3. Allocated Component Metrics:* How to evaluate each component's contribution to the value of performance metrics? *Performance projection* is presented here to resolve the problem. In component based routing protocol designing methodology, we will project each overall performance metric onto allocated performance metrics. Allocated performance metrics is directly correlated to overall performance metrics, and can be summed up to obtain overall performance. For instance, the overall routing controlling overhead ( $O_{\text{OVERALL}}$ ) can be contributed to path discovery component ( $O_{\text{PD}}$ ), route maintenance component ( $O_{\text{RM}}$ ) and topology database maintenance component ( $O_{\text{TM}}$ ).

Thus:

$$O_{\text{OVERALL}} = O_{\text{PD}} + O_{\text{RM}} + O_{\text{TM}} \quad (5.1)$$

For overall data packet loss, it can be decomposed into data packet loss due to failure of path discovery component ( $L_{\text{PD}}$ ) and failure of data packet forwarding ( $L_{\text{DF}}$ ).

$$L_{\text{OVERALL}} = L_{\text{PD}} + L_{\text{DF}} \quad (5.2)$$

For end to end delay ( $D_{\text{OVERALL}}$ ), it comprises time consumed by path discovery component ( $D_{\text{PD}}$ ), data packet forwarding component ( $D_{\text{DF}}$ ), and possibly by route maintenance component ( $D_{\text{RM}}$ ).

$$D_{\text{OVERALL}} = D_{\text{PD}} + D_{\text{DF}} + D_{\text{RM}} \quad (5.3)$$

## 5.2 Hierarchical Structure for Component Dependence Network

In order to understand how each component contributes to overall performance metrics, we have to trace allocated component metrics and derivative component metrics defined above. After collecting all metrics statistics from simulation results, Bayesian network can be created and help us to analyze relationship between components and performance metrics. In our research, the Bayesian network is named as Component Dependence Network (CDN), since it's a Bayesian network that encodes dependence between components and performance metrics as local conditional probability distribution function.

According to the causal relationship between different layers of metrics, we proposed a four layer structure for CDN. As shown in Fig5.1:

*1. Component layer:* consists of routing protocol's activity components: Path Discovery Component (PD), Route Maintenance Component (RM), Topology Database Maintenance Component (TM), and Data Packet Forwarding Component (DF). As shown in Fig5.1, the Path Discovery Component (PD) has two values Hop Based (H) and Path Based (P). Topology Maintenance Component (RM) has two values Hello Assisted (H) and Non-Hello Assisted (NH). Route Maintenance Component (RM) has two values Hop Based (H) and Path Based (P). Data Packet Forwarding Component has two values Hop Based (H) and Path Based (P).

*2. Derivative Component Metric Layer:* contains component metrics listed in the chapter 3.5. The component metrics' values are directly dependent on component layer's value. Parent set for a derivative component metric will contain variables from component layer. All nodes on the Derivative Component Metric Layer have three values as low, middle and high.

*3. Allocated Component Metric Layer:* is directly correlated to overall performance metrics. As shown in the above, overall performance metrics can be directly decomposed into summation of allocated component metrics. Being conditioned by

derivative component metrics layer, this layer is d-separated from component layer, as shown later. All items on this layer have three values as low, middle and high.

4. *Overall Performance Metric Layer*: comprises of traditional overall performance metrics: data packet loss ratio, controlling overhead and end-to-end delay.

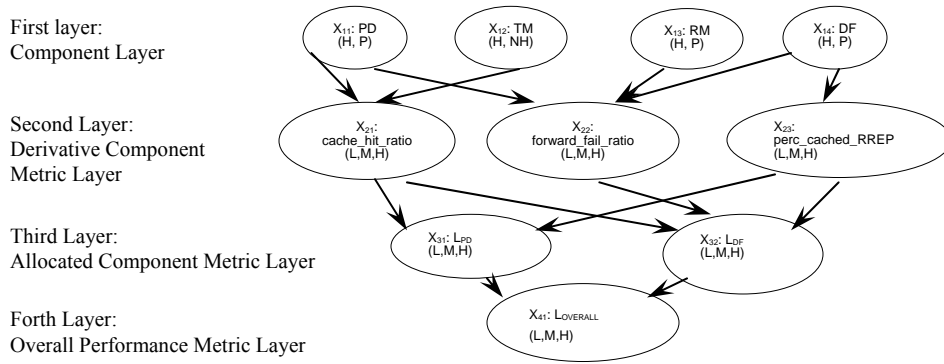


Figure 5.1 The Component Dependence Network.  
Ten variable nodes ( $X_{11}, X_{12}, \dots, X_{41}$ ) are contained.

Noticed in Figure 5.1, for each variable on the  $n$ th layer, its parent set only contains elements on  $(n-1)$ th layer. For instance, parent set of  $X_{31}$  comes from the 2<sup>nd</sup> layer. This is called “cross-layer d-separation” property of CDN, which is a crucial property that simplifies structure learning for CDN. d-separation theorem and its formal logical arithmetic will be introduced in the next section.

### 5.3 d-separations Theorem and Formal Logical Arithmetic

Bayesian Network is a graph based information systems. It encodes local independence by implementing d-separation criterion [6] along all trails between source and destination pair  $\langle x, y \rangle$ . In probability theory, conditional independence can be determined if the following quantitative equation holds:  $p(x|y,Z) = p(x|Z)$ . This is defined as conditional independence for a pair  $\langle x, y \rangle$ , given evidence set  $Z$ . Hence, a formal definition of *Conditional Independence Indicator* is given below:

**Definition 1:**  $\text{Ind}(x, Z, y)$  is a Conditional Independence Indicator for source node  $x$  and destination node  $y$ , given evidence set  $Z$ . Iff  $p(x|y, Z) = p(x|Z)$ , then  $\text{Ind}(x, Z, y) = 1$ . Otherwise,  $\text{Ind}(x, Z, y) = 0$ .

Furthermore, if  $x$ ,  $y$  and  $Z$  can be explicitly described as some variable vertex in a Bayesian network, then Conditional Independence Indicator's value is equivalent to value of *d-separation Indicator*  $D(x, Z, y)$ , which is defined below.

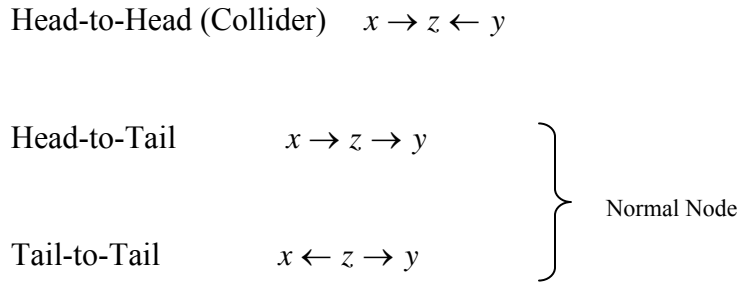
**Definition 2:**  $D(x, Z, y)$  is a d-separation Indicator for source node  $x$  and destination node  $y$ , given evidence set  $Z$ . If source and destination pair  $\langle x, y \rangle$  is d-separated by the evidence set  $Z$ , then  $D(x, Z, y) = 1$ , and  $\text{Ind}(x, Z, y) = 1$ . Otherwise,  $D(x, Z, y) = 0$ , and  $\text{Ind}(x, Z, y) = 0$ .

In Bayesian Network, the quantitative calculation for determining conditional independence in probability theory isn't necessary any more. By searching trails

between source and destination variable nodes  $\langle x, y \rangle$ , we can decide the value of  $D(x, Z, y)$ , according to d-separation criterion. d-separation criterion is implemented along the trail-wise. First of all, trail is defined below. According to Pearl in [6], in directed acyclic graph, a trail (denoted by  $t$ ) is a sequence of consecutive neighboring variables from a source node ( $x$ ) to a destination node ( $y$ ).

**Definition 3:** Denote  $t(x, Z, y)$  as Trail Blockage Indicator. If the trail between  $x$  and  $y$  is active, given evidence set  $Z$ , then this means  $t(x, Z, y) = 0$ . Otherwise, if the trail is blocked, then  $t(x, Z, y) = 1$ .  $Z$  is an evidence set.

Before going to explain the conditions for a blocked trail, we firstly define three categories of variable nodes by distinguishing directions of arrows.



Pearl presented his definition for active trail in [6]. Alternatively, by some logical transformation, we can summarize the conditions for trail's blockage below.

**Theorem 1:** for the  $i$ th trail  $t_i$  between  $x$  and  $y$ , when at least one of criterion 1 and criterion 2 is satisfied, the trail  $t_i$  is blocked by evidence set  $Z$ . Hence, Trail Blockage Indicator  $t_i(x, Z, y) = 1$ . Otherwise,  $t_i(x, Z, y) = 0$ .

**Criterion 1:** Trail  $t_i$  contains head-to-tail node or tail-to-tail nodes, and at least one of them is in evidence set  $Z$ .

**Criterion 2:** Trail  $t_i$  contains head-to-head nodes (colliders), and at least one of them is neither in evidence set  $Z$ , nor has descendants in evidence set  $Z$ .

**Theorem 2:** Iff all trails between  $x$  and  $y$  are blocked by evidence set  $Z$ , then d-separation indicator  $D(x, Z, y)=1$ . Otherwise,  $D(x, Z, y)=0$ .

$D(x, Z, y)=1$  is equivalent to the statement that  $x$  is conditionally independent of  $y$ , given evidence set  $Z$ . Theorem 1 and Theorem 2 actually interpret that conditional independence can be explicitly learned by searching trails with respect to d-separation criterion. Criteria are presented above in text for announcing d-separation. More detailed algorithm is desired to be performed in software programming for implementing d-separation criterion. Therefore, we design an algorithm based on some indicators and logical operations.

For the  $j$ th intermediate node along the  $i$ th trail between  $x$  and  $y$ , define *Collider Indicator* ( $C_{ij}$ ), *Variable-in-Evidence-Set Indicator* ( $E_{ij}$ ), and *Descendant-in-evidence-set indicator* ( $D_{ij}$ ) below:

**Definition 4:** Define  $C_{ij}$  as a Collider Indicator for the  $j$ th intermediate node on the  $i$ th trail.



$$C_{ij} = \begin{cases} 1 & , \text{ the } j\text{th node is a collider} \\ 0 & , \text{ the } j\text{th node is not a collider} \end{cases}$$

**Definition 5:** Define  $E_{ij}$  as a Variable-in-evidence-set Indicator for the  $j$ th intermediate node on the  $i$ th trail.

$$E_{ij} = \begin{cases} 1 & , \text{ the } j\text{th node is a member of evidence set } Z \\ 0 & , \text{ the } j\text{th node is not a member of evidence set } Z \end{cases}$$

**Definition 6:** Define  $D_{ij}$  as a Descendant-in-evidence-set Indicator for the  $j$ th intermediate node on the  $i$ th trail.

$$D_{ij} = \begin{cases} 1 & , \text{ At least one of the } j\text{th node's descendants is a member of evidence set } Z \\ 0 & , \text{ None of the } j\text{th node's descendants is a member of evidence set } Z \end{cases}$$

**Definition 7:** Define  $\alpha_i$  as Collider-in-Evidence indicator below.  $K$  is the number of intermediate nodes on the  $i$ th trail.

$$\begin{aligned} \alpha_i &= \prod_{j=1}^K [C_{ij} \cap (D_{ij} \cup E_{ij})] \\ &= \begin{cases} 1 & , \text{ Along the } i\text{th trail, each Collider or its descendant in the evidence set } Z \\ 0 & , \text{ Along the } i\text{th trail, at least neither one Collider nor its descendant in the evidence set } Z \end{cases} \end{aligned}$$

**Definition 8:** Define  $\beta_i$  as Normal-in-Trail Indicator.  $K$  is the number of intermediate nodes on the  $i$ th trail.

$$\beta_i = \bigcup_{j=1}^K [\overline{C_{ij}} \cap E_{ij}] = \begin{cases} 1 & , \text{ Along the } i\text{th trail, at least one normal node in the evidence set } Z \\ 0 & , \text{ Along the } i\text{th trail, none of normal nodes in the evidence set } Z \end{cases}$$

**Definition 9:** Define  $\gamma_i$  as Collider-in-Trail Indicator below.  $K$  is the number of intermediate nodes on the  $i$ th trail.

$$\gamma_i = \bigcup_{j=1}^K C_{ij} = \begin{cases} 1 & , \quad \text{Along the } i\text{th trail, at least one node is a Collider} \\ 0 & , \quad \text{Along the } i\text{th trail, all nodes are normal node} \end{cases}$$

Following d-separation criterion stated in Theorem 1, and making some transformation by logical arithmetic, below algorithm for trail blockage indicator can be derived.

**Lemma 1:** The Trail Blockage Indicator for the  $i$ th trail between  $\langle x, y \rangle$ , given evidence set  $Z$ , can be expressed as:

$$t_i(x, Z, y) = (\overline{\alpha_i} \cap \gamma_i) \cup \beta_i \tag{5.4}$$

$t_i(x, Z, y) = 1$  indicates that, given the evidence set  $Z$ , the trail between  $\langle x, y \rangle$  is blocked.

$t_i(x, Z, y) = 0$  indicates that, given the evidence set  $Z$ , the trail between  $\langle x, y \rangle$  is unblocked.

**Lemma 2:** d-separation indicator can be expressed as the intersection of all Trail Blockage Indicators between  $x$  and  $y$ :

$$D(x, Z, y) = \bigcap_{i=1}^L t_i(x, Z, y) \tag{5.5}$$

L: the number of trails between  $x$  and  $y$ .

$D(x, Z, y)=1$  indicates that all trails between  $\langle x, y \rangle$  are blocked. Hence, evidence set  $Z$  is a cut set of  $\langle x, y \rangle$ . Source node  $x$  and  $y$  are conditionally independent, when given the cut set  $Z$ .

$D(x, Z, y)=0$  indicates that at least one of trails between  $\langle x, y \rangle$  are un-blocked. Hence, evidence set  $Z$  isn't a cut set of  $\langle x, y \rangle$ . Source node  $x$  and  $y$  are not conditionally independent, when given the evidence set  $Z$ .

**Definition 10:** Cut set  $\text{cut}(x,y)$  is any evidence set  $Z$  that satisfies  $D(x, Z, y)=1$ . The minimal cut set  $\text{cut}_{\min}(x,y)$  is the cut set with the least number of elements.

Lemma 1 and Lemma 2 helps to transform d-separation from text description into logic arithmetic that can be easily implemented by computer. The below example will illustrate how effective is Lemma 1 and 2 in determining d-separation, given different evidence set  $Z$ .

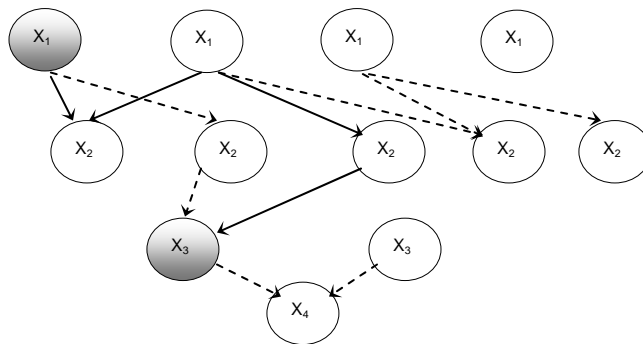


Figure 5.2 A sample CDN. Illustrate the scenario that requires twice Exclude Phase to return minimal cut set.

### Example 1:

Here are two trails between  $\langle X_{11}, X_{31} \rangle$ :

$$t_1: X_{11} \rightarrow X_{22} \rightarrow X_{31}$$

$$t_2: X_{11} \rightarrow X_{21} \leftarrow X_{12} \rightarrow X_{23} \rightarrow X_{31}$$

Case 1: Given evidence set  $Z=\{X_{22}\}$ , according to Definition 4, 5 and 6, along trail  $t_1: C_{11}=0, E_{11}=1, D_{11}=0$ . Hence, from Lemma 1,  $t_1(X_{11}, Z, X_{31}) = 1$ . The second trail  $t_2$  contains four intermediate nodes  $\{X_{21}, X_{12}, X_{23}, X_{31}\}$ . And,  $C_{21}=1, E_{21}=0, D_{11}=0$ ;  $C_{22}=0, E_{22}=0, D_{22}=0$ ;  $C_{23}=0, E_{23}=0, D_{23}=0$ ;  $C_{24}=0, E_{24}=0, D_{24}=0$ . Hence, by Lemma 1,  $t_2(X_{11}, Z, X_{31}) = 1$ . Finally, by Lemma 2,  $D(X_{11}, Z, X_{31})=1$ . Therefore,  $Z=\{X_{22}\}$  is  $\text{cut}_{\min}(X_{11}, X_{31})$ .

Case 2: Let evidence set  $Z=\{X_{22}, X_{23}\}$ , by applying Lemma 1 and Lemma 2, the same result is returned as  $D(X_{11}, Z, X_{31})=1$ . Therefore,  $Z=\{X_{22}, X_{23}\}$  is a cut set of  $(X_{11}, X_{31})$ . However  $Z=\{X_{22}, X_{23}\}$  isn't  $\text{cut}_{\min}(X_{11}, X_{31})$ .

## 5.4 Include-Exclude Algorithm for Structure Induction

As shown in Figure 5.1, the parent set of layer 3 variable is most difficult to be determined because of obscure casual relations between derivative component metrics and allocated component metrics. Include-Exclude algorithm is presented to

resolve the problem. This approach is similar to Grow-Shrink method in [7]. However, through Grow-Shrink method, Shrink phase is only performed once. While in our Include-Exclude method, the multiple Exclude phase are performed to guarantee that the minimal cut set is returned. An example will be presented late to show why multiple Excluding phase is desired.

#### **5.4.1 Introduction to Include-Exclude Algorithm**

##### **Including Phase:**

*Step 1*, initiate an empty evidence set  $E_0 = \{\}$ .

*Step 2*, each time, include a new second layer variable  $X_{2j}$  to create new evidence set  $E_j$ , go to step 3.

*Step 3*, test if  $\text{Ind}(X_{1i}, X_{3k} | E_j) = 1$ . If so, then stop the growing steps, and go to excluding phase. Otherwise, if  $\text{Ind}(X_{1i}, X_{3k} | E_j) = 0$ , go back to Step 2.

##### **Excluding Phase:**

*Step 1*, each time, delete a second layer  $X_{2j}$  variable from evidence set  $E_j$  to create new evidence set  $E_{j-1}$ . Go to step 2.

*Step 2*, test if  $\text{Ind}(X_{1i}, X_{3k} | E_{j-1}) = 1$ . If so, go to step 1, and continue to delete next second layer variable from evidence set  $E_{j-1}$ . Otherwise, if  $\text{Ind}(X_{1i}, X_{3k} | E_{j-1}) = 0$ , go to step 3, When all second layer variables are deleted once from evidence set, go to step 4.

*Step 3*, put back  $X_{2j}$  to evidence set  $E_{j-1}$  to restore evidence set  $E_j$ . And go to step 1.

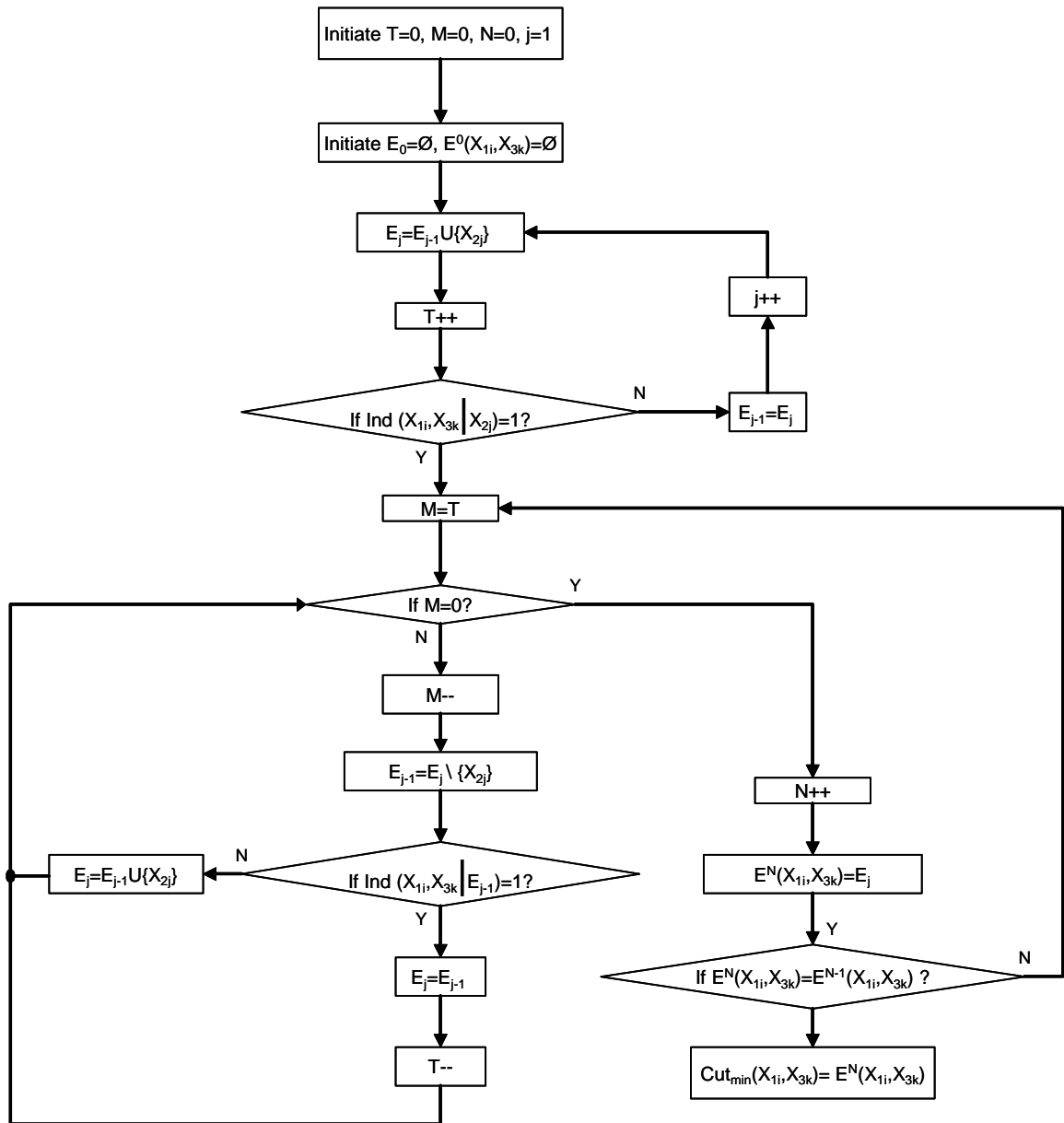


Figure 5.3 Flowchart for Include-Exclude Algorithm

*Step 4*, After all second layer variables are deleted once from evidence set to test the conditional independence, the minimal set for source destination pair  $\langle X_{1i}, X_{3k} \rangle$ , is returned. It's denoted as  $E^1(X_{1i}, X_{3k})$ .

*Step5*, Repeat step 1 to step 4,  $E^n(X_{1i}, X_{3k})$  is returned at the end of each repetition. When  $E^n(X_{1i}, X_{3k}) = E^{n-1}(X_{1i}, X_{3k})$ , the excluding phase is completed, and  $\text{cut}_{\min}(X_{1i}, X_{3k}) = E^n(X_{1i}, X_{3k})$ .

Upon completing Including Phase, the parent set  $\pi(X_{3k})$  is a subset of cut set  $E_j$ . Given  $E_j$ , the first layer variable  $X_{1i}$  and third layer variable  $X_{3k}$  will be conditionally independent to each other. However, we're looking for the minimal cut set providing this local independence in Component Independence Network. Therefore, Excluding Phase is developed to delete unnecessary elements in cut set to generate the minimal cut set for satisfying local independence between the first and third layer variable.

Step 5 in Excluding Phase is the crucial step that guarantees only minimal cut set  $\text{cut}_{\min}(X_{1i}, X_{3k})$  is returned at the end of Exclude-Include algorithm.

**Lemma 3:** Include-Exclude algorithm returns minimal cut set for a pair of variable  $\langle X_{1i}, X_{3k} \rangle$ .

Figure 5.2 is a good example to illustrate that multiple running of Exclude phase is necessary to return  $\text{cut}_{\min}(X_{11}, X_{31})$ . Source and destination pair  $\langle X_{11}, X_{31} \rangle$  is shaded in Figure 5.2.

As following, Include phase and Exclude phase are implemented to locate minimal evidence set  $\{X_{22}\}$ . Suppose that all independence test are Faithful, meaning that if

two variables  $\langle x, y \rangle$  are truly conditionally independent, given  $Z$ . Then the test result is  $\text{Ind}(x, y | Z) = 1$ , otherwise,  $\text{Ind}(x, y | Z) = 0$

***Including Phase:***

Initiate an empty set  $E_0 = \{\}$

Add in a second layer variable  $X_{21}$  to create new evidence set  $E_1 = \{X_{21}\}$ . Because  $\text{Ind}(X_{11}, X_{31} | E_1) = 0$  (by independence test), continue Include Phase.

Create new evidence set  $E_2 = \{X_{22}, X_{21}\}$  by inserting new second layer variable  $X_{22}$ . Since  $\text{Ind}(X_{11}, X_{31} | E_2) = 0$  (by independence test), continue Include Phase.

Continue to insert second layer variable  $X_{23}$  into new evidence set  $E_3 = \{X_{23}, X_{22}, X_{21}\}$ . Because  $\text{Ind}(X_{11}, X_{31} | E_3) = 1$ , stop Include Phase, and go to Exclude Phase.

At end of Include phase, three nodes are included in the returned cut set. The three nodes are  $X_{23}, X_{22}, X_{21}$ . They are shown in the below Figure 5.4 by highlighted red circle board line.

***Excluding Phase 1:***

1. As shown in Figure 5.4,  $X_{23}$  (highlighted in yellow) is deleted from  $E_3$ ,  $E_2 = \{X_{22}, X_{21}\}$ . Because  $\text{Ind}(X_{11}, X_{31} | E_2) = 0$ , restore  $E_3 = \{X_{23}, X_{22}, X_{21}\}$ .

2. As shown in Figure 5.5,  $X_{22}$  (highlighted in yellow) is deleted from  $E_3$ ,  $E_2 = \{X_{23}, X_{21}\}$ . Because  $\text{Ind}(X_{11}, X_{31} | E_2) = 0$ , restore  $E_3 = \{X_{23}, X_{22}, X_{21}\}$ .



3. As shown in Figure 5.6,  $X_{21}$ (highlighted in yellow) is deleted from  $E_3$ ,  $E_2=\{X_{23},X_{22}\}$ . Since  $\text{Ind}(X_{11}, X_{31} | E_2)=1$ , keep  $E_2=\{X_{23},X_{22}\}$ . And return  $E^1(X_{11}, X_{31})=\{X_{23},X_{22}\}$ .

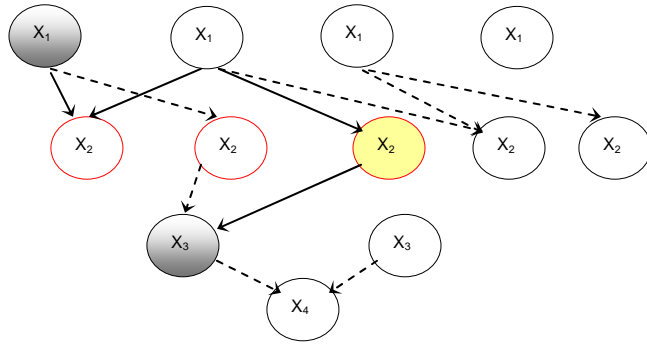


Figure 5.4 Step1 in Exclude phase for sample network

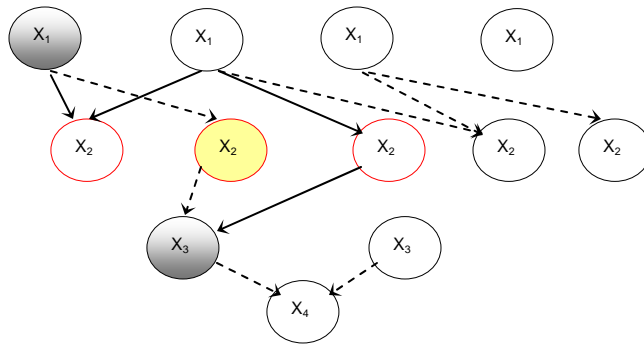


Figure 5.5 Step2 in Exclude phase for sample network

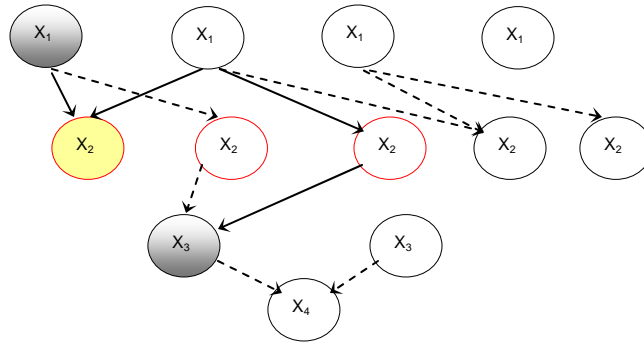


Figure 5.6 Step3 in Exclude phase for sample network

As shown in Example 1,  $\text{cut}_{\min}(X_{11}, X_{31}) = \{X_{22}\}$ . However, the evidence set returned by the first Exclude phase is  $\{X_{23}, X_{22}\}$ . The reason is that at step 1 of Exclude phase,  $X_{23}$  is deleted from the evidence set before the Collide node  $X_{21}$ . According to Theorem 1, the existence of Collide node  $X_{21}$  necessitates including normal node  $X_{23}$  in evidence set, in order to yield  $2^{\text{nd}}$  trail (marked as solid arrows in Figure 5.3) blocked. Since, we don't have a way to specify a good order for an unknown graphical structure (can't observe in advance which node is Collider, or not), the multiple Exclude phase is necessary to obtain the minimal cut set.

***Excluding phase 2:***

1. Delete  $X_{23}$  from  $E_2$ ,  $E_1 = \{X_{22}\}$ . Because  $\text{Ind}(X_{11}, X_{31} | E_1) = 1$ , return  $E^2(X_{11}, X_{31}) = \{X_{22}\}$ .

***Excluding phase 3:***

1. Delete  $X_{22}$  from  $E_2$ ,  $E_1 = \{\}$ . Because  $\text{Ind}(X_{11}, X_{31} | E_1) = 0$ , restore  $E_2(X_{11}, X_{31}) = \{X_{22}\}$ .

Since  $\text{Ind}(X_{11}, X_{31} | E_2) = 1$ , return  $E^3(X_{11}, X_{31}) = E_2 = \{X_{22}\}$ .

Since  $E^3(X_{11}, X_{31}) = E^2(X_{11}, X_{31})$ , Excluding Phase is stopped. And  $\text{cut}_{\min}(X_{11}, X_{31}) = E^3(X_{11}, X_{31}) = \{X_{22}\}$ .

#### **5.4.2 Proof of Include-Exclude Algorithm**

Include-Exclude Algorithm is introduced above. This algorithm is easy to be applied. However, we need to prove Include-Exclude Algorithm can return the minimal cut set, on the assumption that conditional independence can be tested correctly.

For Include phase, the proof is performed by the contradiction proof. We need to exam if the minimal cut set is a sub set of the cut set that returned at the end of Include phase.

For Exclude phase, the correctness is proved in different scenarios. We need to exam if the minimal cut set can be returned at the end of the Exclude phase. In other words, the redundant nodes returned by the Include phase will be deleted by the Exclude phase. Hence, at the end of Include-Exclude phase, only the minimal cut set is returned. Below is the proof.

**Lemma 4:** At end of Including Phase, the minimal cut set is the subset of returned cut set.

Proof by contradiction:

Suppose Minimal Cut Set is  $\text{cut}_{\min}(X_{1i}, X_{3k})$ . And the cut set returned by Inclusion process is  $E_I(X_{1i}, X_{3k})$ .

Suppose  $X_{2j} \in \text{cut}_{\min}(X_{1i}, X_{3k})$ , and  $X_{2j} \notin E_I(X_{1i}, X_{3k})$ .

If  $X_{2j}$  is the 2nd layer node and  $X_{1i}$  is one of the  $X_{2j}$ 's parent node. Here are only four types of 2nd layer nodes, according to the connectivity between  $X_{2j}$  and  $(X_{1i}, X_{3k})$ .

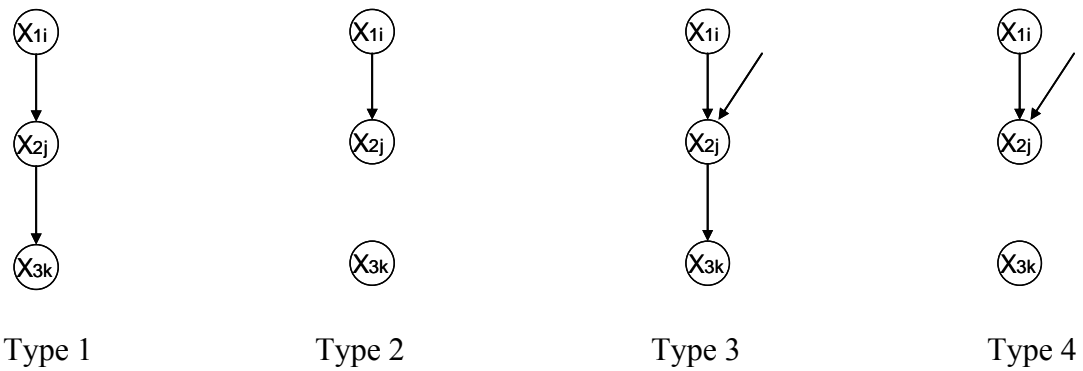


Figure 5.7 Four Types of Nodes on 2<sup>nd</sup> layer

Assume  $X_{2j} \notin E_I(X_{1i}, X_{3k})$ . Then, as shown in Figure 5.7,  $X_{2j}$  must be one of the following two types of 2<sup>nd</sup> layer nodes: Type 2 or Type 4.

However if  $X_{2j}$  is Type 2 or Type 4 node, then must have  $X_{2j} \notin \text{cut}_{\min}(X_{1i}, X_{3k})$ . This contradicts the assumption that  $X_{2j} \in \text{cut}_{\min}(X_{1i}, X_{3k})$ , and  $X_{2j} \notin E_I(X_{1i}, X_{3k})$ . So, if  $X_{2j} \in \text{cut}_{\min}(X_{1i}, X_{3k})$ , then must have  $X_{2j} \in E_I(X_{1i}, X_{3k})$ .

**Lemma 5:** Suppose at end of Excluding Phase, the returned minimal cut set is  $E_E(X_{1i}, X_{3k})$ . If  $X_{2j} \notin \text{cut}_{\min}(X_{1i}, X_{3k})$ , then  $X_{2j} \notin E_E(X_{1i}, X_{3k})$ . And, if  $X_{2j} \in \text{cut}_{\min}(X_{1i}, X_{3k})$ , then  $X_{2j} \in E_E(X_{1i}, X_{3k})$  after Excluding phase.

Before proving the Lemma 5, we first revisit the Definition for Collider-in-Evidence Indicator ( $\alpha$ ), Normal-in-Trail Indicator ( $\beta$ ), Collider-in-Trial Indicator ( $\gamma$ ) and Trail Blockage Indicator ( $t$ )

$\alpha = \begin{cases} 1 & \text{Along the } i\text{th trail, each Collider or its descendant in the evidence set } Z \\ 0 & \text{Along the } i\text{th trail, at least neither one Collider nor its descendant in the evidence set } Z \end{cases}$

$\beta = \begin{cases} 1 & \text{Along the } i\text{th trail, at least one normal node in the evidence set } Z \\ 0 & \text{Along the } i\text{th trail, none of normal nodes in the evidence set } Z \end{cases}$

$\gamma = \begin{cases} 1 & \text{Along the } i\text{th trail, at least one node is a Collider} \\ 0 & \text{Along the } i\text{th trail, all nodes are normal node} \end{cases}$

$$t_i(x, Z, y) = (\overline{\alpha_i} \cap \gamma_i) \cup \beta_i$$

$$D(x, Z, y) = \bigcap_{i=1}^L t_i(x, Z, y)$$

The Lemma 5 can be proved under different scenarios.

**Scenario 1**, If no pure collider node (the type 4 node shown in Figure 5.7) exists on any trail between source and destination pair  $\langle X_{1i}, X_{3k} \rangle$ . Suppose, at the end of Include procedure, Type1, 2 and 3 nodes are returned as elements of evidence set. Then, in Excluding phrase, we consider various trails containing Type 1, 2 and 3 nodes.

**Case1:** Suppose Type 1 node  $X_{2j}$  reside in evidence set returned by Including phase. In Excluding phase, we first delete  $X_{2j}$ , and test the conditional independence. Then,  $\gamma = 0$ ,  $\beta = 0$ .  $D(X_{1i}, Z \setminus X_{2j}, X_{3k}) = 0$ . Hence, Type 1 node  $z$  can't be deleted from evidence set.

**Case2:** Suppose Type2 node  $X_{2j}$  is returned in evidence set by Including phase. In Excluding phase, we delete  $X_{2j}$ . Since  $X_{2j}$  isn't on any trail,  $D(X_{1i}, Z \setminus X_{2j}, X_{3k}) = 1$ . Hence, Type2 node is removed from evidence set.

**Case3:** Suppose Type 3 node  $X_{2j}$  is returned in evidence set by Including phase. In Excluding phase, we first delete  $X_{2j}$ , and test the conditional independence. Then,  $\gamma = 0$ ,  $\beta = 0$ .  $D(X_{1i}, Z \setminus X_{2j}, X_{3k}) = 0$ . Hence, Type 3 node  $X_{2j}$  can't be deleted from evidence set.

As described in above scenarios, if here is no Type4 node returned in evidence set by Including phase, and if  $X_{2j} \notin \text{cut}_{\min}(X_{1i}, X_{3k})$ . then after Excluding phase  $X_{2j} \notin E_E(X_{1i}, X_{3k})$ . If  $X_{2j} \in \text{cut}_{\min}(X_{1i}, X_{3k})$ , then  $X_{2j} \in E_E(X_{1i}, X_{3k})$  after Excluding phase.

**Scenario 2**, Suppose at the end of Including phase, on a trail  $t$ , a set of pure collider node  $X_{2j}$  (the type 4 node shown in Figure 5.7) is returned in evidence set.

**Case1:** Suppose all pure collider nodes  $X_{2j}$  is deleted before normal node  $X_{2j}^N$ . Then when we delete  $X_{2j}^N$ ,  $\alpha = 0$ ,  $\beta = 0$ , and  $\gamma = 1$ ,  $D(X_{1i}, Z \setminus X_{2j}, X_{3k}) = 1$ .

**Case2:** Suppose all pure collider nodes  $X_{2j}$  is deleted after normal node  $X_{2j}^N$ . Then in the first round of Excluding phase, when we delete  $X_{2j}^N$ ,  $\alpha = 1$ ,  $\beta = 0$ , and  $\gamma = 1$ ,  $D(X_{1i}, Z \setminus X_{2j}, X_{3k}) = 0$ . Hence,  $X_{2j}^N$  has to remain in evidence set. Then, after deleting pure collider nodes  $X_{2j}$ , we have to repeat the Excluding phase. In the second round of Excluding phase, when we try to delete  $X_{2j}^N$ ,  $\alpha = 0$ ,  $\beta = 0$ , and  $\gamma = 1$ ,  $D(X_{1i}, Z \setminus X_{2j}^N, X_{3k}) = 1$ .

**Case3:** Suppose pure collider node  $X_{2j}$  is divided into two groups:  $X_{2j}^1$  and  $X_{2j}^2$ . Suppose  $X_{2j}^1$  is deleted before normal node  $X_{2j}^N$ . When delete  $X_{2j}^1$  from evidence set, since pure collider nodes  $X_{2j}^1$  is already out of evidence set, so  $\alpha = 0$ ,  $\beta = 0$ , and  $\gamma = 1$ ,  $D(X_{1i}, Z \setminus X_{2j}^1, X_{3k}) = 1$ . So, the  $X_{2j}^1$  can be deleted from evidence set. The left pure collider node  $X_{2j}^2$  deleted after normal node, and this case is already proved in Case2.

Proved by the above various scenarios, Excluding phase will delete second layer's node which isn't in the minimal cut set. Therefore, as a result of Including and Excluding phase, the cut set returned is the minimal cut set.

### 5.4.3 Parent Set Learning Algorithm

The following steps based on Include-Exclude algorithm is designed to determine parent set for third layer variable:

*Step1:* Fix a third layer variable  $X_{31}$

*Step2:* Determine  $\text{cut}_{\min}(X_{11}, X_{31})$  by Include-Exclude algorithms.

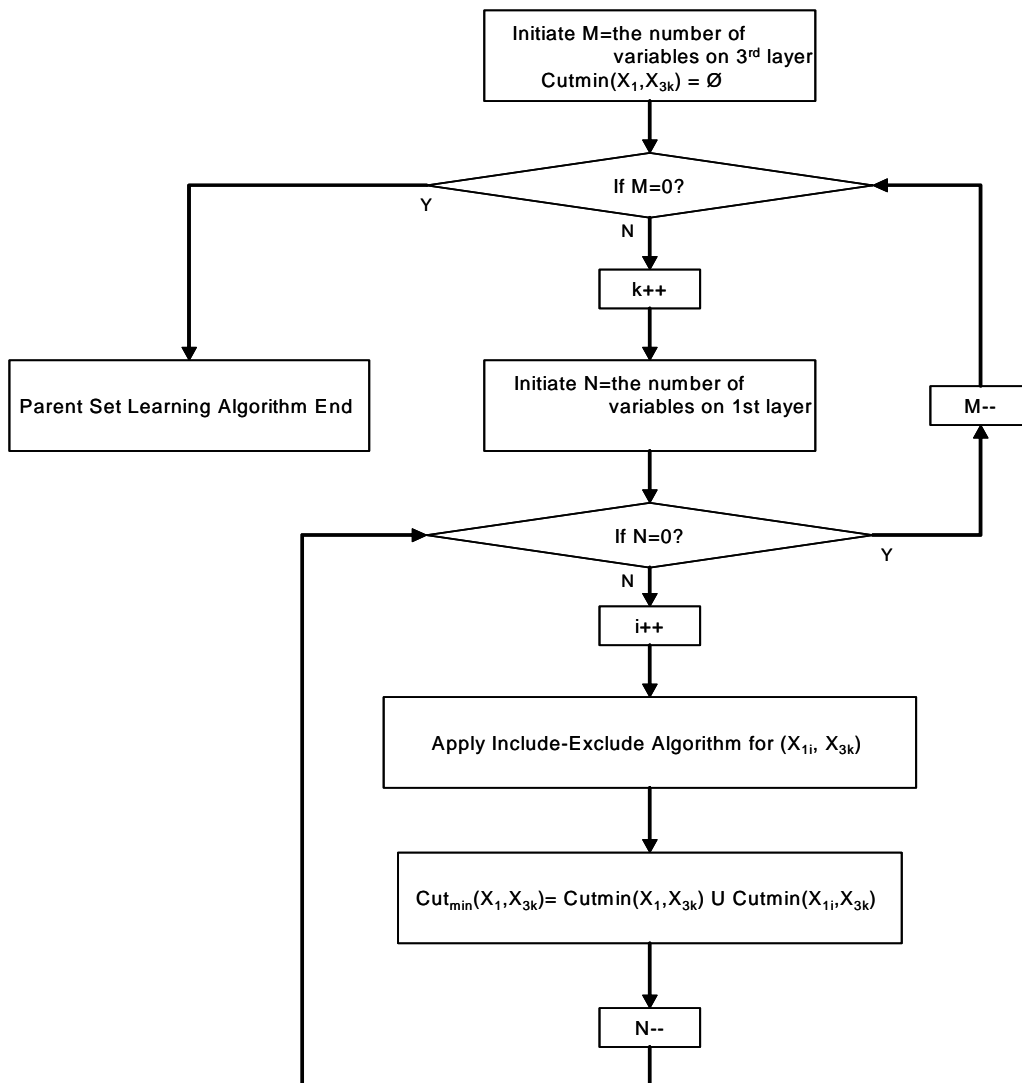


Figure 5.8 Flowchart for learning algorithm for 3<sup>rd</sup> layer's parent



*Step3:* After running Include-Exclude algorithms for all first layer variable  $\langle X_{11}, X_{31} \rangle, \langle X_{12}, X_{31} \rangle \dots \langle X_{1M}, X_{31} \rangle$ , the union of  $\text{cut}_{\min}(X_{1i}, X_{31})$  returned by each run

will be the complete parent set for  $X_{31}$  :  $\pi(X_{31}) = \bigcup_{i=1}^M \text{cut}_{\min}(X_{1i}, X_{31})$

*Step4:* Move to the next third layer variable  $X_{32}$ , and repeat steps 1, 2 and 3, until all third layer variables  $\{ X_{31}, X_{32} \dots X_{3N} \}$  find their parent set  $\pi(X_{31}), \pi(X_{32}) \dots \pi(X_{3N})$ .  $N$  is the number of variables on the third layer.

## 5.5 Statistical Methods in Structure Learning and Significant Component Learning

As shown in this chapter, the Include-Exclude algorithm is easy to be implemented, and the minimum parent set can be located. This is the first time that the component dependence network is designed, and the structure learning algorithm is proposed.

### 5.5.1 Statistical Methods in Structure Learning

During the structure learning, the conditional independence test is performed by partial correlation [26]. Partial correlation is a procedure that allows us to determine what the correlation between any two of the variables would be (hypothetically) if the third variable were held constant. The partial correlation of  $X$  and  $Y$ , with the effects of  $Z$  removed (or held constant), would be given by the formula:

$$r_{xy|z} = \frac{r_{xy} - (r_{xz}) \times (r_{yz})}{\text{sqrt}(1 - r_{xz}^2) \times \text{sqrt}(1 - r_{yz}^2)} \quad (5.6)$$

The above partial correlation is intensively used in structure learning for component dependence network. In the real implementation, it's hard to find the real partial correlation. The sample correlation can be calculated replace the true correlation used in algorithm 5.6. It's easy to calculate the statistical measure of partial correlation by following algorithm:

$$\hat{\rho}_{xy|z} = \frac{N \sum_{z=1}^N r_{xz} r_{yz} - \sum_{z=1}^N r_{xz} \sum_{z=1}^N r_{yz}}{\sqrt{N \sum_{z=1}^N r_{xz}^2 - (\sum_{z=1}^N r_{xz})^2} \sqrt{N \sum_{z=1}^N r_{yz}^2 - (\sum_{z=1}^N r_{yz})^2}} \quad (5.7)$$

In algorithm 5.7, N is the sample size of Z.

In searching structure for component dependence network, the maximum computation complexity can be measured as below:

$$\text{Maximum Computation Complexity} = \left[ \binom{M}{1} + \sum_{k=1}^M \binom{M}{k} - 1 \right] \times N \quad (5.8)$$

In the algorithm 5.8, M is the number of nodes presented on layer 2, and N is the number of nodes shown on layer 3. For example, if 4 nodes on 2<sup>nd</sup> layer, and 2 nodes on 3<sup>rd</sup> layer, then the maximum computation complexity is 36.

### 5.5.2 Statistical Methods in Significant Component Learning

After the structure is determined, a statistical method will be applied to calculate the significance of affection of component to the overall routing performance. Below is the description of the statistical method that applied to calculate the significant component.

First, the definition of entropy is introduced in [27]. Entropy is a functional of the distribution of X. The entropy of a discrete random variable X, given a condition value of y is defined as below:

$$H(X | Y = y) = - \sum_{x \in X} p(x | Y = y) \log p(x | Y = y) \quad (5.9)$$

From 5.9, the entropy  $H(X|Y=y)$  is a measure of uncertainty of a random variable X, when under the condition of  $Y=y$ .

Second, we define following as the conditional deviation:

$$\text{DEV}_{Y=y, Z=z}(X) = \sqrt{\sum_x P(X = x | Y = y, Z = z) \times (X - E_{Y=y, Z=z}(X))^2} \quad (5.10)$$

Third, we define the following as the variance of entropy  $H(X|Y=y)$  due to the change of Y:

$$\text{DEV}_{Y|Z=z, R=r}(H(X | Y = ?, Z = z, R = r)) \quad (5.11)$$

As shown in formula in (5.11), suppose three controlling variables Y, Z and R, then given Z and R, what's the attitude of change of  $H(X|Y,Z=z, R=r)$  due to the change of value of Y. If the deviation is higher across various values of Y, then Y has a greater impact on the certainty of X, when the other two variables Z and R are constant.

Fourth, the expectation of the variance of entropy on Y is defined as the significance indicator of Y on the variable X.

$$\text{Sig}(X | Y) = \text{Exp}_{Z,R} [\text{DEV}_{Y|Z=z,R=r} (H(X | Y, Z = z, R = r))] \quad (5.12)$$

By using an expectation value, the Y's affection on X will be averaged across the different combination of values of Z and R. The whole calculation procedure will be implemented as below:

*Step 1.* Calculate Conditional Probability Table based on the learned network Structure.

*Step 2.* For the variable on the forth layer  $X_{41}$ , calculate its conditional entropy with a fixed value of variable on one of its parent set. The conditional entropy will be calculated according to the formula 5.9. For example, two third layer variables  $X_{31}$  and  $X_{32}$  are included in the parent set of  $X_{41}$ . Then, to calculate the significance indicator of  $X_{31}$  on  $X_{41}$ , variable  $X_{32}$ 's value will be fixed. Then, the conditional entropy will be calculated based on different value of  $X_{31}$ .

$$H(X_{41} | X_{31}, X_{32} = x_{32}^1) = -\sum_{X_{41}} \{P(X_{41} | X_{31}, X_{32} = x_{32}^1) \cdot \text{Log}[P(X_{41} | X_{31}, X_{32} = x_{32}^1)]\}$$

*Step 3.* Calculate the conditional expectation of above conditional entropy, given the conditional probability distribution of  $P(X_{31}|X_{32}=x_{32}^1)$ .

$$E_{X_{31}|X_{32}=x_{32}^1} [H(X_{41} | X_{31}, X_{32} = x_{32}^1)] = \sum_{X_{31}} [P(X_{31} | X_{32} = x_{32}^1) \cdot H(X_{41} | X_{31}, X_{32} = x_{32}^1)]$$

*Step 4.* Calculate the conditional deviation of conditional entropy, given the conditional probability distribution of  $P(X_{31}|X_{32}=x_{32}^1)$ .

$$\begin{aligned}
& Dev_{X_{31}|X_{32}=x_{32}^1} [H(X_{41} | X_{31}, X_{32} = x_{32}^1)] \\
&= \sqrt{\sum_{X_{31}} \{P(X_{31} | X_{32} = x_{32}^1) \cdot [H(X_{41} | X_{31}, X_{32} = x_{32}^1) - E_{X_{31}|X_{32}=x_{32}^1} [H(X_{41} | X_{31}, X_{32} = x_{32}^1)]]\}^2}
\end{aligned}$$

Step 5. Change  $X_{32}$ 's value to  $x_{32}^2$ , and repeat step2, step3 and step4. Until all value of  $X_{32}$  is tried once. Then, following values will be achieved.

$$Dev_{X_{31}|X_{32}=x_{32}^i} [H(X_{41} | X_{31}, X_{32} = x_{32}^i)]$$

Step 6. Calculate the significance indicator of  $X_{41}$  on  $X_{31}$  as below:

$$\begin{aligned}
& Sig(X_{41} | X_{31}) = E_{X_{32}} [Dev_{X_{31}|X_{32}=x_{32}^i} [H(X_{41} | X_{31}, X_{32} = x_{32}^i)]] \\
&= \sum_{X_{32}=x_{32}^i} \{Dev_{X_{31}|X_{32}=x_{32}^i} [H(X_{41} | X_{31}, X_{32} = x_{32}^i)] \cdot P(X_{32} = x_{32}^i)\}
\end{aligned}$$

Step 7. Fix value of  $X_{31}$  instead of  $X_{32}$ , and Repeat Step 2 to Step6, and Calculate the significance indicator of  $X_{41}$  on  $X_{32}$  as below:

$$\begin{aligned}
& Sig(X_{41} | X_{32}) = E_{X_{31}} [Dev_{X_{32}|X_{31}=x_{31}^i} [H(X_{41} | X_{32}, X_{31} = x_{31}^i)]] \\
&= \sum_{X_{31}=x_{31}^i} \{Dev_{X_{32}|X_{31}=x_{31}^i} [H(X_{41} | X_{32}, X_{31} = x_{31}^i)] \cdot P(X_{31} = x_{31}^i)\}
\end{aligned}$$

Step 8. Compare  $Sig(X_{41}|X_{31})$  and  $Sig(X_{41}|X_{32})$ . If  $Sig(X_{41}|X_{31})$  is larger than  $Sig(X_{41}|X_{32})$ , then  $X_{31}$ 's significance indicator will be calculated according to each element in  $X_{31}$ 's parent set. For example, after repeating step 2 to step 7, if  $Sig(X_{31}|X_{21}) > Sig(X_{31}|X_{22}) > Sig(X_{31}|X_{23})$ , then  $X_{21}$  will be studied for its Significance indicator according to parent set  $\langle X_{11}, X_{12} \rangle$ . If  $Sig(X_{21}|X_{11}) > Sig(X_{21}|X_{12})$ , after repeating step2 to step 7, then  $X_{11}$  will be the significant component for the performance metric  $X_{41}$ .

In the real application for calculating the significance component for an overall routing performance metric, the significance indicator is calculated in the way of backward derivation. Take the example of four layer structure shown in Figure 5.2. We will firstly calculate the value of significance indicator between the third layer nodes and the fourth layer node. After locating the third layer node  $X_{3,max}$  which yields the maximum value of significance indicator with the fourth layer node, this  $X_{3,max}$  will be used to find the second layer node ( $X_{2,max}$ ) which has the maximum value of significance indicator with it. Then, after  $X_{2,max}$  is located, the final significant component is the one has the maximum value of significance indicator with  $X_{2,max}$ . This method intuitively agrees with human's derivation method. In order to find the root source for a bad performance, we need to firstly decompose the performance into allocated component metrics. Then, we will look into the derivative component metrics. Finally, by analyzing the routing mechanism, we can make a conclusion that why the routing performance is not satisfying. By applying this structure learning and significance indicator calculation, the human derivation can be replaced by a statistical learning. This is an innovative learning method for routing protocol's performance. An example will be used to show the power of the component dependence network in learning the impact of components to routing performance metrics.

## **5.6 Examples of Application of Component Dependence Network**

Two examples are showing here to show the capability of dependence network in determining significant component for routing protocol. In the first example, we need to figure out which component will impact the Data Packet Loss Ratio. In second example, significant component will be detected to show the impact on End-to-End delay.

All simulations in this study are performed by OPNET. The simulation scenario is the same as the settings in chapter 4.4. 20 mobile stations moved in a field of 2km x 2km. The simulation will study performance at different mobility speed of 0, 10, 20, 30, 40 and 50 meters per second. The Traffic model implemented is Data Traffic (640 bits/sec), Voice Traffic (2560 bits) and Video Traffic (20480 bits/sec). To categorize the level of mobility speed, speed under 10meters/second is Low, and, speed under 30meters/second is Middle, then speed between 30meters/second to 50meters/second is High.

The mobility model is Random Way Point. Each node selects a moving direction randomly, and stops for 1 second after each moving step. The simulation will study performance at different mobility speed of 0, 10, 20, 30, 40 and 50 meters per second. The Traffic model implemented is the raw packet generation provided by OPNET. Each station chooses a random destination to send out data packet at a “Packet Inter-Arrival Time” which is an exponential probability function changes as exponential (1).

To diversify simulation data, seed value changed from 1 to 50 for each scenario. And, each simulation will be run for 10 minutes.

For the first application, data packet loss ratio is a major target. For every 1 minute, a data packet loss ratio will be calculated as below:

#Data Received by destination/#Data sent from application layer

And, to detect actual data packet loss number due to path discovery, each station's distance to its neighbor will be monitored before a data packet sent through a returned path. When the distance is out of transmission range, if the link is a part of returned path, the path discovery will be counted as failure. Hence, even the path discovery will successfully return back a routing path, this is not definitely guarantee the path is a good path. Such as for DSR, some stale cached routing path can be returned back as a good path even it's already broken.

### **5.6.1 Application of component Dependence Network to detect significant component for Data Packet Loss Ratio**

#### **Step1: Structure of Component Dependence Network**

The first layer is component layer which contains four elements.



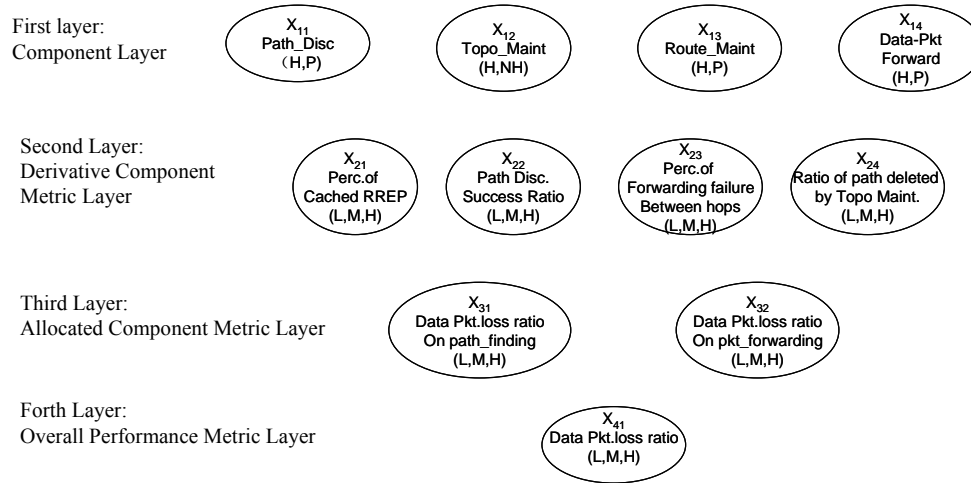


Figure 5.9 Layer Structure of Independence Network for Packet Loss Ratio learning

The second layer is derivative component metrics layer includes four elements. Each performance metric is introduced in chapter 3.5. The third layer is allocated component metric layer containing two elements. The only one element on the fourth layer is overall performance metric, which is data packet loss ratio.

As similar to Fig5.1, the  $X_{11}$ (Path Discovery Component) has two values H(Hop Based) and P(Path Based).  $X_{12}$  (Topology Maintenance Component) contains two values H(Hello Message) and NH(No Hello Message).  $X_{13}$  (Route Maintenance Component) has two values of H(Hop Based)and P(Path Based). And  $X_{14}$  (Data Packet Forwarding Component) has two values of H(Hop Based)and P(Path Based). For all other nodes representing ratio or percentage on 2<sup>nd</sup>, 3<sup>rd</sup> and 4<sup>th</sup> layers, values are the same as low, middle and high. Low means the ratio or percentage values are between 0 to 30%. Middle means the ratio or percentage values are between 30.001%

to 65%. High represents the ratio or percentage values that are between 65.001% to 100%.

**Step2: Structure Learning of Component Dependence Network**

The dependence network is learned for different combination scenarios of data density and mobility speed. As shown in following structure charts, the learning results agree with the previous simulation results. By applying Include-Exclude algorithm, the learned layer structure is shown in Fig.5.10. As reflected in the below chart, each node will be independent to non-descendant node, when its parent nodes are given.

The layer structure is the same for all combination of mobility speed and traffic

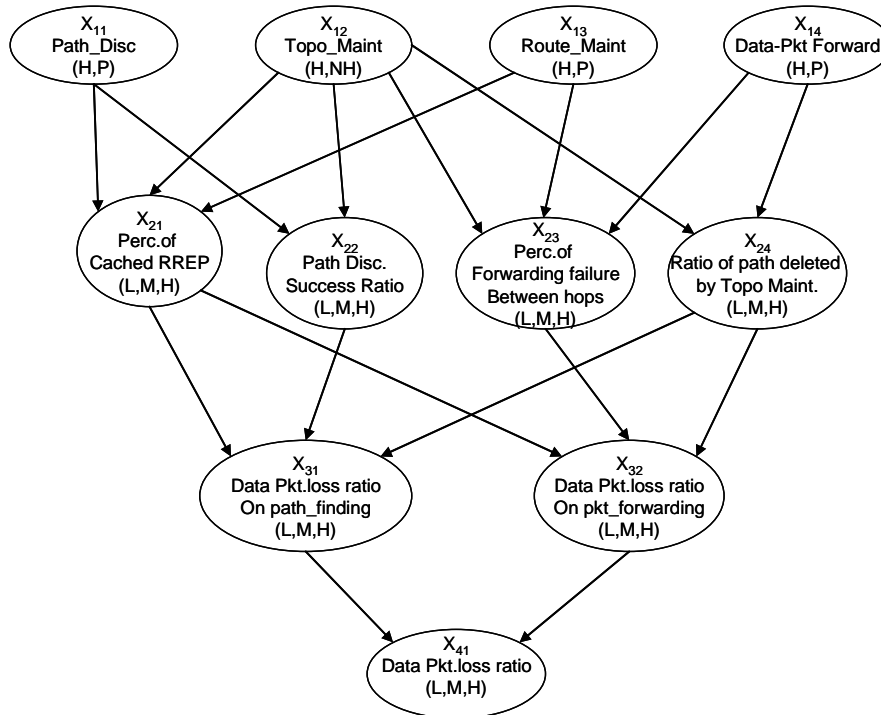


Figure 5.10 Layer Structure Obtained by Include-Exclude Algorithm

density. However, the conditional probability table will be different according to statistical analysis for simulation data.

**Step3: Significant Component Learning for  $X_{41}$**

The conditional probability table is listed below. For Data Traffic, and low mobility speed, below is the conditional probability of  $X_{41}$  given  $X_{31}$  and  $X_{32}$ . And the probability of  $X_{31}$  and  $X_{32}$  is also calculated.

$P(X_{41} X_{31}, X_{32})$ $X_{31}, X_{32}$	$P(X_{41}=L X_{31}, X_{32})$	$P(X_{41}=M X_{31}, X_{32})$	$P(X_{41}=H X_{31}, X_{32})$
$X_{31}=L, X_{32}=L$	0.991	0.009	0
$X_{31}=L, X_{32}=M$	0.899	0.101	0
$X_{31}=M, X_{32}=L$	0.543	0.457	0
$X_{31}=M, X_{32}=M$	0	1	0

Table 5.1 Table of Conditional Probability of  $X_{41}$  given  $X_{31}$  and  $X_{32}$  under Data Traffic and Low Mobility Speed

According to Table 5.1, and the algorithm of 5.9, the entropy of  $X_{41}$  given  $X_{31}$  and  $X_{32}$  is calculated in the table below:

$H(X_{41} X_{31}, X_{32})$ $X_{31}, X_{32}$	$H(X_{41} X_{31}=?, X_{32}=?)$
$X_{31}=L, X_{32}=L$	0.074
$X_{31}=L, X_{32}=M$	0.472
$X_{31}=M, X_{32}=L$	0.995
$X_{31}=M, X_{32}=M$	0

Table 5.2 Table of Entropy of  $X_{41}$  given  $X_{31}$  and  $X_{32}$  under Data Traffic and Low Mobility Speed

In order to calculate the significance indicator  $\text{Sig}(X_{41}|X_{31})$  and  $\text{Sig}(X_{41}|X_{32})$ , the probability distribution of  $X_{31}$  and  $X_{32}$  are provided below:

$X_{31}$ \ P( $X_{31}$ )	P( $X_{31}=?$ )
$X_{31}=L$	0.946
$X_{31}=M$	0.054

Table 5.3 Table of Probability Distribution of  $X_{31}$  under Data Traffic and Low Mobility Speed

$X_{32}$ \ P( $X_{32}$ )	P( $X_{32}=?$ )
$X_{32}=L$	0.980
$X_{32}=M$	0.020

Table 5.4 Table of Probability Distribution of  $X_{32}$  under Data Traffic and Low Mobility Speed

$X_{32}$ \ DEV	$\text{Dev}_{X_{31} X_{32}}[H(X_{41} X_{31}=?, X_{32})]$
$X_{32}=L$	0.209
$X_{32}=M$	0.122

Table 5.5 Table of Deviation of Entropy of  $X_{41}$  given  $X_{32}$  under Data Traffic and Low Mobility Speed

Then, according to algorithm defined in 5.11, deviation for  $H(X_{41}|X_{31}=L, X_{32}=L)$  and  $H(X_{41}|X_{31}=M, X_{32}=L)$  is calculated as above, and also the deviation for  $H(X_{41}|X_{31}=L, X_{32}=M)$  and  $H(X_{41}|X_{31}=M, X_{32}=M)$  is listed in Table 5.5.

Similar to the above calculation, according to algorithm defined in 5.11, the deviation for  $H(X_{41}|X_{31}=L, X_{32}=L)$  and  $H(X_{41}|X_{31}=M, X_{32}=L)$  is calculated as below. And, the deviation for  $H(X_{41}|X_{31}=L, X_{32}=M)$  and  $H(X_{41}|X_{31}=M, X_{32}=M)$

DEV $X_{31}$	$Dev_{X_{32} X_{31}}[H(X_{41} X_{31}, X_{32}=?)]$
$X_{31}=L$	0.055
$X_{31}=M$	0.158

Table 5.6 Table of Deviation of Entropy of  $X_{41}$  given  $X_{31}$  under Data Traffic and Low Mobility Speed

Then, according to 5.12, the significance indicator of  $X_{41}$  given  $X_{31}$  can be calculated as below:

$$\begin{aligned} \text{Sig}(X_{41} | X_{31}) &= P(X_{32} = L) \times Dev_{X_{31}|X_{32}=L}[H(X_{41} | X_{31} = ?, X_{32} = L)] \\ &\quad + P(X_{32} = M) \times Dev_{X_{31}|X_{32}=M}[H(X_{41} | X_{31} = ?, X_{32} = M)] \\ &= 0.207 \end{aligned}$$

In the above formula to calculate the significance indicator of  $X_{41}$  given  $X_{31}$ , the probability distribution of  $X_{31}$  can be read from Table 5.4.

The same, the significance indicator of  $X_{41}$  given  $X_{32}$  can be calculated as below:

$$\begin{aligned} \text{Sig}(X_{41} | X_{32}) &= P(X_{31} = L) \times \text{Dev}_{X_{32}|X_{31}=L}[H(X_{41} | X_{31} = L, X_{32} = ?)] \\ &\quad + P(X_{31} = M) \times \text{Dev}_{X_{32}|X_{31}=M}[H(X_{41} | X_{31} = M, X_{32} = ?)] \\ &= 0.061 \end{aligned}$$

In the above formula to calculate the significance indicator of  $X_{41}$  given  $X_{32}$ , the probability distribution of  $X_{32}$  can be read from Table5.3.

Since  $\text{Sig}(X_{41}|X_{31})$  is larger than  $\text{Sig}(X_{41}|X_{32})$ , it's shown that the  $X_{31}$  has more impact on  $X_{41}$ 's performance. Then, in the next step, we need to focus on  $X_{31}$ , and find out which element on the second layer will affect the  $X_{31}$ 's performance most.

$P(X_{31} X_{21}, X_{22}, X_{24})$ $X_{21}, X_{22}, X_{24}$	$P(X_{31}=L X_{21}, X_{22}, X_{24})$	$P(X_{31}=M X_{21}, X_{22}, X_{24})$	$P(X_{31}=H X_{21}, X_{22}, X_{24})$
$X_{21}=H, X_{22}=H, X_{24}=L$	0.987	0.013	0
$X_{21}=H, X_{22}=H, X_{24}=M$	0.952	0.048	0
$X_{21}=H, X_{22}=H, X_{24}=H$	0.907	0.093	0
$X_{21}=M, X_{22}=H, X_{24}=L$	0.961	0.039	0
$X_{21}=M, X_{22}=H, X_{24}=M$	0.927	0.073	0
$X_{21}=M, X_{22}=H, X_{24}=H$	0.899	0.101	0
$X_{21}=H, X_{22}=M, X_{24}=L$	0.934	0.066	0
$X_{21}=H, X_{22}=M, X_{24}=M$	0.886	0.114	0
$X_{21}=H, X_{22}=M, X_{24}=H$	0.858	0.142	0
$X_{21}=M, X_{22}=M, X_{24}=L$	0.902	0.098	0
$X_{21}=M, X_{22}=M, X_{24}=M$	0.855	0.145	0
$X_{21}=M, X_{22}=M, X_{24}=H$	0.82	0.18	0

Table 5.7 Table of Conditional Probability of  $X_{31}$  given  $X_{21}$ ,  $X_{22}$  and  $X_{24}$  under Data Traffic and Low Mobility Speed

#### Step4: Significant Component Learning for $X_{31}$

The conditional probability table is listed in Table5.7 for  $X_{31}$ . For Data Traffic, and low mobility speed, the Table5.7 is a table of the conditional probability of  $X_{31}$  given  $X_{21}$ ,  $X_{22}$  and  $X_{24}$ . Then, by calculating the entropy of  $X_{31}$  given  $X_{21}$ ,  $X_{22}$  and  $X_{24}$ , Table5.8 is generated below.

$H(X_{31}   X_{21}, X_{22}, X_{24})$	$H(X_{31}   X_{21}=?, X_{22}=?, X_{24}=?)$
$X_{21}, X_{22}, X_{24}$	
$X_{21}=H, X_{22}=H, X_{24}=L$	0.100
$X_{21}=H, X_{22}=H, X_{24}=M$	0.278
$X_{21}=H, X_{22}=H, X_{24}=M$	0.446
$X_{21}=M, X_{22}=H, X_{24}=L$	0.238
$X_{21}=M, X_{22}=H, X_{24}=M$	0.377
$X_{21}=M, X_{22}=H, X_{24}=L$	0.472
$X_{21}=H, X_{22}=M, X_{24}=L$	0.351
$X_{21}=H, X_{22}=M, X_{24}=M$	0.512
$X_{21}=H, X_{22}=M, X_{24}=H$	0.589
$X_{21}=M, X_{22}=M, X_{24}=L$	0.463
$X_{21}=M, X_{22}=M, X_{24}=M$	0.597
$X_{21}=M, X_{22}=M, X_{24}=H$	0.680

Table 5.8 Table of Entropy of  $X_{31}$  given  $X_{21}$ ,  $X_{22}$  and  $X_{24}$  under Data Traffic and Low Mobility Speed

Since here are three controlling variables for  $X_{31}$ , in order to calculate individually how much each parent element will impact performance of  $X_{31}$ , the joint probability distribution are given below for various combination of  $(X_{22}, X_{24})$ ,  $(X_{21}, X_{24})$  and  $(X_{21}, X_{22})$ .

$P(X_{22}, X_{24})$ $X_{22}, X_{24}$	$P(X_{22}=?, X_{24}=?)$
$X_{22}=H, X_{24}=L$	0.49075
$X_{22}=H, X_{24}=M$	0.21714
$X_{22}=H, X_{24}=H$	0.23061
$X_{22}=M, X_{24}=L$	0.00925
$X_{22}=M, X_{24}=M$	0.02561
$X_{22}=M, X_{24}=H$	0.02664

Table 5.9 Table of Joint Probability Distribution of  $X_{22}$  and  $X_{24}$  under Data Traffic and Low Mobility Speed

$P(X_{21}, X_{24})$ $X_{21}, X_{24}$	$P(X_{21}=?, X_{24}=?)$
$X_{21}=H, X_{24}=L$	0.34750
$X_{21}=H, X_{24}=M$	0.10297
$X_{21}=H, X_{24}=H$	0.10503
$X_{21}=M, X_{24}=L$	0.15250
$X_{21}=M, X_{24}=M$	0.13978
$X_{21}=M, X_{24}=H$	0.15222

Table 5.10 Table of Joint Probability Distribution of  $X_{21}$  and  $X_{24}$  under Data Traffic and Low Mobility Speed

$P(X_{21}, X_{22})$ $X_{21}, X_{22}$	$P(X_{21}=?, X_{22}=?)$
$X_{21}=H, X_{22}=H$	0.52466
$X_{21}=M, X_{22}=H$	0.41384
$X_{21}=H, X_{22}=M$	0.03084
$X_{21}=M, X_{22}=M$	0.03066

Table 5.11 Table of Joint Probability Distribution of  $X_{21}$  and  $X_{22}$  under Data Traffic and Low Mobility Speed



Then, according to algorithm defined in 5.11, the deviation for the entropy of  $X_{31}$  under fixed values of  $X_{22}$  and  $X_{24}$  is calculated as above, and also the deviation for for different values of  $X_{21}$ , based on the different combination of values of  $X_{22}$  and  $X_{24}$ , are listed below in Table 5.12.

Dev $X_{22}, X_{24}$	$Dev_{X_{21} X_{22}, X_{24}}[H(X_{31} X_{21}=?, X_{22}, X_{24})]$
$X_{22}=H, X_{24}=L$	0.06347
$X_{22}=H, X_{24}=M$	0.04896
$X_{22}=H, X_{24}=H$	0.01264
$X_{22}=M, X_{24}=L$	0.04496
$X_{22}=M, X_{24}=M$	0.04250
$X_{22}=M, X_{24}=H$	0.04500

Table 5.12 Table of Deviation of Entropy of  $X_{31}$  given  $X_{22}$  and  $X_{24}$  under Data Traffic and Low Mobility Speed

Similar to Table 5.12, the deviation of Entropy of  $X_{31}$  given  $X_{21}$  and  $X_{24}$  is calculated below in Table 5.13.

Dev $X_{21}, X_{24}$	$Dev_{X_{22} X_{21}, X_{24}}[H(X_{31} X_{21}, X_{22}=?, X_{24})]$
$X_{21}=H, X_{24}=L$	0.03614
$X_{21}=H, X_{24}=M$	0.07427
$X_{21}=H, X_{24}=H$	0.04512
$X_{21}=M, X_{24}=L$	0.02480
$X_{21}=M, X_{24}=M$	0.06590
$X_{21}=M, X_{24}=H$	0.06174

Table 5.13 Table of Deviation of Entropy of  $X_{31}$  given  $X_{21}$  and  $X_{24}$  under Data Traffic and Low Mobility Speed

Table 5.12 and Table 5.13 are calculated to determine how much  $X_{21}$  and  $X_{22}$  will affect the changing of uncertainty of  $X_{31}$ . In order to determine the impact of  $X_{24}$ , the deviation of Entropy of  $X_{31}$  given  $X_{21}$  and  $X_{22}$  is calculated below in Table 5.14.

Dev $X_{21}, X_{22}$	$Dev_{X_{24} X_{21}, X_{22}}[H(X_{31} X_{21}, X_{22}, X_{24}=?)]$
$X_{21}=H, X_{22}=H$	0.13514
$X_{21}=M, X_{22}=H$	0.09851
$X_{21}=H, X_{22}=M$	0.09175
$X_{21}=M, X_{22}=M$	0.05847

Table 5.14 Table of Deviation of Entropy of  $X_{31}$  given  $X_{21}$  and  $X_{22}$  under Data Traffic and Low Mobility Speed

Then, according to algorithm 5.12, the significance indicator of  $X_{31}$  given  $X_{21}$  can be calculated as below:

$$\begin{aligned}
 \text{Sig}(X_{31} | X_{21}) &= P(X_{22} = H, X_{24} = L) \times Dev_{X_{21}|X_{22}=H, X_{24}=L} [H(X_{31} | X_{21} = ?, X_{22} = H, X_{24} = L)] \\
 &+ P(X_{22} = H, X_{24} = M) \times Dev_{X_{21}|X_{22}=H, X_{24}=M} [H(X_{31} | X_{21} = ?, X_{22} = H, X_{24} = M)] \\
 &+ P(X_{22} = H, X_{24} = H) \times Dev_{X_{21}|X_{22}=H, X_{24}=H} [H(X_{31} | X_{21} = ?, X_{22} = H, X_{24} = H)] \\
 &+ P(X_{22} = M, X_{24} = L) \times Dev_{X_{21}|X_{22}=M, X_{24}=L} [H(X_{31} | X_{21} = ?, X_{22} = M, X_{24} = L)] \\
 &+ P(X_{22} = M, X_{24} = M) \times Dev_{X_{21}|X_{22}=M, X_{24}=M} [H(X_{31} | X_{21} = ?, X_{22} = M, X_{24} = M)] \\
 &+ P(X_{22} = M, X_{24} = H) \times Dev_{X_{21}|X_{22}=M, X_{24}=H} [H(X_{31} | X_{21} = ?, X_{22} = M, X_{24} = H)] \\
 &= 0.0474
 \end{aligned}$$

Similar to the above method, the the significance indicator of  $X_{31}$  given  $X_{21}$  can be calculated in below algorithm:

$$\begin{aligned}
\text{Sig}(X_{31} | X_{22}) &= P(X_{21} = H, X_{24} = L) \times \text{Dev}_{X_{22}|X_{21}=H, X_{24}=L} [H(X_{31} | X_{22} = ?, X_{21} = H, X_{24} = L)] \\
&+ P(X_{21} = H, X_{24} = M) \times \text{Dev}_{X_{22}|X_{21}=H, X_{24}=M} [H(X_{31} | X_{22} = ?, X_{21} = H, X_{24} = M)] \\
&+ P(X_{21} = H, X_{24} = H) \times \text{Dev}_{X_{22}|X_{21}=H, X_{24}=H} [H(X_{31} | X_{22} = ?, X_{21} = H, X_{24} = H)] \\
&+ P(X_{21} = M, X_{24} = L) \times \text{Dev}_{X_{22}|X_{21}=M, X_{24}=L} [H(X_{31} | X_{22} = ?, X_{21} = M, X_{24} = L)] \\
&+ P(X_{21} = M, X_{24} = M) \times \text{Dev}_{X_{22}|X_{21}=M, X_{24}=M} [H(X_{31} | X_{22} = ?, X_{21} = M, X_{24} = M)] \\
&+ P(X_{21} = M, X_{24} = H) \times \text{Dev}_{X_{22}|X_{21}=M, X_{24}=H} [H(X_{31} | X_{22} = ?, X_{21} = M, X_{24} = H)] \\
&= 0.0473
\end{aligned}$$

$$\begin{aligned}
\text{Sig}(X_{31} | X_{24}) &= P(X_{21} = H, X_{22} = H) \times \text{Dev}_{X_{24}|X_{21}=H, X_{22}=H} [H(X_{31} | X_{24} = ?, X_{21} = H, X_{22} = H)] \\
&+ P(X_{21} = M, X_{22} = H) \times \text{Dev}_{X_{24}|X_{21}=M, X_{22}=H} [H(X_{31} | X_{24} = ?, X_{21} = M, X_{22} = H)] \\
&+ P(X_{21} = H, X_{22} = M) \times \text{Dev}_{X_{24}|X_{21}=H, X_{22}=M} [H(X_{31} | X_{24} = ?, X_{21} = H, X_{22} = M)] \\
&+ P(X_{21} = M, X_{22} = M) \times \text{Dev}_{X_{24}|X_{21}=M, X_{22}=M} [H(X_{31} | X_{24} = ?, X_{21} = M, X_{22} = M)] \\
&= 0.1163
\end{aligned}$$

Comparing significance indicator  $\text{Sig}(X_{31}|X_{24})$  is larger than  $\text{Sig}(X_{31}|X_{21})$  and  $\text{Sig}(X_{31}|X_{22})$ , the  $X_{24}$  has the largest impact on  $X_{31}$ 's performance. In the next step,  $X_{24}$ 's performance will be studied to determine which parent element has the most significant impact on its uncertainty.

#### **Step5: Significant Component Learning for $X_{24}$**

The conditional probability of  $X_{24}$  given its parent set is listed in below table.

$P(X_{24} X_{12}, X_{14})$ $X_{12}, X_{14}$	$P(X_{24}=L X_{12}, X_{14})$	$P(X_{24}=M X_{12}, X_{14})$	$P(X_{24}=H X_{12}, X_{14})$
$X_{12}=H, X_{14}=H$	0	0.462	0.538
$X_{12}=H, X_{14}=P$	0	0.509	0.491
$X_{12}=N, X_{14}=H$	1	0	0
$X_{12}=N, X_{14}=P$	1	0	0

Table 5.15 Table of Conditional Probability of  $X_{24}$  given  $X_{12}$  and  $X_{14}$  under Data Traffic and Low Mobility Speed

Then, by calculating the entropy of  $X_{24}$  given  $X_{12}$  and  $X_{14}$ , Table 5.16 is generated below.

$H(X_{24} X_{12}, X_{14})$ $X_{12}, X_{14}$	$H(X_{24} X_{12}=?, X_{14}=?)$
$X_{12}=H, X_{14}=H$	0.996
$X_{12}=H, X_{14}=P$	1.000
$X_{12}=N, X_{14}=H$	0
$X_{12}=N, X_{14}=P$	0

Table 5.16 Table of Entropy of  $X_{24}$  given  $X_{12}$  and  $X_{14}$  under Data Traffic and Low Mobility Speed

As controlled by simulation results, the  $P(X_{12}=N)=0.5$ ,  $P(X_{12}=H)=0.5$  and  $P(X_{14}=H)=0.5$ ,  $P(X_{14}=P)=0.5$ .

Then, according to algorithm 5.12, the significance indicator of  $X_{24}$  given  $X_{12}$  can be calculated as below:

$$\begin{aligned}\text{Sig}(X_{24} | X_{12}) &= P(X_{14} = H) \times \text{Dev}_{X_{12}|X_{14}=H}[H(X_{24} | X_{12} = ?, X_{14} = H)] \\ &\quad + P(X_{14} = P) \times \text{Dev}_{X_{12}|X_{14}=P}[H(X_{24} | X_{12} = ?, X_{14} = P)] \\ &= 0.7056\end{aligned}$$

Similarly, the significance indicator of  $X_{24}$  given  $X_{14}$  is generated below:

$$\begin{aligned}\text{Sig}(X_{24} | X_{14}) &= P(X_{12} = H) \times \text{Dev}_{X_{14}|X_{12}=H}[H(X_{24} | X_{14} = ?, X_{12} = H)] \\ &\quad + P(X_{12} = N) \times \text{Dev}_{X_{14}|X_{12}=N}[H(X_{24} | X_{14} = ?, X_{12} = N)] \\ &= 0.0014\end{aligned}$$

By comparing the values of significance indicator for giving  $X_{12}$  and  $X_{14}$ , the impact of  $X_{12}$  is much higher than  $X_{14}$ .  $X_{12}$  represents the routing component of “Topology Database Maintenance Component”. This conclusion learned from quantitative analysis agrees with the analysis based on routing mechanism. Therefore, this is a good example to show the power of the component dependence network in detecting the significant component for a routing performance metric. And, illustrated by above example, once the dependence network structure is determined, the significance indicator can be calculated in low computation complexity.

#### **Step6: Parameter Tuning for Significant Component**

The advantage of the component dependence network is that by the quantitative analysis, the significant component can be detected. This self learning method can free the people from analyzing simulation results and making decision based on statistics. And the other advantage of component dependence network is that the significant component can be located, and then we can concentrate the study on this significant component, and make significant parameter tuned. Otherwise, it's very

difficult to determine which parameter should be tuned in order to have a better performance.

After locating the significant component, we need to tune the parameter to find a best value to keep the balance between two routing performance metrics. For example, this is shown in simulation results that frequency of Topology Maintenance's Hello message will affect both data packet loss ratio and routing overhead. Usually, higher is the frequency, lower is the data packet loss ratio and higher is the routing overhead. Then, the question is how to find a right value of Hello message's frequency to meet the requirement of both data packet loss ratio and routing overhead.

For example, for a video traffic with high mobility speed, after the significant component determined as Topology Maintenance Component, the routing overhead and data packet loss ratio will be studied along with different values of frequency of Hello message. The simulation result is listed as below in Figure 5.11.

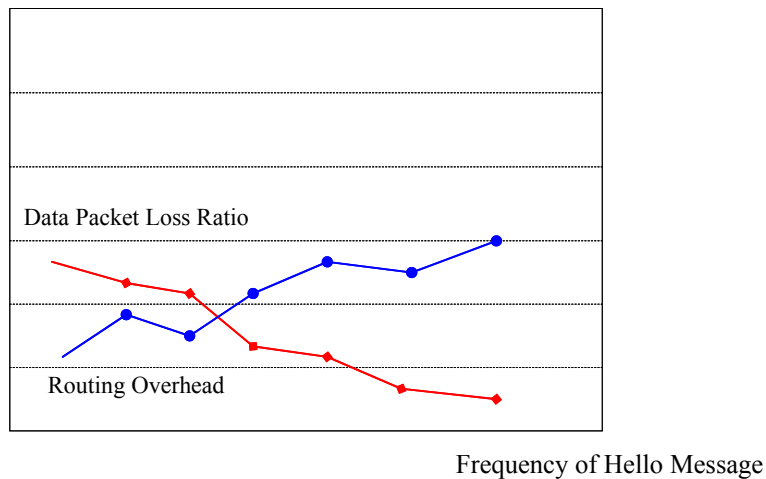


Figure 5.11 Chart of Data Packet Loss Ratio and Routing Overhead against the Frequency of Hello Message

As shown in above figure, higher is the frequency of Hello Message, higher is the routing overhead and lower is the Data Packet Loss Ratio. Then, what's the best frequency of Hello Message if both performance metrics are to be considered? This will be determined by the weight given to different performance metrics. If data packet loss ratio is the more important than routing overhead, then frequency of Hello message can be higher. Otherwise, if the routing overhead is the most critical thing in consideration, then the frequency of Hello message should be lower.

In conclusion, once the significant component is determined, the parameter of the significant component can be tuned accordingly regarding the performance requirement. This is much more applicable than blindly tuning all parameters for all components.

## **5.6.2 Application of component Dependence Network to detect significant component for End to End Delay**

### **Step1: Structure of Component Dependence Network**

The first layer is component layer which contains four elements.

The second layer is derivative component metrics layer includes four elements. Each performance metric is introduced in chapter 3.5. The third layer is allocated component metric layer containing two elements. The only one element on the fourth layer is overall performance metric, which is data packet's End to End Delay.

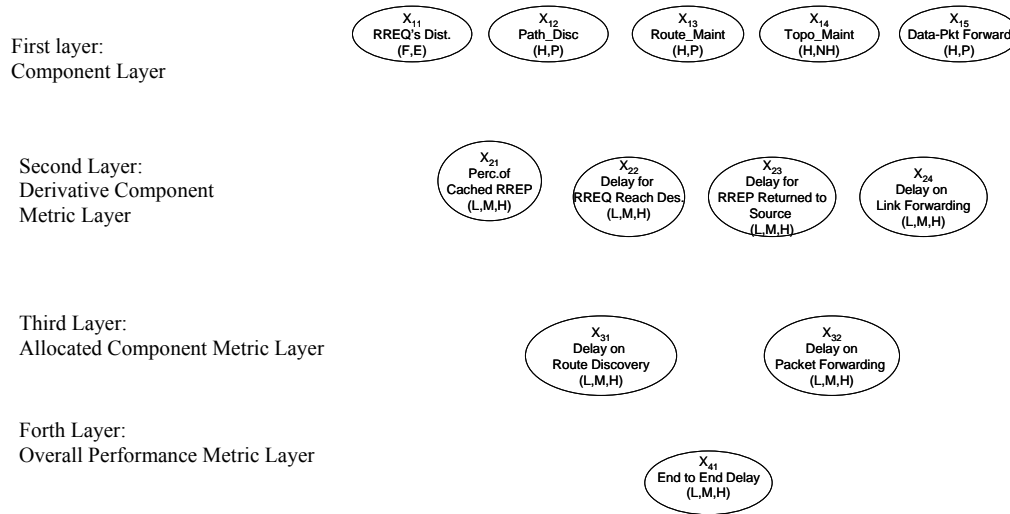


Figure 5.12 Layer Structure of Independence Network for End to End Delay learning

As similar to Fig5.1, the  $X_{11}$ (RREQ's distribution method) has two values F(Fixed Ring) and E(Expanding Ring). For  $X_{12}$  (Path Discovery Component), two values are presented: H(Hop based) and P(Path Based).  $X_{13}$  (Route Maintenance Component) contains two values H(Hop based) and P(Path Based).  $X_{14}$  (Topology Maintenance Component) has two values of H(Hello Based)and NH(NonHello). And  $X_{15}$  (Data Packet Forwarding Component) has two values of H(Hop Based)and P(Path Based). For all other nodes representing ratio or percentage on 2<sup>nd</sup>, 3<sup>rd</sup> and 4<sup>th</sup> layers, values are the same as low, middle and high. Low means data packet's end to end delay time is less than 0.3 second. Middle means data packet's end to end delay time is between 0.3 and 0.6 second. High represents the data packet's end to end delay that is above 0.6 second.



The simulation is accomplished by OPNET. The simulation scenario is the same as the settings in chapter 4.4. 20 mobile stations moved in a field of 2km x 2km. The simulation will study performance at different mobility speed of 0, 10, 20, 30, 40 and 50 meters per second. The Traffic model implemented is Data Traffic (640 bits/sec), Voice Traffic (2560 bits) and Video Traffic (20480 bits/sec). To categorize the level of mobility speed, speed under 10meters/second is Low, and, speed under 30meters/second is Middle, then speed between 30meters/second to 50meters/second is High.

## **Step2: Structure Learning of Component Dependence Network**

The dependence network is learned for different combination scenarios of data density and mobility speed. As shown in following structure charts, the learning results agree with the previous simulation results. By applying Include-Exclude algorithm, the learned layer structure is shown in Fig.5.13. As reflected in the below chart, each node will be independent to non-descendant node, when its parent nodes are determined.

The layer structure is the same for all combination of mobility speed and traffic density. However, the conditional probability table will be different according to statistical analysis for simulation data.

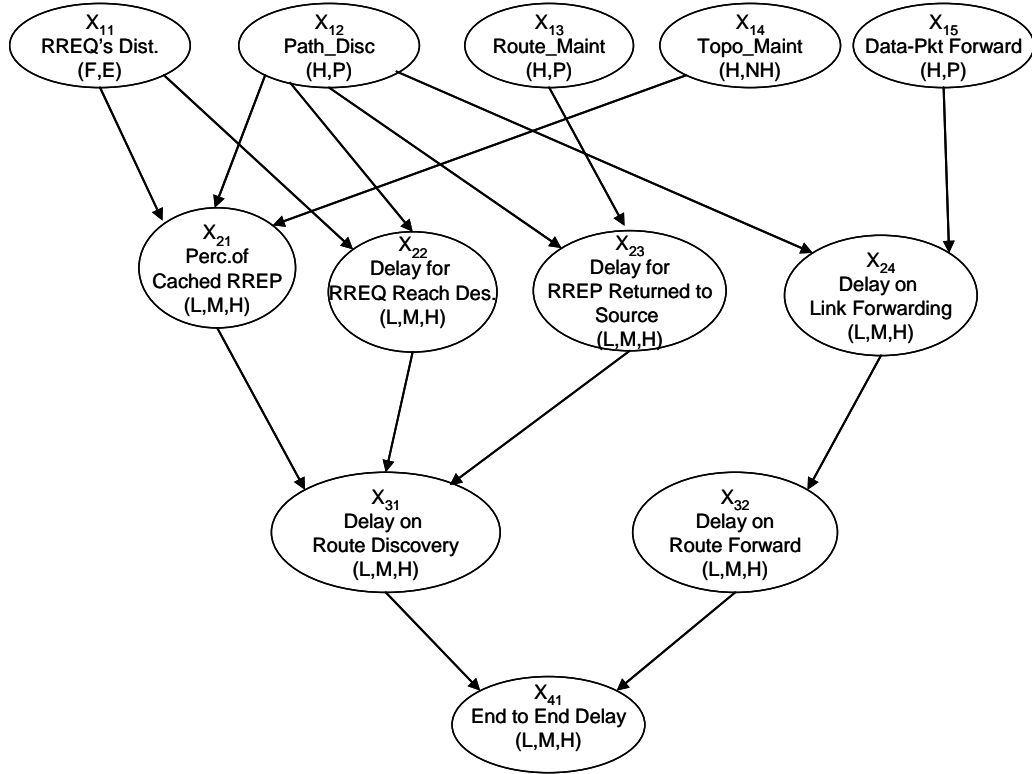


Figure 5.13 Layer Structure Obtained by Include-Exclude Algorithm

**Step3: Significant Component Learning for  $X_{41}$**

The conditional probability table is listed below. For Data Traffic, and low mobility speed, below is the conditional probability of  $X_{41}$  given  $X_{31}$  and  $X_{32}$ . And the probability of  $X_{31}$  and  $X_{32}$  is also calculated.

$P(X_{41} X_{31}, X_{32})$ $X_{31}, X_{32}$	$P(X_{41}=L X_{31}, X_{32})$	$P(X_{41}=M X_{31}, X_{32})$	$P(X_{41}=H X_{31}, X_{32})$
$X_{31}=L, X_{32}=L$	0.781	0.219	0
$X_{31}=L, X_{32}=M$	0.748	0.252	0
$X_{31}=M, X_{32}=L$	0.327	0.673	0
$X_{31}=M, X_{32}=M$	0.269	0.731	0

Table 5.17 Table of Conditional Probability of  $X_{41}$  given  $X_{31}$  and  $X_{32}$  under Data Traffic and Low Mobility Speed

According to Table 5.17, and the algorithm of 5.9, the entropy of  $X_{41}$  given  $X_{31}$  and  $X_{32}$  is calculated in the table below:

$H(X_{41} X_{31}, X_{32})$ $X_{31}, X_{32}$	$H(X_{41} X_{31}=?, X_{32}=?)$
$X_{31}=L, X_{32}=L$	0.758
$X_{31}=L, X_{32}=M$	0.814
$X_{31}=M, X_{32}=L$	0.912
$X_{31}=M, X_{32}=M$	0.061

Table 5.18 Table of Entropy of  $X_{41}$  given  $X_{31}$  and  $X_{32}$  under Data Traffic and Low Mobility Speed

In order to calculate the significance indicator  $\text{Sig}(X_{41}|X_{31})$  and  $\text{Sig}(X_{41}|X_{32})$ , the probability distribution of  $X_{31}$  and  $X_{32}$  are provided below:

$X_{31} \backslash P(X_{31})$	$P(X_{31}=?)$
$X_{31}=L$	0.695
$X_{31}=M$	0.305

Table 5.19 Table of Probability Distribution of  $X_{31}$  under Data Traffic and Low Mobility Speed

$X_{32} \backslash P(X_{32})$	$P(X_{32}=?)$
$X_{32}=L$	0.481
$X_{32}=M$	0.519

Table 5.20 Table of Probability Distribution of  $X_{32}$  under Data Traffic and Low Mobility Speed

$X_{32} \backslash \text{DEV}$	$\text{Dev}_{X_{31} X_{32}}[H(X_{41} X_{31}=?, X_{32})]$
$X_{32}=L$	0.071
$X_{32}=M$	0.347

Table 5.21 Table of Deviation of Entropy of  $X_{41}$  given  $X_{32}$  under Data Traffic and Low Mobility Speed

Then, according to algorithm defined in 5.11, deviation for  $H(X_{41}|X_{31}=L, X_{32}=L)$  and  $H(X_{41}|X_{31}=M, X_{32}=L)$  is calculated as above, and also the deviation for  $H(X_{41}|X_{31}=L, X_{32}=M)$  and  $H(X_{41}|X_{31}=M, X_{32}=M)$  is listed in Table 5.21.

Similar to the above calculation, according to algorithm defined in 5.11, the deviation for  $H(X_{41}|X_{31}=L, X_{32}=L)$  and  $H(X_{41}|X_{31}=M, X_{32}=L)$  is calculated as below. And, the deviation for  $H(X_{41}|X_{31}=L, X_{32}=M)$  and  $H(X_{41}|X_{31}=M, X_{32}=M)$

DEV $X_{31}$	$Dev_{X_{32} X_{31}}[H(X_{41} X_{31}, X_{32}=?)]$
$X_{31}=L$	0.028
$X_{31}=M$	0.425

Table 5.22 Table of Deviation of Entropy of  $X_{41}$  given  $X_{31}$  under Data Traffic and Low Mobility Speed

Then, according to 5.12, the significance indicator of  $X_{41}$  given  $X_{31}$  can be calculated as below:

$$\begin{aligned} \text{Sig}(X_{41} | X_{31}) &= P(X_{32} = L) \times Dev_{X_{31}|X_{32}=L}[H(X_{41} | X_{31} = ?, X_{32} = L)] \\ &\quad + P(X_{32} = M) \times Dev_{X_{31}|X_{32}=M}[H(X_{41} | X_{31} = ?, X_{32} = M)] \\ &= 0.214 \end{aligned}$$

In the above formula to calculate the significance indicator of  $X_{41}$  given  $X_{31}$ , the probability distribution of  $X_{31}$  can be read from Table 5.19.

The same, the significance indicator of  $X_{41}$  given  $X_{32}$  can be calculated as below:

$$\begin{aligned} \text{Sig}(X_{41} | X_{32}) &= P(X_{31} = L) \times \text{Dev}_{X_{32}|X_{31}=L}[H(X_{41} | X_{31} = L, X_{32} = ?)] \\ &\quad + P(X_{31} = M) \times \text{Dev}_{X_{32}|X_{31}=M}[H(X_{41} | X_{31} = M, X_{32} = ?)] \\ &= 0.149 \end{aligned}$$

In the above formula to calculate the significance indicator of  $X_{41}$  given  $X_{32}$ , the probability distribution of  $X_{32}$  can be read from Table5.20.

Since  $\text{Sig}(X_{41}|X_{31})$  is larger than  $\text{Sig}(X_{41}|X_{32})$ , it's shown that the  $X_{31}$  has more impact on  $X_{41}$ 's performance. Then, in the next step, we need to focus on  $X_{31}$ , and find out which element on the second layer will affect the  $X_{31}$ 's performance most.

$P(X_{31} X_{21}, X_{22}, X_{23})$ $X_{21}, X_{22}, X_{23}$	$P(X_{31}=L X_{21}, X_{22}, X_{23})$	$P(X_{31}=M X_{21}, X_{22}, X_{23})$	$P(X_{31}=H X_{21}, X_{22}, X_{23})$
$X_{21}=H, X_{22}=L, X_{23}=L$	0.745	0.255	0
$X_{21}=H, X_{22}=L, X_{23}=M$	0.893	0.107	0
$X_{21}=H, X_{22}=M, X_{23}=L$	0.46	0.54	0
$X_{21}=H, X_{22}=M, X_{23}=M$	0.412	0.588	0
$X_{21}=M, X_{22}=L, X_{23}=L$	0.727	0.273	0
$X_{21}=M, X_{22}=L, X_{23}=M$	0.681	0.319	0
$X_{21}=M, X_{22}=M, X_{23}=L$	0.448	0.552	0
$X_{21}=M, X_{22}=M, X_{23}=M$	0.409	0.591	0
$X_{21}=L, X_{22}=L, X_{23}=L$	0.745	0.255	0
$X_{21}=L, X_{22}=L, X_{23}=M$	0.716	0.284	0
$X_{21}=L, X_{22}=M, X_{23}=L$	0.388	0.612	0
$X_{21}=L, X_{22}=M, X_{23}=M$	0.333	0.667	0

Table 5.23 Table of Conditional Probability of  $X_{31}$  given  $X_{21}$ ,  $X_{22}$  and  $X_{23}$  under Data Traffic and Low Mobility Speed

#### Step4: Significant Component Learning for $X_{31}$

The conditional probability table is listed in Table5.23 for  $X_{31}$ . For Data Traffic, and low mobility speed, the Table5.23 is a table of the conditional probability of  $X_{31}$  given  $X_{21}$ ,  $X_{22}$  and  $X_{23}$ . Then, by calculating the entropy of  $X_{31}$  given  $X_{21}$ ,  $X_{22}$  and  $X_{23}$ , Table5.24 is generated below.

$H(X_{31}   X_{21}, X_{22}, X_{23})$	$H(X_{31}   X_{21}=?, X_{22}=?, X_{23}=?)$
$X_{21}=H, X_{22}=H, X_{23}=L$	0.992
$X_{21}=H, X_{22}=H, X_{23}=M$	0.976
$X_{21}=H, X_{22}=H, X_{23}=M$	0.846
$X_{21}=M, X_{22}=H, X_{23}=L$	0.903
$X_{21}=M, X_{22}=H, X_{23}=M$	0.819
$X_{21}=M, X_{22}=H, X_{23}=L$	0.491
$X_{21}=H, X_{22}=M, X_{23}=L$	0.995
$X_{21}=H, X_{22}=M, X_{23}=M$	0.978
$X_{21}=H, X_{22}=M, X_{23}=H$	0.963
$X_{21}=M, X_{22}=M, X_{23}=L$	0.918
$X_{21}=M, X_{22}=M, X_{23}=M$	0.819
$X_{21}=M, X_{22}=M, X_{23}=H$	0.861

Table 5.24 Table of Entropy of  $X_{31}$  given  $X_{21}$ ,  $X_{22}$  and  $X_{23}$  under Data Traffic and Low Mobility Speed

Since here are three controlling variables for  $X_{31}$ , in order to calculate individually how much each parent element will impact performance of  $X_{31}$ , the joint probability distribution are given below for various combination of  $(X_{22}, X_{23})$ ,  $(X_{21}, X_{23})$  and  $(X_{21}, X_{22})$ .

$P(X_{22}, X_{23})$ $X_{22}, X_{23}$	$P(X_{22}=?, X_{23}=?)$
$X_{22}=L, X_{23}=L$	0.8224
$X_{22}=L, X_{23}=M$	0.0506
$X_{22}=M, X_{23}=L$	0.1193
$X_{22}=M, X_{23}=M$	0.0077

Table 5.25 Table of Joint Probability Distribution of  $X_{22}$  and  $X_{24}$  under Data Traffic and Low Mobility Speed

$P(X_{21}, X_{23})$ $X_{21}, X_{23}$	$P(X_{21}=?, X_{23}=?)$
$X_{21}=L, X_{23}=L$	0.5410935
$X_{21}=L, X_{23}=M$	0.0421565
$X_{21}=M, X_{23}=L$	0.35880425
$X_{21}=M, X_{23}=M$	0.01519575
$X_{21}=H, X_{23}=L$	0.04185225
$X_{21}=H, X_{23}=M$	0.00089775

Table 5.26 Table of Joint Probability Distribution of  $X_{21}$  and  $X_{24}$  under Data Traffic and Low Mobility Speed

$P(X_{21}, X_{22})$ $X_{21}, X_{22}$	$P(X_{21}=?, X_{22}=?)$
$X_{21}=L, X_{22}=L$	0.50441225
$X_{21}=L, X_{22}=M$	0.07883775
$X_{21}=M, X_{22}=L$	0.32848825
$X_{21}=M, X_{22}=M$	0.04551175
$X_{21}=H, X_{22}=L$	0.0400995
$X_{21}=H, X_{22}=M$	0.0026505

Table 5.27 Table of Joint Probability Distribution of  $X_{21}$  and  $X_{22}$  under Data Traffic and Low Mobility Speed



Then, according to algorithm defined in 5.11, the deviation for the entropy of  $X_{31}$  under fixed values of  $X_{22}$  and  $X_{23}$  is calculated as above, and also the deviation for different values of  $X_{21}$ , based on the different combination of values of  $X_{22}$  and  $X_{23}$  are listed below in Table 5.28

Dev $X_{22}, X_{23}$	$\text{Dev}_{X_{21} X_{22}, X_{23}}[H(X_{31} X_{21}=?, X_{22}, X_{23})]$
$X_{22}=L, X_{23}=L$	0.012947
$X_{22}=L, X_{23}=M$	0.052229
$X_{22}=M, X_{23}=L$	0.014067
$X_{22}=M, X_{23}=M$	0.025402

Table 5.28 Table of Deviation of Entropy of  $X_{31}$  given  $X_{22}$  and  $X_{23}$  under Data Traffic and Low Mobility Speed

Similar to Table 5.28, the deviation of Entropy of  $X_{31}$  given  $X_{21}$  and  $X_{23}$  is calculated below in Table 5.29.

Dev $X_{21}, X_{23}$	$\text{Dev}_{X_{22} X_{21}, X_{23}}[H(X_{31} X_{21}, X_{22}=?, X_{23})]$
$X_{21}=L, X_{23}=L$	0.049369
$X_{21}=L, X_{23}=M$	0.019521
$X_{21}=M, X_{23}=L$	0.047842
$X_{21}=M, X_{23}=M$	0.02417
$X_{21}=H, X_{23}=L$	0.042509
$X_{21}=H, X_{23}=M$	0.11738

Table 5.29 Table of Deviation of Entropy of  $X_{31}$  given  $X_{21}$  and  $X_{24}$  under Data Traffic and Low Mobility Speed

Table 5.28 and Table 5.29 are calculated to determine how much  $X_{21}$  and  $X_{22}$  will affect the changing of uncertainty of  $X_{31}$ . In order to determine the impact of  $X_{23}$ , the deviation of Entropy of  $X_{31}$  given  $X_{21}$  and  $X_{22}$  is calculated below in Table 5.30.

Dev $X_{21}, X_{23}$	$Dev_{X_{22} X_{21}, X_{23}}[H(X_{31} X_{21}, X_{22}, X_{23}=?)]$
$X_{21}=L, X_{22}=L$	0.010808
$X_{21}=L, X_{22}=M$	0.017859
$X_{21}=M, X_{22}=L$	0.011332
$X_{21}=M, X_{22}=M$	0.003262
$X_{21}=H, X_{22}=L$	0.047074
$X_{21}=H, X_{22}=M$	0.002558

Table 5.30 Table of Deviation of Entropy of  $X_{31}$  given  $X_{21}$  and  $X_{22}$  under Data Traffic and Low Mobility Speed

Then, according to algorithm 5.12, the significance indicator of  $X_{31}$  given  $X_{21}$  can be calculated as below:

$$\begin{aligned}
 \text{Sig}(X_{31} | X_{21}) &= P(X_{22} = L, X_{23} = L) \times Dev_{X_{21}|X_{22}=L, X_{23}=L}[H(X_{31} | X_{21} = ?, X_{22} = L, X_{23} = L)] \\
 &+ P(X_{22} = L, X_{23} = M) \times Dev_{X_{21}|X_{22}=L, X_{23}=M}[H(X_{31} | X_{21} = ?, X_{22} = L, X_{23} = M)] \\
 &+ P(X_{22} = M, X_{23} = L) \times Dev_{X_{21}|X_{22}=M, X_{23}=L}[H(X_{31} | X_{21} = ?, X_{22} = M, X_{23} = L)] \\
 &+ P(X_{22} = M, X_{23} = M) \times Dev_{X_{21}|X_{22}=M, X_{23}=M}[H(X_{31} | X_{21} = ?, X_{22} = M, X_{23} = M)] \\
 &= 0.0151629
 \end{aligned}$$

Similar to the above method, then the significance indicator of  $X_{31}$  given  $X_{22}$  can be calculated in below algorithm:

$$\begin{aligned}
\text{Sig}(X_{31} | X_{22}) &= P(X_{21} = L, X_{23} = L) \times \text{Dev}_{X_{22}|X_{21}=H, X_{23}=L} [H(X_{31} | X_{22} = ?, X_{21} = L, X_{23} = L)] \\
&+ P(X_{21} = L, X_{23} = M) \times \text{Dev}_{X_{22}|X_{21}=H, X_{23}=M} [H(X_{31} | X_{22} = ?, X_{21} = L, X_{23} = M)] \\
&+ P(X_{21} = M, X_{23} = L) \times \text{Dev}_{X_{22}|X_{21}=H, X_{23}=H} [H(X_{31} | X_{22} = ?, X_{21} = M, X_{23} = L)] \\
&+ P(X_{21} = M, X_{23} = M) \times \text{Dev}_{X_{22}|X_{21}=M, X_{23}=L} [H(X_{31} | X_{22} = ?, X_{21} = M, X_{23} = M)] \\
&+ P(X_{21} = H, X_{23} = L) \times \text{Dev}_{X_{22}|X_{21}=M, X_{23}=M} [H(X_{31} | X_{22} = ?, X_{21} = H, X_{23} = L)] \\
&+ P(X_{21} = H, X_{23} = M) \times \text{Dev}_{X_{22}|X_{21}=M, X_{23}=H} [H(X_{31} | X_{22} = ?, X_{21} = H, X_{23} = M)] \\
&= 0.0469539
\end{aligned}$$

$$\begin{aligned}
\text{Sig}(X_{31} | X_{23}) &= P(X_{21} = L, X_{22} = L) \times \text{Dev}_{X_{22}|X_{21}=H, X_{22}=L} [H(X_{31} | X_{23} = ?, X_{21} = L, X_{22} = L)] \\
&+ P(X_{21} = L, X_{22} = M) \times \text{Dev}_{X_{22}|X_{21}=H, X_{22}=M} [H(X_{31} | X_{23} = ?, X_{21} = L, X_{22} = M)] \\
&+ P(X_{21} = M, X_{22} = L) \times \text{Dev}_{X_{22}|X_{21}=H, X_{22}=H} [H(X_{31} | X_{23} = ?, X_{21} = M, X_{22} = L)] \\
&+ P(X_{21} = M, X_{22} = M) \times \text{Dev}_{X_{22}|X_{21}=M, X_{22}=L} [H(X_{31} | X_{23} = ?, X_{21} = M, X_{22} = M)] \\
&+ P(X_{21} = H, X_{22} = L) \times \text{Dev}_{X_{22}|X_{21}=M, X_{22}=M} [H(X_{31} | X_{23} = ?, X_{21} = H, X_{22} = L)] \\
&+ P(X_{21} = H, X_{22} = M) \times \text{Dev}_{X_{22}|X_{21}=M, X_{22}=H} [H(X_{31} | X_{23} = ?, X_{21} = H, X_{22} = M)] \\
&= 0.0121465
\end{aligned}$$

Comparing significance indicator  $\text{Sig}(X_{31}|X_{22})$  is larger than  $\text{Sig}(X_{31}|X_{21})$  and  $\text{Sig}(X_{31}|X_{23})$ , the  $X_{22}$  has the largest impact on  $X_{31}$ 's performance. In the next step,  $X_{22}$ 's performance will be studied to determine which parent element has the most significant impact on its uncertainty.

#### **Step5: Significant Component Learning for $X_{22}$**

The conditional probability of  $X_{22}$  given its parent set is listed in below table.

$P(X_{22} X_{11}, X_{12})$ $X_{11}, X_{12}$	$P(X_{22}=L  X_{11},X_{12})$	$P(X_{22}=M  X_{11},X_{12})$	$P(X_{22}=H  X_{11},X_{12})$
$X_{11}=F, X_{12}=H$	0.953	0.047	0
$X_{11}=E, X_{12}=H$	0.79	0.21	0
$X_{11}=F, X_{12}=P$	0.938	0.062	0
$X_{11}=E, X_{12}=P$	0.811	0.189	0

Table 5.31 Table of Conditional Probability of  $X_{22}$  given  $X_{11}, X_{12}$  under Data Traffic and Low Mobility Speed

Then, by calculating the entropy of  $X_{24}$  given  $X_{12}$  and  $X_{14}$ , Table5.32 is generated below.

$H(X_{22}  X_{11},X_{12})$ $X_{11}, X_{12}$	$H(X_{22}  X_{11}=?,X_{12}=?)$
$X_{11}=F,X_{12}=H$	0.2735
$X_{11}=E,X_{12}=H$	0.7415
$X_{11}=F,X_{12}=P$	0.3353
$X_{11}=E,X_{12}=P$	0.6994

Table 5.32 Table of Entropy of  $X_{24}$  given  $X_{11}, X_{12}$  under Data Traffic and Low Mobility Speed

As controlled by simulation results, the  $P(X_{11}=F)=0.5$  and  $P(X_{11}=E)=0.5$ . For  $X_{12}$ ,  $P(X_{12}=H)=0.5$  and  $P(X_{12}=P)=0.5$ .

Then, according to algorithm 5.12, the significance indicator of  $X_{22}$  given  $X_{11}$  can be calculated as below:

$$\begin{aligned} \text{Sig}(X_{22} | X_{11}) &= P(X_{12} = H) \times \text{Dev}_{X_{11}|X_{12}=H, X_{13}=H} [H(X_{22} | X_{11} = ?, X_{12} = H)] \\ &\quad + P(X_{12} = P) \times \text{Dev}_{X_{11}|X_{12}=H, X_{13}=H} [H(X_{22} | X_{11} = ?, X_{12} = P)] \\ &= 0.2942 \end{aligned}$$

Similarly, the significance indicator of  $X_{22}$  given  $X_{12}$  is calculated below:

$$\begin{aligned} \text{Sig}(X_{22} | X_{12}) &= P(X_{11} = F) \times \text{Dev}_{X_{11}|X_{12}=H, X_{13}=H} [H(X_{22} | X_{12} = ?, X_{11} = F)] \\ &\quad + P(X_{11} = E) \times \text{Dev}_{X_{11}|X_{12}=H, X_{13}=H} [H(X_{22} | X_{12} = ?, X_{11} = E)] \\ &= 0.0367 \end{aligned}$$

By comparing the values of significance indicator for giving  $X_{11}$ ,  $X_{12}$ , the impact of  $X_{11}$  is higher than  $X_{12}$ .  $X_{11}$  represents the routing component of “RREQ’s Distribution Method”. This conclusion learned from quantitative analysis agrees with the analysis based on routing mechanism.

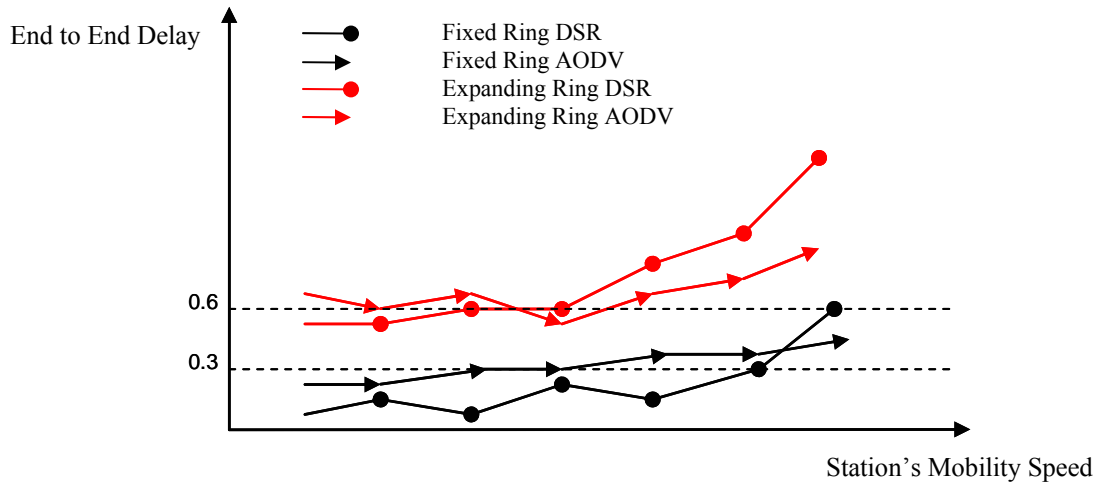


Figure 5.14 Chart of End to End Delay against MANET Station Mobility Speed

As demonstrated in the above Fig.5.14, by using the same RREQ’s dissemination method (fixed ring or expanding ring), DSR has a less End to End delay than AODV.

For low mobility speed stations, DSR allows multiple routing paths cached. So, chance to find a path at intermediate node will be better than AODV. Therefore, DSR's total End to End delay will be shorter than AODV.

For both AODV and DSR, Expanding Ring will take longer to return a routing path. Since Data Packet will wait in queue for RREP's arrival, when Expanding Ring is implemented, more time will be spent in waiting for RREP. Hence, on average, fixed ring will have shorter End to End Delay than expanding ring. The trade off is, routing overhead is more in fixed ring than expanding ring.

Therefore, this is the other good example to show the power of the component dependence network in detecting the significant component for a routing performance metric. And, illustrated by above example, once the dependence network structure is determined, the significance indicator can be calculated in low computation complexity.

#### **Step6: Parameter Tuning for Significant Component**

The advantage of the component dependence network is that by the quantitative analysis, the significant component can be detected. This self learning method can free the people from analyzing simulation results and making decision based on statistics. And the other advantage of component dependence network is that the significant component can be located, and then we can concentrate the study on this significant component, and make significant parameter tuned. Otherwise, it's very difficult to determine which parameter should be tuned in order to have a better performance.

After locating the significant component, we need to tune the parameter to find a best value to keep the balance between two routing performance metrics. For example, this is shown in simulation results that a good TTL value will affect end to end delay and routing overhead. Usually, higher is the TTL, lower is the data packet's end to end delay, and higher is the routing overhead. Then, the question is how to find a right value of TTL value set in RREQ to meet the requirement of both end to end delay and routing overhead.

For example, for a data traffic with high mobility speed, after the significant component determined as RREQ's dissemination method (Expanding ring method will be used to reduce routing overhead), the routing overhead and data packet's end to end delay will be studied along with different values of TTL set in RREQ.

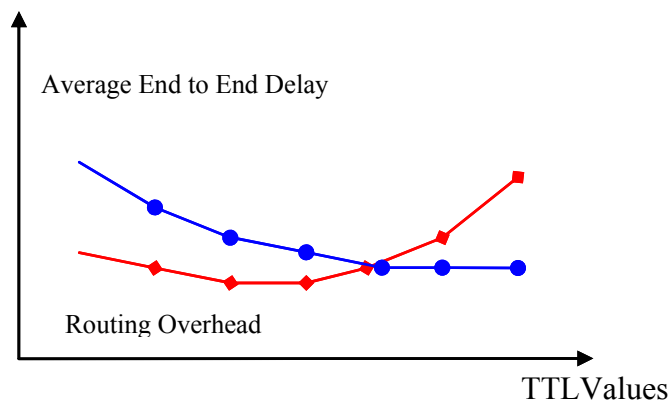


Figure 5.15 Chart of Average Data Packet's End to End Delay and Routing Overhead against TTL Values

As shown in above figure, When TTL increase, average End to End delay will be reduced, since waiting time for RREP will be decreased by average. However,

routing overhead will first decrease then increase. This can be interpreted as when TTL value is small, a lot of RREQ will be terminated before reaching destination, then new RREQ with higher TTL value will be rebroadcasted, hence overall routing overhead is high due to those wasted RREQ dissemination. When TTL increase to a appropriate value, RREQ can reach target destination without too many re-broadcast, then overall routing overhead will be reduced compared to lower TTL. However, when TTL continues to increase, routing overhead will increase significantly, since RREQ will be disseminated too many times (may be in loop).

For End to End delay, after an optimal TTL value is located, the End to End delay will not change too much. Since during End to End delay, path discovery takes the most of the time. After TTL reached optimal values, path discovery time won't be affected too much. Hence, average End to End delay won't decrease after TTL value reaching optimal values.

By conclusion, TTL is a significant component that will affect End to End delay. Hence, TTL's value can be tuned according to simulation results. If routing overhead has more weight than End to End delay, TTL's value stays low. If End to End delay is the major concern, then TTL's value could be higher.



## BIBLIOGRAPHY

- [1] J. Baras, H. Huang, "Component Based Routing: A new Methodology for Designing Routing Protocols for MANET," *Proceedings of the 25th Army Science Conference*, Orlando 2006.
- [2] J. Baras, "Component-Based Routing Protocols for MANETs: Foundations, Methodology, Implementation, Review for a Collaborative Technology," *Alliance in Communications and Networking, Project 1.2*, 2005.
- [3] Das, Perkins, Royer, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks," *Proceedings of INFOCOM 2000 Conference*, 2000.
- [4] Hsu, Bhatia, Takai, Bagrodia, "Performance of Mobile Ad Hoc Networking Routing Protocols in Realistic Scenarios," *Proceedings of IEEE Milcom*, 2003.
- [5] Geiger, Verma, Pearl, "Identifying Independence in Bayesian Networks," *Networks*, Vol.20, 507-534 1990.
- [6] Margaritis, Thrun, "Bayesian Network Induction via Local Neighborhoods," *Technical Report, CMU-CS-99-134*, 1999.
- [7] Heckerman, S. "A Tutorial on Learning with Bayesian Networks," *Technical Report, MSR-TR-95-06*, 1995.
- [8] Choi, Ko, "A Performance Evaluation for Ad Hoc Routing Protocols in Realistic Military Scenarios," *Proceedings of the 9th International Conference on Cellular and Intelligent Communications*, 2004.
- [9] Kahn, R. E. "The Organization of Computer Resources into a Packet Radio Network. *IEEE Transactions on Communications COM-25* (1): 169–178, 1975.

- [10] T.Clausen and P. Jacquet, "Optimized Link State Routing Protocol," <http://www.ietf.org/rfc/rfc3626.txt>.
- [11] Perkins, E. B. Royer, S.Das, "Ad Hoc On Demand Distance Vector Routing," <http://www.ietf.org/rfc/rfc3561.txt>.
- [12] Johnson, D. Malta and Y. Hu, "The Dynamic Source Routing Protocol for MAENT," <http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt>.
- [13] S. R. Das, C. E. Perkins and E. M. Royer, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks," *Proceedings of INFOCOM 2000 Conference*, Tel-Aviv, Israel, March 2000.
- [14] J. Choi and Y. Ko, "A Performance Evaluation for Ad Hoc Routing Protocols in Realistic Military Scenarios," *Proceedings of the 9th International Conference on Cellular and Intelligent Communications (CIC 2004)*, October 2004.
- [15] J. Hsu, S. Bhatia, M. Takai and R. Bagrodia, "Performance of Mobile Ad Hoc Networking Routing Protocols in Realistic Scenarios," *Proc. of the IEEE Milcom*, 2003.
- [16] J. S. Baras and H. Mehta, "A Probabilistic Emergent Routing Algorithm for Mobile Ad hoc Networks," *Modeling and Optimization in Mobile Ad Hoc and Wireless Networks*, March 3-5, 2003.
- [17] M. Dorigo and L.M. Gambardella. "Ant Colony System: A Cooperative learning Approach to the Travelling Salesman Problem", *Biosystems* 43 (1997):73-81
- [18] J. S. Baras, "Component-Based Routing Protocols for MANETs: Foundations, Methodology, implementation", *Review for a Collaborative Technology Alliance in Communications and Networking*, Project 1.2, April 2005.

- [19] Samar, P., Pearlman, M.R. and Haas, Z.J., “Hybrid Routing: The Pursuit of an Adaptable and Scalable Routing Framework for Ad Hoc Networks,” in *Handbook of Ad Hoc Wireless Networks*, M. Ilyas, ed., CRC Press 2002.
- [20] L.E. Miller, “Distribution of Link Distance in a Wireless Network”, *Journal of Research of the National Institute of Standards and Technology*, 106, (2001):401-402.
- [21] G.F. Cooper and E.Herskovits, “A Bayesian Method for the Induction of Probabilistic Networks from Data”, *Machine Learning*, 9, 309-347, 1992.
- [22] Kevin Murphy, “A Brief Introduction to Graphical Models and Bayesian Networks”, <http://www.cs.ubc.ca/~murphyk/Bayes/bnintro.html>, 1998
- [23] Samuel S. Wilks, “Mathematical statistics”, *John Wiley & Sons*, 1963
- [24] T. Roos, P.Grunwald, P. Myllymaki and H. Tirri, “On Discriminative Bayesian Network Classifiers and Logistic Regression”, *Machine Learning*, 59, 267-296, 2005.
- [25] W. Lam and F. Bacchus, “Learning Bayesian belief networks. An approach based on MDL principle”, *Computational Intelligence*, 10, 269-293, 1994.
- [26] Richard Lowry, “<http://faculty.vassar.edu/lowry/ch3a.html>,”
- [27] Thomas M. Cover and Joy A. Thomas, “Elements of Information Theory”, *Wiley Series In Telecommunications*, 1991.
- [28] A. Dembo. “Information inequalities and uncertainty principles,” Technical Report, *Department of Statistics, Stanford University*, 1990.

- [29] R. A. Fisher. "Theory of Statistical Estimation," *Proc. Cambridge Phil. Society*, 22: 700-725, 1925.
- [30] A. Trivino-Cabrere, J Garcia-de-la-Nava, E. Casilari and F. J. Gonzalez-Canete, "An Analytical Model to Estimation Path Duration in MANETs," *The 9th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2006.
- [31] S. Acid and L. M. De Campos, "A Hybrid Methodology for Learning Belief Networks: BENEDICT," *International Journal of Approximate Reasoning*, 27, 235-262, 2001.
- [32] L. Miller, "Distribution of Link Distances in a Wireless Networks," *Journal of Research of the National Institute of Standards and Technology*, 106, 401-412, 2001.
- [33] A.P.Dawid, "Conditional Independence in Statistical Theory," *Journal of the Royal Statistical Society. Series B*, 1-31, 1979.
- [34] J. Liu, "Learning Bayesian Networks with a Hybrid Convergent Method," *IEEE Transactions on Systems, MAN, and Cybernetics-Part A: Systems and Humans*, 436-449, 1999.
- [35] R. Badonnel, R. State, and O. Festor, "Monitoring End-to-End Connectivity in Mobile Ad-Hoc Networks," *International Conference on Network*, 83-90, 2005.
- [36] D. Heckerman, "A Tutorial on Learning With Bayesian Networks," *Technical Report MSR-TR-95-06, Microsoft Research*, 1996.
- [37] G. Allenby, P. Rossi and R. McCulloch, "Hierarchical Bayes Models: A Practitioners Guide," 2005.

- [38] N. Friedman and M. Goldszmidt, "Learning Bayesian Networks with Local Structures," *Learning and Inference in Graphical Models*, 1998.
- [39] W. Lam, "A Distributed Learning Algorithm for Bayesian Inference Networks," *IEEE Transactions On Knowledge and Data Engineering*, 93-105, 2002.
- [40] D. Margaritis and S. Thrun, "Bayesian Network Induction via Local Neighborhoods," *In Advances in Neural Information Processing Systems*, 12, 505-511, 2000.
- [41] L. Campos, "A Scoring Function for Learning Bayesian Networks based on Mutual Information and Conditional Independence Tests," *Journal of Machine Learning Research*, 2149-2187, 2006.
- [42] K. Laskey, "Sensitivity Analysis for Probability Assessments in Bayesian Networks," *Systems, Man and Cybernetics, IEEE Transactions on*, 901-909, 1995.
- [43] E. Castillo and J. Gutierrez, "Sensitivity Analysis in Discrete Bayesian Networks," *IEEE Transactions On Systems, Man, and Cybernetics-Part A: Systems and Humans*, Vol.27, No.4, July 1997
- [44] P. Spirtes and C. Meek, "Learning Bayesian Networks with Discrete Variables from Data," *International Conference on Knowledge Discovery and Data Mining*, 294-299, 1995.
- [45] A.P. Dawid, "Conditional Independence," *Wiley Encyclopedia of Statistical Sciences*, 1997.
- [46] W. Bergsma, "Testing Conditional Independence for Continuous Random Variables," *Journal of the American Statistical Association*, 2004.

- [47] A. McDonald and T. Znati, "Link Availability Models for Mobile Ad-Hov Networks," *Department of Computer Science, University of Pittsburgh*, 1999.
- [48] E. Charniak, "Bayesian Networks without Tears," *Artificial Intelligence Magazine*, 50-63, 1991.