MAC Protocol and Routing Protocol for wireless movable sensor system

Luo Zhe
5050309927
F0603033
Team 5

***Abstract*— Nowadays, Sensors are used in more and more situations. In many situation, sensors are not fixed in a place. They are movable. It means that we must take of special methods in MAC protocol to save energy consumption, and in routing method. Movability indicates the topological structure of sensor network always changes. It becomes a big challenge how to find a short path from sensor to sink. In this report, we choose serval different MAC protocol and method of rounting, and use ns-2 to simulate their performance in a given situation.**

I. Introduction

A. Background

With the development of economy, travel becomes a popular thing for many people. And then, travel agencies take a important role in this domain. But it brings a problem. The people who are touring together want to act free. The guide can't control their behavior. The only thing he can do is to tell them come back in time. If the guide can make them take sensors and then, they form a network. The guide can know where they are and who move too far away.

This small groups of travelers are mobile in a small range of area of scenario of tour, each of their devices are battery-powered, we assume that the application scenario has such features as: mobility, which the nodes can move around freely in the specific area; small scale, the system scale is not necessarily very large, for the specific application; limited number of people and range, the anticipating people are around 10-50 people (the scale of a tourist group) and in the range of about 100 meters (airport, bus, sightseeing sites, etc.).

This network is alterable, so any sensor can change their neighbors. Old neighbor can be removed and new one can be added. The short path to sink will change any moment. So the ad-hoc network theory is very useful. The sensor must find new path if the key point of old short path disappears. Another problem is how to locate the sensor place. We only know the distance between near sensors and the origin sink place. Though these data, we can only calculate the approximate place of each sensor, but it is enough for our need.

This report tries to find a way to solve this problem based on some special routing protocol, MAC protocol and location method. In order to check the efficiency and performance for them, the simulation based on ns-2 will be needed. The mobile sensor network is not only useful in tourist industry, but also can be used in industry control, danger detector, resource estimation, person scout, etc.

II. Routing Protocol

In mobile environment, sensors are invisible for each other, and their neighbors change with time. A individual sensor can't know where it is and how to transmit information to sink except it can connect with sink directly, and the sink has the same difficult as sensors, too. They must take a special rule to overcome this challenge.

In this system, geography information can't became the role to solve this problem. Tourist travel all days, and there is no character with each scenario. So the routing method in mobile sensor network by using the location information [4] [5] is useless. In deed, the number of sensors in this system is limited, flooding method may be a good one. So in method A, a flooding method from sink is described and method A in MAC protocol, is the complement one for it.

But flooding method maybe consumes more than else, because the routing packet must be sent after some time, or the network will disappear. Another solution is to use distributing method: any sensor only needs to know through whom it can sent packet to sink. Every one can tell others they can if it has known through whom it can send packet to sink. This is method B and method B in MAC protocol, is the complement one for it.

A. Method A

The method of Routing is like this. First, an special packet is sent from sink. The packet contains a sign, which means this is a routing packet, a data time of sending from sink, a count of nodes through which this packet has pass and the node number of sending this packet.

Every sensor receives this packet and compares with a table. If it has received the earlier data time packet, it can ignore this packet. Else, it compares with the count of nodes- the less the shorter path is ,then change its table. The table can contain serval nodes. At the same time, it will send back a response packet to sink to show it can be reached if it is the first packet from sink at a new time. At the next circle it increases the count of node and sends this packet to others nodes.

When a sensor receives a packet to the sink from other sensor, it will find a random sensor from the table and send this packet to it in next circle.

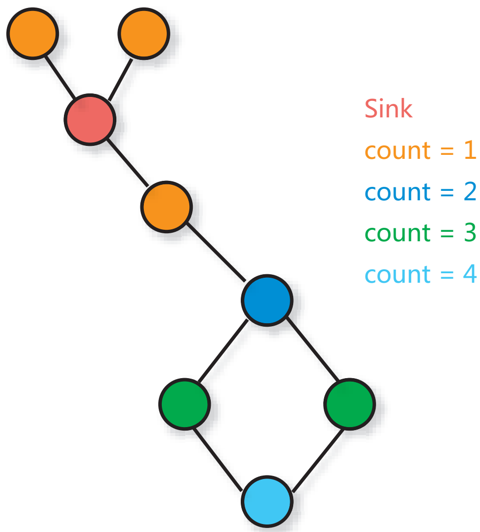

Fig. 1. Layers in Routing

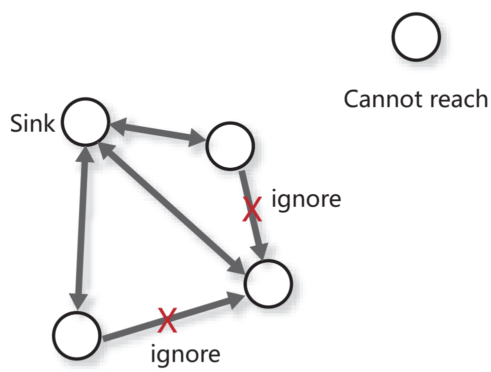

Fig. 2. Shorter path will be found

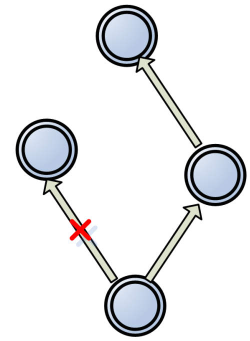

Fig. 3. Transmitting path. If there are more than one node could send to, random chooses one

TABLE I

	Routing Algorithm A
	initialization: For each node i
	Set routing table Null($n = 0$) (it contains tt, n, N_j(node number),$j = 0..n$, C(count)) every node has a fix number i, for the sink $i = 0$.
a)	**Sink node broadcast:** for sink
	The sink broadcast routing packet, which contains $flag$(indicate it is routing packet),Sid(source ID)=0, t(transmit time)=now time, c(count of nodes through which this packet has pass)=0
b)	**Other node response :** For each node i, who receives the routing packet,
	if t is earlier than tt in table of this node, do nothing. else, if $C > c$, then $n = 1$, $N_1 = Sid, C = c$ send a response packet to sink. set $c = c + 1$,$Sid = i$ broadcast the routing packet at next circle else, if $C = c$, then $n = n + 1$, $N_n = Sid$
c)	**transmit response packet:** For each node i who receives response packet from other node to sink
	1:if $n \neq 0$ check all $N_j, j = 0..n$ 2:if node N_j can't connect, delete N_j ($N_j = N_n, n = n - 1$) 3:random j in 1..n, send the packet to the node N_j
	Algorithm end

B. Method B

In this method, routing packet is unnecessary. At each circle, nodes check their neighbors whether a shorter path will be establish, and refresh their tables. According to the data flow and the length of path, nodes decide node to which data should be sent. This method is better than Method A because of saving packets from sink to every node.

III. MAC Protocol

Duty cycling is a widely used mechanism in wireless sensor networks (WSNs) to reduce energy consumption due to idle listening, but this mechanism also introduces additional latency in packet delivery. There are many method to realize this mechanism with low latency such as DW-MAC[2], RMAC[1],etc. In this system, there is no such high demand for latency, so we take CDMA as main communication method and avoid data confliction.

A. Method A

We use the synchronal MAC. At start of circle, there is Sync frame to synchronize clock. Then, nodes can transmit data each other by CDMA. Uplink and downlink use different frequences. Every node has a unique number for distinction. So we can arrange a orthogonal code for each node. It uses it own code to send signal.

And then, every node needs to sent BC(Broadcast), which contains a sign whether table is null its position and routing packet if existed. Every node who receives BC decode it and check. Now, every node knows their routing whether still existed. If the sender find the routing nodes are not existed or invalid, they will be deleted from routing table. The BC is also used as distance measure tool.

If there is packet needed to send, the node choose a random node in the table, and send CN(Connect) which contains the node number with which it real wants to communicate. If a node doesn't have packet to send or receive, it can sleep. It is time to send and receive data. There is no confliction with each other because every circle, each node only sent data to one node. After node received the data, it sends ACK. After node finishes above, it sleeps.

Because the location of nodes changes with time, nodes need to communicate in every circle, in order to calculate their new location. The idle listening doesn't exist in this way. The

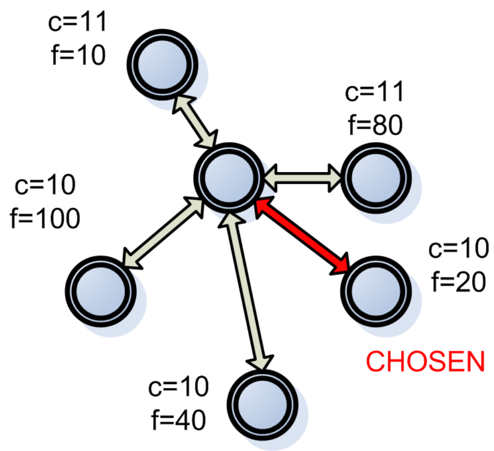

Fig. 4. Best path to sink now

TABLE II

	Routing Algorithm B
	initialization: For each node i
	Set routing table Null($n = 0$) (it contains n, N_j(node number), $j = 0..n$, $C_j(count)$,$F_j(flow)$) every node has a fix number i, for the sink $i = 0$.
a)	**Every node broadcast:** For each node i
	Every node broadcast itself, which contains Sid(source ID), $c = min(C_j) + 1$, f(data flow send at last circle) Sink has c=1, f=0
b)	**node refreshes table :** For each node i, who receives nodes broadcast,
	according c and f of every node i, function$F(c, f)$, estimate the better path(small c and f), record their c and f as C_j, F_j
c)	**transmit packet:** For each node i who has packet to sink
	Send packet to the best node in the table, (according to F(c,f))
	Algorithm end

nodes don't need to wake up frequently due to the velocity of people.

B. Method B

This MAC is similar with above one. But there is not routing packet in it. BC includes c, f in addition. Every circle table is needed to restructure according to $F(c, s)$. Choose best node to send considering count of nodes needed to pass to sink and data flow. After receiving data, node need calculate data flow been sent next circle f.

C. Method C

Above two MAC protocols use CDMA, but it brings a big demending for hardware, that power control is necessary. So the next MAC protocol use TDMA for avoiding collision. In this practical situation, node can't be far away, time delay is not important, and hops is very limited. So we can transmit data through one hop per cycle.

At first, there is synchronization frame. And then, every node broadcasts its layer distance from sink in its own time slot and it is also used for measure distance(BC). Following, every node sends SCH(schedule) in its own slot to tell the node the schedule which it will transmit data to. Last, at the plan time, the nodes wake and transmit data. The schedule time depends on transmitter number.

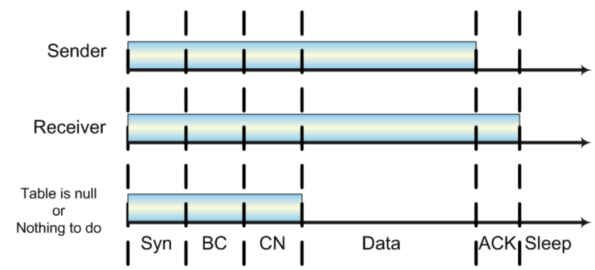

Fig. 5. MAC CDMA

TABLE III

	MAC Protocol A
a)	initialization: For each node i
	synchronization
b)	**BC:** for every node i
	code and send a sign of whether table $\neq NULL$, its position and routing packet if $\exists$ routing packet need to send
c)	**CN :** For each node i, who receives the BC,
1:	decode BC for every code
2:	if $\exists$ routing packet then refresh table
3:	check table, if $\exists$ N_j not in BC, or $\exists$ N_j in BC but the table of node N_j is empty delete N_j from table.
4:	according to BC measure distance from neighbor nodes and calculate approximate position
5:	if table= $NULL$ then sleep
6:	if $\exists$ packet to send then according to table choose one random node N_j send N_j as CN.
7:	go d).
d)	**check receiver :** For each node i
1:	decode CN for every code.
2:	if i in CN, record which node wants to send data to it.
3:	if $\nexists$packet needed to send and receive. sleep else go e)
e)	**data:** For each node i who has something to send or receive
i:receiver)	according to the list of nodes want to send data to it, decode data. after finish return ACK and sleep
ii:sender)	code data and send data, after finish check ACK and sleep
	Algorithm end

IV. Location

Locating is a not easy thing for this system, because sensors are mobile. radioLocation consist of two components. One is distance measure and the other is triangulation[6]. In this system, distance measure can't be very accurate, direction of radio is unbelievable and the neighbor sensors may not be enough. So we can only estimate their position.

The problem can be described as

$$\begin{pmatrix} (X_1 - U_x)^2 + (Y_1 - U_y)^2 + (Z_1 - U_z)^2 \\ (X_2 - U_x)^2 + (Y_2 - U_y)^2 + (Z_2 - U_z)^2 \\ \vdots \\ (X_n - U_x)^2 + (Y_n - U_y)^2 + (Z_n - U_z)^2 \end{pmatrix} = \begin{pmatrix} R_1^2 \\ R_2^2 \\ \vdots \\ R_n^2 \end{pmatrix}$$

But in practice, R_i is not accurate and the equations can't lead to solution. We use the (U_x, U_y, U_z) which minimize

$$\sum_{i=1}^{n} \frac{(\sqrt{(X_i - U_x)^2 + (Y_i - U_y)^2 + (Z_i - U_z)^2} - R_i)^2}{n}$$

Because people can't move quickly, the sensor position must be close to its origin position. So we can try position near last,

TABLE IV

	MAC Protocol B
a)	initialization: For each node i
	synchronization
b)	**BC:** for every node i
	code and send a sign of its path state and its position.
c)	**CN :** For each node i, who receives the BC,
1:	decode BC for every code
2:	calculate all $F(c,f)$,sort and refresh table
3:	according to BC measure distance from neighbor nodes and calculate approximate position. if necessary, send it to sink
4:	if table= $NULL$ then sleep
5:	if ∃ packet to send then according to table choose best node N_j send N_j as CN.
7:	go d).
d)	**check receiver :** For each node i
1:	decode CN for every code.
2:	if i in CN, record which node wants to send data to it.
3:	if ∄packet needed to send and receive. sleep else go e)
e)	**data:** For each node i who has something to send or receive
i:receiver)	according to the list of nodes want to send data to it, decode data. after finish return ACK, calculate data flow need to send next circle as f and sleep
ii:sender)	code data and send data, after finish check ACK and sleep
	Algorithm end

TABLE V

	MAC Protocol C
a)	initialization: For each node i
	synchronization
b)	**BC:** for every node i
	send a sign of its path state and its position in its time slot.
c)	**SCH :** For each node i, who receives the BC,
1:	distinguish node by time slot
2:	calculate all $F(c,f)$,sort and refresh table
4:	according to BC measure distance from neighbor nodes and calculate approximate position. if necessary, send it to sink
3:	if table= $NULL$ then sleep
5:	if ∃ packet(received from other node) to send then according to table choose best node N_j send N_j as CN.
7:	go d).
d)	**check receiver :** For each node i
1:	distinguish node by SCH time slot for every code.
2:	if i in SCH, record which node wants to send data to it.
3:	if ∄packet needed to send and receive. sleep else go e)
e)	**data:** For each node i who has something to send or receive
i:receiver)	according to the list of nodes want to send data to it, wake when their time slots, or sleep. after finish return ACK, calculate data flow need to send next circle as f and sleep
ii:sender)	at its time slot, wake, and send data, after finish check ACK and sleep
	Algorithm end

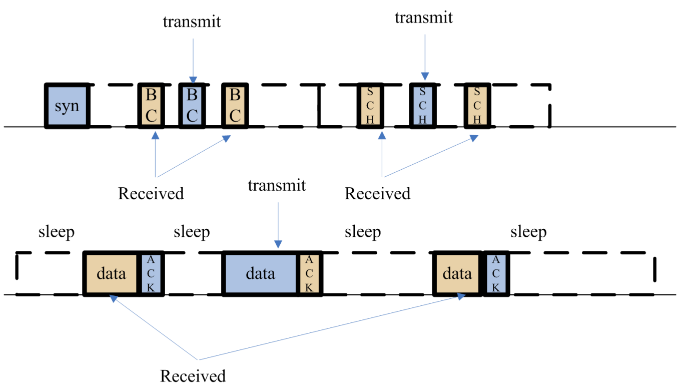

Fig. 6. MAC TDMA

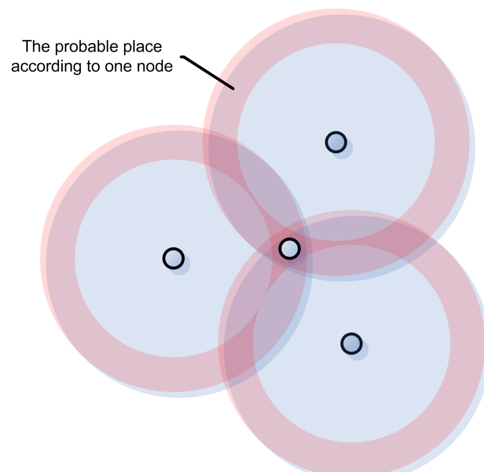

Fig. 7. Basic theory: three neighbors decide one position

and find most believable position. If there is no such position, glance all position to find position near most believable one, and then try to find most believable one.

V. Simulation: Next Step

In ns-2, it offers some basic classes of physic layer, MAC protocol, routing protocol and application layer. In this project, MAC protocol, routing protocol and application layer must be rewritten in C++. It is difficult work for us and we now construct a simple and basic scenario. In this scenario, physic layer uses two-ray model, MAC protocol uses 802.11, routing protocol uses DSDV, application layer uses tcp and ftp.We use 15 nodes for simulation, and node 0 is the sink. node 0 remain static and other nodes move away from node 0. Every node sends information to node 0.

From the following figures, we can find that using this method, connection is not stable. The larger the distance between sink and node is, the harder node transmits data to sink. And there is a big delay to construct a connection with sink in this method.

Fig.8 to 10 show the position of nodes at different time. At first, all node get together. With time passing, nodes begin to move and become separate. At starting, all node can connect to sink. But because the bandwidth is limited, some node can't establish ftp connection at first such as node 1 and 14. At the last stage of simulation, some some nodes get connection successfully, and some nodes lost their connection such as node 3 and 5. Node 4 is connected at first and still connected at last though it is far away from sink. The Fig. 11 shows the congestion windows of TCP, every successful transmission increases it by 1 and it becomes half if packet is lost.

The following Fig. 12 to 14 show the congestion window in different MAC and routing protocols. We can see that TDMA reduces the collision. Once connection is established, it is

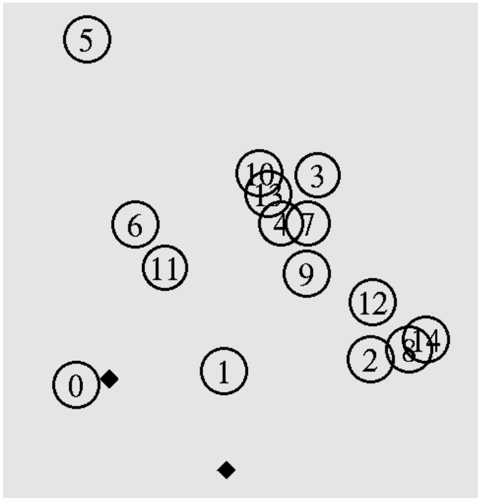

Fig. 8. position in 125s

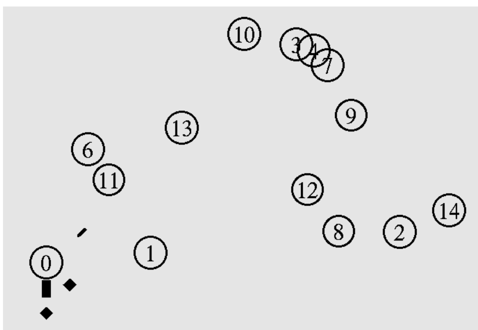

Fig. 9. position in 200s

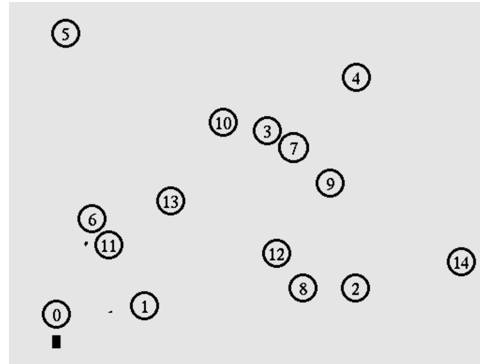

Fig. 10. position in 270s

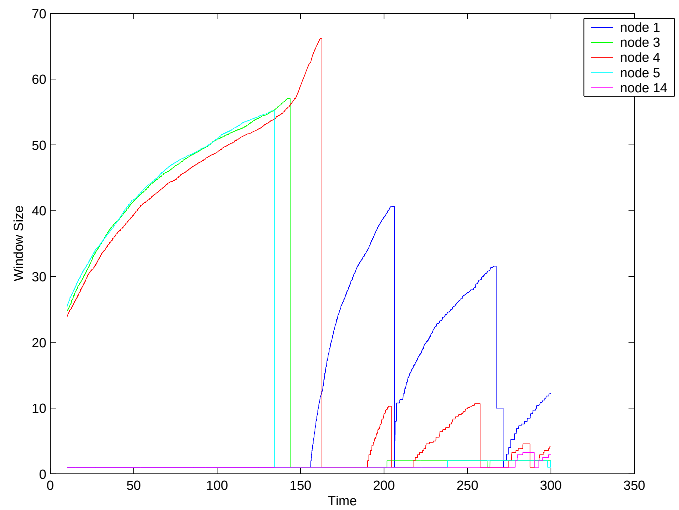

Fig. 11. congestion window in 802.11 and DSDV

harder to lost than that in 802.11. But it leads that far node has less chance to transmit to sink due to bandwidth. DSR needs more control packet to establish connection, so when collisions often happen, the farer the node is, the less chance it has to transmit.

In mobile model, Random Direction Mobility Model and Random Waypoint Mobility Model[3] are useful. In this problem, node always move away from node 0. There are some interesting places for nodes, and nodes usually move towards these places. After they reach the place, they will stay for a while and then find next place to move. So we decide to customize the moving path of each node.

Next step is to simulation the whole system by rewrite .cc file and test the performance of different method in different mobile situation [3](using different path script), and we will find whether this model can work correctly and efficiently.

March 29, 2009

References

[1] Shu Du, Amit Kumar Saha and David B. Johnson,"*RMAC: A Routing-Enhanced Duty-Cycle MAC Protocol for Wireless Sensor Networks,*" INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE 6-12 May 2007 Page(s):1478 - 1486 Digital Object Identifier 10.1109/INFCOM.2007.174 .

[2] Yanjun Sun, Shu Du, Omer Gurewitz David and B. Johnson.,"*DW-MAC: A Low Latency, Energy Efficient Demand-Wakeup MAC Protocol for Wireless Sensor Networks,*" International Symposium on Mobile Ad Hoc Networking & Computing Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing Hong Kong, Hong Kong, China SESSION: Link layer design and rescheduling Pages 53-62 Year of Publication: 2008 ISBN:978-1-60558-073-9

[3] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra,"*Performance Analysis of Mobility-assisted Routing,*"International Symposium on Mobile Ad Hoc Networking & Computing Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing Florence, Italy SESSION: Mobility models Pages: 49 - 60 Year of Publication: 2006 ISBN:1-59593-368-9

[4] Qing Fang Jie Gao and Leonidas J. Guibas,"*Locating and Bypassing Holes in Sensor Networks,*" Mobile Networks and Applications Volume 11 , Issue 2 (April 2006) Pages: 187 - 200 Year of Publication: 2006 ISSN:1383-469X IEEE INFOCOM 2004 0-7803-8356-7/04 (C) 2004 IEEE

[5] Ljubica Blazevic, Jean-Yves Le Boudec, and Silvia Giordano, "*IEEE A Location-Based Routing Method for Mobile Ad Hoc Networks,*"Mobile Computing, IEEE Transactions on Volume 4, Issue 2, March-April 2005 Page(s):97 - 110 Digital Object Identifier 10.1109/TMC.2005.16

[6] Chris Savarese, Jan M. Rabaey and Jan Beutel "*LOCATIONING IN DISTRIBUTED AD-HOCWIRELESS SENSOR NETWORKS,*" Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on Publication Date: 2001 Volume: 4, On page(s): 2037-2040 vol.4 Meeting Date: 05/07/2001 - 05/11/2001 Location: Salt Lake City, UT, USA ISBN: 0-7803-7041-4 References

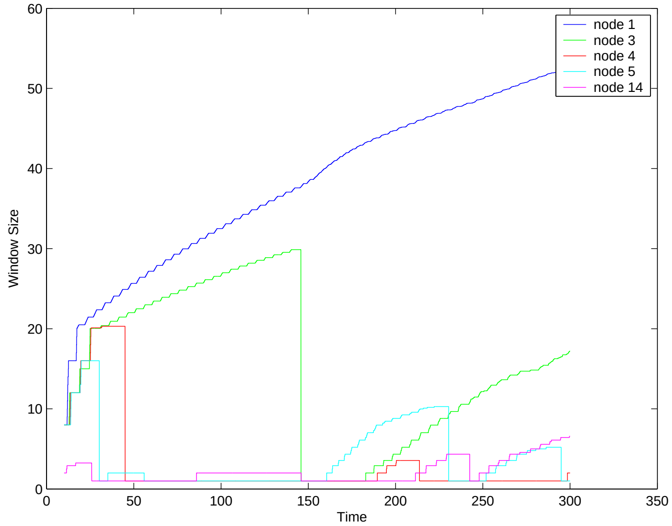

Fig. 12. congestion window in TDMA and DSDV

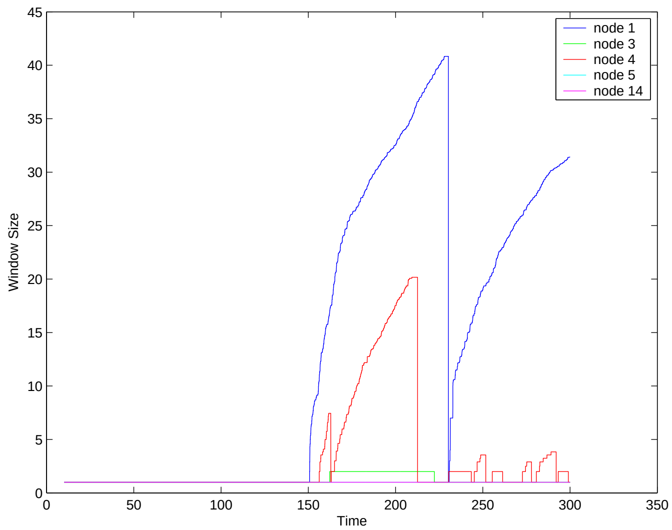

Fig. 13. congestion window in 802.11 and DSR

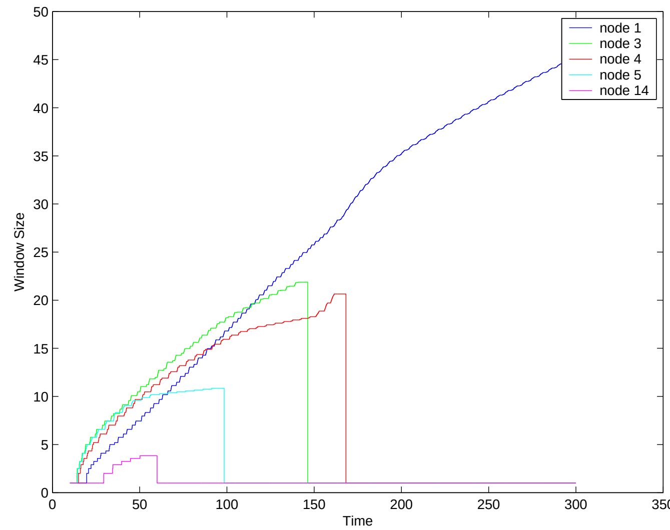

Fig. 14. congestion window in TDMA and DSR

Cited: 4 INSPEC Accession Number: 7167752 Digital Object Identifier: 10.1109/ICASSP.2001.940391 Current Version Published: 2002-08-07