A Reliable Cluster-based Routing Algorithm for MANET

Hiroyuki Narumi*, Yoh Shiraishi**, and Osamu Takahashi**

*Graduate School of Systems Information Science, Future University Hakodate, Japan
**Future University Hakodate, Japan
{g2108031*, siraisi**, osamu**}@fun.ac.jp

Abstract - It is known that conventional reactive routing protocols in Mobile Ad-hoc Networks (MANET) decrease the packet delivery rate (PDR) in high-mobility and -traffic environments. This is because temporary congestion occurs due to control packets flooding the route establishment and the route break in long route communications. Generally, clustering is an effective method for improving the capability for dealing with node mobility. However, current clustering methods have the long route communication problem described above. Therefore, we propose a reliable clustering method, called the "Cluster-by-Cluster routing method", for reining in the emergence of long route communications by cascading short route communications. Furthermore, we implement our proposed method on Dynamic Source Routing (DSR), which is a representative of reactive routing protocol, and evaluate the effectiveness with computer simulation. As a result, in a high-mobility environment (10.0 m/s node velocity, pause time 0.0 s), our method improves PDR about 52% better than DSR and decreases the routing overhead by about 92%. In addition, in a large-scale network (250 nodes), our method improves PDR by about 22% and decreases the routing overhead by about 95%. By these results, we conclude that our proposed method can be used for reliable communication in MANET.

Keywords: Routing, Cluster, Large-Scale Network, Long Route Communication

1 INTRODUCTION

As mobile computers, such as handheld computers and tablets, continue to show improvements in convenience, memory capacity, storage capacity, and mobility, Mobile Ad hoc Networks (MANET) have become more popular. A MANET is characterized by multi-hop wireless links in the absence of any cellular infrastructure as well as by frequent host mobility. Reactive protocols, such as Dynamic Source Routing (DSR) [1], are representative protocols in MANET. We show the performance deterioration of the reactive protocol with long route communications in a high-mobility environment through basic experiments. In response to this problem, we propose a reliable Cluster-by-Cluster routing method for dividing an end-to-end long route into a combination of short routes. Furthermore, we apply our method to DSR and evaluate routing performance and scalability. The Cluster-by-Cluster routing results are compared to the results of a popular reactive routing protocol – DSR. Finally we show the effectiveness of Cluster-by-Cluster routing in high-mobility and -traffic environments.

The rest of this paper is structured as follows. Section 2 presents the problem of long route communications in MANET. Section 3 presents common clustering scheme in MANET. Section 4 presents our proposed method. Section 5 analyzes and evaluates experimental results. Section 6 concludes this paper and provides a brief outlook on our future work.

2 BASIC EXPERIMENT

We evaluated a reactive protocol's unicast performance in networks of varying sizes (50, 100, 150, 200, and 250 nodes) using the network simulator – version 2 (ns-2) [6]. In the simulation, we used the 802.11 standard with a transmission range of 250m. Furthermore, we chose DSR as a routing protocol, and node density of 100 nodes/km^2. All nodes were constantly (i.e, 0s pause time) moving according to the random waypoint model. A constant node velocity of 2.0 m/s was used. To measure the packet drop rate, we set up one constant bit-rate communication between the given source and destination node.

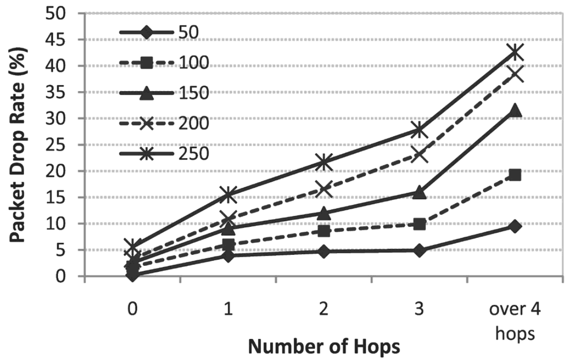


Figure 1: Packet drop rates vs. route length

Figure 1 shows the packet drop rates of the respective network sizes versus route lengths of delivered data packets. The packet drop rates in each network size increase, as route lengths get longer. For all network sizes, the packet drop rates increase quickly when route lengths are long (over 4 hops). The packet drop rate with over 4-hop communication in a large-scale network (250 nodes) increases to about 45%. As just described, long route communications in MANET cause the packet drop rate to increase. This is because extensive congestion occurs by the flooding of control packets when a source node establishes a route to long-distance destination node, and forwarding nodes drop data packets by end-to-end route breaks when they move constantly and quickly.

Consequently, it is important to reduce such flooding of control packets in the route discovery process and emergence of long route communications drop many data packets.

3 RELATED WORK

In this section, we present a common clustering scheme in a MANET, and describe an issue about long route communications in the scheme.

3.1 Clustering in MANET

To efficiently manage a network for routings (especially, the route discovery process) in large-scale and high-mobility ad hoc networks, many clustering schemes and hierarchical routing protocols commencing with the Cluster Based Routing Protocol (CBRP) have been proposed [2, 3, 4, 9, 10, 13, 14]. Figure 2 shows an example of a network formation constructed using the common clustering method used in these protocols. A cluster is a group of nodes with one of them assigned as a cluster head. A cluster is identified by its Cluster-ID, and they are either overlapping or disjointed. A cluster head will have complete information about group membership. Each node in the network has determined its corresponding cluster head(s) and, therefore, has information as to which cluster(s) it belongs to. All nodes within a cluster, except the cluster head, are called member nodes of its cluster. Any node a cluster head may use to communicate with an adjacent cluster is called a gateway node.

By dividing the network into several clusters, we are able to drastically reduce control packets called Route Requests (RREQs)[1] when a source node establishes a source route to the destination node. The route discovery process of conventional ad hoc routing protocols floods the route establishment with a large amount of control packets. However, in the hierarchical routing protocol, only cluster heads are flooded with RREQ packets in search for the source route (Figure 3). Therefore, a source node can quickly create a source route to a particular destination node. For these reasons, it is known that the hierarchical routing method has the capacity to deal with node movement compared to conventional ad hoc routing protocols such as DSR.

However, in a large-scale network, in which there are comparatively many clusters, a situation occurs where nodes physically located far from each other must communicate over two or more clusters. In such situation, the conventional hierarchical routing methods described above create long routes between source and destination nodes. As described in the previous section, if the source node sends data packets using the long route, the packet drop with the route break will increase with the common clustering method, as well as conventional ad hoc routing protocols. To deal with such packet drops caused by the movement of forwarding nodes, several hierarchical routing protocols have "Route Maintenance" and "Local Route Repair" schemes to salvage dropped packets. However, there are very few approaches for reining in the emergence of long route communications.

[1] Request message of the Route Discovery process.

We adopt the concepts of an overlay clustering and our Cluster-by-Cluster routing method to reduce long route communications. Finally, we show that our Cluster-by-Cluster routing method obtains better packet delivery rates and produces lower network traffic than ever before.

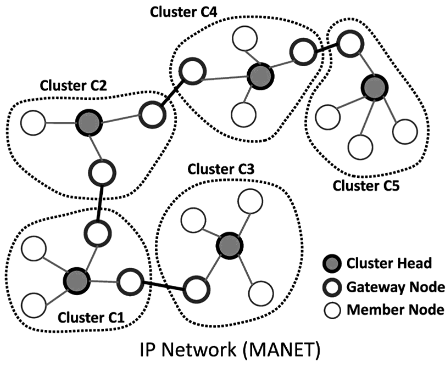


Figure 2: Clustering in MANET

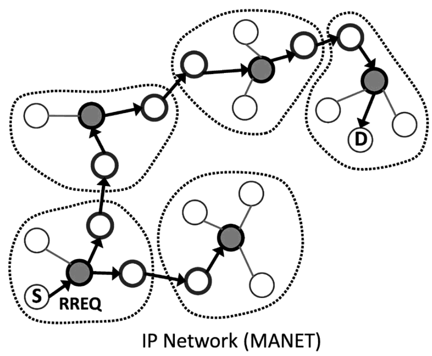


Figure 3: Route discovery process in conventional hierarchical routing method

4 PROPOSED METHOD

In this section, we design Cluster-by-Cluster routing method, and define a protocol layer at which Cluster-by-Cluster routing operates.

4.1 Overview

Cluster-by-Cluster routing forms an overlay network on a physical network, and groups physically close nodes as overlay clusters, and data packets a source node sent out are forwarded cluster by cluster. Figure 4 shows an example of this process. All nodes have complete information on the relative location of all current clusters, as described in a later section. Therefore, the source node is able to generate a proper "cluster path" in which data packets delivered on the overlay network and forwarding nodes can specify "temporal" gateway nodes currently situated at boundaries of each cluster on the path. Consequently, we can divide one physically long route communication into several physically short route communications by specifying such gateway nodes as temporal destination nodes. This leads to a reduction in long route communications, even if a source and a destination node are physically located far from each other. Figures 4 and 5 depict examples of the route divide. Suppose source node S sends data packets to destination node D (Figure 4). Conventional hierarchical and ad hoc routing protocols create a long source route of 6 hops (Figure 5b). On the other hand, Cluster-by-Cluster routing makes 5 short source routes leveraging information about the relative location of all current clusters. In particular, a 6-hop route in conventional protocols (Figure 5b) is divided into 0-, 0-, 0-, 1-, and 1-hop short routes in Cluster-by-Cluster routing (Figure 5a). "0 hop" means a direct communication between a source and a destination node.

To dynamically construct these route divide process, we propose four core algorithms, as described below.

1) Clustering Algorithm
2) Acquisition of Cluster Adjacency Information
3) Cluster Path Decision
4) Cluster-by-Cluster Routing

We present the above four algorithms in the following sections.

4.2 Clustering Algorithm

We apply Random Landmarking (RLM) [5] as the base of clustering algorithm in Cluster-by-Cluster routing to map the physical topology to its overlay topology. RLM is one of the representative clustering methods for constructing P2P networks in MANETs, such as MADPastry [11, 12]. Each RLM node has a unique but dynamic overlay ID called node ID. Since there are generally no stationary works without any fixed cluster head, it uses a set of *landmark keys*. Landmark keys are simply overlay IDs that divide the overlay ID space into equal-sized segments. For example, in a hexadecimal-based ID space, an appropriate set of landmark keys could be: C000…000, C100…000, C200…000, . . . , CE00…000, CF00…000. These landmark keys are broadcasted within the corresponding cluster as part of beacon messages by cluster heads. RLM constructs clusters of physically close nodes that share a common landmark key. Therefore, physically close nodes in RLM are also quite likely to be close to each other in the overlay ID space. Figure 6 shows dynamic clustering in RLM. In our proposed method, we made two changes to RLM, "Node ID Publication" and "Node ID Lookup".

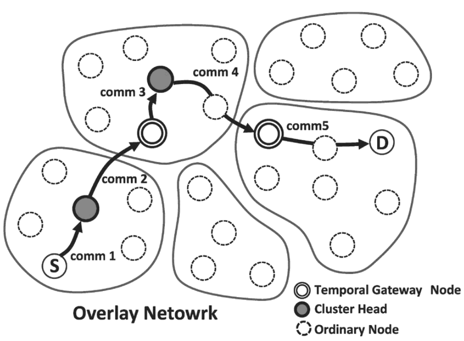
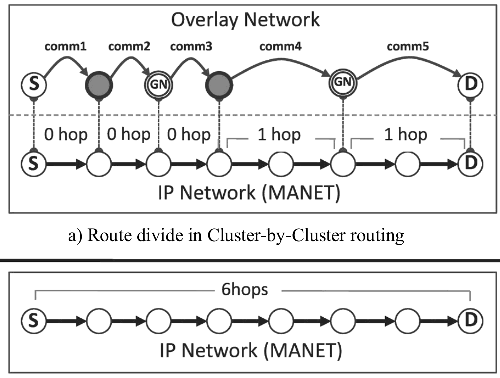


Figure 4: Cluster-by-Cluster routing



a) Route divide in Cluster-by-Cluster routing

b) Long route communication in conventional routing

Figure 5: Long route communication and combination of short route communications

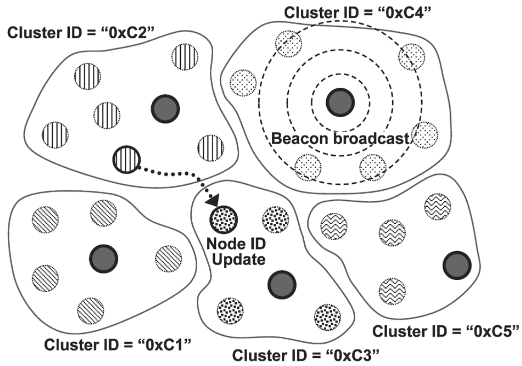


Figure 6: Clustering and node ID update in Random Landmarking

Node ID Publication. A node moved to a different cluster has to update its node ID to indicate its current position on the overlay network. As a result, it simply replaces its current landmark key with a new one. Then it is able to rejoin the new cluster (Figure 6). To send data packets to any destination node, a source node has to have information about the "current" node ID of the destination. Therefore, it publishes the new node ID to the node that has the numerically closest node ID to its node ID. All nodes on the overlay network obtain its current node ID, and can send data packets to it. We modified this publication process to optimize with Cluster-by-Cluster routing, as shown in Figure 7.

Node ID Lookup. To communicate with any node, a source node obtains that node's current node ID. Figure 8 shows the node ID lookup process in our method.

Def inition:
・***hash(A)*** **:** the hash value of *A*
・***generateStoreMsg(K, I, A)*****:** Store message contains *K*, *I*, and *A*.
・C_i **:** the cluster ID of the cluster C_i, $0 \leqq i < n$, where *n* is the number of clusters.

(a) Sender:
(1) $A \leftarrow$ my node address; $I \leftarrow$ my node ID;
(2) $K \leftarrow hash(A)$;
(3) $M \leftarrow generateStoreMsg(K, I, A)$;
(4) send *M* to a cluster head has cluster ID C_i, s.th. $|C_i - I|$ is minimal;

(b) Receiver:
(1) store <*K*, *A*, *I*> pair of *M*;

Figure 7: Node ID publication

Definition:
・ ***hash(D)*** **:** the hash value of D
・ ***generateLookupMsg(K, I)*** **:** Lookup message searches for K.
・ C_i **:** the cluster ID of the cluster C_i, $0 \leqq i < n$, where n is the number of clusters.

(a) Sender:
(1) $D \leftarrow$ destination node address; $I \leftarrow$ my node ID;
(2) $K \leftarrow hash(D)$;
(3) $M \leftarrow generateLookupMsg(K, I)$;
(4) send M to a cluster head has cluster ID C_i, s.th. $| C_i - K |$ is minimal;

(b) Receiver:
(1) return $<K, A, I>$ pair to node I;

Figure 8: Node ID lookup

4.3 Acquisition of Cluster Adjacency Information

To divide long route communications effectively, Cluster-by-Cluster routing collects each cluster's adjacency information, exploiting beacon messages broadcasted from cluster heads. Figure 9a shows this process. The beacon message contain a source cluster head's "cluster ID", which is called a landmark key in RLM. Then the cluster head broadcasts the beacon piggybacked with the RREQ message within its cluster. Figure 9b shows the beacon reception process. A node that received the beacon will return a "neighbor feedback" message to the source cluster head as a "temporal" gateway node if it is located at a boundary between the source cluster and the other different clusters. Otherwise, it forwards or discards the beacon as necessary. Figure 10a shows the neighbor feedback transmission process, and Figure 10b depicts the neighbor feedback reception process. Gateway nodes send the neighbor feedback piggybacked with the Route Reply (RREP) [2] message. Consequently, the cluster head that received the feedback can create a temporal bi-directional route to the gateway, the neighboring cluster. Cluster-by-Cluster routing exploits these bi-directional routes to efficiently minimize the flood of traffic during the route discovery process.

Definition:
・ ***generateBeacon (Adj, K, Sq)*** **:** Beacon message contains Adj, K, and Sq.

(a) Sender:
(1) $C \leftarrow$ my cluster ID; $Sq \leftarrow$ current beacon sequence number;
(2) $Adj \leftarrow$ cluster adjacency information I collected;
(3) $B \leftarrow generateBeacon(Adj, K, Sq)$;
(4) broadcast B piggybacked on RREQ with TTL = k;
(5) schedule next transmission time;

(b) Receiver:
(1) remove expired information from my Beacon Table;
(2) $C \leftarrow$ cluster ID of sender cluster head;
(3) $Hr \leftarrow$ number of hops of B;
(4) $Hc \leftarrow$ number of hops of the beacon of my current cluster;
(5) if ($Hr < Hc$) {
(6) join to the cluster C;
(7) $T \leftarrow$ current TTL of B;
(8) if ($T < k$) {
(9) re-broadcast B with sequence number of $T+1$;
(10) }
(11) }
(12) else {
(13) // I am a gateway node
(14) return neighbor feedback to the sender cluster head of C;
(15) }
(16) store the route to the sender cluster head;

Figure 9: Beacon transmission and reception algorithm

[2] Reply message of the Route Discovery process.

Definition:
・ ***generateClusterAdjacencyInf(Sq, C_m)*** **:** Cluster adjacency information generated from Sq and C_m.
・ ***generateNeighborFeedback(Adj)*****:** Neighbor feedback contains Adj.
・ ***p*** **:** Maximum number of neighbor feedback which returns to a cluster head.

(a) Sender:
(1) monitor surround in random seconds;
(2) $C_d \leftarrow$ destination cluster ID;
(3) $S \leftarrow$ sum of neighbor feedbacks that other nodes sent to the cluster C_d;
(4) if ($S < p$) {
(5) $Sq \leftarrow$ current sequence number of B; $C_m \leftarrow$ my node ID;
(6) $Adj \leftarrow generateClusterAdjacencyInf(Sq, C_m)$;
(7) $F \leftarrow generateNeighborFeedback(Adj)$;
(8) return F to the cluster head of C_d;
(9) }

(b) Receiver:
(1) $F \leftarrow$ received neighbor feedback;
(2) // sender is a gateway node
(3) extract the route to the sender from F and store it;
(4) $Sq \leftarrow$ sequence number of F;
(5) $Sq_c \leftarrow$ sequence number of last feedback that I received before;
(6) if ($Sq < Sq_c$) {
(7) $Adj \leftarrow$ cluster adjacency information that is extracted from F;
(8) store Adj;
(9) }

Figure 10: Neighbor feedback transmission and reception algorithm

Figure 11 shows the acquisition process of cluster adjacency information. Suppose node A belongs to the cluster C2. If node A receives the beacon from the cluster head of C1, then A recognizes itself as the gateway node between clusters C1 and C2. Node A sets the information that C1 is next to C2 with neighbor feedback, and then returns it to C1's cluster head. C1's cluster head analyzes the feedback and updates its cluster adjacency information. After that, C1's cluster head broadcasts the beacon with the up-to-date cluster adjacency information next time, as shown in Figure 12. All cluster heads perform this operation periodically, and all nodes that receive beacons store the adjacency information each time. Finally, all nodes can obtain information about the relative location of all current clusters, as described later in Section 4.4.

Short Route Cache. Each node that received a beacon stores the route to the sender cluster head. In addition, each cluster head that received a neighbor feedback stores the route to the sender gateway node. This leads to effective packet forwarding and the prevention of route discovery in Cluster-by-Cluster routing.

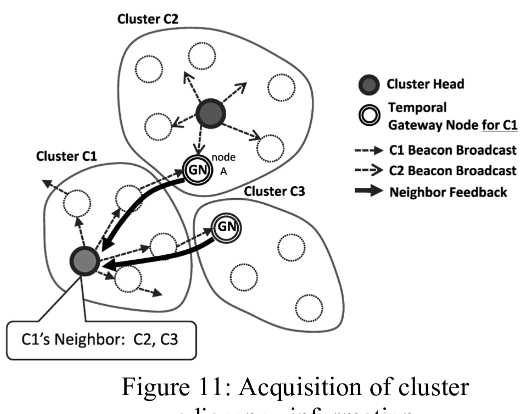


Figure 11: Acquisition of cluster adjacency information

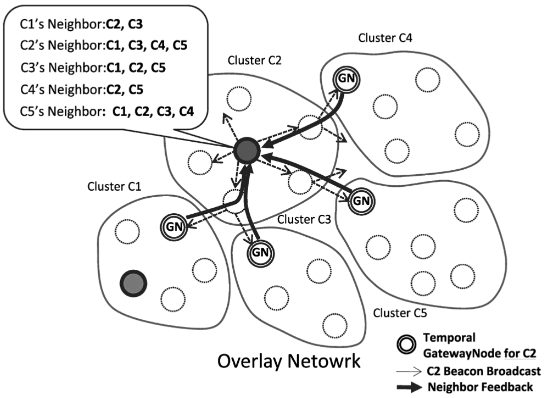


Figure 12: Acquisition of all cluster adjacency information

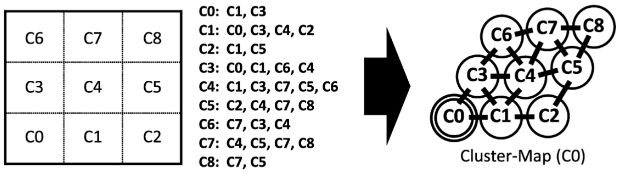
4.4 Cluster Path Decision

All nodes are able to create a cluster path to any destination cluster autonomously by exploiting its own cluster adjacency information. Before creating the path, each node generates a "cluster map", which is the relative cluster location map that base point is its cluster from the cluster adjacency information. We define clusters next to the source cluster as "Level 0", and define clusters next to Level 0 clusters as "Level 1", and so on. Figure 13 shows an example of the cluster map of cluster C0. Suppose there are nine clusters on the overlay network, and C0's cluster head collected the cluster adjacency information as shown in Figure 13a. Then nodes belonging to cluster C0 can generate the cluster map of cluster C0, as shown in Figure 13b.

By exploiting this cluster map, Cluster-by-Cluster routing creates a cluster path to any cluster the destination node belongs to. Figure 14 shows the cluster path decision algorithm. Suppose node S, belonging to cluster C0, sends data packets to any node belonging to cluster C4. From cluster C0's viewpoint, cluster C4 is a Level 1 cluster. Consequently, node S picks out Level 0 clusters next to cluster C4 from the cluster map. In this case, node S obtain two Level 0 clusters (C1 and C3), it chooses only one cluster according to any metric. Suppose node S chooses C3 as Level 0, the cluster path decision process is terminated. Therefore, node S obtains the path: "C0, C3, C4". Finally, as described in Section 4.1, we can shorten the physical route per communication using this cluster path.

4.5 Cluster-by-Cluster routing

A Cluster-by-Cluster routing delivers data packets cluster by cluster using the cluster path. Figure 15 shows the routing algorithm and, as mentioned above, Figure 4 depicts an example of this process. When a source node sends a data packet to a destination node, it first sends the packet to its own cluster head using a cached route obtained from the beacon. Then, the cluster head forwards it to the appropriate gateway node of the next cluster using a cached route obtained from the neighbor feedback. Cluster heads and gateway nodes of the clusters on the cluster path forward the packet until it reaches the destination cluster. Finally, the gateway node of the destination cluster forwards it to the destination node directly.



a) C0's cluster adjacency information

b) Cluster map of cluster C0

Figure 13: Cluster map generation

```
Def inition:
・route[]: the cluster path to the destination cluster.

Cx ← destination cluster ID;
l ← level of Cx;
route[l] ← Cx;
while ( l > 0 ) {
    Cx ← cluster ID of level (l – 1) cluster that Cx is adjacent cluster;
    l ← l – 1;
}
// route[] is a cluster path to the destination cluster
```

Figure 14: Cluster path decision algorithm

```
if ( I am a f inal destination node ) {
    pass a packet to the upper layer;
}
else if ( I am belonging to a destination cluster ) {
    directly forward a packet to a final destination node;
}
else if ( I am a cluster head ) {
    forward a packet to a gateway node of next adjacent cluster;
}
else {
    // I am a gateway or an ordinary node
    forward a packet to my cluster head;
}
```

Figure 15: Cluster-by-Cluster routing algorithm

4.6 Layer Definition

In the TCP/IP model, the Cluster-by-Cluster routing method lies on the Internet layer, as shown in Figure 16. Cluster-by-Cluster "overhears" the surrounding packets to control the transmission of the neighbor feedback. Therefore, Cluster-by-Cluster routing operates essentially in promiscuous mode. We can use any ad hoc routing protocol supporting the promiscuous mode in Cluster-by-Cluster routing.

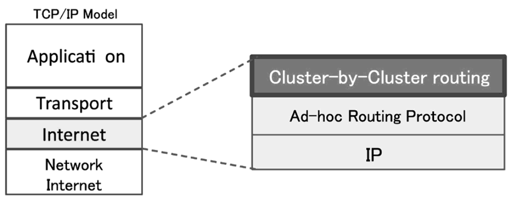


Figure 16: The TCP/IP layer model

5 SIMULATION RESULTS

To evaluate the performance of Cluster-by-Cluster routing, we implemented it as a routing agent in ns-2 [6]. Additionally, in this simulation, Cluster-by-Cluster routing operates on the popular reactive routing protocol - DSR.

Cluster-by-Cluster routing's results were compared to those of DSR.

5.1 Parameters

Table 1 lists the simulation parameters. For all simulations, we used the 802.11 standard with a transmission range of 250 m. Furthermore, we chose a node density of 100 nodes/km^2. All nodes were constantly (i.e. 0s pause time) moving with V_{const} according to the random waypoint model, except 9 nodes allocated in a reticular pattern. That is, this network was in a high-mobility environment. Each simulation run lasted twenty simulated minutes. For Cluster-by-Cluster routing, above-mentioned 9 nodes operated as the cluster head. On the other hand, for DSR, they operate as just fixed node. The Flow State option for DSR was disabled in our simulations.

To measure the packet delivery rate and the routing overhead, two random nodes sent out a UDP message to a random node every second between 50 and 1150 seconds.

5.2 Evaluation Metrics

We defined three metrics to evaluate the performance of both routing agents.

Packet Delivery Rate - the percentage of all data packets eventually delivered to the correct destination node. This metric is calculated by the following formula.

$$\text{Packet Delivery Ratio (\%)} = \frac{\#\ \text{of Packets Received}}{\#\ \text{of Packets Sent}} \times 100 \quad (1)$$

Routing Overhead - the total network traffic in packets that is created during the twenty simulated minutes. In the case of Cluster-by-Cluster routing, this figure comprises all route-level packets that are created by a Cluster-by-Cluster routing node: publications, lookups, beacons, and neighbor feedback. This metric is calculated by the following formula.

$$\text{Routing Overhead} = (\#\ \text{of Control Packets Sent}) + (\#\ \text{of Control Packets Forwarded} \quad (2)$$

Average Number of Hops - the average length of the route used by data packet transmission. In the case of Cluster-by-Cluster routing, this figure comprises messages created by a Cluster-by-Cluster routing node: publications and lookups.

Table 1: Simulation parameters

Network Parameters	
Node Density	100nodes/km^2
Number of Nodes	50, 100, 150, 200, 250
Simulati onTime	1200secs
Node VelocityV_{const}	1.4, 2.5, 5.0, 7.5, 10.0 (m/s)
Node Movement Model	Random Waypoint
Pause Time	0 sec
Transmission Range	250m
Bandwidth	2Mbps
Parameters of DSR	
dsragent_salvage_with_cache	true
dsragent_use_tap	true
dsragent_ring_zero_search	true
dsragent_enable_fl owstate	false
Parameters of Cluster-by-Cluster routi ng	
Number of Clusters	9
Beacon Transmission Period (unstable state)	2secs
Beacon Transmission Period (stable state)	30secs (1.4 – 7.5m/s), 20secs (10.0m/s)
BeaconTTL (*k*)	1
Maximum Number of Neighbor Feedback (*p*)	3

5.3 Node Mobility

Figure 17 shows the packet delivery rates that Cluster-by-Cluster routing and DSR obtain with varying node velocities (1.4, 2.5, 5.0, 7.5, and 10.0 m/s). A network size of 100 nodes was used. Cluster-by-Cluster routing obtains packet delivery rates of above 97% for all node velocities. On the other hand, the packet delivery rates of DSR drop quickly to about 42% as the node velocity increases. As a result, Cluster-by-Cluster routing's packet delivery rate versus node velocity is at most about 52% better than that of DSR with 10.0 m/s node velocity. Furthermore, as shown in Figure 18, Cluster-by-Cluster routing decreases the routing overhead about 92% more than that of DSR.

There are two reasons for DSR's markedly lower success rates.

(i) Emergence of congestion by the route discovery when a source node builds a long route to a destination node.

(ii) Drop of packet by node movement in a long route communication.

In this simulation, two DSR nodes send a message every second. If these nodes are far from each destination node, they create a long route by the route discovery. The long route is difficult to build in a high-mobility environment. If the source node fails to create the route, it retries the route discovery. This leads to extensive congestion. Once congestion occurs, many nodes cannot create their route to the destination. As a result, they discard their own messages, as their route discoveries are timed out. Even if the source node can create a long route, forwarding nodes may drop the message due to their movement. Thus, a long route communication in a high-mobility and -traffic environment is a threat to the communication reliability of the entire network. Figure 18 backs up this analysis. Figure 18 shows the routing overhead and the average number of hops versus node velocity. As the node velocity increases, the routing overhead of DSR increases drastically. This is because a large amount of congestion occurs due to the increase in long route communications, as shown in the average number of hops in Figure 18. From the increment of the overhead, it seems that congestion occurs in 5.0, 7.5, and 10.0 m/s.

In contrast, Cluster-by-Cluster routing uses only cached short routes that are likely up-to-date, as shown in Figure 18. Therefore, there is little overhead by the route discovery in a high-mobility network.

As shown in these results, Cluster-by-Cluster routing has the capacity to deal with node movement.

5.4 Network Size

Figure 19 shows the packet delivery rates that Cluster-by-Cluster routing and DSR obtain in networks of varying size (50, 100, 150, 200, and 250 nodes). A constant node velocity of 1.4 m/s was used. The packet delivery rates of DSR drop gradually to about 70% as the network size increases. On the other hand, Cluster-by-Cluster routing obtains packet delivery rates of above 92% for all network sizes. As a result, Cluster-by-Cluster routing's packet delivery rate versus network size is at most about 22% better than that of DSR in the networks with 250 nodes. Figure 20 shows the routing overhead and the average number of hops versus the network size. In the network of 250 nodes, Cluster-by-Cluster routing decreases the routing overhead about 95% more than that of DSR.

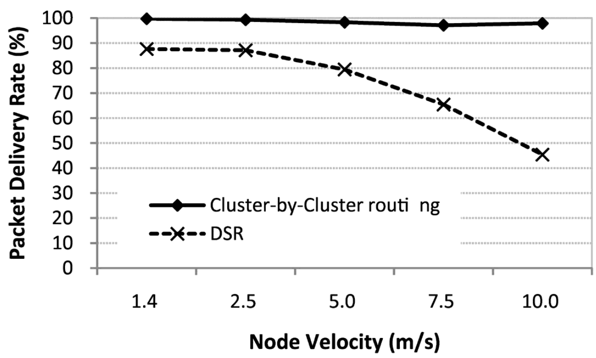


Figure 17: Packet delivery rates versus node velocity

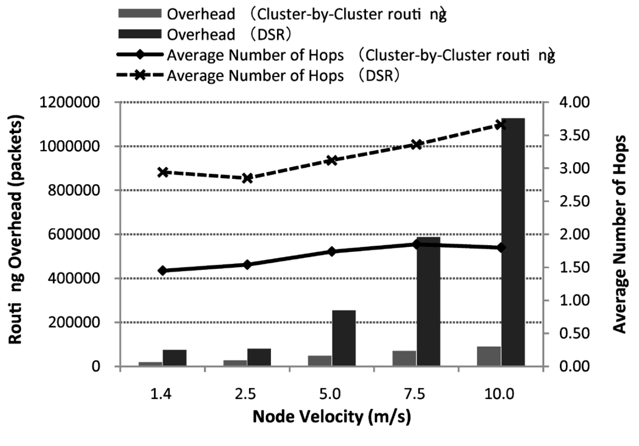


Figure 18: Routing overhead versus node velocity

The reason that packet delivery rates of DSR drop gradually with the increase in network size is the same as reason (i) in the previous subsection. In the large networks (especially 200 and 250 nodes), many DSR nodes relate to the route discovery process of one source node. Therefore, they flood the entire network with many RREQ messages. As a result, the congestion caused the deterioration of the packet delivery rate.

In contrast, Cluster-by-Cluster routing can control the emergence of long route communications. Therefore, there is also little overhead by the route discovery in the large networks. That is, Cluster-by-Cluster routing can achieve stable communications in large-scale networks because Cluster-by-Cluster routing does not use network resource.

As shown in these results, Cluster-by-Cluster routing has the capacity to deal with large-scale networks.

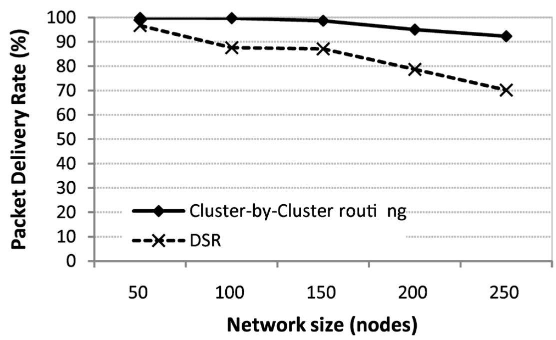


Figure 19: Routing overhead versus network size

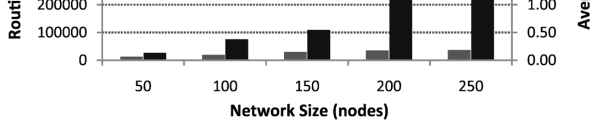


Figure 20: Routing overhead versus network size

6 CONCLUSION

We proposed a Cluster-by-Cluster routing method for controlling the emergence of long route communications due to clustering on the overlay network. Our simulation results show that our approach outperforms a popular reactive ad hoc routing protocol (DSR) in the scenarios considered – both in terms of packet delivery rates and routing overhead.

As future work, we plan to set the algorithm to take into account the movement of cluster heads in this Cluster-by-Cluster routing method. Furthermore, we will evaluate our method in varying combination of simulation and compare other popular ad hoc routing protocols such as AODV [7] and OLSR [8].

REFERENCES

[1] DSR (Dynamic Source Routing) : RFC4728 http://www.ietf.org/rfc/rfc4728.txt

[2] J.Y.Yu and P.H.J.Chong, “A survey of clustering schemes for mobile ad hoc networks, ” IEEE Communications Surveys & Tutorials, vol.7, no.1, pp.32–48, 2005.

[3] M. Jiang, J. Li, and Y.C. Tay, “Cluster based routing protocol (CBRP),” functional specification, IETF Internet Draft, MANET working group, *draft-ietf-manet-cbrp-spec-01.txt*, Aug. 1999.

[4] K.Lee and C.Hwang, “Hierarchical Configuration Scheme for Scalable Ad Hoc Networks,” In Proc. of ICCSA, LNCS 3981, pp.816-826, 2006.

[5] R.Winter, T.Zahn, and J.Schiler. “Random Landmarking in Mobile, Topology-Aware Peer-to-Peer Networks,” In Proc. of FTDCS, May 2004.

[6] The Netowork Simulator version 2 (ns-2), http://www.isi.edu/nsnam/ns/

[7] AODV (Ad hoc On-Demand Distance Vector) : RFC3561, http://www.ietf.org/rfc/rfc3561.txt

[8] T. Clausen, P. Jacquet Ed. "RFC 3626: Optimized Link State Routing Protocol," Internet Engineering Task Force (IETF) RFC, http://ietf.org/rfc/rfc3626.txt

[9] T.Ohta, S.Inoue, Y.Kakuda, and K.Ishida. “An Adaptive Mutihop Clustering Scheme for Ad Hoc Networks with High Mobility,” In Proc. Of IEICE, Vol.E86-A No.7, pp.1689-1697, June 2003.

[10] R.Oda, T.Ohta, and Y.Kakuda. “An Autonomous Clustering-Based Hierarchical Routing Using Decentralized Route Information Management Scheme for Mobile Ad Hoc Networks,” ICDCS Workshops 2008: pp.569-574, 2008.

[11] T.Zahn and J.Schiller. “DHT-based Unicast for Mobile Ad Hoc Networks”. In Proc. Of IEEE Percom. March 2006.

[12] T.Zahn and J.Schiller. “MADPastry: A DHT Substrate for Practicably Sized MANETs,” In Proc. Of ASWN, June 2005.

[13] G.He, S.Zhou, and S.Liu. “A K-hop Based Clustering Algorithm in Ad Hoc Networks,” In Proc. Of NAS 2008, pp.151-152, 2008.

[14] F.Garcia, J.Solano, and I.Stojmenovic, "Connectivity based k-hop clustering in wireless networks," *Telecommun. Syst.*, vol. 22, pp. 205-220, 2003.