Implementation and Performance Evaluation of MFlood protocol for Mobile Ad Hoc Networks

AL-Gabri Malek[1], LI Chunlin[1], Wang Bo [2]
[1]Department of Computer, Wuhan University of Technology, Wuhan (430000)
[2]Wuhan Universities, Wuhan (430000)
E-mail:malekye@gmail.com

Abstract
In this paper, we adding MFlood protocol to NS2 and analyze the features and models of wireless network in details; discuss the simulation process and implementation of wireless network on the basis of NS2, also MFlood protocol and AODV protocol self-defined in wireless network. Nam is used to display the process of simulation. Gnuplot and Gawk Tools are used to produce plot data files and icons to clarify the communication process.NS2 is a simulation tool designed specifically for communication networks. The main functionalities of NS2 are to set up a network of connecting nodes and to pass packets from one node to another.A detailed introduction to the design, implementation, and use of network simulation tools are presented. The requirements and issues faced in the design of simulators for wired and wireless networks are discussed. Abstractions such as packet- and fluid-level network models are covered.
Keywords: network simulation; MFlood protocol; AODV protocol; Ad Hoc network; protocol analysis

0 Introduction

Network simulation software for network research can provide researchers with a convenient, efficient verification and analysis tool [1]. At present, there is a lot of well-known network simulation software such as MIL3's OPNET software, Cadence's VCC software and so on. But they are too expensive for research units and institutions. NS2 is free and open software, and very usefully [2].

Ns began as a variant of the REAL network simulator in 1989 [3]and has evolved substantially over the past few years. In 1995, ns development was supported by DARPA through the VINT project at LBL, Xerox PARC, UCB, and USC/ISI. Currently, ns development is support through DARPA with SAMAN and through NSF with CONSER, both in collaboration with other researchers including ACIRINs has always included substantial contributions from other researchers, including wireless code from the UCB Daedelus and CMU Monarch projects and Sun Microsystems [4].

NS are used widely in network research to verify new protocols, modifications to existing protocols and new ideas [5]. The tool used in many cases is NS2, which provides a wide range of protocol implementations and simulation tools. The nature of the NS2 protocols means that they are often based on theoretical models and possibly lacks the behavior of real networks[6][7].

NS2 is an object-oriented simulation tool to simulate and analyze network elements; it is also a powerful tool to develop new protocol and scheme.

About this paper, we firstly, introduce about NS2, secondly, simulation of wireless network (AODV, MFlood), thirdly, environment of NS2, fourthly, talk about implementation of NS2,fifthly,talk about Comparative between AODV and MFlood Protocols and finally, talk about conclusion.

In this paper, the simulation and realization of wireless network are demonstrated and AODV protocol and MFlood protocol are simulated. Finally, the simulation results are presented and analyzes.

1 Simulation of Wireless Network

There are three general approaches to performance evaluation [8]: (1) prototyping: build it (or a scaled down version of it) and see how it works; (2) analytical modeling: build a mathematical model of it and use it to analyze the system; and (3) simulation: build a software model of the system. Prototyping is often not feasible, or time consuming especially for large scale systems; it also provides limited controllability and observability...

The establishment of special components can be composed of the required specific application, business flow, agents, links, routing and node model [9]. These components are required to be tested and compiled so as to ensure whether they can be correctly used [10].

After the completion of the simulation, the trace files should be analyzed to obtain valuable information, also, Nam can be used to monitor the network simulation process. The analysis results from simulation can help to decide whether it needs to change the configuration topology and business simulation so as to trigger other simulations for relative objective simulation results [10].

NS2 [11] is used to simulate the transmission performance of MFlood protocol in wireless network, where the MAC protocol is based on 802.11 and the propagation model is TwoRayGround. The objectives and the process of the simulation are as follows:

First, the scenes and data flow are simulated according to the requirements, where at random moment, two arbitrary nodes can be chosen for communication and the nodes maintain moving state. Due to the limitation of communication distance and network bandwidth, some packet may be lost. Furthermore, the performance of self defined MFlood protocol can be demonstrated on the basis of the analysis results of bandwidth and packet loss of AODV protocol and self-defined MFlood protocol[12].

2 Environment of NS2

We use the following tools cbrgen data stream to generate a random scenario, 25 nodes in the random selection of 20 pairs of nodes, start the 20 cbr data streams, each stream per second was born the size of a 512-byte packets. Mflood.tcl in our tests, the wireless network transmission rate is only 2Mbit / s, so now the network is a very heavy load. Order to generate data flow scenario is as follows [12]:

Steps for creating scenario files for mobility simulations:

1. Go to ns directory and run "configure" (you probably have done that already while building ns). This creates a makefile for setdest.

2. Go to indep-utils/cmu-scen-gen/setdest. Run "make", which first creates a stand-alone object file for ~ns/rng.cc (the stand-alone version doesnot use Tclcl libs) and then creates the executable setdest.

3. Run setdest with arguments as shown below: ./setdest [-v version of setdest; 1. for original 1999 CMU version or 2. for modified 2003 U.Michigan version] [-n num_of_nodes] [-p pausetime] [-s maxspeed] [-t simtime] [-x maxx] [-y maxy] > [outdir/movement-file]

ns cbrgen.tcl –type cbr –nn 25 –seed 1 –mc 20 –rate 4 >cbrn25m20r4

And ns setdest -n 25 –p 120 -M 15 –x 1500 –y 300 > snp25p25M10

In 1500 * 300 square scenes, there are 25 nodes which have random speed and direction with maximum speed of 20 m/s, and duration time of 900s.

Initiated, every flow per second can produce a data packet of 512 bytes.

3 implementation of NS2

According to the tradition of general protocol, source files such as in tcl/lib/ns-mobilenode.tcl NS2 2.29

MFlood install method:

i. modify ns-lib.tcl

When a wireless node is created, its default routing agent is specified in the arguments, such as opt(rt). In /tcl/lib/ns-lib.tcl. the **"create-wireless-node"** procedure will check the routing protocol type. Here we can add a new type as "FIXRT" in the "switch" block. Also add "set ragent [$self create-fixrt-agent $node]". For example,[12]

As we add wireless ad hoc network routing protocols under the NS running Tcl specified routing protocol will only be used when in fact the name of routing protocols (that is, a string, for example, correspond to our agreement MFlood) , we need to amend the NS system Tcl code is set when the routing protocol will call for MFlood increase in our new Agent / MFlood category (that is, C + + under MFlood category). This need to modify ~ ns / tcl / ns-lib.tcl documents Simulation category in the create-wireless-node member functions (the function is defined as the Simulation instproc create-wireless-node args)

Add the following code

```
Switch-exact $routingAgent_{;# Original code, in accordance with set of routing
                                protocols called
                             ; # Different initialization function agreement
                           }
```

```
MFlood {
    set ragent [$self create-mflood-agent $node]
}
```

This means that when the wireless mobile node for the specified routing protocol MFlood will call create-mflood-agent member function to initialize. Then we in the ~ ns / tcl / ns-lib.tcl document the definition of create-mflood-agent: a member function, the code is as follows [13]:

```
}
Simulator instproc create-mflood-agent {node} {
        set ragent [new Agent/MFlood [$node id]]
    $node set ragent_ $ragent
    return $ragent
}
```

From the create-mflood-agent function definition can be seen, created the Agent / MFlood of an object, the object will eventually be a mobile node and $ node bind together. NS modify the system in the Tcl file, a useful technique is to have a reference code NS. For example, we add a routing protocol MFlood category, in the context of these systems can be configured to have reached an agreement AODV reference code, such as the search string "AODV", and then add AODV in accordance with the agreement to add our own MFlood agreement. In this way, we do not need to know the whole NS system Tcl process can achieve its own aims, but also conducive to a targeted understanding of the part we are concerned [11].

ii. modify ns-packet.tcl

This definition of a name for the Baotou MFlood. In ~ ns / tcl / lib / ns-packet.tcl listed a number of defaults will be used in Baotou, in which we need to add the name of Baotou Mflood [11]. Add the name of the needs and mflood.cc document in front of the definition of PacketHeader / MFlood line

Here are some of the code:

```
foreach prot {
    MFlood
AODV
……
}
```

iii. modify packet.h

In order to create a package type MFlood (Note MFlood baotou above is not the same concept, they can also use different names), we need to modify ~ ns / common / packet.h, increase the definition of packet type, first of all, we need to enum packet_t that we add a new type, the definition of PT_MFLOOD. Then, the class p_info category p_info () constructor to add new packet type is given his name:

```
name_[PT_MFLOOD]="MFlood";
```

Incidentally, add a new category and new Agent header format, in accordance with the NS2 programming examples will be relatively simple to do and norms, of course, can be proficient in NS in accordance with their needs for expansion and modification of [12].

iv. modify Makefile

At the conclusion of the Agreement and the realization of the definition of (the concrete realization of the code see appendix), we have to increase to the new file to compile and go to connect to the NS. This need to modify ~ ns / Makefile file, an increase of the new type of compiler.

Our paper on the ~ ns / flood directory in the Makefile in the definition of variables OBJ_CC add the following line [13]:

```
mflood/mflood.o mflood/mflood-seqtable.o\
```

v. at NS directory:

create new file for mflood, after that copy all the files mflood.cc、mflood.h、flood-packet.h、mflood-seqtable.cc、and mflood-seqtable.h inside this file.and add this code

```
else if (strcasecmp (argv[1], "port-dmux") == 0) {
        _TclObject *obj;
        _port_dmux_ = (NsObject *) obj;
        _return TCL_OK;
    }
}
return Agent::command(argc, argv);
}
```

In mflood.cc file.

In file mflood.h add this line **#define _NOW_ (Scheduler::instance().clock())** and **NsObject *port_dmux_;**

```
private:
    u_int32_t myseq_;
```

};

#endif

The self-defined routing protocol class here is MFlood. For the purpose of carrying out retransmission packet detection (that is, each packet forwards only for once), it also maintains long routing table to record the previous-send packet. The routing table class is MFlood_Rtable, which is consist of routing table items (MFlood_RTEntry class), and each MFlood_RTEntry item records the data packets

Transmitted for each data source. MFlood protocol maintains a sequence number SEQ for each node. If data source sends a packet, then SEQ will add 1 and SEQ will be added to packets header information, from which other nodes can decide whether they have transmitted the packets [14].

According to the tradition of general protocol, source file such as mflood.h /.cc (the definition and realization of the Protocol) should be created. Below is the definition of MFlood Routing Protocol, in which recv function and command function are inherited from Agent class. This kind of inter-layer relationship has been defined in Tcl code of NS2.After the completion of test steps, the performance of the protocol can be analyzed through the experimental data statistics.

4 Comparative between AODV and MFlood Protocols

Through the simulations of AODV protocol and Mflood protocol, the relevant working information is demonstrated. For instance, the pdfraction situation of wireless network bandwidth is shown in Figure 8.

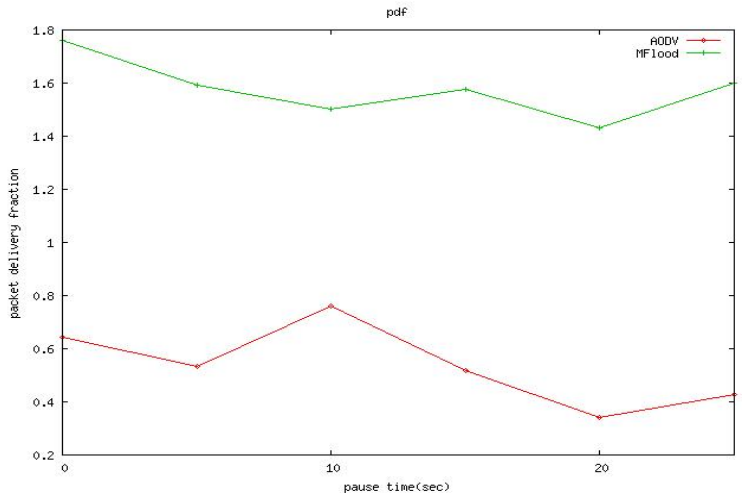

Figure 8. Pdfraction bandwidth of two protocols

Based on figure 8, we evaluated the performance of the algorithms in terms of Packet Delivery Fraction (PDF) by keeping the speed constant and varying the pause time. Initially scenario has been setup for a small network of 25 nodes. As depicted in Figure 8, all the two protocols AODV and MFlood become even more crucial when the pause time of nodes is low, and pause time is very small. Mflood performance is better than AODV. Simulations of wireless network with two different routing protocols are introduced to explain the simulation method of NS2 and the analysis of the simulation results is presented. Figure 8 displays the broadcast packet when AODV changes the routing information. It is seen that AODV transfers packets in the form of unicast. After each node receives the broadcast routing information, the routing mode will be recorded in

its routing table. When data packet waits for transmission, each node needs to check its routing table and send the packet to the designated node until that packet reaches the destination or the packet is lost. Only when exchanging the routing information, will AODV transfer packets in the form of broadcast.

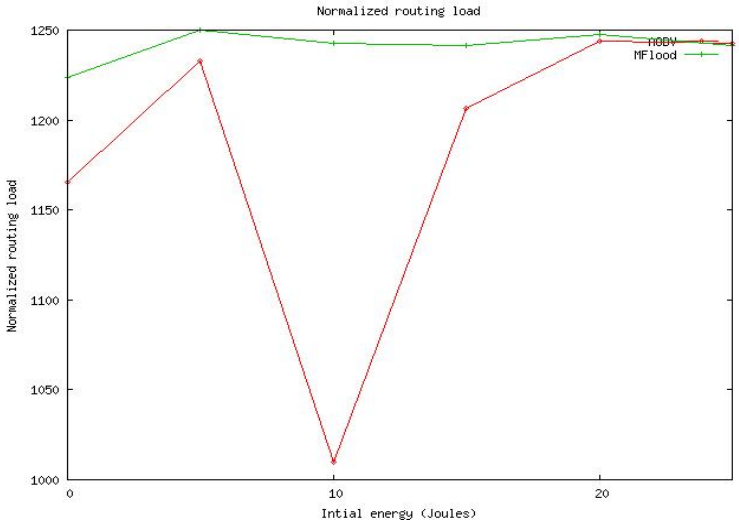

Figure 9. Total energy consummation of two protocols

As shown in Figure9 the energy consummation of AODV is lower than the energy consummation of MFlood. So Energy consummation is a critical parameter in the MANET. Battery-powered devices try to conserve energy by transmitting only when absolutely necessary. An attacker can attempt to consume batteries by requesting routes or forwarding unnecessary packets to a node. [14]

It can be seen from the above figures, in MFlood protocol, the data are transferred in the form of broadcast. When one node sends data packet, it will broadcast this data packet to all the nodes around. Then the node accepted the packet will check the data packet, if it has not sent the packet and the packet destination is not itself, the node will forward the packet in the form of broadcast again. This process will last until the data packet reaches its destinations or the packet is lost for unreachable destination, the process is quite favorable for the work mode of the self-defined protocol in this paper.

5 Conclusion

On the basis above, a number of different topology and data flow are alternated for the simulation experiment and the results are nearly the same. In aspects of pdfraction bandwidth and packet loss rate, AODV protocol overwhelms MFlood protocol. Especially in the aspect of packet loss rate, it is not like the traditional thought which considering that the packet loss rate of broadcast is lower than that of unicast.

On the contrary, the packet loss rate of unicast is much lower. However, in high-speed movement circumstances, the packet loss rate of MFlood keeps almost unchanged.

Simulations of wireless network with two different routing protocols are introduced to explain the simulation method of NS2 and the analysis of the simulation results is presented.

NS2 is an object-oriented simulation tool to simulate and analyze network elements; it is also a powerful tool to develop new protocol and scheme.

NAM is used to display the process of simulation. Tools such as Gnuplot and Gawk are used to produce plot data files and icons to clarify the communication process and make comparison.

References

[1] Yuh-Ren Tsai and Shiuh-Jeng Wang ,2007. Two-tier authentication for cluster and individual sets in mobile ad hoc networks doi:10.1016/j.comnet.2006.06.010
[2] S. Keshav , 1997.Cornell University http://www.cs.cornell.edu/skeshav/real/overview.html
[3] Wang, S. Y., and Kung, H. T., 2002. A new methodology for easily constructing extensible and high-fidelity TCP/IP network simulators, doi: 10.1016/S1389-1286(02)00254-2
[4] FastRunner,2008. Network Simulator - Event simulator targeted at networking research.
< http://www.icewalkers.com/Linux/Software/535070/Network-Simulator.html>
[5] Sam Jansen and Anthony Mcgregor. 2008. Static virtualization of C source code. Doi: 10.1002/spe.v38:4
[6] Sam Jansen and Anthony McGregor. 2006. Performance, Validation and Testing with the Network Simulation Cradle. Doi : 10.1109/MASCOTS.2006.40
[7] Sam Jansen and Anthony McGregor,2005. Performance, Validation and Testing with the Network Simulation Cradle < http://portal.acm.org/citation.cfm?id=1158096 >
[8] Jan Kwiatkowski, Marcin Pawlik and Dariusz Konieczny ,2008.Comparison of Execution Time Decomposition Methods for Performance Evaluation ,doi: 10.1007/978-3-540-68111-3_123
[9] Shudong Chen, Zengde Wu, Wei Zhang and Fanyuan Ma,2004. PBiz: An E-business Model Based on Peer-to-Peer Network,doi: 10.1007/b97162
[10] Tarek H. Ahmed ,2005. Simulation of Mobility and Routing in Ad Hoc Networks using Ant Colony Algorithms,doi: 10.1109/ITCC.2005.257
[11] Ros, F.J. and Ruiz, P.M,2004. "Implementing a New Manet Unicast Routing Protocol in NS2
<http://masimum.dif.um.es/nsrt-howto/html/nsrt-howto.html>
[12]Ke,C.H,2004."using-MFLood-in-ns-2.27environment
<http://140.116.72.80/%7Esmallko/ns2/mflood.htm >
[13] zhangyang,2008.Add mflood in ns2 < http://qzone.qq.com/blog/25276109-1216995005 >
[14] Jianxin Wang, Bei Peng and Weijia Jia ,2004.Design and Implementation of Virtual Computer Network Lab Based on NS2 In the Internet,doi: 10.1007/b98796