Connectionless Approach: A Localized Scheme To Mobile Ad Hoc Networks

Yao Ho
University of Central Florida

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation
CONNECTIONLESS APPROACH – A LOCALIZED SCHEME TO MOBILE AD HOC NETWORKS

by

YAO HUA HO
B.S. University of Central Florida, 2001
M.S. University of Central Florida, 2002

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the School of Electrical Engineering and Computer Science in the College of Engineering and Computer Science at the University of Central Florida Orlando, Florida

Summer Term
2009

Major Professor: Kien A. Hua
© 2009 Yao Hua Ho
ABSTRACT

According to a Gartner Group (www.gartner.com) report in September 2008, the worldwide telecommunications market is on pace to reach $2 trillion in 2008. Gartner predicts that by 2012, the ratio of mobile to fixed connections will exceed 4-to-1. The North American mobile data market grew to 141.1 million connections in 2007, with a compound annual growth rate of 41.7 percent. It is believed that a large portion will be ad hoc and multi-hop connections, which will open many opportunities for Mobile Ad hoc NETwork (MANET) applications and Wireless Mesh Network (WMN) applications. A MANET is a self-organizing multi-hop wireless network where all nodes participate in the routing and data forwarding process. Such a network can be easily deployed in situations where no base station is available, and a network must be build spontaneously. In applications such as battlefield communications, national crises, disaster recovery, and sensor deployment, a wired network is not available and ad hoc networks provide the only feasible means of communications and information access. Ad hoc networks have also become commonplace for gaming, conferencing, electronic classrooms, and particularly vehicle-to-vehicle communications. A Wireless mash network (WMN) is collection of mesh clients and mesh nodes (routers), with mesh nodes forming the backbone of the network and providing connection to the Internet and other network. Their rapid deployment and ease of maintenance are suitable for on-demand network such as disaster recovery, homeland security, convention centers, hard-to-wire buildings and unfriendly terrains.

One important problem with MANET is the routing protocol that needs to work well not just with a small network, but also sustain efficiency and scalability as the network gets expanded and the application transmits data in greater volume. In such an environment, mobility, channel
error, and congestion are the main causes for packet loss. Due to mobility of mobile hosts, addressing frequent and unpredictable topology changes is fundamental to MANET research. Two general approaches have been considered: connection-oriented approach and connectionless-oriented approach. In the former, the emphasis is on how to reconnect quickly with low overhead when a broken link occurs. Examples of this approach includes [5], [9], [10], [16], [26], [28], [29], [34], [44], and [45]. In contrast, connectionless-oriented approach focuses on minimizing the occurrence of broken links. We proposed one such scheme called Connectionless Approach (CLA) [21] and [22]. In CLA, the network area is divided into non-overlapping grid cells, each serving as a virtual router. Any physical router (i.e., mobile host), currently inside a virtual router, can help forward the data packet to the next virtual router along the virtual link. This process is repeated until the packet reaches its final destination. Since a virtual link is based on virtual routers which do not move, it is much more robust than physical links used in the connection-oriented techniques. Simulation results in our previous works [21] and [22], based on GloMoSim [60], indicate that CLA performs significantly better than connection-oriented techniques (i.e., AODV, DSR, LAR, GRID, TMNR, and GPSR).

The contribution of this work consists of investigating and developing new Connectionless-Oriented Approach for Mobile Ad Hoc Network. Two of the greatest impacts of this research are as follows. First, the new approach is targeted towards robustly support high mobility and large scale environment which has been adapted for vehicle-to-vehicle environment in [20]. Second, the detailed simulations which compare eight representative routing protocols, namely AODV, DSR, LAR, GRID, TMNR, GPSR, CBF, and CLA, under high-mobility environments. As many important emergent applications of the technology involved high-
mobility nodes, very little is known about the existing routing methods perform relative to each other in high-mobility environments. The simulation results provide insight into ad hoc routing protocols and offer guidelines for mobile ad hoc network applications.

Next, we enhanced and extend the connectionless-oriented approach. The current connectionless-oriented approach, however, may suffer from packet drops since traffic congestion is not considered in the packet forwarding policy. We address this weakness by considering the connectionless-oriented approach with a collision avoidance routing technique. After that, we investigate techniques to enforce collaboration among mobile devices in supporting the virtual router functionality. Many works have been published to combat such problem - misbehaving nodes are detected and a routing algorithm is employed to avoid and penalize misbehaving nodes. These techniques, however, cannot be applied to the connectionless-oriented approach since any node in the general direction towards the destination node can potentially help forward the data packets. To address the security and cooperation issues for connectionless-oriented approach, we introduce a cooperation enforcement technique called 3CE (3-Counter Enforcement). In addition, wireless mesh networks have become increasingly popular in recent years. Wireless mesh network (WMNs) are collection of mesh clients and mesh nodes (routers), with mesh nodes forming the backbone of the network and providing connection to the Internet and other network. We propose a paradigm that combines virtual routers and mesh nodes to create a hybrid network call VR-Mesh Network. This hybrid network can reduce number of mesh node needed without decrease the performance of the network.
ACKNOWLEDGMENTS

Achievement of an important goal in life is often done not in isolation, but rather with the support of caring people. I take this opportunity to thank who have supported me in this endeavor.

As my advisor, Professor Kien A. Hua receives my utmost gratitude. Dr. Hua has been my advisor and my friend since I studied for my bachelor degree here at U.C.F. He opened my eyes to the world of research in computer science. He has helped me to discover my talent which I can only find by going through the Ph.D. training. Dr. Hua has lead me to achieve a Ph.D., a degree, which I once thought it was unattainable. The concern and care for his students is exemplary, both in academic and other matters. My achievements that I have accomplished so far and will attain in the future are the result of his hard work. I am grateful to have him as my advisor. Thanks also to all the members of my committee for their helpful comments on my research work and for being flexible with dates and times of all the qualifiers. I am indebted also to all the teachers at the universities and all the colleagues of our Data System Group.

Many thanks go to my wife Chi-Hsiao, who has been patient and understanding through this process. She has devoted and sacrificed herself to support me over the years. I thank my son and daughter, William and Charlotte who are always cheering me up with their smile. I thank my sister, Ai, who has been a source of motivation and support. Also, I thank my mother and father, Mei-Hui Lin and Kenneth who has supported me financially as well as spiritually with their endless love and patience.
TABLE OF CONTENTS

LIST OF FIGURES .................................................................................................................. x
LIST OF TABLES ................................................................................................................... xii

1. INTRODUCTION ..................................................................................................................1

2. CONNECTIONLESS APPROACH (CLA) .............................................................................3
   2.2 Proposed Solution: Connectionless Approach (CLA) ..................................................8
       2.2.1 Virtual Grid Cell ......................................................................................................9
       2.2.2 Initialization Phase .............................................................................................10
       2.2.3 Location Discovery .............................................................................................10
       2.2.4 Path Computation .............................................................................................12
           2.2.4.1 Establish the Reference Line .................................................................13
           2.2.4.2 Determine the Reference Points ..........................................................14
           2.2.4.3 Select Grid Cells for the Path .................................................................16
       2.2.5 Data Forwarding .................................................................................................17
       2.2.6 Grid Path Maintenance .....................................................................................20
       2.2.7 Low Node Density Environment and Obstacle Environment .........................22
       2.2.8 Optimization – Caching Overheard Location Information ...............................25

2.3 Simulation Study .............................................................................................................25
   2.3.1 Simulation Parameters .........................................................................................26
   2.3.2 Performance Metrics ..........................................................................................26
   2.3.3 Simulation Results ..............................................................................................27
       2.3.3.1 Effect of Mobile Speed ..............................................................................27
       2.3.3.2 Effect of Pause Time ..................................................................................31
       2.3.3.3 Effect on Number of Communication Sessions .........................................35
       2.3.3.4 Effect of Network Density ..........................................................................38
       2.3.3.5 Effect of Terrain Area (Scalability) .............................................................42

3. DYNAMIC ROUTE DIVERSION .....................................................................................49
   3.2.1 Connectionless Approach with Dynamic Route Diversion (CLA-DRD) .............53
       3.2.1.1 Connectionless Approach (CLA) ...............................................................53
       3.2.1.2 Dynamic Route Diversion for CLA (CLA-DRD) ........................................55
   3.2.2 Contention Based Forwarding with Dynamic Route Diversion .........................63
       3.2.2.1 Contention-Based Forwarding (CBF) ..........................................................63
3.2.2.2 Dynamic Route Diversion for CBF (CBF-DRD)..............................64

3.3.1 Performance metrics ........................................................................68

3.3.2 Simulation Parameters .......................................................................69

3.3.3 Simulation Results .............................................................................69
  3.3.3.1 Effect on Number of Communication Session to Fraction of Packet Delivered........................................................................70
  3.3.3.2 Effect on Number of Communication Session to End-to-End Delay .................................................................................70
  3.3.3.3 Effect on Number of Communication Session to Normalized Routing Load.................................................................71

3.4 Discussion ............................................................................................72

4. COOPERATION ENFORCEMENT..............................................................74

4.2 Node Configuration and Tamper Proof Module .......................................78

4.3 3C Module ...........................................................................................80
  4.3.1 Counters Update during the Location Discovery Phase......................82
    4.3.1.1 Initiate Location Discovery .........................................................83
    4.3.1.2 Receive Location Discovery Packet .........................................84
    4.3.1.3 Forward or Reply Location Discovery Packet .............................85
  4.3.2 Counters Update during the Data Forwarding Phase .........................86
    4.3.2.1 Receive Data Packet .................................................................86
    4.3.2.2 Forward Data Packet .................................................................87
  4.3.3 Initiate Misbehavior Detection ............................................................89
  4.3.4 Detection Mode ................................................................................91
  4.3.5 Misbehavior Detection Procedure .....................................................91

4.4 Simulation Study ....................................................................................93
  4.4.1 Schemes Implemented ......................................................................94
  4.4.2 Simulation Setup ...............................................................................94
  4.4.3 Metric .............................................................................................95
  4.4.4 Experimental Results .......................................................................96
    4.4.4.1 Packet Delivered Ratio ...............................................................96
    4.4.4.2 Misbehaving Node Detection Ratio ..........................................100
    4.4.4.3 False Accusation Ratio ...............................................................100
    4.4.4.4 Control Overhead Ratio .............................................................102
    4.4.4.5 End-to-End Delay ....................................................................103
    4.4.4.6 Active Detection Ratio ...............................................................104
5. HYBRID MOBILE AD HOC AND MESH NETWORKS ........................................108
  5.1 Related Work..........................................................................................109
  5.2 Hybrid of Virtual Router and Mesh Network...........................................112
    5.2.1 Route Request..................................................................................113
    5.2.2 Route Reply....................................................................................114
    5.2.3 Data Forwarding...............................................................................116
    5.2.4 Route Update...................................................................................117
  5.3 Simulation Study.......................................................................................118
    5.3.1 Schemes Implemented.......................................................................118
    5.3.2 Simulation Setup...............................................................................118
    5.3.3 Performance Metrics.........................................................................121
    5.3.4 Experimental Results.........................................................................122
6. CONCLUDING REMARKS AND FUTURE WORKS......................................126
  6.2.1 Security and Privacy in Wireless Mesh Network.................................129
  6.2.2 Wireless Vehicular Communication....................................................130
7. REFERENCES...............................................................................................131
LIST OF FIGURES

Figure 1. Connectionless Approach. ................................................................. 4
Figure 2. Virtual Cell....................................................................................... 9
Figure 3. Grid path. ..................................................................................... 13
Figure 4. Reference line. ............................................................................. 14
Figure 5. Reference points. ....................................................................... 15
Figure 6. Determine the reference points and grid path. ......................... 17
Figure 7. Forwarding Delay.......................................................................... 20
Figure 8. Recovery Cells = Neighbors(A) \ Neighbors(C). ....................... 22
Figure 9. Recover Cell = (Neighbors (A) \ Neighbors (B)) \cup (Neighbors(B) \ Neighbors(C)). ................................................................. 22
Figure 10. Low Density or Obstacle Environment....................................... 24
Figure 11. Effect of speed on fraction of packets delivered. .................... 29
Figure 12. Effect of speed on end-to-end delay........................................ 29
Figure 13. Effect of speed on normalized routing load. ......................... 30
Figure 14. Effect of speed on packet duplication........................................ 30
Figure 15. Effect of pause time on fraction of packets delivered............. 33
Figure 16. Effect of pause time on fraction of end-to-end delay.............. 33
Figure 17. Effect of pause time on normalized routing load.................. 34
Figure 18. Effect of pause time on packet duplication............................. 34
Figure 19. Effect of number of communication sessions on fraction of packets delivered..... 36
Figure 20. Effect of number of communication sessions on end-to-end delay. .................... 37
Figure 21. Effect of number of communication sessions on normalized routing load............. 37
Figure 22. Effect of number of communication sessions on packet duplication................................................................. 38
Figure 23. Effect of node density on fraction of packets delivered........... 40
Figure 24. Effect of node density on fraction of end-to-end delay.......... 41
Figure 25. Effect of node density on normalized routing load................ 41
Figure 26. Effect of node density on normalized packet duplication........ 42
Figure 27. Effect of terrain area on fraction of packets delivered........... 44
Figure 28. Effect of terrain area on fraction of end-to-end delay............. 44
Figure 29. Effect of terrain area on normalized routing load................... 45
Figure 30. Effect of terrain area on packet duplication............................ 45
Figure 31. Dynamic packets rerouting as a collision avoidance mechanism. 50
Figure 32. Connectionless Approach. ......................................................... 54
Figure 33. Data header ................................................................. 56
Figure 34. Packet Queue in MAC layer ........................................... 57
Figure 35. Congestion Notification Packet ................................... 58
Figure 36. Data Forwarding Procedure ........................................... 59
Figure 37. DRD Request Packet Format ........................................ 60
Figure 38. DRD Request Procedure .............................................. 61
Figure 39. DRD Reply Packet Format ........................................... 62
Figure 40. Example of Dynamic Route Diversion .............................. 63
Figure 41. Contention-Based Forwarding ....................................... 64
Figure 42. DRD Request Packet Format for CBF-DRD ....................... 65
Figure 43. DRD Request Procedure for CBF-DRD ............................. 66
Figure 44. Example of CBF-DRD .................................................. 67
Figure 45. Effect of number of communication sessions on fraction of packets delivered. .... 70
Figure 46. Effect of number of communication sessions on end-to-end delay ..................... 71
Figure 47. Effect of number of communication sessions on normalized routing load .......... 72
Figure 48. Layer Structure ........................................................... 79
Figure 49. Update the counters during the Location Discovery phase ............... 88
Figure 50. Virtual link for a Detection packet .................................... 93
Figure 51. Packet Deliver Ratio (P) with 10% Malicious Nodes ............ 97
Figure 52. Packet Deliver Ratio (P) with 20% Malicious Nodes ............ 97
Figure 53. Packet Deliver Ratio (P) with 10% Malicious Nodes ............ 99
Figure 54. Packet Deliver Ratio (P) with 20% Malicious Nodes ............ 99
Figure 55. Control Overhead Ratio (C) with 20% Malicious Nodes ......... 103
Figure 56. End to End Delay (E) with 20% Malicious Nodes ............ 104
Figure 57. Active Detection Ratio (A) with 20 m/s and 20% malicious nodes ......... 105
Figure 58. Number of detecting nodes needed per malicious node at different speed .......... 106
Figure 59. Virtual Router Approach ............................................... 111
Figure 60. Hybrid Network .......................................................... 112
Figure 61. Virtual Link and Physical Link ........................................ 113
Figure 62. Virtual routers and the route from source to VR-Destination ............ 115
Figure 63. Network area with 25 mesh nodes .................................. 119
Figure 64. Square grid topology with different number of mesh nodes ............ 121
Figure 65. Effect on number of mesh node to fraction of packet delivered .......... 122
Figure 66. Effect on number of mesh node to end-to-end delay .................. 123
Figure 67. Effect on number of mesh node to normalized routing load .......... 124
LIST OF TABLES

Table 1. Performance Trade-Off between Different Approaches ........................................ 47
Table 2. Applicable Environments of Different Approaches .............................................. 48
Table 3. Detection Ratio (D) of CLA and CBF with 3CE ..................................................... 100
Table 4. False Accusation Ratio (F) of CLA and CBF with 3CE ........................................ 101
Table 5. Parameters of mesh and mobile nodes ................................................................. 120
Wireless networks have become increasingly popular in recent years. There are two variations of mobile wireless networks: *infrastructure mobile networks* and *infrastructureless mobile networks*. The latter are also known as ad-hoc networks. This type of network has no fixed routers. Instead, mobile nodes also function as routers which discover and maintain communication connections. Thus, a *mobile ad hoc network* (MANET) is a self-organizing multi-hop wireless networks where all nodes participate in the routing and data forwarding process.

In a MANET, communication connections need to adapt to frequent unpredictable topology changes due to the mobility, energy constraints, and limited computing power of mobile devices. When a disconnection occurs, reconnection must be established quickly with little overhead. Many routing protocols have been proposed for MANETs such as *Ad hoc On Demand Distance Vector* (AODV) [45], *Cluster Gateway Switch Routing* (CGSR) [10], *Contention-Based Forwarding* (CBF) [16], *Distance Routing Effect Algorithm for Mobility* (DREAM) [1], *Dynamic Destination Sequenced Distance-Vector Routing* (DSDV) [44], *Dynamic Source Routing* (DSR) [26], *Global State Routing* (GSR) [9], *Location-Aided Routing* (LAR) [29], *Location-Based Routing* (TMNR) [5], *Trajectory Based Forwarding* [42], *Wireless Routing Protocol* (WRP) [35], and *Zone Routing Protocol* (ZRP) [18]. These types of algorithms follow a connection-oriented approach. By connection-oriented, we mean that mobile nodes need to establish a connection using either route discovery or a routing table before two mobile units can communicate. In practice, the mobility of some of the nodes can be high, causing frequent
reconnections. Such overhead wastes energy and causes discontinuity in the communications. The jitter effect is particularly undesirable for streaming applications such as voice and video.

Reducing the frequency of reconnections is a hard problem as it is an innate property of mobility beyond the control of any routing algorithm. This fact has motivated us to avoid using any “fixed” connections at all. The contribution of this proposal is the introduction of a connectionless approach to MANET communications. This new paradigm, called *Connectionless Approach for MANET’s* (CLA) in this paper, is unique in that its performance is essentially unaffected by node mobility, and therefore more suitable for mobile applications.
2. CONNECTIONLESS APPROACH (CLA)

In the connectionless approach, we divide the network area into small “virtual cells.” Each node uses the location information obtained by using technology such as the Global Positioning System (GPS) [14] to determine which virtual cell the node is currently on. Unlike the conventional connection-oriented approaches, our technique does not associate a communication session with a specific route in terms of a hop-by-hop connection. Instead, communication between any two nodes is conducted over a path of virtual cells which connects the nodes (see Figure 1). In other words, a communication session is defined by a grid path. Only nodes on a cell along this path are responsible for forwarding the data passing through. At any time, nodes can move in and out of a cell. When a node leaves the transmission path, it is no longer responsible for forwarding of the data. Similarly, when a node moves into an active communication path, this node is required to participate in the communication session by helping to forward the data. We note that our strategy is effective even when the precision of the location information system is not “high” (as is the case with GPS). Although discrepancies in the location information might cause a node to fail to forward data for a given grid cell, there is an equal probability that a similar position deviation will cause some other node near this cell to forward the data. Besides, nodes on a cell can also cover for each other should this condition arise.
The idea of dividing a network area into smaller “virtual grid areas” is not new. In [33], a grid is used in a distributed location service to track mobile node locations. In another paper [58], nodes in the same grid cell coordinate with each other to determine which will sleep and for how long in order to conserve energy. In [34], grids are used as cluster units. Similar to the cluster based approaches, GRID reduces the cost of route maintenance by dividing the network area into fixed-size grid cells with nodes within each cell forming a cluster. Within each cell, a gateway is selected to maintain route, forward data, and maintain cluster members. This strategy has the disadvantages of the cluster-based approaches. First, the network throughput can be severely limited due to the fact that only a few gateway nodes can forward data. Second, in the GRID approach, the number of hops in a route is fixed by the size of the grid. The average number of hops per route depends on the grid size. A smaller grid size will have a higher average number of hops per route. Although increasing the grid size reduces the average number of hops, it will also result in weaker connectivity of communication that, in turn, will result in frequent route breaks among gateways. Third, this strategy requires each cell leader to
periodically notify its existence to other nodes within the cell. Fourth, GRID does not address the issue of route break due to a failure at a gateway node.

Recently, a hop-by-hop connection-oriented approach, called *Trajectory Based Forwarding* (TBF) [42], has been proposed. Similar to DREAM and Cartesian Routing [15] each node in TBF needs to establish the connection to the next hop before forwarding the data, and therefore a connection-oriented approach. To determine the next hop, a node compares the distances of its neighbor nodes to the trajectory of the route stored in the packet, and then selects the next hop that lies more or less on the trajectory. This scheme incurs the following overhead: (1) each node must maintain and update the location information of its every neighbor; (2) a node needs to compare the locations of all its neighbors with the trajectory in order to select the next hop; and (3) maintaining the trajectory information is very expensive. Consequently, this scheme is limited to applications in which the communicating nodes move very slowly. The study in [42] assumed that the destination nodes do not move.

From the above discussion, we observe that existing communication techniques for MANET’s take the connection-oriented approach. The various drawbacks limit their applications to smaller networks with lower mobility. The long delay associated with the repair of a route break also precludes these techniques from media streaming applications. The Connectionless approach, introduced in the following Section 2.2, has none of the above problems.

### 2.1 Related Work

A routing protocol for MANETs needs to be robust in adapting to frequent and unpredictable topology changes due to inherent mobility, bandwidth and energy constraints, and limited
computing power of mobile devices. When a disconnection occurs, reconnection must be established quickly with little overhead. Many routing protocols have been proposed for MANETs such as Ad hoc On Demand Distance Vector (AODV) [45], Cluster Gateway Switch Routing (CGSR) [10], Connectionless Approach (CLA) [21], Contention-Based Forwarding (CBF) [16], Distance Routing Effect Algorithm for Mobility (DREAM) [3], Dynamic Destination Sequenced Distance-Vector Routing (DSDV) [44], Dynamic Source Routing (DSR) [26], Greedy Perimeter Stateless Routing (GPSR) [27], Location-aware Routing Protocol (GRID) [34], Global State Routing (GSR) [9], Location-Aided Routing (LAR) [29], Trajectory Based Forwarding [42], Location-Based Routing (TMNR) [5], Wireless Routing Protocol (WRP) [35], and Zone Routing Protocol (ZRP) [18].

Early-generation routing protocols, such as DSDV, WEP, and GSR, establish communication link by maintaining routing information in a routing table at each node. A drawback to this solution is that every node needs to update its routing table and propagate the update, as the network topology changes, in order to maintain a consistent view of the network. This operation incurs excessive network traffic and computation overhead. Later techniques, such as DSR and AODV, attempt to reduce unnecessary network traffic by initiating route request on-demand. This type of routing protocols establishes communication links by flooding the network to find a route to the destination node. This strategy is simple and robust; however, it is not energy efficient and can cause severe media congestion.

Another approach to reduce network flooding is to leverage location information obtained from GPS (Global Positioning System) [14] or other location services [32] and [49]. For instance, LAR uses location information to limit the area of flooding, thereby reducing the
number of route request messages. These schemes result in better power conservation and improve network scalability.

Some other techniques reduce not only number of route requests but also route maintenance costs. This type of approach, such as GRID and CGSR, organizes mobile nodes into clusters. Each cluster has a cluster head and a number of gateways. Two clusters communicate via a gateway node within their communication range. An obvious advantage of this environment is that only cluster-heads and gateway nodes need to rebroadcast messages. However, the network throughput can be limited by the number of gateway nodes. Furthermore, cluster management incurs overhead.

To address mobility issues, one-hop approaches, such as TBF, TMNR, and GPSR, have been proposed. In these schemes, instead of the need to establish a complete connection from the source to the destination, the node only needs to establish the connection to the next hop (i.e., one hop) and forward the data. To determine the next hop, a node compares the distances of its neighbor nodes to the destination node (i.e., GPSR), the next waypoint (i.e., TMNR), or a trajectory (i.e., TBF).

CBF is more recent techniques developed for routing in MANETs. While GPRS, TMNR, and TBF need to maintain (proactively or reactively) neighbor nodes location information and establish a connection to the next hop before forwarding a data packet, CBF simply forward data packets without first establishing the link to the next node. In CBF, a forwarding node transmits a data packet as a single-hop broadcast to all its neighbors. These neighbors compete with each other for the “right” to forward the packet. During this contention period, a node determines how well it is suited to be the next hop for the packet. The node that
wins the contention *suppresses* the other nodes, thus establishing itself as the next forwarding node. This contention is based on the distances of the nodes to the destination. A drawback of this strategy in a high-density environment is that several neighboring nodes might have similar distances to the destination. Consequently, they will all establish themselves as the next hop and forward the data packet. This incurs unnecessary network traffic and wastes power of the mobile nodes. A solution, suggested in [16], is for each contestant node to report its qualification for forwarding the data packet and wait for the current forwarding node to select the winner for the next hop. We did not consider this strategy in our study since it is similar to GPSR. Since both CBF and CLA can robustly support high mobility, these two schemes have been adapted for vehicle-to-vehicle environment in [17] and [20].

2.2 Proposed Solution: Connectionless Approach (CLA)

In mobile ad hoc networks, the network topology changes rapidly as a result of host. To minimizing the occurrence of broken links, we proposed a connectionless-oriented approach called Connectionless Approach (CLA).

As in many routing protocols [1], [5], [13], [27], [29], [30], [31], [42], [53], [54], [57], and [58], we also assume that all the nodes can obtain location information provided by technologies such as the *Global Positioning System* (GPS) [14]. This is a reasonable assumption because of the increasing availability of these devices and because the GPS service is provided without charge. If GPS is not available, one can conceive that nodes calculate their position with a local scheme - a research area that has recently been well studied in [32] and [49]. In our presentation, we use *xy*-coordination. In fact, devices such as GPS can provide 3-D location in longitude, latitude, and altitude.
2.2.1 Virtual Grid Cell

The network area is divided into small “virtual cells.” These cells, defined by the coordinates of their upper right and lower left corners, are each assigned a unique cell ID. Each virtual cell has eight neighboring cells (see Figure 2(A)). Note that the network area can also be divided into “cellular-like” cells. For simplicity, we use square cells in the discussion of this paper and our simulation study.

(A) Cell E has eight neighboring cells A, B, C, D, F, G, H, and I.
(B) R is farthest distance between any two nodes in two neighboring cells.

Figure 2. Virtual Cell.

We size our virtual cells based on the nominal radio range \( R \) as follows. Assume each virtual grid cell is a square with \( x \) units on each side. The distance between the two possible farthest nodes in any two neighboring cells must not be larger than \( R \) (see Figure 2 (B)). Therefore, we have:

\[
\sqrt{2}(2x) \leq R \quad \text{or} \quad x \leq \frac{R}{2\sqrt{2}}
\]

In other words, the virtual grid cell is designed such that, for two neighboring cells, all nodes in one cell can communicate with all nodes in the neighboring cells.
2.2.2 Initialization Phase
When a new node enters the network area, this node first contacts any nearby node to get the partition information. The partition information given out by a node \( n \) includes the size of the terrain area, the size of the virtual cell, and the cell ID and the coordinates of the two corners of the virtual cell \( n \) is currently on. With this partition information, the new node can easily compute the location and ID information of other cells in the entire terrain area, since every virtual grid cell is relative to its neighboring cells in terms of the coordinates and cell ID. The new node can also determine which virtual cell it is on by comparing its own location with the coordinates of the virtual cells.

2.2.3 Location Discovery
When a source node \( S \) wants to send data to a destination node \( D \), \( S \) needs to know the destination cell, i.e., the cell containing \( D \). First, node \( S \) searches its own Location Cache to find the location information of \( D \). If the location information is available and fresh enough, the Path Computation routine is called to determine the grid path to the destination cell. We will describe this path computation process in the next section. If no location information is found in the cache, \( S \) initiates the Location Discovery process to find the location of the destination node \( D \).

To initiate location discovery, node \( S \) broadcasts a LOCATION DISCOVERY message to all neighboring nodes within the transmission range. A LOCATION DISCOVERY packet contains the source node ID, destination node ID, location information of source node, and a unique request ID.

When a node receives a LOCATION DISCOVERY packet, it checks if it is the destination node. If so, it returns a LOCATION REPLY message to the source of the LOCATION DISCOVERY.
The **LOCATION REPLY** packet contains the location information of the destination node. Since the source node’s location information is included in the **LOCATION DISCOVERY** packet, the destination node can call the *Path Computation* routine to determine the grid path used to reply.

When a node receives a **LOCATION DISCOVERY** packet and it is not the destination node, it first stores this request ID in its *Request ID Cache* if this node never saw this **LOCATION DISCOVERY** packet before. Next, this node checks its *Location Cache* to see if the location information of the destination node is available. We note that the *Location Cache* implements an LRU (Least Recently Used) replacement policy and keeps only the location information that is within the freshness threshold. If the information is available in the cache, the node will reply with the destination location information in the **LOCATION REPLY** packet; otherwise, the node propagates the **LOCATION DISCOVERY** packet as a local broadcast. When the source node receives the **LOCATION REPLY** packet, the source node will call the *Path Computation* routine, discuss in the next section, to determine the grid path to the destination node.

To avoid flooding the network with **LOCATION DISCOVERY** messages, we can apply a probabilistic delay technique (i.e., [19], [41], [51], and [56]). In this scheme, when a node \( n \) receives a **LOCATION DISCOVERY** packet, \( n \) delays the forwarding for a random time interval. During this period, if \( n \) hears any neighboring node forwarding the **LOCATION DISCOVERY** packet, then \( n \) will not need to forward the same message. We can even farther reduce flooding by considering location information in the delay computation. The idea is to use a shorter delay for a node that is farther away from the sender. This allows the **LOCATION DISCOVERY** messages to travel faster compared to random delay. We use this technique in our forwarding procedure.
We will discuss the forwarding procedure and the delay computation in more detail in Section 3.5.

When replying the Location Discovery, our approach uses path computation routine to compute a path back to source node. In the LOCATION REPLY packet, we only included the location information of the destination node. For other connection-oriented approaches such as DSR, the route reply uses a hop-by-hop route to be relayed back to source. This approach has higher probability of fail reply due to route failure, especially in a high mobility environment.

Contrary to Route Discovery of connection-oriented approaches, LOCATION DISCOVERY carries only location of the source node that allows the destination to compute a reference line. Any nodes along this reference line could help relaying the LOCATION REPLY back to the source. We will describe the path computation and data forwarding in Section 3.4 and Section 3.5.

2.2.4 Path Computation

In our connectionless approach, we do not need to maintain a hop-by-hop route between the source and destination nodes. Our technique selects a list of grid cells that form a “connecting” path between the source and destination. An example is illustrated in Figure 3. It shows that the lightly shaded cells are part of the grid path selected. Nodes within each of these cells alternate in forwarding data toward the destination node. Two distinct routes are shown in Figure 3 to illustrate the forwarding of two different data packets at different times. Notice that we do not need to use every cell to forward data. In Figure 3, the data packets from m or n can be relayed by a node that is closest to the destination among all the nodes that resides along the path and can overhear the packets. This allows packets to skip grids in a dense network. We shall describe the data forwarding strategy in more detail shortly.
When a node leaves the selected cells, it is no longer obliged to forward the data. Similarly, if a node enters the grid path, this node must participate in the data forwarding. A delay forwarding scheme, discussed in Section 3.5, is used to coordinate the nodes within a grid path to take turns forwarding the data. We observe that a grid path is much more robust than a traditional hop-by-hop connection. The latter would fail if any one node along the route “fails.” In contrast, a grid path is much more tolerant of “node failures” since a neighboring node can dynamically substitute for the failed node with no overhead. This characteristic makes the connectionless approach more suitable for ad-hoc networks. We describe the three steps in path computation in the following subsections.

2.2.4.1 Establish the Reference Line

We define the destination cell as the cell containing the destination node. Similarly, the source cell is the cell containing the source node. To determine a virtual grid path between a source

Figure 3. Grid path.
node and a destination node, we first establish the “reference line” between the source cell and the destination cell. The reference line (RL) is the straight line that connects the center of the source cell \((X_S, Y_S)\) and the center of the destination cell \((X_D, Y_D)\) as illustrated in Figure 4. It shows the reference line between the two cells, one at the upper right and the other at lower left corners. Obviously, the coordinates of the two end points define the reference line.

![Figure 4. Reference line.](image)

2.2.4.2 Determine the Reference Points

Once the reference line has been established, we need to determine the reference points. The reference points (RP’s) on a reference line are the interceptions of the reference line and either the vertical or horizontal centerlines of the considered grid cells (See Figure 5).

To compute the coordinates of the reference points, we first need to determine if the reference line is either in a vertical or horizontal orientation as follows. Let the coordinates of the source node be \((X_S, Y_S)\), and that of the destination node be \((X_D, Y_D)\). If \(\left| \frac{X_D - X_S}{Y_D - Y_S} \right| > 1\), the reference line is horizontal; otherwise, it is vertical. If this ratio is equal to 1, we consider the reference line as in vertical orientation.
The reference points divide the reference line into equal-length segments. As seen in Figure 5, the three reference points divide the reference line into four RL-segments. The number of such segments can be computed as follows. If the RL is horizontally oriented, then

$$RL\_Segment\_Count = \left\lceil \frac{|X_D - X_S|}{cell\_size} \right\rceil; \quad (1)$$

otherwise,

$$RL\_Segment\_Count = \left\lceil \frac{|Y_D - Y_S|}{cell\_size} \right\rceil. \quad (2)$$

We first consider the case when the reference line is in a horizontal orientation. The distance between any two adjacent reference points along the vertical axis, called the $y$-Increment (see Figure 6), can be computed as follows:

$$y\text{-Increment} = \frac{Y_D - Y_S}{RL\_Segment\_Count} \quad (3)$$

Using Equation (3), we calculate the coordinates $RP_n (X_n, Y_n)$ of the $n^{th}$ reference point, from the center of the source cell as follows:
\[ X_n = \begin{cases} 
X_S + \frac{X_D - X_S}{|X_D - X_S|} (n \times \text{cell\_size}) & X_S \neq X_D \\
X_S & X_S = X_D 
\end{cases} \quad (4) \]

\[ Y_n = Y_S + (n \times y\text{-Increment}) \quad (5) \]

Similarly, for a vertically oriented reference line, the distance between any two adjacent reference points along the horizontal axis, called the \textit{x-Increment}, can be calculated as follows:

\[ x\text{-Increment} = \frac{X_D - X_S}{RL\_Segment\_Count} \quad (6) \]

Using Equation (6), we determine the coordinates \( R_{Pn}(X_n, Y_n) \) of the \( n^{th} \) reference point from the center of the source cell as follows:

\[ X_n = X_S + (n \times x\text{-Increment}) \quad (7) \]

\[ Y_n = \begin{cases} 
Y_S + \frac{Y_D - Y_S}{|Y_D - Y_S|} (n \times \text{cell\_size}) & Y_S \neq Y_D \\
Y_S & Y_S = Y_D 
\end{cases} \quad (8) \]

2.2.4.3 Select Grid Cells for the Path

Once the reference points have been determined, the grid path consists of those grid cells overlapping with at least one reference point. If a reference point is on the border of two adjacent cells, we arbitrarily choose one of the two according to some convention (e.g., the one on the “left” or the “bottom”). A grid path is illustrated in Figure 6 consisting of the shaded cells, each overlaps with one reference point. In this example, one of the reference points \( R_{P2} \) falls on a cell border; and we pick the cell at the bottom to serve in the path. Once the grid path is determined, the source node can start to transmit data.
2.2.5 **Data Forwarding**

To transmit a data packet, the source node includes the following information in the data header: *Source Node ID, Source Cell ID, Destination Node ID, Destination Cell ID, x-Increment, y-Increment, RL_Segment_Count*, and *Current Cell ID*. The *Current Cell ID* is the ID of the cell containing the node that is about to forward the data packet. Thus, each intermediate node updates this header field before relaying the data packet.

When a node $n$ receives a data packet from $m$, the data forwarding procedure is as follows:

1. If $n$ is the destination node, $n$ does not forward the data.

2. If $n$ is not on the grid path, $n$ does not forward the data.

3. If Steps 1 and 2 fail (i.e., $n$ might need to forward the data), $n$ delays the forwarding for a certain time interval. During this delay period, $n$ discards the same packets arriving from the upstream. Furthermore, it will cancel the forwarding if either of the following two conditions becomes true:
- \( n \) hears the same downstream packet.

- \( n \) hears the same packet from a neighboring node on the same grid cell.

At the end of the delay period, if the forwarding decision has not been cancelled, \( n \) forwards the data.

We explain the above three steps in more detail as follows:

- **Step 1:** The node \( n \) checks the Destination Node ID in the data header to verify if \( n \) is the destination node. If the packet is for \( n \), it does not forward the data.

- **Step 2:** Only nodes on the grid path need to forward data. The node \( n \), upon receiving a packet, checks if it is on the grid path as follows. Suppose \( n \) is currently on the cell \( N \) with center at \((X_N, Y_N)\); \((X_S, Y_S)\) is the coordinate of the center of the source cell \( S \); and the reference line is horizontally oriented. \( n \) first determines its nearest reference point \((RP_k)\) as follows:

\[
k = \min\left(\frac{|X_N - X_S|}{grid\_size}, RL\_Segment\_Count\right)
\]

That is, the \( k \)th reference point is the reference point nearest to node \( n \). The position of \( RP_k \) can be computed using Equations (4) and (5). If the reference point falls within the region of the cell \( N \), then node \( n \) is on the grid path. In this case, \( n \) saves the location information of the source and destination nodes to its location cache for future use.

- **Step 3:** If \( n \) is on the grid path, and is not the destination, then \( n \) delays the forwarding for a time interval computed as follows:
\[ \text{DELAY} = \frac{\alpha}{\text{Dist}_n}, \quad (9) \]

where \( \alpha \) is a constant of maximum delay in \( \mu \text{sec} \); and \( \text{Dist}_n \) is the distance between node \( n \) and the center of node \( m \)'s cell denoted by the \textit{Current Cell ID} in the packet header. In other words, \( \text{Dist}_n \) approximates the distance between \( n \) and the sender \( m \) of the data packet. Equation 9 computes a shorter delay for a node farther from the sender. Notice we can always factor other parameters into the delay computation such as workload or battery power level. That is, we use a larger delay for a node with low battery power to conserve its energy.

Let \( A \) and \( B \) be two cells on the grid path. If cell \( A \) is closer to the source cell than cell \( B \), then cell \( A \) is in the upstream of cell \( B \). We also say cell \( B \) is in the downstream of cell \( A \).

With the information in the packet header, node \( n \) can compute the grid path, as discussed in Section 3.4 at the beginning of the communication session and use this path information to categorize arriving packets into \textit{upstream packets} and \textit{downstream packets}. During the delay period, \( n \) discards the same packets arriving from the upstream. This is illustrated in 0. It shows that both nodes \( n \) and \( p \) are within the broadcast range of the sender \( m \). However, since node \( n \) is further away from the sender, \( n \) has a shorter delay. Consequently, the packet forwarded from node \( m \) to node \( p \) will be discarded, and the longer route (more hops) through node \( p \) will not be used. In general, this scheme results in fewer hops and helps the data packets to travel faster.

During the delay, node \( n \) cancels the forwarding decision if it detects the same packet arriving from downstream or from a neighboring node on the same grid cell. \( n \) can verify if a packet is from such a neighboring node by checking the \textit{Current Cell ID} in the packet header. If the
Current Cell ID is the same as the ID of the cell currently containing \( n \), then the packet is coming from a neighboring node.

By the end of the delay period, if \( n \) has not cancelled the forwarding decision, \( n \) should relay the data packet.

*Both node \( n \) and \( p \) are within the radio range of the sender, node \( m \). However, node \( n \) is farther away, and therefore has a shorter forwarding delay. As a result, the longer route through node \( p \) will not be used.*

**Figure 7.** Forwarding Delay.

### 2.2.6 Grid Path Maintenance

In path maintenance, we need to consider two scenarios: (1) the source node or the destination node moves off the source cell or the destination cell, respectively. (2) Some intermediate cell in the grid path becomes empty, i.e., no mobile nodes are found in the cell.

To handle Case 1, the source node attaches its current location information to the data packet. For destination node, it periodically checks its own location to see if it is out of the
destination cell. If it is, it sends a LOCATION UPDATE packet to update its location with the source node. With the updated location information, the source node can call the Path Computation routine to select a new grid path. Similarly, the destination node can also activate the Path Computation routine to find a new grid path should the source node moves outside the original source cell.

To handle Case 2, i.e., a cell in a grid path may momentarily become empty, we allow the nodes on some of the neighboring cells to help forward the data. We refer to such a neighboring cell as a Recovery Cell in this paper. We illustrated the neighboring cells of a given cell in Figure 2(A). To facilitate our discussion, we use “Neighbors(A)” to denote the set of cells neighboring to a cell A. Let us consider three consecutive cells A, B, and C in a grid path. If cell B becomes empty, we determine the recovery cells as follows.

\[ \text{If } \text{Neighbors}(A) \cap \text{Neighbors}(C) \neq B \text{ then} \]

\[ \text{Recovery Cells} = \text{Neighbors}(A) \cap \text{Neighbors}(C) \quad \text{/*See Figure 8*/} \]

\[ \text{else} \]

\[ \text{Recovery Cell} = (\text{Neighbors (A) } \cap \text{Neighbors (B)}) \cup (\text{Neighbors(B) } \cap \text{Neighbors(C)}) \]

\[ \text{/*See Figure 9*/} \]

When a node \( r \) not on the grid path overhears a data packet, \( r \) checks if it is on the recovery cell with respect to this data packet. This can be easily determined since \( r \) can compute the grid path using the information in the packet header. If the node \( r \) is on a recovery cell, it will listen to see if any node forwards the data packet within the maximum \( \text{DELAY} \). When this delay expires, \( r \) will help to forward the data packet if no other node has forwarded it.
In general, a grid path is much less likely to fail than a traditional hop-by-hop route. Therefore, the overhead associated with grid path maintenance is significantly lower than the overhead incurred by route maintenance in traditional connection-oriented techniques.

![Figure 8](image.png)

**Figure 8.** Recovery Cells = Neighbors(A) \(\cap\) Neighbors(C).

![Figure 9](image.png)

**Figure 9.** Recover Cell = (Neighbors (A) \(\cap\) Neighbors (B)) \(\cup\) (Neighbors(B) \(\cap\) Neighbors(C)).

### 2.2.7 Low Node Density Environment and Obstacle Environment

We note that there may be situations in which we cannot find a non-empty recovery cell to help forward the data. This can happen in a low density, a heterogeneous network, or an obstacles environment in which the number of mobile nodes is low, unevenly distributed, or blocked by obstacles. In this situation, a node, that had forwarded a packet but did not hear any node forwards the packet again, will need to ask for help from its neighboring cells (See 0). This node will send exactly the same data packet again but with a special one-bit field in the packet header.
to solicit help from any node to forward the packet. We call this type of data packet HELP FORWARD data packet. When a node $n$ receives such a data packet from a node $m$, $n$ tries to help as follows:

1. If $n$ is not on the grid path or recovery cell, $n$ delays the forwarding for a certain time interval $DELAY\_Help$; otherwise, $n$ delays the forwarding for half of the $DELAY\_Help$ time interval.

2. During this delay period, $n$ will cancel the forwarding if any one of the following three conditions becomes true:
   - $n$ hears the same HELP FORWARD data packet forwarded by some other node.
   - $n$ hears the normal data packet with the same packet ID as the HELP FORWARD data packet. When a node is on a grid path or in a recovery cell, it modifies the HELP FORWARD data packet back into a normal data packet before forwarding it.
   - $n$ hears a STOP HELP control packet. The STOP HELP control packet is sent by the node $m$ since it heard some node forwarding the HELP FORWARD data packet.

3. At the end of the delay period, if the forwarding decision has not been cancelled, $n$ forwards the data as follows:
   - If $n$ is not on the grid path, $n$ simply forwards the HELP FORWARD data packet.
   - If $n$ is on the grid path, $n$ first changes the HELP FORWARD data packet into a normal data packet, and then forwards it.

In the above procedure, the delay is computed as follows.
\[
\text{DELAY-\_Help} = \left| \frac{\alpha}{2 \cdot \text{Dist}_D} - \frac{\alpha}{2 \cdot \text{Dist}_n} \right|, \quad (10)
\]
where \( \alpha \) is a maximum delay constant in \( \mu \text{sec} \), \( \text{Dist}_D \) the distance between node \( n \) and the center of the cell denoted by the Destination Cell ID in the packet header, and \( \text{Dist}_n \) the distance between node \( n \) and the center of the cell denoted by the Current Cell ID (cell of sender node \( m \)) in the packet header. The significance of this equation is to select a node farther away from \( m \) and closer to the destination node to forward the data packet. We also note that the node selection strategy in Step 1 uses half of the \( \text{DELAY\_Help} \) to favor a node on the original grid path. This node, in Step 2, will modify the HELP FORWARD data packet back into a normal data packet before forwarding it. This effectively saves the forwarding nodes in the subsequent hops the cost of sending the STOP HELP control packets.

* In a low density environment or an obstacle environment, nodes can ask help from their neighbors to forward data packets.

**Figure 10.** Low Density or Obstacle Environment.
2.2.8 Optimization – Caching Overheard Location Information

When a node receives a LOCATION DISCOVERY message, a LOCATION REPLY message, a LOCATION UPDATE message, or a regular data packet; it can learn about the locations of the sender and/or the receiver. Such location information can be cached locally to save location discovery process in the near future. The benefit of this approach is limited by the freshness of the information. We can only use the information if it has not expired.

2.3 Simulation Study

To evaluate our connectionless approach, we performed simulations using a network simulator called GloMoSim [60]. This simulator, developed at UCLA, is a packet-level simulator specifically designed for ad-hoc networks. It follows the OSI 7-layer network communication model. Although, popular simulators such as NS-2, OPNET Modeler, and GloMoSim provide advanced simulation environments to test and debug network protocols, we prefer GloMoSim due to its ability to handle high mobility of nodes and its scalability of handle large number of nodes and size of network area. Unlike other simulators, GloMoSim uses the parallel discrete-event simulation capability provide by Parsec [1].

Eight routing protocols were simulated and compared – AODV, DSR, LAR, GRID, TMNR, GPSR, CBF, and our ConnectionLess Approach (CLA) for MANET’s. While DSR and LAR were proposed in 1998, other five protocols were proposed within the last five years to address advancements in mobile applications. In particular our focus is on high-mobility environments, as many important emergent applications of this technology involve high-mobility nodes. As an example, cars in a vehicle-to-vehicle network are typically moving at speeds exceeding 30 mph. Very little is known about how existing routing methods perform relative to
each other in such high-mobility environments. Our purpose is to investigate the impact of high mobility on different routing protocols under various scenarios. The simulation results provide insight into ad hoc routing protocols and offer guidelines for mobile ad hoc network applications. We perform sensitivity analysis with respect to mobile speed, pause time, number of communication sessions, node density, and terrain area to investigate their effect on performance.

2.3.1 Simulation Parameters

The field configuration is a $1000m \times 1000m$ field, unless it is specified otherwise by the network scenarios. The radio propagation range for each node is 250 meters and channel capacity is 2 Mbits/sec. We used the random waypoint mobility model [60]; that is, each node randomly selects a destination point. When the node reaches this destination point, it pauses for a period of time, and then selects another destination point. Traffic applications are constant bit rate sessions. Each data packet is 512 bytes and the senders are chosen randomly among the nodes. Multiple simulation runs (100 runs per setup on average) with different seed numbers were conducted for each scenario and collected data were averaged over those runs.

2.3.2 Performance Metrics

The routing protocols are compared according to the following four metrics.

(i) fraction of packets delivered – measures the ratio of the data packets delivered to the destinations and the data packets generated by the CBR source. This number indicates the effectiveness of a protocol.

(ii) end-to-end delay – measured in milliseconds, includes processing, route discover latency, queuing delays, retransmission delay at the MAC, and propagation and transmission times. This number measures the total delay time from a sender to a destination.
(iii) **normalized routing load** – measures the number of routing packets transmitted per distinct data packet delivered to a destination. The routing overhead is an important metric for comparing these protocols as it measures the scalability of a protocol, and its efficiency in terms of throughput and power consumption.

(iv) **packet duplication** – measures the average number of duplicate packets per distinct data packet received by the destinations. A protocol with a high number of duplicate packets can congest the network and waste power of mobile nodes.

The first three metrics were suggested by the IETF MANET working group for routing protocol evaluation [11], and were also used in [6][12].

### 2.3.3 Simulation Results

In our simulation study, we performed sensitivity analysis to investigate the effect of various network parameters on the routing protocols. We present our simulation results in this section.

#### 2.3.3.1 Effect of Mobile Speed

This study is based on 200 nodes with 20 communication sessions. We set up our simulation with zero pause time to stress the mobility in the network. To understand the effect of mobile speed on performance, we varied the speed of the mobile nodes between 10 m/s (or 22 miles/hour) and 25 m/s (or 56 miles/hour).

The simulation results are presented in Figure 11 - Figure 14. They show performance trade-off in some techniques. Although DSR performs comparably to CLA and CBF in terms of end-to-end delay (Figure 12) and number of control packets transmitted per data packet (Figure 13), DSR does poorly in delivering data to their destination (Figure 11). This can be attributed to
the fact that DSR needs to rediscover routes more frequently as node mobility increases. Similarly, GRID, TMNR, GPSR, LAR, and AODV have high end-to-end delay and control packet overhead because links break frequently due to high node mobility (Figure 12 and Figure 13). Under this condition, they need to send more ROUTE DISCOVERY messages. In addition, LAR suffers from inaccurate prediction of the request zone (used to limit the flooding area), which makes flooding the entire network more common.

In the cases of TMNR and GPSR, the high control overhead is caused by maintaining neighbor information (locations), and high end-to-end delay is caused by the inaccurate (outdated) neighbor information. The inaccurate neighbor information causes TMNR and GPSR to forward to non-existing neighboring nodes. In Figure 11, as the mobility increases, the performance of TMNR and GPSR degrades rapidly due to outdated information. Similarly, GRID also has high control overhead caused by maintaining information on the gateway node for each grid (see Figure 13). In Figure 12, the high end-to-end delay in GRID is not only caused by the inaccurate gateway information but also the fact that only a few selected gateway nodes can forward data. The limited number of forwarding nodes (gateway nodes) causes the network throughput to decrease in Figure 11.

In contrast, since CLA and CBF have no connection to break and maintain or neighbor information to update, they have low control overhead, short end-to-end delay, and high successful delivery ratio. Between these two techniques, Figure 14 shows the number of packet duplication for CBF is three times higher compared to that of CLA. This is due to the fact that CLA only allows nodes in selected grid path to forward data packets. In addition, CBF has the “fan-out” effect that is similar to the broadcast storm problem [41] and [56] when forwarding
data packets. We note that we did not study the effect of mobility beyond 25 m/s (or 56 miles/hours) because the performance comparisons can be extrapolated from the trends in the performance behavior.

**Figure 11.** Effect of speed on fraction of packets delivered.

**Figure 12.** Effect of speed on end-to-end delay.
Figure 13. Effect of speed on normalized routing load.

Figure 14. Effect of speed on packet duplication.
2.3.3.2 **Effect of Pause Time**

In this study, we fixed the number of nodes at 200, their speed at 20 \( m/s \), the number of communication sessions at 20, and varied the pause time between 0 to 600 seconds to investigate its effect on performance.

The simulation results are plotted in Figure 15 – Figure 18. We note that as the pause time becomes very long, communication connections are less likely to break and most protocols display about the same performance. Nevertheless, we observe trade-off in the performance metrics among different protocols as the pause time is shorter, i.e., higher mobility. We discuss these conditions as follows.

In Figure 15, AODV has very high fractions of packet delivered under short pause time. This is due to the fact that AODV periodically maintains a local routing table in each node, and a data package can be dynamically rerouted to a new next hop if the current “next hop” has moved away. This helps to reduce the number of lost packets. This strategy, however, incurs a high number of control packets per data packet due to the maintenance of the local routing tables, as seen in Figure 17.

Figure 16 indicates that DSR performs well in terms of end-to-end delay. This is, however, due to the fact that DSR takes relatively longer time to establish a route. Longer routes take too long to connect and many of them become broken soon after they are established under high mobility (i.e., short pause time). Consequently, we observed mostly short routes, two to three hops, in our simulation study with small end-to-end delay. This also explains the low fractions of packets delivered in DSR because many packets delivered over long routes are lost (see Figure 15). DSR also has low control overhead according to Figure 17. Nevertheless, this
is due to the high percentage of packet loss (see Figure 15) and we do not take into account the control packets for these lost packets. For LAR, frequently flooding the entire network is caused by inaccurate predication of the request zone due to short pause time. Flooding the entire network will cause high end-to-end delay (see Figure 16) and high number of packet duplication (see Figure 18). As the pause time increase, LAR performs better due to the more accurate prediction of the request zone.

For GRID, the periodic update of a gateway node is required to notify its existence to the other nodes in its grid even if it stays at the same location. Therefore, the number of control packets stays the same after pause time reaches 100 seconds in Figure 17. Since only gateway nodes can forward data packet, the performance of end-to-end delay also stays the same after pause time reaches 100 seconds in Figure 16. Similar to GRID, TMNR and GPSR have basic maintenance cost (i.e., control overhead) associated with periodic update of neighbor information in Figure 17. Compared to GPSR, TMNR has higher control overhead caused by maintaining additional routing table information. This routing table is used when the destination is near.

From Figure 15 – Figure 18, we see that the performance curves of CLA are essentially flat. CBF has similar performance in terms of Fraction of Packet Delivered, End-to-End Delay, and Normalized Routing Load. This indicates that both CBF and CLA are unaffected by node mobility (i.e., speed or pause time). This means that CBF and CLA are very robust and suitable for a wide range of mobile applications. In terms of the number of duplicate packets received by destinations, most of the routing protocols have on average 2 packets, except for CBF and CLA (Figure 18). Between these two, CBF has 3 times more packet duplication (10 – 12 duplicate packets) compared to CLA.
Figure 15. Effect of pause time on fraction of packets delivered.

Figure 16. Effect of pause time on fraction of end-to-end delay.
Figure 17. Effect of pause time on normalized routing load.

Figure 18. Effect of pause time on packet duplication
2.3.3.3 Effect on Number of Communication Sessions

In this study, we performed sensitivity analysis with respect to the number of communication sessions. We ran our simulation with speed fixed at 20 m/s, pause time at zero, and number of nodes at 200. We varied the number of communication sessions between 5 and 40.

The simulation results are shown in Figure 19 – Figure 22. Again, they show the trade-off among fractions of packets delivered, end-to-end delay, control overhead, and packet duplication. In Figure 19 and Figure 21, AODV and LAR achieve high packet delivered ratio with the cost of high control overhead. For AODV, each node periodically maintains the state of the routing table. This is the reason that AODV has high overhead in Figure 21. For LAR, the overhead cost comes from maintaining and updating request zone and expected zone in Figure 21. Inaccurate request zone causes LAR re-issue route request by floor the network. As the result, LAR has higher end-to-end delay and higher control overhead.

From the simulation, we notice that DSR does not perform well when mobility reaches above 20 m/s (see Figure 19 to Figure 21). In DSR, only source nodes maintain the route. When a route breaks, the source node will attempt to use any other route that it happens to know about or issues another route request to find a new route. However, with mobility that reaches above 20 m/s, a source node cannot robustly adapt to the changes of topology due to high mobility.

For GRID, TMNR, and GPSR, the low fractions of packet delivered ratio and high control overhead are the result of outdated information. For GRID, this out of date information is the gateway nodes in each of the grid cell. For TMNR and GPSR, this out of date information is the neighboring nodes of each node. And since only gateway nodes will forward the data packets, the gateway nodes become bottleneck due to unbalance workload in GRID.
Only contention based forwarding (CBF) and connectionless approach (CLA) can robustly adapt to the changes in the number of communication sessions to maintain good performance regardless of the network conditions. By robust, we mean that both CBF and CLA achieved high successful delivered rate (see Figure 19), low end-to-end delay (see Figure 20), and low control overhead (see Figure 21). Between these two, CLA has a significantly lower number of packet duplications compared to CBF (see Figure 22). This can be attributed to the fact that CLA limits the forwarding area to a grid path.

![Effect of Number of Communication Sessions](image)

**Figure 19.** Effect of number of communication sessions on fraction of packets delivered.
Figure 20. Effect of number of communication sessions on end-to-end delay.

Figure 21. Effect of number of communication sessions on normalized routing load.
2.3.3.4 Effect of Network Density

In this study we assumed the nodes move constantly at 20 m/s without pausing and that each maintains 20 concurrent communication sessions. To examine the effects of network density, we ran the simulation with different numbers of nodes: 50, 100, 150, 200, and 400 nodes.

The results of this study are plotted in Figure 23 - Figure 26. In terms of fraction of packets delivered in Figure 23, AODV and LAR tend to perform the best out of eight routing protocols under density of 100 nodes and 200 nodes, respectively. However, as number of node increases, the number of control packets per data packet also increases for AODV and LAR (see Figure 25).

Again, DSR does not perform well when mobility is above 20 m/s, even if the number of node increases. For GRID, the number of gateway nodes is fixed due the number of grid is fixed
(i.e., one gateway node per grid). Therefore, since only gateway nodes allow to forward data packets, increase the number of nodes in the network did not improve the performance of GRID.

For TMNR and GPSR, the increasing number of nodes in the network causes the two protocols to have more neighbor nodes to maintain. As the results, the number of control packets is also increased in Figure 25. As density increases, the time to determine which neighboring nodes that is closer to destination/next anchor to be the next hop also increase. This selection process become more time consuming as number of nodes increase. Therefore, end-to-end delay also increases for TMNR and GPSR in Figure 24. Since this greedy forwarding approach will choose the next forwarding hop closest to destination and furthest from current node, the connection between the current node and the selected next hop will be very weak (i.e., faster out of radio rage of each other). Therefore, this causes low fraction of packets delivered rate show in Figure 23.

When the node density is sufficiently high (i.e., 150 nodes or more), the CLA is the only scheme that consistently displays good performance under all four metrics. We note that “150 nodes in a 1,000\(m \times 1,000m\) terrain or 100 grid cells” is still a reasonably practical scenario. We observe that the “end-to-end delay” curve of the connectionless technique (CLA) behaves irregularly when the node density is very low, i.e., 50 and 100. This is due to the fact that data packets that fail to reach their final destination are not taken into account in the computation of the end-to-end delay. As a result, the average end-to-end delay is small because only packets relayed over a few hops make it to the destination. This measure increases when there are 100 nodes in the network because the node density now becomes sufficiently high to support longer hop-by-hop connections. In fact, when the density drops below 50 nodes in 1000\(m \times 1000m\)
field (i.e., 140m × 140m per node or 3 grids per nodes), TMNR, CBF, and CLA can no longer forward the data packets. If we continue to increase the number of nodes in the network, the contention based forwarding (CBF) and connectionless approach (CLA) eventually have the option to select the shorter hop-by-hop connections for each packet transmission, resulting in very good end-to-end delay. From this point forward, the performance of the contention based forwarding (CBF) and the connectionless technique (CLA) becomes “flat” given the fixed terrain dimensions. As the density increases, the number of packet duplications increases rapidly for CBF. Thus, CBF is not scalable for high density environments.

![Figure 23. Effect of node density on fraction of packets delivered.](image-url)
Figure 24. Effect of node density on fraction of end-to-end delay.

Figure 25. Effect of node density on normalized routing load.
Figure 26. Effect of node density on normalized packet duplication.

2.3.3.5 Effect of Terrain Area (Scalability)

To study if the techniques under consideration can scale up to facilitate large-area deployment, we increased both the network area and the number of nodes in order to maintain a constant node density (i.e., averaging $70m \times 70m$ per node or two nodes per grid cell):

- $500m \times 500m$ area and 50 nodes
- $1,500m \times 1,500m$ area and 450 nodes
- $1,000m \times 1,000m$ area and 200 nodes
- $2,000m \times 2,000m$ area and 800 nodes

In these four simulation runs, we fixed the node speed at 20 m/s, the pause time at zero, and the number of communication sessions at 20.

The results are presented in Figure 27 – Figure 30. As we increase the network area, the performance of DSR, GRID, TMNR, and GPSR degrade very quickly in terms of the fraction of data packets delivered successfully. Similarly, LAR, AODV, GRID, TMNR, and GPSR degrade rapidly in terms of end-to-end delay and control overhead. When scale up the network in terms
of area and number of nodes, the number of hop or the distance of a connection between source and destination becomes longer. Therefore, route maintenance becomes more costly in term of control packets per data packet for most routing protocols. Also, a longer route has more chance to break due to any one of the node in a connection fail or out of reach. Thus, fraction of packets delivered is also lower as the scale of the network increases for DSR, GRID, TMNR, and GPSR. Although, AODV and LAR can achieve high fraction of packet delivered, both protocols do not perform well in terms of the control overhead and end-to-end delay.

Only the CBF and CLA perform well under three metrics – fraction of packets delivered, end-to-end delay, and normalized routing load. However, we observe that only CLA does not increase the number of packet duplications with the increases in the network area and the number of nodes. Thus, only CLA can scale up to support larger networks. In contrast, CBF degrades rapidly in terms of number of packet duplication when the network area and number of nodes increase.
Figure 27. Effect of terrain area on fraction of packets delivered.

Figure 28. Effect of terrain area on fraction of end-to-end delay.
Figure 29. Effect of terrain area on normalized routing load.

Figure 30. Effect of terrain area on packet duplication.
2.4 Discussion

We introduced a connectionless approach to wireless mobile ad hoc networks called Connectionless Approach (CLA) for MANET’s. Some of the key advantages of this scheme are as follows:

- It has no communication connections to break or maintain. This results in very low control overhead.
- When a node moves away or runs out of battery power, a nearby node can take over the data-forwarding task with no delay. This ensures low packet loss.
- Transmission delayed is optimized for each packet transmission with no overhead. This results in excellent end-to-end delay.

We present a comparative study of eight routing protocols for ad hoc networks in high-mobility environments. The detailed simulators, implemented using GloMoSim, allow us to perform fair and accurate comparisons of these techniques with a broad range of network parameters including mobility, pause time, the number of communication sessions, density, and size of terrain.

We summarize the performance characteristics of these techniques in TABLE 1. In our study we observed that AODV, DSR, LAR, GRID, TMNR, and GPSR have to make a trade-off between the fraction of packets delivered, the end-to-end delay, and the normalized routing load. Although both CBF and CLA perform well in terms of all three metrics; CBF has much higher number of packet duplications compared to all of the other protocols. CLA, on the other hand, is not suitable to low-density environments (i.e., below 50 nodes in 1000m × 1000m field).
In Table 2, we summarize the characteristics of the environments suitable for each protocol. Multi-hop based approaches, such as DSR, AODV, and LAR, are more suitable for conference or meeting applications where mobility is low, pause time is long, communication load is light, density is low, and terrain size is small. Notice that only AODV and DSR do not require any location information provided by GPS. For the Cluster based approach in general, long pause time is needed to maintain up-to-date cluster membership. GRID can support moderate node speed because it only maintains the gateway nodes. However, short pause time can cause GRID to constantly reelect the gateway nodes. Thus, GRID adapts well in an environment with a moderate speed and long pause time. Our simulation shows, the pause time can have great effect on GRID’s performance. However, network throughput is limited by only gateway nodes allow to forward for the cluster approach. For One-hop based approach such as TMNR and GPSR, the need to maintain neighboring information causes the performance to decrease under high mobility. Compare the two protocols, GPSR can adapt to low density due to geographical anchor locations. For connectionless approach, both CBF and CLA perform well and suitable for most environments. However, CBF tend to have higher number of packet
duplication. This leads to media congestion, waste of power, and lower network throughput. Cluster based (i.e. GRID) and one-hop based (i.e. TMNR and GPSR) are suitable for mid-range scale of network such as disaster recovery or sensor network with location service available. Connectionless based approach (i.e. CBF and CLA) is suitable for vehicle network or battle field that has a large terrain, high node density, and nodes moving in a high speed with short pause time. Connectionless based approach can also perform well in a more static environment where nodes move in low speed and have long pause time. Between CBF and CLA, CLA tend to have higher communication load and low end-to-end delay which is ideal for voice and video application. While the results can provide guidelines, the final selection of a routing protocol should also take into account considerations specific to a given application.

Table 2. Applicable Environments of Different Approaches

<table>
<thead>
<tr>
<th>Different Approaches</th>
<th>Mobility</th>
<th>Pause Time</th>
<th>Comm. Load</th>
<th>Density</th>
<th>Terrain Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>AODV</td>
<td>Low</td>
<td>Long</td>
<td>Low</td>
<td>Low</td>
<td>Small</td>
</tr>
<tr>
<td>DSR</td>
<td>Low</td>
<td>Long</td>
<td>Low</td>
<td>Low</td>
<td>Small</td>
</tr>
<tr>
<td>LAR</td>
<td>Low</td>
<td>Long</td>
<td>Low</td>
<td>Low</td>
<td>Small</td>
</tr>
<tr>
<td>GRID</td>
<td>Medium</td>
<td>Long</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>TMNR</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>GPSR</td>
<td>Medium</td>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>CBF</td>
<td>High*</td>
<td>Short**</td>
<td>Medium</td>
<td>Medium</td>
<td>Large</td>
</tr>
<tr>
<td>CLA</td>
<td>High*</td>
<td>Short**</td>
<td>High</td>
<td>High</td>
<td>Large</td>
</tr>
</tbody>
</table>

* High mobility means the protocols able to adapt High mobility as well as Low mobility.

** Short pause time means the protocols able to adapt Short pause time as well as Long pause time.

Notes that such property (i.e., ability to adapt High mobility will also able adapt Low mobility) is similar with other environment characteristics.
3. DYNAMIC ROUTE DIVERSION

In a Mobile Ad Hoc Network (MANET), communication connections need to adapt to frequent unpredictable topology changes due to the mobility, energy constraints, and limited computing power of the mobile hosts. Early solutions address this fundamental requirement by employing techniques that can reconnect a broken link quickly with low overhead. This strategy, however, cannot cope with a high frequency of broken links in a high mobility environment. To address this problem, a few connectionless-oriented techniques, e.g., *Connectionless Approach* (CLA), have emerged. These schemes rely on any mobile hosts along the general direction towards the destination node to help forward the data packets. Extensive simulation results have shown that these methods are more robust, and perform significantly better than connection-oriented techniques. The current connectionless methods, however, may suffer from packet drops since traffic congestion is not considered in the packet forwarding policy.

A standard solution is to leave congestion control to the MAC layer. When serious congestion is confirmed, the source node is informed to search for another route. This simple approach incurs delay, computation overhead, and packet losses. These problems become more visible in traffic-intensive environments such as multimedia applications, where congestion is more probable and the negative impact of packet loss on the service quality is of more significance. Better solutions for congestion have been proposed (e.g., CADV [35], CRP [55], DLAR [32], extension-AODV [36], and extension-DSR [40]). These schemes consider congestion in initial routing to avoid the aforementioned problems. Similar techniques are not available for connectionless-oriented MANETs. In fact, it is difficult to adapt the existing techniques since there is no hop-by-hop route in connectionless-oriented routing. In this chapter,
we investigate a cross-layer design for connectionless-oriented MANETs. We are motivated to make CLA, already a good technique, better.

![Diagram of connectionless approach and connectionless approach with dynamic rerouting](image)

**Figure 31.** Dynamic packets rerouting as a collision avoidance mechanism.

The focus of this chapter is to improve the connectionless approach by taking into consideration traffic congestion in order to minimize packet drops. This is achieved by taking...
into account the workload of individual virtual routers, and dynamically reroute packets as necessary to prevent virtual router overloaded. Dynamic route diversion is illustrated in Figure 31. In Figure 31.a, CLA tries to minimize the number of hops for each of the three communication sessions resulting in the contention area. This problem is addressed in Figure 31.b by detouring some of the data packets from the otherwise contention area.

The primary contributions are as follows:

- We introduce dynamic route diversion mechanism to address the drawback of the current connectionless-oriented approach.
- We apply the dynamic route diversion model to improve two connectionless-oriented techniques:
  - Connectionless Approach with Dynamic Route Diversion (CLA-DRD).
  - Contention-Based Forwarding with Dynamic Route Diversion (CBF-DRD).

We present simulation results to show that with the Dynamic Route Diversion extension, CLA-DRD and CBF-DRD can prevent contention areas and achieve significantly better performance over the original techniques.

3.1 Related Work

Congestion is a dominant reason for packet drops in MANET. In [35], authors found that AODV is ineffective under stressful network traffic situations and proposed a modified version called CADV. CADV select nodes with short queuing delay in routing protocol. While this modification may improve the route quality, the issue of long delay and high overhead when a new route needs to be discovered remain unsolved. Also, this approach only improves the
performance of AODV when an existing route becomes heavily congested. A dynamic load-aware routing protocol (DLAR) was proposed in [32]. DLAR is similar to CADV, the difference being that a node which low routing load is favored to be included in the routing path during the route discovery phase.

CADV, DLAR, as well as most on-demand routing protocols, are single-path. Multipath protocols may be used to reduce the delay due to new-route discoveries. Protocols such as extension-AODV [36] and extension-DSR [40] are example of multipath version of existing on-demand sing-path protocols. This type protocol operates proactively and requires heavy overheads and, therefore, it may not perform as well as an on-demand protocol does in MANETs.

Recently, [55] proposed CRP, a cache-based on-demand multipath routing protocol. CRP balances network routing load better than other on-demand multipath protocols because it sends packets on multiple paths simultaneously based on the current network congestion situation. However, none of the congestion control techniques can be adapt to Connectionless-Oriented approach since there is no hop-by-hop route in connectionless-oriented routing.

3.2 Proposed Solution: Dynamic Route Diversion (DRD)

As we have discussed, Connectionless Approach (CLA) [21] technique and Contention Base Forwarding (CBF) [16] technique are robust with respect to node mobility. Broken links are infrequent because any nodes that happen to be along the forwarding direction can help to relay the data packets. Using only the location of the destination node as the general forwarding/routing direction, however, may result in many forwarding paths crossing each other forming the undesirable contention areas. The consequences are packet drops and nodes within
these contention areas unable to access the media. In this section, we introduce two new connectionless-oriented routing techniques to address this problem.

3.2.1 **Connectionless Approach with Dynamic Route Diversion (CLA-DRD)**

To make the chapter self contained, we first briefly describe our previous work, CLA, in more detail can be found in Chapter 2. We then explain the proposed CLA-DRD technique in details in the following subsection.

3.2.1.1 **Connectionless Approach (CLA)**

In *Connectionless Approach* (CLA), the network area is divided into small non-overlapping grid cells (see Figure 32(a)). Instead of maintaining a hop-by-hop route between the source and destination node, the source selects a list of grid cells that form a “connecting” path between the source and destination. The location of destination is discovered by the CLA’s *location discovery procedure* where a simple broadcasting technique [19] is employed. From a different perspective, each grid cell can be viewed as a *virtual router* in the sense that any physical router (i.e., a mobile node) currently within the virtual router can alternate in forwarding data toward the next virtual router. The communication path consisting of consecutive virtual routers form a *virtual link* (see Figure 32(a)). For example, if node \( j \) and node \( k \) received a data packet from node \( i \) (see Figure 32 (b)). Both node \( j \) and node \( k \) are within the radio range of the sender, node \( i \). However, node \( k \) is farther away, and therefore has a shorter forwarding delay. As a result, node \( k \) will forward the data packet from node \( i \). In this chapter, we use the terms “virtual router” and “grid cell” interchangeably. Similarly, we also use the terms “virtual link” and “forwarding grid path” interchangeably.
Given a virtual router (a grid cell), its physical routers (nodes) compete to forward the data packets according to a *data forwarding procedure*. This function computes a shorter delay for a node farther from the sender and closer to the destination. In this environment, a virtual link is considered broken if one of its virtual routers becomes empty. This is addressed by replacing the empty virtual router with a neighboring virtual router. The fundamental advantages of CLA are twofold. First, a virtual link is much less likely to be broken than a standard route used in conventional connection-oriented techniques; and second, unlike standard routes, the robustness of virtual link is not sensitive to the mobility inherent in MANET.

![Diagram](image)

**Figure 32.** Connectionless Approach.
3.2.1.2 Dynamic Route Diversion for CLA (CLA-DRD)

In order to minimize the size and the number of contention areas, we need to modify the simple forwarding procedure used in CLA. To monitor the congestion and avoid collision within the neighboring (radio range) area, each node tracks any overheard packets in its cell. The congestion is determined based on the ratio between the number packets currently waiting for forward. We can use a variety of metrics at a node to monitor congestion status. For instance, it can be the percentage of all packets discarded for lack of buffer space, the average queue length, the number of packets timed out and retransmitted, the average packet delay, and the standard deviation of packet delay. In all cases, rising numbers indicate growing congestion. For ease of presentation and as a proof of concept, we only use the number of packets currently waiting for forward as criteria to determine the congestion.

Using the location discovery procedure (see Section 2.2.3) in CLA, the location of the destination node is known and the source node is able to begin a communication session. Using the location of the destination node, the source node can compute the Selected Grid Path using path computation procedure (see Section 2.2.4) in CLA. The path computation procedure is based on the line-of-sight between the source node and the destination node. For ease of discussion, we omitted the details of location discovery procedure and path computation procedure which can be found in [21] (or Section 2.2.3) and [22] (or Section 2.2.4). To transmit a data packet, the source node includes the following entries in the data header: Source Node ID, Source Cell ID, Destination Node ID, Destination Cell ID, Current Cell ID, and Selected Grid Path (see Figure 33). The Current Cell ID is the ID of the cell containing the node that is about to forward the data packet. Thus, each intermediate node updates this header field before
relaying the data packet. *Selected Grid Path* is a list of cell IDs which form a connecting path between Source Cell and Destination Cell (see Figure 32).

At any instant in time, a node is associated with a particular virtual router or grid cell. Each node uses a *Job Table* to keep track of current forwarding jobs (packets) in its cell. Each record in this table represents a data forwarding job in progress, which is distinct by a pair of attributes – *Source ID* and *Destination ID*. To age out outdated forwarding jobs, the *Job Table* maintains a *Time to Live (TTL) timer* with each job entry. To avoid contention areas, each node also tracks the workload of its virtual router or cell by maintaining a *Forwarding Job Counter*, the number of forwarding jobs in progress within the grid cell.

<table>
<thead>
<tr>
<th>Source Node ID</th>
<th>Destination Node ID</th>
<th>Current Cell ID</th>
<th>Source Cell ID</th>
<th>Intermediate Cell ID (1)</th>
<th>Intermediate Cell ID (2)</th>
<th>...</th>
<th>Intermediate Cell ID (n)</th>
<th>Destination Cell ID</th>
</tr>
</thead>
</table>

**Figure 33.** Data header.

**Packet Processing Procedure:** When a node receives a data packet, it processes the packet as follows:

In the node’s MAC layer, it has a *Packet Queue* (i.e., see Figure 34). We observe the packets incoming rate and packets’ outgoing rate.

- \( r_i = \text{incoming rate of packets} \)
- \( r_o = \text{out going rate of packets} \)

The incoming rate is updated using a 3-point smooth function as follow:

\[
R_i = \frac{r_{i-1} + r_i + r_{i+1}}{3}.
\]

Like the incoming rate, the outgoing rate is updated as follow:
\[ R_o = \frac{r_{o-1} + r_o + r_{o+1}}{3}. \]

![Packet Queue in MAC layer.](image)

Figure 34. Packet Queue in MAC layer.

The nodes periodically exchange their \( R_i \) and \( R_o \) with other nodes in the same grid cell every \( \text{Sync} \) time units; and synchronization is done by setting the \( \text{GridID}_Ri \) and \( \text{GridID}_Ro \) in each node to the maximum of all the values within the grid. In other words, nodes in the same grid cell generally see the same value for the \( \text{GridID}_Ri \) and \( \text{GridID}_Ro \).

**Data Forwarding Procedure:** When data forward from Source to Destination, “upstream” is any cell or node that is closer to Source, compare to a node itself. Similarly, “downstream” is any cell or node that is closer to Destination, compare to a node itself. When a node \( n \) receives a data packet from node \( m \), the data forwarding procedure is as follows:

1. If \( n \) is the destination node, \( n \) does not forward the data.

2. If \( n \) is not on the Selected Grid Path (see Figure 32), \( n \) does not forward the data.

3. If the \( \text{GridID}_Ri > \text{GridID}_Ro + \text{threshold} \) (i.e., buffer overflow),
   - \( n \) does not forward the data
   - \( n \) broadcasts locally (i.e., one-hop) Congestion Notification (see Figure 35 for the Congestion Notification Packet format) to notify \( m \) that congestion occurred.

If Steps 1, 2, and 3 fail (i.e., \( n \) might need to forward the data), do the following:
a. \( n \) calls Job Counter Update procedure to update the *Forwarding Job Counter*.

b. \( n \) computes a delay time interval: 
\[
\text{DELAY} = \frac{\alpha}{\text{Dist}_n},
\]
where \( \alpha \) is a constant of maximum delay in \( \mu \text{sec} \); and \( \text{Dist}_n \) is the distance between node \( n \) and node \( m \).

c. \( n \) delays the forwarding for the computed time interval, \( \text{DELAY} \). During this delay period, \( n \) discards the same packets arriving from the upstream. Furthermore, it will cancel the forwarding if any one of the following three conditions becomes true:

- \( n \) hears the same packet in downstream (i.e., closer to the destination node).
- \( n \) hears the same packet from a neighboring node on the same grid cell.
- \( n \) hears the Congestion Notification from its cell or a cell in the downstream of the Selected Grid Path.

At the end of the delay period, if the forwarding decision has not been cancelled, \( n \) forward the data.

|--------------------|----------------|----------------|---------------------|---------------------|-------------------------|-------------------------|-------|-------------------------|

Selected Grid Path

**Figure 35.** *Congestion Notification Packet.*

As discussed before, the *threshold* (i.e., number of packets drop) can be based on the nodes’ carrying capacity, percentage of all packets discarded for lack of buffer space, the average queue length, the number of packets timed out and retransmitted, the average packet delay, and the standard deviation of packet delay. We note in Step 4.b that the delay function computes a shorter delay for a node farther from the sender. A flowchart for the above procedure is given in Figure 36.
Figure 36. Data Forwarding Procedure
**Dynamic Route Diversion Request**: When a *non-congested* node, $m$, receives a *Congestion Notification*, $m$ checks if this notification packet is arriving from a downstream cell (i.e., a cell that is closer to the destination node) of the *Selected Grid Path*. If this is the case, $m$ tries to detour the subsequent data packets from the contention area by broadcasting a special route request called *Dynamic Route Diversion Request* (*DRD Request*). This *DRD Request* has three groups of entries (see Figure 37) as follows:

- The first group consists of only the *DRD Source Cell ID* entry. This entry has $m$’s Cell ID to identify the corresponding virtual router as the virtual source node for this *DRD Request*.

- The second group is a list of *DRD Destination Cell ID* entries. Each of these list entries identifies a distinct downstream cell of the *Selected Grid Path*, as a candidate for the destination cell.

- The third group is a list of *DRD Intermediate Cell ID* entries. The virtual routers corresponding to these entries form a grid path (called *DRD Path*) from the source cell identified in the first group entry to one of the candidate destination cells identified in the second group entry.

![Figure 37. DRD Request Packet Format.](image)

**Dynamic Route Diversion Request Procedure**: When a node $n$ receives a *DRD Request*, if it is not a congested node, $n$ performs the following Dynamic Route Diversion
Request Procedure. Nodes that are already congested do not broadcast any DRD related packets to avoid further exacerbating the congestion.

1. \( n \) checks the second group entry of the \textit{DRD Request} to determine if it is on one of the candidate destination cells. If it is, \( n \) adds its cell ID to the DRD Path list and replies with a DRD Reply packet using the hops identified in the DRD Path, but in the reverse order.

2. \textit{Otherwise}, \( n \) checks the third group entry of the \textit{DRD Request} to determine if its grid cell is already in the \textit{DRD Path}. If so, \( n \) simply discards the packet. \( n \) checks the second group entry of the \textit{DRD Request} to determine if it is on one of the candidate destination cells. If it is, \( n \) adds its cell ID to the DRD Path list and replies with a DRD Reply packet using the hops identified in the DRD Path, but in the reverse order.

\textit{Otherwise}, \( n \) adds its Cell ID to the \textit{DRD Path} list and forwards the \textit{DRD Request}. The flowchart for the above procedure is given in Figure 38.

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{figure38}
\caption{DRD Request Procedure.}
\end{figure}

\textbf{DRD Reply Procedure:} The format of the \textit{DRD Reply} packet is shown in Figure 39. Its data entries are similar to those of the \textit{DRD Request} packet. When a node, \( m \), receives a \textit{DRD Reply} packet, \( m \) performs the following DRD Replying procedure:

1. \( m \) checks the second group’s first entry of the \textit{DRD Reply} packet to determine if \( m \) is in the source virtual router (DRD Source Cell). If it is, \( m \) will modify the data
packet’s *Selected Grid Path* by inserting *DRD Path*. Likewise, if *m* is one of the potential destination cells, it will reroute the data packets using the DRD path.

2. Otherwise, *m* checks the third group entry to determine if it is on the grid path leading to the source virtual router. If it is, *m* forwards the *DRD Reply*.

3. Otherwise, *m* discards the *DRD Reply* packet.

<table>
<thead>
<tr>
<th>DRD Reply Cell ID</th>
<th>DRD Source Cell ID</th>
<th>DRD Destination Cell ID (2)</th>
<th>...</th>
<th>DRD Destination Cell ID (x)</th>
<th>DRD Intermediate Cell ID (n)</th>
<th>DRD Intermediate Cell ID (n - 1)</th>
<th>...</th>
<th>DRD Intermediate Cell ID (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Possible Destinations</td>
<td>Reverse DRD Path</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 39.** *DRD Reply* Packet Format.

We give an example in Figure 40 to illustrate the Dynamic Route Diversion Procedures. In this example, the source node and the destination node are in cells 3A and 3N, respectively. The initial grid path consists of 13 grid cells, <3A, 3B, 3C, 3D, 3E, 3F, 3G, 3H, 3I, 3J, 3K, 3L, 3M, 3N>. Let us say, the virtual router corresponding to cell 3D becomes congested, and it broadcasts a *Congestion Notification*. When the adjacent virtual router at 3C receives this notification, it initiates a detour process by broadcasting a *DRD Request*. This *DRD Request* has following entries in its second group as possible destination virtual routers: 3D, 3E, 3F, 3G, 3H, 3I, 3J, 3K, 3L, 3M, 3N. Due to the contention area (see Figure 40), when the *DRD Request* packet arrives at the virtual router at 3K, the *DRD Path* entry is <3C, 2D, 1E, 1F, 1G, 1H, 2J, 3K>. This virtual router responds by sending a *DRD Reply* back to the virtual router at 3C that initiated the *DRD Request*. With the detour information (i.e., *DRD Path*), the virtual router at 3C modifies the *Selected Grid Path* to <3A, 3B, 3C, 2D, 1E, 1F, 1G, 1H, 2J, 3K, 3L, 3M, 3N>;
and the data forwarding process resumes along this new grid path. We note that the grid cells corresponding to the detour path are underscored in the new grid path.

![Diagram](image)

**Figure 40. Example of Dynamic Route Diversion**

### 3.2.2 Contention Based Forwarding with Dynamic Route Diversion

Similar to the CLA-DRD, we extend another connectionless based approach, Contention Based Forwarding (CBF), to improve the performance with Dynamic Route Diversion. To make the paper self contained, we briefly describe Contention-Based Forwarding (CBF) technique. We then explain the proposed CBF-DRD technique in detail in the following subsection.

#### 3.2.2.1 Contention-Based Forwarding (CBF)

In CBF, a node forwards the packets as a single-hop broadcast to all neighbors. The neighbors compete with each other for the “right” to forward the packet. During this contention period, a node determines how well it is suited as a next hop for the packet. The node that wins the contention suppresses the other nodes, thus establishes itself as the next forwarding node. This contention is based on the distance of the nodes to the destination (see Figure 41(a)). For
example, if node $j$ and node $k$ received a data packet from node $i$ (see Figure 41 (b)). Both node $j$ and node $k$ will calculate a contention (e.g., delay) timer according to their respective distances, $Dist_{j,d}$ and $Dist_{k,d}$, to destination node $d$. In this case, node $k$’s timer expires first (i.e., $Dist_{j,d} < Dist_{k,d}$) and broadcasts the data packet to the next node. This will cancel node $j$’s timer to prevent multiple next hops and packet duplication.

![Diagram](image)

**Figure 41.** Contention-Based Forwarding.

### 3.2.2.2 Dynamic Route Diversion for CBF (CBF-DRD)

We use the similar approach as CLA-DRD. Each node keeps the $R_l$ and $R_o$ within its neighbor
(within its communication range). When congestion occurs in an area, a node simply broadcasts a *Congestion Notification* to initiate a *DRD Request* which finds a hop-by-hop route to dynamically detour from the contention area (see Figure 44). The node that initiated the *DRD Request* (i.e., DRD Source Node) will include the distance (i.e., *DRD Distance*) between the destination node and itself. We can draw a circle using the distance value as the radius and the destination node as the center (see Figure 44). Nodes that are outside this circle (i.e., its distance > *DRD Distance*) use Dynamic Route Diversion to forward data packets. When the forwarding node locate within this circular area (i.e., its distance ≤ *DRD Distance*), it switches to the original CBF technique and continuously forwards data packets to destination.

To modify *DRD Request* procedure for CBF, the list of possible destinations is replaced by the distance value called *DRD Distance* (see Figure 42). It is the distance between *DRD Request* node and the destination node (see Figure 44). Any node with the distance between itself and the destination node less than the *DRD Distance* is the possible destinations for *DRD Request*.

![Figure 42. DRD Request Packet Format for CBF- DRD.](image)

**Dynamic Route Diversion Request Procedure:** When a node \( n \) receives a *DRD Request*, \( n \) performs the following Dynamic Route Diversion Request Procedure:

1. \( n \) checks the second group entry of the *DRD Request* to determine if its distance to the destination node is less than the *DRD Distance*.  

   65
2. \( n \) checks the third group entry of the DRD Request to determine if its node ID is already in the DRD Path.

If Steps 1 and 2 fail,

- \( n \) adds its Node ID to the DRD Path list and forwards the DRD Request; otherwise
- \( n \) adds its Node ID to DRD Path list and replies with a DRD Reply using the hops identified in the DRD Path, but in the reverse order.

The flowchart for the above procedure is given in Figure 43. The DRD Reply Procedure is similar for CLA. We replaced the concept of cell with node.

![Flowchart](image-url)

**Figure 43.** DRD Request Procedure for CBF-DRD.

We give an example in Figure 44 to illustrate the Dynamic Route Diversion Procedures for CBF. Let us say, there is a contention area between Source Node and Destination Node, and node \( n \) broadcasts a Congestion Notification. When node \( b \) receives this notification, it initiates a detour process by broadcasting a DRD Request. This DRD Request contains a distance value of \( b \) to Destination Node. Due to the
contention area, when the *DRD Request* packet arrives at node *e*, the *DRD Path* entry is \(<b, c, d, e>\). Node *e* responds by sending a *DRD Reply* back to the node *b* that initiated the *DRD Request*. With the detour information, the data is forwarded using this *DRD Path* to route around the contention area. Once the data packet travel to node *e*, the Contention-Based Forwarding is used from node *e* to the Destination node.

![Diagram of CBF-DRD](image)

**Figure 44.** Example of CBF-DRD

### 3.3 Simulation Study

To evaluate our Dynamic Route Diversion approach, we performed simulations using a network simulator called GloMoSim [60]. This simulator, developed at UCLA, is a packet-level simulator specifically designed for ad-hoc networks. It follows the OSI 7-layer network communication model. Four routing protocols were simulated and compared – *Connectionless Approach* (CLA), *Connectionless Approach with Dynamic Route Diversion* (CLA-DRD),
Contention-Based Forwarding (CBF), and Contention-Based Forwarding with Dynamic Route Diversion (CBF-DRD). In [22], we presented an in-depth evaluation that indicates the connectionless-oriented techniques (i.e., CLA and CBF) perform significantly better than connection-oriented techniques (i.e., AODV, DSR, LAR, GRID, TMNR, and GPSR). Thus, it is only crucial to compare CLA-DRD and CBF-DRD with CLA and CBF.

3.3.1 Performance metrics

The routing protocols are compared according to the following three metrics which were suggested by the IETF MANET working group for routing protocol evaluation [11].

- **Fraction of Packet Delivered** – measures the ratio of the data packets delivered to the destinations and the data packets generated by the CBR source. This number indicates the effectiveness of a protocol.

- **End-To-End Delay** – measured in milliseconds, includes processing, route discover latency, queuing delays, retransmission delay at the MAC, and propagation and transmission times. This number measures the total delay time from a sender to a destination.

- **Normalized Routing Load** – measures the number of routing packets transmitted per distinct data packet delivered to a destination. The routing overhead is an important metric for comparing these protocols as it measures the scalability of a protocol, and its efficiency in terms of throughput and power consumption.
3.3.2 Simulation Parameters

The field configuration is a $1000m \times 1000m$ field, unless it is specified otherwise by the network scenarios. The radio propagation range for each node is 250 meters and channel capacity is 2 Mbits/sec. We used the random waypoint mobility model [60]; that is, each node randomly selects a destination point. When the node reaches this destination point, it pauses for a period of time, and then selects another destination point. Traffic applications are constant bit rate sessions. Each data packet is 512 bytes and the senders are chosen randomly among the nodes.

We ran our simulation with the speed of the mobile nodes at random between 0 m/s and 25 m/s (or 56 miles/hr), random pause time between 0 to 20 second, and number of nodes at 200. The maximum delay $\alpha$ for forwarding procedure is 15 ms. The Maximum Time to Live (TTL) for Job Timer in a Job Table is 20 ms. To period update the Job Counter, the $Grid_{ID-R_i}$ and $Grid_{ID-R_o}$ synchronization time (Sync) is set to 10 ms. The threshold is set to 20% of buffer size. As discussed before, the threshold can be based on the nodes’ carrying capacity, percentage of all packets discarded for lack of buffer space, the average queue length, the number of packets timed out and retransmitted, the average packet delay, and the standard deviation of packet delay. Multiple simulations runs (100 runs per setup on average) with different seed numbers were conducted for each scenario and collected data were averaged over those runs.

3.3.3 Simulation Results

We present the simulation results in this section. We study the effect to the number communication sessions. We varied the number of communication sessions between 10 and 100. The simulation results are shown in Figure 45 – Figure 47.
3.3.3.1 Effect on Number of Communication Session to Fraction of Packet Delivered

In Figure 45, CLA and CBF show the performance decrease when number of communication session increase to 40. This is the reason that contention area of CLA and CBF increased when number of communication session is close to half of total number of nodes. CLA-DRD and CBF-DRD can robustly adapt to the changes in the number of communications to maintain good performance regardless of the network conditions. By robust, we mean that the performance of both CLA-DRD and CBF-DRD stay unaffected with high successful delivered rate when increase of number of communication session.

![Figure 45. Effect of number of communication sessions on fraction of packets delivered.](image)

3.3.3.2 Effect on Number of Communication Session to End-to-End Delay

In Figure 46, increasing the number of communication sessions causes both CLA and CBF have longer end-to-end delay due congestions. The reason is that the collision cause data packet to
drop or forwarding nodes are waiting for the medium in contention area. However, CLA-DRD and CBF-DRD avoid the contention area by using different routes to forward data around this area when contention is detected. Compare to CBF-DRD, CLA-DRD has lower end-to-end delay because the *DRD Path* is still maintaining connectionless approach in terms of cell-by-cell (see Section 3.2.1.2). For CBF-DRD, the *DRD Path* is a hop-by-hop connection with probability of route break. Thus, it has slightly higher end-to-end delay.

![Figure 46. Effect of number of communication sessions on end-to-end delay.](image)

### 3.3.3.3 Effect on Number of Communication Session to Normalized Routing Load

In Figure 47, the CLA-DRD and CBF-DRD have higher control overhead when number of communication is less than 2/3 of total number of nodes. In fact, both approaches require a constant number control overhead. This is not severely affected by number of communication session. There are two reasons that CLA-DRD and CBF-DRD have higher normal routing load.
First, both approaches need to periodic update the Forwarding Job Counter. Second, the procedure of discovering a route around the congestion area also introduces additional control overhead. However, as the number of communication increases, the number of contention areas also increases. In CLA and CBF, data packets are forwarded to contention areas frequently. Those data packets are either lost or waited for media due to congestion. As a result, the need to re-establish connection path or direction of destination nodes is more frequent. This causes both high control overhead and low fraction of packet delivered ratio.

![Effect on Number of Communication Sessions](image)

**Figure 47.** Effect of number of communication sessions on normalized routing load.

### 3.4 Discussion

Connectionless-oriented approach is an exciting new routing technique for more robust communications in mobile ad hoc networks. However, current connectionless-oriented schemes do not take into consideration network congestion in routing data packets. As a result,
contention areas may occur resulting in packet drops. We addressed this problem in this paper by introducing dynamic packets rerouting as a mechanism to avoid collisions. This model was applied to improve the original Connectionless Approach (CLA) and Contention-based Forwarding (CBF) technique with the contention avoidance routing capability.

To extend the CLA technique, we introduced the “virtual router” and “virtual link” concepts. Unlike the CLA that only forwards data, the virtual router (cell) is able to select a new virtual link (route) if necessary. This new perspective of the CLA technique allowed us to formulate the contention avoidance problem as load balancing the virtual routers to prevent router overloaded.

The CBF technique was extended by detecting and avoiding contention areas at each node. Similar to avoiding busy virtual routers in CLA, the data packets are dynamically detoured from the contention areas in CBF to prevent further contention.

To assess the effectiveness of the proposed Dynamic Route Diversion techniques, we performed simulation study using GloMoSim. The simulation results indicate that both new techniques, CLA-DRD and CBF-DRD, offer significant performance gain, with CLA-DRD possessing a small performance edge over CBF-DRD. Both achieve high fraction of packet delivery and low end-to-end delay with minimum increases in control overhead. Furthermore, we observed the workload more balanced and evenly distributed among the mobile hosts resulting in longer life spent for the network.
4. COOPERATION ENFORCEMENT

Mobile Ad hoc NETworks (MANETs) have attracted great research interest in recent years. A mobile ad hoc network is a self-organizing multi-hop wireless network where all hosts (often called nodes) participate in the routing and data forwarding process. The deployment of ad hoc networks does not rely on fixed infrastructures such as router and base station, thereby posing a critical requirement on the nodes to cooperate with each other for successful data transmission. Many works (e.g., [7], [8], and [25]) have pointed out that the impact of malicious and selfish users must be carefully investigated. Existing cooperation enforcement techniques ([7], [8], [25], [28], [37], and [38]) cannot be adapted for some of recent advance in routing protocols. In particular, we are interested in the new Connectionless-Oriented Approach ([21] and [22]). We investigate two such techniques, namely Connectionless Approach (CLA) [21] and Contention-Based Forwarding (CBF) [16], in this paper. These techniques do not maintain a hop-by-hop route for a communication session to minimize the occurrence of broken link. In CLA, the network area is divided into non-overlapping grid cells, each serving as a virtual router. Any physical router (i.e., mobile host), currently inside a virtual router, can help forward the data packet to the next virtual router along the virtual link. This process is repeated until the packet reaches its final destination. Since a virtual link is based on virtual routers which do not move, it is much more robust than physical link. Another scheme, CBF, simply forwards data packets to the next hop without first having to establish the one-hop connection. The nodes that happen to be in the general direction towards the destination node help to forward the data packets.

The goal of this research is to address the security and cooperation issues for connectionless-oriented approach (i.e., CBF [16] and CLA [21]) in wireless ad hoc networks.
There can be both selfish and malicious nodes in a mobile ad hoc network. The selfish nodes are most concerned about their energy consumption and intentionally drop packets to save power. The purpose of malicious node is to attack network using various intrusive techniques. In general, nodes in an ad hoc network can exhibit Byzantine behaviors. That is, they can drop, modify, or misroute data packets. As a result, the availability and robustness of the network are severely compromised. Many works ([7], [8], [25], [28], [37], and [38]) have been published to combat such problem - misbehaving nodes are detected and a routing algorithm is employed to avoid and penalize misbehaving nodes. These techniques, however, cannot be applied to the connectionless-oriented approach since any node in the general direction towards the destination node can potentially help forward the data packets.

The primary contributions of this chapter are as follows:

- We introduce a cooperation enforcement technique, called 3CE (3-Counter Enforcement), for the connectionless-oriented approach.
- We apply the 3CE method to two connectionless-oriented techniques:
  - Connectionless Approach (CLA), and
  - Contention-Based Forwarding (CBF).

We present simulation results to show that with the 3CE features, CLA and CBF can prevent malicious nodes and enforce the cooperation among nodes to maintain the good performance of the network.
4.1 Related Work

The deployment of ad hoc networks does not rely on fixed infrastructures such as router and base station, thereby posing a critical requirement on the nodes to cooperate with each other for successful data transmission. There are several Cooperation Enforcement Techniques to deter misbehaving nodes. In Zhou and Haas [62], authors employ asynchronous threshold security and share refreshing for distributed certification authorities for key management in mobile ad-hoc networks. They take advantage of inherent redundancies in mobile ad hoc networks given by multiple routes to enable diversity coding, allowing for Byzantine failures given by several corrupted node or collusions. The approach is a potentially strong prevention mechanism; however, to the best of our knowledge, the impact on performance of a large scale network and ability to adapt to high mobility has not been published.

Smith, Murthy, and Garcia-Luna-Aceves [52] examine the routing security of distance vector protocols in general and develop countermeasures for vulnerabilities by protecting both routing messages and routing update. They propose sequence numbers and digital signatures for both routing messages and updates. However, distance vector protocols are not suitable to large scale and high mobility network as studied in CBF [16] and CLA [21]. And it is difficult to employ [52] in such a network environment and to adapt to new types of routing protocols.

Buttyan and Hubaux [8] propose incentives to corporation by means of so-called nuggets. Nuggets serve as a per-hop payment in every packet or counter to the secure module in each node to encourage forwarding. Similar approach called Confidant protocol proposed by Buchegger and Le Boudec [7] which propagates the bad reputation of node to more than one node. However, this type of approaches cannot be employ by connectionless-oriented approach. The reasons are
follows: First, malicious nodes can easily cheat the proposed protocols by create and forward packets to a none-existing node or a random node to increase the nuggets or the counters since there is no pre-determined route or a next hop in connectionless-oriented approach. Second, in large scale networks, the connections between two nodes can have large number of hops. Thus, to establish a connection might be very costly or not affordable to some nodes in terms of nuggets.

Marti, Giuli, Lai, and Baker [37] observe increased throughput in mobile ad-hoc network by complementing DSR with a watchdog (for detection of malicious behavior) and a ‘pathrater’ (for trust management and routing policy, every path used is rated), which enable nodes to avoid malicious nodes in their routes. Their approach does not punish malicious nodes that do not cooperate, but rather relieves them from the burden of forwarding for other. In other words, the malicious nodes are rewarded in their behavior. Jiang, Sheu, Hua, and Ozyer [25] proposed a finite-state model to penalize the misbehavior nodes and allow them to rejoin only if the behavior improved. However, in Connectionless Approach (CLA) and Contention-Based Forwarding (CBF), there is no pre-determining next hop. Thus, it is impossible to employ a misbehavior detection mechanism (i.e., watchdog) and a malicious node avoidance routing protocol (i.e., path rater).

Yi, Naldburg, and Kravets [59] propose a modification of AODV with security metrics to path computation and selection. They define trust levels according to organizational hierarchies with a shared key for each level, so that nodes can state their security requirements when requesting a route and only nodes that meet these requirements can participate in the routing.
Again, it is not suitable to connectionless based approach due to no per-determined route or selection process for a route.

Therefore, existing cooperation enforcement techniques cannot be adapted for the \textit{Connectionless-Oriented} Approach since any node in the general direction towards the destination node can potentially help forward the data packets.

\textbf{4.2 Node Configuration and Tamper Proof Module}

The proposed technique is based on nodes with the following configuration. First, nodes are equipped with wireless interface cards that can be switched to detection mode to “detect” data transmission on a “suspicious” node in their proximities. Second, connectionless-oriented routing protocol is employed in the network layer. Without loss of generality, we base our discussion on the more recent techniques developed for routing in MANETs (i.e., Connectionless Approach routing protocol (CLA) and Contention-Based Forwarding (CBF)). Nevertheless, the technique can be incorporated into any location-aid protocols to protect nodes against uncooperative behaviors. Third, reliable communication protocols such as TCP cannot be employed in this type of routing protocols. While TMNR and TBF need to maintain (proactively or reactively) neighbor nodes location information and establish a connection to the next hop before forwarding a data packet, CBF and CLA simply forward data packet without first establishing the link to the next node. Any node that happens to be in the general direction towards the destination node can compete for the “right” to forward data packets.

In addition, similar to the techniques presented in [8] and [25], we also equip each node with a tamper resistant module. All other hardware and software components are susceptible to illicit modifications. We notice that a tamper-proof security module remains controversial [46],
but it proves to be inevitable in a large scale and high mobility network environment. Our approach guarantees that as long as the tamper resistant module is not compromised, nodes cannot benefit from uncooperative behaviors. Some mission critical data is stored in the tamper resistant module. This information include: 1) a unique ID of the node; 2) a pair of public/private keys; 3) a \textit{Forward Request Counter} that counts number of packets that are received and need to be forwarded; 4) a \textit{Forward Counter} that counts number of packets have been forwarded; 5) a \textit{Location Discovery Counter} that counts number of Location Discovery packets initiated by a node; 6) a \textit{Session Table} that keeps track ongoing communication sessions; 7) a \textit{Counter Update Procedure} that updates the three counters; 8) a \textit{Misbehavior Detection Procedure} that initiates the detection to identify a malicious node. Since the tamper proof module maintains information of three counters that are used to determine maliciousness of a node and initiate the detection, hereafter we also refer to this module as the 3C Module, and the proposed technique as the 3CE or 3C Enforcement technique.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{layer_structure.png}
\caption{Layer Structure.}
\end{figure}
The 3C Module inspects Location Discovery packets, Location Reply packets and data packets exchange between the network layer and the MAC layer (see Figure 48); and the module updates the counters as follows: 1) When a new packet arrives at a non-destination node, it updates (i.e., increment by one) its **Forward Request Counter**; 2) When a node forward a packet, it updates (i.e., increment by one) its **Forward Counter**; and 3) When a note initiates a Location Discovery packet, it updates (i.e., increment by one) its **Location Discovery Counter**. In addition, the 3C Module constructs and adds 3C’s header (i.e., the value of three counters) to the Location Discovery packet as in various layers of the OSI model.

### 4.3 3C Module

In a connection-oriented (i.e., hop by hop route) approach, before a node can start a data transmission session to another node, the protocol needs to issue a route request to find a route to the destination node. However, in connectionless-oriented approach, only the location of the destination node is needed. Thus, a Location Discovery packet is broadcasted to find only the destination’s location. Once its location is determined, intermediate nodes can forward data packet according to the general direction towards the destination; and all packets exchanged between nodes are examined by the nodes’ 3C Module.

In a 3C Module, three counters (i.e., **Forward Request Counter**, **Forward Counter**, and **Location Discovery Counter**) are updated according to the counter update procedure. These counters are maintained by the node’s own 3C Module (see Figure 48). Similar to [8] and [25], we assume the 3C Module is a tamper resistant module that malicious users cannot contaminate it. The details of the counters update procedure will be discussed in Section 4.3.1 and 4.3.2.
When a source node $S$ initiates a Location Discovery packet, node $S$’s 3C Module adds the 3C’s header to the Location Discovery packet as in various layers of the OSI model. **3C header** contains the value of three counters (i.e., *Forward Request Counter*, *Forward Counter*, and *Location Discovery Counter*) of node $S$. Based on this header, neighboring nodes of $S$ can decide to forward or discard the Location Discovery packet. If a node $n$ “suspects” the source node is misbehaved, $n$ invokes its **Misbehavior Detection Procedure**. A node suspects another node is misbehaving if one of the following is true: a) the *Forward Ratio* (i.e., ratio of *Forward Counter* to *Forward Request Counter*) of $S$ falls below the *Forward Ratio* of $n$; or b) the *Request Ratio* (i.e., ratio of the *Location Discovery Counter* to *Forward Counter*) of $S$ rises above the *Request Ratio* of $n$. If so, $n$ exchanges 3C information (i.e., the value of the three counters) with its neighboring nodes to determine the network condition in the local area (i.e., $n$’s neighboring nodes). If the source node $S$ is identified (by **Misbehavior Detection Procedure**) as misbehaving, its neighboring nodes will penalize this node by not forwarding $S$’s Location Discovery packets.

In order for malicious nodes to rejoin the network, non-malicious nodes still allow malicious nodes to participate in forwarding data. Unlike many techniques that avoid the malicious nodes during the routing procedure, our approach allows malicious nodes to rejoin the network by contributing its share (i.e., forwarding data for others) of network workload. This way, nodes are given more incentive to act collaboratively. By forwarding data packets for other nodes, a malicious node can increase its *Forward Counter*. When its ratio of *Forward Request Counter* to *Forward Counter* rises above threshold $\alpha$ and its ratio of *Location Discovery Counter* to *Forward Counter* falls below threshold $\beta$, the malicious node will again be allowed
to join the network, i.e., its neighboring nodes again help forward its Location Discovery packets. We elaborate the above processes in the following sections.

### 4.3.1 Counters Update during the Location Discovery Phase

As mentioned earlier, a node needs to find the location of the destination before it can start to send data packets in connectionless-oriented protocols such as CBF and CLA. A node can initiate a Location Discover procedure, receive a Location Discovery packet, or forward/reply a Location Discovery packet. To initiate a Location Discover procedure, a source node broadcasts a Location Discovery packet.

**Location Discovery packet:** Location Discovery packet contains the following information: source node ID\(\text{source}_\text{ID}\), source node’s location \(\text{S}_\text{cell}_\text{ID}\), destination node ID \(\text{destination}_\text{ID}\), destination node’s location \(\text{D}_\text{cell}_\text{ID}\), forward node ID \(\text{forward}_\text{ID}\), and forward node’s location \(\text{F}_\text{cell}_\text{ID}\).

When a node receives a Location Discovery packet, it checks if it is the destination node. If so, it returns a Location Reply packet that contains its location \(\text{D}_\text{cell}_\text{ID}\); otherwise, if the node did not see this Location Discovery packet before, it adds its ID and its cell ID (i.e., forward node ID – \text{forward}_\text{ID} and the currently location – \text{F}_\text{cell}_\text{ID}) and broadcasts the Location Discovery packet to other nodes. In Figure 49, we show the data forwarding procedure for CLA in Routing Layer. The same procedure can be applied to CBF.

**Session Table:** Each node maintains a *Session Table* in its 3C Module to keep track all the ongoing communication session. An ongoing communication session is identified by a session ID which is a pair of source ID and destination ID of the communication session. This table contains the following information for each entry (i.e., communication session): *session ID*
(i.e., a pair of source_ID and destination_ID) and a time to live (TTL) timer. An entry is deleted from the Session Table when one of the following information is true: (i) A communication session ended; (ii) Entry’s TTL (time to live) timer expired; (iii) Entry belongs to an identified malicious node. An entry’s TTL timer is reset when a packet received such that: a) the packet corresponds to this entry (i.e., source_ID and destination_ID = session_ID); and b) it is not from a malicious node.

4.3.1.1 Initiate Location Discovery

When a Location Discover procedure in the routing layer passes an initiated Location Discovery packet to the 3C Module, it processes the packet and updates the Location Discover Counter as follows (see Figure 49):

1. The 3C Module determines if this Location Discovery packet belongs to one of the initiator’s (i.e., the source node’s) ongoing communication session in the Session Table. If it does, go to Step 2; otherwise, go to Step 3.

2. The 3C Module increments the Location Discover Counter by one and adds it to the Session Table (and go to Step 3).

3. The 3C Module adds a 3C header containing the values of the three counters (i.e., Forward Request Counter, Forward Counter, and Location Discover Counter) to this Location Discovery packet before passing it to the MAC Layer for broadcast to other nodes.

In the connectionless-oriented approach, the destination of a communication session is periodically updated according to the mobility of the destination node. The location of the source
node is updated by piggybacking the location information in the data packets. However, a source node sometime needs to re-discover the location of a destination node due to packet losses caused by congestion, mobility, or channel errors. Thus, we differentiate between the initial location discovery and the location discovery that is re-establishing an ongoing communication session.

4.3.1.2 Receive Location Discovery Packet

When a Location Discovery packet broadcast from a node $m$ to any of its one-hop neighbor node $n$, $n$’s MAC Layer passes the packet to its 3C Module for processing the Location Discovery packet and updating the **Forward Request Counter** as follow (see Figure 49):

1. The 3C Module determines if $m$ is the source node that initiated this Location Discovery packet (i.e., packet’s $source_{ID} = packet’s forward_{ID}$). If so, go to Step 2; otherwise, go to Step 3.

2. If $m$ is the source node of this Location Discovery packet, the 3C Module in $n$ uses the information in the packet’s 3C header to determine if there is a need to start the detection procedure to examine $m$’s behavior. We will discuss when to initiate the misbehavior detection and the procedure for misbehavior detection in 4.3.3 and 4.3.5, respectively. If node $m$ is confirmed to be misbehaving, the 3C Module of node $n$ discards the packet (as punishment); otherwise, go to Step 3.

3. Node $n$ keeps records of ongoing communication session in its **Session Table**. If the arriving Location Discovery packet’s $source_{ID}$ and $destination_{ID}$ matches an entry in node $n$’s **Session Table** (e.g., packet’s $source_{ID} + destination_{ID} = session_{ID}$), its
3C Module resets the *time to live* (TTL) timer of the corresponding entry. Next, the Location Discovery packet is then passed on to the routing layer (Step 5).

4. If the Location Discovery packet is not belonged to any ongoing session in the *Session Table* (e.g., packet’s `source_ID + destination_ID ≠ session_ID`), the 3C Module updates the *Session Table* and increases the **Forward Request Counter** by one. The 3C Module then passes the Location Discovery packet to the routing layer for further processing (Step 5).

5. Depending on different routing protocols (e.g., *CLA* and *CBF* protocol), node *n* can discard the packet, continue to forward (i.e., pass back down to lower layers), or initiate a reply procedure (i.e., reach the destination). In Figure 49, we show the routing protocol for CLA in the Routing Layer.

### 4.3.1.3 Forward or Reply Location Discovery Packet

Depending on the role of a node in a communication session (e.g., forwarding node or destination node), a node can forward the Location Discovery packet, reply the Location Discovery packet with a Location Reply, or discard the Location Discovery packet according to its routing protocol. A Location Reply packet is generated by a node’s Routing Layer when a Location Discovery packet arrived at a destination. This destination node needs to reply the source node of the Location Discovery packet. If a node is the destination, its Routing Layer generates a Location Reply packet and passes this reply packet to 3C Module.

When Routing Layer submits a Location Discovery packet or a Location Reply to 3C Module, 3C Module processes the Location Discovery packet and updates the **Forward Counter** as follows:
1. 3C Module determines if the Location Discovery packet or the Location Reply packet matches an entry in the *Session Table*. To determine if the Location Reply packet matches an entry in the *Session Table*, 3C Module simply reverses the order of *source_ID* and *destination_ID* of this packet. If the packet matches an entry in the *Session Table*, go to Step 2. Else, the packet is discarded because a malicious node can generate dummy packet to increase its *Forward Counter* to avoid detection.

2. 3C Module increases the *Forward Counter* by one. Then, the Location Discovery packet or the Location Reply packet is passed to MAC Layer.

### 4.3.2 Counters Update during the Data Forwarding Phase

Once the location of the destination node is determined, the source node can start a communication session. In connectionless-oriented approach, nodes simply forward data packets without first establishing the link to the next node. Any node that happens to be in the general direction towards the destination node can compete for the “right” to forward data packets. When a source node \( s \) starts to send the data packet from routing layer to 3C Module, \( s \)’s 3C Module simply passes the data packet to the MAC layer without updating any counter.

#### 4.3.2.1 Receive Data Packet

When a node \( n \) receives a data packet, its MAC Layer passes the data packet to its 3C Module. Then, 3C Module updates the *Forward Request Counter* as follows:

1. 3C Module determines if the data packet corresponds to a communication session in \( n \)’s *Session Table*. If so, go to Step 2.
2. n’s 3C Module resets the time to live (TTL) timer of the corresponding entry in the Session Table and pass the data packet to the routing layer. Depend on different routing protocols, the data packet is either discarded or forwarded.

3. If the data packet is not belonged to any ongoing session in the Session Table, the 3C Module updates the Session Table and increases the Forward Request Counter by one. The 3C Module passes the Location Discovery packet to the routing layer for further processing (e.g., discard or forward data packet).

4.3.2.2 Forward Data Packet

Depend on the routing protocol, the data packet is either discarded or forward (see the Routing Layer in Figure 49). In connectionless-oriented approach, every node has equal probability of participate in the data forward procedure. If the routing layer decides to forward data packet, it then returns a data packet to 3C Module. The 3C Module processes the data packet and updates the Forward Counter as follows:

1. 3C Module determines if the data packet matches any entry in the Session Table. If so, it increases the Forward Counter by one and passes the data packet to the MAC layer.

2. Else, the data packet is discarded. We discard any packets that are not in the Session Table due to the same reason as discussed in Section 4.3.1.3. A malicious node can generate dummy packets to avoid evoking the Misbehavior Detection procedure.
Figure 49. Update the counters during the Location Discovery phase.
4.3.3 Initiate Misbehavior Detection

By modifying its own routing protocol, a malicious node can intentionally drop (i.e., discard) packets to save its power. However, in the connectionless-oriented approach, every node has an equal chance to participate in a forwarding process. Thus, 3C Module needs to determine to whether to “invoke” the Misbehavior Detection procedure. In order to determine if there is a need to invoke the Misbehavior Detection procedure, 3C Module exams the 3C header in the Location Discovery packet and calculates two ratios, \( \text{Forward Ratio (FR)} \) and \( \text{Request Ratio (RR)} \) as follow:

\[
\text{Forward Ratio}_i (FR_i) = \frac{\text{Forward Counter}_i}{\text{Forward Request Counter}_i},
\]

\[
\text{Request Ratio}_i (RR_i) = \frac{\text{Location Discovery Counter}_i}{\text{Forward Counter}_i},
\]

where \( i \) is the node that initiated this Location Discovery packet (i.e., the source node).

When a node \( n \) receives a Location Discovery packet from a node \( m \), \( n \)’s 3C Module checks if \( m \) is the initiator (i.e., source node) of this Location Discovery packet using the information included in the packet (see Section 4.3.1). If \( m \) is not the initiator, \( n \)’s 3C Module does not invoke the detection procedure. Then, this Location Discovery packet passes to the \textbf{Counter Update procedure} for further process (see Figure 49). If \( m \) is the initiator of this Location Discovery packet, \( n \)’s 3C Module checks the 3C header included in this Location Discovery packet for the following conditions:
1. $FR_m < FR_n$

2. $RR_m > 1.2 \times RR_n$

If one of the above condition is true, $n$’s 3C Module broadcasts a 3C packet (including $n$’s 3C information) to its one-hop neighbor nodes. When a node receives $n$’s 3C packet, it replies with its own 3C information. When $n$ receives its neighboring nodes’ replies, $n$ calculates the *Local Average Forward Ratio (LAFR)*. This ratio is calculated as follow:

$$LAFR_n = \frac{\sum_{i=1}^{k} (FR_i) + FR_n}{k + 1}$$

, where $k$ is number of neighboring nodes for $n$ (excluding $m$).

In MANET, network conditions, such as density and congestion, can change dynamically. Thus, the *Local Average Forward Ratio*$_n$ (LAFR$_n$) is merely the local network condition around $n$. If $FR_m \geq LAFR_n$, it means that network condition at area of $m$ might be congested which causes $m$ not forward packets. Thus, we do not need to invoke the Misbehavior Detection procedure. On the other hand, if $FR_m < LAFR_n$, then $m$ might be misbehaving by not forwarding packets. In this case, $n$ activates its Detection Mode. Notice that all the neighboring nodes of $m$ and $n$ can activate its Detection Mode (but not at same time) because their Forward Ratios are similar. When a node activates its Detection Mode, it continues to forward for other nodes except for the suspicious node.

To avoid evoking the Misbehavior Detection procedure, malicious nodes can initiate dummy packets to increase their own Forwarding Counter. Although, by doing so, malicious nodes defeat the purpose of saving power. Nevertheless, 3C Module can prevent this
misbehavior act by compare the outgoing packets against the Session Table. If the packet does not match any entry in the Session Table, 3C Module discards this dummy packet.

### 4.3.4 Detection Mode

The Detection Mode has two states: Listening-State and Detecting-State. Initially, a node in the Detection Mode is set to Listening-State. In the Listening-State, a node $n$ waits for a random period of time. During this delay period of time, $n$ does the following:

1. If $n$ hears a Detection packet from another node to test node $m$ (i.e., the suspect node), $n$ resets the delay time. A Detection packet is generated by Misbehavior Detection procedure to test a suspicious node.

2. If $n$ hears a Detection packet been forwarded by $m$, $n$ exits the Detection Mode. By exiting the Detection Mode, $n$ forwards $m$’s Location Discovery packet. Similarly, all other nodes that are in their Detection Mode (Listen-State) hear $m$ forwarded the Detection packet will exist their Detection Mode.

At the end of delay period, node $n$ enters the Detecting-State. In the Detecting-State, $n$ invokes the Misbehavior Detection procedure to determine if $m$ is a malicious node.

### 4.3.5 Misbehavior Detection Procedure

The detection mechanism can be implemented as a software application as proposed in [8] for lower cost. Alternatively, it can also be implemented as a build-in component of the temper resistant module for better security. Without loss the generality, we base our discussion on the latter option.
The purpose of the Misbehavior Detection procedure is to detect uncooperative behaviors that result in disruption or degradation of data transmission. We focus on network layer attacks and do not address lower level threats such as physical layer jamming and MAC layer disruptions. The attacks contained by the Misbehavior Detection Module are as follows. First, the Misbehavior Detection procedure is invoked if there is a suspicion of dropping packets was detected during the location discovery phase. Second, the Misbehavior Detection procedure captures malicious users who deliberately discard packets that they are obligated to forward either for selfish purposes or to mount denial of service attacks.

When a node \( n \) invokes its Misbehavior Detection procedure to detect a suspect node \( m \), the procedure is as follows:

1. \( n \) calculates a virtual link (see Figure 1(a)) using the location information (i.e., cell ID) contained in \( m \)’s Location Discovery packet.

2. Based on this virtual link, \( n \) generates a Detection packet (i.e., similar to regular data packet). The source location and destination location of this Detection packet are as follow:
   - Source node’s location (\( S\_cell\_ID \)) of this Detection packet is the cell behind of \( n \), relative to \( m \).
   - Destination node’s location (\( D\_cell\_ID \)) of this Detection packet is the cell behind of \( m \), relative to \( n \).

3. Next, \( n \) broadcasts this Detection packet. All the neighboring nodes of \( m \) are in Detection Mode and will not forward this Detection packet.
4. \( n \) waits for a \( t \) period of time (\( t = \) maximum delay time in the routing layer).

5. At the end of the delay, if \( n \) does not receive the Detection packet forwarded by \( m \) (i.e., \( forward_{\text{ID}} = m \)), \( n \) repeats the process again for two times (total of 3 times).

If \( n \) receives the detection packet which is forwarded by \( m \), \( n \) (and all the neighboring nodes of \( m \)) exits the Detection Mode. \( n \) forwards \( m \)'s Location Discovery packet because \( m \) has passed \( n \)'s Misbehavior Detection procedure. If \( n \) does not receive the detection packet from \( m \), \( n \) punishes \( m \) by discard \( m \)'s Location Discovery packet for period of \( t_{\text{punish}} = C \times (LAFR_n - FR_m) \). Thus, the punishment period is proportion to individual (misbehaving) node’s misbehaved level.

![Virtual link for a Detection packet.](a)

![Radio Range of n](b)

**Figure 50.** Virtual link for a Detection packet.

### 4.4 Simulation Study

We conducted various experiments to verify the effectiveness of the proposed 3CE (3-Counter Enforcement) scheme in enhancing performance of mobile ad hoc network. In this section, we first introduce the simulation setup and parameters. We then study the proposed technique based on various performance metrics.
4.4.1 Schemes Implemented

We implemented three schemes, namely the reference scheme, the defenseless scheme and the proposed 3CE scheme, for performance evaluation. In the reference scheme, all the nodes act collaboratively and relay data for each other. In the defenseless scheme, a certain fraction of nodes are misbehaving as they failed to participate in forwarding procedure. In other words, these nodes discard any packets not destined at them. No detection or prevention mechanism is implemented so that the network is totally “defenseless”. Finally, in the proposed 3CE scheme, misbehaving nodes are detected and punished. A malicious node can recognize itself is been punished when Location Discovery packets of the node has been dropped four consecutively times. Once malicious nodes recognized themselves been punished, they participate in forwarding data to rejoin the network. We varied the Initiate Detection Threshold (IDT) from 1.0, 1.2, 1.4, and 1.6. This threshold determines the percentage of a node require to participated in forward procedure in order not to initiate the 3C’s detection procedure. For example, when the threshold is set to 1.2, a node is allowed of 20% of packet drop due to either network condition or mobility.

4.4.2 Simulation Setup

All the experiments were conducted using GlomoSim [60]. Experiments were based on a mobile ad hoc network with 450 nodes and 90 communication sessions within a 1500 by 1500 meter two dimensional space. Each communication session, source node and destination node are randomly selected (i.e., both normal nodes and misbehaving nodes). Traffic applications are constant-bit-rate sessions. Each data packet is 512 bytes. For the CLA, each grid cell is 100 by 100 meter. The maximum delay time (t) is set to 2 seconds. All nodes employ 802.11 at the MAC layer.
Each node has a radio range of about 250 meters. The random waypoint model was used to model the mobility of hosts. Multiple simulation runs (100 runs per setup on average) with different seed numbers were conducted for each scenario and collected data were averaged over those runs. The total simulation duration for each run was 60 minutes (3600 seconds). We varied the number of misbehaving nodes (i.e., 5%, 10%, 20%, and 30% of total number of nodes) and node mobility (i.e., 10 m/s to 25 m/s or 22 mile/hr to 56 mile/hr). Initially, misbehaving nodes drop all the received packets. Once misbehaving nodes have been identified (i.e., all their Location Discovery packets are drop by other neighboring nodes), they behave normally until they are no longer identified as misbehaving nodes (i.e., their Location Discovery packets are forwarded by others).

### 4.4.3 Metric

In the experiments, we evaluated the proposed scheme based on the following six metrics: (i) **Packet delivered ratio** ($P$): The ratio of the data packets delivered to the destinations and the data packets generated by the CBR source. This measures the rate at which effective data transmission is performed. It is also a good indicator of the degree of collaboration among the nodes. (ii) **Misbehaving node detection ratio** ($D$): The ratio of the number of misbehaving nodes that were correctly identified to the total number of misbehaving node that have actually acted uncooperatively during the simulation. (iii) **False accusation ratio** ($F$): The ratio of the number of 3C Modules that incorrectly accused benign hosts to the overall number of misbehaving nodes that 3C Module identified. (iv) **Control overhead ratio** ($C$): The ratio of the number of routing packets transmitted per distinct data packet delivered to a destination. (v) **End-to-end delay** ($E$): The number measured in milliseconds, includes detecting and processing
malicious nodes delay, route discover latency, queuing delays, retransmission delay at the MAC, and propagation and transmission times. This measures the total delay time from a sender to a destination (without communication sessions that belong to misbehaving nodes). (vi) **Active detection ratio** ($A$): The ratio of the number of nodes activated their Detection Mode per misbehaving node’s location discover packet.

### 4.4.4 Experimental Results

We present the simulation results in this section.

#### 4.4.4.1 Packet Delivered Ratio

By employing the proposed scheme, significantly more data can be successfully delivered to the destinations since nodes are now required to participating in data forwarding. Figure 51 and Figure 52 depict the practical scenarios where the number of malicious node is 10% and 20% of the total nodes. We observe in the case of fewer malicious nodes (less than 10%), the two protocols with 3CE (i.e., **CLA-3C** and **CBF-3C**) have very close throughput to the references **CLA** and **CBF** (i.e., **CLA-Reference** and **CBF-Reference**). Notice that the performance of 3CE scheme is slightly less than the reference scheme. This is due to two reasons: 1) misbehavior nodes are not 100% detected (i.e., see section 4.4.4.2, the 3CE’s misbehaving detection ratio is about 87%); and 2) the false accusation ratio is not 0% (i.e., see section 4.4.4.3, some nodes are been miss identified as malicious nodes). Also, notice that in the reference scheme, even all the nodes act collaboratively and relay data for each other, network condition (e.g., channel error, congestion, and mobility) are still the main causes for packet loss. Never the less, the results show that the proposed 3CE scheme can minimize the effect of malicious nodes to the network.
In both cases, the proposed technique improves the deliver ratio by more than 25% compared to the defenseless scheme.

**Figure 51.** Packet Deliver Ratio ($P$) with 10% Malicious Nodes.

**Figure 52.** Packet Deliver Ratio ($P$) with 20% Malicious Nodes.
In Figure 53 and Figure 54, we varied the *Initiate Detection Threshold* from 1.0, 1.2, 1.4, and 1.6. This means a node is allowed to drop from 0%, 20%, 40%, and 60% of total packets received without initiate 3CE’s detection mode. We observe in the case of *Initiate Detection Threshold* = 1.6 (60% of tolerable drop rate), the performances of the proposed technique decreased to the defenseless scheme. However, if we decrease the *Initiate Detection Threshold* = 1.0 (0% of tolerable drop rate), we do not gain additional improvement on the deliver ratio. In fact by doing so, the performances of Misbehaving node detection ratio ($D$), False accusation ratio ($F$), Control overhead ratio ($C$), and Active Detection ratio ($A$) are decreased (see following sections). Based on the simulation results, the ideal *Initiate Detection Threshold* is 1.2 (20% of tolerable drop rate) with probability of error of 5%.

Another important factor to the performance of packet deliver ratio is the speed of mobility. Due to mobility of mobile hosts, addressing frequent and unpredictable topology changes is fundamental to MANET research. As the mobility of node (e.g., speed) increase, the performance of all three schemes (i.e., 3CE, reference, and defenseless) are decreased. Similarly, when we increased mobility and number of malicious nodes (see Figure 53 and Figure 54), the packet deliver ratio is also decreased as the result. However, consider of mobility increased from 10 m/s (or 22 miles/hour) to 25 m/s (or 56 miles/hour), the deliver ratio is only drop average 20%. Thus, the protocol is still suited for many applications (e.g., video and audio) with error correction code.
Figure 53. Packet Deliver Ratio ($P$) with 10% Malicious Nodes.

Figure 54. Packet Deliver Ratio ($P$) with 20% Malicious Nodes.
4.4.4.2 Misbehaving Node Detection Ratio

We list the results of misbehaving node detection ratio for various simulation scenarios in Table 3. They indicate that the proposed misbehaving node detection mechanism is very effective. In most cases with the \textit{Initiate Detection Threshold} = 1.2 (or 20% of tolerable drop rate), the 3CE’s detection ratio is about 87%. However, when the threshold increased to 1.6 (60% of tolerable drop rate), the 3CE’s detection ratio decrease to about 50%. This indicated that it important to select the acceptable threshold for the proposed technique. The results (with correct threshold selected) demonstrate that on-demand misbehaving node detection is applicable. Since the proposed 3CE technique can adapt by the connectionless oriented approach, it is ideal for highly dynamic MANETs such as vehicle-to-vehicle networks.

| Table 3. Detection Ratio (D) of CLA and CBF with 3CE. |
|---------------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Protocol                        | Speed (m/s)     | 10              | 15              | 20              | 25              | Tolerable Drop rate (%) | 0% | 20% | 40% | 60% | 0% | 20% | 40% | 60% |
| CLA                             | CLA             | CBF             | CLA             | CBF             | CLA             | CBF             | CLA             | CBF             | CLA             | CBF             | CLA             | CBF             |
| 5% misbehaving nodes            | 96%            | 96%            | 89%            | 87%            | 77%            | 75%            | 59%            | 62%            | 96%            | 94%            | 88%            | 85%            | 75%            | 76%            | 55%            | 54%            |
| 10% misbehaving nodes           | 93%            | 92%            | 91%            | 90%            | 79%            | 77%            | 58%            | 60%            | 92%            | 91%            | 89%            | 87%            | 77%            | 74%            | 54%            | 52%            |
| 20% misbehaving nodes           | 94%            | 93%            | 91%            | 90%            | 75%            | 72%            | 58%            | 60%            | 93%            | 92%            | 85%            | 87%            | 73%            | 73%            | 54%            | 52%            |
| 30% misbehaving nodes           | 93%            | 91%            | 91%            | 88%            | 68%            | 66%            | 52%            | 54%            | 91%            | 92%            | 87%            | 85%            | 66%            | 65%            | 50%            | 49%            |

4.4.4.3 False Accusation Ratio

We report the false accusation ratios of the proposed 3CE scheme under various scenarios in Table 4. We conclude that in all node mobility scenarios with the \textit{Initiate Detection Threshold} =
1.2 (or 20% of tolerable drop rate) the false accusation ratio is very low. We observe that this ratio is higher when the speed of nodes is increased. This is due to the fact that some of the suspect nodes moved out of the detection node’s radio range and were thus incorrectly classified by 3CE’s Misbehaving Detection procedure as misbehaving nodes, thereby lifting the false accusation ratio. In addition, by decrease the Initiate Detection Threshold, the false accusation ratio is increased. The reason is that nodes drop packets due to the network condition were identify as misbehaving nodes. Nevertheless, further investigation of simulation log files shows that under simulation configuration with the Initiate Detection Threshold = 1.2, on average less than four nodes was incorrectly accused. Both results indicate that the proposed detection mechanism is able to detect most of the in-cooperative nodes with very low false accusation ratio.

### Table 4. False Accusation Ratio (F) of CLA and CBF with 3CE.

<table>
<thead>
<tr>
<th>Speed (m/s)</th>
<th>Tolerable Drop rate (%)</th>
<th>0%</th>
<th>20%</th>
<th>40%</th>
<th>60%</th>
<th>0%</th>
<th>20%</th>
<th>40%</th>
<th>60%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>CLA</td>
<td>CBF</td>
<td>CLA</td>
<td>CBF</td>
<td>CLA</td>
<td>CBF</td>
<td>CLA</td>
<td>CBF</td>
<td>CLA</td>
</tr>
<tr>
<td>5% misbehaving nodes</td>
<td>8%</td>
<td>7%</td>
<td>0%</td>
<td>1%</td>
<td>3%</td>
<td>5%</td>
<td>8%</td>
<td>7%</td>
<td>5%</td>
</tr>
<tr>
<td>10% misbehaving nodes</td>
<td>8%</td>
<td>11%</td>
<td>1%</td>
<td>1%</td>
<td>4%</td>
<td>7%</td>
<td>8%</td>
<td>7%</td>
<td>7%</td>
</tr>
<tr>
<td>20% misbehaving nodes</td>
<td>9%</td>
<td>10%</td>
<td>1%</td>
<td>1%</td>
<td>5%</td>
<td>7%</td>
<td>8%</td>
<td>8%</td>
<td>9%</td>
</tr>
<tr>
<td>30% misbehaving nodes</td>
<td>12%</td>
<td>11%</td>
<td>2%</td>
<td>3%</td>
<td>6%</td>
<td>7%</td>
<td>8%</td>
<td>9%</td>
<td>17%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Speed (m/s)</th>
<th>Tolerable Drop rate (%)</th>
<th>0%</th>
<th>20%</th>
<th>40%</th>
<th>60%</th>
<th>0%</th>
<th>20%</th>
<th>40%</th>
<th>60%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>CLA</td>
<td>CBF</td>
<td>CLA</td>
<td>CBF</td>
<td>CLA</td>
<td>CBF</td>
<td>CLA</td>
<td>CBF</td>
<td>CLA</td>
</tr>
<tr>
<td>5% misbehaving nodes</td>
<td>7%</td>
<td>8%</td>
<td>3%</td>
<td>2%</td>
<td>6%</td>
<td>6%</td>
<td>7%</td>
<td>8%</td>
<td>12%</td>
</tr>
<tr>
<td>10% misbehaving nodes</td>
<td>8%</td>
<td>8%</td>
<td>2%</td>
<td>2%</td>
<td>7%</td>
<td>6%</td>
<td>8%</td>
<td>8%</td>
<td>14%</td>
</tr>
<tr>
<td>20% misbehaving nodes</td>
<td>10%</td>
<td>11%</td>
<td>2%</td>
<td>2%</td>
<td>7%</td>
<td>8%</td>
<td>10%</td>
<td>11%</td>
<td>18%</td>
</tr>
<tr>
<td>30% misbehaving nodes</td>
<td>13%</td>
<td>16%</td>
<td>4%</td>
<td>5%</td>
<td>11%</td>
<td>9%</td>
<td>10%</td>
<td>11%</td>
<td>23%</td>
</tr>
</tbody>
</table>
4.4.4.4 Control Overhead Ratio

With 20% of malicious nodes and the *Initiate Detection Threshold* = 1.2 (or 20% of tolerable drop rate), we observe that the Control Overhead Ratio is higher when the speed of nodes is increased (see Figure 55). Similar to False Accusation Ratio, this is due to the fact that some of the suspect nodes moved out of the detection node’s radio range and were thus cause some nodes to invoke 3CE’s Misbehaving Detection procedure, thereby lifting the Control Overhead Ratio. However, this is inevitable in most on-demand misbehaving node detection approaches. With the *Initiate Detection Threshold* = 1.0 (or 0% of tolerable drop rate), we also observe that the Control Overhead Ratio is increased four times higher compare to the simulation results with the *Initiate Detection Threshold* ≥ 1.2. Again similar to False Accusation Ratio, more incidences of Detection procedure were invoked to due to the low threshold.
**Figure 55.** Control Overhead Ratio (C) with 20% Malicious Nodes.

### 4.4.4.5 End-to-End Delay

We report the increasing of end-to-end delay in Figure 56. With 20% of malicious nodes, we observe that the proposed scheme incurs minimum end-to-end delay. In most of cases, the length of delay increases approximately five milliseconds compared the reference schemes. This can due to the fact that other nodes can continue to forward data packet while one node is detecting a malicious node. Also, malicious nodes are unable to utilize the network resource once they are identified. Since we punish the misbehaving nodes by not forwarding their Location Discovery packet for a period of time, we did not include the communication sessions which the source nodes are misbehaving nodes. In addition, the *Initiate Detection Threshold* does not affect end-
to-end delay due the characteristics of the connectionless-oriented approach where anyone within the virtual router (i.e., CLA) or the neighboring node within the general direction of destination (i.e., CBF) can alternate in forwarding data toward the next virtual router or the next node.

![Graph showing end-to-end delay with 20% malicious nodes.]

Figure 56. End to End Delay (E) with 20% Malicious Nodes.

### 4.4.4.6 Active Detection Ratio

With speed of 20 m/s and 20% of malicious nodes, we observe that the number of nodes activated Detection Mode per malicious node’s location discover packet (that attempt to establish a connection) becomes fixed even the number of nodes in the network increased from 450 nodes to 1800 nodes (see Figure 57).
In fact, if a malicious node is stationary, the maximum number of neighboring nodes that are in the Detection Mode (i.e., Detecting-State) is six (see Figure 1 (a)). If a malicious node is moving at speed of 20 m/s, then the moving range (i.e., a circle with radius of \( r \)) within the maximum delay time (\( t = 2 \) seconds) of the Detection Mode is as follow:

\[
r = \text{speed} \times \text{time} = 20(\text{m/s}) \times 2(s) = 40(m)
\]

With radio range of a node is 250 meters; the radius of circular area of the maximum area of neighboring nodes that can activate Detection Mode is as follow:

\[
r_{\text{Detection}} = r + \text{radio range} = 40(m) + 250(m) = 290(m)
\]

Thus, the maximum number of neighboring nodes that are in the Detection Mode is seven nodes (see Figure 1 (b)). In order for a malicious node to move out of area where its neighboring nodes have activated the Detection Mode, the malicious node needs to travel of 540 meters (i.e.,
290 m + 250 m). With maximum moving speed of 20 m/s, the time a malicious node to move out of this area is 27 seconds (i.e., 540(m) / 20 (m/s)). Thus, the upper bond of Active Detection Ratio (A) is 7 nodes per 27 seconds (or 0.26 nodes per second). This confirms with our simulation study. In fact, the result in Figure 57 shows that our approach is able to adapt under high mobility (i.e., variety of applications – vehicular networks) and high density networks (i.e., scalable).

![Diagram of detecting nodes and malicious node](image1)

**Figure 58.** Number of detecting nodes needed per malicious node at different speed.

### 4.5 Discussion

In this chapter, we proposed an efficient 3CE (3-Counter Enforcement) scheme to enforce collaboration for the connectionless-oriented approach (i.e., CLA and CBF) in mobile ad hoc network. Our contributions are as follows. 1) We introduce an on-demand approach to misbehaving-node detection for the connectionless-oriented approach. Since the connectionless-oriented approach addresses highly dynamic networks (i.e., vehicle-to-vehicle networks), the
existing misbehaving-node detection techniques are not suitable. Our approach supports this type of routing protocol under high mobility environments. 2) Each node maintains three counters to represent its own status (i.e., reputation). Since nodes only determine their neighboring nodes’ counters information when a location discovery phase, no additional information is needed under a normal operation (i.e., nodes behave normally). 3) With large number of nodes and high mobility, the proposed approach enforces the cooperation on-demand with minimum increase of delay.

We conducted various experiments to study the effectiveness and efficiency of the proposed 3CE technique. The simulation results indicated that the proposed technique is very effective in enforcing collaboration. The degree of collaboration is significantly strengthened as the network throughput is greatly improved compare to a defenseless network. Such improvement is accomplished with almost no false accusation of cooperative nodes. As of efficiency, the proposed scheme incurs minimum delay.
Wireless mesh networks have become increasingly popular in recent years. Wireless mesh network (WMNs) are collection of mesh clients and mesh nodes (routers), with mesh nodes forming the backbone of the network and providing connection to the Internet and other network. Their rapid deployment and ease of maintenance are suitable for on-demand network such as disaster recovery, homeland security, convention centers, hard-to-wire buildings and unfriendly terrains.

The primary interests in mesh networks are extending the reach of exiting wired network and achieving high network capacity. To extend the coverage and capacity of the network, one can simply increase number of mesh nodes. However, the average cost of setup single mesh node is around $1,000 to $2,000. In order to setup a large scale mesh network such as Nortel’s Mobile City project in Taipei-Taiwan or SingTel Cisco Wireless Mesh Network in Singapore, its cost and time are unavoidable. Thus, we need an effective solution for the transition period from a partial to fully deployment of a mesh network.

Recently, in Mobile Ad Hoc Network (MANET), a new routing scheme, called Connectionless-Oriented Approach, has emerged. One of these types of scheme called Connectionless Approach (CLA) [21]. In CLA, the network area is divided into small non-overlapping grid cells (see Chapter 2). Instead of maintaining a hop-by-hop route between the source and destination node, the source selects a list of grid cells that form a “connecting” path between the source and destination. From a different perspective, each grid cell can be viewed as a virtual router in the sense that any physical router (i.e., a mobile node) currently within the
virtual router can alternate in forwarding data toward the next virtual router. The communication path consisting of consecutive virtual routers form a *virtual link* (see Chapter 2).

Given a virtual router, its physical routers compete to forward the data packets according to a data forwarding procedure. This function computes a shorter delay for a node farther from the sender and closer to the destination. In this environment, a virtual link is considered broken if one of its virtual routers becomes empty. This is addressed by replacing the empty virtual router with a neighboring virtual router. The fundamental advantages of CLA are twofold. First, a virtual link is much less likely to become broken than a standard route used in conventional connection-oriented techniques; and second, unlike standard routes, the robustness of virtual link is not sensitive to the mobility inherent in MANET.

In this chapter, we propose a paradigm that combines virtual routers and mesh nodes to create a hybrid network call VR-Mesh Network. This hybrid network can reduce number of mesh node needed without decrease the performance of the network. The primary contributions of this chapter are as follows:

- We introduce a hybrid network call VR-Mesh Network.
- We proposed a new routing protocol for the VR-Mesh network.

We present simulation results to show that VR-Mesh Network can maintain the similar performance of the network with fewer number of mesh nodes.

### 5.1 Related Work

Wireless mesh networks have become increasingly popular in recent years. Wireless mesh network (WMNs) are collection of mesh clients and mesh nodes (routers), with mesh nodes
forming the backbone of the network and providing connection to the Internet and other network. There are a large number of protocols and implementations in both wireless mesh, each with differing goals and design criteria. In the following we will briefly introduce a few of the most commonly see protocols in the wireless mesh network (WMN).

Topology Broadcast based on Reverse-Path Forwarding (TBRPF) [43] is a proactive, link-state routing protocol designed for mobile ad-hoc networks, which provides hop-by-hop routing along minimum hop paths to each destination node. TBRPF has two modes: Full Topology and Partial Topology. In Full Topology mode, each node is provided with the state of every link in the network. This mode is useful for spare topologies and when full topology information is needed. In Partial Topology, each node is provided with only enough information to compute min-hop paths to all other nodes. Currently, TBRPF protocol is used in the WIMENET routers produced by PacketHop Inc. and Firetide Inc.

Ad-hoc On-Demand Distance Vector Routing (AODV) [45] and Dynamic Source Routing (DSR) [26] are reactive, on-demand routing protocol where nodes do not need to maintain a routing table according to the current network topology. Instead, routes are created on demand and initiated by the source nodes. These schemes flood the network to discovery alternative routes to the destination. When a link break occurs along the route due to network topology changes, the source node can attempt to use any other route it happens to know about, or flood the network again to find a new route. This is referred to as route maintenance. Since route discovery and route maintenance are very expensive, these schemes are not suitable for applications with a frequent link breakage due to high mobility. Currently, AODV and DSR are used in Kiyon Inc.’s Autonomous Network and MSR’s WIMENET testbed, respectively.
Instead of choosing a static route ahead of time, Extremely Opportunistic Routing (ExOR) [4] defers the choice of the next forwarding node until the reception of the packet which is to be routed. The forwarding is done by the node closest to the destination, so the route is built dynamically. ExOR protocol is used by the MIT’s RoofNet Project.

In VRA environment, a virtual router is dynamically created and associated with a geographic location. Two nearby virtual routers are interconnected by multiple physical links referred together as a virtual link between the two routers. The physical link is provided by mobile devices currently in the proximity of the router (see Figure 59). In other words, these mobile devices take turn to help forward the data packets. A communication path between a source node and a destination node through intermediate virtual routers is called a route although it consists of virtual links, instead of physical links as in traditional route techniques. Since virtual routers do not move, this scheme is much less susceptible to link breaks.

![Virtual Router Approach](image)

**Figure 59.** Virtual Router Approach.
5.2 Hybrid of Virtual Router and Mesh Network

We consider a generic network consists of mobile nodes and mesh nodes with wireless communication capability (see Figure 60). Mobile nodes can either directly or relay by other mobile nodes to connect mesh nodes. Mesh nodes form the backbone of the wireless network and connect mobile nodes to the Internet through gateway routers.

![Figure 60. Hybrid Network.](image)

In a mesh network, mesh nodes are stationary. Thus, communication connections to mesh nodes are more reliable. However, connections between mobile nodes to mobile nodes suffer constant link breaks due to mobility of nodes. In Virtual Router Approach, those connections are established as virtual links or virtual route. A virtual link is represented by number of virtual routers instead of specific mobile nodes. A virtual router is dynamically
created and associated with a geographic location (see Figure 61). Using stationary virtual routers guarantees high data deliver rate even when nodes themselves have high mobility.

![Diagram](image-url)  
**Figure 61.** Virtual Link and Physical Link.

A virtual router is made up of one or more physical “routers” (or mobile nodes), and it may serve multiple communication sessions simultaneously. With Virtual Router Approach, we can replace some of mesh nodes with virtual routers to reduce number of mesh node in the network. This can reduce the cost of the mesh network and redirect the workload of overloaded mesh nodes. In this section, we introduce a new routing protocol for this hybrid network, called VR-Mesh Network, consists of mesh nodes and virtual router.

### 5.2.1 Route Request

In VR-Mesh Network, we use a route request procedure similar to the one in DSR although other routing techniques can also be applied. When a mobile node needs to establish a connection to the Internet through a mesh node, it first checks if it has a route established. If not, it initiates a
Route Request to any node that has direct connection to a mesh node (i.e., one-hop neighbor of a mesh node). We refer any node that has a direct connection to any mesh node as Virtual Destination (VR-Destination). Notice that a mobile node can be a one-hop neighbor of a mesh node but do not have direct connection to a mesh node due to traffic congestion in the area or workload of the mesh node.

An intermediate node receiving the request appends its ID to the packet and forwards it if this node has not seen the request before. Since mesh nodes are stationary, VR-Destination (i.e., a direct connection to a mesh node) does not need to forward Router Request further (i.e., to a mesh node) to establish communication connections between mesh nodes to mesh nodes. Different WMN routing protocols, discussed in Section 5.1, can be apply for establish a route within the mesh nodes. To avoid flooding the network with Route Request messages, a probabilistic delay technique is used.

5.2.2 Route Reply
Once a VR-Destination (i.e., a one-hop neighbor of mesh node) receives the Route Request, it can send a Route Reply to the source to establish the virtual routers and a route as follows. The VR-Destination will include its own ID, the node ID of the source, the list of nodes that the Route Request packet has traversed, and the ID of this route. The ID of a route is generated by the destination node consisting of its own ID concatenated with a locally generated unique number to ensure the uniqueness of the route ID in the network. The Route Reply packet is routed by the nodes listed in the packet, but any neighboring nodes can overhear the packet and join the node currently broadcasting the Route Reply packet to form a virtual router. A node ignores the Route Reply packet if this node has already joined a virtual router. When the Route
Reply packet eventually reaches the source, we have established a route consisting of the virtual routers. This process is illustrated in Figure 62. It will be clear later how these virtual routers different from mobile nodes serving as “routers” in traditional rerouting techniques.

Figure 62. Virtual routers and the route from source to VR-Destination.

The distance of a node to the VR-Destination is expressed in terms of hop count, i.e., number of hops to the destination node. We call this measure the Distance to Destination (DTD) in this paper. With this definition, the DTD of the relaying node is clear. As an example in 4, the DTD of the relaying node $r_3$ is 3. The DTDs of overhearing nodes are determined as follows. A relaying node of a Route Reply includes its DTD in the Hop Count field of the packet. A relaying node can compute its own DTD as the minimum of all the Hop Count fields it has overheard for a given Route Reply. As an example illustrated in 4, $n$ overhears a Route Reply broadcast by nodes that are 1, 2, and 3 hops away from the destination $d$. Thus, $n$’s DTD is 1.
This DTD tells $n$ whether it is farther from or closer to the destination compared to some other node $m$ in the route. We will explain in the next subsection how DTD is used in data forwarding.

### 5.2.3 Data Forwarding

To transmit a data packet, the source node includes the following information in the data header: *Source Node ID*, *VR-Destination Node ID*, *Packet ID*, *Route ID*, *Virtual Hop Count*, and *Traversed Nodes*. The *Source Node ID* and *VR-Destination Node ID* field refer to the node IDs of the source and VR-Destination, respectively. *Packet ID* refers the ID the source assigns to this data packet. *Route ID* refers to ID of the route this data packet will be forwarded along. *Virtual Hop Count* refers to the sender’s DTD; that is, every intermediate node will update this field before relaying the packet. The DTD is determined as explained in Section 5.2.2. *Traversed Nodes* refers to a list of nodes that relay this data packet; that is, every intermediate node appends its node ID to this list before relaying the packet. We will explain the purpose of this list in Section 5.2.4.

When a node $n$ receives a data packet from $m$, the data forwarding procedure is as follows:

1. If $n$ is the VR-Destination, $n$ forwards the data to a mesh node.
2. If $n$ has seen the data packet, $n$ does not forward the data.
3. If $n$ is not in the route, $n$ does not forward the data.
4. If $n$ does not belong to the same Virtual Router as $m$, $n$ does not forward the data.
5. If all previous steps fail (i.e. $n$ might need to forward the data), $n$ delays the forwarding.
6. During this delay period, $n$ will cancel the forwarding if $n$ hears the same packet again.
7. At the end of the delay period, if the forwarding decision has not been cancelled, \( n \) forwards the data.

In the Step (3 of the above procedure, \( n \) determines if it is part of the route by comparing the Route ID field of the data packet with all the Route ID’s it has captured from the overheard Route Reply packets.

In Step (4 of the procedure, \( n \) determines if it is part of the same virtual router by checking if its own DTD for this route is one less hop than the value indicated in the Virtual Hop Count field of the data packet. We permit only the nodes in the same Virtual Router to participate in the forwarding of a data packet from \( m \) in order to limit the number of competing nodes.

In Step (5 of the data forwarding procedure, node \( n \) sets its delay as \( \text{rand}(n \rightarrow seed) \times t \) seconds, where the function \( \text{rand}(n \rightarrow seed) \) computes a random number using a predetermined seed at node \( n \).

### 5.2.4 Route Update

Since nodes that receive a Route Reply can move away from their Virtual Router, the VR-Destination (i.e., any node within this virtual route) can recruit replacement nodes by periodically sending out an unsolicited Route Reply, called Route Update, to the source. The VR-Destination includes its own node ID, the node ID of the source, the list of nodes traversed by the latest data packet (i.e., nodes listed in the Traversed Nodes field) received by the VR-Destination from the source, and the ID of this route. The ID of a route is generated by the node within the virtual router of VR-Destination as discussed in Section 5.2.2. The nodes listed in the Route Update
packet relay the Route Update to the source node. Each of these nodes establishes a virtual router identified by the corresponding node ID, and these virtual routers define a new route between the source and VR-Destination.

5.3 Simulation Study

We conducted various experiments to verify the effectiveness of the proposed VR-Mesh Network scheme in enhancing performance of hybrid of mobile ad hoc and mesh network. In this section, we first introduce the simulation setup and parameters. We then study the proposed technique based on various performance metrics.

5.3.1 Schemes Implemented

We implemented two schemes, namely the reference scheme and the proposed VR-Mesh scheme for performance evaluation. In the reference scheme, only mesh nodes relay data to the base station (a randomly selected mesh node). To connect the base station, mobile nodes need to directly connect to a mesh node. Destination Sequenced Distance Vector (DSDV) routing protocol is used between mesh nodes and base station mesh node (i.e., internet gateway). Other routing protocols can be applied. In the proposed VR-Mesh scheme, mobile nodes are able to route data to VR-Destinations (mobile nodes directly connection to mesh nodes) using Virtual Router. A mobile node establishes a communication connection between itself and a mesh node as virtual link. The connection between a mesh node and base station mesh node is established using DSDV routing protocol.

5.3.2 Simulation Setup

All the experiments were conducted using GlomoSim [60]. This simulator, developed at UCLA, is a packet-level simulator specifically designed for ad-hoc networks. It follows the OSI 7-layer
network communication model. The field configuration is a 2500 by 2500 meter two dimensional space (see Figure 63). All nodes employ 802.11 at the MAC layer. Two types of node – mesh node and mobile node. Mesh nodes are stationary and placed in a square grid topology (see Figure 63). According to [47], triangular and square grid topologies result in significantly better coverage than hexagonal and random topologies. In the mesh topology, one randomly selected mesh node is the base station (BS) and others are relay stations (RSs).

![Figure 63. Network area with 25 mesh nodes.](image)
For mobile nodes, the random waypoint model was used to model the mobility of hosts. That is, each node randomly selects a destination point. When the node reaches this destination point, it pauses for a period of time, and then selects another destination point. We set up our simulation with 300 nodes, each moves at 20 meters per second (m/s) with zero pause time. Radio range is approximately 500 meter for mesh nodes and 250 meter for mobile nodes. Table 5 gives quick overview of the parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mesh Node</th>
<th>Mobile Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>25, 16, 9, and 4</td>
<td>300</td>
</tr>
<tr>
<td>Mobility</td>
<td>Stationary</td>
<td>20 m/s w/ zero pause time</td>
</tr>
<tr>
<td>Radio Range</td>
<td>500 meter</td>
<td>250 meter</td>
</tr>
<tr>
<td>Routing Protocol</td>
<td>DSDV</td>
<td>Virtual Routing approach</td>
</tr>
</tbody>
</table>

Multiple simulation runs (100 runs per setup on average) with different seed numbers were conducted for each scenario and collected data were averaged over those runs. The total simulation duration for each run was 10 minutes (600 seconds). Traffic applications are TCP session involving 1/10 of all mobile nodes or 15 sessions. Each data packet is 512 bytes. We reduce the number of mesh nodes in the network to verify the performance of replacing mesh nodes with virtual routers as follows (see Figure 64):

- 25 mesh nodes in 5 by 5 square grid topology.
- 16 mesh nodes in 4 by 4 square grid topology.
- 9 mesh nodes in 3 by 3 square grid topology.
- 4 mesh nodes in 2 by 2 square grid topology.
5.3.3 Performance Metrics

The routing protocols are compared according to the following three metrics which were suggested by the IETF MANET working group for routing protocol evaluation [11].

- **Fraction of Packet Delivered** – measures the ratio of the data packets delivered to the base station (BS) of mesh node and the data packets generated by the CBR source. This number indicates the effectiveness of a protocol.

- **End-To-End Delay** – measured in milliseconds, includes processing, route discover latency, queuing delays, retransmission delay at the MAC, and propagation and transmission times. This number measures the total delay time from a sender to the BS of mesh node.

- **Normalized Routing Load** – measures the number of routing packets transmitted per distinct data packet delivered to the BS mesh node. The routing overhead is an
important metric for proposed approach as it measures the scalability of a protocol, and its efficiency in terms of throughput and power consumption.

### 5.3.4 Experimental Results

We study the effect of decreasing the number of mesh nodes. We reduce the number of mesh nodes from 25 to 4. In Figure 65, the fraction of packet delivered of proposed VR-Mesh scheme decreases only 18% in average when we reduce 84% of mesh nodes in the network. Thus, the VR-Mesh scheme can maintain 80% of performance of the fully deployed mesh network using virtual routers with only 16% of mesh nodes.

![Figure 65. Effect on number of mesh node to fraction of packet delivered.](image)

In Figure 66, we compared the end-to-end delay of the VR-Mesh scheme with the Reference scheme. Our scheme reduced the end-to-end delay by 8 ms (26%). By comparing the fully deployed mesh network (i.e., 25 mesh nodes) with the VR-Mesh scheme with only 4 mesh
nodes, the end-to-end delay increased only 7 ms in average. This is due to source nodes are able to establish connections to mesh nodes using virtual routes and VR-Destinations.

In the Reference scheme, the end-to-end delay is also increase when we reduce number of mesh nodes in the network. The reason is that source nodes are unable to communicate directly to a mesh node. For a source node to establish a communication connection (i.e., directly connect to a mesh node), the source node needs to move within communication range of mesh node. Thus, this contributes to the decrease of the end-to-end delay performance of Reference scheme. Notice that this end-to-end delay does not include the data packets that are unsuccessfully delivered.

![Effect on Number of Communication Sessions](image)

**Figure 66.** Effect on number of mesh node to end-to-end delay.

In Figure 67, the VR-Mesh scheme has higher control overhead compare to the Reference scheme when the number of mesh nodes is less than 16. There are two reasons that the VR-Mesh scheme have higher normalized routing load. First, when the mesh nodes are not
fully deployed (i.e., less than 25 mesh nodes), source nodes need to establish and maintain the communication connection between mesh nodes. Second, when counting number control overhead, we do not include the unsuccessful delivered packets. Therefore, successful delivered packets for Reference scheme are usually within the communication range (i.e., one hop distance) of Mesh Nodes.

![Effect on Number of Communication Sessions](image)

**Figure 67.** Effect on number of mesh node to normalized routing load.

### 5.4 Discussion

In this chapter, we proposed a paradigm that combines virtual routers and mesh nodes to create a hybrid network called VR-Mesh Network. A virtual router is made up of one or more physical “router”, and it may serve multiple communication sessions simultaneously. Combining virtual routers, the proposed hybrid network is able to extend the coverage and capacity of the network. Our contributions are as follows. 1) We introduce a hybrid network called VR-Mesh Network. This hybrid network can achieve the same performance with fewer mesh nodes in the network. 2)
We proposed a new routing protocol for this hybrid network to reduce the number to route breaks. By reducing the number of route breaks, we improved the performance of successful packet delivery ratio and the end-to-end delay.

To assess the effectiveness of the proposed VR-Mesh network, we conducted various experiments to study the effectiveness and efficiency of the proposed hybrid network using GloMoSim. The simulation results indicated that the proposed technique achieve high fraction of packet delivery and low end-to-end delay with minimum increases in control overhead. Furthermore, the VR-Mesh network can reduce number of mesh nodes needed in the network without scarifies the performances.
6. CONCLUDING REMARKS AND FUTURE WORKS

6.1 Concluding Remarks
With the growing rate of mobile data market, wireless networks have become one of most popular network communication connection. One of variations of mobile wireless network called Mobile Ad Hoc Network (MANET). In a MANET, communication connections need to adapt to frequent unpredictable topology changes due to the mobility, energy constraints, and limited computing power of mobile devices.

In this dissertation, first, we propose a new approach called Connectionless Approach. In our approach, the network area is divided into small non-overlapping grid cells. Instead of maintaining a hop-by-hop route between the source and destination node, the source selects a list of grid cells that form a “connecting” path between the source and destination. From a different perspective, each grid cell can be viewed as a virtual router in the sense that any physical router (i.e., a mobile node) currently within the virtual router can alternate in forwarding data toward the next virtual router. The communication path consisting of consecutive virtual routers form a virtual link.

Given a virtual router, its physical routers compete to forward the data packets according to a data forwarding procedure. This function computes a shorter delay for a node farther from the sender and closer to the destination. In this environment, a virtual link is considered broken if one of its virtual routers becomes empty. This is addressed by replacing the empty virtual router with a neighboring virtual router. The fundamental advantages of CLA are twofold. First, a virtual link is much less likely to become broken than a standard route used in conventional connection-oriented techniques; and second, unlike standard routes, the robustness of virtual link
is not sensitive to the mobility inherent in MANET. Extensive simulation results have shown that our method is more robust, and performs significantly better than connection-oriented techniques.

In Chapter 3, we pointed out the performance of the virtual router approach suffers from packet drops since traffic congestion is not considered in the packet forwarding policy. A standard solution is to leave congestion control to the MAC layer. When serious congestion is confirmed, the source node is informed to search for another route. This simple approach incurs delay, computation overhead, and packet losses. These problems become more visible in traffic-intensive environments such as multimedia applications, where congestion is more probable and the negative impact of packet loss on the service quality is of more significance. Better solutions for congestion have been proposed (e.g., CADV, CRP, DLAR, extension-AODV, and extension-DSR). These schemes consider congestion in initial routing to avoid the aforementioned problems. Similar techniques are not available for connectionless-oriented MANETs. In fact, it is difficult to adapt the existing techniques since there is no hop-by-hop route in connectionless-oriented routing. In Chapter 3, we proposed a cross-layer design, called Dynamic Route Diversion (DRD), for connectionless-oriented MANETs. DRD improves the connectionless approach by taking into consideration traffic congestion in order to minimize packet drops. This is achieved by taking into account the workload of individual virtual routers, and dynamically reroute packets as necessary to prevent virtual router overloaded. The simulation results indicate that DRD offers significant performance gain by achieve high successful packet delivery rate and low end-to-end delay with minimum increases in control overhead. Furthermore, we observed the
workload more balanced and evenly distributed among the mobile hosts resulting in longer life spent for the network.

In Chapter 4, we address the security and cooperation issues for Virtual Router approach by proposed an efficient and on-demand 3CE (3-Counter Enforcement) scheme and to enforce collaboration. Many works have pointed out that the impact of malicious and selfish users must be carefully investigated. The selfish nodes are most concerned about their energy consumption and intentionally drop packets to save power. Since the Virtual Router approach addresses highly dynamic networks (i.e., vehicle-to-vehicle networks), the existing misbehaving-node detection techniques are not suitable. Our approach supports this type of routing protocol under high mobility environments. Each node maintains three counters to represent its own status (i.e., reputation). Since nodes only determine their neighboring nodes’ counters information when a location discovery phase, no additional information is needed under a normal operation (i.e., nodes behave normally). With large number of nodes and high mobility, the proposed approach enforces the cooperation on-demand with minimum increase of delay.

Wireless mesh networks have become increasingly popular in recent years. Wireless mesh network (WMNs) are collection of mesh clients and mesh nodes (routers), with mesh nodes forming the backbone of the network and providing connection to the Internet and other network. The primary interests in mesh networks are extending the reach of exiting wired network and achieving high network capacity. In Chapter 5, we propose a paradigm that combine virtual routers and mesh nodes to create a hybrid network. This hybrid network can reduce number of mesh node needed without decrease the performance of the network. We consider a generic network consist of mobile nodes mesh nodes with wireless communication capability. Mobile
nodes can either directly or relay by other mobile nodes to connect mesh nodes. Mesh nodes form the backbone of the wireless network and connect mobile nodes to the Internet through gateway routers.

In this dissertation, we proposed and investigate the new communication protocol for mobile ad hoc networks, vehicular networks, and mesh networks. The focus on these techniques is how to make the communication protocol to adapt and cooperate under highly dynamic network environment (e.g., high mobility and network congestion). The related works are discussed and explained their advantage and disadvantage. The extensive amount of experiments and simulations are provided to compare the proposed techniques with other existing ones. The results indicate that they outperform recent techniques by a significant margin and are suitable for highly dynamic ad hoc networks and mesh networks.

6.2 Future Works
My future research directions will carry on from the current academic accomplishments. Based on experience and knowledge, I will continue the research in the area of Mobile Ad Hoc Networks, Wireless Mesh Networks, and Intelligent Transport System. To provide a sound foundation for virtual router approach that can support large-scale network, diverse network environment, and different applications will remain my interest in the future. Some selected topics are as follows.

6.2.1 Security and Privacy in Wireless Mesh Network
Mesh networks are vulnerable to set of security challenges that include authentication, access control, and authorization, privacy and trust, encryption, key and identity management, DoS attacks, intrusion detection and prevention, and security policies. Traditional routing algorithm
cannot be directly applied to Mesh network, because they do not defend global attackers. One way to defend the attacks is to design a routing algorithm to address security and privacy issues. I would like to devise the problem with two foci: (1) design a routing protocol to defend against an attacker and to protect node privacy. (2) find ways for mesh nodes and users to collaborate to detect the attacks.

### 6.2.2 Wireless Vehicular Communication

The wireless vehicular communication can be typically identified as vehicle-to-person communication, vehicle-to-infrastructure communication, vehicle-to-vehicle communication, and vehicular communication networks. The combination of unique features of wireless vehicular communications and networking issues opens new opportunities for many interesting research areas, for example, real time safety applications, and intelligent diver information services.

Throughout the world, many national or international projects in government, industry, and academia have been devoted to the establishment of ambitious research programs, such as the European eSafety initiative, the German Ministry of Education and Research sponsored Wheels project, the US programs derived from the Intelligent Vehicle Initiative, and the Japanese InternetITS and AHS programs. In order for universal realization of wireless vehicular communications, many research challenges still need to be addressed to create good-performance, highly scalable, robust and secure vehicular technologies. It would be interesting to see if we could build a similar Virtual Router network for vehicular communication that include cooperation enforcement, route diversion, and privacy and security protection. I am interested in exploring those challenging problems.
7. REFERENCES


