

# Topology Control with Two-Hop Forest Construction in Ad Hoc Networks

Rattana Chawengmahapeeti

MS TS Program, Faculty of Science and Technology, Assumption University  
Bangkok, Thailand  
<rattana\_cha@yahoo.com>

## Abstract

*The emerging implementations of ad hoc networks depend on multiple technological factors which are a subject of intensive studies. Due to the limited processing capabilities of the network nodes, limited bandwidth, as well as power constraints, the analytical description of the routing process in multihop ad hoc networks poses significant challenges. The topology control deals with the increased routing complexity by reducing the average node degree in the network. A two-hop forest construction for topology control is proposed and studied in this paper. The quality of service in terms of latency and throughput is considered for the case of unicast routing on the basis of computational experiments.*

**Keywords:** Topology control, two-hop forest, ad hoc network.

## Introduction

The ad hoc networks represent sets of nodes that exchange information on a multihop basis. Due to the decentralized nature of the routing process, routing decisions must be made locally which increases the risk of the occurrence of congestion at some parts of the network and underutilization of the remaining parts. The prospective ad hoc networks should implement the all-port communication model where all input and output ports of a network node can be used simultaneously for the exchange of information among connected first neighbors. The increased amount of incoming traffic at the input ports with possible relaying through a single output port results in congestion due to the limited channel bandwidth. As greater is the number of input/output ports, as greater is the traffic variability. An additional constraint is the overall power consumption of the multiple ports. When dealing with the energy efficiency, most studies are concerned with the transmission range rather than with the number of ports. However, an alternative approach is to control the number of active ports so that a full

connectivity is still preserved and at the same time the congestion is reduced using appropriate topology control techniques. The node degree distribution plays a significant role in this process. The average node degree is an indicator of the typical number of ports being used per node in a network. In planar networks, typical values of the average value vary within the range from 3 to 5. It is beneficial to study computationally the performance of ad hoc networks for various connectivity schemes as the analytical consideration of this problem is far from completion at present. The hybrid routing techniques combine the topology control with routing to deal with the processing limitations of the nodes and with the noise in the wireless channels. For compatibility with the existing hardware solutions, a single first-in first-out (FIFO) queue and a single routing server per node are supposed to handle the incoming/outgoing traffic from/to multiple input/output ports. For example, within the existing IEEE 802.11x standards (IEEE 2007), only one port per node can be served at a time. The future extensions to real multi-port networks depend on the cost of implementation of nodes having a prescribed number of active

ports which satisfy the electromagnetic compatibility (EMC) standards.

The problem statement of this paper is to study and compare the performance of ad hoc networks with and without topology control in order to contribute to the development of an efficient hybrid routing protocol. Topology control algorithms are targeted for the construction of non-overlapping clusters (zones) which can be interconnected through gateway nodes. An emphasis is put on the use of a limited node degree that could prevent abrupt failures of more frequently used nodes. Due to the lack of statistical regularity of randomly distributed ad hoc networks, the simulation studies do not always reveal certain characteristics of the impact of the topology to the routing process. One can use simplifying assumptions for the initiation of computational experiments with emphasis on the topology control and its effect to the routing process. The development of a semi-analytical model and its software implementation for testing routing over shortest paths in arbitrary network configurations is reported in this study.

### Analytic Model

#### Unicast Routing

Unicast routing model is assumed where a source node can be connected to only one destination node at a time. This is a reasonable assumption that can simplify the analytical model. In particular, one can be interested in the worst case traffic flow that can occur in an edge between arbitrary relaying nodes  $i$  and  $j$  with  $i, j = 1, 2, \dots, N$ , where  $N$  is the number of nodes in the network. Assuming also a single multihop path between arbitrary source  $s$  and destination  $d$  nodes, let  $N_s$  be the number of source nodes that use the edge  $i \rightarrow j$  for routing. Also, let  $N_d$  be the number of possible destinations after traversing the edge  $i \rightarrow j$ . In unicast routing mode, the worst case traffic flow is related to the maximum number  $N_{s,d}$  of paths that may simultaneously attempt to use the said edge, where

$$N_{s,d} = \min(N_s, N_d). \tag{1}$$

Having a knowledge about the frequency of use of an edge (number of paths) and the average arrival rate per node,  $\lambda$ , it is possible to estimate:

- the traffic flow through an arbitrary edge  $i \rightarrow j$ . This results in the collection of overlapping flows that can be viewed in a three-dimensional (3D) histogram; and
- the total arrival rate,  $\lambda_{TOTAL}$ , at the queues of relaying nodes. This results in the collection of rates that can be viewed in a two-dimensional (2D) histogram.

#### Connectivity Matrix

The frequency of use of the links depends on the underlying topology. The simplest way to describe the topology in a computational experiment is based on the creation of a binary connectivity matrix, **cm**, where one can keep separately in the memory both the connectivity matrix of the plain scheme without topology control and the connectivity matrix of the resultant topology. The rows,  $i$ , and columns,  $j$ , of the connectivity matrix represent arbitrary directed edges  $i \rightarrow j$  for one-hop transmissions among first neighbors as shown schematically in Fig. 1.

$$\mathbf{cm}[i][j] = \begin{matrix} \text{Columns, } j \\ \begin{bmatrix} 0 & 1 & \dots & 0 \\ 1 & 0 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 0 & 1 & \dots & 0 \end{bmatrix} \\ \text{Rows, } i \end{matrix}$$

Fig. 1. A schematic view of a connectivity matrix.

The process of topology control can be viewed from algorithmic point of view as a black box with an input connectivity matrix of the plain scheme with no topology control and with an output containing the resultant topology as shown schematically in Fig. 2. The content of the black box can be modified to produce various different topologies with different network diameters and node degree distributions.

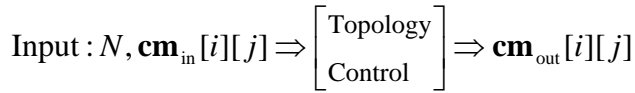


Fig. 2. A schematic view of a topology control algorithm.

**Node Design**

The nodes in ad hoc networks should be capable of receiving and transmitting information simultaneously in the form of packets of variable length from/to multiple input/output ports. The average waiting time depends on both the inter-arrival time distribution and the packet length distribution. The individual average arrival rates,  $\lambda_i$ , must satisfy the inequality  $\lambda_i \leq R$ , where  $R$  is the average transmission rate per port. The total average arrival rate,  $\lambda_{TOTAL}$ , is limited by the relaying capabilities of a node:

$$\lambda_{TOTAL} \leq R * (\text{Number of Ports}), \quad (2)$$

where  $\lambda_{TOTAL} = \lambda_1 + \lambda_2 + \dots$ , is the total arrival rate at the input of a node. It is based on the sum of multiple packet flows arriving at the distinct input ports of the node.

**Node Delay**

The average latency,  $T$ , can be given as a dimensionless ratio in per cent with respect to a reference flow,  $\lambda_{Reference}$ , related to the flow generated by a single source for which the average latency,  $T_{Reference}$ , is assumed to be known in advance. For simplicity, M/M/1 queuing (Kleinrock 1976) in an open network is assumed. Then the normalized average latency per node is given by the expression:

$$T_{\%} = (T / T_{Reference}) \times 100\% \quad (3)$$

The average end-to-end latency per hop in a multihop path is obtained accordingly:

$$T_{\text{End-to-End, \%}} = \frac{\sum_{k=1}^{\text{Hops}} T_k}{\text{Hops} * T_{\text{Reference}}} \times 100\%, \quad (4)$$

where  $T_i, i = 1, 2, \dots$ , are the average latencies per node. For simplicity, it is assumed that the packet servicing capabilities of the nodes and

their output ports are equivalent. Then the normalized delays will show as to how many times the said delays are greater than the reference delay.

**Throughput Calculation**

The throughput is determined assuming that the traffic flow,  $\lambda$ , generated by a single source is the same for all active nodes. Then the number of overlapping paths through an edge  $i \rightarrow j$  can be obtained as schematically shown in Fig. 3.

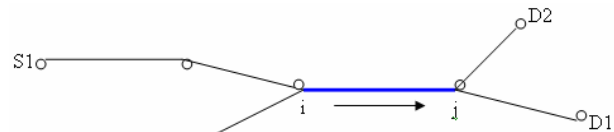


Fig. 3. A sample path overlapping on edge  $i \rightarrow j$ .

Knowing the throughput of the individual edges of the network topology, one can calculate the end-to-end throughput of a source-destination pair as being equal to the worst throughput of an edge along the single shortest path being chosen for packet delivery:

$$\text{End-to-End Throughput} = \min(\text{Edge Throughput}). \quad (5)$$

The edge throughput can be given as a dimensionless ratio in per cent:

$$\text{Edge Throughput} = \frac{\min(R, \lambda_{\text{Output Port}})}{\lambda_{\text{Output Port}}} * 100\%. \quad (6)$$

**Numerical Technique**

The parameters used for the computational experiments are fixed to certain normalized values, as follows:

- average source rate of a single active node,  $\lambda = 0.1$ ;
- average port rate,  $R = 1$ .

Therefore, a single node will use no more than 10% of the available channel bandwidth per port. It is to be expected that the increase of the number of nodes will follow to a rapid congestion for a fixed value of  $\lambda$ .

The computational experiments can be performed for an arbitrary number of nodes where the upper limit depends only on the limited operational memory and processing speed of the computer configurations in use. The computational experiments produce the data for both latency and throughput graphs (Chawengmahapeeti 2006).

**Sample Topologies**

Randomly generated arbitrary topologies from 25 to 100 nodes are used. Sample networks as shown in Figs. 4, 5, and 6. A 5x5 mesh topology of 25 nodes is used as well, as an example of a homogeneous planar network. The dashed lines represent skipped edges (not used after applying topology control) as well as gateway edges. The solid lines show preferred neighbors. The topology control is used to separate the networks into trees and to interconnect the trees into a forest. One can notice that the gateway of one zone can be connected to more than one gateway from another zone. That is why the number of paths between zones is not unique and several alternative paths can be traversed depending on the gateway interconnections.

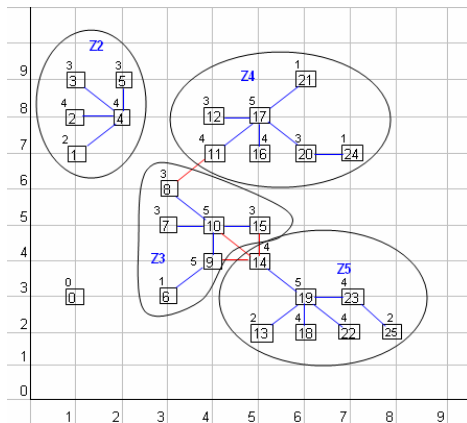


Fig. 4. A sample 25-node network (see also Nikaiein and Bonnet 2004). In fact, the topology contains 26 nodes, where node 0 is disconnected from the other nodes in the network.

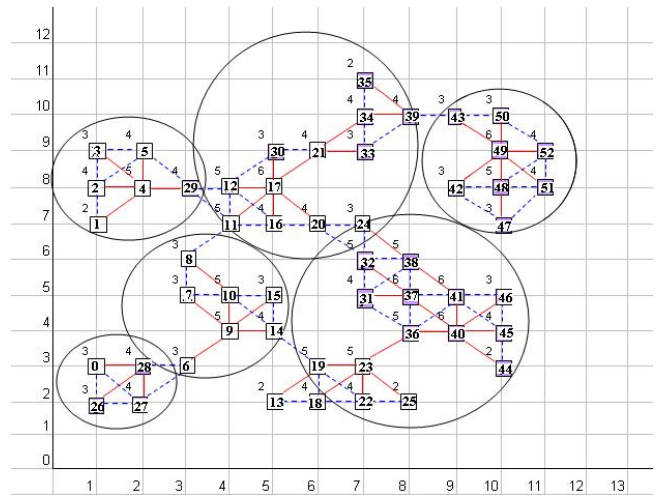


Fig. 5. A sample 53-node network.

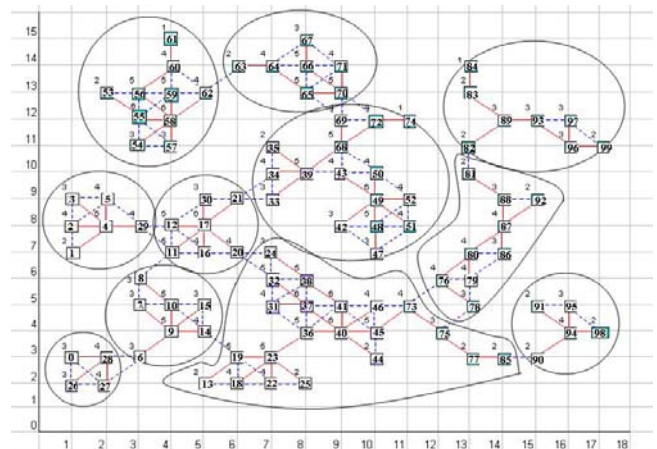


Fig. 6. A sample 100-node network.

**Computational Experiments**

The topology control technique used as a basis for this study has been introduced recently in a series of papers (Nikaiein *et al.* 2000; Nikaiein *et al.* 2000b; Nikaiein *et al.* 2001; Nikaiein *et al.* 2002; Nikaiein 2003; Nikaiein and Bonnet 2004) as a part of the Distributed Dynamic Routing (DDR) protocol. It relies on a cluster-based approach for constructing a forest of trees formed from connected nodes.

**Forest Construction Algorithms**

Several forest construction algorithms will be considered and compared to the standard forest algorithm (Nikaiein *et al.* 2000) which is obtained by choosing a preferred neighbor having the highest number of neighbors within communication range and highest network identification (NID) number

when one must choose between two nodes of the same node degree. The algorithms are listed below.

**Algorithm 1:** The standard forest (Nikaein *et al.* 2000) is constructed by choosing a preferred neighbor having the highest number of neighbors within communication range and lowest NID when one must choose between two nodes of the same node degree as shown in Fig. 4. This modification is in fact logically equivalent to the standard forest but will obviously produce a different forest. It is used to illustrate the multitude of forests that can be obtained with similar techniques.

**Algorithm 2a:** The forest is constructed by choosing a preferred neighbor having the number of neighbors closest to a *degree threshold* equal to 3 within communication range and highest degree/NID when one must choose between two nodes of the same proximity to the threshold.

**Algorithm 2b:** The forest is constructed by choosing a preferred neighbor having the number of neighbors closest to a *degree threshold* equal to 4 within communication range and highest degree/NID when one must choose between two nodes of the same proximity to the threshold.

Algorithms 1 and 2a/b rely on first neighbor information. The said algorithms can be modified to handle two-hop information in a number of different ways. Here the *cumulative* approach will be used where the total sum of the node degree of a given first neighbor and the node degrees of the neighbors of that first neighbor is obtained. Algorithms 3 and 4a/b below are based on this concept.

**Algorithm 3:** The forest is constructed by choosing a preferred neighbor providing the highest cumulative number of node degrees within *two hops* away and highest NID when one must choose between two nodes of the same cumulative node degree.

**Algorithm 4a:** The forest is constructed by choosing a preferred neighbor providing the cumulative number of node degrees within *two hops* away being closest to a *degree threshold* equal to 3 within communication range and highest degree/NID when one must choose between two nodes of the same proximity to the threshold.

**Algorithm 4b:** The forest is constructed by choosing a preferred neighbor providing the highest cumulative number of node degrees within *two hops* away being closest to a *degree threshold* equal to 4 within communication range and highest degree/NID when one must choose between two nodes of the same proximity to the threshold.

Algorithms 1, 2a/b, 3, and 4 are compared to the plain scheme with no topology control and also with the standard forest construction algorithm. It is to be expected that the network diameters of the forests produced by threshold-based Algorithms 2 and 4 should increase when compared to Algorithms 1 and 3. This increase depends on the exact network configuration and variations of the network diameter should appear on a wide range. However, the main concepts behind the forest construction are not the minimization of the number of hops but rather a simplification of the path discovery and loop avoidance when using shortest paths.

Algorithms 3 and 4a/b are two-hop ones, indirectly based on a concept of two-hop authentication by Yau and Mitchell (2004). The reason for including such algorithms with higher connectivity latency is to attempt to stabilize the routing performance by choosing parts of the network with better overall connectivity. Degree thresholds 3 and 4 are considered as two separate cases because it is not a priori known what is the optimal threshold and one can study the differences in terms of latency and throughput for each case. If one chooses thresholds greater than four in planar networks, the case will be similar to the standard forest algorithm. Thresholds equal to one or two would follow to zones that contain nodes predominantly connected in a line.

Therefore, thresholds 3 and 4 are a subject of special interest for this study for both one-hop and two-hop algorithms. A threshold equal to 3 should follow to an increase of the number of trees in the forest and also to an apparent reduction of their size. However, this would increase the number of gateways between the zones and therefore slightly increase the number of alternative shortest paths to be used for routing.

There are two possible ways in sending and collecting two-hop information. The first way is to collect initially the information from first neighbors and calculate individually the cumulative one-hop node degree. Then the cumulative node degree can be broadcast back to first neighbors and this process can be accomplished in two consecutive routing steps. The second way is to distribute the node degree information among first and second neighbors with the use of broadcast beacon packets with time to live (TTL) equal to two hops. During the second routing steps the nodes are simply forwarding beacon signals to second neighbors.

The reason to discuss both ways is that there is an essential problem is the collection of information from selfish nodes. It is assumed that the nodes in the network are cooperative, however, in reality the level of cooperation may depend on the software implementation in each node. With the first approach, during the second routing step a selfish node would apparently report a lower cumulative node degree not to be chosen frequently as a preferred neighbor. Hopefully, with the second approach, the selfish node can drop two-hop beacon packets or intentionally modify them. However, if a two-hop acknowledgement is used and all the beacon packets are authenticated, (Yau and Mitchell 2004) the modification of the content of the beacons will be detected. If the selfish nodes attempt to simply drop beacon packets, the effect of packet dropping can be detected statistically and the node can be forced to relay more traffic or simply be excluded from the network as a penalty.

The results presented in this paper are obtained assuming that all the nodes are cooperative in order to study the optimal capabilities of the topology algorithms.

## Results and Discussion

The most useful statistical information on latency and throughput is obtained on the basis of the average values for different path lengths. Fig. 7 shows the end-to-end latency per hop for different path lengths of the sample 25-node topology in Fig. 4. The observed result here is that the plain scheme (full connectivity, no topology control) demonstrates the highest latency. It turns out that the performance of the forest in terms of latency outperforms substantially the plain scheme. A possible explanation can be found in the increased number of hot spots in the plain scheme due to the more frequent overlapping of shortest paths. This find can be considered as an indication of the effectiveness of tree-based topology control.

The second find is that the latency of the two-hop forest (Algorithm 3) is the smallest one. Smaller latency is observed only for the two-hop forest with threshold = 3 (Algorithm 4). Unfortunately, threshold = 3 follows to an apparent increase of the network diameter. Therefore, the two-hop Algorithm 3 is the most effective one for this sample topology.

Fig. 8 shows the throughput per hop for different path lengths of the sample 25-node topology. The throughput reduces steadily with the increase of the path length.

It should be mentioned that the average end-to-end latency is calculated on the basis of the samples of non-congested paths only. The average end-to-end throughput is calculated on the basis of all the paths.

The ad hoc topologies are quite asymmetric typically and contain multiple bottlenecks. In order to study the performance of the proposed algorithms on a homogeneous topology, a 5x5 mesh of 25 nodes is considered. Fig. 9 shows the end-to-end latency per hop for different path lengths of the sample 25-node mesh topology.

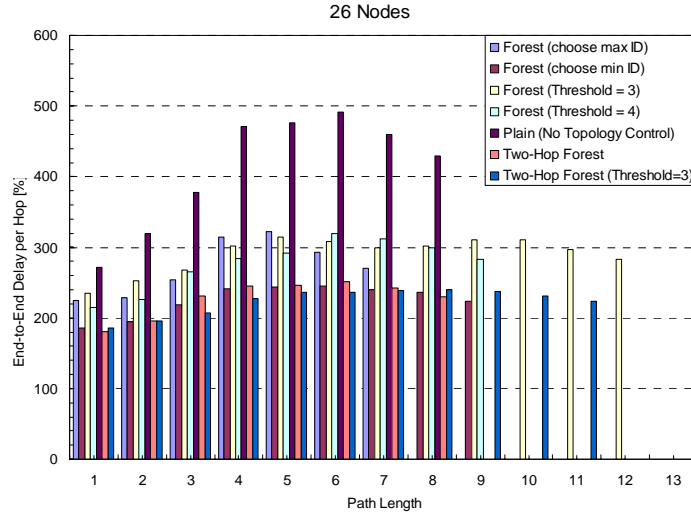


Fig. 7. End-to-end latency per hop for different path lengths, 25 nodes.

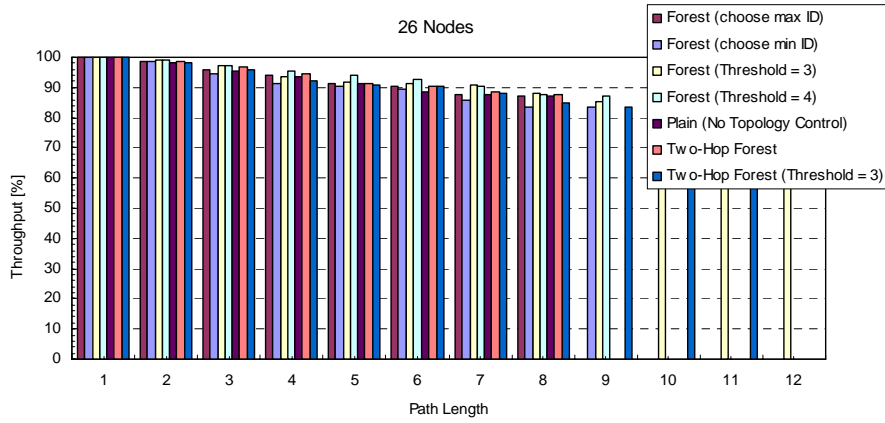


Fig. 8. End-to-end throughputs per hop for different path lengths, 25 nodes.

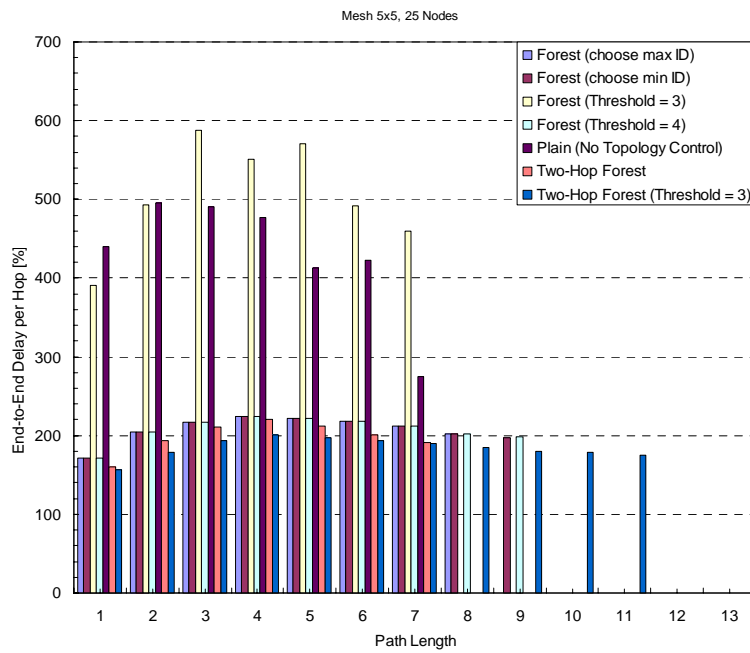


Fig. 9. End-to-end latency per hop for different path lengths, 25 nodes (mesh).



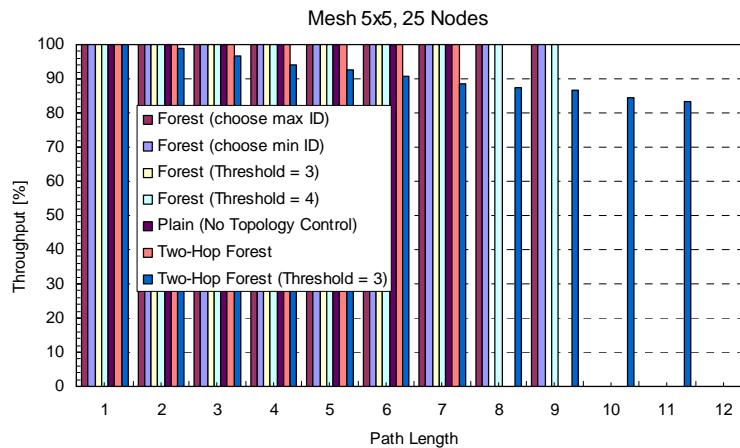


Fig. 10. End-to-end throughputs per hop for different path lengths, 25 nodes (mesh).

The same observations as for the previous 25-node scheme are valid here for the 25-node homogeneous topology. The two-hop Algorithm 3 performs better than the others and use of the the plain scheme results in a substantial increase of the latency. An interesting find is that the Algorithm 2b (Threshold = 3) produces the worst latency. This can be partially explained with the observation that the average node degree of the mesh equals 4 and the threshold = 3 would result in poor zone segmentation. Not surprisingly, even the throughput drops rapidly for the two-hop Algorithm 4a with the same Threshold = 3 as shown in Fig. 10.

Similar performance of the forest construction algorithms is observed also with the increase of the number of nodes up to 100 nodes (Chawengmahapeeti 2006). It appears that the two-hop forest construction Algorithm 3 can be used successfully for both homogeneous and non-homogeneous distributions of nodes in planar networks. Another advantage of this algorithm is that selfish nodes (intentionally reporting low network degrees in their beacon signals) could also be used for routing if the two-hop authenticated beacon signals from all of their neighbors indicate an increased average value of the connectivity in the local environment of the selfish node, which would allow other nodes to use the said selfish node more frequently as a preferred node. This would turn the selfish node from a leaf node into a branch or even a gateway node. The increased latency

in collecting two-hop information about second neighbors is therefore compensated with the improved reliability of the constructed zones. The number of zones generated by the different algorithms for the sample topologies in Figs. 4, 5, and 6 is shown in Table 1.

The algorithm can be adaptively chosen and modified according to the network conditions. In static non-congested networks, the standard forest construction would be sufficient to handle the traffic. With the increase of the overall uncertainty of the network in terms of traffic, mobility, and faulty conditions, one can choose the two-hop Algorithm 3 and generate a number of zones comparable with the standard forest. In non-homogeneous topologies with high mobility, smaller zone diameters are the preferred choice and algorithms 2a and 4a would produce an increased number of small zones with an average degree of connectivity approaching the degree thresholds of 3 and 4. The aspects of mobility are considered here indirectly in terms of the zone diameter and the number of simultaneous connections within the all-port model that can be established in a long-term period. The first aspect requires a small zone diameter for low latency in the exchange of control packets. The second aspect requires a reduced node degree of fast moving nodes to avoid short-term link establishments which would interrupt the data forwarding.

Apparently, the forest construction algorithms applied to homogeneous topologies may produce a single zone containing all the



nodes in the network as for the case of the 5x5 mesh. However, this is a worst case scenario and the slightest distraction of the homogeneity would result in the creation of a number of zones.

Table 1. A comparison of the number of zones for different algorithms.

Algorithm	Number of Zones			
	25 nodes (mesh)	25 nodes	53 nodes	100 nodes
Standard Forest	1	4	7	11
Algorithm 1	1	3	7	11
Algorithm 2a	4	5	9	17
Algorithm 2b	1	3	8	14
Algorithm 3	1	3	4	8
Algorithm 4a	1	5	9	15
Algorithm 4b	2	4	5	7
Plain Scheme	none	none	none	none

In ad hoc networks, the nodes are randomly distributed and even local homogeneity can seldom be observed. The increased number of zones raises the question of the power consumption of gateway nodes. All the nodes in the network are assumed to have equal capabilities in forwarding traffic. Most of the gateway nodes are leaf nodes from the constructed trees. It means that an increased traffic will be carried between zones through a limited number of gateways. In static networks, this would follow to a rapid battery exhaustion of the gateway nodes especially if cases of bottlenecks between two partially connected parts of a large network. However, the increased mobility would follow to dynamic zone modifications which would reduce the long-term gateway service. Also, it is not obvious whether the leaf nodes or the roots of the trees would forward more traffic. There can be several gateway nodes connecting two zones, but there is always only one root of the tree in each zone. With the increase of the zone

diameter, the root becomes more congested and the leaf nodes on the average are less congested. It can be assumed that for zones of middle size both gateways and roots carry similar traffic on the average. The intermediate branch nodes of the trees seem to forward less traffic. Branch nodes can serve as gateway nodes as well. For an arbitrary topology, it is important to locate the hot spots, where the overlapping of multiple paths would result in congestion. The cumulative two-hop information about the node neighborhood helps in having an appropriate root of the tree, which in turn eases the traffic forwarding because not all the traffic is being handled by the root but also by the multiple branches that forward inter-zone traffic from/to the gateway nodes. In such cases, a branch node can serve as a local root for a limited number of leaf nodes. The two-hop forest algorithm introduced in this paper can be considered as a contribution to the research on hybrid routing protocols where the importance of the implementation of topology control is better understood.

### Conclusion

The main contributions of this paper can be summarized as follows:

- A framework for numerical evaluation of routing in ad hoc networks is developed. This approach allows one to test quickly the performance of arbitrary networks on the basis of computational experiments.
- Two modified forest constructions are proposed and studied. The first one deals with the choice of a preferred neighbor for tree construction with respect to a degree threshold in order to minimize the variability of incoming traffic. The second one uses two-hop neighbor information to increase the topological stability and cope with selfish nodes.
- The proposed forest constructions are compared with the standard forest and with the plain scheme. The choice of a preferred neighbor with respect to the average value of the node degree in the

network follows to a slightly better performance than the standard forest. The main advantage here is that the node degrees of tree-connected nodes are more uniformly distributed across the network, thus preventing the quick battery exhaustion.

- The comparison with the plain scheme reveals a substantial reduction of the end-to-end latency for a significant number of nodes. This is interpreted in terms of the reduced variability of traffic through the nodes. On the contrary, the plain scheme increases the risk of the occurrence of hot spots when using shortest paths.

## References

- Chawengmahapeeti, R. 2006. Performance evaluation of a hybrid routing protocol for mobile ad hoc networks. MS TS Thesis, Assumption University, Bangkok, Thailand.
- IEEE 802.11 LAN/MAN Wireless LANs. 2007. Available: <http://standards.ieee.org/getieee802/802.11.html>.
- Kleinrock, L. 1976. Queueing Systems, Volumes 1 and 2. John Wiley & Sons, New York, NY, USA.
- Nikaein, N.; Labiod, H.; and Bonnet, C. 2000. DDR: Distributed Dynamic Routing Algorithm for Mobile Ad Hoc Networks. Proceedings of the 1<sup>st</sup> ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2000), Boston, MA, USA, 11 August, pp. 19-27, Association for Computing Machinery (ACM), New York, NY, USA.
- Nikaein, N.; Wu, S.; Bonnet, C.; and Labiod, H. 2000b. Designing Routing Protocol for Mobile Ad Hoc Networks. Proceedings of DNAC, Paris, France, 14-17 November, Kluwer Academic Publishers.
- Nikaein, N.; Bonnet, C.; Moret Y.; 2001. HARP-Hybrid Ad Hoc Routing Protocol. Proceeding of IST, Tehran, Iran, 1-3 September, International symposium on telecommunications.
- Nikaein, N.; Bonnet, C.; Moret, Y.; and Rai, I. A.; 2002. 2LQoS – Two – Layered Quality of Service Model for Reactive Routing Protocols for Mobile Ad Hoc Networks. Proceedings of SCI, Orlando, USA, 14-18 June, 6th World Multiconference on Systemics, Cybernetics and Informatics.
- Nikaein, N.; 2003. Architecture and Protocols for Supporting Routing and Quality of Service for Mobile Ad Hoc Networks. Ph.D. Thesis, Reference 2808 EPFL.
- Nikaein, N.; and Bonnet, C. 2004. Topology Management for Improving Routing and Network Performances in Mobile Ad Hoc Networks. Mobile Networks and Applications, Kluwer Academic Publishers, Dordrecht, The Netherlands, 9(6): 583 – 594.
- Yau, P.-W.; Mitchell, C.J.; 2004. 2HARP: A Secure Routing Protocol to Detect Failed and Selfish Nodes in Mobile Ad Hoc Networks. Proceedings of the 5<sup>th</sup> World Wireless Congress (WWC 2004), San Francisco, CA, USA, 15-28 May, pp. 1-6, Delson Group Inc., Saratoga, CA, USA.