An alternative packet distribution procedure for mobile network simulation*

Shiyi WU and Christian BONNET

Mobile Communication Department
Institut Eurecom
2229, route des cretes
06904 Sophia-Antipolis, France
{Shiyi.WU, Chrisitian.Bonnet}@eurecom.fr

Abstract

Most researchers use ns2 to analyze the performance of routing protocols for ad hoc network. But the simulation executing time increases quickly as the mobile networks size, and augmenting. In order to reduce simulation running time for mobile networks, this paper proposes an alternative procedure, selective propagation, to send packet in wireless media layer of ns2. The idea is to calculate a list of receivers in the coverage range before sending a packet. The comparison shows this mechanism can gain up to 50 % of time for executing simulations.

Keywords – network simulator, ad hoc network, ns2, packet distribution

I. Introduction

Most researchers use ns2 to analyze the performance of routing protocols for ad hoc network. Ns2 is a discrete event driven simulator targeted at networking research. Ns2 provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks [1]. Several unicast routing protocols for ad hoc network have been implemented in ns2.

The time used in simulation takes a great part of research. When we used ns2 version 2.1b7 to do simulations for testing the performance of routing protocol for ad hoc networks, we found the simulation time increased quickly when the network size and messages augmenting. In order to reduce simulation time, this paper proposes an alternative packet distribution procedure, selective propagation for mobile network simulator.

In section II, we review the packet sending procedure in ns2 for wireless networks. Then we present selective propagation in section III. Section IV gives performance comparison of these two procedures. At last, section V is our conclusions and future work.

II. Packet propagation in ns2

Figure 1 shows low layers structure of a mobile node in ns2. For a mobile node, the NetIF is a WirelessPhy object.

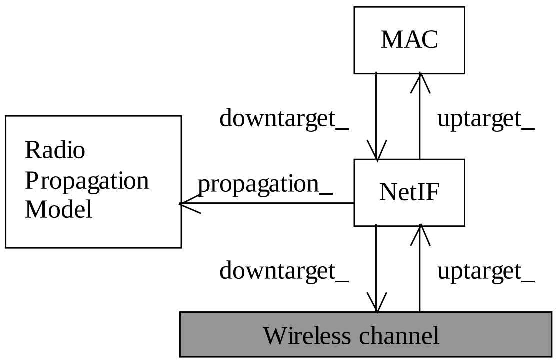

Figure 1 : Schematic of a mobile node (low layers) in ns2

The procedure of sending packet through wireless interface in ns2 is in accordance with real world. All mobile nodes have a WirelessPhy object, which is connected to a wireless channel object (Figure 1). Accordingly, the wireless channel object has a list of WirelessPhy objects which are attached to itself. WirelessPhy object inserts transmission power into packet (Pt) before forwarding this packet to wireless channel. Upon receiving a packet sent by a node

* This work is partially supported by HITACHI EUROPE

(said tnode), wireless channel object estimates the propagation delay to all other nodes in its list ($rnode_1$, …, $rnode_{n-1}$) (Figure 2). Before calculating the distance between two nodes, the position of both nodes are updated. The distance divided by 300,000km/s gives the propagation delay to the receiver. Then, wireless channel object sends this packet to them.

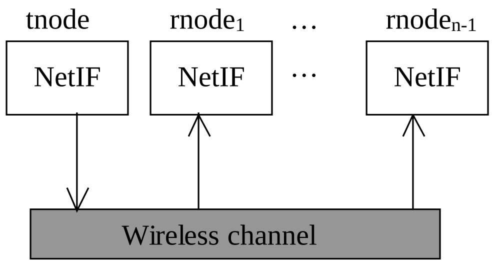

Figure 2 : Propagation packet

Upon receiving first bit of a packet from wireless channel, WirelessPhy object asks propagation model object to calculate receive power level of this packet (Pr) (see Figure 1) based on the distance, Pt, antenna gain of both nodes and so on. Propagation object uses Friss free space attenuation equation at near distances and an approximation to Two ray Ground at far distance. The approximation assumes specular reflection off a flat ground plane. The cross-over distance is calculated with

$$d^2 = \frac{(4\pi)^2 * L * h_t^2 * h_r^2}{\lambda^2}$$

where L is system loss, h_t, h_r is antenna height of transmitter /receiver, λ is the wave length of carrier. If the distance is greater than cross-over distance, receive power level (P_r) is calculated with Two ray Ground formula

$$P_r = \frac{P_t * G_t * G_r * h_t^2 * h_r^2}{d^4}$$

where P_t is the transmission power level, G_t and G_r are antenna gain of sender and receiver. Otherwise, if the distance is smaller than cross-over distance, Friss free space attenuation formula is used to compute P_r

$$P_r = \frac{P_t * G_t * G_r * \lambda^2}{(4\pi d)^2 * L}$$

Having receive power level (P_r) of a packet, WirelessPhy object should decide which action would be done to this packet. Figure 3 shows the procedure of this decision. WirelessPhy object has two thresholds: receive power threshold (RXThresh) and carrier sense threshold (CSThresh). If the P_r is greater than RXThresh, WirelessPhy object forwards the packet to it up layer (uptarget_). Otherwise, if the P_r is smaller than RXThresh but greater than CSThresh, WirelessPhy object finds that the channel is busy but cannot correctly receive the packet, it would report an error to the upper layer. If the P_r is smaller than CSThresh, WirelessPhy object receives nothing, it drops the packet silently. Same as delay calculation, during P_r computation, the positions of sender and receiver and the distance of two nodes are updated again to get accurate value.

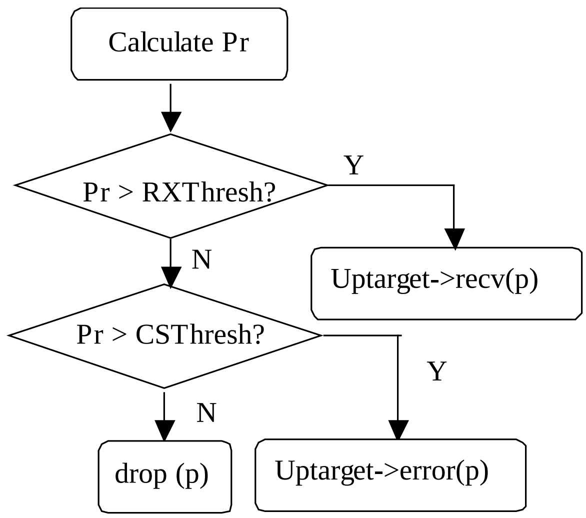

Figure 3: Procedure of WirelessPhy decision

The computation of sending a packet can be estimated like that. Suppose there are *n* mobile nodes in the simulator. In this mechanism, to send a packet, the position of nodes is updated *4*(n-1)* times (*2*(n-1)* times for tnode and *2* times for *n-1* rnodes). All nodes should be noted although some of them are far from tnode. *2*(n-1)* times of distance calculation and *(n-1)* times P_r computation. The fact is that movement speed of mobile node is not much smaller than light speed. During a propagation delay, a node won't move significantly. So, it's no need to update location of node so frequently and calculate distance two times during sending-receiving a message.

III. Selective propagation

Instead of notifying all nodes in the wireless network, we propose to calculate the list of nodes which can detect the packet before sending a packet.

The reason is that all configurations are given at the beginning of simulation and only transmission power Pt would be changed during simulation (power control and/or energy consumption). The wireless channel object can read the parameters of WirelessPhy object connecting to it and determine the list of receivers with a distance matrix of mobile nodes. The structure is also changed (see Figure 4)

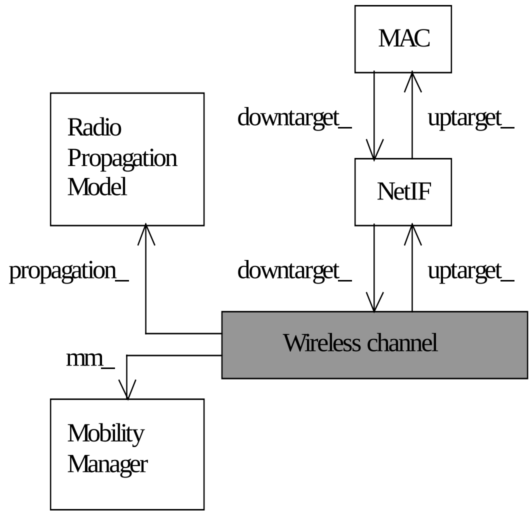

Figure 4 : Schematic of a mobile node (low layers) in selective propagation

A mobility manager object is introduced to manage this matrix. Using this matrix, wireless channel object decides which nodes will receive the packet sent by tnode according to the transmission power, distance and the CSThresh of receive nodes.

Mobility manager has to decide when and how to update the distance matrix. There are two important moments. One is when the movement of nodes causes significant propagation loss change (e.g. 1db) and the other is when a node arrives at its destination since it would change its behavior, which will as a result modify the prevision. We give each distance in matrix an attribute to say when to update this distance. To estimate the time of propagation loss change 1dB (t_{1dB}), the procedure is like that. At a moment t_0, the distance of two nodes is d, and receive power level is P_r. The movement speed of two nodes is v_a and v_b. After a moment, said t, the new distance, in the worst case, is $d' = d \pm (v_a + v_b) * t$. So the ratio of P_r' and P_r [1]is

$$\frac{p_r'}{p_r} = \left(\frac{d}{d'}\right)^{\alpha} \le \frac{1}{\left(1 \pm \frac{(v_a + v_b) * t}{d}\right)^{\alpha}}$$

where the value of α depends on whether d is greater than cross over distance calculated by (1). Given the difference of P_r' and P_r to 1dB, we have

$$1 \pm \frac{(v_a + v_b) t_{1dB}}{d} = 10^{\pm\frac{1}{\alpha * 10}}$$

when taking minus, t_{1dB} has a smaller value. So

$$\frac{(v_a + v_b) t_{1dB}}{d} = 1 - 10^{-\frac{1}{\alpha * 10}}$$ □

During mobility manager construction, it calculates the left part value for α = 2 and α = 4. The latest one among t_{1dB} and the moments of nodes arrival at their destinations is the time to update the distance between these two nodes. Another good result brought by equation (4) is that t_{1db} is direct proportion to actual distance. One node is further away from another node, the request for updating the distance between them is less frequent since the effect from one to another is more feeble.

When mobility manager receives a request for a distance between two nodes, it checks firstly whether the value should be updated. If it is the case, it updates the positions of both nodes and calculates their distance. At the same time, it determines the next update time using the above procedure. Then mobility manager replies the request.

At the side of wireless channel, this object calls the propagation object to compute the receive power level of a packet to all wireless nodes in the maximum coverage range of the sender. Wireless channel uses formula (2) or (3) to calculate the radius of maximum coverage range with maximum Pt and minimum CSThresh of the WirelessPhy objects connected to it. This radius could be evaluated when first packet arrives at this object. Wireless channel will only forward the packet to those nodes when the receive power level is greater than their CSThresh.

---

[1] We suppose the distance change do not pass cross-over distance

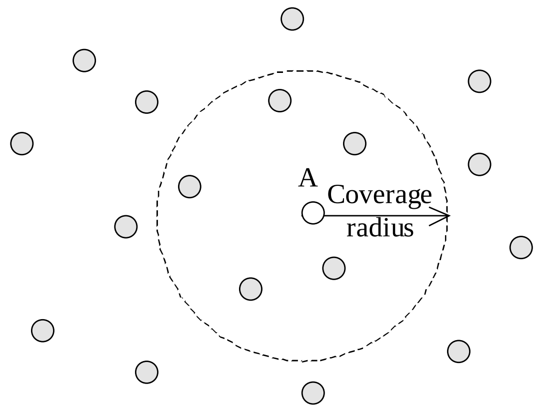

Figure 5: An ad hoc network example

For example in Figure 5, when node A sends a message, in fact only 4 nodes in the circle can hear this message with receive power higher than CSThresh. But ns2 will still distribute this message to all nodes (17 nodes). Consequently, there are 34 times distance calculation from node A (17 times for estimating propagation delay and 17 times for receive power level calculation). Using selective propagation in this example, the computation can be greatly reduced. Supposing all nodes are identical and those nodes send and receive packet with same criteria. The coverage range of node A, the circle, is the maximum coverage range. Then receive power level will be calculated only for these 4 nodes in the circle. Consequently these 4 nodes will be notified of the packet sent by node A. In the worst case, the number of distance calculation is reduced to 17 times and P_r evaluation is reduced to 4 times.

IV. Performance comparison

We modified the code of ns2 and integrated our proposition directly into simulator to compare the speed of two mechanisms. In the rest text, we call the code of ns2 without change as normal ns2 and the modified one as selective propagation.

The radio model uses characteristics similar to Lucent's WaveLAN with a 2Mb/s bit rate and a radio range of 250 meters [5].

Traffic sources are CBR unicast. Ten source-destination pairs are spread randomly over network. Each source generates 4 packets of 512 byte data per second. DSR is chosen as ad hoc unicast routing protocol.

The mobility model uses the random waypoint model in a 2000m $\times$ 2000m field. The pause time is 0 and node movement speed is uniformly distributed between 0 and a maximum speed. Simulations are run for 900 simulated seconds. Each source sends 3200 packets to their destination during a simulation.

We vary three key parameters: the number of nodes from 50 to 150; the number of sources 10 sources, 20 source and 30 sources; maximum movement speed from 1m/s to 30m/s. Each data point represents an average of 10 runs with different randomly generated mobility scenarios. For fairness, identical mobility and traffic scenarios are used across two procedures.

First we change networks size but use identical traffic models and same maximum speed. Figure 6 shows the comparison result. Selective propagation runs more quickly than normal ns2. The bigger the network size is , the more time it gains. We economize about 30% simulation time when there are 150 mobile nodes.

The second comparison aims to test scalability of movement speed. We fix the number of nodes to 50 and the number of sources to 10. The result is shown in Figure 7. Both mechanisms have the same tendency. Selective propagation is always 25% more rapid than normal ns2.

At last, we try to increase number of sources to see the performance of these two mechanisms. The number of nodes is 100 and their maximum movement speed is 10 m/s. We change the number of sources as 10 20 and 30. From Figure 8, we can see that selective propagation can save much more time than normal ns2 when traffic augmenting . The gain can get 55% at 30 sources. The simulation executing time of selective propagation increases more slowly than normal ns2 when traffic increasing.

The comparison results show that selective propagation can save up to 55% simulation time for a mobile network with 50 to 150 nodes and sending totally 32000 to 96000 data packets during the simulations. Selective propagation is more scalable than normal ns2 when number of nodes and traffic increasing. So this mechanism is more appreciative for big mobile network.

V. Conclusions and Future work

This paper proposed an alternative procedure for packet sending in simulator to economize simulation executing time of ns2 for mobile network. The broadcast capacity of wireless interface makes simulator consume lots of time for propagating packet. The mechanism used in ns2 is to send packet to all the other wireless nodes and calculate receive power level in each node, then according to the predefined thresholds, wireless interface decides different reaction. Our philosophy is the computation of receive power level should be done in wireless channel instead of wireless interface of each node. If two nodes are far away from each other, it's no use to calculate the precise distance between them and send packet. Based on this idea, this paper discusses a selective propagation mechanism. This mechanism aggregates distance calculation and notifies only a subset of nodes in certain region. The simulation results show this mechanism can save in some case 55% simulation executing time compared with ns2. Therefore, the selective propagation mechanism is suitable for big mobile network simulation.

This mechanism is proposed to improve the performance of ns2 for mobile network. We will study other simulators such as OPNET, GLOMOSIM to have a general idea.

Reference

[1] "The Network Simulator - ns-2" http://www.isi.edu/nsnam/ns/

[2] Josh Broch, David A. Maltz, David B. Johnson, Yi-Chun Hu and Jorjeta Jetcheva, "A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols", *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, ACM, Dallas, TX, Oct. 1998.

[3] Samir R. Das, Charles E. Perkins and Elizabeth M. Royer, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks", *IEEE Personal Communications* , Volume: 8 Issue: 1 , Feb. 2001, Page(s): 16 –28

[4] Kevin Fall and Kannan Varadhan, "ns Notes and Documentation", 2000, available from http://www-mash.cs.berkeley.edu/ns/

[5] Bruce Tuch. "Development of WaveLAN, an ISM band wireless LAN", *AT&T Technical Journal*, 72(4): 27-33, July/Aug 1993

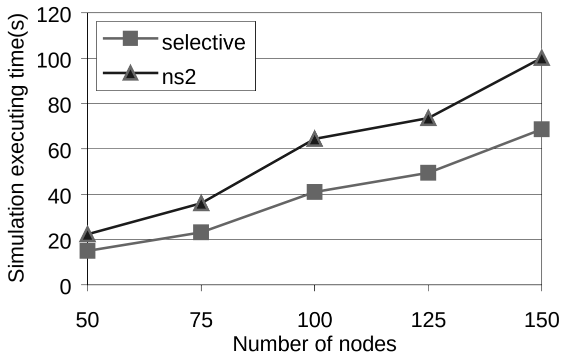

Figure 6 : Simulation executing time vs. number of nodes

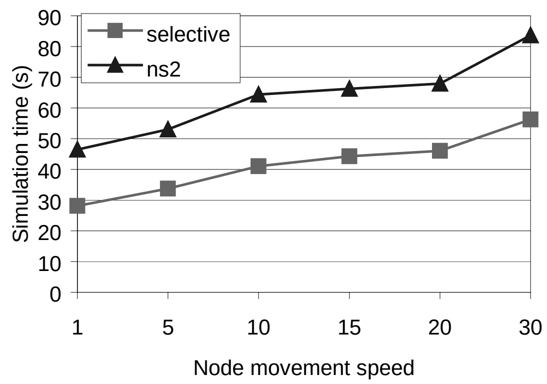

Figure 7 : Simulation executing time vs. maximum movement speed

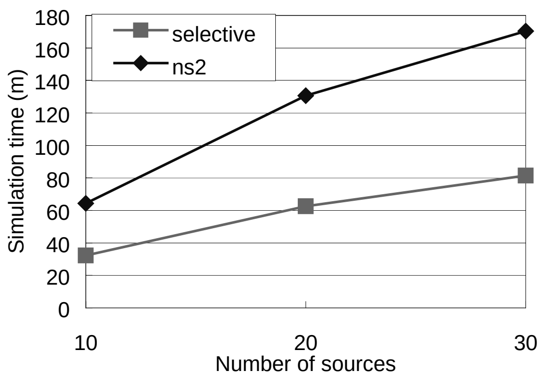

Figure 8 : Simulation executing time vs. number of sources