On the motivation behind Clustering Protocols in Ad-Hoc Networks

Mohit Garg
Guides: D. Manjunath and R. K. Shyamasundar

***Abstract*— Though ad-hoc networks have been studied in great details by researchers all over the world, not much work has been done in analytically understanding the behaviour of the routing protocols. In this work, we analytically analyse the performance of hierarchical routing protocols *vis a vis* reactive routing protocols. We formulate the problem in a general framework and derive conditions under which one will be preferable over the other. We also comment on the various parameters involved and how can they be modelled.**

The results should be useful for generic as well as specific performance comparison studies between clustering and reactive protocols.

***Index Terms*— Analytical comparison, Ad-Hoc routing, Clustering, Reactive, Flooding, Overhead, Power Control.**

I. Introduction

AD-HOC Networks consist of nodes (usually mobile), scattered around in a geographical area without any fixed infrastructure. Though primarily developed for military applications, MANets (Mobile Ad-Hoc NETworks) have attracted a lot of industry and research interest recently.

The major challenge in MANets is to route packets without any control on the topology of the network – since the nodes are freely mobile. The fact that there is no fixed infrastructure (like base stations in cellular networks) available which can be relied upon makes this problem all the more difficult to solve. Also, since each of the nodes is usually a portable device, power management strategies become important considerations in selecting routing schemes to deploy in practical situations.

Broadly speaking, one can divide the available schemes in literature into three broad categories :-

- *Pro-Active Routing*: In these approaches, each node pro-actively maintains routes to all other nodes in the network by periodically exchanging link-status information with neighbouring nodes. These schemes require an overhead to exchange link-status information regularly. Well known example is DSDV [11].
- *Reactive Routing*: In these approaches, routes are computed on an 'on-demand' basis *i.e.* as and when a source node requires, a route is computed by complete or partial flooding of the network. These schemes incorporate certain overheads for route discovery. [9], [10].
- *Hierarchical Schemes*: These techniques, tend to achieve some sort of a compromise between pro-active and reactive approaches. They define and form logical entities called 'clusters' which are nothing but a local group of nodes. Almost all the schemes which fall in this category maintain some sort of pro-active approach to routing within the clusters (intra-cluster routing) while finding inter-cluster routes on an on-demand basis *i.e.* reactively. [7], [8].

Looking at the wide spectrum of protocols available, it becomes extremely essential for network designers to be aware of conditions under which each of the above class of protocols will be better than the other two.

From our experience in fixed wireline networks, it is clear that 'pro-active' protocols will perform better in case of low mobility of nodes since the link-status update overhead will be low if the network scenario is relatively 'static'. But, in the case of a highly dynamic environment, the 'pro-active' schemes will lead to intolerably large overheads making them unviable for deployment in such situations.

The case of 'cluster-based'[1] and reactive protocols is, however, not very clear. Both have certain overheads and one needs to investigate under which conditions will one be better than the other.

In this work, we develop a general framework which can act as an aid to understanding the relative performance of 'reactive' and 'hierarchical' protocols. The remainder of this document is organised as follows – In Section II, we discuss work related to our analysis, Section III formulates the problem, Section IV contains the actual analysis and we finally conclude in Section V.

[1]We will be using the terms 'cluster-based' and 'hierarchical' interchangeably. Ditto for 'reactive' and 'flooding'.

II. Related Work

Many studies have been performed [3], [6] to study the behaviour of various protocols in different scenarios. Unfortunately, almost all of the analysis has been simulation based. It is well noted in [1] that *'Simulation results, though extremely useful, are often limited in scope to specific scenarios'*. Thus, there is a need to analytically analyse the performance of the three routing frameworks *vis a vis* each other.

Santivàñez, *et al* in [1] present an asymptotic analysis of some common protocols in terms of their overheads. Their analysis is mostly 'order of magnitude' based but provides vital insights into the performance of the algorithms. In our work, we specifically analyse 'hierarchical' protocols *vis a vis* 'reactive' schemes and give a more detailed analysis by characterising different terms involved in the exact expressions for the overheads in the two frameworks.

Jacquet and Viennot [2] also perform an analytical analysis of the generic 'pro-active' and 'reactive' protocols, but they do not consider 'hierarchical' schemes.

Simulation based comparison of overheads are also available in literature [6]. In almost all of them, clustering performs better than reactive schemes as the network size increases. We also derive a similar result on the number of 'flows' in the network.

We present our analysis next.

III. Problem Formulation

In a flooding based environment, each source node will initiate a route finding request as and when needed and this request will propagate through the network causing flooding in some manner depending on the actual protocol used. Here, we analyse the performance of a clustering scheme built over and above this protocol w.r.t the unclustered network. We will compare their *overheads* and hence power and bandwidth consumption[2].

A. Overheads

Before we formulate the problem, it is important to understand the meaning of the term *'overhead'*. Strictly speaking, any packet(s) exchanged in a network other than the data packets is an overhead. Since these packets eat up the available bandwidth, it becomes imperative to keep these overhead packets to the minimum possible required for the proper functioning of the network. For a given routing scheme, the overheads can be classified in the following three categories [1] :-

- *Pro-Active Overhead*: This is due to the link-status updates that need to be exchanged between nodes to pro-actively maintain routes to one or more destinations in the network. This is a major contributor to overhead in the pro-active routing framework. The reactive approaches are free from this overhead, but the hierarchical schemes have some pro-active overhead for intra-cluster routing and also for cluster maintenance.
- *Reactive or Flooding Overhead*: This occurs while route discovery in flooding type scenarios *i.e.* whenever a 'source' node wishes to establish a route to a destination, some amount of extra control packets are exchanged between nodes in the network for computing the required route. This overhead is absent in pro-active protocols but is the major contributor for reactive protocols. Some amount of flooding overhead also exists in hierarchical protocols.
- *Sub-Optimal Route Overhead*: Each routing protocol optimises the route computation based on some metric. But as time progresses, due to the mobility of the nodes, the route may become sub-optimal. Thus, this overhead needs to be taken into account while comparing two protocols. [3]

B. Assumptions

Now that we have defined the meaning of 'overheads', we formulate the scenario under which we will be analysing clustering and flooding. We do not assume any specific protocols but rather develop a framework under which one can study the differences between a reactive protocol *vis a vis* a clustering protocol which satisfy certain basic assumptions which we list down presently. Let network A be the unclustered network and B be the clustered one.

- The clustering scheme in network B uses a pro-active protocol for intra-cluster routing and the *same* reactive routing protocol as network A for inter-cluster route discovery.
- We do not assume a fixed number of nodes in the network. We allow nodes to join and leave the network with time *i.e.* the number of nodes in the network at any given time is a random process. We assume the time stationarity of this process.
- Since the total number of nodes in the network is not fixed, we can assume that all clusters are statistically I.I.D. in their characteristics.

[2]If similar power control measures are adopted in clustering and reactive protocols, one with a lower *overhead* will be better off in terms of power consumption metrics.

[3]We will show in the appendix that this term in the overhead is exactly the same in both the hierarchical and flooding protocols if the hierarchical protocol is built over the same flooding protocol.

- We do not assume anything on the transmission ranges of the nodes. They may be changed adaptively for power and energy conservation. But both networks A and B will have follow the same algorithm for power management.
- We assume that the link-status and cluster maintenance overhead is independent of the data traffic being generated in the network and only depends on the link-states and mobility condition, *i.e.* clusters are formed and maintained even when data is not being transmitted in the network.
- We also assume a concept of 'border' or 'gateway' nodes in each cluster which will have complete information about the clusters to which they belong to and hence will take part in the flooding process in network B. Most of the clustering protocols incorporate a similar concept in them.

Since we wish to understand the behaviour of the clustering scheme on a given network, we make the conditions in A and B exactly identical. By 'identical' we mean that the two networks have the same topology at each instant of time with same sets of nodes communicating at any given instant of time in both the networks. In other words we will do a *sample path* based comparison of the two schemes *i.e.*

$$P_A(\mathbf{X}, t) = P_B(\mathbf{X}, t) \quad \forall \;\; t \tag{1}$$

where, $P_A(\mathbf{X}, t)$ denotes the node positions at time t in network A, similarly for B. We will refer to this as the 'location sample path'. Also,

$$LS_A(\mathbf{X}, t) = LS_B(\mathbf{X}, t) \quad \forall \;\; t \tag{2}$$

where, $LS_A(\mathbf{X}, t)$ denotes the link-states at time t in network A, similarly for B. We will refer to this as the 'link-status sample path'. Finally,

$$A(\mathbf{S}, \mathbf{D}, t) = B(\mathbf{S}, \mathbf{D}, t) \quad \forall \;\; t \tag{3}$$

i.e. the source-destination pair vectors at time instant t is same in both the networks. In simple words, the assumption is that at a given time instant if S was transmitting data to D in network A, it will also be doing so in network B. This will be referred to as the 'flow sample path'.

With this common base, we will derive conditions in which network B will be better in terms of overhead. It is clear that the network with a lower overhead will be able to achieve a higher data throughput in the given time interval.

IV. Analysis

Consider a sample path for the network processes in the two networks. Both A and B follow the same sample path. We will analyse the network in a time window and compare the overheads in the clustered (network B) vs. the non-clustered (network A) environments. We will first formulate the problem exactly and then go on to replace quantities by their expected values so as to be able to quantify the parameters for the entire ensemble. Finally, we will characterise the parameters under some assumptions.

A. The Model

Consider an observation time interval of t_0. Let there be M communicating pairs of nodes (or *flows*) during this interval. Denote the time for which a route between the m^{th} pair, already computed to remain valid by $T^{k,m}$, where k denotes the k^{th} computation of the route. This is a random variable. We will show in the appendix that this time is same for both the clustered and the unclustered network since the routes actually computed will be identical – at least under commonly used topological path metrics. [4]

We will characterise the distributions of all the random variables later. Let Ct_0 are the total number of clustering packets exchanged in the time interval t_0. This includes the cluster maintenance messages and the link state updates propagated in the clusters. *i.e.*

$$Ct_0 = \text{LSUs} + \zeta \tag{4}$$

where, 'LSUs' denotes the number of link-status updates propagated in the network, and ζ denotes the total cluster maintenance overhead messages exchanged, in the time interval t_0 under consideration.

Let R_m be the number of times the m^{th} route is recomputed in the time duration t_0. A point to note is that if the source and destination nodes belong to the same cluster in network B, this route recomputation will not be done through the flooding protocol, rather will be taken care of by the intra-cluster link-status updates. We do not consider this special case since the probability of this event happening is quite low especially in large networks. But it is to be noted that by taking this computation as a reactive overhead in B, we will be over-estimating the cluster overhead.

Thus, for clustering to be better than the reactive routing protocol, the following inequality needs to be satisfied:-

[4] In the following analysis, uppercase denotes random variables

$$Ct_0 + \sum_{m=1}^{M}\sum_{k=1}^{R_m}\Gamma_B^{k,m} \leq \sum_{m=1}^{M}\sum_{k=1}^{R_m}\Gamma_A^{k,m} \qquad (5)$$

In (5), $\Gamma_A^{k,m}$ denotes the overhead messages exchanged for the computation of the k^{th} route for the m^{th} flow in network A. Note that since the time instant at which the route is computed in both the networks is the same, (since the location and flow sample paths are the same implying identical paths existing and hence paths being invalidated at the same instants of time and thus being recomputed at the same time), this overhead can be divided into the overhead per cluster ($\Gamma_{A,i}^{k,m}$) of network B by considering the clusters in network B at the instant of computation of the route. In other words, we are considering 'dotted clusters' in network A identical to those in B and divide the overhead into each of these 'dotted clusters'. $\Gamma_B^{k,m}$ has a similar meaning in network B.

In all flooding based protocols, each node relays a route formation request at most once. Since most of the flooding in network B will take place via 'border' nodes, all nodes in a cluster may not be involved in route computation. In network A, there is no concept of 'border' nodes and thus all nodes in a cluster may be flooded. We denote the fraction of nodes in B (w.r.t. number of participating nodes in cluster i in network A) participating in route discovery in the i^{th} cluster by $\Phi_i^{k,m}$ where $\Phi_i^{k,m} \leq 1$. This parameter is defined only for network B.

Also, we will drop the subscript A in $\Gamma_{A,i}^{k,m}$ since this is only defined for network A. Thus, rewriting (5), we get:-

$$Ct_0 \leq \sum_{m=1}^{M}\sum_{k=1}^{R_m}(\Gamma_A^{k,m} - \Gamma_B^{k,m}) \qquad (6)$$

i.e.,

$$Ct_0 \leq \sum_{m=1}^{M}\sum_{k=1}^{R_m}(\sum_{i=1}^{N^{k,m}}\Gamma_i^{k,m} - \sum_{i=1}^{N^{k,m}}\Phi_i^{k,m}\Gamma_i^{k,m}) \qquad (7)$$

where, $N^{k,m}$ denotes the number of clusters which relay the k^{th} route computation for the m^{th} flow. Note that though the cluster topology is likely to change for route computations at different instants of time, $N^{k,m}$ only concerns itself with the number of clusters involved in the route computation and does not 'physically' identify the individual clusters.

Now, for a given location sample path, the location of nodes does not change with the traffic conditions. Note that the clustering overhead (Equation 4) is independent of the data traffic generated in the network, and only depends on the location and link-status sample paths. Thus, from (7), we can see that as M increases, the right hand side changes non-decreasingly (as $\Phi_i^{k,m} \leq 1$). This means that as the number of communicating pairs increase, the number of route find requests increases leading to an increased savings in the clustered network w.r.t. the route computation overhead. Thus, for a fixed clustering overhead, the case for the clustering scheme becomes stronger.

Hence, we can conclude from (7) that for any given clustering scheme, we will always be able to find a threshold for the data traffic which will make the implementation of the scheme feasible over a purely reactive approach.

B. Taking ensemble averages

Equation 7 captures the trade-off involved in a particular sample path. Now assuming *ergodicity* of the processes involved, we divide both sides by t_0 and take limits as $t_0 \to \infty$. Thus,

$$\lim_{t_0\to\infty}\frac{1}{t_0}Ct_0 \leq \lim_{t_0\to\infty}\frac{1}{t_0}[\sum_{m=1}^{M}\sum_{k=1}^{R_m}\sum_{i=1}^{N^{k,m}}(\Gamma_i^{k,m} - \Phi_i^{k,m}\Gamma_i^{k,m})] \qquad (8)$$

Applying *ergodicity* by replacing time averages by expectations, we get

$$E[Ct_0] \leq E[\sum_{m=1}^{M}\sum_{k=1}^{R_m}\sum_{i=1}^{N^{k,m}}(\Gamma_i^{k,m}) - \Phi_i^{k,m}\Gamma_i^{k,m})] \qquad (9)$$

The limits of the summations need to be characterised since we will not be able to take expectations inside the summations unless the limits are fixed. Thus,

- Since R_m is a random process w.r.t. m, we will replace it by $E(R_m)$. The justification for this step is that R_m for a given flow will depend only location sample paths. Specifically, it will depend only on the amount of time for which the m^{th} flow is active and the location of nodes in that time interval. We therefore, replace R_m by the expected value of the number of times the m^{th} flow needs to be recomputed. Here the expectation runs over all the location sample paths.
- For the computation of a fresh route from S to D, we can assume that all the clusters relay the request. Thus, $N^{k,m}$ may be taken to be the total number of clusters in the network. Note that this will result in an upper bound w.r.t. the overhead since route repairing can be done locally which may not be flooded to all clusters in the network. One

may characterise this more closely w.r.t. the actual reactive protocol under consideration. Now, since we assume I.I.D. clusters, we will replace this by the average number of clusters in the network, $E(N_c)$.

Using the above, we can rewrite (9) as follows:-

$$E(Ct_0) \leq \sum_{m=1}^{M} \sum_{k=1}^{E(R_m)} \sum_{i=1}^{E(N_c)} (E(\Gamma_i^{k,m}) - E(\Phi_i^{k,m}\Gamma_i^{k,m})) \quad (10)$$

C. Characterising the R.H.S. Variables

We now list some observations which may help in characterising the various terms in (10).

- $T^{k,m}$ depends only on the mobility of the nodes. At a coarse level, one can assume a uniform and stationary rate of link degradations say, λ_{lb}. Let the average route length be $L^{k,m}$ – this can be taken stationary w.r.t. both k and m. This implies that $T^{k,m}$ will also be stationary. Thus, we get a Poisson distributed link breakage process where $T^{k,m}$ is the time duration for which no link along the route breaks. Hence,

$$f_{T^{k,m}}(t^{k,m}) = \lambda_{lb} L^{k,m} e^{-\lambda_{lb} L^{k,m} t^{k,m}} \quad (11)$$

  $E(R_m)$ can be computed once $T^{k,m}$ is characterised. Hekmat and Mieghem in [5] do a numerical simulation based analysis for the average minimum hop count between two nodes and come up with a unimodal distribution. One can go deeper into the distribution of $L^{k,m}$ if more is known about the location process in the network.
- If we assume that once a request arrives in a cluster, it is forwarded to all cluster members in the reactive framework, $E(\Gamma_i^{k,m})$ can be taken to be the average number of nodes in each cluster ($E(N_i)$). Such an assumption is typically true of protocols like DSR [10] wherein each node sends the route request to all its neighbours. Note that the average number of nodes per cluster will depend on the actual clustering scheme being used.
- $\Phi_i^{k,m}$ is the fraction of nodes (w.r.t. network A) in cluster i, which take part in the flooding route request in network B. This consists of all border nodes plus some core nodes in the cluster. (see appendix for a detailed discussion on routing).

  Let there be B_i border nodes in cluster i. We assume that all these nodes take part in route discovery. The number of border nodes in the cluster is a strong function of the topology of the network. Since things are random, it is in general difficult to estimate the topological characteristics. But some simple bounds can be calculated – the connectedness of the network implies that each cluster must have at least one border node. This gives a lower bound on the number of border nodes as the number of clusters (N_c). An upper bound can also be derived considering the links in the network. Let Υ_N be the total number of edges in the network and Υ_c of these be intra-cluster edges. Thus, the number of border nodes is bounded by:-

$$N_c \leq \sum_{i=1}^{N_c} B_i \leq 2(\Upsilon_N - \Upsilon_c) \quad (12)$$

  Simulations and prior knowledge about the clustering scheme may be used to better characterise B_i.

  Now, let the each of the core nodes take part in the routing process with a probability p_c. This probability will be large in case of sparsely distributed network and lower for more dense topologies. Thus,

$$E(\Phi_i^{k,m}\Gamma_i^{k,m}) = E(B_i) + p_c(E(N_i) - E(B_i)) \quad (13)$$

  One can compute $E(N_i)$ once the clustering scheme is known. $E(B_i)$ can also be approximated by:-

$$E(B_i) = E\left(\frac{\sum_{j=1}^{E(N_c)}(B_j)}{N}\right)E(N_i) \quad (14)$$

  Thus, we can derive some approximate expression for the flooding overhead in the cluster-based network.
- M is the traffic threshold which can be computed from the equation once the other parameters are known.

D. Clustering Overhead

The L.H.S. of (10), is the average total clustering overhead in time interval t_0. Rewriting (4)

$$Ct_0 = \text{LSUs} + \zeta \quad (15)$$

where, 'LSUs' denotes the number of link-status updates propagated in the network, and ζ denotes the total cluster maintenance overhead messages exchanged, in the time interval t_0 under consideration. We will compute the $E(Ct_0)$ next.

A point to note here is that only those changes which may lead to a route being recomputed need to be considered significant enough to generate an update in the clustering environment. Also, only -ve status changes need to be sent immediately, all +ve changes may be sent along with the next periodic update. Again this depends on whether the clustering protocol has a provision for pro-active LSUs or only reactive. Even if was purely reactive LSUs, we can still divide the updates into -ve and +ve updates. In the current analysis, we will consider purely reactive LSUs for both +ve and -ve changes.

Taking the rate of -ve updates being the same one which leads to a link being broken (λ_{lb} and hence we can relate LSUs and $T^{k,m}$). Also note that each LSU will be propagated to all nodes in the cluster and hence the overhead for -ve LSUs will be equal to $\lambda_{lb} t_0 N_i^2$ for the i^{th} cluster. Similarly, if the +ve LSUs are generated at the rate of λ_{lf} per second per node, we get the overhead for +ve LSUs to be $\lambda_{lf} t_0 N_i^2$ for the i^{th} cluster.

Another important observation is that each cluster rearrangement (node joining/leaving) message that is generated is either due to a link breaking (-ve LSU), or a new link forming (+ve LSU). Now, lets assume that each -ve link status update leads to a member leaving a cluster with probability 'p^-'. Note that whenever a member leaves a cluster, there has to be a -ve link status update generated. A cluster rearrangement also takes place when a new member joins the network. Let the probability that a +ve LSU implying a new member joining be p^+.

Thus, if a member leaves one cluster or a new member joins the system, some cluster maintenance packets will be generated along with the LSUs. The number of cluster maintenance packets generated will be equal to the number of clusters that can 'listen' to the 'orphan' node. Let the 'cluster degree' (no. of clusters near the node) of the 'orphan' (j^{th}) node be $D_{c,j}$. So, we get $D_{c,j} + 1 + 2$ messages being generated – 1 for request, $D_{c,j}$ replies to the request,1 for request to join and 1 for accepting the request. [5]

Again, a new link being established or an old link breaking may lead to a border nodes being created. Now, since border nodes and core nodes may be handled differently in different protocols (e.g. a border node may be a member of more than one cluster, therefore, its motion may affect more than one cluster), these may lead to different overheads which can be computed using similar analysis as above. We do not consider this distinction between border and core nodes in the current analysis.

Hence, we get

[5]Also, note that this may lead to some additional overhead w.r.t. the clustering scheme if some penalties are paid by the clusters when their sizes go above or below some thresholds (this is done to keep the sizes under control). Since this is specific to the clustering scheme, we do not discuss it here.

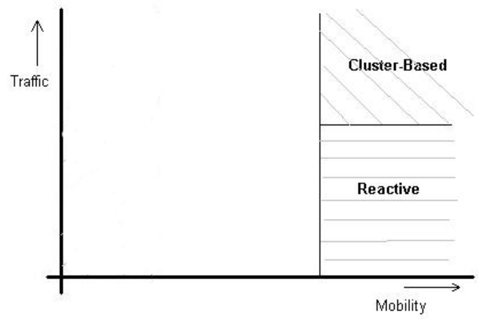

Fig. 1. Deployment conditions for routing schemes.

$$E(Ct_0) = E(C^+t_0) + E(C^-t_0) \tag{16}$$

where,

$$E(C^+t_0) = \sum_{i=1}^{E(N_c)} \lambda_{lf} t_0 \{E(N_i^2) + p^+[E(D_c) + 3]E(N_i)\} \tag{17}$$

and,

$$E(C^-t_0) = \sum_{i=1}^{E(N_c)} \lambda_{lb} t_0 \{E(N_i^2) + p^-[E(D_c) + 3]E(N_i)\} \tag{18}$$

Note that as λ_{lb} and λ_{lf} decrease, the clustering overhead decreases. In the limit as they approach zero, we get the static case in which there will be no need for cluster overhead messages. This observation agrees with the choice of hierarchical routing as the scheme in wireline networks.

In both the above equations, $E(D_c)$ needs to be computed. A good approximation to this parameter can be taken as

$$E(D_c) = E(\frac{\sum_{j=1}^{N} D_{c,j}}{N}) \tag{19}$$

i.e. expectation of the average of the cluster degrees of all nodes.

Thus, we have been able to characterise the clustering overhead and relate the R.H.S. and L.H.S. of (10). Using this general model, one should be able to compare common hierarchical protocols with reactive routing schemes.

V. Conclusion

An analytical framework for the comparison of hierarchical and reactive routing schemes was developed in this work. It was shown that the optimal paths computed in both the cases will be identical. Based on this important observation, we formulated the problem and derived the conditions under which cluster based schemes will perform better than purely reactive approaches. Finally, we also characterised the clustering overheads and other parameters in the model.

Another interesting result observed was that for a given network, cluster based schemes will always perform better than purely reactive schemes when the number of flows increases beyond a threshold. Thus strengthening the case for the deployment of hierarchical schemes in high data traffic networks. Hence, we can broadly represent the routing schemes as in Fig. 1.

The model is a general framework, yet takes a detailed view of the overhead packets being exchanged in both the environments and relates events in one to those in another thereby coming up with a analytical tool to compare the performance.

APPENDIX I
IDENTICAL PATH COMPUTATION IN THE TWO NETWORKS

Each routing algorithm tries to optimise on the basis of some metric. This metric may not be the "ideal metric" that should be minimised in order to get the best performance in the network. Also, a route once computed may slowly degrade to a non-optimal route as the nodes move and the locations change. Thus, this gives rise to the 'sub-optimal route overhead'.

What we will show is that the optimal route (w.r.t. the desired metric) computed in both A and B is the *same* – at least for topological metrics (*e.g.* min. hops or min. energy metrics [4]). This will imply that the 'sub-optimal' routing overhead will be the same in both the scenarios and we need not be bothered about this overhead while comparing the two approaches.

Consider a source node S which wishes to establish a route to a node D. We have two possible cases:-

- Both S and D belong to the same cluster in network B. Thus an optimal route to D w.r.t. the desired metric is already known to S in network B. Uniqueness of the optimal route guarantees that the reactive protocol in A will compute this optimal route. Hence both are the same in this case.
- In case S and D belong to different clusters, a reactive routing request will be initiated in network B. We will trace the path that this request takes and show that the optimal route is computed in network B too.

In any reactive routing protocol, more than one route will be computed to the destination via the flooding algorithm. Now, some agent (typically, the destination) will collect all these routes and choose the one that optimises the metric under consideration *i.e.*, it chooses

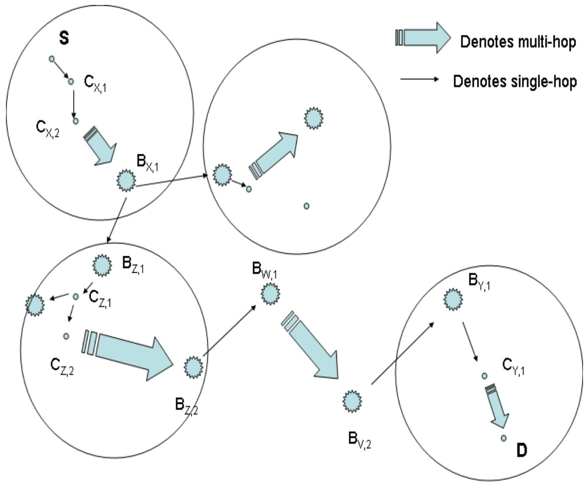

Fig. 2. Tracing the routing request in network B (not to scale)

the *globally optimal* route. Thus it is enough to prove that if the globally optimal route is retained as a possible option for the selecting agent to choose in the clustered network, then this globally optimal route will be chosen as the route between S and D, thereby proving the claim.

Let S and D belong to cluster X and Y respectively. Let the globally optimal path from S to D, be given by the sequence of nodes $\{S, C_{X,1}, C_{X,2}, ..., B_{X,1}, B_{Z,1}, C_{Z,1}, C_{Z,2}, ..., B_{Z,2}, B_{W,1}, ..., B_{V,2}, B_{Y,1}, C_{Y,1}, ...D\}$, where $B_{i,j}$ and $C_{i,j}$ denote the border and code nodes in cluster i respectively. *i.e.*, the optimal route from S to D in network A can be decomposed into border and code nodes in network B. This can always be done since in order for a core node to communicate to the outside of the cluster, it must route its packets via a border node of its cluster. Thus, the above decomposition is always valid. We will show that a path with exactly this sequence of nodes will be available to the selection agent for selection of the optimal path.

It is clear that since the above path is optimal, all segments of the path will also be optimal *i.e.* any subpath between any two nodes will be the optimal path between those two nodes. Consider a routing request originating at S in network B. Refer Fig. 2

1) Since, S knows that D is not a member of its cluster, it will forward the route request to all the border nodes B_X's of its cluster. Now, because S knows optimal paths to all these border nodes via the intra-cluster link-status updates, it will route these requests via the optimal paths (which will be routed via the sequence of core nodes $\{C_{X,1}, C_{X,2}..., B_{X,1}\}$). Thus, the globally optimal path containing border node $B_{X,1}$ is not lost at this

stage.

2) Next, all border nodes of cluster X will relay the request to adjacent clusters *i.e.* to the border nodes of the adjacent clusters which are their one-hop neighbours. One of these relayed messages reaches $B_{Z,1}$ directly from the node $B_{X,1}$. Thus, the optimal sequence is still intact.
3) After receiving the request, node $B_{Z,1}$ will forward it to all border nodes belonging to its cluster via optimal paths and also to all adjacent cluster border nodes which are its one-hop neighbours. Thus, due to the optimality of the $B_{Z,1}$ to $B_{Z,,2}$ segment, the sequence of nodes from $B_{Z,1}$ to $B_{Z,2}$ will be the same as that in the optimal path in network A. Again, the optimal path is still intact.
4) Step 3 will be repeated until the destination node cluster Y is reached. In Y, since $B_{Y,1}$ will already know the optimal path to D via the cluster routing-table protocol, it will forward the request to D via this optimum path.

Thus, the above arguments show that the the routing request will travel via the globally optimal path and reach the destination node in network B too, thereby enabling the selecting agent to choose this as the optimal route between the given source-destination pair.

In a nutshell, the above argument means that *only* and *all* those paths in which every segment is ‘locally optimal’ are chosen as candidates for the route in network B as compared to *all* paths in network A. Now, since the globally optimal path will consist of locally optimal segments, it has to be one of the candidates for the final selection.

Hence, *identical* optimal routes are computed in both network A and B.

Some points to keep in mind

1) The above result will hold for all metrics which will satisfy the classical dynamic programming technique which is used in the Viterbi Receivers and the Bellman-Ford algorithm. Validity in case of other metrics will have to be investigated.
2) The routing metric may be a traffic (bandwidth) related metric in which we want to find the best bandwidth route from S to D. Now since both A and B have different total overheads, hence traffic conditions will be different in both. Thus, the result that the same path is computed in both the networks may not hold in case of traffic related metrics. This needs to be investigated further and we do not concern ourselves with this in this work.
3) Also note that the traffic conditions in both the networks being different may lead to a sub-optimal route (w.r.t. the routing metric) being computed in A which will mean that the routes in A and B may not be the same. This is so because in case the route request via the optimal route gets delayed due to higher traffic conditions there, it is possible that an intermediate forwarding node does not forward this request further since it would have already received a request from a sub-optimal path which had a lower traffic at that point in time *i.e.* since each node forwards the routing request only once (as most protocols assume), the traffic conditions may lead to sub-optimal paths being computed. This actually amounts to the routing protocol failing in its purpose. This possibility can be taken care of in the design of the routing protocol. We can overcome this in at least two ways:-
   - A condition in most routing protocols is that each node forwards the routing packet only once for each computation of a route. We may wish to relax this assumption to make sure that all the possible candidate routes are collected at the destination node which then makes a decision about which route is the best. Though this may lead to more overhead, it will discount the possibility of a sub-optimal route being computed w.r.t. the routing metric. Thus, an optimal route is computed in A and thus, both the networks have the same route.
   - The route selection can be done at intermediate nodes as well with each node waiting for some time to collect packets corresponding to the same route request before forwarding the best out of those to its neighbours. This will increase delays in route computation but will help in obtaining the optimal route.

   In both the above ways, we are assuming that the relative traffic delays between two probable routes a given network are not exorbitantly high thereby rendering any waits useless.
   The above two are only indicative methods, and the specific routing protocol may come up with better techniques to overcome the problem of sub-optimality w.r.t. route discovery due to traffic conditions. (Note that this sub-optimality has nothing to do with the comparison of networks A and B, but is rather an issue to be resolved by the reactive routing protocol itself).

ACKNOWLEDGEMENT

I would like to thank Prof. D. Manjunath for his unending help in discussing the problem. I would also like to thank Prof. R. K. Shyamasundar, for his constant encouragement and giving me an opportunity to work in this field.

REFERENCES

[1] C. A. Santivàñez, B. McDonald, I. Stavrakakis, and R. Ramanathan, "On the scalability of ad hoc routing protocols", *Proceedings of IEEE INFOCOM 2002*, Vol. 3, Jun 23-27, 2002.

[2] P. Jacquet and L. Viennot, "Overhead in Mobile Ad-Hoc Network Protocols", *INRIA Research Report 3965, Institut National de Recherche en Informatique et en Automatique (INRIA)*, France, Jun. 2000.

[3] C. E. Perkins, E. M. Royer, S. R. Das, and M. K. Marina, "Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks", *IEEE Personal Communications Magazine*, Vol. 8, No. 1, Feb. 2001.

[4] V. Rudoplu, and T. H. Meng,"Minimum Energy Mobile Wireless Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, Aug. 1999.

[5] R. Hekmat and P .V. Mieghem, "Degree Distribution and Hopcount in Wireless Ad-hoc Networks", *Proceedings of the 11th IEEE International Conference on Networks (ICON 2003)*, Sydney, Australia, Sep. 28-Oct. 1, 2003.

[6] S. Y. Ni, Y. C. Tseng, Y. .S. Chen, and J. P. Shen, "The Broadcast Storm Problem in a Mobile Ad Hoc Network", *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Washington, United States, 1999.

[7] A. B. McDonald and T. F. Znati, "A Mobility-Based Framework for Adaptive Clustering in Wireless Ad Hoc Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, Aug. 1999.

[8] C. R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 7, Sept. 1997.

[9] C. E. Perkins and E. M. Royer, "Ad-hoc On-demand Distance Vector Routing", *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications 1999*, 25-26 Feb. 1999.

[10] D. B. Johnson, D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks", *Mobile Computing, edited by Tomasz Imielinski and Hank Korth, Chapter 5*, pp. 153-181, Kluwer Academic Publishers, 1996.

[11] C. R. Perkins and P. Bhagwat, "Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for mobile computers", *Proceedings of ACM SIGCOMM*, Oct. 1994.