Implementation of MANET routing protocols on OMNeT++

A. Ariza-Quintana E. Casilari A. Triviño Cabrera

Dpto. Tecnología Electrónica, University of Málaga

Campus de Teatinos, 29071 Málaga (Spain),

Tfno.: 34-952132755; FAX 34-952131447

aarizaq@uma.es ecasilari@uma.es atc@uma.es

ABSTRACT

This work describes how some of the most popular ad hoc routing protocols (DYMO, DSR, AODV) have been implemented for the INET library of OMNeT++ simulator tool. The developed modules have been programmed so they can be easily extensible to include future routing MANET protocols.

Keywords

OMNeT++, MANET, Ad hoc networks, DSR, AODV, DYMO.

1. INTRODUCTION

Simulation is a basic tool for the study and analysis of communication protocols, especially in the case of Mobile Ad Hoc Networks (MANETs). The need of simulation in this field of telematics is founded in two main reasons: Firstly, simulation permits to model and evaluate extensive MANET networks with multiple nodes which are still not existent or are quite difficult (and costly) to implement by means of actual devices in realistic scenarios. The fact is that most examples in the literature of actual ad hoc networks that have been deployed for experimental evaluation just consist of a reduced number of nodes (normally less than a dozen). Secondly, the complexity and the nature of the factors (radio propagation, node mobility, protocol interaction, etc) that impact on the performance of MANETs prevents from defining an analytical mathematical model able of characterising this type of networks without enormous simplifications.

Nowadays there exist many event-driven licensed and unlicensed simulation environments for the emulation of MANETs. By far, the most popular tool is Network Simulator-2 or NS-2 [1], followed by OPNET [2], Ncutns [3] and OMNeT++ [4]. Among all these alternatives, OMNeT++ is gaining an increasing attention by the MANET research community. As the main advantages of OMNeT++ we can mention:

1. It is an open source code tool, so the programmer can add new libraries or adapt the existing ones according to the requirements of the scenario and technologies to emulate.

2. It possesses a friendly and simple programming interface, which helps the development of new code.

3. It presents a modular structure, which eases the implementation of new communication protocols. Thus, the inclusion of a new protocol does no require a perfect knowledge of how other parts of the code have been implemented. On the contrary, the different layers and protocols communicate by means of messages. So, the programmer just requires being on familiar terms with the format of these messages.

4. The OMNeT++ code is especially efficient, which results in an important reduction of the simulation time when compared with other simulation tools (in particular, NS-2).

5. OMNeT++ is widely supported in Internet due to the extension of OMNet++ community.

Nowadays the libraries of OMNeT++ incorporate the UDP/TCP/IP protocol stack as well as different types of Layer-2 MAC (Medium Access Control) interfaces, including IEEE802.11, which is the most utilised layer-2 protocol for the analysis and evaluation of MANETs. Consequently OMNeT++ is an attractive tool to be employed in studies about MANETs. However, the distributed nature of ad hoc networks prevents that classical IP routing protocols (such as OSPF) can be utilised. So, many specific MANET routing protocols have appeared in the literature during last ten years. None of these protocols have been programmed and incorporated in the basic libraries of OMNeT++. In this work we describe the definition of OMNeT++ modules that implement three of the most employed ad hoc routing protocols: AODV (Ad hoc On Demand Distance Vector, [5]), DSR (Dynamic Source Routing [6]) and DYMO (Dynamic MANET On-demand Routing Protocol [7]).

2. IMPLEMENTATION OF AD HOC ROUTING PROTOCOLS

Aiming at incorporating some ad hoc routing schemes in OMNeT++, we have preferred to adapt existing public versions of the protocols (available and intended for other environments) rather than reformulating the programming of the routing algorithms (mainly described in IETF Requests For Comments). In this way, we benefit from the utilisation of a mature code that has been already validated in different scenarios and which notably reduces the implementation time and the programming costs. Additionally, the adaptation of a previously developed software allows a more rigorous comparison of results with those studies that do not employ OMNeT++.

There exist different available implementations of ad hoc routing protocols. However, we considered as the best candidates to be migrated to OMNet++ those that were conceived to work with the Linux Kernel. These versions render possible the definition of actual ad hoc networks and are quite employed to set up small and

OMNeT++ 2008, March 3, 2008, Marseille, France.

economic prototypes consisting of several linux-equipped portable devices (mainly PDAs, laptops or embedded PCs). The adoption of any of these implementations in the OMNet++ environment would permit to contrast the simulation results with those generated by a realistic MANET application. In this sense, one of the most popular and employed versions of AODV and DSR for the definition of actual MANETs is the software developed by the University of Uppsala [8]. So, this open-source code was selected as the base software to integrate these routing protocols in OMNet++. Similarly, for the case of DYMO, we opted for the Linux software designed in the University of Murcia [8]. This version of DYMO was derived from the implementation of Uppsala (AODV-UU) which eased its integration in OMNet++ once that we adapted AODV and DSR to this simulation environment.

2.1 MANET Manager Module

For the inclusion of the MANET routing protocols in OMNet++ we have defined (see Figure 1) a new module (*manetRouting*) which includes in turn a managing module (*Manet Manager*). Figure 2 shows in a schematic way the relationship between the created MANET routing module and the (IP) network layer of OMNet++ through the *Manet Manager*. This managing module, apart of triggering the ad hoc routing procedures, will be in charge of controlling the IP tables. Thus, for example, at the beginning of the simulation the *Manet Manager* will erase all the entries in the IP tables so that the network layer will be forced to transfer any generated packet to this new module, which will initiate a route request to find an ad hoc path to the destination node of the packets. The process of route request (and route reply) will be accomplished by the corresponding module that implements the active ad hoc routing algorithm. For this purpose, the adapted implementations of the protocols were included as three different modules (one for each protocol) depending as 'leaf' modules of the *Manet Manager*. Once the routes are created by the routing protocols the *Manet Manager* will update the IP tables accordingly. In any moment, the route discovery process is triggered if the IP layer does not find in the IP tables any adequate entry for an incoming packet. As DSR, AODV and DYMO are incorporated in different modules, just the module of the routing protocol to be employed is initiated. This optimises the utilisation of the resources in the simulation equipment as long as no copies of the unused modules are kept active in the system memory.

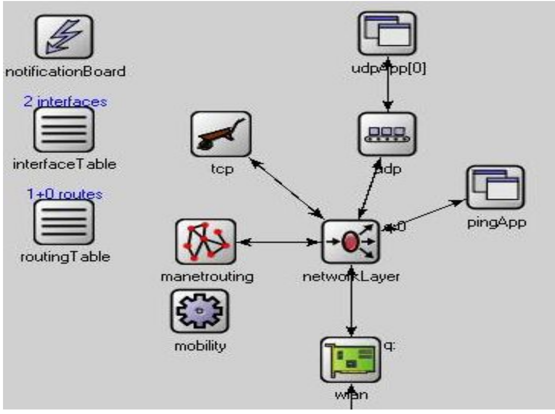

Figure 1. Connection of the ad hoc routing module to the network layer

2.2 Message formats and protocol operation

As the basic event in OMNet++ is the message, the first step to develop the new MANET modules is to define and configure the formats of the messages that each protocol will utilize. The implementation of the messages is based on C language structures ('structs'), so that every field of the message is characterised as an element of the consequent 'struct', which will be accessed by the code. All these fields are defined as the public attributes of the C++ classes that describe the messages.

The decision of codifying the packets by means of C structures (instead of employing the proprietary OMNeT++ message format) was made in order to minimize the changes in the adaptation of the original ad hoc routing code and also to improve the adaptability of the code to other simulation tools, such as ns2.

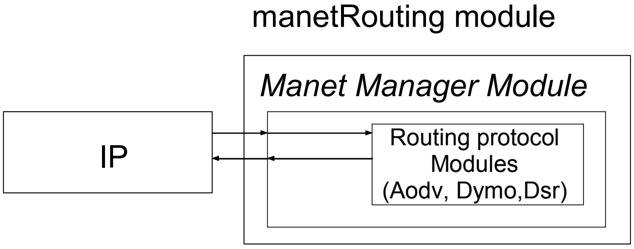

Figure 2. Scheme of the management module of the ad hoc routing protocols. Every routing protocol is executed as 'leaf' module of the *Manet Manager*

All the created messages (summarised in Table 1) have been defined in a similar way and include a special field that permits to transport information in a binary vector. This field is employed to convey the specific protocol data related to the MANET routing algorithms. The copy operators are in charge of duplicating this information as the packets (and the OMNet++ messages) progresses through the ad hoc network.

Table 1. Defined messages for the Routing Protocol Modules

Protocol	Class from which it is derived	It provides:	Type of Message
AODV	`cPacket`	TLV Fields	RREQ, RREP, RRER, HELLO
DYMO	`cPacket`	TLV Fields	RE, UERR, RRER, HELLO
DSR	`IPDatagram`	Header Dsr	RREQ, RREP, RRER, Source Forwarding.

Note: TLV (Time Length Value)

In the case of AODV and DYMO, when the IP layer receives a packet it searches the destination in the corresponding IP table contained in the *routingTable* module (see Figure 1). Provided that no adequate entry exists in the table for the packet, it will be sent to the *manetRouting* module. At this point, this routing module buffers the packet and executes the ad hoc route discovery process in order to find an active path to the destination. As soon as a valid route is discovered, the IP routing table is updated and the packet is transferred to the IP layer so it can be transmitted to the following hop. On the other hand, DSR is a source routing scheme and proposes an alternative routing to the IP layer as long as it includes in the packet header the whole set of IP addresses of the nodes that the packet must traverse to reach its destination. In this case, the *Dsr* module does not modify the IP routing table. Alternatively, all the possible paths to a destination from a source nodes are stored in a specific internal structure. So, the IP-layer does not perform during the routing of a packet and this operation is entirely supported by the Dsr module.

To improve the efficiency in the utilisation of the memory by the software, for the case of DSR, a common stack is employed by the Dsr modules of all the MANET nodes. The goal of this stack is to reserve a memory space for the structures that are required to be defined for the management of the packet transmissions. So, when a new packet must be sent, one of the structures is extracted from the stack. Similarly, as soon as the packet is received and its corresponding structure is not necessary any longer, the structure is reinserted in the stack instead of being erased. Just in case that the stack is empty, the program would require to reserve more memory space to continue the simulation. Making so and just with a small stack of structures, the number of operations of memory reservations (and consequently the simulation time) can be significantly reduced. In particular, this fact has been checked by means of different simulations with networks of 100 nodes, heavy traffic conditions and a stack of only 30 structures.

The wireless and cooperative nature of ad hoc networks also required a modification of the search mechanism in the IP tables. In a conventional IP device this search is mainly based on the IP prefix. Conversely, in the case of MANET routing, the IP address of the incoming packet must coincide completely with one of the IP-table entries. In a 'classical' IP routing procedure, the goal is to determine the output interface through which the received packet must be transmitted. However, in most MANET nodes the output interface is unique (e.g.: an 802.11 interface) and the router basically must decide which is the IP address of the neighbour node to which the packet must be transferred. Once this address is found in the IP table, the corresponding MAC address of the neighbour is resolved by means of an Address Resolution Protocol (ARP or ARP-like protocol).

2.3 Protocol Extensions

Apart from the adaptation of the existing code of the routing protocols, several optional extensions have been incorporated in the developed software.

In the case of DSR, the mechanism of cost assignation ETX [10] has been included. This mechanism proposes to evaluate the quality (and availability) of the bidirectional connection between neighbour nodes so that the link quality can be taken into account in the routing decision.

To configure alternative routes to the same destination, nodes are allowed to respond to the same route solicitation if it is received through different paths. Therefore, if a Route Request (RREQ) is received after a previous RREQ from the same origin has been responded, a node can decide to send a new Route Reply (RREP) message to the origin to build up a different path. In order to avoid long paths and to reduce interferences and power consumption in the nodes, this decision is assumed to be taken if the new detected path has a lower number hops (or optionally a lower cost) than the previously configured route. Additionally, for DSR, the software incorporates the technique of path-cache. Similarly to the link-cache policy (already implemented in the version of Uppsala), the path cache allows the utilisation of alternative paths when the active route is broken, but without the need of being continuously calculating the shortest path (e.g.: through Dijkstra algorithm). This reduces the computational cost.

For the implemented version of DYMO, the software has been extended so that nodes can learn new ('gratuitous') routes activating a 'promiscuous' mode at the MAC layer, that is to say, by listening and reading all the Route Reply (RREP) packets received at the layer-2 although they are not forwarded to their own IP address. In this sense, as 'sniffed' RREP packets in DYMO (and AODV) do not indicate explicitly indicate the IP address of the neighbour that originated it, the node in promiscuous mode must utilise an ARP-type table that permits to match the MAC address of the neighbour and its corresponding IP addresses. This will allow the promiscuous node to passively discover new routes to the nodes mentioned in the RREP packet.

3. CONCLUSIONS

This work has described the adaptation of diverse implementations of ad hoc routing protocols so they can be incorporated in Omnet++ INET library. In particular, the adaptation has been focused on the protocols AODV, DSR and DYMO, which are among the most employed routing schemes in studies and prototypes concerning MANETs. The developed modules have permitted to integrate in a powerful and popular simulation tool an existing and validated software that is presently performing in actual implementations of ad hoc networks. Therefore, the simulations with OMNeT++ could be compared even with the results obtained with real MANET testbeds. In order to maintain the potentiality and compatibility of the INET library, the programmed code has just slightly modified this library. In this sense, the modifications are limited to the Ip.cc module (to enable the sending of ad hoc packets and the search in the IP routing tables according to the ad hoc protocols) as well as to the Ieee802.cc module (with the goal of permitting link layer feedback). The developed software is publicly available at [11].

4. ACKNOWLEDGMENTS

This work was partially supported with public funds by the Spanish National Project No.TEC2006-12211-C02-01.

5. REFERENCES

[1] Network Simulator 2, http://www.isi.edu/nsnam/ns/

[2] Opnet, http://www.opnet.com

[3] NCTUns, http://NSL.csie.nctu.edu.tw/nctuns.html.

[4] OMNeT++, http://www.omnetpp.org

[5] Perkins, C., Royer, E. and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", RCF-3561, IETF, 2003.

[6] D. Johnson, Y. Hu, D. Maltz "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4", RCF-4728, IETF, 2007.

[7] I. D. Chakeres and C. E. Perkins "Dynamic MANET On-demand Routing Protocol". IETF Internet Draft, draft-ietf-manet-dymo-10.txt, July 2007 (Work in Progress).

[8] Implementation of AODV and DSR routing protocols University of Uppsala (Sweden), available at: http://www.it.uu.se/research/group/coregroup/software

[9] F. Ros, Implementation of DYMO routing protocol, University of Murcia (Spain), available at: http://masimum.dif.um.es/?Software:DYMOUM

[10] D. S. J. De Couto, D. Aguayo, J. Bicket, Robert Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing", Mobicom 2003, San Diego, 2003.

[11] A. Ariza, Implementation of Ad hoc routing protocols for OMNet++, University of Malaga, software available at: http://webpersonal.uma.es/~AARIZAQ/