Variable Update Period for DSDV Protocol

Zhijia Fang，Qinghua Shi，Lin Yao
College of Computer Science and Technology
Shandong University
Jinan 250101，Shandong Province，P.R. China
fangzhijia1234@126.com

***Abstract*: DSDV is a Distance-Vector routing protocol, which demands that every node periodically broadcast updated information. After analyzing the character and recent research on DSDV, the paper proposes a variable update cycle algorithm for DSDV to solve the problem that the cost and bandwidth waste largely increase because of broadcast updating when a increasing number of nodes enter the Ad Hoc. It sets different update cycle according to different node and the whole circumstance of Ad Hoc network. In order to cut down unnecessary resource waste, the algorithm mainly takes into account the remnant battery power in the node and the rate of topology change. The simulation results indicate that improved DSDV can evidently reduce bandwidth waste and prolong life-span of the nodes.**

Key Word：Table-driven；Broadcast update；Hub node；Node elimination

I. INTRODUCTION

Ad Hoc is a wireless communication system, designed for military communications[1]. It has some characters, for example, no center, self-organization, multi-hop routing, independent network-forming, mobile nodes, which have lots of advantages when used in non-military circumstance. Because of multi-hop, the change of network's topology makes traditional Internet-based routing protocol hardly adaptive to the need of high rate in topology change. In addition, routing protocol should consider low bandwidth, high error rate and low energy consumption of Ad Hoc[2].

DSDV(Destination-Sequenced Distance-Vector Routing) is a table-driven routing protocol for Ad Hoc[3]. It was originally adapt to the Ad Hoc which has few nodes. Now, it becomes a research focus and is under the development. The paper improves the current Ad Hoc protocol and proposes a more efficient algorithm. Lastly, we get the experimental result reflecting the real circumstance from simulation.

II. BACKGROUND

A. DSDV Algorithm

In table-driven routing protocol, every node keeps at least one table to restore routing information from this node to others. Every node sends and receives update information, updates local tables to maintenance consistent accurate current routing information in whole network. When resource sends message, it can find appropriate routing. Different table-driven routing protocols differ in the number of routing tables they need and the way they send update information.

DSDV is a Distance-Vector and table-driven routing protocol improved from traditional Bellman-Ford routing protocol on the basis of RIP[4]. It solves a problem, such as routing circle and endless count DBF algorithm results in. It adopts shortest path rule: every node locally preserves a routing table which includes all the other nodes this node can get to, the next node on the way to destination and hops, the serial number distributed by destination and so on. Usually, every node sends ascending destination serial number. Then it can differentiate new and old routing in order to avoid circle. Every node periodically sends local routing table to all its neighbors and it also sends information when its routing tables change.

In DSDV routing protocol, routing table is periodically time-driven, included in the topology update information. The update manner is used for fast changing network, by informing new nodes about network topology when they join. DSDV can reduce end-to-end delay and to some extent satisfies kinds of applications in QoS request[5]. And DSDV triggers update in a all-updating way, saying every node's routing table can turn to a steady state in a short time. So we consider making DSDV a appropriate routing protocol in WMN and its recovered nodes area. Nodes in WMN don't have a high moving rate as in MANET, so it will be a longer time in periodic update. If some information about update is included in DSDV routing table, the protocol will improve the sending rate of data packets.

B. Evaluation

The update manner in DSDV protocol has some disadvantages as follows;

a. Weak scalability. No matter how much change the network has, every node must periodically broadcast update information. Routing table's capability and bandwidth is increasing as much more nodes join. It is the main disadvantage in DSDV. As long as the topology changes, DSDV is unstable until update information broadcasts all over the network. If a node has a large moving rate, DSDV will have difficulty in operating.

b. Low packet transmission rate and bad throughput

978-1-4577-0321-8/11/$26.00 ©2011 IEEE

performance. Packet transmission rate is the ratio between the number of packets destination receives and that source sends[6]. Although the cost of table-driven routing protocol is nearly the same when nodes move faster and faster, it can't convergence in time when topology changes fast. Maybe this situation result in lots of unreliable routing or causes routing circle or lead to lost of packets. The packet transmission rate sharply drops when topology changes rapidly.

c. Large cost on broadcast. Because every node periodically broadcasts update information, bandwidth is occupied and part of battery power is used for broadcast. All these lead to larger routing cost, longer delay, lower switching ratio. It may bring out node's early disappearance to those Ad Hoc networks that don't have too much battery power[7].

It concludes that the periodic update manner of DSDV is important to protocol itself. On the basis of the analysis of DSDV's update manner, the paper proposes a variable update cycle method for broadcast update, in order to solve the problem such as large cost and node's early disappearance caused by periodic update.

III. VARIABLE UPDATE PERIOD ALGORITHM

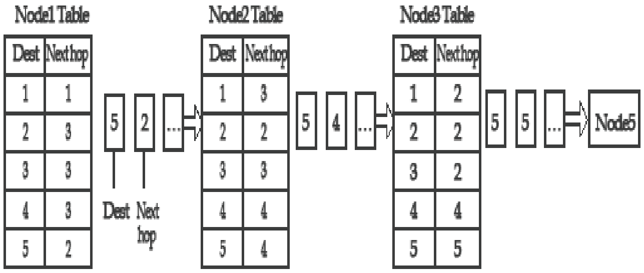
A. Improved Algorithm Model

In DSDV, packets choose routing by routing tables when forwarded. Every node's routing table includes all other node's address and next hop's address[8]. In Fig.1: If node1 communicates with node5, and node1 knows after looking up table that node5 is the next hop, it will send packets to node2. The table shows the next hop of destination node5 is node4 when received by node2, and packets will be sent to node4 from node2. The node4 will do the same thing when the table arrives. At last, the packets get to destination node5.

In the process, original DSDV adopts periodic manner in broadcast update. To get a variable period, every routing table sets two columns noting the last and current broadcast update cycle from this node to others.

For example the format of node1's routing table is TABLE I.

In the table, *other* is all other nodes, *desti* is destination node, *next hop* is next hop, *t_last* is the node's last broadcast cycle, *t_cur* is the current broadcast cycle.



Node1 Table

Dest	Nexthop
1	1
2	3
3	3
4	3
5	2

Node2 Table

Dest	Nexthop
1	3
2	2
3	3
4	4
5	4

Node3 Table

Dest	Nexthop
1	2
2	2
3	2
4	4
5	5

Fig.1 Communication with five nodes

TABLE I

ROUTING TABLE OF FIVE NODES

other	desti	next_hop	t_last	t_cur
2				
3				
…				
n-1				
n				

In the algorithm, there are two factors that influence computation of the next broadcast cycle:

a. The residual battery power. Algorithm ensures that some nodes don't die too early to lengthen the survival time and improve robustness of the whole network. It's focus is battery optimization, so the influence of battery is primarily considered. The value of next cycle is inverse proportional to residual battery power.

b. Recent broadcast cycle. The computation of next cycle should consider last two cycles to avoid evident offset, and unitary network doesn't have a grave jitter. This is also the reason why DSDV adapts to the Ad Hoc whose topology changes a little. The value of next cycle is proportional to last two cycles with a given definite value.

Core idea: Set a threshold for residual battery power in every node. Update with initial value when battery power is higher than threshold. Otherwise, use this iterative algorithm to compute next update cycle.

Suppose ξ is percentage of residual battery power, T_1 is the last broadcast cycle, T_2 is the current broadcast cycle, α is weighting factor(decides which influences more in computing), η is a starting threshold(usually 0.7). Then next broadcast cycle is:

$$T=(\alpha \bullet T_1+(1-\alpha) \bullet T_2)/（\xi+1-\eta） \qquad (1)$$

In formula, we use $\xi+1-\eta$ in stead of ξ in order to avoid a phenomenon that cycle may be very large when there is little battery power. This is the process: last broadcast cycle is discarded when current broadcast update is over. Then the current cycle is evaluated with the last, and the computed cycle is considered as "current".

B. Iterative Process

step1:

Initialize routing table in every node. Evaluate *t_last* and *t_cur* and other columns are evaluated as original DSDV.

step2:

Broadcast update.

step3:

Check if residual battery power reaches threshold after broadcast update. If it is, compute next update cycle using formula 3-1 and evaluate new cycle. Otherwise, update with initial value.

step4:

Check if Ad Hoc terminates. If broadcast goes on, go to step2. Otherwise, stop the algorithm.

In iteration, the operation, initialization and computation of other data are all according to original DSDV[9].

C. Qualitative Evaluation

Algorithm adds two periodic items in the periodic table on the basis of original DSDV protocol and estimates residual battery power. According to the theory analysis above, improved algorithm has the following advantages and disadvantages.

Advantages：

a. It optimizes Ad Hoc network and reduces bandwidth occupied by the broadcast update. In the original DSDV, every node must periodically broadcast update information. Even if the node isn't included in the key path, it must do as those in the path. This update manner causes part bandwidth waste. Especially when the network congests, the appearance of broadcast information will evidently cause bad load.

b. When topology changes a little, for example, nodes join or leave, or move in the network, the whole performance of algorithm isn't influenced very much. The primary reason is that algorithm adopts periodical iteration and the next cycle is referred to the cycle value before. The computed value and the recent value is more or less the same. The value shows a smooth changing trend.

c. Algorithm prolongs the life of battery. It considers another important problem is that Ad Hoc can load a mass battery and computes different broadcast cycle as to different residual power node. Algorithm tries to prolong every node's survival time to avoid premature disappearance and promote the existence of the whole Ad Hoc[10][11]. The improvement of robustness is important to the transmission ability in the interest of load capacity, for it makes nodes leave from the Ad Hoc later, without "barrel effect".

Disadvantages：

a. There are two periodic items in very routing table of the node. With the increment of the number of nodes, the total computation of algorithm is exponentially increasing and the cost of node's memory is linearly increasing. For the nodes that are weak in computing and memory, the algorithm maybe make network badly loaded[12].

b. When topology changes a lot, the existing problem of slow convergence is not evidently solved. The reason is that algorithm mainly considers cycle by computing different update cycle for different nodes, not considering key path.

IV. EXPERIMENT RESULTS AND DISCUSSION

On the basis of the algorithmic model and analysis, the paper compares original DSDV with advanced DSDV by NS-2. The experiment sets 50 nodes in a 1000m*1000m square. Every node refers to MAC level protocol in IEEE802.11 and Network interface refers to WaveLan wireless interface standard. There is CBR data flow about 512B/s packets in experiment. Nodes never stop on the move and experiment lasts for 300s. In original DSDV, update period is 15s and in advanced is initialized 15s too. Packet delivery rate and cost of routing are studied in experiment.

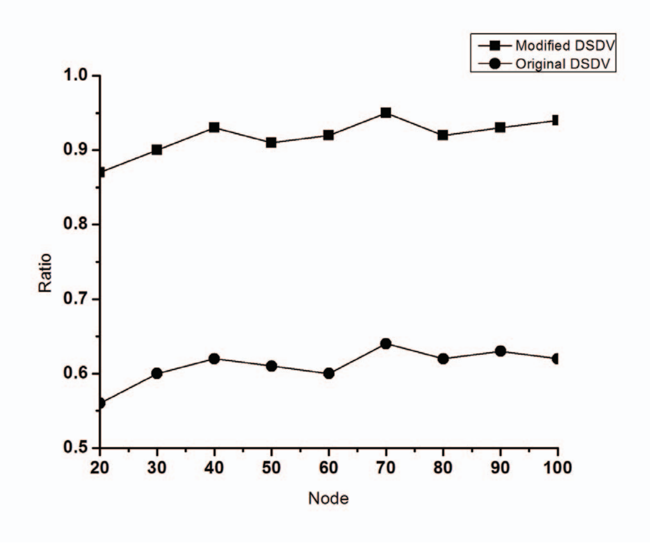


Fig.2 Ratio of different nodes

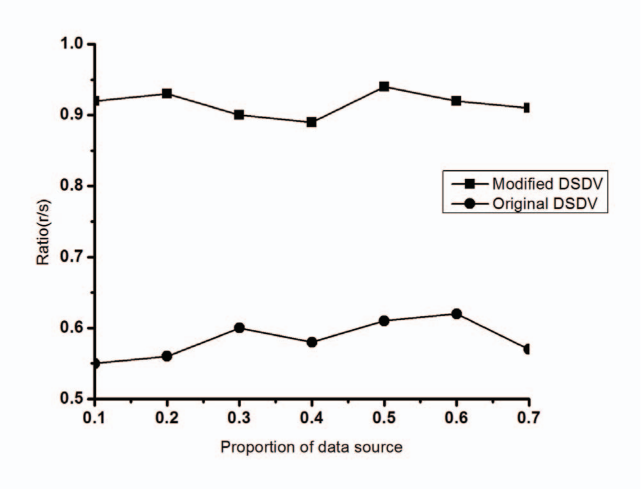


Fig3.Ratio of different proportion of data source

a. Packet delivery rate. Packet delivery rate is the value of received packets divided by sent packets from source. Fig2 and Fig3 shows results of packets delivery rate in different number of nodes and different network load in original and advanced protocol. It's clear that packet delivery rate is higher in advanced DSDV than original and data trends smoothly without large undulation. The reason is that algorithm sets different update period for different node to lower bandwidth usage by reducing unnecessary waste.

b. Routing cost. Fig4 and Fig5 shows results of routing cost in different number of nodes and different network load in original and advanced protocol. In the contrast, advanced DSDV improves network cost in the interest of variable update cycle. It's unnecessary for AdHoc to update frequently.

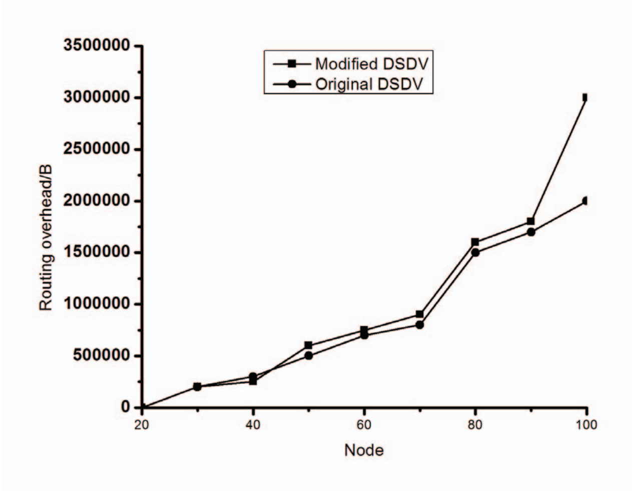


Fig4. Overhead of different nodes

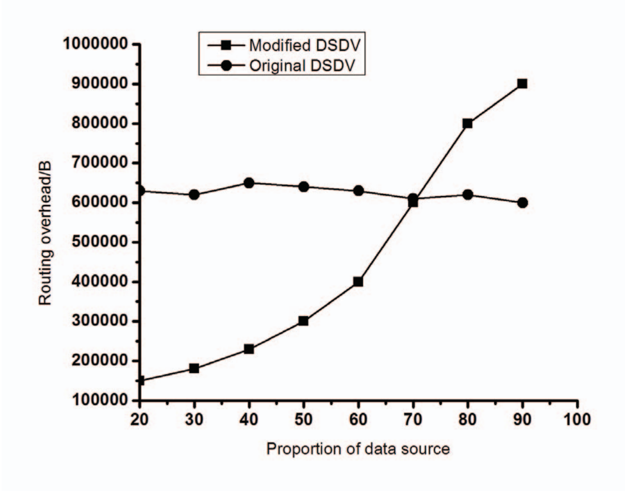


Fig5. Overhead with different proportion of data source

In stead, it brings asynchronous communication by computing adaptive cycle for every node. In chart, it's not effective in advanced algorithm with few nodes because it need to compute variable update cycle. But it is clear when the number of nodes is larger than a value.

V. CONCLUSION

The paper proposes a variable update cycle algorithm on the basis of DSDV protocol in Ad Hoc network, and shows advantages of improved protocol in theory and simulation. Variable update cycle algorithm is established by advanced routing table structure on the basis of original rule[13][14]. The simulation shows that it improves throughput and survival ability by sacrificing occupancy rate of CPU and memory.

With the development of Ad Hoc network, we are increasingly demanding routing protocol. It will adequately optimize network by combining original routing protocol with variable cycle mechanism in order to efficiently cut down packet loss rate, delay, load balancing and solve the problem of low loaded battery. I will perfect the algorithm on the basis of DSDV.

REFERENCES

[1] Chenlinxing, Zengxing, Caoyi. Mobile Ad Hoc [M]. Beijing Electronics Industry, 2006 7-9, 48-51.

[2] Zhengxiangquan. Ad Hoc Technology application [M]. Beijing Tinghua, 2004 161-179.

[3] ROSATIL, BER IOL IM, REALIG. On ant routing algorithms in Ad Hoc networks with critical connectivity [J]. Ad Hoc Network, 2008, 6(6): 827-859.

[4] CARO GD, DUCATELLE F, GAM BARDELLA L M. Ad Hoc Net An adaptive nature inspired algorithm for routing in mobile Ad Hoc networks [J]. European Transact ions on Telecommunications, 2005, 16(2): 443-455.

[5] WANG H, SH I Z, GEA F,. An optimized ant colony algorithm based on the gradual changing orientation factor for multi constraint QoS routing [J]. Computer Communications, 2009(32): 586-593.

[6] KADONOD, IZUM IT, OOSHITA Feta l. An ant colony optimization routing based on robustness for Ad Hoc network s with GPS [J]. Ad Hoc Network s, 2009, 8(1): 63-76.

[7] WANG JP, OSAGIE E, THULAS IRAMAN P,. HOPNET: A hybrid ant colony optimization routing algorithm for mobile Ad Hoc network [J]. Ad Hoc Network s, 2009, 8 (4): 690-705.

[8] HAIL J, LIU X, X IA Y D. Research on multicast routing protocols for mobile Ad Hoc networks [J]. Computer Networks, 2008, 52(5): 988-997.

[9] ROYERE, PERKINS C. Multicast operation of the Ad Hoc on demand distance vector routing protocol [C] / Proceedings of Mobi−Com. Seattle: [s. n.], 1999.

[10] CH IANG CC, GERLA M, LEE SJ. On demand multicast routing protocol in multi hop wireless mobile network s [J]. Mobile Networks and Applications, 2002, 7(6): 441-453.

[11] MENCHACA ▪ MENDEZL R, VAISHAM PAYAN R, GARC IA ▪ LU ▪ NA ▪ ACEVES JJ,. DPUMA: A highly efficient multicast routing protocol for mobile Ad Hoc networks [M]. Berlin: Springer, 2005: 178- 191.

[12] YUN SY, YI KC, HAN C A genetic algorithm for energy efficient based multicast routing on MANET [J]. Computer Communications, 2008, 31(4): 858 - 869.

[13] CHIANG TC, LIU CH, HUANG Y M. A near optimal multicast scheme for mobile Ad Hoc networks using a hybrid genetic algorithm [J]. Expert Systems with Applications, 2007, 33(3): 734 -742.

[14] HUANG CJ. Using particle swam optimization for QoS in Ad Hoc multicast [J]. Engineering Application s of Artificial Intelligence, 2009, 22 (8): 1188- 1193.