

# Destination-Sequenced Distance Vector (DSDV) Protocol

Guoyou He  
Networking Laboratory  
Helsinki University of Technology  
ghe@cc.hut.fi

## Abstract

*An ad hoc network is a collection of mobile nodes forming an instant network without fixed topology. In such a network, each node acts as both router and host simultaneously, and can move out or join in the network freely. The instantly created network does not have any base infrastructures as used in the conventional networks, but it is compatible with the conventional networks. DSDV is a modification of the conventional Bellman-Ford routing algorithm. It addresses the drawbacks related to the poor looping properties of RIP in the face of broken links. The modification adapted in DSDV makes it a more suitable routing protocol for ad hoc networks. This paper reviews the DSDV protocol, and analyzes the properties of DSDV when it is used for ad hoc networks routing.*

## 1 Introduction

Ad hoc networks differ significantly from conventional networks in the dynamic topology of interconnections and automatic administration for setting up the network. From a graph theory point of view, an ad hoc network is a graph  $G(N, E(t))$ , which composes of a set of nodes  $N$ , and a set of edges  $E(t)$ . Each mobile host can be a node of the graph. Each edge of the set  $E(t)$  is formed by two nodes within the service range, it can be unidirectional or bi-directional.  $E(t)$  changes with time as the mobile nodes in the ad hoc network freely move around. The topology of the ad hoc can be arbitrary at any time. With the change of the topology of an ad hoc network, the nodes in the network have to update their routing information automatically and instantly. Routing protocols in packet-switched networks traditionally use either distance-vector or link-state routing algorithm. Both of them allow a host to find the next hop to reach the destination via the “shortest path”. The metric of the shortest path might be number of hops, time delay in milliseconds, total number of packets queued along the path, or something similar. Such shortest path protocols have been successfully used in many dynamic packet switched networks. In principle, any such protocol can also be used in ad hoc networks. The main drawbacks of both link-state and distance-vector protocol are that they take too long to converge and have a high message complexity. Because of the limited bandwidth of wireless links in an ad hoc network, message complexity

must be kept low. In addition, the rapidly changing topology requires that the routing protocols can find routes quickly. So new routing protocols have to be developed to fulfill this basic philosophy. The destination sequenced distance vector (DSDV) protocol [7] is an adaptation of the classical Bellman-Ford [11] routing protocols. It is specifically targeted for the ad hoc networks. It addresses the long-lived loops and counting to infinity problems of the conventional distance-vector routing protocols.

## 2 Overview of Routing Protocols

The most popular packet routing protocols in conventional networks are primarily categorized as link-state and distance-vector.

**Link State Routing Protocol** [7, 11] keeps a routing table for complete topology, which is built up finding the shortest path of link cost. The information of link cost is transmitted periodically by all nodes using flooding technique. Each node updates its routing table using new link cost information gathered. Link cost information may be inconsistent because of dynamic behavior of topology or wireless medium, such as instantaneously incorrect long propagation delays etc. This may result in short-lived long routing loops, which disappear on link updates.

**Distance Vector Routing Protocol** [7, 11] operates by having each node  $i$  maintains a table, which contains a set of distance or cost  $\{d_{ij}(x)\}$ , where  $j$  is a neighbor of  $i$ . Node  $i$  treats the neighbor  $k$  as the next hop for a data packet destined for node  $x$ , if  $d_{ik} = \min \forall j \{d_{ij}(x)\}$ .

The routing table gives the best distance to each destination and which route to get there. To keep the distance set in the table up to date, each router exchanges information with all its neighbors periodically. If, as a result, a minimum distance to any neighbor of a node changes, this process will be repeated until all the nodes have updated the routing information. However, distance vector routing algorithm can cause both short-lived and long-lived loops due to updating the routing table with stale information. Though the looping problem can be eliminated using inter-nodal coordination method, which requires the routers to coordinate themselves mutually by confirmation messages in relatively stable environment. However, ad hoc networks are rapidly

changing mobile environments. This inter-nodal coordination mechanism is difficult to be used in the ad hoc networks.

The well-known implementation of distance vector routing, routing information protocol (RIP) [11], has been successfully used within local networks. However, it introduces counting to infinity problem due to network topology changes such as link or router failures. To improve the convergence of RIP, triggered updates, split horizon, poison reverse and path hold-down mechanisms [10, 11] have been integrated into different RIP implementations. Unfortunately, they solve some problems and introduce others at the same time. The usefulness of RIP to ad hoc environments is restricted due to the rapid topology change.

For the reasons mentioned above, both conventional link state and distance vector routing algorithms are not suitable for ad hoc network routing. The drawbacks of these protocols have to be fixed. DSDV is one of the proposed routing protocols for ad hoc networks. It aims to overcome the looping problem of RIP and retain the simplicity of RIP.

### 3 DSDV Protocol

Destination sequenced distance vector routing (DSDV) is adapted from the conventional Routing Information Protocol (RIP) to ad hoc networks routing. It adds a new attribute, sequence number, to each route table entry of the conventional RIP. Using the newly added sequence number, the mobile nodes can distinguish stale route information from the new and thus prevent the formation of routing loops.

#### Packet Routing and Routing Table Management [7]

In DSDV, each mobile node of an ad hoc network maintains a routing table, which lists all available destinations, the metric and next hop to each destination and a sequence number generated by the destination node. Using such routing table stored in each mobile node, the packets are transmitted between the nodes of an ad hoc network. Each node of the ad hoc network updates the routing table with advertisement periodically or when significant new information is available to maintain the consistency of the routing table with the dynamically changing topology of the ad hoc network.

Periodically or immediately when network topology changes are detected, each mobile node advertises routing information using broadcasting or multicasting a routing table update packet. The update packet starts out with a metric of one to direct connected nodes. This indicates that each receiving neighbor is one metric (hop) away from the node. It is different from that of the conventional routing algorithms. After receiving the

update packet, the neighbors update their routing table with incrementing the metric by one and retransmit the update packet to the corresponding neighbors of each of them. The process will be repeated until all the nodes in the ad hoc network have received a copy of the update packet with a corresponding metric. The update data is also kept for a while to wait for the arrival of the best route for each particular destination node in each node before updating its routing table and retransmitting the update packet. If a node receives multiple update packets for a same destination during the waiting time period, the routes with more recent sequence numbers are always preferred as the basis for packet forwarding decisions, but the routing information is not necessarily advertised immediately, if only the sequence numbers have been changed. If the update packets have the same sequence number with the same node, the update packet with the smallest metric will be used and the existing route will be discarded or stored as a less preferable route. In this case, the update packet will be propagated with the sequence number to all mobile nodes in the ad hoc network. The advertisement of routes that are about to change may be delayed until the best routes have been found. Delaying the advertisement of possibly unstable route can damp the fluctuations of the routing table and reduce the number of rebroadcasts of possible route entries that arrive with the same sequence number.

The elements in the routing table of each mobile node change dynamically to keep consistency with dynamically changing topology of an ad hoc network. To reach this consistency, the routing information advertisement must be frequent or quick enough to ensure that each mobile node can almost always locate all the other mobile nodes in the dynamic ad hoc network. Upon the updated routing information, each node has to relay data packet to other nodes upon request in the dynamically created ad hoc network.

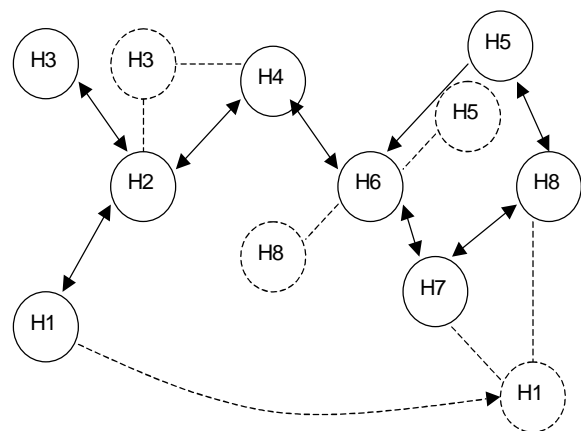
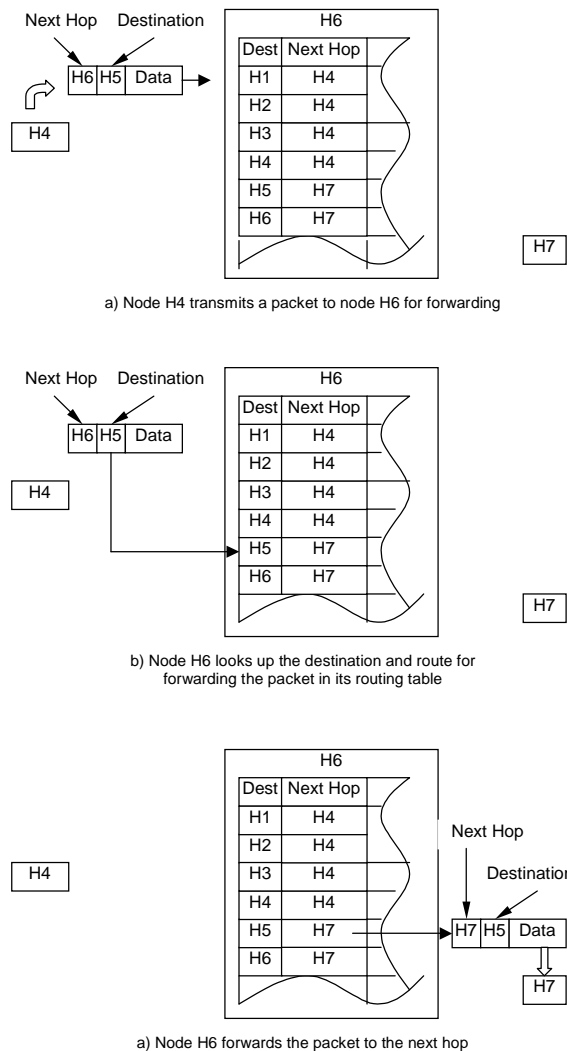


Figure 1: An example of the ad hoc networks

**Table 1: The routing table of node H6 at one instant [7]**

Dest	Next Hop	Metric	Seq.No.	Install
H1	H4	3	S406_H1	T001_H6
H2	H4	2	S128_H2	T001_H6
H3	H4	3	S564_H3	T001_H6
H4	H4	1	S710_H4	T002_H6
H5	H7	3	S392_H5	T001_H6
H6	H6	0	S076_H6	T001_H6
H7	H7	1	S128_H7	T002_H6
H8	H7	2	S050_H8	T002_H6

Figure 1 shows an example of an ad hoc network before and after the movement of the mobile nodes. Table 1 is the routing table of the node H6 at the moment before the movement of the nodes. The Install time field in the routing table helps to determine when to delete stale routes.

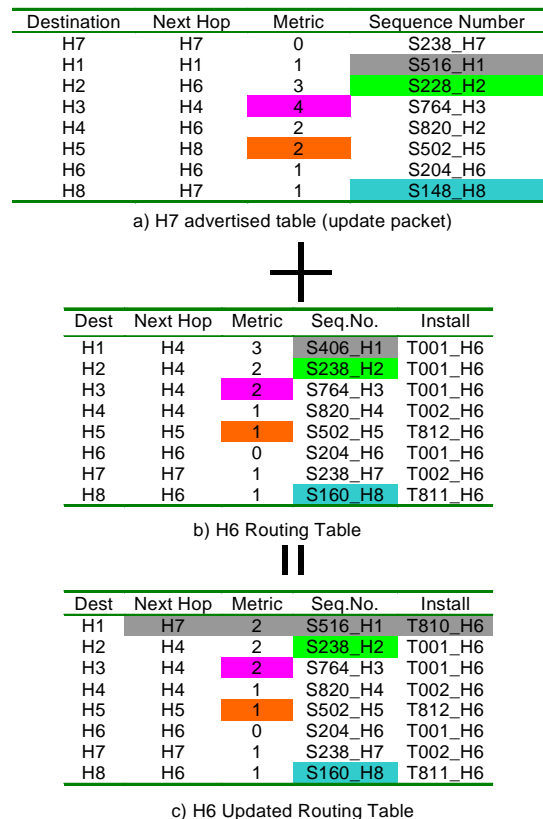


**Figure 2: DSDV packet routing example [12]**

Figure 2 shows an example of packet routing procedure in DSDV. Node H4 wants to send a packet to the node H5 as shown in Figure 1. The node H4 checks its routing table and locates that the next hop for routing the packet

is node H6. Then H4 sends the packet to H6 as shown in Figure 2a. Node H6 looks up the next hop for the destination node H5 in its routing table when it receives the packet (Figure 2b). Node H6 then forwards the packet to the next hop H7 as specified in the routing table as shown in Figure 2c. The routing procedure repeated along the path until the packet finally arrives its destination H5.

In the routing information updating process, the original node tags each update packet with a sequence number to distinguish stale updates from the new one. The sequence number is a monotonically increasing number that uniquely identifies each update from a given node. As a result, if a node receives an update from another node, the sequence number must be equal or greater than the sequence number of the corresponding node already in the routing table, or else the newly received routing information in the update packet is stale and should be discarded. If the sequence number of one node in the newly received routing information update packet is same as the corresponding sequence number in the routing table, then the metric will be compared and the route with the smallest metric will be used.



**Figure 3: An example of updating route information**

In addition to the sequence number and the metric for each entry of the update packet, the update route

information contains also both the address of the final destination and the address of the next hop. There are two types of update packets, one is called full dump, which carries all of the available routing information. The other is called incremental, which carries only the routing information changed since the last full dump. Figure 3 shows an example that a node handles an incremental update packet.

Figure 3 indicates that the node H7 in Figure 1 advertises its routing information with broadcasting the update packet to its neighbors. When the node H6 receives the update packet, it will check the routing information of each item contained in both the update packet and the its routing table and update the routing table. The entries with higher sequence numbers are always entered into the routing table (e.g., the entry H1 has newer sequence number - S516H1 in the update packet in Figure 3a. This sequence number is entered into the updated routing table Figure 3c after the routing update.), regardless of whether each of them have a higher metric or not. If an entry has the same sequence number, the route with smaller metric is entered into the routing table (e.g., the entry H5 has the same sequence number – S502\_H5 in both the update packet in Figure 3a and the current routing table in Figure 3b, but the entry H5 in the current routing table in Figure 3b has lower metric, so it enters the updated routing table in Figure 3c.). The items with old sequence numbers in the update packet are always ignored (e.g., H2 and H8 have old sequence number respectively in the update packet in Figure 3a, both of them are ignored in the updated routing table in Figure 3c.).

Each node in an ad hoc network must periodically transmit its entire routing table (full dump) to its neighbors most likely using multiple network protocol data units (NPDUs) [7]. The full dumps of the nodes can be transmitted relatively infrequently when little movement of mobile nodes is occurring. Incremental update packets are transmitted between the full dumps for partial changes of the routing table such as receiving new sequence numbers and fewer significant route changes (as shown in Figure 3a). The incremental routing update should be fitted in one NPDU. The mobile nodes are expected to determine the significance of the routing information changes to be sent out with each incremental advertisement. As shown in Figure 3a, the significant route change of H1 has to be included first, then the others with sequence number changes. When the significant changes increase with frequent varying of the network topology and the size of an incremental approaches the maximum size of a NPDU, a full dump is scheduled to make the next incremental become smaller.

### Responding to Topology Changes

Links can be broken when the mobile nodes move from place to place or have been shut down etc. The broken link may be detected by the communication hardware or be inferred if no broadcasts have been received for a while from a former neighbor. The metric of a broken link is assigned infinity. When a link to next hop has broken, any route through that next hop is immediately assigned an infinity metric and an updated sequence number. Because link broken qualifies as a significant route change, the detecting node will immediately broadcast an update packet and disclose the modified routes.

To describe the broken links, any mobile node other than the destination node generates a sequence number, which is greater than the last sequence number received from the destination. This newly generated sequence number and a metric of infinity will be packed in an update message and flushed over the network. To avoid nodes themselves and their neighbors generating conflicting sequence numbers when the network topology changes, nodes only generate even sequence numbers for themselves, and neighbors only generate odd sequence numbers for the nodes responding to the link changes.

Destination	Next Hop	Metric	Sequence Number
H7	H7	0	S238_H7
H1	H1	∞	S517_H1
H2	H6	3	S228_H2
H3	H4	4	S764_H3
H4	H6	2	S820_H2
H5	H8	2	S502_H5
H6	H6	1	S204_H6
H8	H7	1	S148_H8

a) H7 advertised table (update packet)



Dest	Next Hop	Metric	Seq.No.	Install
H1	H4	3	S516_H1	T001_H6
H2	H4	2	S238_H2	T001_H6
H3	H4	2	S764_H3	T001_H6
H4	H4	1	S820_H4	T002_H6
H5	H5	1	S502_H5	T812_H6
H6	H6	0	S204_H6	T001_H6
H7	H7	1	S238_H7	T002_H6
H8	H6	1	S160_H8	T811_H6

b) H6 Routing Table



Dest	Next Hop	Metric	Seq.No.	Install
H1	H7	∞	S517_H1	T810_H6
H2	H4	2	S238_H2	T001_H6
H3	H4	2	S764_H3	T001_H6
H4	H4	1	S820_H4	T002_H6
H5	H5	1	S502_H5	T812_H6
H6	H6	0	S204_H6	T001_H6
H7	H7	1	S238_H7	T002_H6
H8	H6	1	S160_H8	T811_H6

c) H6 Updated Routing Table

Figure 4: An example of link broken

Figure 4 illustrates an example of link broken. We assume the link between the node H1 and H7 is broken

in Figure 1. Node H7 detects the link broken and broadcasts an update packet (Figure 4a) to node H6. Node H6 updates its routing table with the newly received routing information (odd sequence number – S517\_H1 and  $\infty$  metric) of entry H1 (Figure 4c). It means that the link to node H1 is broken. If any other nodes send route update information of node H1 with even sequence number generated by node H1 previously, it is smaller than the current sequence number - S517\_H1 in Figure 4c, to node H6, which knows that the route information is stale, thus routing loop is prevented. If other nodes generate a newer odd sequence number with an infinity metric for node H1 and it is sent to node H6, which knows that the link to node H1 is broken via the odd sequence number and infinity metric.

The routes to a lost node will be re-established when the lost node comes back to the network and broadcasts its next update message with an equal or later sequence number and a finite metric. The update message will be disseminated over the whole network to indicate that the broken links have come back into service again. In any case, the entry containing a finite metric and an equal or later sequence number will supersede the corresponding entry with a metric of infinity in the routing table of a node.

DSDV also contains substantially more procedures for handling layer-2 and layer-3 routing and for dealing with the extension of base station coverage. The details of these procedures are referred to [7].

#### 4 Problems of DSDV

The main purpose of DSDV is to address the looping problem of the conventional distance vector routing protocol and to make the distance vector routing more suitable for ad hoc networks routing. However, DSDV arises route fluctuation because of its criteria of route updates. At the same time, DSDV does not solve the common problem of all distance vector routing protocols, the unidirectional links problem.

##### Damping Fluctuation

Fluctuation is a general problem arising in DSDV by the following criteria of route updates:

- Routes are always preferred if the sequence numbers are newer, routes with older sequence numbers are discarded.
- A route with a sequence number equal to that of an existing route is preferred if it has a better metric, and the existing route is discarded or stored as less preferable.

The broadcasts of routing information by mobile nodes are asynchronous events, though some regularity is

expected. In the case of many mobile nodes independently transmitting update messages and having markedly different transmission intervals, it may turn out that a particular mobile node receives new update packets in a way that causes this mobile node to consistently change route back and forth between different next hops, even though no network topology change has taken place. This fluctuation happens because of the above two route selection criteria. Conceivably, a mobile node can always receive two routes with equal sequence numbers or with a newer sequence number one after the other via different neighbors to the same destination, but the mobile node always gets the route with the worse metric first. This situation leads to the fluctuation with a continuing burst of new update packets. This can happen when there are many mobile nodes, all transmitting their updates irregularly. Alternatively, if the mobile nodes are acting independently and with and with markedly differently transmission intervals, the situation can occur with fewer mobile nodes.

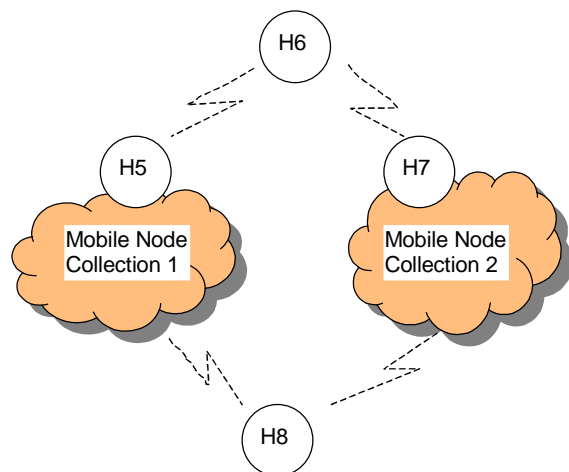


Figure 5: Receiving Fluctuating Routes

Suppose in the case of Figure 5, there are enough mobile nodes to cause the fluctuation problem. Two separate collections of mobile nodes both connect to a common destination H8 but with no other mobile nodes in common. Let's assume that all mobile nodes are transmitting updates every 15 seconds, that mobile node H5 has a route to H8 with 20 hops, and that mobile node H7 has a route to H8 with 19 hops, but that the routing information update from H5 arrives at H6 about 8 seconds before the routing information update from H7 to H6. This might happen every time when a new sequence number is issued from the mobile node H8. Thus the routing information in H6 for H8 is fluctuated back and forth for every newly issued sequence number of H8.

For solving this problem, DSDV requires that a mobile node receives a new update message, the route with the later sequence number must be available for use for the mobile node forwarding decisions, but it does not have to advertise the update immediately unless the route is to a previously unreachable destination, that the mobile node has to first consult its route settling time table as illustrated in Table 2 to decide the length of time to wait for a route with a better metric before advertising the update message. It requires that each mobile node keep two routing tables, one for forwarding packets, the other for advertising the incremental routing information packets. The settling time is stored in the latter with fields, destination address, last settling time, and average settling time. It is calculated by maintaining a running, weighted average over the most recent updates of the routes for each destination. The average settling time is used to determine the delay of an update advertisement.

**Table 2: Route settling time table of H6 at one instant.**

Dest Addr	Last settling time (Sec.)	Average settling time (Sec.)
H1	15	13
H2	13	11
H3	15	13
H4	8	8
H5	8	8
H6	8	8
H7	8	8
H8	12	11

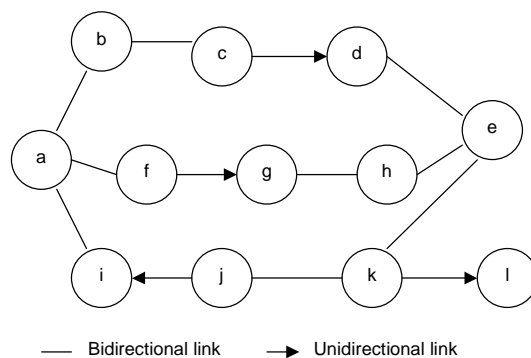
Note: The time values here is only for demonstration, not reality values

The average settling time in Table 2 is calculated with  $Average\ settling\ time = \sum x_j T_j / \sum x_j$ , where  $x_j$  is the metric (hops) from currently node to node  $j$  along the path to a destination,  $T_j$  is the last settling time of node  $j$ .

Suppose that a new routing information update arrives at H6, and sequence number in the new entry is newer than the sequence number in the currently used entry but has a worse metric. Then H6 must use the new entry in making subsequent forwarding decisions. However, H6 does not have to advertise the new route immediately and can consult its route settling time table to decide how long to wait before broadcasting the update. The average settling time is used to decide the delay (e.g., delay = Average settling time  $\times$  2) before advertising a route.

### Unidirectional Links

DSDV assumes that all wireless links in an ad hoc network are bi-directional. However, this is not true in reality. Wireless media is different from wired media due to its asymmetric connection. Unidirectional links are prevalent in wireless networks.



**Figure 6: An ad hoc network with unidirectional and bi-directional links [8]**

The presence of unidirectional links creates the following problems for DSDV as illustrated in Figure 4.

- **Knowledge Asymmetry:** Over the unidirectional links, the sink nodes know the existence of the source nodes, but the source nodes cannot assume the existence of the sink nodes.
- **Sink Unreachability:** In DSDV, the destination node initiates the path updates. Over a unidirectional link, there might be no way that a sink node can broadcast its existence, e.g. the node  $l$  in Figure 6.

In fact, these problems are very serious for DSDV in the case of Figure 6. As DSDV can only use bi-directional links for routing packets, it will ignore the links  $\overset{\rightarrow}{cd}$ ,  $\overset{\rightarrow}{fg}$ ,  $\overset{\rightarrow}{ji}$ , and  $\overset{\rightarrow}{kl}$ . As a result, the network will be perceived as three different parts.

A proposed solution for handling unidirectional links problem is presented in [8]. It requires that each node maintain enough information to distinguish between bi-directional and unidirectional links to its neighbors. Once knowledge of link orientations is available, appropriate routing decisions can be made. However, this proposed approach incurs higher communication and storage overheads of  $O(n^2)$ . The details of this approach are referred to [8].

### Other Drawbacks

It is difficult to determine the maximum setting time. DSDV does not support multi-path routing. The destination central synchronization suffers from latency problem. It has excessive communication overhead due to periodic and triggered updates. Each node must have a complete routing table.

## 5 Evaluation of DSDV

### Complexity

In DSDV, the time complexity is  $O(d = \text{network diameter})$  [9], and the communication complexity (link addition/failure) is  $O(N = \text{number of nodes in the network})$  [7,9].

### Performance

DSDV requires that each node maintain two tables. The bulk of the complexity in DSDV is generating and maintaining these tables. The updates are transmitted to neighbors periodically or scheduled as needed. As growing of mobility and number of nodes in the network, the size of the bandwidth and the routing tables required to update these tables grows simultaneously. The overhead for maintaining and updating these tables will increase correspondingly. It is natural that heavy routing overhead will degrade the performance of the network.

Simulation results in [2] shows that DSDV fails to converge if nodes don't pause for at least 300 seconds during movement; the packet delivery ratio is in the range of 70%-92% at higher rate of mobility; packet loss is mainly caused by stale routing entries; in periodic updates transmission, routing overhead is constant with respect to the mobility rate; nearly optimal path can be selected in routing procedure.

Another simulation was done in [5] under the condition of 1000m X 1000m rectangular movement region, 350m constant radio range for each mobile node, 0.4-0.6 m/sec for low mobility, 3.5-4.5 m/sec for high mobility, 512 bytes for the packets length, 1-10 connections/node, and 30 mobile nodes. The result is shown in Table 3.

**Table 3: Simulation result for DSDV in [5]**

	Low mobility (0.4-0.6 m/sec)	High mobility (3.5-4.5 m/sec)
Delivery ratio of packets (%)	98-100	98-100
End-to-end delay (ms)	6.3-6.6	7.2-7.7
Normalized routing load	0.07 (3 connections/node)	0.24 (3 connections/node)
	0.025 (9 connections/node)	0.07 (9 connections/node)

From the result in Table 3, we can see that the end-to-end delay and the routing load increase with the mobility; but the routing load decreases with the number of connections of each node at same mobility.

### Stability and scalability

DSDV guarantees loop-free routes in routing packets. It uses incremental and full dump for updates. The incremental update is used so that the entire routing table needs not to be transmitted for every change in the

network topology. However, DSDV requires the full dump update periodically, regardless of the number of changes in the network topology. In this aspect, DSDV is not efficient in route updating. It also limits the number of nodes that can join in the network. In addition, whenever the topology of a network changes, DSDV is unstable until update packets propagate throughout the network.

### QoS Routing with DSDV

The requirements of QoS routing in a wireless network that supports multimedia traffic and/or interconnected to a wired ATM backbone were identified in [2]. For general purpose, the requirements of QoS routing in ad hoc networks that support real time traffic are summarized here: (1) *Band reservation*: The ad hoc network must allocate bandwidth at call setup time in order to support real time connections. (2) *QoS routing*: To support QoS for real time traffic, the mobile nodes not only need to know the minimum delay path to destination but also need to have the knowledge of the bandwidth available on that path. At call setup time, the bandwidth has to be available and reserved. Otherwise the call setup request will be rejected. Thus, traditional distance vector routing algorithms are not adequate. Routing with QoS is required for efficiently managing bandwidth resources. (3) *Congestion control*: Even though using QoS routing can manage the bandwidth resources at call setup time, network congestion due to the dynamics of mobility and of traffic patterns has to be controlled via applying selective packet dropping and input rate control, etc. (4) *Mobility*: The inter-working of mobility as well as the allocation and maintenance of bandwidth resources are critical to an ad hoc network, especially when it interconnects to a wired backbone.

A QoS routing simulation was done in a wireless network, which interconnected to ATM in [3]. In this case, QoS routing information permits to extend the ATM virtual circuit service to the wireless network with renegotiation of QoS parameters at the gateway. In the simulation, a cluster TDMA architecture [3] was used. Only bandwidth was considered for the quality of service due to bandwidth guarantee is the most critical requirement for real time application. The available bandwidth computation is carried out independently at each node and is piggybacked on the DSDV loop-free routing algorithm.

The simulation was done in the environment consists of 20 mobile nodes randomly roaming in a 1000 X 1000 feet square, the radio range is 500 feet, the routing update period is 100 ms. After every routing update period, a mobile moves  $x$  feet with probability  $P$ . Two cases with average speeds 11 km/hour and 55 km/hour were considered respectively. It is assumed that virtual circuit calls between arbitrary source and destination pairs are generated according to a Poisson distribution.

Call duration is exponentially distributed. It was monitored that the total packet throughput versus offered traffic load as stations move at various speeds. The performance of the cluster TDMA architecture was measured.

In the simulation, the TDM frame is same as the routing update period (100 ms) with 10 data slots per frame. Real time traffic was modeled by a continuous source, which generates a packet per TDM frame. Packets will be dropped if their “age” in the network is over 10 TDM frames. Each real time connection is mapped into a virtual circuit. A call is accepted only when the available bandwidth is above the required QoS. Otherwise, the call is rejected. When one part of the network becomes congested because of virtual circuit routing due to node movements, and no enough bandwidth is available to support the initial QoS, low priority packets are dropped first. Only high priority packets are transmitted and QoS is gracefully degraded. If the congested part can't transmit even the high priority packets, the virtual circuit is temporarily broken down.

An experiment with cluster TDMA with Bellman-Ford routing was also done in [3]. The results of both of the experiments are given in Table 4. The details of the simulation are referred to [3].

**Table 4: QoS Routing Performance in DSDV [3]**

Node speed	11 km/hour		55 km/hour	
	DSDV+QoS	B-F	DSDV+QoS	B-F
Routing	88.4%	76.5%	88.2%	73.5%
Packet received	88.4%	76.5%	88.2%	73.5%
Packet dropped (low priority)	10.2%	15.4%	10.0%	18.3%
Packet dropped (high priority)	1.4%	8.1%	1.8%	8.2%
Offered load (pkt/frame)	6.19	11.80	5.66	11.80
Throughput (pkt/frame)	5.47	9.02	4.99	8.68
Packet delay (ms/packet)	189	229	207	232
Packet in loop	0%	0.2%	0%	0.2%
Out-of-seq. pkts	11.2%	11.3%	13.3%	17.8%

The simulation shows that QoS routing can be effectively used with DSDV in the support of real time traffic in multi-hop wireless network. Compared to B-F routing, the simulation results illustrate that the DSDV+QoS algorithm has better packet delivery rate, less packet delay, less dropped packets and less packets in loop.

## 6 Current Status

DSDV is a well-known routing algorithm proposed for ad hoc network routing, but it has many problems as mentioned in previous sections. Currently, there are no standard specifications and commercial implementations

available yet for DSDV, but one DSDV simulator has been implemented with C++ at UC Berkeley [13].

Research work on improvement of DSDV is still active. Many improved protocols based on DSDV have been developed. These improvements of DSDV include Global State Routing (GSR) described in [4], Fisheye State Routing (FSR) described in [6], Ad Hoc On-Demand Distance Vector Routing (AODV) [1], etc. DSDV is a protocol under study. In my opinion, it will be substituted with its improvements in real applications in the future.

## 7 Summary

DSDV is an adaptation of classical distance vector routing protocol to ad hoc networks. In DSDV, two routing tables are maintained at each of the nodes. One of them is the routing table, which contains a complete list of addresses of all other nodes in the network. The other contains the setting time data for each destination node. It is used to determine the time for update advertisement. The routing of updates and packets between nodes is based on these tables. Along with each node's address, the routing table contains the address of next hop, route metric, destination sequence number, etc. Route updates are broadcasted periodically or scheduled as needed in the network. Routes are always selected with later sequence number. If the sequence numbers are identical, the route with smallest metric will be selected. These criteria guarantee loop-free routes, but they also induce routes fluctuation. DSDV is a bi-directional protocol, which unavoidably has the unidirectional links problem in ad hoc networks. It is expected further studies on the problems of DSDV as mentioned in previous sections. Compared to Bellman-Ford routing, QoS with DSDV routing has substantial improvement in the routing results, but it is expected that further researches including multi-path routing, QoS multicasting in DSDV and ad hoc networks. Many improvements of DSDV have been developed. Hopefully, these newly proposed protocols will be used in reality.

## References

- [1] Boppana Rajendra V. Marrina Mahesh K. and Konduru Satyadeva P.: An Analysis of Routing Techniques for Mobile and Ad Hoc Networks, 1999. <[http://www.cs.utsa.edu/faculty/boppana/papers/Hip\\_c99-preprint.pdf](http://www.cs.utsa.edu/faculty/boppana/papers/Hip_c99-preprint.pdf)>
- [2] Broch Joch, Maltz David A., etc.: A performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols, Fourth Annual



- ACM/IEEE International Conference on Mobile Computing and Networking, 1998.  
<<http://www.cs.pdx.edu/~singh/courses/ass1pap.pdf>>
- [3] Chen T.-W., Tsai J. T. and Gerla M.: QoS Routing Performance in Multihop, Multimedia, Wireless Networks, 1997.  
<<http://www1.ics.uci.edu/~atm/adhoc/paper-collection/gerla-qos-routing-icupc97.pdf>>
- [4] Chen Tsu-Wei and Gerla Mario: Global State Routing: A New Routing Scheme for Ad Hoc Wireless Networks.  
<<http://www1.ics.uci.edu/~atm/adhoc/paper-collection/gerla-gsr-icc98.pdf>>
- [5] Das Samir R., Castaneda Robert and Yan Jiangtao, etc.: Comparative Performance Evaluation of Routing Protocols for Mobile, Ad Hoc Networks.  
<<http://www.cs.washington.edu/homes/rimli/papers/manet.pdf>>
- [6] Pei Guangyu, Gerla Mario, Chen Tsu-Wei: Fisheye State Routing in Mobile Ad Hoc Networks.  
<<http://systems.cs.colorado.edu/grunwald/MobileComputing/Papers/fisheye-state-routing-in-mobile-ad-hoc-networks.pdf>>
- [7] Perkins Charles E., Bhagwat Pravin: Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, London England UK, SIGCOMM 94-8/94.  
<<http://www.cise.ufl.edu/~helal/6930F01/papers/DS DV.pdf>>
- [8] Prakash Ravi: Unidirectional Links Prove Costly in Wireless Ad Hoc Networks, ACM 1999 1-58113-174-7/99/08.  
<<http://www.cs.pitt.edu/~kirk/cs3550/p15-prakash.pdf>>
- [9] Royer Elizabeth M.: A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks, IEEE Personal Communications, April 1999.  
<<http://www.ces.clemson.edu/~rsass/courses/NRG/Papers/Royer.pdf>>
- [10] Schmid Andreas and Steigner: Avoiding Counting To Infinity In Distance vector Routing, Kluwer Academic Publishers, Netherlands, 2001.  
<<http://www.uni-koblenz.de/~steigner/labor/papers/ripmti.pdf>>
- [11] Tanenbaum Andrew S.: Computer Networks, Prentice Hall, Inc., 1996, ISBN 7-302-02410-3.
- [12] Thampuran Santhosh R.: Routing Protocols for Ad Hoc Networks of Mobile Nodes.  
<<http://www-unix.ecs.umass.edu/~sthampur/Papers/AdhocRouting.pdf>>
- [13] The Vint Project: The ns Manual (formerly ns Notes and Documentation), 28 February, 2002, [referred 02.04.2002].  
<<http://www.isi.edu/nsnam/ns/doc>>