**Sudan University of Science & Technology**

**College of Graduate Studies**

# Compare the performance of algorithms guidance in wireless networks that use infrastructure using a program Omnet ++

# مقارنة اداء خوارزميات التوجية في الشبكات اللاسلكية التي تستخدم بنية تحتية بإستخدام برنامج Omnet++

*In partial Fulfillment of the Requirements for the Degree of*

*Master of Sciences (MSc.) in(Computer Networks)*

Submitted By:

Farha Abd Algfar Ali

Supervisor:

Dr. Hassab Elgawi Osman

Khartoum ,sudan,2013

# *Abstract*

Since the topology of the mobile ad-hoc network (MANET) is constantly changing, the issue of routing packets between any pair of nodes becomes a challenging task. Most protocols are based on reactive routing instead of proactive. Multicast routing is another challenge because the multicast tree is no longer static due to the random movement of nodes within the network. Routes between nodes may potentially contain multiple hops, which is more complex than the single hop communication**.**

In this thesis some recent developed routing protocols are reviewed and compared using OMNeT++ (ver 4.3) Simulation. The ultimate goals are: to learn about each protocol for this kind of networks and to define and know how it works to avoid a particular problem within the network, and to find mechanisms leading to multi-hop ad hoc networks. For the comparative analysis protocols are divided into three groups: 1) Reaction Protocols (AODV & DSR ), 2) Proactive protocols (OLSR & TBRPF( , and 3) hybrid protocols.

In network routing, problems related to bandwidth and a flood between the node and its neighbors as well as the problem of reducing the vulnerability of the network in the transmission process are appeared. The goal here is to calculate the optimum route according to one or several restrictions (number of hops and bandwidth maximum and minimum of the end-to-end delay).

Results suggested that more delays in the protocols in the process is the
.(exchange of information (DSR

# المستخلص

منذ طوبولوجيا شبكة مخصصة النقالة (مانيه) تتغير باستمرار، فإن مسألة توجيه الحزم بين أي زوج من العقد تصبح مهمة صعبة. وتعتمد معظم بروتوكولات التوجيه على رد الفعل بدلا من استباقية. توجيه الإرسال المتعدد هو تحد آخر لأن شجرة الإرسال المتعدد لم تعد ساكنة بسبب الحركة العشوائية من العقد داخل الشبكة. يحتمل أن تحتوي على مسارات متعددة .القفزات بين العقد، الذي هو أكثر تعقيدا من الاتصالات قفزة واحدة

في هذه الأطروحة يتم مراجعة بعض بروتوكولات التوجيه المتقدمة الحالية ومقارنتها المحاكاة. الأهداف النهائية هي: لمعرفة كل بروتوكول لهذا النوع OMNeT ++ - 4.3 باستخدام من الشبكات وتحديد ومعرفة كيف يعمل لتجنب مشكلة معينة داخل الشبكة، وإيجاد الآليات التي تؤدي إلى تعدد هوب مخصصة الشبكات. تنقسم البروتوكولات  إلى ثلاث مجموعات: 1) البروتوكولات رد الفعل

(AODV DSR) ،بروتوكولات الاستباقية (2 TBRPFو OLSR) .و 3) البروتوكولات الهجينة

التوجيه في الشبكة، والمشاكل المتعلقة النطاق الترددي وظهر طوفان بين العقدة وجيرانها وكذلك الحد من مشكلة ضعف الشبكة في عملية الإرسال. والهدف هنا هو: معرفة البروتوكول الأمثل وفقا لواحد أو عدة قيود (عدد القفزات وعرض النطاق الترددي الحد الأقصى والحد الأدنى من التأخير من النهاية إلى النهاية). وبناء علي التجارب التي تم تطبيقها علي برنامج المحكاة(Omnet++ ) ومن خلال النتائج التي تحصلنا عليها نجد ان اكثر البروتوكولات تاخراً في عملية تبادل المعلومات هو(DSR).

# *Acknowledgements*

I take this opportunity to extend my sincere thanks to the family of Sudan University of Science and Technology and the school of Graduate Studies in particular. Special thanks and credits goes to Dr. Hassab Elgawi Osman who supervised this research and helped me in many directives and guidance.

Farha Abd Algfar Ali
Khartoum, Sudan 2015

# *Table of Contents*

# *List of Figures*

# *List of Tables*

# *Chapter 1*

## *Ad-Hoc Networking*

Ad hoc network works with the idea that wireless node within a network can communicate directly with each other, without the use of any network infrastructure. However, these networks are limited by the fact that each wireless node must be within transmission range of each other, and this leads to small networks as shown in Figure 1.1. One of the other major problems in ad hoc networks lays on how packets are move in and share with other nodes.

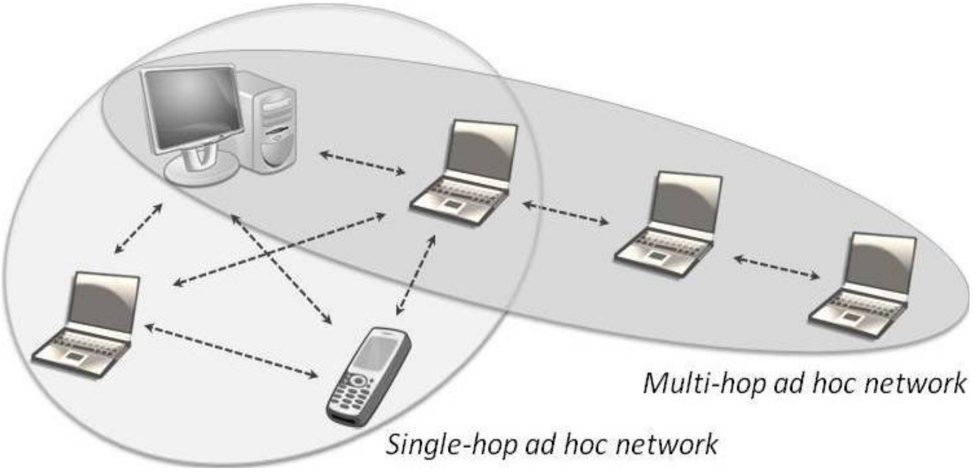


**Fig 1.1 limited range of wireless network transmission.**

Generally, there are two main link layer technologies used, however to connect a larger area requires an ad hoc multi-hop network architecture.

## 1.1 Historical Background of Ad Hoc Networking

Ad hoc networking is not a new technology, but has been developed more than 30 years ago. The word ad hoc derives from Latin and means *"for a particular purpose"*. Figure 1.2 summarizes the chronological development of ad-hoc networks.

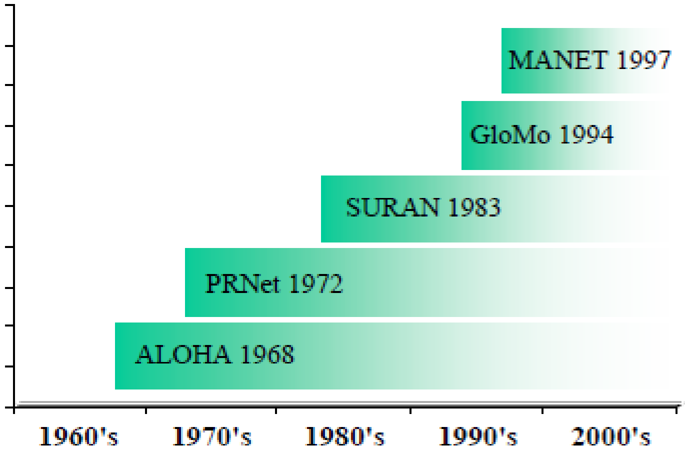


**Fig 1.2** Chronology of ad hoc Networks**.**

The origin of ad hoc networking can be traced back as far as to the ALOHA System project that was started in 1968. The ALOHA network was build to connect Hawaii university facilities with a prototype radio-linked-sharing network, but also in order to study computer communication using radio and satellites.

Based on the experience of the ALOHA network DARPA began the development of Packet Radio Network (PRNet) in 1972. Even though PRNet initially used at centralized control stations, it evolved quickly to work at distributed basis. On the routing protocol side, PRNet introduced the first proactive, multi-hop routing algorithm. Even though PRNet demonstrated the feasibility of ad hoc networking idea, there remained many major issues that could not yet be solved.

To extend the PRNet technology further, DARPA initiated the Survivable Adaptive Networks (SURAN) project in 1983. The project was designed to solve the detected problems that can be summarized to three concrete goals:

- To develop a small, low-cost, low-power radio that could support more sophisticated packet radio protocols.
- To demonstrate algorithms that could extend the network scalability to thousands of nodes.
- To develop some techniques that would increase networks robustness and make it survivable.

One of DARPA's latest development efforts is the Global Mobile (GloMo) project initiated in 1994.

In 1997 Mobile Ad hoc networks (MANET) were created to provide communication between peers without any network infrastructure. To improve the provided services on those networks, many quality of service (QoS) frameworks have been proposed to improve the performance of ad hoc networks. However, talking about QoSs is beyond the scope of this thesis.

## 1.2 Communications at Ad Hoc Networking

In ad hoc network, a node can successfully communicate with nodes within its transmission range. This is including nodes outside this range, but close enough to detect the signal and or are in the carrier sensing range. Such a network is formed dynamically and nodes organize themselves. The main restriction of an ad hoc network is its area, since any node has to be within the transmission range of any other node it communicates with. To extend the area of ad hoc network, )e.g., to enable communication between two distant nodes) intermediate nodes have to be inserted. This is the characterization of a multi-hop ad hoc network. Since nodes are free to move independently and in any direction delivering packets to a specific destination becomes a complex task due to frequent and unpredictable topology changes.

Despite promising in many applications these types of networks face challenges when it comes to issues related to connectivity, security, quality of service, energy consumption, and routing optimization. Indeed, not only do ad hoc networks inherit the classical problems of wireless networks (e.g. air medium propagation, unreliability, and hidden nodes), but also generate new problems and complexities (e.g. autonomy, lack of infrastructure, dynamicity, and scalability).

## 1.3 Research Problem

There are many kinds of protocols used in the ad hoc network, including reactive and proactive protocols. This research aims at comparing the performance of these protocols.  OmNet++ is used to simulate the network.

## 1.4 Organization of Forthcoming Chapters

This research consists of five chapters, where the first chapter Introduction to the Ad Hoc networks,

Chapter two talks about Mobile Ad-Hoc Network (MANET

Chapter threes presents the definition of the protocols and how they work

Chapter four provides description and analysis of routing protocols using omNet++ simulation program.

Chapter five summaries the research.

# *Chapter 2*

# ***Mobile Ad-Hoc Network (MANET)***

The mobile ad hoc network (MANET) is a network formed – by MANET working group - without any central administration. Therefore, the network nodes have to serve also as routers and hosts. The network nodes are mobile and they are able to communicate wirelessly with each other by sending and receiving data packets.

The working group has the following goals:

- Trying to standardize an intra-domain uni-cast routing protocol.
- Going to address the security issues in intended usage environments.
- Going to address also the layering more advanced services, such as multicast and QoS extensions, on top of the initial routing technology.

This advances in mobile network technologies is lifted up by other commercial initiatives, such as IEEE 802.11 Wireless LAN (WLAN) standard. In addition to IEEE 802.11, it is worth to mention Bluetooth (launch in 1998) that is the first commercial ad hoc radio system predicted to be used on a large scale.

## 2.1 MANET Features

MANET has the following unique features:

1) **Autonomous terminal**. In MANET, each mobile terminal is an autonomous node, which may function as both a host and a router. In other words, besides the basic processing ability as a host, the mobile nodes can also perform switching functions as a router. So usually endpoints and switches are indistinguishable in MANET.

2) **Distributed operation**. Since there is no background network for the central control of the network operations, the control and management of the network is distributed among the terminals. The nodes involved in a MANET should collaborate amongst themselves and each node acts as a relay as needed, to implement functions e.g. security and routing.

3) **Multi-hop routing**. Basic types of ad hoc routing algorithms can be single-hop and multihop, based on different link layer attributes and routing protocols. Single-hop MANET is simpler than multihop in terms of structure and implementation, with the cost of lesser functionality and applicability. When delivering data packets from a source to its destination out of the direct wireless transmission range, the packets should be forwarded via one or more intermediate nodes.

4) **Dynamic network topology**. Since the nodes are mobile, the network topology may change rapidly and unpredictably and the connectivity among the terminals may vary with time. MANET should adapt to the traffic and propagation conditions as well as the mobility patterns of the mobile network nodes. The mobile nodes in the network dynamically establish routing among themselves as they move about, forming their own network on the fly.

5) **Fluctuating link capacity**. The nature of high bit-error rates of wireless connection might be more profound in a MANET. One end-to-end path can be shared by several sessions. The channel over which the terminals communicate is subject to noise, fading, and interference, and has less bandwidth than a wired network. In some scenarios, the path between any pair of users can traverse multiple wireless links and the link themselves can be heterogeneous.

6) **Light-weight terminals**. In most cases, the MANET nodes are mobile devices with less CPU processing capability, small memory size, and low power storage. Such devices need optimized algorithms and mechanisms that implement the computing and communicating functions.

## 2.2 MANET Applications

The set of applications for MANETs is diverse, ranging from large-scale, mobile, highly dynamic networks, to small, static networks that are constrained by power sources. Besides the legacy applications that move from traditional infrastructured environment into the ad hoc context, a great deal of new service scan and will be generated for the new environment. Typical applications include:

1) **Military battlefield.** Military equipment now routinely contains some sort of computer equipment. Ad hoc networking would allow the military to take advantage of commonplace network technology to maintain an information network between the soldiers, vehicles, and military information head quarters.

2) **Commercial sector.** Ad hoc can be used in emergency/rescue operations for disaster relief efforts, e.g. in fire, flood, or earthquake. Emergency rescue operations must take place where non-existing or damaged communications infrastructure and rapid deployment of a communication network is needed.
3) Local level. Ad hoc networks can autonomously link an instant and temporary multimedia network using notebook computers or palmtop computers to spread and share information among participants at a e.g. conference or classroom. Another appropriate local level application might be in home networks where devices can communicate directly to exchange information.
4) Personal Area Network (PAN). Short-range MANET can simplify the intercommunication between various mobile devices (such as a PDA, a laptop, and a cellular phone). Tedious wired cables are replaced with wireless connections .Such an ad hoc network can also extend the access to the Internet or other networks by mechanisms e.g. Wireless LAN (WLAN), GPRS, and UMTS. The PAN is potentially a promising application field of MANET .

## 2.3 Challenges and Barrier

The MANET challenges include the flowing points:

1) **Routing.** Since the topology of the network is constantly changing, the issue of routing packets between any pair of nodes becomes a challenging task.
2) **Security and Reliability.** In addition to the common vulnerabilities of wireless connection, an ad hoc network has its particular security

problems due to e.g. nasty neighbour relaying packets. The feature of distributed operation requires different schemes of authentication and key management. Further, wireless link characteristics introduce also reliability problems, because of the limited wireless transmission range, the broadcast nature of the wireless medium, mobility-induced packet losses, and data transmission errors.

3) **Quality of Service (QoS).** Providing different quality of service levels in a constantly changing environment will be a challenge. The inherent stochastic feature of communications quality in a MANET makes it difficult to offer fixed guarantees on the services offered to a device. An adaptive QoS must be implemented over the traditional resource reservation to support the multimedia services.
4) **Internetworking.** In addition to the communication within an ad hoc network, internetworking between MANET and fixed networks (mainly IP based) is often expected in many cases. The coexistence of routing protocols in such a mobile device is a challenge for the harmonious mobility management.
5) **Power Consumption.** For most of the light-weight mobile terminals, the communication-related functions should be optimized for lean power consumption. Conservation of power and power-aware routing must be taken into consideration.

## 2.4 MANET Topologies

The MANET connectivity is based on peer communication. This is an important difference compared to cellular networks using base stations and fixed infrastructure. In addition to sending data packets directly, the nodes may also need other nodes to relay the traffic, as presented in Figure 2.1. The described situation is multi-hopping.

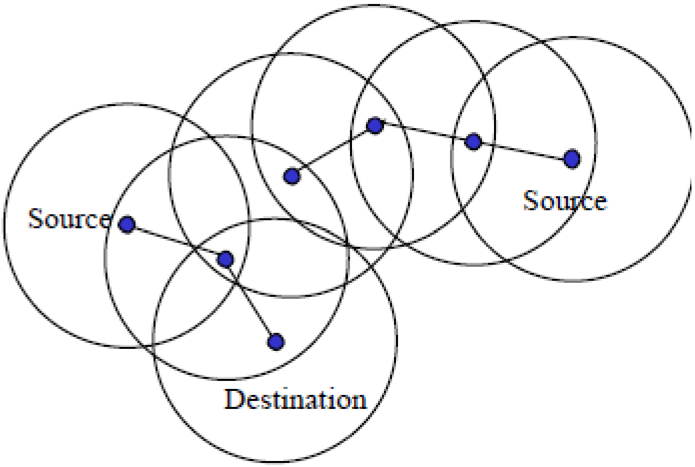


**Fig 2.3 Multi-hopping MANET Communications.**

## 2.4.1 Single-hop and Multi-hop

Because there are no separate terminals and radio units, MANETs have their own network topology that is either a *single-hop* or a *multi-hop*. While single-hop network nodes send data directly from source to destination, in multi-hop network nodes can use other nodes to relay their traffic.

Multiple hops increase the transmission delay, but it can be compensated with increased link rate. Therefore, the end-to-end delay may actually benefit from multiple hops. In fact, multi-hopping may even be necessary to be able to reach a very distant node in available frequency range.

The large transmission range causes interference and reduces the effective bandwidth available to the network nodes by increasing the number of nodes competing for the same network bandwidth. Therefore, it is beneficial to use multi-hopping (Figure 2.1) or at least control the transmission range as presented in single-hop ad hoc networking example in Figure 2.2.

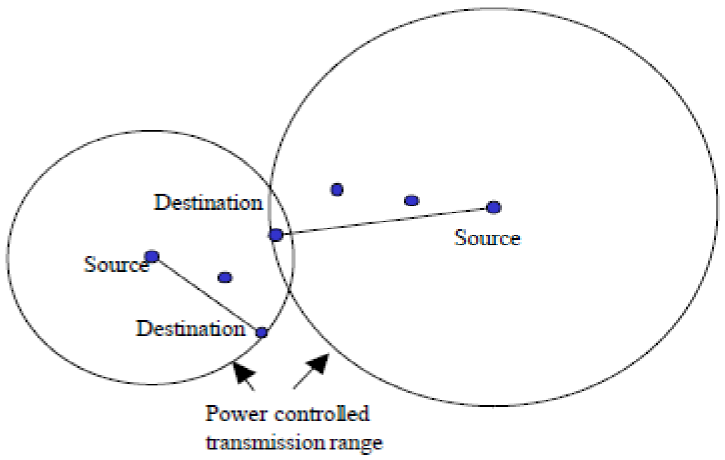


**Fig 2.4 Single-hop MANET Communications.**

The beneficial of multi-hop networking  against single-hopping are the following:

- Increases the network scalability.
- Reduces the interference.
- Increases the overall network throughput.
- Decreases the delay seen by application.
- Reduces the energy consumption in data transmission.

## 2.5 MANET Technologies

Several technologies enable MANET, both in Wireless Personal Area Networks (WPAN) and Wireless Local Area Networks (WLAN). By definition, WPANs are limited to a few meters, with data rates up to several hundred Kilobits per second. As for WLANs, their communicating range is larger and they can achieve data rates of tens of Megabits per second.

Generally two main link layer technologies for MANET are used:

1. The IEEE 802.11 standard for WLANs and
2. The Bluetooth specifications for WPANs.

# *Chapter 3*

# *Mobile Ad Hoc Routing Protocols*

## 3.1 Routing in Mobile Ad-Hoc Networking

MANET nodes cooperate to route packets to each other to allow communication through hops. This property is not sufficient to ensure network connectivity. Packets must be forwarded and path from the source to its destination should be determined via a process so-called *guidance*. A simple way to perform routing is to enter static routes through the integration of static routing table in each node according to the topology.

In Mobile ad hoc multi-hop routing protocols nodes dynamically create routing among themselves to form their own network. These protocols apply routing schemes. Two techniques to calculate the best route:

1. **Distance-vector routing.** Cost is calculated to reach the destination by using different road standards. To calculate the road Bellman Ford algorithm can be used. Distance-vector routing protocols has a low computational complexity and overhead message. Algorithm struggle to converge on the new topology when network conditions are changes.
2. **link-state routing.** Identify nodes that are connected to other nodes and each node attempts to construct a map of the network connection. Dijkstra algorithm is used to how to calculate best route rather than sharing routing tables with the neighborhood. Two techniques are used, well known in wired networks (Routing Information Protocol

(RIP) Open Shortest Path First (OSPF)) and hop Declaration Ad hoc routing protocols (AODV and OLSR).

## 3.2 Traditional Network Routing algorithms

Routing algorithms have been developed on the routing protocols used in packet switched networks in the link-state and OSPF, or distance vector algorithms RIP. This division also provides a good way to classify routing protocols according to the information that they use.

In the link-state algorithms routing table is formed of link status information and this information is collected from all the links that are placed between the other nodes in the network.

Distance vector algorithm is another popular routing algorithm which is based on Shortest Path First algorithms typically stores information only about the next stage to the desired destination. Name remote vectors derived from the cost metric, which is typically the distance and stored in its routing table. Distance vector algorithms are easy to program and require less memory than link-state algorithms. They also are more local routing updates, because it did not all information is stored in each node. Therefore, remote vector algorithms also have disadvantages such as: very slow convergence.

## 3.3 Mobile Network Routing algorithms

Protocols are divided in terms of exchange of information into three groups:

1. **Reaction protocols:** is to reduce the number of messages exchanged information control when the contract idle any traffic and their reaction only to active node when you decide to send information, at the time of the

beginning of the network to route discovery process, which can be very expensive in terms of resources.

2. **Proactive protocols:** is the exchange of information on a regular basis and control in order to maintain information about the network topology and to obtain information on the routes to all destinations.
3. **Hybrid protocols:** are a mixture of reaction and proactive in the process of unification.

**Fig 3.5** shows the classification based on routing protocols.

| | **Reactive Protocols** | **Proactive Protocols** |
|---|---|---|
| **Link State Protocols** | DSR, TORA | OLSR, TBRPF, TORA, LANMAR/FSR |
| **Distance Vector Protocols** | AODV | |

Route is classified into three types:

1. Pre-active, if a route has not been used;
2. Active, if a route is being used;
3. Post-active, if a route was used before but now is not.

## 3.4 Reactive Protocols (RP)

Reactive protocols provide ways for the flow of information only upon request. This means that the Protocol will react to find a way when the source node needs to send information to the destination node. During the remaining time, *active* or *idle* node to participate in the process of forwarding service contract to another source. Reactive protocols does not consume a lot of resources when idle, but to discover the way (they do not have any information topology), they ask the floods between the network and in this case consume a large amount of network resources. Examples of reactive protocols are AODV and DSR.

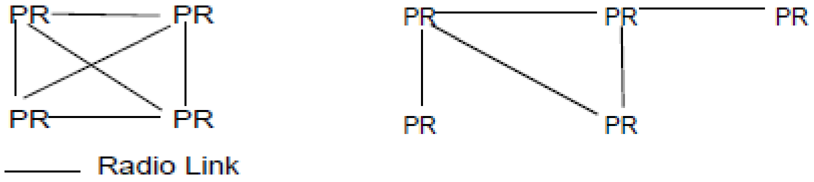
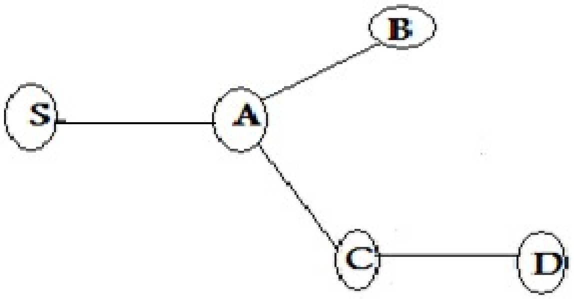


**Fig 3.6** Examples of PR-network.

## 3.4.1 AODV (Ad hoc Demand Distance Vector)

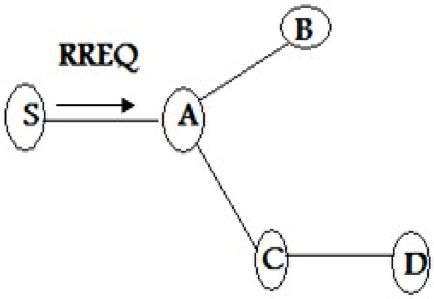
AODV is a reactive protocol used in many systems and has a lot of successes which belongs to a class of routing protocols distance vector. In these protocols to reach the destination mobile node uses next-hop allow smaller space in the number of hops between him and destination nodes do not keep mobile information the contract was not worried hand information on active traffic flows. In the event that the node expressed a desire to send packets to the destination begins checking if she has a way available in its routing table and in the absence of a road it will embark on the road discovery by sending request packet route (RREQ) to locate the receiver and distributed this packet to its neighbors. Receivers of RREQ look forward to determine if they have a route available to the recipient in their routing tables. If the neighbors do not know the destination or there was no way exists the neighbors turn to rebroadcast the packet RREQ to their neighbors and keep trace. Thus packets are sent RREQ across the network to the packet arrives on the one hand, or a mobile node that contains the road to this recipient. The following forms describes the process of Route Discovery:



1. Node S needs a route to D

2. Creates a Route Request (RREQ)

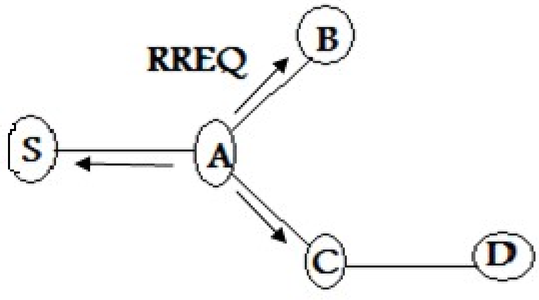
- Enters D’s IP addr, seq , S’s IP addr, seq, hopcount (=0)



3. Node S broadcasts RREQ to neighbors

4. Node A receives RREQ Makes a reverse route entery for S

Dest =S, nexthop=S, hopcount = 1 , It has no routes to D, so it rebroadcasts RREQ



5. Node A receives RREQ

- Makes a reverse route entery for S. dest=S, nexthop=S, hopcount=1

- It has no routes to D, so it rebroadcasts RREQ

6. Node C receives RREQ

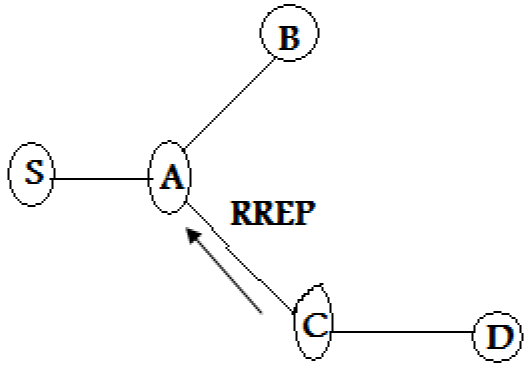
- Makes a reverse route entry for S. dest=S, nexthop=A, hopcount=2

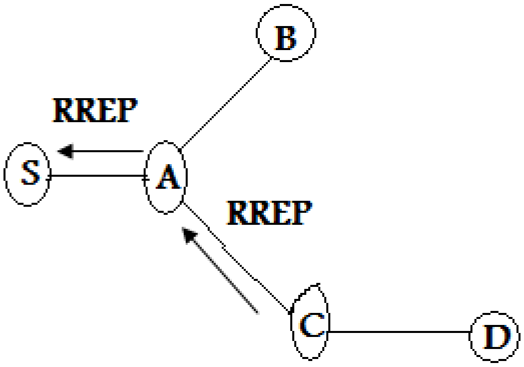
- It has a route to D, and the seq for route to D is >= D’s seq in RREQ

C creates a Route Reply (RREP)

- Enters D’s IP addr, seq, S’s IP addr, hopcount to D (=1)

- Unicasts RREP to A





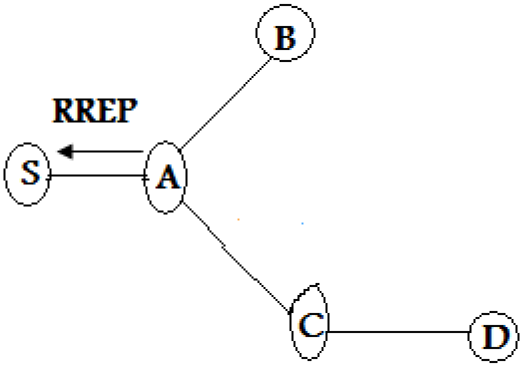
7. Node A receives RREP

- Makes a forward route entry to D. dest=D, nexthop=C, hopcount=2

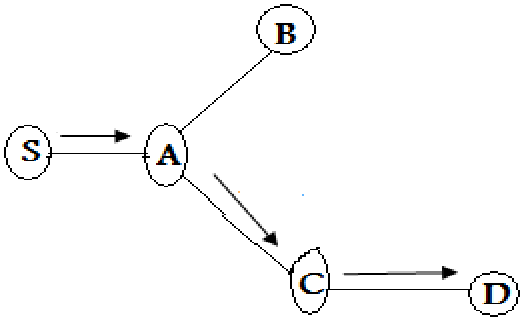
- Unicasts RREP to S

8. Node S receives RREP

-Makes a forward route entry to D. dest=D, nexthop =A, hopcount = 3



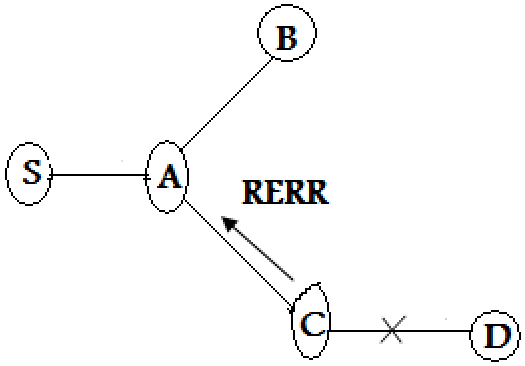
AODV Data Delivery:



9. Node S receives RREP

- Makes a forward route entry to D. dest=D, nexthop =A, hopcount = 3
- Sends data packet on route to D

As in the case of a break in the link are sent a message mis as in the following



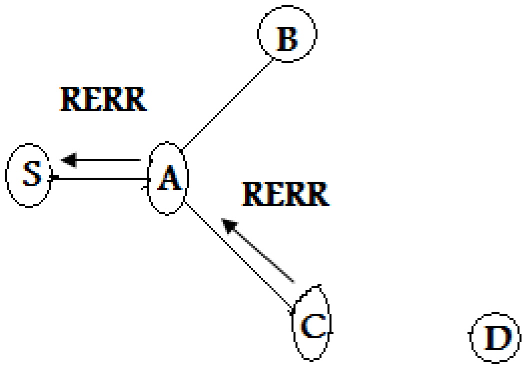
Steps:

1. Link between C and D breaks

2. Node C invalidates route to D in route table

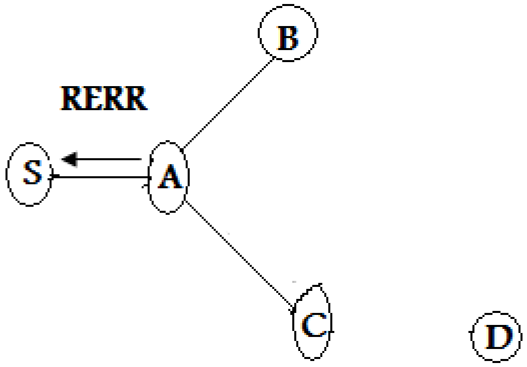
3. Node C creates Route Error message

- Lists all destinations that are now unreachable
- Sends to upstream neighbors



4. Node A receives RERR

- Checks whether C is its next hop on route to D
- Deletes route to D (makes distance -> infinity)
- Forwards RERR to S



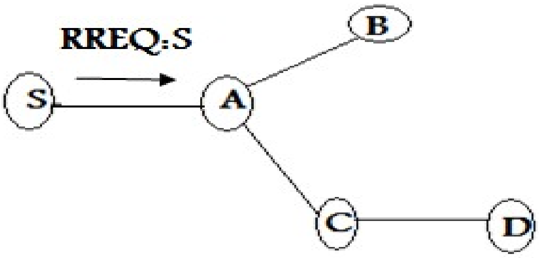
5. Node S receives RERR

- Checks whether A is its next hop on route to D
- Deletes route to D
- Rediscovers route if still needed

### 3.4.2 DSR (Dynamic Routing Source)

DSR is also a reactive protocol, and it works in a way very similar fashion to AODV. The main difference is in the action direction. AODV is driven table, while DSR uses source routing in idle nodes and when a node decide to send traffic to a destination, it will proceed without creating local routing tables, and all intermediate node adds and prints identity to the packet header. The disadvantage of source routing is, the additional overhead in packets.

Example for Route Discovery in DSR:

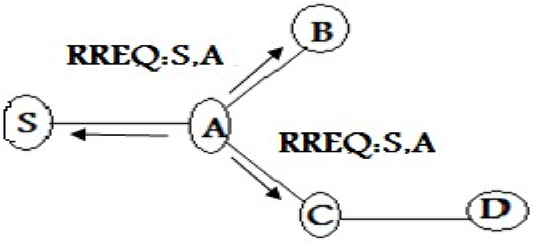


1. Node S needs a route to D.

2. Broadcasts RREQ packet.

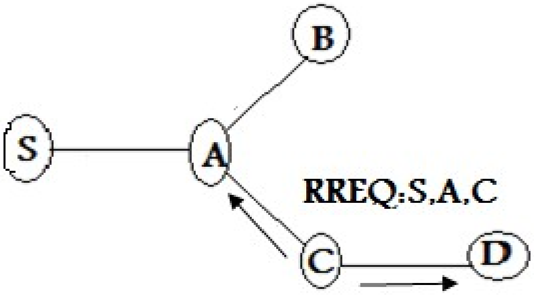
3. Node A receives packet, has no route to D.

- Rebroadcasts packet after adding its address to source Route.



4. Node C receives RREQ, has no route to D.

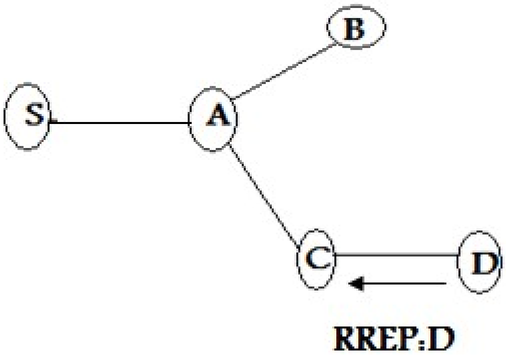
- Rebroadcasts packet after adding its address to source Route.



5. Node D receives RREQ, unicasts RREP to C

- Puts D in RREP source route

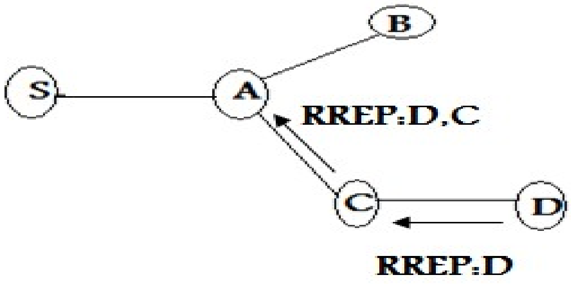
**Step of  Route Reply in DSR;**



6. Node C receives RREP

- Adds its address to source route

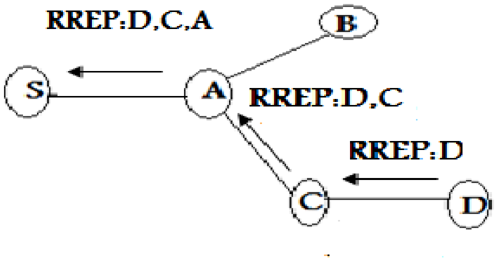
- Unicasts to A



7. Node A receives RREP

- Adds its address to source route

- Unicasts to S



8. Node S receives RREP

- Uses route for data packet transmissions

## 3.5 Proactive Protocols

Proactive routing protocol saves the information that is updated regularly on the network in each node. The way to get this information through messages sent or periodically raised in response to significant events that occur in the network. This information allows each node to keep knowledge topology offers the possibility of calculating the route for each destination in the network. Examples of proactive routing protocols are:

1. DSDV (Destination sequence distance vector),
2. OLSR (Mohsen link-state routing), and

3. TBRPF (Topology broadcast at the direction reverse path).

### 3.5.1 OLSR (Mohsen Link State Routing Protocol)

In OLSR it is possible to maximize the dissemination of information packets in the network and reduce the overhead due to flooding procedure. The protocol is based on the concept of multiple relays. Each node selects A multiple relay (MPR) according to the following rule: choose Minimum subset of the adjacent nodes which can cover all the neighbors hop with symmetric connections. Figure 3.2 shows a comparison between the classical and floods MPR. The denser the network, the more efficient is the MPR. OLSR protocol is able to identify a list of several points relays that will cover all the neighbors. Each station has a multiple sequence of choice, which will serve as a vector for this station. When a list of selected MPRs is sent to all network nodes (MPR flooding), each node receives this information will be able to build a topology map and calculate the best route for each destination.

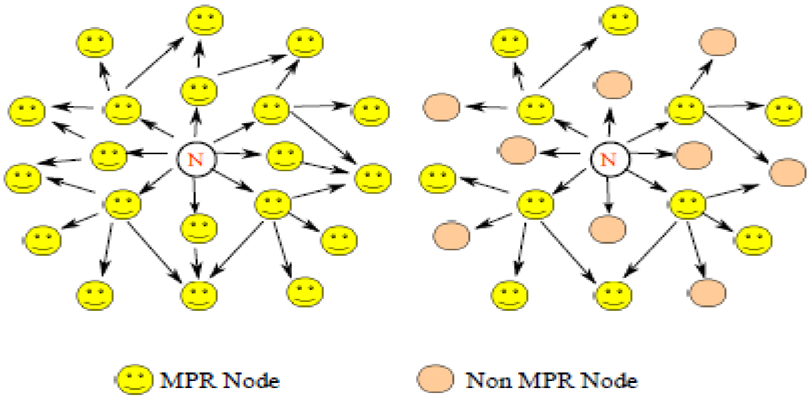


**Fig 3.7 Using the discovery floods.**

### 3.5.2 TBRPF (Topology Broadcast based on Reverse Path Forwarding)

TBRPF protocol is proactive seeks to limit the use of the bandwidth of the control messages. Unlike OLSR which keeps partial topology of network (using MPR), all nodes in TBRPF have full knowledge of the network topology. This knowledge is more expensive, but allows the use of multiple paths tracks. Shortest path tree is maintained at each node to all other nodes. There a unique tree at each node. Trees are constructed by itself according to the update messages.

## 3.6 Reactive Versus Proactive Protocols

Proactive protocols follow roads to all destinations and major benefit of this approach is that it imposes a minimum delay at the start of communication with arbitrary point. However, it suffers from additional traffic control caused by continuous updates. This can lead to waste scarce resources on bandwidth unused identify ways. Also lead to create more congestion. Due to the higher priority of control packets, data packets will be lost regularly, which led to re-send and congestion even further. Proactive routing protocols do not fit well in the environment fast-moving, compared to reactive protocols. Table 3.1 summarizes the comparison between two types of protocols and Table 3.2 shows the characteristics of the protocols.

**Table 3.1 Overall comparison between proactive and reactive routing Protocols**.

| **Compared feature** | **On-demand, reactive** | **Table-driven, proactive** |
|---|---|---|
| Availability of routing information | Available when needed | Always available regardless of need |
| Routing philosophy | Flat | Mostly flat, except CGSR |
| Periodic updates | Not required | Required |
| Coping with mobility | Use localized route discovery as in ASB and SSR | Inform other nodes to achieve a consistent routing table |
| Signaling traffic generation | Grows with increasing mobility of active routes | Greater than that of on-demand routing |
| Quality of service support | Few can support QoS, although most support shortest path | Mainly shortest path as the QoS metric |

**Table 3.2 shows the characteristics of the protocols.**

| | **FSR** | **OLSR** | **TBRPF** | **AODV** | **DSR** |
|---|---|---|---|---|---|
| **Routing Philosophy** | Proactive | Proactive | Proactive | On-Demand | On-Demand |
| **Routing Metric** | Shortest Path | Shortest Path | Shortest Path | Shortest Path | Shortest Path |
| **Frequency of Updates** | Periodically | Periodically | Periodically , As needed (link changes) | As needed (data traffic) | As needed (data traffic) |
| **Use Sequences Numbers** | Yes | Yes | Yes (HELLO) | Yes | No |
| **Loop-Free** | Yes | Yes | Yes | Yes | Yes |
| **Worst Case exists** | Yes | No | No | Yes (full flooding) | Yes (full flooding) |
| **Multiple Paths** | Yes | No | No | No | Yes |
| **Storage Complexity** | O(n) | O(n) | O(n) | O(e) | O(e) |
| **Common Complexity** | O(n) | O(n) | O(n) | O(2n) | O(2n) |

Performance of routing protocols is not well understood and there are several methods that can be used to classify routing protocols by certain considerations including:

- whether to use hierarchical addressing scheme.
- whether the protocol is capable of multicast routing. Table 3.3 shows some of the proposed routing protocols and classified into categories proactive or reactive.

**Table 3.3** Comparison of some routing protocols.

| **Routing Protocol** | **Way of obtaining routing information** |
|---|---|
| ABR | Reactive |
| AODV | Reactive |
| CBGR | Proactive |
| CBRP | Reactive |
| DSDV | Proactive |
| DSR | Reactive |
| FSR | Proactive |
| OLSR | Proactive |
| SSR | Reactive |
| STAR | Proactive |
| TORA | Reactive |
| WRP | Proactive |
| ZRP | Proactive & reactive |

# *Chapter 4*

# ***Simulation and Results***

The performance of routing protocols is compared and simulated using OMNet++.

## 4.1 OMNeT++ Overview

OMNeT++ is an object-oriented modular discrete event network simulation framework. It has a generic architecture, so it can be used in various problem domains for example:

- modeling of wired and wireless communication networks
- protocol modeling
- modeling of queueing networks

OMNeT++ itself is not a simulator of anything concrete, but rather provides infrastructure and tools for writing simulations. One of the fundamental ingredients of this infrastructure is component architecture for simulation models. Models are assembled from reusable components termed modules. Well-written modules are truly reusable, and can be combined in various ways like LEGO blocks.

Modules can be connected with each other via gates (other systems would call them ports), and combined to form compound modules. The depth of module nesting is not limited. Modules communicate through message passing, where messages may carry arbitrary data structures.

Modules can pass messages along predefined paths via gates and connections, or directly to their destination; the latter is useful for wireless simulations, for example. Modules may have parameters that can be used to customize module behavior and/or to parameterize the model’s topology. Modules at the lowest level of the module hierarchy are called simple modules, and they encapsulate model behavior. Simple modules are programmed in C++, and make use of the simulation library.

OMNeT++ simulations can be run under various user interfaces. Graphical, animating user interfaces are highly useful for demonstration and debugging purposes, and command-line user interfaces are best for batch execution.

The simulator as well as user interfaces and tools are highly portable. They are tested on the most common operating systems (Linux, Mac OS/X, Windows), and they can be compiled out of the box or after trivial modifications on most Unix-like operating systems.

OMNeT++ also supports parallel distributed simulation. OMNeT++ can use several mechanisms for communication between partitions of a parallel distributed simulation, for example MPI or named pipes. The parallel simulation algorithm can easily be extended, or new ones can be plugged in. Models do not need any special instrumentation to be run in parallel – it is just a matter of configuration. OMNeT++ can even be used for classroom presentation of parallel simulation algorithms, because simulations can be run in parallel even under the GUI that provides detailed feedback on what is going on.

## 4.2 Setup and Experimentation

Steps to setup the simulation and run the experiments are as follow:

### 4.2.1 Simulation Steps

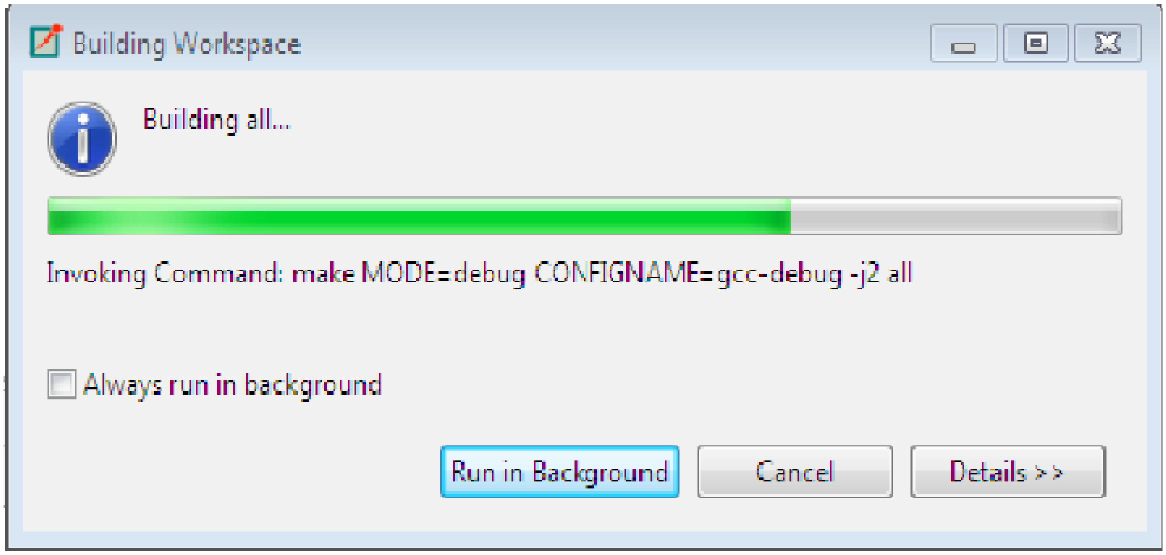
To run the experiments using OMNeT++ we followed these steps:

**Step1:**

Download the Inet package that contains the routing protocols from web site http://inet.omnetpp.org/index.php?n=Main.Download.

**Step2:**

Import the Inet package from File → Import → Existing Projects into Workspace and select the Inet package path, then build it inside the OMNeT++ from Project → Build All and wait until the building finish.



Building all...

Invoking Command: make MODE=debug CONFIGNAME=gcc-debug -j2 all

□ Always run in background

**Fig 4.8** Building Inet Package

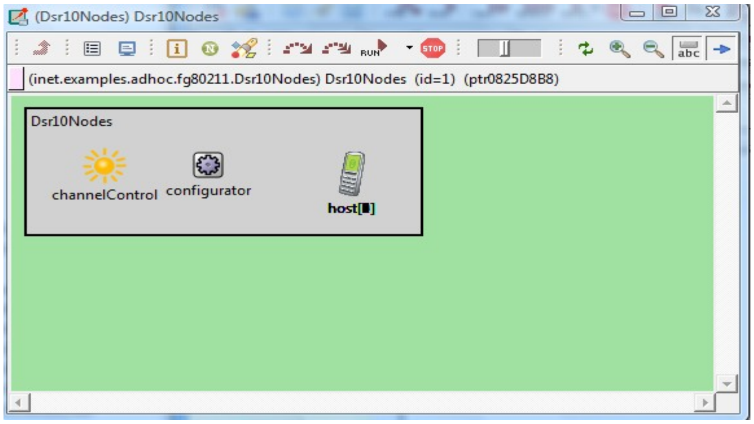
## 4.2.2 Experimentation

To see the delay in the transmission process, two types of protocols are tested, first the test is performed in a network containing 10 devices and the other containing 30 devices.

**<u>The first test in 10 devices</u>**

### The first protocol running ( Dsr in 10 node)

The flowing figure is the Modules' relationships and communication links are stored as plain text Network Description (NED) files and can be modeled graphically as shown in Figure 4.2.



(Dsr10Nodes) Dsr10Nodes

(inet.examples.adhoc.fg80211.Dsr10Nodes) Dsr10Nodes (id=1) (ptr0825D8B8)

**Fig 4.9The network model.**

**Step1:**

After the selected and existing protocols we doing right click on .ini file inside protocol's files and select Run As →OMNeT++ Simulation as shown in Figure 4.3.

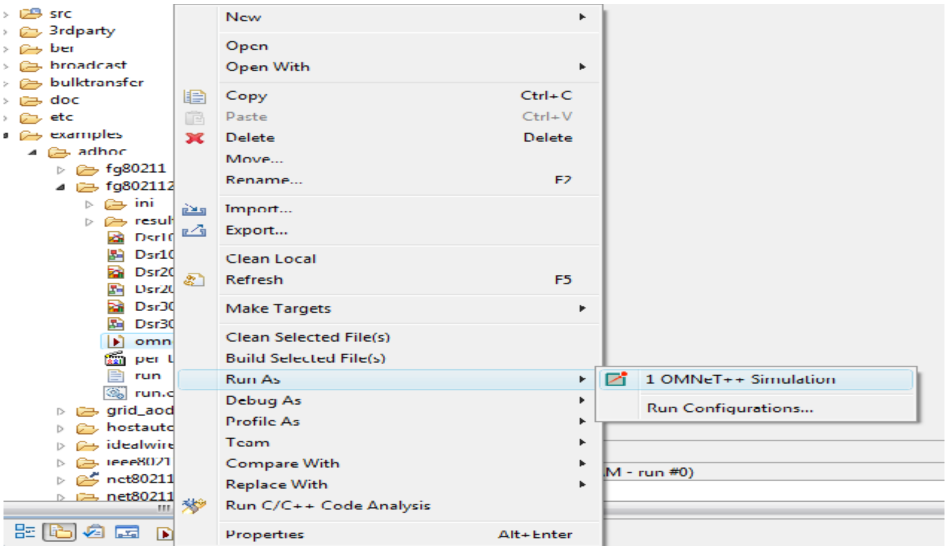
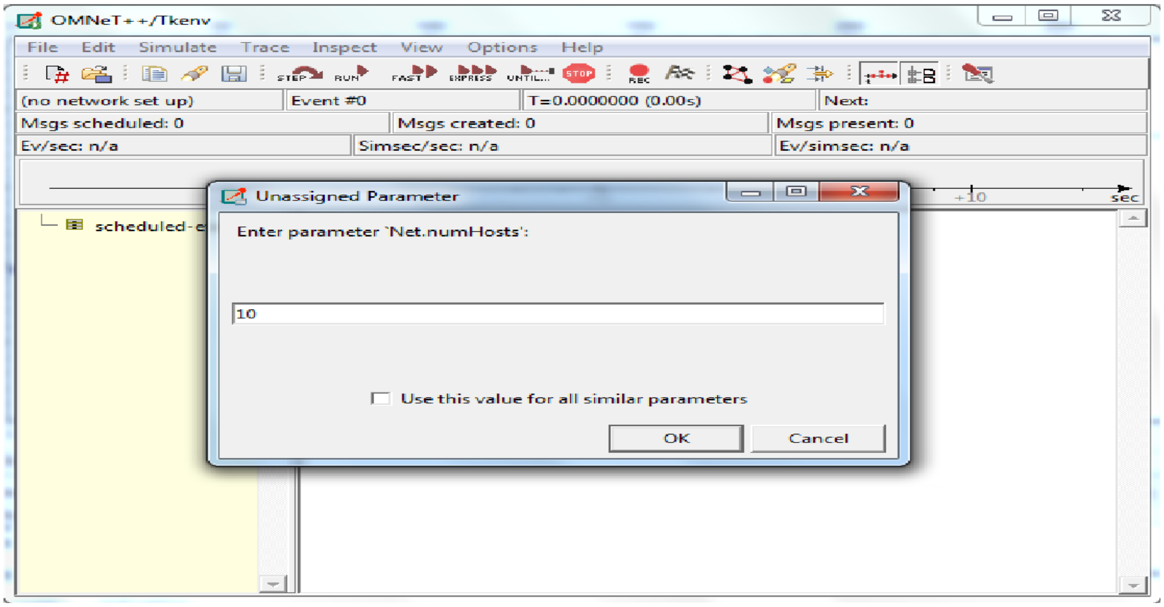


**Fig 4.10** Run simulation.

**Step2:**

After selecting the run command the following window in Figure 4.4 will be displayed to determine the number of hosts.


OMNeT++/Tkenv
File Edit Simulate Trace Inspect View Options Help
STEP RUN FAST EXPRESS UNTIL... STOP REC
scheduled-e
+10
sec
Unassigned Parameter
OK
Cancel

(no network set up) | Event #0 | T=0.0000000 (0.00s) | Next:

Msgs scheduled: 0 | Msgs created: 0 | Msgs present: 0

Ev/sec: n/a | Simsec/sec: n/a | Ev/simsec: n/a

Enter parameter `Net.numHosts':

10

□ Use this value for all similar parameters

**Fig 4.11** Window to determine the number of hosts.

If the 10 nodes were entered this window will be displayed to show the animation of the hosts and the sending of the packets.

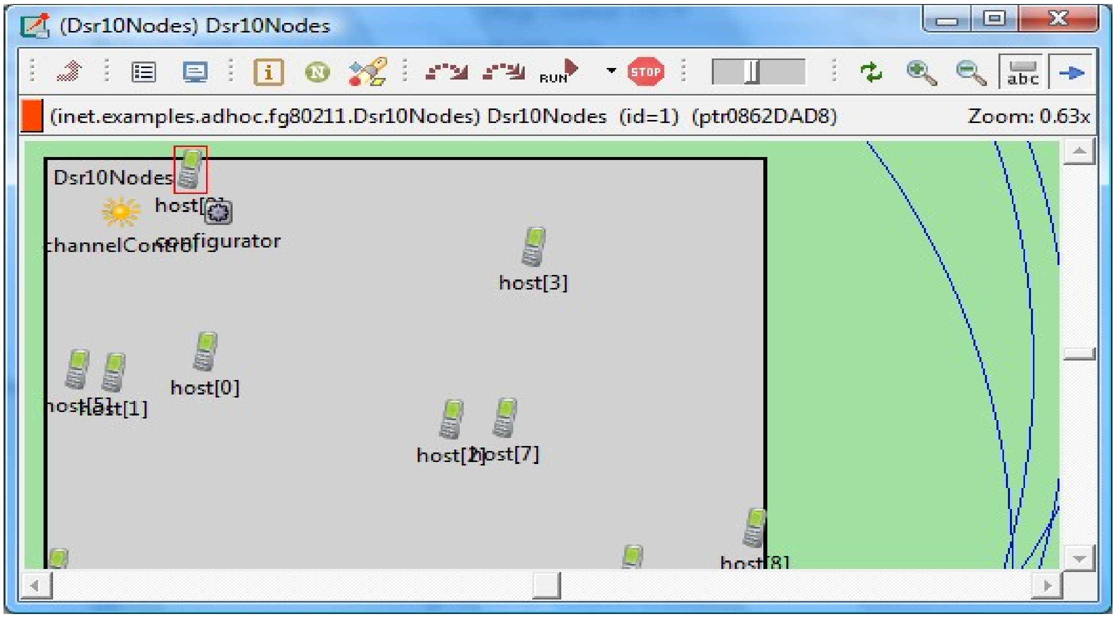
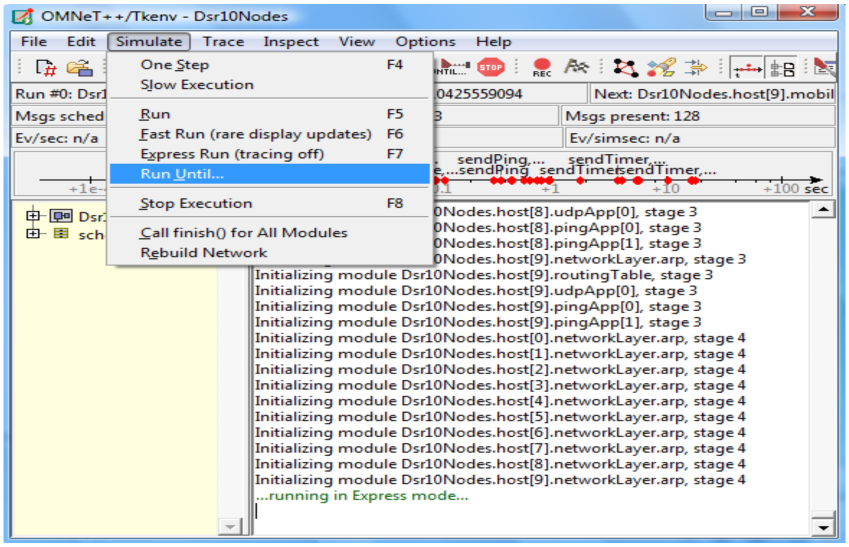


**Fig 4.12** Node animation window.

**Step5:**

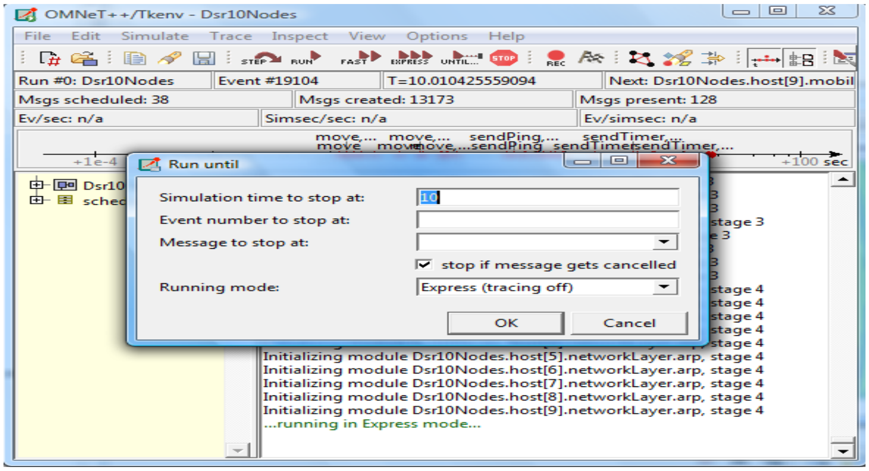
To determine the total simulation time select from Simulate menu → Run Until and specify the time, as shown in Figure 4.6.



```
Run #0: Dsr1          0425559094          Next: Dsr10Nodes.host[9].mobil
Msgs sched            3                   Msgs present: 128
Ev/sec: n/a                               Ev/simsec: n/a
```

```
0Nodes.host[8].udpApp[0], stage 3
0Nodes.host[8].pingApp[0], stage 3
0Nodes.host[8].pingApp[1], stage 3
0Nodes.host[9].networkLayer.arp, stage 3
Initializing module Dsr10Nodes.host[9].routingTable, stage 3
Initializing module Dsr10Nodes.host[9].udpApp[0], stage 3
Initializing module Dsr10Nodes.host[9].pingApp[0], stage 3
Initializing module Dsr10Nodes.host[9].pingApp[1], stage 3
Initializing module Dsr10Nodes.host[0].networkLayer.arp, stage 4
Initializing module Dsr10Nodes.host[1].networkLayer.arp, stage 4
Initializing module Dsr10Nodes.host[2].networkLayer.arp, stage 4
Initializing module Dsr10Nodes.host[3].networkLayer.arp, stage 4
Initializing module Dsr10Nodes.host[4].networkLayer.arp, stage 4
Initializing module Dsr10Nodes.host[5].networkLayer.arp, stage 4
Initializing module Dsr10Nodes.host[6].networkLayer.arp, stage 4
Initializing module Dsr10Nodes.host[7].networkLayer.arp, stage 4
Initializing module Dsr10Nodes.host[8].networkLayer.arp, stage 4
Initializing module Dsr10Nodes.host[9].networkLayer.arp, stage 4
...running in Express mode...
```

**Fig 4.13** Specified the simulation time.


OMNeT++/Tkenv - Dsr10Nodes
File Edit Simulate Trace Inspect View Options Help
STEP RUN FAST EXPRESS UNTIL... STOP REC
move,... move,... sendPing,... sendTimer,...
move move move,...sendPing sendTimer sendTimer,...
+1e-4
+100 sec
Dsr10
sched
Run until
OK
Cancel

Run #0: Dsr10Nodes | Event #19104 | T=10.010425559094 | Next: Dsr10Nodes.host[9].mobil
Msgs scheduled: 38 | Msgs created: 13173 | Msgs present: 128
Ev/sec: n/a | Simsec/sec: n/a | Ev/simsec: n/a

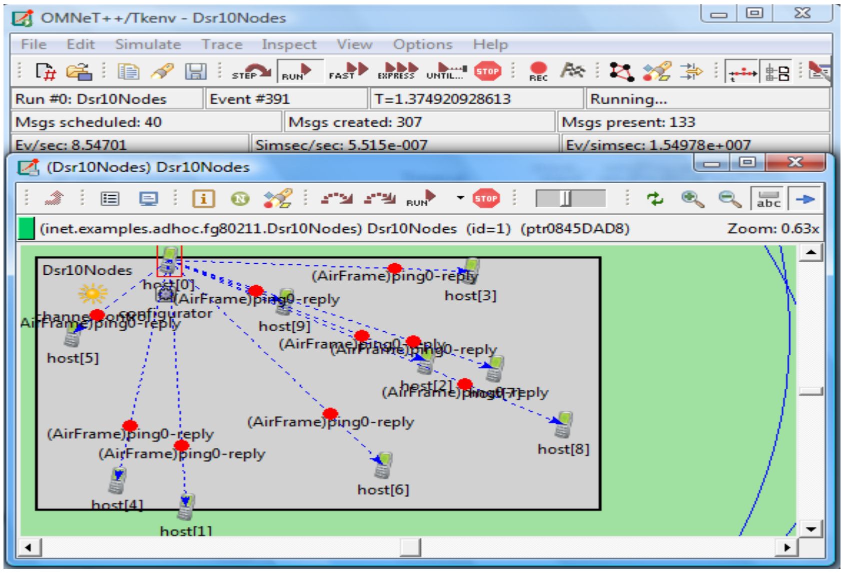
Simulation time to stop at: 10
Event number to stop at:
Message to stop at:
☑ stop if message gets cancelled
Running mode: Express (tracing off)

```
Initializing module Dsr10Nodes.host[5].networkLayer.arp, stage 4
Initializing module Dsr10Nodes.host[6].networkLayer.arp, stage 4
Initializing module Dsr10Nodes.host[7].networkLayer.arp, stage 4
Initializing module Dsr10Nodes.host[8].networkLayer.arp, stage 4
Initializing module Dsr10Nodes.host[9].networkLayer.arp, stage 4
...running in Express mode...
```

**Fig 4.14** Total simulation time.

**Step6:**

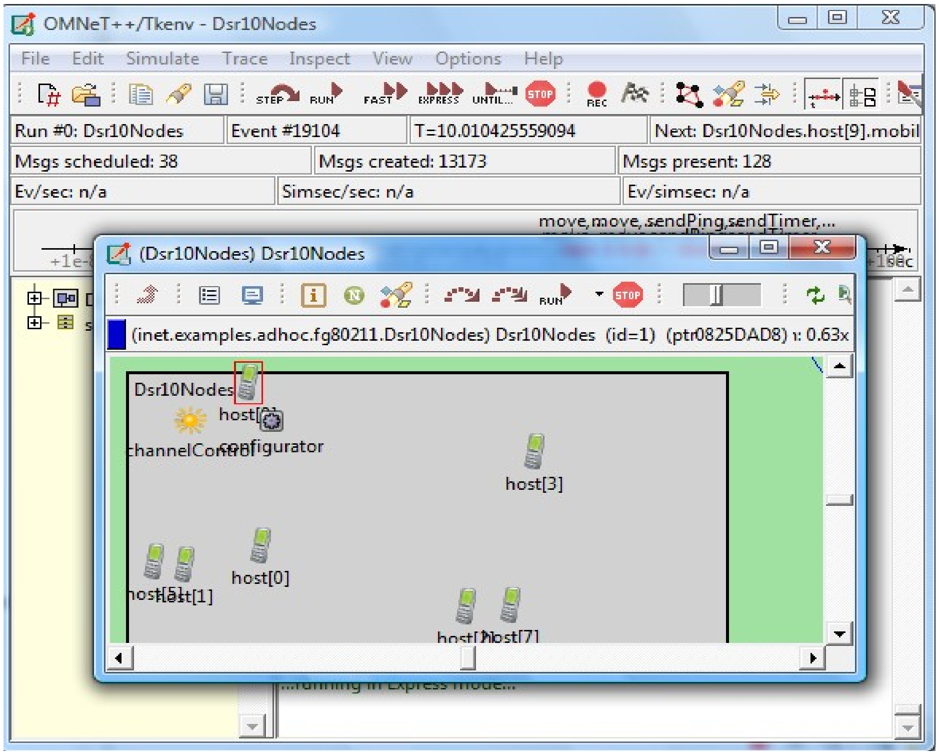
To start the simulation click on the RUN button in the simulation window, display the running of the simulation, also show simulation while sending Request Messages (RREQ) , as shown in Figure 4.8.



Run #0: Dsr10Nodes | Event #391 | T=1.374920928613 | Running...
Msgs scheduled: 40 | Msgs created: 307 | Msgs present: 133
Ev/sec: 8.54701 | Simsec/sec: 5.515e-007 | Ev/simsec: 1.54978e+007

(inet.examples.adhoc.fg80211.Dsr10Nodes) Dsr10Nodes (id=1) (ptr0845DAD8) Zoom: 0.63x

**Fig 4.15** Running and sending the simulation.



Run #0: Dsr10Nodes | Event #19104 | T=10.010425559094 | Next: Dsr10Nodes.host[9].mobil

Msgs scheduled: 38 | Msgs created: 13173 | Msgs present: 128

Ev/sec: n/a | Simsec/sec: n/a | Ev/simsec: n/a

(inet.examples.adhoc.fg80211.Dsr10Nodes) Dsr10Nodes (id=1) (ptr0825DAD8) ι: 0.63x

**Fig 4.16** End of the simulation.

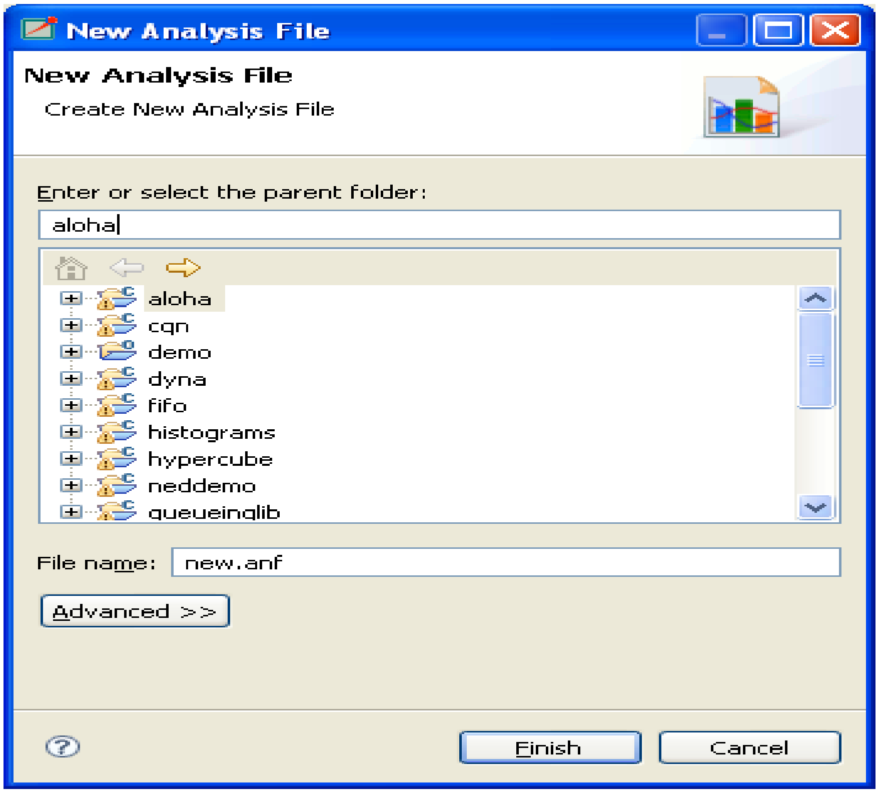
**Step7:**

After the simulation is done we need to create the analysis file to analyze the simulation result for the routing protocol.

### 4.2.3 Analyzing the Results

Analyzing the simulation result is a process includes several steps, first step filter and transform the data, then chart the result. The OMNeT++ programming support using the analysis file (.anf), and it's a statistical analysis tool integrated into the Eclipse environment to obtain the results from the data.

To create a new analysis file, choose File → New → Analysis File from the menu. Select the folder for the new file and enter the file name. Press Finish to create and open an empty analysis file, as shown in Figure 4.10.



New Analysis File

Create New Analysis File

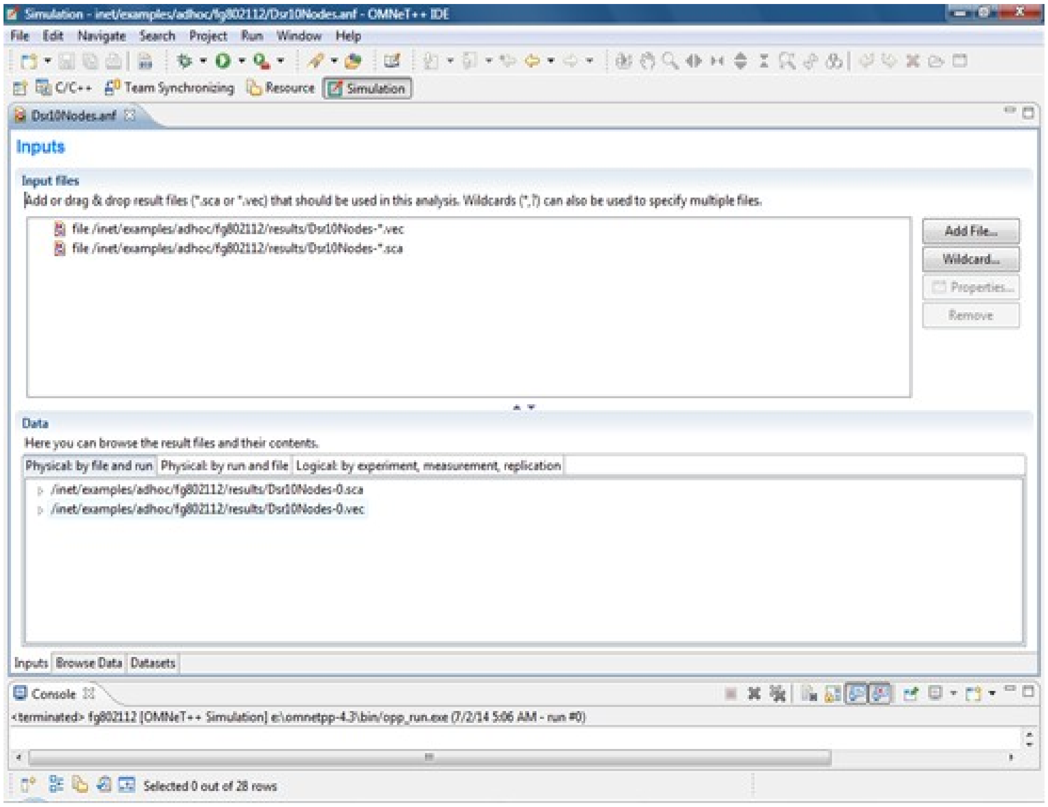
Enter or select the parent folder: aloha

File name: new.anf

**Fig 4.17** New Analysis File.

Then open the New Analysis File with Analysis Editor by double-click on the result file in the Project Explorer View. The Analysis Editor is implemented as a multi-page editor to determine what result files to take as inputs, what data to select from them, what processing steps to apply, and what kind of charts to create from them.

On the upper half of the first page determine the input file that serve as input for the analysis by click on Add File button, the lower half shows what files matched the input specification and what runs they contain, as shown in Figure 4.11.



Add or drag & drop result files (*.sca or *.vec) that should be used in this analysis. Wildcards (*,?) can also be used to specify multiple files.

file /inet/examples/adhoc/fg802112/results/Dsr10Nodes-*.vec
file /inet/examples/adhoc/fg802112/results/Dsr10Nodes-*.sca

Here you can browse the result files and their contents.

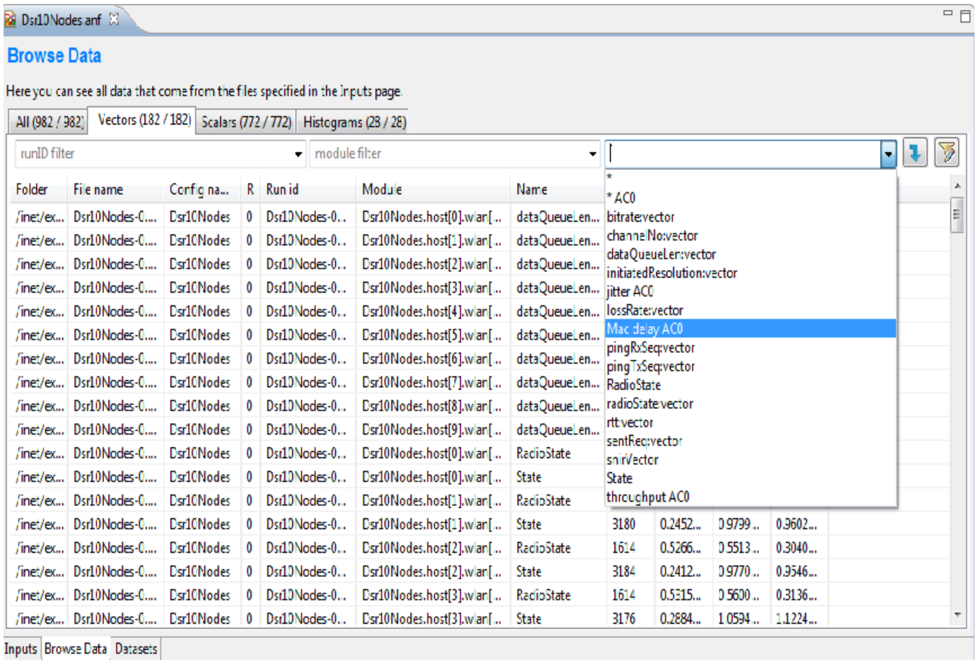
/inet/examples/adhoc/fg802112/results/Dsr10Nodes-0.sca
/inet/examples/adhoc/fg802112/results/Dsr10Nodes-0.vec

```
<terminated> fg802112 [OMNeT++ Simulation] e:\omnetpp-4.3\bin/opp_run.exe (7/2/14 5:06 AM - run #0)
```

Selected 0 out of 28 rows

**Fig 4.18** Determine input files for data analysis.

The second page of the Analysis editor displays results (vectors, scalars and histograms), and the results can be sorted and filtered.

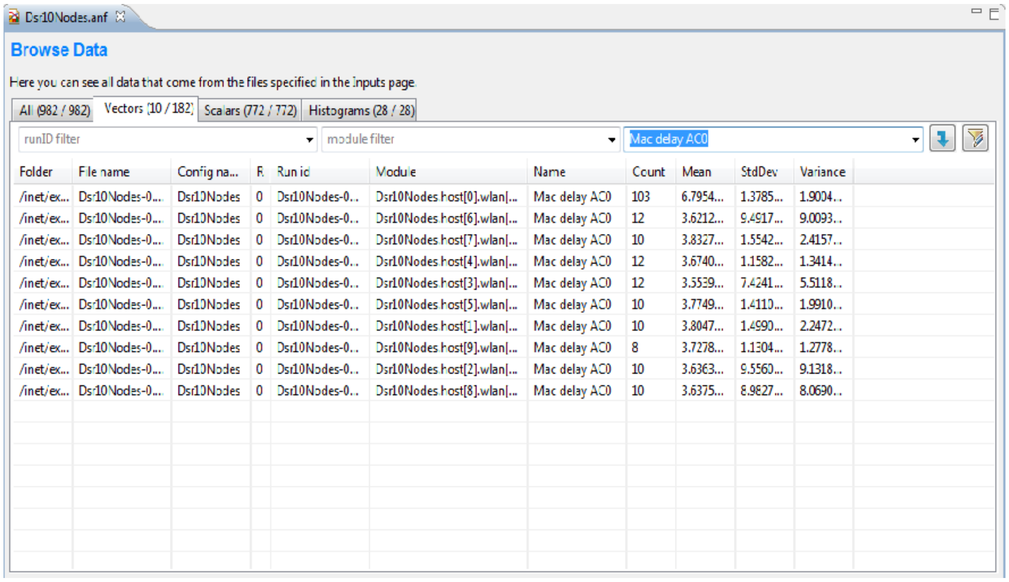


Here you can see all data that come from the files specified in the Inputs page.

| Folder | File name | Config na... | R | Run id | Module | Name | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| /inet/ex... | Dsr10Nodes-0.... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[0].wlan[... | dataQueueLen... | | | | |
| /inet/ex... | Dsr10Nodes-0.... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[1].wlan[... | dataQueueLen... | | | | |
| /inet/ex... | Dsr10Nodes-0.... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[2].wlan[... | dataQueueLen... | | | | |
| /inet/ex... | Dsr10Nodes-0.... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[3].wlan[... | dataQueueLen... | | | | |
| /inet/ex... | Dsr10Nodes-0.... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[4].wlan[... | dataQueueLen... | | | | |
| /inet/ex... | Dsr10Nodes-0.... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[5].wlan[... | dataQueueLen... | | | | |
| /inet/ex... | Dsr10Nodes-0.... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[6].wlan[... | dataQueueLen... | | | | |
| /inet/ex... | Dsr10Nodes-0.... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[7].wlan[... | dataQueueLen... | | | | |
| /inet/ex... | Dsr10Nodes-0.... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[8].wlan[... | dataQueueLen... | | | | |
| /inet/ex... | Dsr10Nodes-0.... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[9].wlan[... | dataQueueLen... | | | | |
| /inet/ex... | Dsr10Nodes-0.... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[0].wlan[... | RadioState | | | | |
| /inet/ex... | Dsr10Nodes-0.... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[0].wlan[... | State | | | | |
| /inet/ex... | Dsr10Nodes-0.... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[1].wlan[... | RadioState | | | | |
| /inet/ex... | Dsr10Nodes-0.... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[1].wlan[... | State | 3180 | 0.2452... | 0.9799... | 0.9602... |
| /inet/ex... | Dsr10Nodes-0.... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[2].wlan[... | RadioState | 1614 | 0.5266... | 0.5513... | 0.3040... |
| /inet/ex... | Dsr10Nodes-0.... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[2].wlan[... | State | 3184 | 0.2412... | 0.9770... | 0.9546... |
| /inet/ex... | Dsr10Nodes-0.... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[3].wlan[... | RadioState | 1614 | 0.5315... | 0.5600... | 0.3136... |
| /inet/ex... | Dsr10Nodes-0.... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[3].wlan[... | State | 3176 | 0.2884... | 1.0594... | 1.1224... |

**Fig 4.19** Browsing vector and scalar data generated by the simulation.

We can filter the simulation result to show the performance of the protocol by different metrics such as Mac delay, to select one of them click on static name filter as shown in Figure 4.13.



Here you can see all data that come from the files specified in the Inputs page.

| Folder | File name | Config na... | R | Run id | Module | Name | Count | Mean | StdDev | Variance |
|---|---|---|---|---|---|---|---|---|---|---|
| /inet/ex... | Dsr10Nodes-0... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[0].wlan[... | Mac delay AC0 | 103 | 6.7954... | 1.3785... | 1.9004... |
| /inet/ex... | Dsr10Nodes-0... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[6].wlan[... | Mac delay AC0 | 12 | 3.6212... | 9.4917... | 9.0093... |
| /inet/ex... | Dsr10Nodes-0... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[7].wlan[... | Mac delay AC0 | 10 | 3.8327... | 1.5542... | 2.4157... |
| /inet/ex... | Dsr10Nodes-0... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[4].wlan[... | Mac delay AC0 | 12 | 3.6740... | 1.1582... | 1.3414... |
| /inet/ex... | Dsr10Nodes-0... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[3].wlan[... | Mac delay AC0 | 12 | 3.5539... | 7.4241... | 5.5118... |
| /inet/ex... | Dsr10Nodes-0... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[5].wlan[... | Mac delay AC0 | 10 | 3.7749... | 1.4110... | 1.9910... |
| /inet/ex... | Dsr10Nodes-0... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[1].wlan[... | Mac delay AC0 | 10 | 3.8047... | 1.4990... | 2.2472... |
| /inet/ex... | Dsr10Nodes-0... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[9].wlan[... | Mac delay AC0 | 8 | 3.7278... | 1.1304... | 1.2778... |
| /inet/ex... | Dsr10Nodes-0... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[2].wlan[... | Mac delay AC0 | 10 | 3.6363... | 9.5560... | 9.1318... |
| /inet/ex... | Dsr10Nodes-0... | Dsr10Nodes | 0 | Dsr10Nodes-0... | Dsr10Nodes.host[8].wlan[... | Mac delay AC0 | 10 | 3.6375... | 8.9827... | 8.0690... |

**Fig 4.20** Select the Mac delay.

To show the Mac delay of the protocol in vector, after selecting the Mac delay meter right click on the host and select plot, the plot will be displayed as shown there in Figure 4.14.

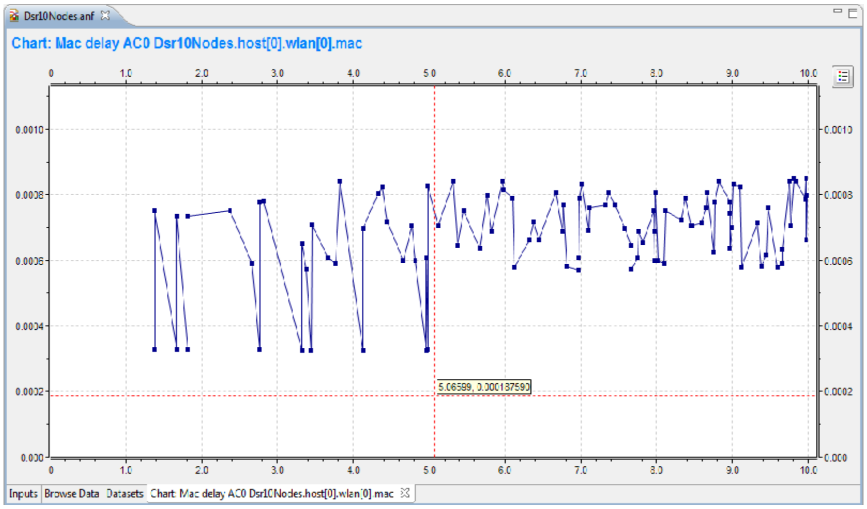


**Fig 4.21** Mac delay of host 0 in Dsr.

## The second protocol running (Aodv in 10 node)

The result is showed in Figure 4.15.

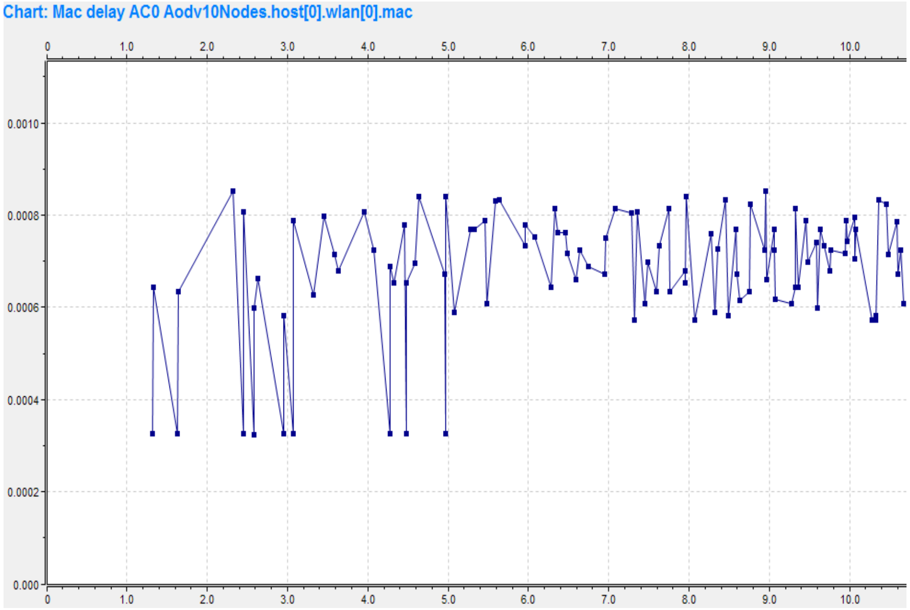


**Fig 4.22** Mac delay of host 0 in Aodv.

## The third protocol running (Olsr in 10 nodes)

Figure 4.16 displays results for the Olsr protocol.

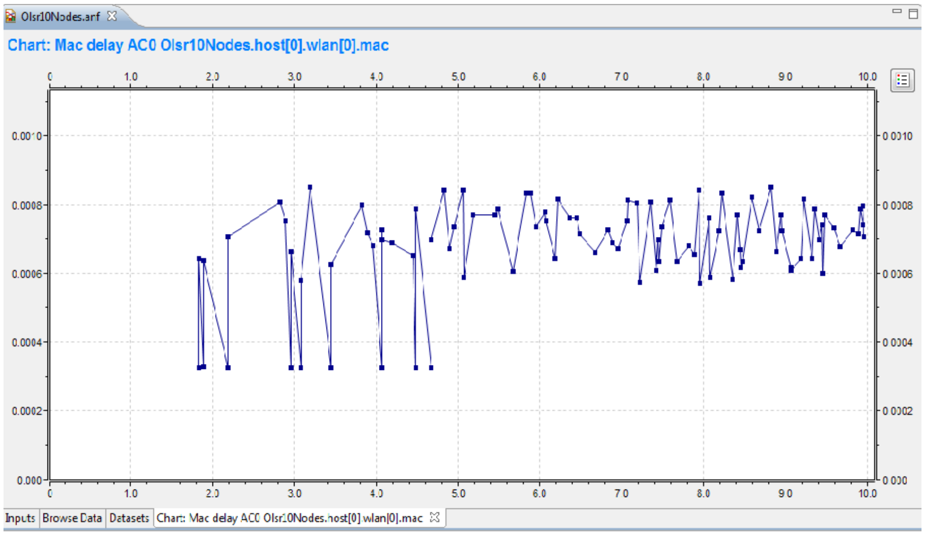


**Fig 4.23** Mac delay of host 0 in Olsr.

## <u>The second test in 30 device</u>

Implementation steps are the same steps as the above.

## The result of first protocol running (Dsr in 30 nodes)

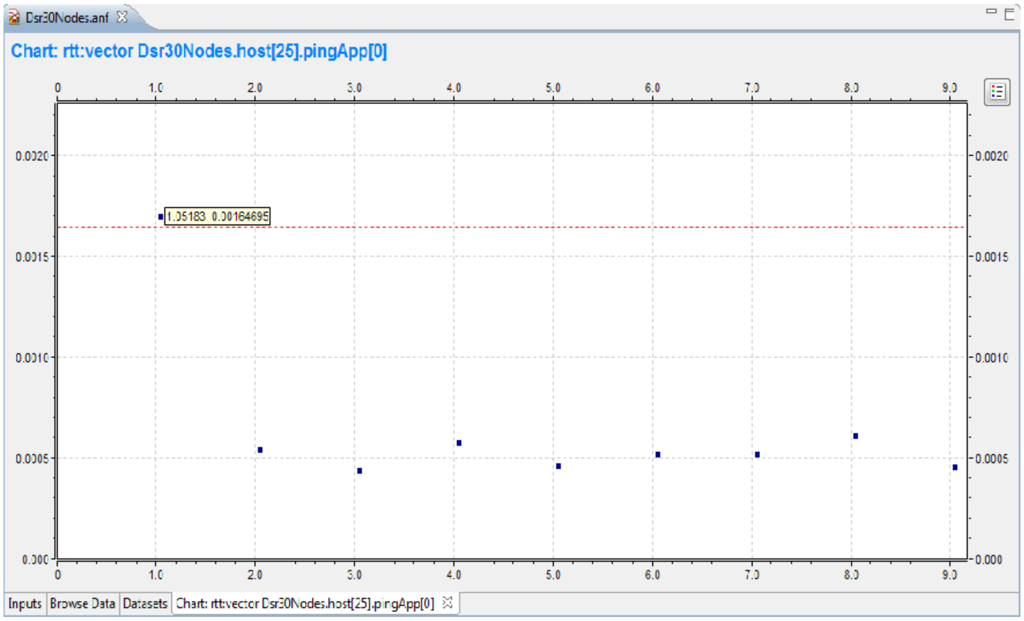


**Fig 4.24** Mac delay of host 0 in Dsr.

The result of second protocol running (Aodv in 30 nodes)

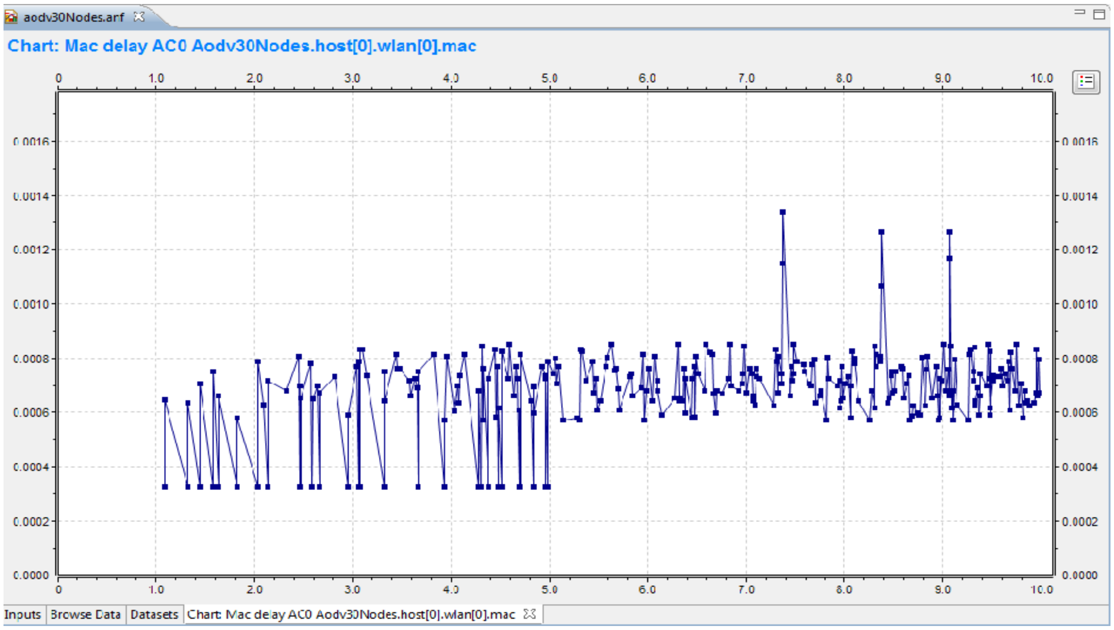


**Fig 4.25** Mac delay of host 0 in Aodv.

## The result of third protocol running (Olsr in 30 nodes)

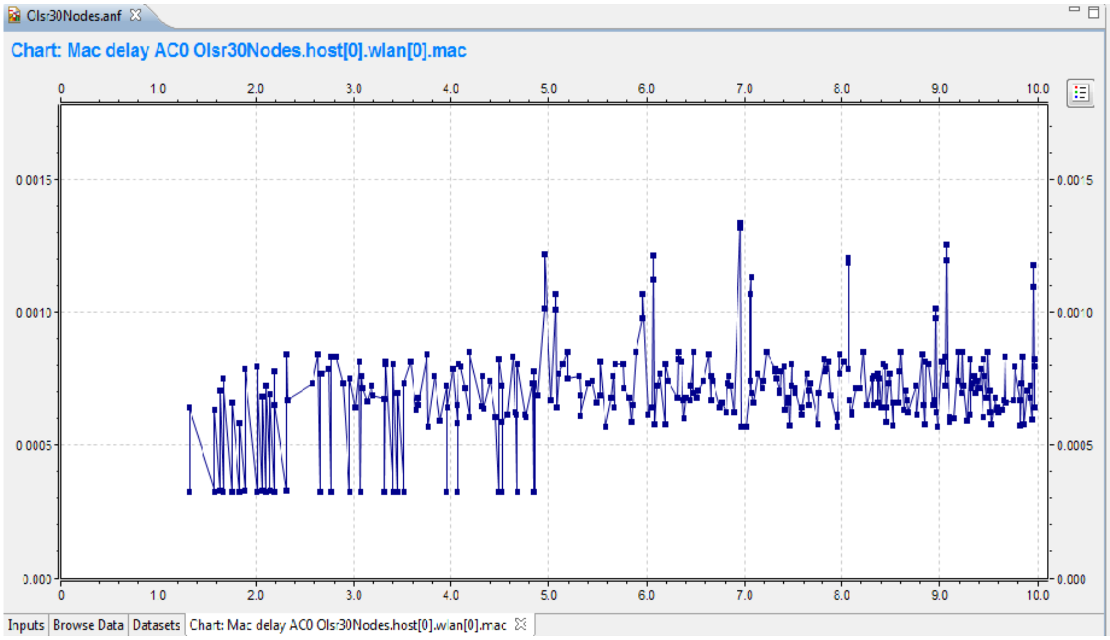


**Fig 4.26 Mac delay of host 0 in Olsr**

After using the program (omnet ++) to simulate the work of protocols and protocol analysis, we find that delays in the transmission process (exchange of information)in (dsr) protocols are more compared to other protocols. Delays are more when the number of devices in a network are getting larger.

*Chapter 5*

# *Summary and Conclusions*

Ad hoc networking can be applied anywhere where there is little or no communication infrastructure or the existing infrastructure is expensive or inconvenient to use. Ad hoc networking allows the devices to maintain connections to the network as well as easily adding and removing devices to and from the network.

The mobile ad hoc network (MANET) is a network formed without any central administration. Therefore, the network nodes have to serve also as routers and hosts. The network nodes are mobile and they are able to communicate wirelessly with each other by sending and receiving data packets.

There are multiple properties associated with MANET networks, including:

- **Dynamic topology:** nodes capable of routing packets, arbitrary transfer and this movement lead to a structural change randomly and rapidly.
- **Limited physical security attacks**: such as denial of service or eavesdropping, are often easier to launch wireless networks from wired networks in dedicated networks, but at least have the advantage of being completely decentralized.
- **Omnidirectional radio broadcasting:** packets are broadcast to all neighboring nodes, at one time for the exchange of data between nodes near each other and this is not possible.

- **Potential unidirectional links:** produce due to the intervention group transfer fading.

# *Bibliography*

1. Jeff Kennington, Eli Clinics, Dinesh Rajan. “Wireless Network Design: Optimization Models and Solution Procedures”.

2. Jun-Zhao Sun. " Mobile Ad Hoc Networking: An Essential Technology for Pervasive Computing "

3. Klaus Nieminen . "Introduction to Ad Hoc Networking"

4. Wenjia Li and Anupam Joshi. "Security Issues in Mobile Ad Hoc Networks - A Survey"

5. Jayesh Kataria, P.S. Dhekne BARC and Sugata Sanyal TIFR. "ACRR: Ad-hoc On-Demand Distance Vector Routing with Controlled Route Requests"

6. Scott F. Midkiff, Chair, Luiz A. DaSilva, Nathaniel J. Davis, Ira Jacobs and Charles P. Koelling. "Mobile Ad-hoc Network Routing Protocols: Methodologies and Applications"

7. Xin Yu. "Distributed Cache Updating for the Dynamic Source Routing Protocol"

8. Baruch Awerbuch & Amitabh Mishra. "Dynamic Source Routing (DSR)Protocol"

9. Xiaoyan Hong, Kaixin Xu, Mario Gerla. "Scalable Routing Protocols for Mobile Ad Hoc Networks”

10. http://inet.omnetpp.org/index.php?n=Main.Download