Routing Protocols Simulation of Wireless Self-organized Network Based on NS-2

Qian CAI

School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou Jiangxi, China 341000

Keywords: NS-2, Network simulation, Self-organized network, Routing Protocol.

Abstract. This paper carries out simulation in terms of table-driven routing protocol and source-initiated on-demand protocol that are typical of DSDV, DSR and AODV based on NS-2 network simulation tools according to different routing discovering strategies. Then through setting different simulation scenarios, this study contrasts performance difference of average end-to-end delay of the three protocols in the case of gradually increasing the number of mobile nodes and improving movement speed. It finally analyzes the characteristics and the advantages and disadvantages of the three routing protocols.

Introduction

Ad Hoc, as a kind of wireless self-organized multi-hop wireless network, is composed of a series of mobile nodes without base station support. It features self-organizing, invulnerability and dispensable with fixed infrastructure support, so it is widely applied in various fields. Due to different communication efficiency of routing protocols for Ad Hoc network, it is very important to choose the appropriate routing protocol for fast communication of the network.. According to the characteristics of mobile Ad Hoc network, MANET has developed many unicast routing protocol based on different strategies, which can be ranged among table-driven routing protocol, source-initiated on-demand protocol and hybrid routing protocol according to difference of driving modes. Frequently used table-driven routing protocol and source-initiated on-demand protocol are shown as in Fig. 1.

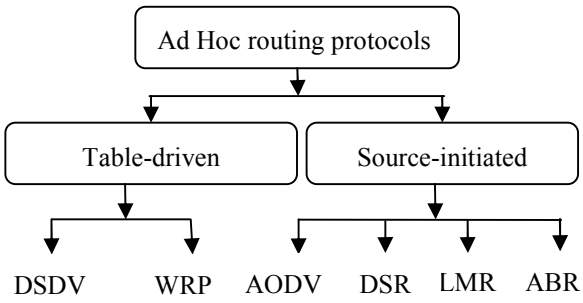

Figure 1 Ad Hoc Network Routing Protocols Classified According to Driven Mode

Table-driven Routing and Source-initiated On-demand Protocols

In table-driven routing protocol, each node maintains a table that contains routing information reaching to other nodes. When change has been detected in the topology structure of the network,

the node will send update message. Once receiving the message, the other nodes will renew their own routing tables instantly. As consistent, timely and accurate routing information are maintained, the routing table can precisely reflect the topology structure of the network.

Once the source node sends a message, it will immediately obtain the routing available to reach the destination node. Therefore, the time delay of this routing protocol is small, but the overhead is big. DSDV (Destination Sequenced Distance Vector) is a typical instance of such a protocol.

The routing tables of DSDV protocol contain information about all possible destination nodes and distance to reach them. They manage to maintain the connectivity of network nodes with periodic broadcasting in the network. Their entries include destination nodes, hop count and destination sequence number. The destination sequence number is assigned by the destination node, mainly used for judging whether the routing is out of date and preventing routing loops.

Source-initiated on-demand protocol is a routing algorithm which looks up the routers when in need of sending data. In this routing protocol, the nodes do not need to maintain routing information accurately and in time. Only when the destination node sends a message, the source node starts lookup process in the network routing till it finds the corresponding routing. Compared with the table driven routing protocol, the overhead of on-demand routing protocol is smaller, but the packet transfer delay is longer. This kind of protocols includes DSR (Dynamic Source Routing) and AODV (Ad Hoc On-demand Distance Vector Routing).

DSR uses source routing algorithm rather than hop routing method. In DSR, each node has a high-speed buffer used to store all routes of which the destination nodes are known. DSR mainly includes two processes: route discovery and route maintenance. When node S sends data to node D, it first checks whether there is any route that is not out of date to the destination node in the cache. If so, it will directly use the available routing. Otherwise, it will start the routing discovery process.

AODV employs distributed routing based on routing table and uses three major messages: routing requests (RREQ), routing response (RREP) and routing error (RRER). AODV is a combination of DSR and DSDV: firstly, borrows the base of route discovery and maintenance from DSR; secondly, it employs DSDV hop-by-hop routing, ordinal number and the periodic updating mechanism in the phase of routing maintenance.

A Brief Introduction to NS-2 Simulation Software

NS-2 (Network Simulator, version 2) is currently one of the mainstream, open source Network simulation software, developed by UC Berkeley University in 1989, which provides a strong support for simulation of TCP, routing and multicast protocols on both wired and wireless networks. Because of its open source—all source codes are free and open, anyone can obtain, use and modify them. Researchers all over the world extend and update its function every day, adding new protocol support and functional modules. It is also research one of the most widely used network simulation software in the current network field.

The essence of NS-2 is an object-oriented and discrete event driven simulator, which uses virtual clock and whose simulations are all driven by discrete event. It can be used for simulation of different communication networks as well as having powerful functions and rich modules. some of the simulation of the modules that have been implemented are: network transmission protocols, such as TCP and UDP; traffic source flow generator, such as FTP, Telnet, Web CBR and VBR; routing queue management mechanism, such as Droptai, RED and CBQ; routing algorithms, such as Dijkstra; and wireless networks such as WLAN, Ad hoc routing, mobile IP and satellite

communication network, etc. NS-2 has implemented multicast and some of the MAC sub-layer protocols for simulation on local area network (LAN).

As NS-2 employs C + + and OTcl as development language, it achieves convenient and rapid configuration of various components and parameters such as simulation topology, nodes and links by means of simulation codes writing with the easy-to-use Tcl/OTcl script. NS is actually an OTcl script interpreter, which contains simulation event scheduler, network component object library, etc. Event scheduler controls the simulating process, activates current event in the event queue at an appropriate time and performs the event. Network components simulate communication between network devices or nodes, which by making simulation scenarios and processes exchange particular groups to simulate the real network condition, and then records the performance in the log file (called the Trace file) for the user's analysis to get the simulation results. NS use this division model which not only improves the simulation efficiency, but also speeds up the simulation as well as providing the simulation configuration with flexibility and ease of operation. After years of development it has become a simulation tool involving all aspects of network. In addition, it has been widely used in teaching and learning with network technology.

Simulation of Routing Protocols

The experiment performs simulation of DSDV, DSR and AODV routing protocols in the two scenarios listed in Table 1 employing NS-2 simulator and provides a comparative analysis in terms of end-to-end delay. In Scenario 1, simulation is performed with different node numbers (20, 30, 40, 50 or 60) moving at speed up to 30m/s while in Scenario 2, it is done with the same node numbers (40) move at different maximum speed (up to 10m/s, 20m/s, 30m/s, 40m/s or 50m/s).

Table 1 Simulation Scenario Parameters Configuration

Parameters configuration	Scenario 1	Scenario 2
Data flow	CBR	CBR
Node numbers	20~60	40
Maximum speed(m/s)	30	10~50
Dwell time (s)	0	0
Duration of the simulation (s)	180	180
Movement zone (m)	500*500	500*500

Communication scenario (exemplified by 20 nodes in Scenario 1): ns cbrgen.tcl–type cbr –nn 20–seed 1–mc 10–rate 10.0 >cbrcj1-20

Movement scenario (exemplified by the maximum speed 20m/s in Scenario 2): ./setdest –n –p 0 –M30 –t 100 –x 500 –y 500 >scencecj1-20

The source codes of node number setting in TCL script and trace files are shown as below (exemplified by the maximum speed 50m/s in Scenario 2).

```
#Set trace file
set opt(tr) trace2-50.tr;
#Set number of nodes
set opt(nn)    40
#Set traffic pattern file
set opt(cp)     "cbrcj2-50"
```

#Set node movement scenario
set opt(sc) "scencecj2-50"

After the script is edited, use NS-2 to perform network transmission simulation and demonstrate it through NAM (Network Animator). The function of the software is run animation according to Trace output file of specific format in network simulation so as to observe the traces and grouping dada flow in network simulation of the packet flow and data flow. Fig. 2 shows the NAM simulation image of this experiment.

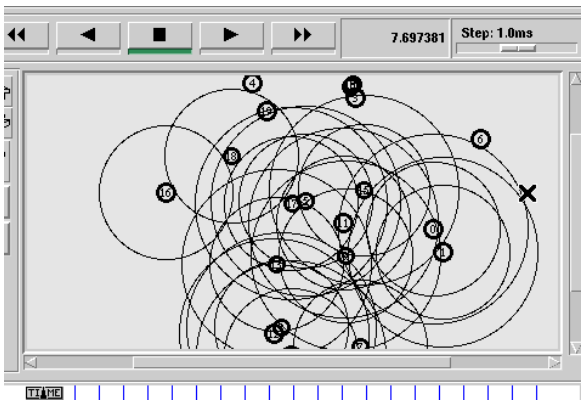

Figure 2 NAM Simulation Image of Node Number as 20 in Scenario 1

An Analysis of the Simulation Results

This experiment uses the AWK tool to analyze the results of the simulation log file. AWK is an excellent text processing tool, which scans each row in the Trace file, search the pattern that matches the given content entered in the command line. After finding matched content it performs programming analysis. The key AWK codes for endpoint to endpoint delay time are as follows:

```
if (action=="s" || action== "r" || action=="f" )
    { if(action=="s"&&trace=="AGT"&&type=="cbr")
     {sends++;}
     If (packet_id > highest_packet_id)
          {highest_packet_id = packet_id;}
          if(start_time[packet_id] == 0)
          {start_time[packet_id] = time;}
if (action =="r" && trace== "AGT" && type== "cbr")
     {if(first==0){
     first_received_time= time;
     first=1;}
     receives++;
     end_time[packet_id] = time;
    } else
```

end_time[packet_id] = -1;

for (packet_id = 0; packet_id <= highest_packet_id ; packet_id++) {packet_duration = end_time[packet_id] - start_time[packet_id];

if (packet_duration >0) end_to_end_delay += packet_duration;}

Avg_delay = end_to_end_delay / (receives);}

pdfraction = (receives/sends)*100;

printf(" Total packet sends: %d \n", sends);

printf(" Total packet receives: %d \n", receives); printf(" Packet delivery fraction: %s \n", pdfraction);

printf(" Average End-to-End delay:%f s \n" , avg_ delay);

printf("first packet received time:%f s\n", first_received_time);

Figure 3 AWK Data Analysis in terms of Node Maximum Speed at 50 m/s in Scenario 2

This experiment uses Gnuplot to perform interactive drawing in terms of AWK data analysis results.

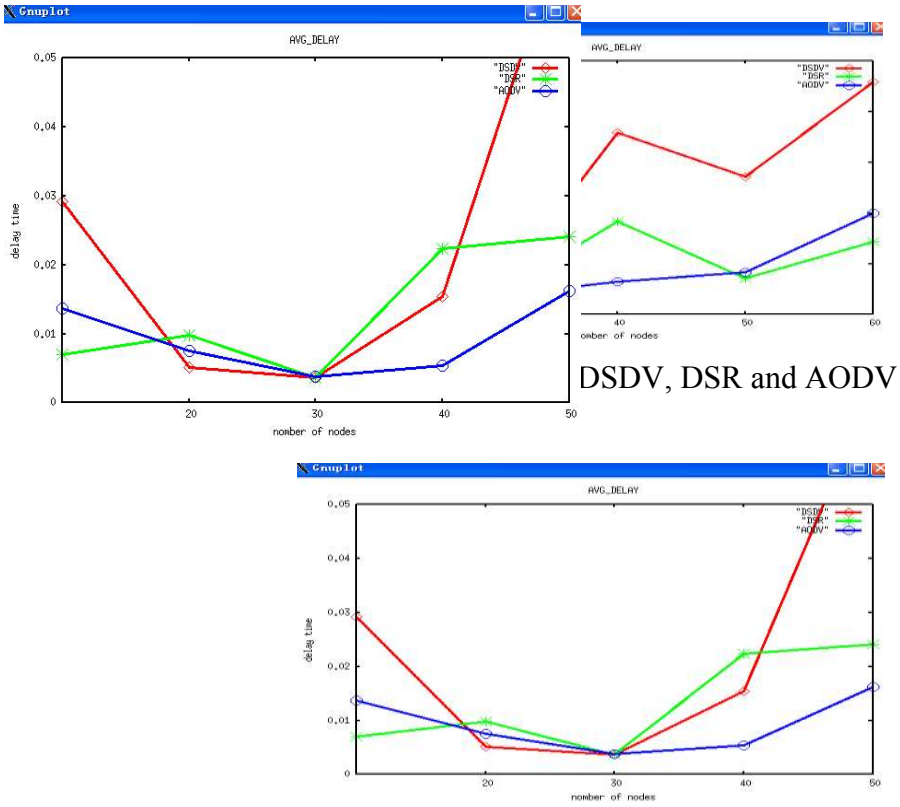

Figure 5 End to End Delay of DSDV, DSR and AODV in Scenario 2

From the results of the experiment, in Scenario 1, in the movement of nodes at the same top speed of 30 m/s, when the number of nodes exceeds 30, end-to-end delay of all the three protocols has increased. In particular, the delay increment of DSDV is the highest. In Scenario 2, as the movement speed of nodes goes beyond 30 m/s, end-to-end delay of all the three protocols has increased. In particular, the delay increment of DSDV is the highest. After extracting the value of first packet received time of the simulation experiment ten times in the two scenarios, DSDV, DSR, AODV average time is 7.7 s, 2.60 s, 2.59 s respectively. It is obvious that the average time of DSDV is much higher than other two protocols. This is because although DSDV is a table driven routing protocol, it is not necessary that it has available paths in its routing table. So in the case of a route failure, it has to find an effective path again for updating the routing table, thus making it a time-consuming process. Therefore, in a scenario of fast moving nodes and a large number of nodes, using AODV and DSR routing protocols can achieve high efficiency.

Summary

This paper has applied NS-2 simulation technology in the wireless self-organized network. Through simulating network configurations in different environments, it designs simulation cases, writes source codes and sets simulation parameters while using NAM animation to perform demonstration, AWK to analyze the simulation results and the Gnuplot program for drawing. Furthermore, it probes into the performance difference of the average end-to-end delay between the three protocols in the case of gradually increasing the number of mobile nodes and their movement speed. It is finally found that DSDV routing protocol as a typical active routing protocol has a poor performance in a large-scale network with a variable structure as in the experiment of this research.

Acknowledgement

This research was financially sponsored by the Youth Science Foundation under the Provincial Department of Education of Jiangxi (Project No.: GJJ14458).

Author in Brief

Qian CAI, born in November 1980, male, master, lecturer, whose research direction is self-organized network.

References

[1] Z. H. Ke, R. X. Cheng and D. J. Deng, NS2 Simulation Experiment—Multimedia and Wireless Network Communications, Publishing House of Electronics Industry, Beijing, 2009.

[2] H. J. Huang, S. L. Feng, L. J. Qin and H. Z. Chen, Network and Protocol Simulation, Beijing, Posts & Telecom Press, 2010.

[3] L. M. Xu, B. Pang and Y. Zhao, NS and Network Simulation, Beijing, Posts & Telecom Press, 2003.

[4] X. R. Xie, Computer Network, fifth ed., Publishing House of Electronics Industry, Beijing, 2009.

[5] L. P. Fang, S. P. Liu, et al, Basis and Application of NS-2 Network Simulation, National Defend Industry Press, Beijing, 2008.

[6] Y. Jiang, J. H. Gu, P. L. Du, et al, Reseach of wireless sensor network test-beds, Computer Technology and Development. 9, 20 (2010) 188-192.

[7] D. Wang, X. M. Wang, S. B. Wu, et al, Design of simulator for wireless mutimedia sensor networks, Computer Technology and Development. 9, 21 (2011) 1-5.