WEIGHTED CRITICAL PATH ROUTING PROTOCOL FOR MOBILE AD HOC NETWORKS

Ihab El Kabary
Faculty of Computer and Information Sciences
Ain Shams University

Amal Elnahas
Faculty of Media Engineering and Technology
The German University in Cairo

Said Ghoniemy
Faculty of Computer and Information Sciences
Ain Shams University

Keywords: Ad-hoc networks, hybrid routing algorithms, mobile network routing protocol.

Abstract: Designing a routing protocol that can adapt to the changes in the underlying network conditions, as well as incorporating a minimum overhead is a challenging task for ad-hoc networks. In this paper, we present the Weighted Critical Path Routing (WCPR) protocol that strives to incorporate the merits of reactive and proactive ad hoc routing schemes. The aim of our work is to achieve low latency between highly active pairs of nodes, thus increasing the overall performance of the network without dramatically increasing the routing overhead. The genuine aspect of WCPR is that it initially starts-off as a conventional reactive Dynamic Source Routing (DSR) protocol. The network traffic is monitored in attempt to gradually discover pairs of highly interactive nodes in the network. Critical Paths are then constructed between these pairs of nodes and proactively safe guarded. The established CPs are treated differently depending on the amount of traffic consumed by each. WCPR is evaluated through simulation experiments and proved to outperform DSR in terms of delay with minimal increase in overhead.

1 INTRODUCTION

A mobile ad hoc network (MANET) is a self-configuring network composed of mobile nodes that operate without the need for any established infrastructure. Such form of networks is needed in many situations where no fixed communication infrastructure is available or where this fixed infrastructure is expensive to establish in terms of time or money constraints. Examples are battlefield applications, emergency relief operations, and others. In an ad-hoc network, nodes are free to move randomly and can act as both hosts and routers at the same time. Due to their limited transmission range, every node in the ad hoc network is not aware of the complete topology of the whole network and multiple hops maybe needed for one to exchange data with another node not in its direct transmission range.

The dynamic nature of MANET caused by continuously changing network topology and traffic pattern renders the design of a suitable routing protocol a challenge. Previously proposed routing protocols fall mainly into two divert categories based on their mode of operation: proactive protocols and reactive protocols. Proactive routing protocols, such as DSDV (C. Perkins, 1994), WRP (S. Murthy, 1996), CGSR (C. Chiang and M.Gerla, 1997) and OLSR (T. Clausen and Behrmann, 2001) exchange routing information periodically between nodes and maintain a set of available routes ready to be used by nodes at all time. On the other hand, reactive protocols such as DSR (Johnson and Maltz, 1996), AODV (Perkins and Royer, 1999), ABR (Toh, 1997), PLBR (R. Sisodia and Murthy, 2002), LAR (Ko and Vaidya, 1998) and DZALAR (Elnahas, 2005) attempt to perform a route discovery operation on demand when a specific route is needed. An obvious trade-off exists between the routing overhead and the delay in constructing a route when attempting to use proactive or reactive protocols. Proactive protocols can provide low latency and good reliability but at the same time are associated with high overhead specially with the increase of the

El Kabary I., Elnahas A. and Ghoniemy S. (2006).
WEIGHTED CRITICAL PATH ROUTING PROTOCOL FOR MOBILE AD HOC NETWORKS.
In *Proceedings of the International Conference on Wireless Information Networks and Systems*, pages 51-58

number of nodes in the network. The high overhead, measured in terms of the number of routing packets transmitted in the network, is caused by the need to periodically maintain all routes even if they are not needed, which affects the most valuable resource in a MANET, the bandwidth. On the other hand, reactive protocols efficiently make use of bandwidth as they delay the route discovery mechanism until a route is requested thus dramatically decreasing the routing overhead but enduring an apparent increase in latency.

Hybrid adaptive routing protocols have been suggested in an attempt to balance the overhead and the adaptability to network conditions by implementing both proactive and reactive protocols in different regions or at different times in the same network. A variety of hybrid protocols, such as ZRP (Haas, 1997), CEDAR (P. Sinha and Bharghavan, 1999), ZHLS (Joa-Ng and Lu, 1999) and SHARP (V. Ramasubramanian, 2003), have been suggested combining proactive and reactive routing mechanisms in various ways.

In an earlier work presented in (I. kabary, 2006), we proposed a hybrid routing protocol that attempts to incorporate the merits of both proactive and reactive schemes taking a totally different approach by focusing on locality of calls in order to achieve low latency. The idea of our protocol is to initially starts-off as a conventional reactive routing protocol like the DSR (Johnson and Maltz, 1996) and then attempt to monitor the network traffic patterns in order to discover pairs of nodes that exchange information more often than others. The routes between these pairs of highly active nodes (called critical paths) are proactively safe-guarded to ensure minimum routing delay . In this way, routes that are frequently used are maintained, while other routes between low active nodes, are created on demand. This approach ensures the efficient use of scarce bandwidth and at the same time decreases the latency between highly active pairs of nodes and between the intermediate nodes found within the critical route paths. Achieving low latency between pairs of highly active nodes does not come without a price. Safeguarding the critical paths causes an increased overhead as pointed in (I. kabary, 2006).

In this paper, we present the WCPR protocol that attempts to decrease the overhead entailed by safeguarding the critical paths. The idea of WCPR is to treat different critical paths differently depending upon their criticality. Thus, how frequent a certain path is safeguarded depends on how critical it is. This criticality is measured depending on the amount of traffic consumed by each path.

The rest of this paper is organized as follows. The next section sheds some light on related work in hybrid and adaptive routing protocols. The description of WCPR is presented in section 3. In section 4 we perform a thorough investigation on the performance of WCPR and finally the summary of our contribution and the future work are presented in section 5.

2 RELATED WORK

A variety of hybrid ad hoc routing protocols have been developed like ZRP (Haas, 1997), CEDAR (P. Sinha and Bharghavan, 1999), ZHLS (Joa-Ng and Lu, 1999) and SHARP (V. Ramasubramanian, 2003). Each protocol exploits the benefit of proactive and reactive shemes in different ways.

The ZRP (Zone routing Protocol) is one of the first known hybrid routing protocols based on defining a zone around each mobile node consisting of its k-neighbors. A proactive routing protocol is used to perform routing within the zone while on-demand reactive routing is used between nodes in different zones. The proactive routing protocol is used to provide each node with a view of its surrounding routing zone topology. On the other hand, global route discovery is initiated through a process called bordercasting. Bordercasting allows a node to send packets to its peripheral nodes only (nodes lying on the boundary of the route zone) and preventing other nodes accessing the packet. So route discovery is efficiently established via bordercasting a route request to the entire source node's peripheral nodes, which in turn bordercast the request to their peripheral nodes and so on if the destination is not within their respective routing zones. Once the destination is discovered in one of the zones, a route reply is echoed back to the source in the form of a reversed list of peripheral nodes between the source and destination that the route request passed through. In this way, ZRP focuses on decreasing the route discovery overhead.

CEDAR (Core Extraction Distributed Ad hoc Routing) is a robust QoS routing protocol that is built on the idea of dynamically electing a set of distributed nodes which form the core of the MANET. This is done by approximating a minimum dominating set of the MANET. Each core host maintains the local topology of hosts in its domain and performs route computation on behalf of these hosts. Then QoS routing is achieved by propagating the bandwidth availability information throughout the core nodes. When a path is requested between two nodes, a shortest widest path (a path with maximum bandwidth) is calculated using information gained by these core nodes.

In the ZHLS (Zone-based Hierarchical Link State) routing protocol, at design time, the network is divided into non-overlapping zones. Initially, each node knows its position and therefore its zone ID through Global Positioning System (GPS) by mapping its physical location to the zone map. Then, each node only knows the node connectivity with its zone and the zone connectivity of the entire MANET. When a node needs to send data to a specific destination, the source needs to search for the zone ID of the destination node before any data can be transmitted. First the source node checks the intrazone routing table, if the destination is found then it is within the same zone. If not, a location request is sent to every other zone until the zone ID of the destination is identified. In this way, overhead can be decreased dramatically.

On the other hand, SHARP attempts to automatically find the balance point between proactive and reactive routing. This is done by adjusting the degree to which route information is propagated proactively versus the degree to which it needs to be discovered reactively. SHARP is distinguished in a way that it enables each node to use a different application-specific performance metric to control the adaptation of the routing layer, not just focusing on decreasing route overhead.

All of the previous hybrid routing protocols have not specifically focused on the issue that usually mobile nodes interact and send data to a relatively small number of nodes when compared to the total number of nodes in the MANET and usually from this small set of nodes lies an even smaller set of nodes that receives data at a relatively high rate (we call them hot destinations). The WCPR protocol focuses on this observation and attempts to identify and proactively maintain routes between each node and its hot destinations while using an on-demand reactive protocol when attempting to communicate with the rest of the nodes in the network. By this way WCPR is genuine when compared to other hybrid routing protocols as it strives to achieve call locality and low latency between pairs of highly interactive nodes.

3 WIGHTED CRITICAL PATH ROUTING PROTOCOL

The proposed WCPR protocol is composed of five main modules, as explained in this section in details. An overview of the protocol is presented followed by a description of each of the modules.

3.1 Protocol Overview

Our proposed protocol is based on the earlier version published in (I. kabary, 2006), where highly active nodes are identified and critical paths (CPs) are constructed and maintained between those nodes. CPs are periodically maintained by checking the validity of those paths every inspection interval. All CPs are treated alike and inspection interval value is the same for all paths.

Table 1: Critical path categories.

Category	Outgoing Traffic %	Inspection Interval
Category 1	25%	12s
Category 2	50%	6s
Category 3	75%	3

In this work, in attempt to decrease the overhead associated with maintaining critical paths, we propose treating critical paths emerging from the same source node (SRC) differently depending upon their criticality. This criticality is measured based on the amount of traffic consumed by each CP. The higher the traffic consumed through a CP, the higher its criticality becomes and the more attention will be given to this CP. Hence, inspection interval value varies according to the degree of criticality. By increasing the inspection interval for some paths, overhead is reduced.

In Table 1 , critical paths are categorized into three different categories, representing three different levels of CP criticality. The number of categories is a tuning parameter that can be set according to the different traffic types and requirements.

The outgoing traffic from a source to a destination is measured as an activity ratio calculated as follows:

$$ActivityRatio_{SRC}(DST) = \frac{PktSent(DST)}{\sum_{i=1}^{d} PktSent(i)} \quad (1)$$

where d is the number of destination nodes for a certain source SRC.

3.2 Protocol Design

The WCPR is composed of the following five components:

- CPDA: Critical Path Detection Algorithm
- CPCA: Critical Path Construction Algorithm
- CPIA: Critical Path Inspection Algorithm
- CPRA: Critical Path Re-construction Algorithm
- CPBA: Critical Path Break-up Algorithm

The CPDA attempts to discover pairs of highly active nodes and calls on the CPCA to establish a critical path between those nodes. The CPIA proactively safe guards the previously established CPs and checks that the CP is always valid with no broken links. If the CPIA detects an invalid CP, it calls on the CPRA to immediately re-establish an alternative CP having the same source and destination. Finally CPBA decides when a CP is no longer distinguished and its no longer a benefit to keep the CPIA proactively safe guarding it. The next ssubsections explain the functionality of the five algorithms in more details.

3.2.1 Critical Path Detection Algorithm (CPDA)

The CPDA is responsible for detecting destination nodes, called hot destinations, receiving a relatively high rate of traffic from data transmissions, according to the categories described in Table 1. Information about data exchange between each node and all its destinations is accumulated in the HotDestinationTable, shown in Table 2, stored at each source node.

Table 2: Hot Destination Table.

Destination	PacketsSent	ActivityRatio

Initially, the HotDestinationTable of each node is empty. When the source node SRC attempts to send packets to a destination node DST, an entry is made in the HotDestinationTable of the SRC node with DST written in the Destination field and the PacketsSent (initially set to 0) is incremented by the number of packets sent to that specific destination DST. Every time the SRC node sends packets to any destination, the PacketsSent field of that specific destination is incremented in the HotDestinationTable of the SRC. The degree of interaction between the SRC node and a specific destination is set in the ActivityRatio field. Depending on the value of the ActivityRatio (AR), the category of the CP is determined and saved.

In order to dynamically adapt to the changes in the traffic, a CP category may change with time according to the outgoing traffic patterns of the SRC nodes. Accordingly, CPDA must be able to upgrade/downgrade the category of an already established CP. This can be seen in lines 23 to 33 of Figure 1. The upgrade/downgrade mechanism works by creating a packet that contains the new category of the CP, this packet is forwarded to the DST node of the CP, which in turn updates the InspectionInterval according to the new category as Table 1 implies.

Checking the CPs is only done every β packets sent by the source node in order to decrease the overhead associated with creating, maintaining or breaking up a critical path.

```
1.OnNodeTransmission(pkt)
2.{
3. DST = GetPcktDestn(pkt)
4. if(HotDestTable.Contains(DST)==true)
5.   IncrementHotDestEntry(DST)
6. else
7.   AddHotDestEntry(DST)
8.   totalPkts=0
9. foreach HotDestEntry e in HotDestTable
10.{  totalPkts =
       totalPkts + e.pktsSent }
11.pktsSent = GetPacketsSentBy(DST)
12.path = RetrievePathFromCache(SRC,DST)
13.AR = pktsSent/totalPkts;
14.if (pktsSent % beta == 0)
15.{
16.   // Creating new CPs
17.if (AR>0.75 AND not IsCP(path))
18.    CPCA (SRC,DST,Category3)
19.if (AR>0.5 AND not IsCP(path))
10.    CPCA (SRC,DST,Category2)
21.if (AR>0.25 AND not IsCP(path))
22.    CPCA (SRC,DST,Category1)
23.  // Upgrading CPs
24. else if(AR>=0.75 AND
25.      (path.Cat==1 OR path.Cat==2))
26.     UpgradeCP(SRC,DST,3)
27.  else
28.   if(AR>=0.5 AND (path.Cat==1))
29.     UpgradeCP(SRC,DST,2)
30.    // Downgrading CPs
31.   else if(AR<0.75 AND
32.      (AR>=0.5 AND path.Cat==3))
33.       DowngradeCP(SRC,DST,2)
34.     else if(AR<0.5 AND
35.       (AR>=0.25 AND path.Cat==2))
33.       DowngradeCP(SRC,DST,1)
34.    // Decompose CP
35.      else if(AR<0.25 AND IsCP(path)
36.        CPBA(SRC,DST)
37.}}
```

Figure 1: CPDA algorithm.

3.2.2 Critical Path Construction Algorithm (CPCA)

The CPCA algorithm, described in Figure 2, is simple yet crucial to the overall protocol. It is concerned with the creation of the CP between a pair of nodes that have already been identified by the CPDA as a pair of highly active nodes. Once the CPCA is called upon by the CPDA with parameters SRC and DST; CPCA marks the route that contains the SRC and DST as source and destination nodes in the cache entries of the DSR as a CP, along with its category. Not only that, but the CPCA creates a CriticalPathAdvertisement packet containing the complete CP and sends this packet to the destination node to add it to its data structure found at each node called CriticalPathTable, that enlists all CPs that this node is part of. The CriticalPathTable is a vital data structure and its role will be evident in the CPIA explained in the next subsection.

3.2.3 Critical Path Inspection Algorithm (CPIA)

CPIA module is in charge of monitoring the CPs and periodically assuring that they are ready to be used instantly by the SRC nodes. The OnTimerElapsed function will be recalled for each CP according to

```
1.CPCA (SRC, DST, CATEGORY)
2.{
3. MarkAsCrtclPath(SRC,DST,CATEGORY
4. CPAdvPacket = CreateCrtclPathAdvPckt()
5. ForwrdToDSTNode(CPAdvPacket)
6.}
```

Figure 2: CPCA.

the category of this CP. The OnTimerElapsed function is attached to each CP and triggered on different time intervals depending on the category of the CP as described in Table 1. This function is responsible for sending CPValidation packets from the DST to the SRC informing the SRC that the CP is still intact. On the other hand, if the SRC does not receive a CPValidation packet within a specific time interval, it will mean that the CP is broken and an alternative CP should be established. The CPIA is shown in Figure 3 and Figure 4 . It is split into two parts, one used by the DST nodes and the other used by the SRC nodes.

```
1.CP.OnTimerEllapsed()
2.{
3. CP.DST.SndValdtnPckt(CP.SRC)
4. ReschdleTimer(CP.Inspection_Interval)
5.}
```

Figure 3: CPIA (DST).

```
1.CP.OnTimerEllapsed()
2.{
3. if(CP.CheckForValdtnPckt()==false)
4.   CPRA(CP.SRC, CP.DST)
5. CP.ValidtnReset()
6.}
```

Figure 4: CPIA (SRC).

It is worth mentioning that line 4 of CPIA in Figure 3 is responsible for rescheduling the next time OnTimerElapsed will be called, since this Inspection-Interval could be changed by the CPDA due to the upgrade/downgrade mechanism of CPs as explained before. Also, the OnTimeElapsed is triggered at the SRC node with a small lag in time to give chance for the CPValidation packet to be received from the DST.

3.2.4 Critical Path Re-construction Algorithm (CPRA)

The CPRA, in Figure 5, will be initiated by CPIA when a CP has been broken. CPRA is responsible for rediscovering an alternative path and when found will call upon CPCA giving the new CP the same category as the broken CP it replaced.

```
1.CPRA(SRC, DST, CATEGORY)
2.{
3. DSRRouteRedscvry(SRC,DST)
4. CPCA(SRC,DST,CATEGORY)
5.}
```

Figure 5: CPRA.

3.2.5 Critical Path Break-up Algorithm (CPBA)

CPBA is called upon by CPDA when the proactive maintenance of a CP made by the CPIA is no longer beneficial, as the use of the CP has deteriorated, and the overhead made in keeping the CP valid is useless. The CP should return to be a normal path once more as shown in Figure 6.

```
1.CPBA(SRC,DST)
2.{
3. UnmarkPath(SRC,DST)
4. RemoveCP(p)
5.}
```

Figure 6: CPBA.

4 PERFORMANCE EVALUATION

Simulation experiments have been conducted in order to evaluate our proposed protocol. This section details the simulation environment, as well as presenting the analysis of the results obtained.

4.1 Simulation Environment

Our simulation model considers a 1000x1000 m square area. Fifty nodes are involved in the simulation, each has a wireless transmission range of 250 meters. Each simulation runs for 300 seconds. Mobile nodes move within the simulation area according to the Random Waypoint (RWP) mobility model, where each node randomly chooses a point to move to at a randomly selected velocity. The node, then, pauses for a certain pause time before repeating the same pattern again. All nodes have a uniform speed distributed between 0 and 10 m/s. The pausetime, which reflects the node ś degree of mobility, ranges from 0 to 300 seconds. When pausetime is 0 seconds, it means that all nodes are in continuous motion and the ad hoc network is in a high degree of mobility. When pausetime is 300 seconds, it means that all nodes are stationary throughout the simulation. From the 50 nodes, 8 nodes engage in initiating constant-bit-rate (CBR) connections with bit rates of

Table 3: Simulation Parameters.

Simulation Parameter	Value
Simulation time	300s
Simulation area	$1000x1000m^2$
Number of mobile nodes	50
Transmission range	250 m
Radio Propagation model	Two-Ray ground
Antenna Type	Omni-directional
Mobility model	RWP
Speed	Between (0, 10) m/s
Pause time	0,50,100,150,200,250,300 s
CBR sources	8
Packet rate	2 packets/sec
Packet size	512 Bytes
β	20 packets

two packets per second, as in (V. Ramasubramanian, 2003). The complete set of simulation parameters are listed in Table 3.

In order to effectively test the performance of WCPR protocol we created a generic traffic pattern scheme for each CBR source node (SRC) in which the node transmits data to 6 different destination nodes (DST) at different times with variable transmission lengths. This will cause each CBR SRC node to have several CPs with variant categories. These paths will inevitably have their categories changed (upgraded/downgraded) as the simulation runs. Examples of such different traffic patterns generated by 8 different source nodes (nodes 1 till 8) are shown in Figure 10, where destination nodes were chosen randomly.

4.2 Simulation Results

The performance of the WCPR protocol is compared to that of the DSR (Johnson and Maltz, 1996) and to the Critical Path Routing (CPR) protocol (I. kabary, 2006). The DSR were chosen for comparison as WCPR is considered a modification added to the basic DSR protocol. Three performance metrics were considered, namely: average end-to-end delay, packet delivery ratio, and route overhead.

Results in Figure 7 and 8 show that the performance of WCPR outperforms DSR and is nearly identical to that of CPR in terms of packet delivery ratio and average end-to-end delay. In various pause-times CPR merely had a 1% advantage in both these metrics due to the fact that in WCPR the lowest InspectionInterval given to the most critical CPs (3 seconds) was used with all CPs of CPR. The main advantage of WCPR is in decreasing the routing overhead. WCPR was accompanied with less overhead throughout the various pause-times when compared to that of CPR by 18% when pause-times ranged between 0 and 150 seconds as shown in Figure 9.

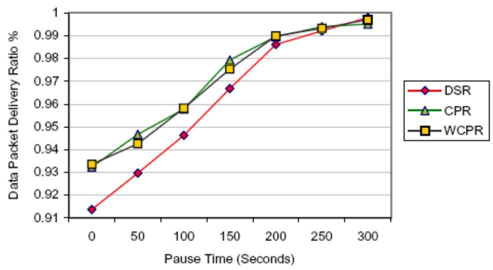

Figure 7: Packet Delivery Ratio.

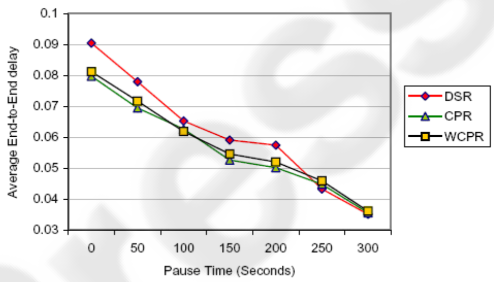

Figure 8: Average end-to-end delay.

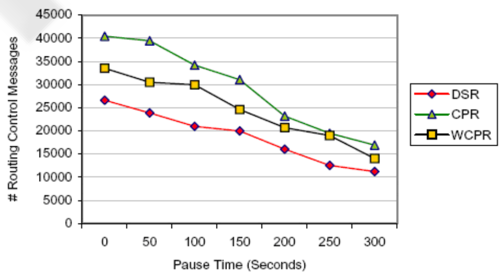

Figure 9: Routing overhead.

5 CONCLUSION

This paper presents the design, implementation and evaluation of the adaptive WCPR protocol which attempts to benefit from the merits of both the reactive DSR protocol, which saves the bandwidth, and the proactive scheme which results in lower latency.

WCPRś genuine aspect is that it focuses on achieving low latency between pairs of highly active nodes in the MANET.

The proposed protocol is an evolution of the CPR with the focus on further decreasing the control overhead entailed in the operation of the CPR protocol. The idea of WCPR is that it treats different CPs emerging from the same source node differently depending upon their criticality. This criticality is measured depending on the amount of traffic consumed by each CP. In other words, the higher the traffic consumed through a CP, the higher its criticality becomes and the more attention will be given to this CP. Simulation results showed that our efforts paid off and routing overhead in WCPR was decreased by 18.3% in comparison to CPR at relatively high degrees of mobility (ranging from 0 to 150 seconds).

Future work will focus on attempting to establish Quality of Service (QoS) measurements in the CPs created by the protocol. Clearest example will be to allow the CP to satisfy certain levels of bandwidth requirements. Also monitoring the effect of giving higher priorities to packets that are being sent through CPs seems to be very interesting. We will also focus on trying to avoid creating CPs that incorporate nodes that are constrained and have low battery life times remaining in attempt to increase the life time of these nodes. Finally we will focus on comparing WCPR with other protocols, specifically hybrid routing protocols.

REFERENCES

C. Chiang, H. Wu, W. L. and M.Gerla (1997). Routing in clustered multi-hop mobile wireless networks with fading channel. In *IEEE Singapore International Conference on Networks, SICON, Singapore*.

C. Perkins, P. B. (1994). Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *ACM SIGCOMM, London, UK*.

Elnahas, Ghoniemy, K. E.-K. (2005). Dead-zone avoidance algorithm for location based routing protocols. In *The 2nd International Conference on Computers and Information Systems, Cairo, Egypt*.

Haas, Z. (1997). The routing algorithm for the reconfigurable wireless networks. In *International Conference on Universal Personal Communications, ICUPC, San Diago, USA*.

I. kabary, A. Elnahas, S. G. (2006). Critical path routing protocol for ad-hoc networks. In *IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN), Innsbruck, Austria*.

Joa-Ng, M. and Lu, I. (1999). A peer-to-peer zone based two-level link state routing for mobile ad hoc networks. In *IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, pp. 1415-1425*.

Johnson, D. and Maltz, D. (1996). Dynamic source routing in ad hoc wireless networks. In *Mobile Computing, Kluwer Academic Publishers, pp 153-181*.

Ko, Y. and Vaidya, N. (1998). Location-aided routing (lar) in mobile ad hoc networks. In *ACM MOBICOM, Dallas, Texas*.

P. Sinha, R. S. and Bharghavan, V. (1999). Cedar: A core extraction distributed ad hoc routing algorithm,. In *IEEE Journal on Selected Areas in Communications, vol. 17,no. 8, pp. 1454*.

Perkins, C. and Royer, E. (1999). Ad hoc on-demand distance vector routing. In *IEEE Workshop on mobile Computing Systems and Applications*.

R. Sisodia, B. M. and Murthy, C. (2002). A preferred linked-based routing protocol for ad hoc wireless networks. In *Journal of Communications and Networks, vol. 4, no. 1, pp.14-21*.

S. Murthy, J. J. G.-L.-A. (1996). An efficient routing protocol for wireless networks. In *ACM Mobile Networks and Applications Journal, Special Issue on Routing in Mobile Communication vol. 1, no. 2, pp. 183-197*.

T. Clausen, G. Hansen, L. C. and Behrmann, G. (2001). The optimized link state routing protocol, evaluation through experiments and simulations. In *IEEE Symposium on Wireless Personal Mobile Communication*.

Toh, C. (1997). Associativity-based routing for ad hoc mobile networks. In *Wireless Personal Communications, vol. 4, no. 2*.

V. Ramasubramanian, Z. Haas, E. S. (2003). Sharp: A hybrid adaptive routing protocol for mobile ad hoc networks. In *ACM Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc), Annapolis, Maryland*.

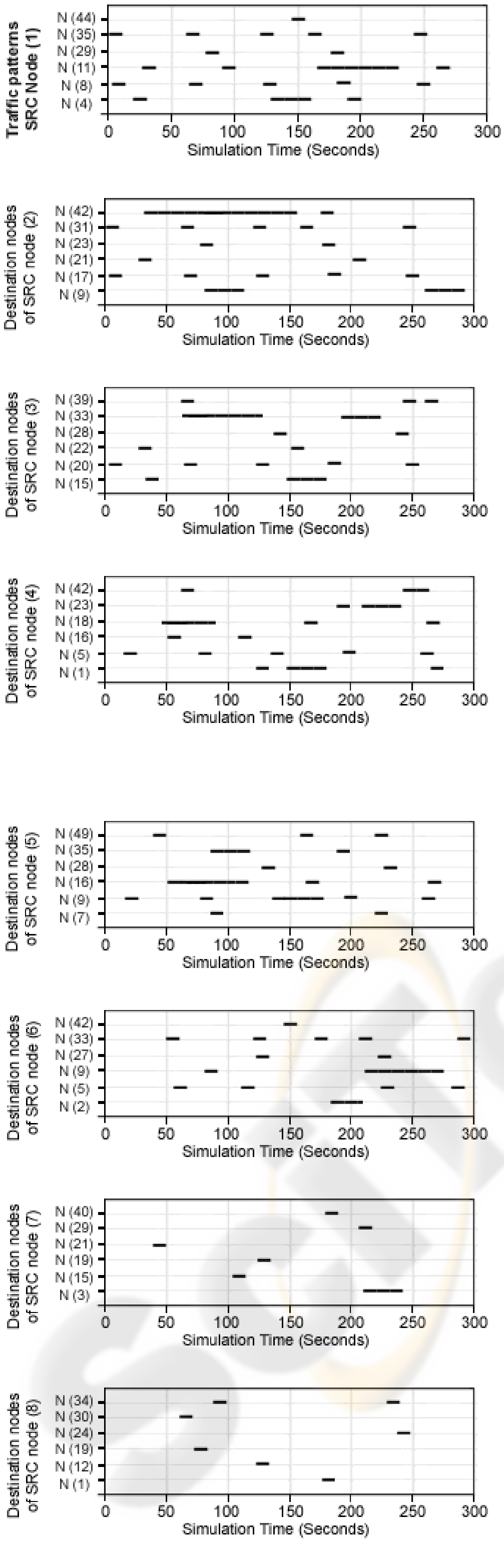

Figure 10: Traffic patterns for 8 sources.