Journal of Universal Computer Science, vol. 13, no. 1 (2007), 32-55
submitted: 29/8/05, accepted: 21/6/06, appeared: 28/1/07 © J.UCS

An Adaptive Hierarchical Extension of DSR: The Cluster Source Routing

Farid Jaddi
(IRIT-ENSEEIHT, France
Farid.Jaddi@enseeiht.fr)

Béatrice Paillassa
(IRIT-ENSEEIHT, France
Beatrice.Paillassa@enseeiht.fr)

Abstract: Numerous studies have shown the difficulty for a single routing protocol to scale with respect to mobility and network size in wireless ad hoc networks. This paper presents a cluster-based extension of the DSR protocol called Cluster Source Routing (CSR)[1]. The proposed approach improves the scalability of DSR in high-density and low-mobility networks. The originality of our proposal is an adaptive use of DSR and CSR routing modes according to network density and node mobility in order to produce less overhead and perform efficient routing. Indeed, adaptation is a key feature for a routing protocol since network dynamics can suddenly and widely change in wireless ad hoc networks. Thus, the DSR-CSR protocol achieves enhanced performance over a broader {network density, node mobility} domain as shown by simulations.
Key Words: wireless networks, ad hoc routing, clustering, adaptation, scalability.
Category: H.3.7, H.5.4

1 Introduction

An ad hoc network is a collection of wireless mobile nodes forming a temporary network without any fixed infrastructure. The applications of ad hoc networks range from conference meetings to disaster recovery (flood, earthquake, ...) and military operations. Though ad hoc networks are attractive (low-cost and quick deployment), they face several limitations due to the shared wireless medium. In particular, numerous studies have shown the difficulty for a single routing protocol to scale with respect to mobility, traffic load and network size [Broch et al. 1998] [Perkins et al. 2001] [Santivanez et al. 2002]. Flat routing protocols such as DSR (Dynamic Source Routing) [Johnson et al. 2004] and AODV (Ad Hoc On Demand Distance Vector) [Perkins et al. 2003] are efficient regarding to overhead minimization and packet delivery ratio in small networks and have been promoted by the IETF Manet working group [Manet]. However, their flooding-based route discovery involves multiple collisions due to broadcast. Besides, it becomes overhead-expensive on large networks configurations [Tseng et al. 2002].

[1] This work is supported by the NEWCOM Network of Excellence (european project)

Therefore, numerous hierarchical routing schemes have been designed to make ad hoc protocols more scalable such as Cellular-DSR, OLSR, dominating-set based and clustering-based schemes. Basically, the main principle is to select nodes which have extended routing functions to enhance routing efficiency. Cellular-DSR is an extension of DSR designed for a hierarchical hybrid network architecture [Jetcheva et al. 2003]. The scalability of DSR is improved through the partitioning of the network. However, C-DSR relies on the use of fixed base stations which prevents a pure ad hoc deployment. The Optimized Link State Routing (OLSR) protocol improves the classical link state routing algorithm [Clausen and Jacquet 2003]. Nodes named multipoint relays (MPRs) are selected to perform efficient flooding and to minimize link state information exchanges. Hence, OLSR minimizes the control overhead, especially on large and dense networks. Connected Dominating Sets (CDS) algorithms aim to form a virtual backbone to facilitate routing and reduce flooding effects [Wu and Li 1999] [Stojmenovic et al. 2002] [Dai and Wu 2003]. Each node either belongs to the CDS (it is said to be dominant node) or is a neighbor of a dominant node. CDS algorithms improve energy comsumption and decrease MAC collisions due to broadcast.

Clustering algorithms set up and maintain a hierarchical scheme which improves the network management. They are based on the partition of the network into logical groups. A cluster head is selected in each group. Then, cluster heads are interconnected by gateways (nodes which belongs to more than one group). A CDS is thus formed by cluster heads and gateways. Clustering schemes mainly differ on the cluster heads election criterion, the clusters size, and the cluster heads functions. The Linked Cluster Architecture (LCA) selects cluster leaders on the node identifiers (ID) [Baker et al. 1981]. Such an algorithm is quite simple to implement but node ID does not ensure that the most suitable nodes, in terms of mobility or connectivity, are elected as cluster heads. To reduce the number of clusters, [Parekh 1994] selects the most connected nodes as cluster leaders (i.e nodes which have the highest number of neighbors). The DCA/DMAC algorithm generalizes these approaches: it uses a weight as the election criterion [Basagni 1999]. The weight definition varies according to the application and combinates different parameters such as IP address and transmission power. [McDonald and Znati 1999] choose the node mobility, expressed by the path availability between cluster members, as its criterion to increase the clusters lifetime. All these protocols form 2-hop-diameter clusters and cluster members are 1-hop away from the cluster head. The algorithm described in [Amis et al. 2000] forms d-hop clusters and requires two stages composed of d rounds of flooding in the d-hop neighborhood to select the cluster head. Such proposal provides larger clusters but is overhead-expensive and quite complex to set up. Once the architecture is set, nodes perform the hierarchical scheme.

Adaptive Routing using Clusters (ARC) protocol distinguishes intra and inter cluster procedures [Belding-Royer 2002]. A clustering hierarchy is built in the ARC protocol and routing operations are performed in combination with an on-demand protocol like AODV.

Flooding-based protocols such as DSR or AODV are well adapted to small networks while hierarchical schemes provide benefits on larger networks. Since the network size of Manets can suddenly vary through networks merging and partitioning, combining the benefits of both approaches according to networks conditions will increase the routing performance. There are high-mobility (emergency services or troops deployment) and low-mobility (battlefield protection, medical care for the wounded) phases in numerous ad hoc applications such as military operations or disaster recovery. In these conditions, a cluster-based protocol performs efficient routing during low-mobility periods while a flooding routing is suitable to high-mobility periods. Therefore, we propose to adapt the routing mode to network dynamics.

The Cluster Source Routing (CSR) extension we have developed aims to increase the scalability of DSR with regard to network size and node mobility. The point of our proposal is to transfer the Route Discovery procedure to an upper level in a clustering architecture: the upper level cluster leader acts as a central Route Cache and Route Discovery is performed through unicast communications between cluster leaders. Thus, the network flooding due to the DSR Route Discovery is avoided. The main originality of our proposal is that each node can autonomously move from one mode to another ($DSR \leftrightarrow CSR$) according to its adaptation criteria (node mobility and node density). Besides, an upper level cluster leader has a global knowledge of the network: it can provide benefits in QoS issues such as load balancing.

The remainder of the paper is organized as follows. The computation of adaptation criteria is considered in Section 2. Section 3 introduces the CSR extension and describes its routing mechanisms while Section 4 presents CSR clustering procedures. Simulation results are exposed to validate the CSR extension. Finally, Section 5 concludes the paper.

2 Adaptation

In ad hoc networks, protocols have to adjust their behavior to various network conditions. In [Obaidat et al. 2004] and [Qingwen et al. 2004], link quality and energy conservation are considered to improve the protocol performance. A routing example of adaptation is the Independent Zone Routing protocol [Samar et al. 2004]. IZR is an hybrid protocol which improves the well known ZRP [Haas 1997]. A Routing Zone, whose radius can exceeds one hop, is defined for each node. A mobile station uses a proactive protocol to be aware of its zone topology. Inter-zones routing is achieved on demand. In IZR, each node dynamically adapts its zone radius by monitoring received control traffic in order to reduce control overhead. Therefore, an ad hoc routing protocol which can adapt not only its variables but also its routing algorithm to the network dynamics will outperform classical routing schemes.

From the CSR point of view, it appears that an upper level of Route Cache is useful as the node density grows. This trend is confirmed by simulation in Section 3.4. The required overhead to perform a DSR Route Discovery grows much faster versus density than the CSR one. Moreover, simulation results show that the set up of the clustering architecture is more interesting with regard to overhead reduction on dense network conditions (see Section 4.4). Thus, the node density appears as a pertinent adaptation criterion. Mobility also influences the CSR scheme because high mobility actually implies much stale information in Route Caches and requires consequent overhead in order to maintain the architecture. In such conditions, CSR will not perform well and nodes must pass into native DSR mode. Hence, mobility must also be considered as an adaptation criterion. As a result, we defined our adaptation criterion as a combination of mobility and density.

In our architecture, the local mobility and density estimations are necessary for *0-cell* cluster set up. The global mobility estimation is required to switch between modes. In order to evaluate adaptation criteria, various mobility and density metrics are available. These metrics can be computed by using periodic emission for each node. For example, the link duration exposed in [Boleng et al. 2002] provides a local mobility and density estimation. The local density is estimated by the number of neighbors. The duration of a link is defined as the time that two nodes are within transmission range of each other. For each node, the metric is obtained by averaging individual link durations experienced with all neighbors. The link duration succeeds in capturing nodes mobility and providing density metric but generates some additional overhead. The estimation of local mobility can also be calculated by using a quantity, introduced by [Jiang et al 2004], which predicts the link status for a period, T_p, in the future. Based on exponentially distributed epochs model, the probability $L(T_p)$ that the link may really last to the end of T_p is calculated by each node which forwards a packet. The implementation of such a metric only requires a 1-byte field, which can be transmitted through the DSR option mechanisms. The pair $(T_p, L(T_p))$ is used to adapt the link caching time-out in DSR but can also be used to make our routing scheme more adaptive. However, this estimation relies on a specific mobility model and its performance highly depends on the network mobility conditions. We propose the use of routing information such as overhead routing traffic and Route Cache to estimate mobility and density. Although this method is less precise than link duration, it is a more economic way to obtain adaptation criteria in terms of overhead diminution. The Route Cache can provide an estimation of the number of neighbors and thus a density estimation can be deduced from stored routes. Link caches are more efficient than path caches and should be preferred in most of DSR implementations [Hu and Johnson 2000][Johnson et al. 2004]. An estimator of link duration can be obtained without any additional overhead from the lifetime of each cached link.

The number and frequency of DSR Route Errors, Route Requests and Route Replies heard by nodes reveal some global mobility in the network. Global mobility could also be deduced from Route Cache. The Link-MaxLife Cache described in [Hu and Johnson 2000] maintains a Stability Table containing the observed stability of each node stored in cache. Thus, each node can calculate a global network mobility by averaging stabilities of its Stability Table. In [Altalhi and Richard 2004], a mobility indicator is obtained through the monitoring of broken paths and Route Cache management. With regard to our proposal, nodes can switch from DSR to CSR mode if the network stability and the local density are sufficient.

Mobility and density estimations are also employed to adjust timers values (Route Request timer, Route Cache lifetime, . . .) of the CSR procedures which are presented as follows.

3 CSR routing procedures

3.1 The clustering model

The extension procedures we are going to present are totally transparent and ensure full compatibility between native DSR and DSR-CSR nodes. In fact, the DSR packet format is conserved. Native DSR and DSR-CSR nodes can communicate since CSR integrates the DSR protocol. The CSR procedures are carried out through the DSR option mechanisms. Appropriate option codes are chosen to allow native DSR nodes to treat packets if necessary.

The network is partitioned according to a 2-level hierarchical architecture (Figure 1) [Jaddi and Paillassa 2004] [Jaddi and Paillassa 2005]. The lower level of cluster is the cell (*0-cell* cluster). Each node within the cell is 1-hop away from the Cluster Head. Communication between *0-cell* clusters is completed through gateway nodes. The upper level of cluster (*1-server* cluster) is formed by a set of cells. The associated cluster leader is named Server. Each node can have four status:

- *Undefined*: the node has not yet obtained a valid status and is running the native DSR protocol.
- *Node*: a station which can use the CSR mode.

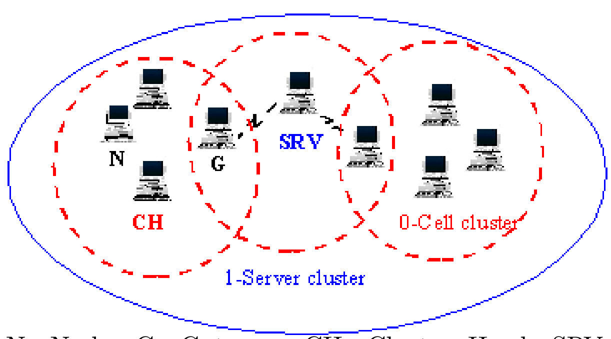
– *Cluster Head*: the cluster leader of the *0-cell* cluster.

– *Server*: the cluster leader of the *1-server* cluster. Its routing information is stored in two tables:

- *Mobile-Cluster Head Table (MCH)*: the localized mobile nodes are classified by cells (identified by Cluster Heads).
- *Cluster Head-Cluster Head Table(CHCH)*: it is a link-cache which indicates routes between Cluster Heads.

Server can be used to implement QoS mechanisms in the network such as path selection or load balancing. On receiving a CSR Route Request, Server can provide the source with different distinct paths to reach the destination. This mechanism can be used whenever such paths exist. In case of route failure, the source node will have another independent route to the destination.

Server has a global view of the network traffic since it handles all CSR Route Requests. Manet routing protocols usually consider that the best route between two nodes is the route with the minimum number of hops. Server can record which routes are already used and can set up a load balancing mechanism. When different routes between a source and a destination are available, Server selects the least loaded path instead of the minimum hop one.

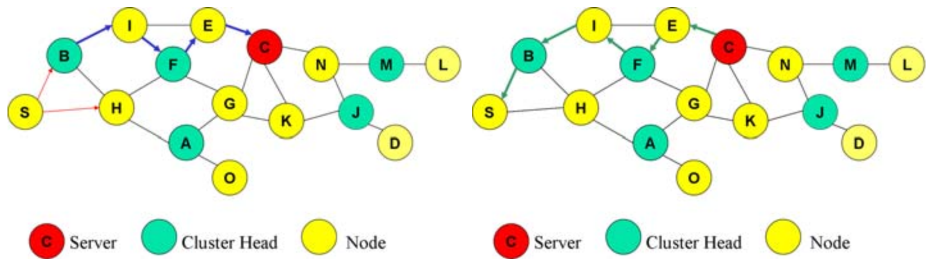


N: Node, G: Gateway, CH: Cluster Head, SRV: Server

Figure 1: Csr model

3.2 CSR Route Discovery

CSR improves the DSR Route Discovery mechanism in term of required overhead. In fact, Route Discovery relies on the cluster architecture in order to avoid



(a) CSR Route Request (b) CSR Route Reply

Figure 2: CSR Route Discovery

the network flooding.

When a node has a packet to send, it first searches its Route Cache for a suitable route to the destination. If the node does not have such a route, it initiates a Route Discovery by locally broadcasting a Route Request in its cell according to the non-propagating Route Request optimization of DSR (Time To Live=1). The Cluster Head of the cell checks its own cache and replies if a path to the destination is found. Native DSR nodes and CSR nodes can also reply if they have such a route. If no route is known, the Cluster Head forwards the Route Request to the Server (Figure 2(a)). The Cluster Head has obtained the path to Server through the Topology Discovery procedure (see Section 4.2.3). The Server checks whether the targeted destination is already localized. If so, it builds the path between the source and the destination by using its topology knowledge and sends it back to the source node within a Route Reply (Figure 2(b)). Otherwise, Server questions all the Cluster Heads it is aware of to find the targeted destination. Each Cluster Head looks for the destination in its cell by sending a non-propagating Route Request. When a Cluster Head localizes the destination, it sends back a positive reply to the Server.

The Server updates its topology information and replies to the source node. If it cannot localize a searched destination, it sends back a Route Error packet to the source to indicate an unreachable destination. On receiving this Route Error from the Server, the node initiates a DSR Route Discovery.

3.3 CSR Route Maintenance

Route Error is detected from the native DSR Route Maintenance. When a Cluster Head forwards a Route Error, it checks if it is the last Cluster Head which has forwarded the data packet. If so, it applies the Registration procedure (Route Error is piggybacked in the Registration packet) to perform a local repair. Thus, Server updates its routing information and removes any broken link from its

CHCH table.

On receiving a DSR Route Error packet during a communication, the source node initiates a CSR Route Discovery if the architecture is operational and if it does not have any other route. The DSR Route Error is piggybacked to the Server in the CSR Route Request. Server removes the broken link from its table if necessary and performs a new CSR Route Discovery.

3.4 Evaluation

We use the QNAP2 simulation environment to check if the CSR main principles (Route Discovery optimization by using an upper level of Route Cache) provides significant improvement. QNAP2 modeling permits to obtain quick results.

3.4.1 Simulation model

QNAP2 is a simulation tool based on the queue theory [Veran and Potier 1984]. Traffic sources are constant bit rate (CBR). Mobility of the destinations is taken into account by a movement probability. The station is reachable according to this probability. We calculate a 95% confidence interval over 10 simulations for all the observed performances criteria (relative intervals are less than 1%). A connectivity degree is defined, representing the number of neighboring Cluster Heads each Cluster Head can reach. We use two topologies for our evaluation. The low connectivity topology has an average connectivity degree equal to 2. The high connectivity degree has an average connectivity degree equal to 4. The node density is defined as the number of nodes per *0-cell* cluster. We run simulations with a nodes density set to 3, 4 and 5. Each simulation lasts 180000s. All the links are considered bidirectional.

To compare native DSR and DSR-CSR modes, we choose the following metrics:

- *Routing overhead*: the number of control packets which are transmitted.
- *Data*: the number of delivered data packets which are transmitted.
- *Efficiency*: defined by $\frac{Data}{Data+Overhead}$

Measurements have been performed in number of packets since DSR and DSR-CSR have nearly the same packet format. For multi-hop communication, each transmission of the packet is taken into account.

3.4.2 Results

Different network dynamics affect the control overhead of each routing Discovery procedure. Among them, the node density and the number of different paths in the network are key characteristics. To highlight their influences, we run simulations on different topologies with several node densities. The average connectivity degree of a topology gives an idea of the total number of paths in the network. For each topology, the connectivity degree remains constant while the density is varying. Figure 3 illustrates the DSR-CSR/native DSR efficiency ratio versus mobility on two different topologies and for three node densities. On the low-connectivity topology, at low density, the native DSR slightly produces less overhead than the DSR-CSR mode. However, the CSR mode becomes more efficient when the node density increases. This improvement is more significant on the high-connectivity topology. The native DSR Route Discovery mainly depends on the total number of nodes of the network n. The required control traffic overhead per Route Request is roughly $O(n)$. As for CSR mode, its Route Discovery procedure overhead is a function of the number of clusters and the average distance in number of hops between the *1-server* Cluster Head and a station. The low-connectivity topology contains few distinct paths. Thus, at low density, native DSR and DSR-CSR have almost equal efficiency. However, the benefit of DSR-CSR grows with node density because the number of stations increases while the connectivity degree remains constant. The influence of the number of distinct paths is shown by the results on the high-connectivity topology where the difference between the two modes is more marked. As for low-connectivity topology, this difference increases with node density. With numerous distinct paths, the flooding approach is much more expensive in terms of control overhead. As expected, the DSR-CSR mode is gainful on dense network configuration.

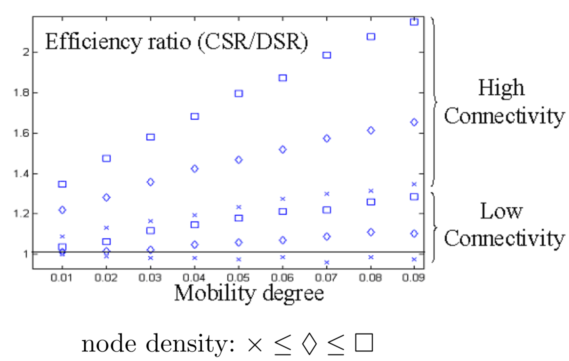


Figure 3: Efficiency ratio on two topologies for different node densities

4 CSR clustering procedures

The criterion used to elect the cluster leaders on each level is a combination of a mobility metric and a density metric (see Section 2). Thus, the least mobile and most connected nodes are selected. The Route Request and Route Reply packets are used to perform clustering procedures. CSR information is conveyed by using DSR option mechanisms.

4.1 *0-cell* cluster

Each node must obtain a status (Node or Cluster Head) to proceed with the CSR mode set up. Each Node is managed by a Cluster Head which forwards its Route Requests to the Server.

4.1.1 The GetStatus procedure

The set up of cells is based on the highest-connectivity degree algorithm, although the election criterion based on mobility and density degrees is preferred [Parekh 1994]. When a node enters CSR routing mode, it initiates the GetStatus procedure. Nodes which do not belong to a cluster are called uncovered nodes. To obtain a status, each uncovered node locally broadcasts a Route Request which contains its election criterion and indicates its undefined status (a Status packet). A specific option code is used to prevent neighboring native DSR nodes from processing the packet [Johnson et al. 2004]. Once the Status packet is broadcast, the node waits for a GetStatus period. If a packet from a Cluster Head is received before GetStatus expires, the node initializes its status to Node. Else, on receiving a Status packet, the node checks its routing mode:

- CSR mode: it compares the packet election criterion with its own.
- DSR mode: it checks its adaptation criterion. If its criterion is suitable enough to switch to CSR mode, the node starts the GetStatus procedure.
- native DSR: it just discards the packet (unknown option code).

If the node has the local highest criterion (the lowest ID is preferred in case of tie), it sets up its status to Cluster Head and broadcasts a Cell Maintenance packet indicating its status. Thus, its neighbors take the Node status and stop their GetStatus procedure. If a node does not have the local highest criterion and does not hear any Cluster Head, it becomes itself Cluster Head at the end of the procedure.

4.1.2 Cell maintenance

Periodically, each Cluster Head locally broadcasts a Cell Maintenance packet to maintain its cell. If Node has not heard any Cluster Head during a Status period, it applies again the GetStatus procedure. The selected Cluster Head revocation algorithm is LCC (Least Cluster Change) in order to control the number of Cluster Heads [Chiang et al. 1997]: when two Cluster Heads are within transmission range, the lower-criterion one gives up its role and becomes Node. Thus, Cluster Heads are at least 2-hop away.

4.2 *1-server* cluster

The 1-server cluster formation (Cluster Head Registration and Server election) and maintenance (Topology Discovery, multiple Servers) are described.

4.2.1 Cluster Head Registration

Once a node became a Cluster Head, it has to register to the Server. Thus, the Cluster Head checks the availability of Server and obtains a path to reach Server. Server can also localize the Cluster Head with respect to its neighboring Cluster Heads.

The Cluster Head broadcasts a Registration packet to reach its neighboring Cluster Heads (Figure 4). The TTL is set to three because neighboring Cluster Heads are at most 3-hop away from each other. When a registered Cluster Head receives a Registration packet, it forwards the packet to the Server (DSR Route Reply). Unregistered Cluster Heads just discard the Registration packet. Server updates its tables from each Registration packet but only replies to the first one. Thus the Cluster Head has a route to the Server. In fact, from each Registration packet, Server learns a new link between Cluster Heads and can fill in both its CHCH and MCH tables. The lack of Server availability is detected after three unsuccessful registration tries. Then, the Cluster Head initiates a Server Election.

4.2.2 Server Election

The Server is elected among Cluster Heads and selected on the election criterion. At the beginning of the procedure, Cluster Heads initialize their candidate criterion variable with their election criterion value and the candidate address variable with their own address. Each Cluster Head which initiates an election broadcasts an Election packet in its 3-hop neighborhood. This Election packet is a DSR Route Request which contains the election criterion of the Cluster Head. The Election packet is forwarded like a DSR Request by nodes (CSR and/or

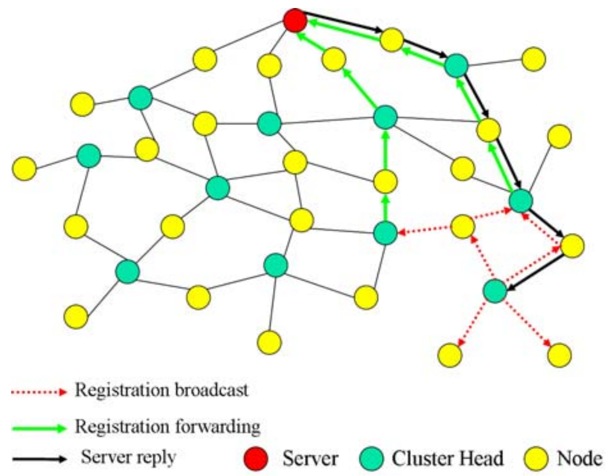


Figure 4: Cluster Head Registration and Topology Discovery

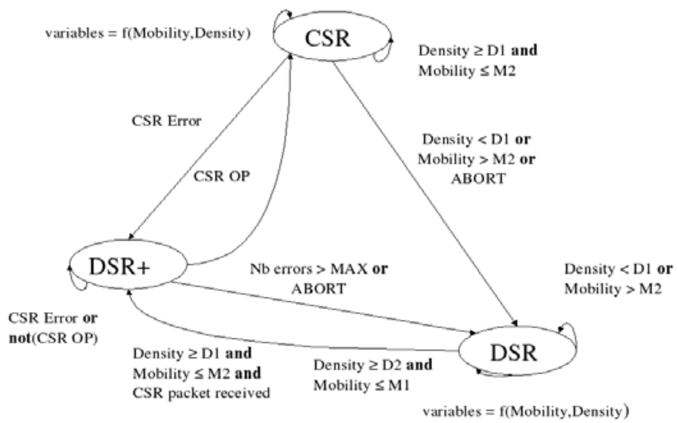
DSR) and only treated by Cluster Heads. When an Election packet is received, the packet criterion is compared with the candidate criterion. If the packet criterion is better, Cluster Head resets its election timer and records the candidate address, the candidate criterion and the route to reach it. The Cluster Head appends its address in the record listing, sets the TTL to three and re-broadcasts the Election packet to its neighboring Cluster Heads. This functioning ensures that the best Election packet reaches all the connected Cluster Heads. On receiving a packet with a criterion worse than its candidate criterion, a Cluster Head just discards the packet. At the end of the election (detected on timer expiration), the best candidate declares itself as the Server and each Cluster Head registers to the new Server. If the network is connected, the best Cluster Head is elected as Server since all Cluster Heads have received its election packet. Else, a Server is elected in each partition of the network.

4.2.3 Topology Discovery

Periodically, each Cluster Head applies the Registration procedure (see Figure 4). Thus, Server can update its tables and maintain routes between Cluster Heads. This procedure is also used to recover a valid route when a link failure is detected in the path to Server. The Registration timer is adapted from the local mobility estimation. A flag in the Registration packet indicates whether the Cluster Head has to wait for a Server Reply or not.

4.2.4 Multiple Servers

In case of network partition, each new network will set up a CSR architecture if the {mobility, density} conditions are favorable. If two CSR networks merge, one Server has to give up its role. Periodically, each Server broadcasts a maintenance packet by using the DSR flooding procedure. On receiving a maintenance packet from a higher-criterion Server, a Server gives up its role. It then becomes Cluster Head and registers to its new Server. Cluster Head will register to the higher-criterion Server in case of Server change.



CSR OP: operational CSR mode

Figure 5: CSR states

4.3 CSR robustness

In case of error, nodes recover the CSR architecture through CSR clustering procedures. Cluster Head detects that Server is unreachable during CSR Route Discovery or Registration procedures. It then performs the election procedure. Node detects the Cluster Head is unreachable during CSR Route Discovery or Cell maintenance procedures. It then performs the GetStatus procedure.

A node switches to DSR mode if it experiences more CSR errors than the predefined value MAX. The MAX value is set to 3 in our simulations. CSR errors are caused by failures in setting up the architecture. On receiving an ABORT packet from the Server, a node also switches to DSR mode. Server sends an ABORT packet when it is about to give up its role.

Mobility and density metrics can be considered in an individual or combined way. In our simulations, we separately used the mobility and density metrics to perform the mode switching. The number of Route Errors is selected as the mobility metric and the number of neighbors in the route cache provides the density metric. Metrics are periodically computed. The computation timer value is set to 6s. The mode change is detailed below. Each node which runs DSR-CSR protocol could experience three states (Figure 5):

- *DSR*: The node uses DSR Route Discovery and DSR Route Maintenance. If network dynamics are favorable (high density and low mobility), the node enters *DSR+* state. Two thresholds of mode switching are defined for both the Mobility metric ($M1 < M2$) and the Density metric ($D1 < D2$):

  - $Mobility > M2$ or $Density < D1$: the node stays in *DSR* mode (high mobility and/or low density).

  - $Mobility \leq M2$ and $Density \geq D1$: the node changes from *DSR* to *DSR+* mode if it receives a CSR packet (average mobility and density).

  - $Mobility \leq M1$ and $Density \geq D2$: the node switches from *DSR* to *DSR+* mode (low mobility and high density).

  The simulation results we present in Section 4.4.3 have been obtained with $M1 = 2$, $M2 = 4$, $D1 = 2$ and $D2 = 5$. Thresholds values could be adjusted through extensive experiments.

- *DSR+*: The node uses DSR Route Discovery and Route Maintenance. However, clustering procedures are used to set up or to recover the CSR architecture. If clustering procedures succeed, the node enters CSR state. Else, it goes into *DSR* state. After its election, Server sets a timer and waits for Cluster Heads Registration to obtain its routing information. On timer expiration, Server is operational. When Server sends back a Registration Reply, it indicates to Cluster Head whether the CSR architecture is active or not. During *DSR+* state, Server could stop *CSR* mode by broadcasting an ABORT packet in the network (for example, few registered Cluster Heads indicating a low global density) and pass into *DSR* mode.
  After its election, each Cluster Head registers to the Server. If the Registration procedure fails, Cluster Head will initiate a Server election. On MAX election failures, Cluster Head passes into *DSR* state. When Cluster Head receives a Registration Reply, it checks whether the CSR mode is operational or not. If so, it enters the *CSR* state and signals operational CSR mode to its cluster members through each Cell Maintenance packet. Else, it sets a timer and only goes into *CSR* state on its expiration. On receiving an ABORT, each Cluster Head switches to *DSR* state.
  On receiving a Cell maintenance packet, the node checks whether the CSR

is operational (use of CSR Route Discovery) or not (use of DSR Route Discovery). Each Node switches to *DSR* state if it receives an ABORT.

– *CSR*: Node uses CSR Route Discovery and Route Maintenance. CSR mode is operational and Cluster maintenance procedures are applied. Server broadcasts an ABORT message when it is about to give up its role because of network dynamics and switches to *DSR* state. If Server receives a packet from a higher criterion Server, it becomes Cluster Head and enters the *DSR+* state. If Server is unreachable, the Cluster Head locally broadcasts a Cell Maintenance packet indicating to its cluster members that CSR architecture is not operational. Then, it applies the Registration procedure and switches to *DSR+* mode. On receiving an ABORT, Cluster Heads and Nodes switch to *DSR* state.

4.4 Evaluation

The overhead introduced by clustering procedures is thoroughly evaluated on ns2 to determine on which network conditions the CSR mode is valuable. We also study the aggregate overhead of CSR in procedures of both routing and clustering. The CSR performance is compared to AODV and DSR ones.

4.4.1 Simulation model

We used the ns2.26 distribution to evaluate clustering procedures [NS]. We used a TwoRayGround propagation model and a 802.11 mac layer with standard values. The transmission range is set to 150 meters. The duration of all simulations is 1000 seconds. We conduct simulations for 50, 100 and 150 nodes in a 1000 x 1000 simulation area. Each point has been calculated over 10 different mobility scenarios.

4.4.2 Validation of CSR Clustering

The study focuses on the control overhead introduced by the formation and the maintenance of the architecture. The Random Waypoint mobility model is used with a pause time of 30 seconds and a maximum speed varying from 1 m/s to 20 m/s. The overhead is measured in number of control packets per node. Figure 6 shows the *0-cell* overhead per node as a function of the node speed. The overhead is evaluated for three densities (50, 100 and 150 nodes). Firstly, we noticed the overhead per node remains nearly constant as a function of mobility whatever the density. Thus, the CSR cluster architecture is stable and its overhead is controlled with regard to mobility. Secondly, the *0-cell* overhead significantly decreases as the density increases (50% decrease from 50 to 100 density, 20%

decrease from 100 to 150 density). As expected, the CSR architecture requires less overhead as the node density increases.

This trend is also apparent during the *1-server* election procedure. Figure 7 presents the overhead per node during the Server election versus node density. It shows the election overhead over an average mobility ($Vmax = 10m/s$). The required overhead for Server election decreases as the density increases. Such results confirm that the Server election generates less overhead on high-density configurations. We have computed overhead under different mobility conditions and have obtained similar results. It appears that the Server election is performed quickly with regard to mobility (two times the duration of a network flooding in the worst case). The node mobility does not significantly affect results.

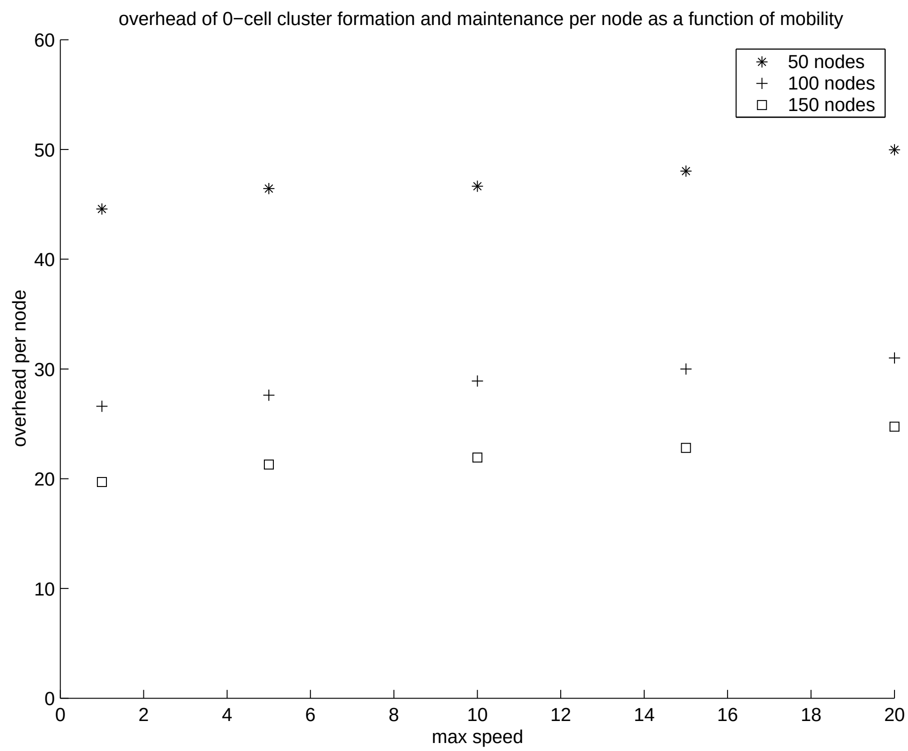


Figure 6: Overhead per node for *0-cell* procedures

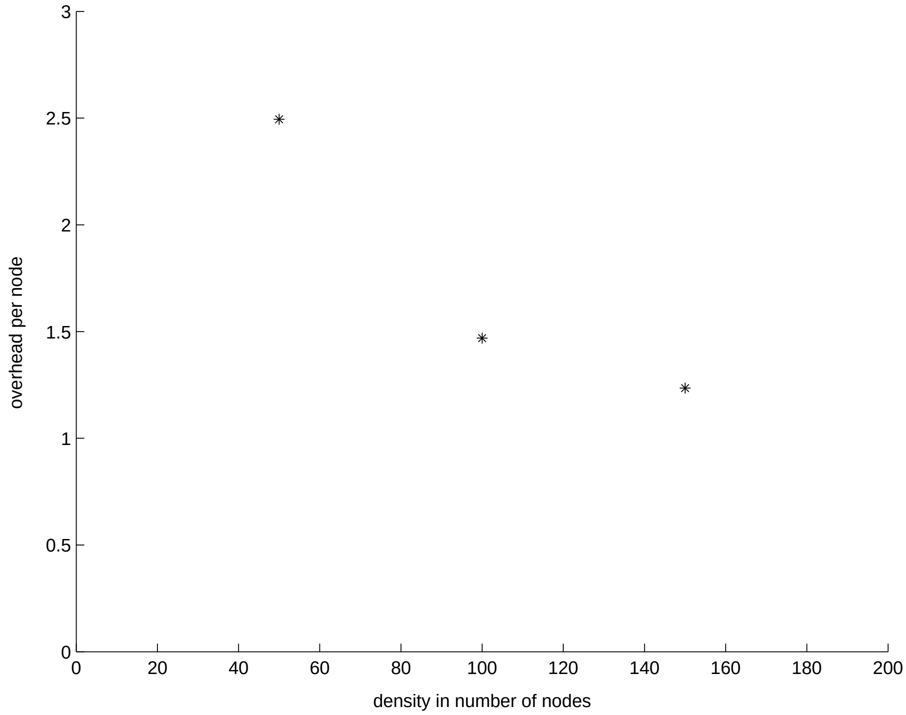


Figure 7: Overhead per node during a Server election

4.4.3 Performance Comparison

We compare AODV, DSR and CSR over different mobility conditions and data loads. The Random Waypoint mobility model is used with a pause time of 100, 200, 300, 400 and 500 seconds and a maximum speed varying from 1 m/s to 20 m/s. Traffic sources are Constant Bit Rate (CBR). The source-destination pairs are selected randomly. We run simulations for 10 and 20 traffic sources with a packet rate of 4 packets/s. We choose the following metrics to evaluate the relative performance of the protocols:

- *Routing overhead*: the number of transmitted control packets. For multi-hop communication, each transmission of the packet is taken into account.
- *Data*: the number of delivered data packets.
- *Normalized Data Load*: defined by $\frac{Data}{RoutingOverhead}$

Routing Overhead and Data are considered in terms of packets as the cost to gain access to the medium is more important than the byte transmission cost as in [Broch et al. 1998] [Perkins et al. 2001].

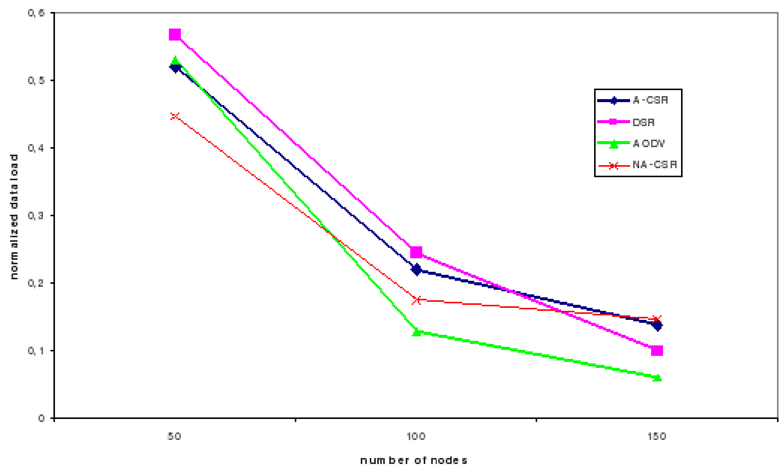


Figure 8: Normalized data load ratio with 10 connections

We first observe normalized data load as a function of node densities with 10 connections (Figure 8). We want to determine if the mode switching is valuable. Two versions of the CSR extension are considered. NA-CSR (Non-Adaptive CSR) implements all CSR features except the mode switching. A-CSR (Adaptive CSR) performs mode switching according to the mobility and density metrics. Each point is averaged over 15 scenarios with different pause times. For 50 nodes, NA-CSR performs poorly compared to other protocols. Network conditions are not suitable for a hierarchical routing protocol. A-CSR normalized data load is closer to DSR one. Simulations show that most of CSR nodes switch to DSR mode. However, some CSR nodes experiment CSR mode when their local network conditions are favourable to a mode switching. This additional overhead penalizes A-CSR. AODV performs better than A-CSR and NA-CSR for 50 nodes. AODV generates more overhead packets since nodes periodically broadcast Hello messages. Therefore, its normalized data load is less than the other protocols for 100 and 150 nodes. For 100 nodes, DSR protocol still has the best normalized data load and A-CSR performs better than NA-CSR. For 150 nodes, A-CSR and NA-CSR perform better than DSR since the network conditions are suitable enough for the hierarchical extension. CSR versions scale better with regards to node density. Besides, A-CSR globally outperforms NA-CSR. A-CSR

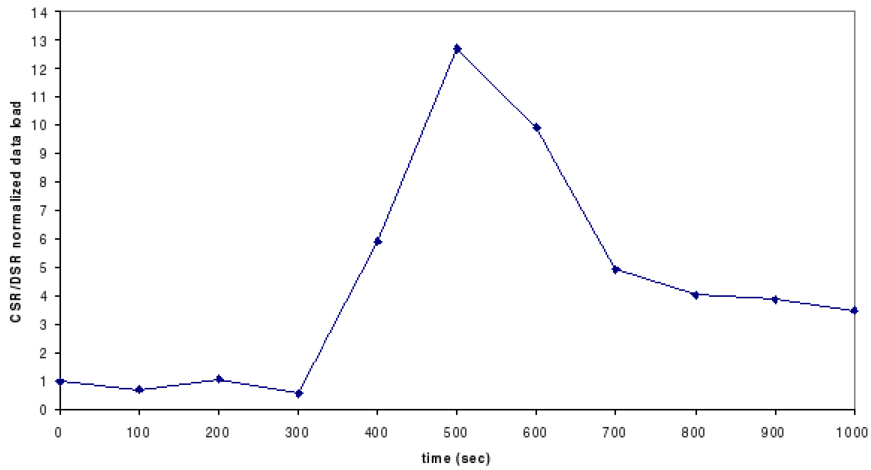
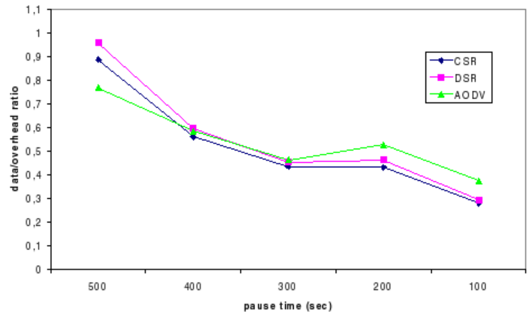


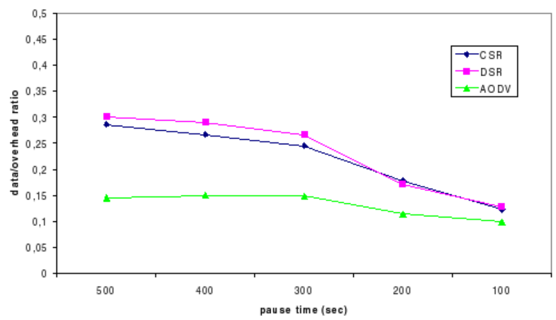
Figure 9: CSR/DSR normalized data load ratio pause time=300s 150 nodes

fits its behavior according to the node density by using the adaptive routing mode switching. A-CSR achieves the best overall routing performance in term of normalized data load. Later on, CSR extension designates the A-CSR one.

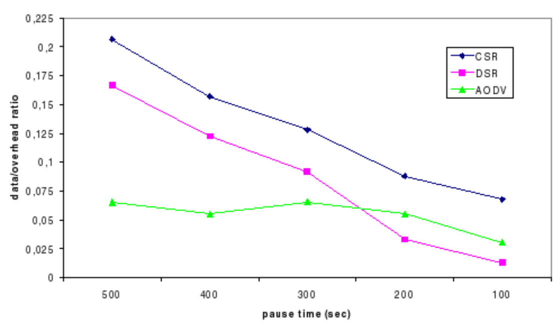
Figure 9 illustrates both DSR-CSR and native DSR behaviors in case of mobility. It shows the DSR-CSR/native DSR normalized data load ratio versus time for a mobility scenario. The network is composed of 150 nodes. The pause time is set to 300s and the maximum speed is 15 m/s. We used 10 traffic sources. During the first 300s, DSR slightly performs better than CSR. During this period, there is no mobility and each source uses Route Discovery procedure only once. Then, source nodes use their cached routes. Even if the node density is high, the cost of the CSR architecture set up and maintenance is higher than the overhead minimization obtained with the CSR Route Discovery procedure. The nodes movement at time=300s implies many DSR Route Discoveries. When there are simultaneous DSR Route Discoveries, the number of packets collisions is increased. After the mobility period is over, most DSR-CSR nodes switch to CSR mode and use CSR Route Discovery. The DSR-CSR/DSR ratio is above 6. The DSR-CSR extension becomes more efficient since the CSR Route Discovery benefit is larger than the clustering cost. The CSR gain decreases at t=600s and t=900s when the nodes move. These events involved a mode switching from CSR to DSR and then from DSR to CSR for many nodes. The CSR clustering architecture has to be set up again. However, the efficiency ratio is still around 4.



(a) 50 nodes



(b) 100 nodes



(c) 150 nodes

Figure 10: Normalized Data Load with 10 sources for various number of nodes

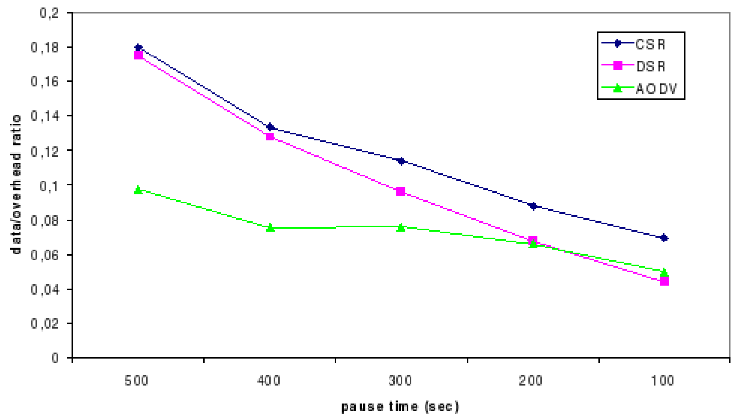


Figure 11: Normalized Data Load with 20 sources for the 100 nodes model

Figure 10 illustrates the normalized data load of AODV, DSR and CSR as a function of pause time. We used 10 traffic sources for this set of experiments. We first observe AODV data load is lower when the pause time is high. AODV generates control packets because each node periodically transmits Hello messages. Under low mobility conditions, Route discovery procedures are not widely used and this periodic control traffic penalizes AODV performance. However, as the mobility increases, the difference between AODV and the other two protocols decreases. For the 50 nodes configuration, AODV performs even better than DSR and CSR at high mobility (figure 10(a)). For the 150 nodes configuration, AODV becomes more efficient than DSR when the pause time is under 200s (figure 10(c)). Das, Perkins and Royer have obtained similar results: AODV is more scalable than DSR with regard to the number of nodes, the number of sources and mobility [Perkins et al. 2001]. DSR extensively uses route caching and this strategy is overhead expensive under high mobility conditions because a lot of cached routes become stale. For the 50 nodes configuration, DSR and CSR have the same behavior according to mobility even if DSR performs slightly better. The network conditions in term of density are not suitable enough to switch to DSR mode and most of CSR nodes stay in DSR mode. However, some CSR nodes try to switch to CSR mode because they experiment high local density. This behavior generates additional overhead especially under low mobility. The same trend appears for the 100 nodes configuration (figure 10(b)). However, the difference between DSR and CSR is reduced as the mobility increases because the node density is more suitable to switch to CSR mode and Route Discovery procedures are more frequently used. For the 150 nodes experiments, CSR produces a better data/overhead ratio even for low mobility. The node density is high and most of CSR nodes switch to CSR mode. This difference between DSR and CSR increases with mobility because Route Discovery procedures are extensively used. The benefits of the CSR routing optimization are far greater than the cost of the architecture set up and maintenance.

Figure 11 shows the influence of data load on the relative performance of protocols. When the number of sources increases, CSR performs better than DSR on the 100 nodes network. The performance difference between AODV and DSR also decreases when the number of sources is higher. With 10 connections, DSR had a better data/overhead ratio than CSR and AODV (see Figure 10(b)). This trend is reversed with 20 connections: CSR outperforms DSR and the improvement increases with the mobility. AODV performs better than DSR when the pause time is less than 200s. These results show that AODV and CSR scale better than DSR with respect to traffic load.

5 Conclusion

The paper has presented the Cluster Source Routing extension, a method to adapt the routing protocol to various conditions of mobility and density in wireless ad hoc networks. DSR and CSR perform best in different mobility, density domains and their combination enhances the global routing performance. Our proposal relies on a 2-level hierarchical scheme (*0-cell* and *1-server* clusters) which improves the scalability of the DSR routing protocol. Route Requests are forwarded to the *1-server* leader, considered as an upper level of Route Cache , to prevent network flooding. Then, data are transferred according to native DSR. Clustering procedures are defined to set up and maintain the CSR architecture. Each station individually adapts its routing mode (DSR or CSR) according to mobility and density. In fact, routing adaptation to network dynamics is a major benefit in wireless networks as network conditions can unpredictably fluctuate. Adaptation criteria computation methods are presented to enable the change between modes and to adjust the routing variables.

Simulations show the advantage of CSR in term of control traffic, especially in dense and low-mobility networks. The adaptive routing mode switching enables enhanced performance over a wider range of node densities.

References

[Altalhi and Richard 2004] Altalhi, A.H., Richard, G.G.: “Virtual Paths Routing: a Highly Dynamic Routing Protocol for Ad Hoc Wireless Networks”; Proc. of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, (Mar 2004), 81-86.

[Amis et al. 2000] Amis, A.D., Prakash, P., Vuong, T.H.P., Huynh, D.T.: "Max-min d-cluster formation in wireless ad hoc networks"; Proceedings of IEEE INFOCOM 2000, Vol.1, 32-41.

[Baker et al. 1981] Baker, D.J., Ephremides, A.: "The architectural organization of a mobile radio network via a distributed algorithm"; IEEE Transactions on communications, 29(11), (Nov 1981), 1694-1701.

[Basagni 1999] Basagni, S.: "Distributed clustering for ad hoc networks"; Proceedings of ISPAN, Australia (Jun 1999), 310-315.

[Belding-Royer 2002] Belding-Royer, E.: "Hierarchical Routing in Ad hoc Mobile Networks"; Wireless Communication &Mobile computing (2002), 512-532.

[Boleng et al. 2002] Boleng, J., Navidi, W., Camp, T.: "Metrics to Enable Adaptive Protocols for Mobile Ad Hoc Networks"; Proc. International Conference on Wireless Networks (ICWN'02), 293-298.

[Broch et al. 1998] Broch, J., Maltz, D., Johnson, D., Hu, Y., Jetcheva, J.: "A Performance Comparison of Multi-Hop Wireles Ad Hoc Network Routing Protocols"; Proc ACM/IEEE Mobicom 98, Dallas Texas (Oct 1998), 85-97.

[Chiang et al. 1997] Chiang, C., Wu, H., Liu, W., Gerla, M.: "Routing in Clustered Multihop, Mobile Wireless Network with Fading Channel"; Proc. of IEEE Singapore International Conference On Networks (SICON'97), Singapore (Apr 1997), 197-211.

[Clausen and Jacquet 2003] Clausen, T., Jacquet, P.: "Optimized Link State Routing Protocol (OLSR)"; IETF RFC 3626 (Oct 2003).

[Dai and Wu 2003] Dai, F., Wu, J.: "Distributed dominant pruning in ad hoc networks"; Proceedings of the IEEE ICC, Anchorage(May 2003), 353-357.

[Haas 1997] Haas, Z.J., Pearlman, M.R.: "The Performance of a New Routing Protocol for the Reconfigurable Wireless Networks"; Comm. ICC 98 (Jun 98), 156-160.

[Hu and Johnson 2000] Hu, Y., Johnson, D.: "Caching strategies in On-Demand Routing Protocols for Wireless Ad Hoc Networks"; ACM Mobicom 2000, Boston, (Aug 2000), 231-242.

[Jaddi and Paillassa 2004] Jaddi, F., Paillassa, B.: "A Cluster Procedure for the Dynamic Source Routing Protocol in Ad hoc Networks"; 3^{rd} Med-Hoc-Net 2004, Bodrum (Jun 2004).

[Jaddi and Paillassa 2005] Jaddi, F., Paillassa, B.: "An Adaptive Hierarchical Extension of the DSR: The Cluster Source Routing"; 1st ACIS International Workshop on Self-Assembling Wireless Networks, SAWN, Baltimore (May 2005),460-467.

[Jetcheva et al. 2003] Jetcheva, J.G., Hu, Y., PalChaudhuri, S., Saha, A.K., Johnson, D.B.: "Design and Evaluation of a Metropolitan Area Multitier Wireless Ad Hoc Network Architecture"; Proc. of the 5th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA), Pittsburg (Oct 2003), 32-43.

[Jiang et al 2004] Jiang, S., Liu, Y., Jiang, Y., Yin, Q.: "Provisioning of adaptability to variables topologies for routing schemes in MANETs"; IEEE Journal on Selected Areas in Communications, Vol.22 (7), Sep 2004, 1347-1356.

[Johnson et al. 2004] Johnson, D., Maltz, D., Hu, Y.: "The Dynamic Source Routing Protocol for Mobile Ad Hoc Network"; IETF Manet Working Group, Internet Draft, work in progress (July 2004).

[Manet] The MANET working group: `http:www.ietf.org/html.charters/manet-charter.html`

[McDonald and Znati 1999] McDonald, A.B., Znati, T.F.: "A mobility-based framework for adaptive clustering in wireless networks"; IEEE journal on selected areas in communications, 17(8), Aug 1999, 1466-1486.

[NS] ns-2 simulator: `http://www.isi.edu/nsnam/ns/`

[Obaidat et al. 2004] Obaidat, M.S., Green, D.G.: "SNR-WPA: an Adaptive Protocol for Mobile 802.11 Wireless LANs"; 2004 IEEE International Conference on Communications (Jun 2004), 3911-3915.

[Parekh 1994] Parekh, A.K.: "Selecting Routers in Ad Hoc Wireless Networks "; Proc. SBT/IEEE ITS (1994).

[Perkins et al. 2001] Perkins,C., Belding-Royer, E., Das, S., Marina, M.: "Performance Comparison of Two On Demand Routing Protocols for Ad Hoc Network"; IEEE personal communications (Feb 2001), 16-28.

[Perkins et al. 2003] Perkins,C., Belding-Royer, E., Das, S.: "Ad hoc On demand Distance Vector (AODV) Routing"; Internet RFC 3561, July 2003.

[Qingwen et al. 2004] Qingwen, L., Shengli, Z., Giannakis, G.B.: "TCP Performance in Wireless Access with Adaptive Modulation and Coding"; 2004 IEEE International Conference on Communications (Jun 2004), 3989-3993.

[Samar et al. 2004] Samar, P., Pearlman, M.R., Haas, Z.J.: "Independent Zone Routing: an Adaptive Hybrid Routing Framework for Ad Hoc Wireless Networks"; IEEE/ACM Transactions on Networking (Aug 2004), 595-608.

[Santivanez et al. 2002] Santivanez, C., McDonald, B., Stavrakakis, I., Ramanathan, R.: "On the Scalability of Ad Hoc Routing Protocols"; Proceedings of INFOCOM 2002, New York (Jun 2002),1688-1697.

[Stojmenovic et al. 2002] Stojmenovic, I., Seddigh, M., Zunic, J.: "Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks"; IEEE Transactions on Parallel and Distributed Systems, Vol. 13(1), Jan 2002, 14-25.

[Tseng et al. 2002] Tseng, Y.C., Ni, S.Y., Chen, Y., Sheu, J.: "The Broadcast Storm Problem in a Mobile Ad Hoc Network"; Wireless Networks,8 (2-3), 2002, 153-167.

[Veran and Potier 1984] Veran, M., Potier, D.: "QNAP 2: A Portable Environment for Queueing Networks Modelling"; Proc. Int. Conf. Modelling Techniques Tools Perform. Anal., Paris (May 1984).

[Wu and Li 1999] Wu, J., Li, H.: "On calculating connected dominating sets for efficient routing in ad hoc wireless networks"; Proceedings of the ACM International Workshop on Discrete Algorithms an Methods for Mobile Computing and Communications, Seattle (Aug 1999), 7-14.