Efficient Information Dissemination in Dynamic Networks

Zhiwei Yang
1. Dept. of Computer Science, Sun Yat-sen University
2. College of Computational Science, Zhongkai University of Agriculture and Engineering
Guangzhou 510006, China
yangzhw@mail2.sysu.edu.cn,

Weigang Wu, Yishun Chen, Jun Zhang
1. Dept. of Computer Science, Sun Yat-sen University
2. The Key Laboratory of Machine Intelligence and Sensor Networks, Ministry of Education
Guangzhou 510006, China
{wuweig, chenysh35, issai}@mail.sysu.edu.cn

***Abstract*—Dynamic network is the abstraction of networks with frequent topology changes arising from node mobility or other reasons. With the model of dynamic network, distributed computation problems can be formally studied with rigorous correctness. Information dissemination is one of such problems and it has received much attention recently. However, existing works focus on the time cost of dissemination, i.e. how fast the information can be disseminated to all nodes, and communication cost has been largely ignored. Our work focuses on high communication efficiency in information dissemination with correctness property guaranteed. We achieve this by making use of cluster-based hierarchy. Clustering has been widely studied and used in wireless networks to reduce communication cost. However, to the best of our knowledge, it has never been considered in the study of dynamic networks. In this paper, we firstly propose a dynamic network model, named (*T*, *L*)-HiNet, to extend existing dynamic network model with clusters. (*T*, *L*)-HiNet includes several properties defining the dynamics of cluster hierarchy in a dynamic network. Base on (*T*, *L*)-HiNet, we design hierarchical information dissemination algorithms for different scenarios of dynamics. The correctness of our algorithms is proved and the performance is analyzed. Compared with the algorithm recently proposed by Kuhn, Lynch and Oshman [7], our design can significantly reduce communication and our objective is fully achieved.**

Keywords—Information dissemination; dynamic network; cluster; distributed algorithm; system model.

I. INTRODUCTION

Due to node mobility and other reasons, the topology of a computer network, e.g. wireless ad hoc network or a peer-2-peer overlay network, may change frequently from time to time. Dynamic network is the general abstraction of such networks with formalized network models [1, 2]. Based on these models, various communication and computation problems can be addressed with formally proved correctness [3, 4].

Information dissemination is one of such problems, where the network nodes need to disseminate information throughout the network [5, 6]. It can be defined as the *k*-token dissemination problem [7, 8], where each node receives an initial set of tokens drawn from some domain, such that the total number of tokens in the input to all nodes is *k*. The goal is for each node to collect and output all *k* tokens.

Due to unceasing change and limited connection, information dissemination in dynamic network is a challenging task. With different dynamics models, quit a number of algorithms [9, 10, 5, 7, 8, 11] have been designed for this problem. Almost all these works focus on how to propagate information fast against topology changes, and another important issue, communication cost has been largely ignored. Communication cost in distributed computation is crucial in many instances of dynamic networks, e.g. mobile ad hoc networks and wireless sensor networks [12, 13].

This motivates us to pursue communication efficiency in information dissemination in dynamic networks. Basically, we introduce clusters into dynamic networks and make use of the cluster-based hierarchy to reduce transmissions conducted. Clustering has been widely studied and adopted in ad hoc networks to reduce communication cost and improve scalability in various networking and application problems [14, 15, 16, 17].

However, to the best of our knowledge, clustering has never been considered in the study of dynamic networks. Although clustering should be helpful in information dissemination, as the first attempt, making use of cluster hierarchy in dynamic networks is a challenging task. Firstly, hierarchy makes the network architecture more complex, and it will change with the network topology changes. Therefore, more dynamic factors need to be modeled and defined. Secondly, with hierarchy, the upper layer algorithms, like information dissemination algorithms, must operate differently at ordinary nodes and cluster head nodes, which are obviously more complex than those based on a flat architecture.

In this paper, we firstly propose a new dynamic network model, called (*T*, *L*)-HiNet, which extends existing models by including cluster hierarchy. The key point lies in the definition of different stability properties to model the dynamics of the

0190-3918/13 $26.00 © 2013 IEEE
DOI 10.1109/ICPP.2013.74

hierarchy, including set of cluster heads, set of members in a cluster, and connectivity among cluster heads. Based on (T, L)-HiNet, we design information dissemination algorithms for different scenarios of dynamics. All the algorithms can make clusters to reduce communication cost. Clusters can also help speed up the procedure of information dissemination if the hierarchy is stable enough.

We prove the correctness of our algorithms, in terms of information delivery, and analyze their performance in terms of both communication cost and time cost. The algorithm recently proposed by Kuhn and Lynch et al. [7] is chosen for comparison purpose. The results show that our algorithms can reduce communication cost significantly while keeping the time cost similar or even smaller in some cases.

The rest of the paper is organized as follows. Section 2 briefly reviews existing work on dynamic networks, including both network models and information dissemination algorithms. We describe our new dynamic network model (T, L)-HiNet in Section 3, including the basic system model describing the underlying environment, the CTVG graph model to represent the hierarchical network, and the stability properties to model dynamics of the hierarchy. In Section 4, we describe our hierarchical information dissemination algorithms and prove their correctness. Section 5 analyzes the performance of our algorithms and shows advantage against existing ones. Finally, Section 6 concludes the paper with future directions.

II. Related Work

Network model is the basis of dynamic network study. Dynamic network models are usually defined by extending graph models for general networks and the key point is how to reflect the dynamics of the network topology. Existing models define network topology changes from two different points of view, i.e. edge-centric models and graph-centric models. Edge-centric models focus on the changes of edges against time, i.e. edges may appear or disappear and consequently cause topology changes. Graph-centric models concern more about the overall properties in the graph level, such as connectivity, diameter.

Edge-centric models can be found in [1, 4, 18]. Kempe et al. [1] proposed the temporal network (G, λ), where λ is a time labeling of G specifying the time at which the edge e is available. Clementi et al. [18] have introduced the notion of Edge-Markovian Dynamic Graph (EMDG for short), which refers to stochastic edge time-dependency in evolving graphs. Ferreira et al. [4] is also an edge-centric model, which views a dynamic network as the evolution of topology graph, i.e. a sequence of static graphs. Avin et al. [19] considers that topology changes only occur once within every certain number of rounds.

Graph-centric models are proposed in [3, 7]. The notion of dynamic diameter is proposed by Kuhan et al. [3], which is a bound on the time required for each node to be causally influenced by each other node. The dynamic diameter extends the concept of normal diameter of a graph with consideration of dynamic changes. Kuhn and Lynch et al. [7] propose the T-interval connected model, which stipulates that for every T consecutive rounds, there exists a stable connected spanning subgraph.

Time-Varying Graph (TVG) [20] is a general dynamic network model, which integrates previous models with a unified framework. With TVG, a dynamic network is modeled as $G=(V, E, T, \rho, \zeta)$, where V and E are the vertex and edges in the topology graph respectively, T is the lifetime of the network, ρ indicates whether a given edge is available and ζ represents the time taken to cross an edge. It can switch between graph-centric and edge-centric perspectives on the dynamics.

All these models consider flat network architecture. To reflect cluster-based hierarchy in a dynamic network, new models are necessary.

Information dissemination was first studied in the static networks [9]. In a static graph network, k-token dissemination [11] can be completed in time $O(n+k)$ via flooding, where n is the total number of nodes. Gossiping is commonly used probabilistic approach to information dissemination in static environments [21, 22, 23, 24], where each node chooses one or a few nodes at random in each round to exchange information with them. Kempe et al. [22] use gossip-style local communication to provide simple and fault-tolerant protocols. Mosk-Aoyama et al. [21] consider a randomized gossip mechanism for information dissemination.

Information dissemination in dynamic networks is usually realized by flooding or similar approaches [2, 4, 10, 25, 26], which is necessary to theoretically guarantee the delivery of information to all nodes. The algorithm proposed by Clementi et al. [18] is based on the EMDG model with consideration of edge birth rate and death rate. Baumann et al. [10] proposed the k-active flooding protocol, where each node will forward information received for k times so as to speed up the propagation of information.

Dell et al. [5] studied information dissemination from the view of network connectivity. They show that one piece of information (i.e. one token) can be disseminated with guaranteed delivery to the whole network once the underlying network is connected at any time. This has been proved to be the weakest requirement for information dissemination in dynamic networks. Kuhn and Lynch et al. [7] studied k-token dissemination based on the T-interval connectivity model. Following this conclusion in [5], 1-token dissemination can be correctly completed via flooding in 1-interval connected networks in n-1 rounds. They design algorithms for general case of T-interval connected networks, and show that the computation can be sped up by factor T compared with 1-interal connected networks. Heaupler et al. [8] improved the work in [7] by making use of network coding to speed up the procedure of disseminating.

All the above information dissemination algorithms focus on how to guarantee the correctness of information delivery and communication cost is never considered. Since real deployed dynamic networks, like mobile ad hoc networks or wireless sensor networks, are resource constrained, how to achieve information dissemination with guaranteed correctness and low cost is obviously significant. This motivates us to design communication efficient algorithms by making use of cluster-based hierarchy.

III. (*T*, *L*)-HiNet: A Hierarchical Dynamic Network Model

(*T*, *L*)-HiNet is a new model of dynamic networks, which includes cluster hierarchy. In the following, we firstly describe the system model, which includes basic assumptions on the underlying network and clusters. The core part is the definition of (*T*, *L*)-HiNet, a model of the stability properties of cluster-based hierarchy. We also define CTVG, a graph model used in the definition of (*T*, *L*)-HiNet to represent the underlying network.

A. System Model

We consider an ad hoc network with a set of nodes, each of which has a unique identifier. The network nodes communicate with each other via multi-hop paths. The neighborhood among network nodes is determined by the communication range of the wireless transmission. Each node is equipped with the capability of probing neighbors. The topology of network may change from time to time, due to node mobility or other reasons.

A cluster-based hierarchy is constructed by clustering nodes into clusters. Since the network topology changes from time, the cluster-based hierarchy will change accordingly and re-clustering is possible. The clustering procedure can be carried out by clustering algorithms, which is out of the scope of this paper. We just assume the existence of such hierarchy, with following characteristics.

Each node belongs to at most one cluster at any given time but it may change its cluster from time to time. There is one and only one cluster head in each cluster and each cluster has a unique cluster identifier. For simplicity of presentation, the node ID of cluster head is used as the cluster ID in this paper. The members of a cluster are neighbors of the cluster head and obviously they know each other. Cluster heads may be connected via ordinary nodes along a path selected by the routing protocol or clustering algorithm. The ordinary nodes in the path are called cluster gateway nodes. Fig. 1 shows an example network with cluster-based hierarchy constructed.

B. Definition of CTVG

Now, we define the new dynamic network model CTVG, the extension of TVG with cluster-based hierarchy.

Definition 1. (Cluster-based Time-Varying Graph, CTVG): $G=(V, E, T, \rho, \zeta, C, I)$, where:

- V: the set of nodes that present in the network.
- $E \subseteq V \times V \times L$, the edge matrix represents the relationship between the nodes. L stands for some property associated with edge, such as the bandwidth of a link.
- Γ: lifetime of the system. Γ is divided into consecutive rounds, each round present one send/receive time. Then, the lifetime can be denoted by $\{t_0, t_1, .., t_i, ..\}$, where $[t_i, t_{i+1})$ is one round.
- ρ: $E \times \Gamma \rightarrow \{0,1\}$, the presence function indicating whether a given edge is available at a given time.
- $\zeta : E \times \Gamma \rightarrow \Gamma$, the latency function representing the time taken to cross an edge.

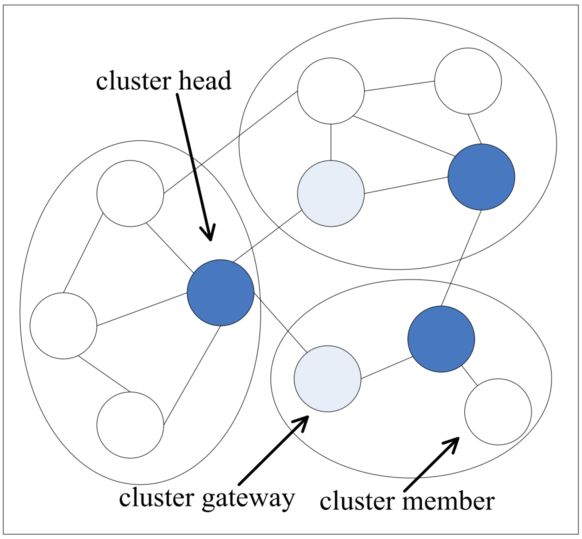

Fig. 1 An Example Network with Clusters

The above elements have been defined in the TVG model. To represent clusters, we introduce two new elements *C* and *I*. *C* is used to represents what status of the node in cluster and *I* is used to represent which cluster the node belongs to.

- C: $V \times \Gamma \rightarrow \{h, g, m\}$, the status of each node at a given time. The meaning of the status value: *h* indicates that the node is a cluster head; *g* indicates that the node is a cluster gateway node and responsible for forwarding packets between clusters; *m* indicates that this node is a common node, i.e. a cluster member.
- I: $V \times \Gamma \rightarrow N$, ID of the cluster that the node belongs to at a given time. N is the ID of the cluster.

With the definition above, a dynamic network with clusters can be clearly represented by CTVG. The dynamics of the network will change the network topology and also clusters from time to time. Therefore, we need to model the dynamics of the cluster-based hierarchy, i.e. how the hierarchy will change against time.

C. Definition of (*T*, *L*)-HiNet

We model the dynamics of cluster-based hierarchy in terms of how the hierarchy keeps stable. The following notions may be used in our dynamic models.

- V_h^i: is the set of cluster head nodes in the time $[t_i, t_{i+1})$, where $V_h^i \subseteq V$ and $\forall v \in V, C(v,i) = h \Leftrightarrow v \in V_h^i$.
- M_k^i: is the cluster member set of cluster k in time $[t_i, t_{i+1})$, where $M_k^i \in$ V and $\forall v \in V, I(v,i) = k \Leftrightarrow v \in M_k^i$.

To model the dynamics of cluster-based hierarchy, i.e. the changes of clusters, we define the following stability properties. Basically, we borrow the concept of T-interval from [7], to define the stability of clusters. T-interval denotes every T consecutive rounds. T can be any number, from 1 to ∞.

Definition 2. (T-interval Stable Cluster Head Set, T_s): We say the cluster head set is T-interval stable that:

$$\forall i, j \in [0, T-1) \Rightarrow V_h^i = V_h^j$$

This property requires that the set of cluster head keeps unchanged during the T-interval time. Obviously, the larger the T is, the more stable the cluster head set is. Especially, an ∞-interval stable cluster head set can be constructed using some special cluster head nodes as in [16].

Definition 3. (T-interval Stable Cluster, T_c): We say a cluster k is T-interval stable that:

$$\forall i, j \in [0, T-1) \Rightarrow M_k^i = M_k^j$$

Definition 4. (T-interval Stable Hierarchy, T_h): We say the cluster-based hierarchy is T-interval stable that the structure of hierarchy keeps unchanged in T-interval time that:

$$\forall k \in N, \forall i, j \in [0, T-1) \Rightarrow V_h^i = V_h^j \;\&\; M_k^i = M_k^j$$

Obviously, in a T-interval Stable Hierarchy, the cluster head set must be T-interval stable (Definition 2) and all the clusters are T-interval stable (Definition 3).

Definition 5. (T-interval Cluster Head Connectivity, T_d, and T-interval Cluster Head Subgraph, Υ): We say a dynamic network has T-interval cluster head connectivity if $\forall i \in N, \exists \Upsilon \subseteq G_i : V_\Upsilon \supseteq V_h^i$, Υ is connected, and $\forall j \in [i, i+T-1), \Upsilon \subseteq G_j$. Accordingly, the subgraph Υ is called T-interval Cluster Head Subgraph.

Definition 6. (L-hop Cluster Head Connectivity): We say that the connectivity among cluster heads is L-hop if:

$$\min\{L \mid \forall S \subset V_h^i, v \in V_h^i \setminus S, \text{ and } \exists u \in S, distance(u,v) \le L\}$$

L in fact indicates the maximum value of the shortest path between any two cluster heads that connected directly or by only gateway nodes. This is a key parameter that affects the procedure of information dissemination, as shown later in algorithm design and analysis.

L can be delicately controlled by clustering algorithms, as in WCDS-based clusters [12, 13]. Interestingly, in a 1-hop cluster-based network as assumed in our system, the value of L is not more than three.

Definition 7. (T-interval L-hop Cluster Head Connectivity): The cluster-based hierarchy is with both T-interval cluster head connectivity and L-hop cluster head connectivity in the subgraph Υ.

Definition 8. (T-interval L-hop Hierarchy Network, (T, L)-HiNet): a dynamic network with T-interval stable hierarchy (Definition 4), and T-interval L-hop cluster head connectivity (Definition 7) is called a T-interval L-hop Hierarchical Network, denoted by (T, L)-HiNet.

The seven definitions above describe the dynamics of a cluster-based hierarchy from different points. Fig. 2 shows the relationship among the definitions by a tree structure, where a higher level definition is the combination of its children ones in the lower level.

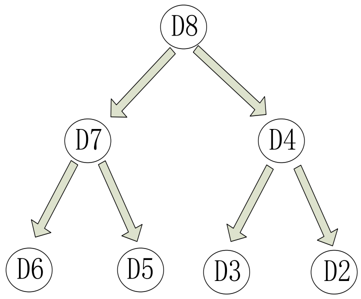

Fig.2 Relationship among definitions on dynamics of clusters

IV. Hierarchical Information Dissemination Algorithms

Based on the hierarchical dynamic network model proposed in the previous section, we design two hierarchical information dissemination algorithms. The first algorithm is for the general case of (T, L)-HiNet, and the second algorithm is for the worst case of (1, L)-HiNet. The correctness proof is also presented following each algorithm. We also consider unchanged cluster head set, a special case of (T, L)-HiNet and discuss how to achieve higher efficiency than that in general one.

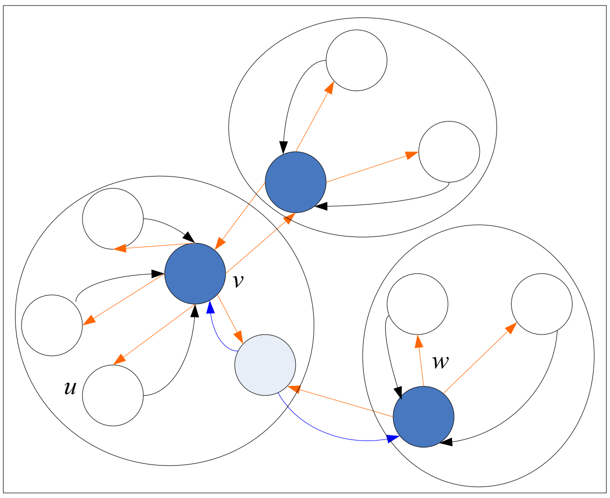

Fig. 3 An example illustration of Algorithm 1

With the example scenario shown in Fig. 3, we briefly describe the basic procedure to help the understanding of our algorithms later. When a node u wants to disseminate the token t, it will send t to its cluster head v. Then node v sends token t to all its neighbor nodes via broadcasting. When a gateway node receives token t, it will further propagate t by sending it towards other clusters. When another cluster head w receives token t, it will broadcast the token to all neighbors as v has done. To guarantee the token will be delivered to all nodes, each cluster head with token t will broadcast the token once in each T time interval. A cluster head can stop broadcasting t after a specific number of time intervals, which is necessary to guarantee that t will be delivered to all nodes. The number of time intervals needed will be discussed in the algorithm operations.

A. The Algorithm for (T, L)-HiNet

a) Description of algorithm operations

To disseminate k tokens, the procedure is divided into M phases, each consisting of T rounds. The value of M will be discussed later. According to the role of a node, i.e. member, cluster head, or gateway, different operations are executed. The pseudo code of the algorithm is shown in Fig. 4.

Algorithm 1. k-token dissemination in (T, L)-HiNet

```
For cluster member:
for i=0,…,M-1 do
  if clustherhead is changed then
    TS←∅
    TR←∅
  for r=0,...,T-1 do
    if TA ≠ TS ∪ TR then
      t←max(TA\(TS ∪ TR))
      send t to its cluster head
      TS←TS ∪ {t}
    receive t' from its cluster head
    TA←TA ∪ {t'}
    TR←TR ∪ {t'}
return TA

For cluster head and gateway:
for i=0,…,M-1 do
  for r=0,...,T-1 do
    if TS ≠ TA then
      t←min(TA\TS)
      broadcast t to neighbors
      TS←TS ∪ {t}
    receive t1,...,ts from neighbors
    TA←TA ∪ {t1,...,ts}
  TS←∅
return T A
```

Fig. 4 Pseudo code of k-token dissemination in (T, L)-HiNet

Each node needs to maintain three sets. TA is the set of tokens ever collected; TS is set of tokens broadcast by the node in the current phase; and TR is the set of tokens received from the current cluster head. Each token is stamped with a unique id, and the id is comparable with others.

Each member node u executes the algorithm with M phases. In the beginning of each phase, node u needs to check if its cluster head has been changed. If its cluster head is different from the one in previous phase, it needs to empty the two sets: TR and TS. Then u will execute T rounds of operations as follows.

In each round, node u will first check if any tokens collected have not been known by the current cluster head, i.e. TA ≠ TS ∪ TR. Then, it chooses t, the token with the maximum id among these unknown by cluster head and sends it to the current cluster head. Token t will be put into TS accordingly.

Node u also needs to receive tokens from its cluster head in each round. Node u updates its token sets TA and TR by adding the token received. Then, one round ends.

The operations at cluster head and gateway nodes are the same. Like cluster members, a cluster head/gateway node executes the algorithms with M phases, and each phase consists of T rounds. In the end of each phase, it needs to empty the set TS.

In each round, a cluster head/gateway needs to first check if any tokens have not been sent to its members/neighbors by broadcasting. It will choose token t with the minimum id that has not sent in current phase. The cluster head/gateway updates its token sets TA by adding the token received and its token sets TS by adding the token sent. Then, one round ends.

b) Correctness proof

We now prove the correctness of our hierarchical information dissemination algorithm by showing that each node will have all the k tokens after they finish executing the algorithm.

Lemma 1. *For any node v in a round r that dist(v,u)≤r≤T, either v has collected t or at least (r-dist(v,u)) different tokens.*

Proof. This lemma is borrowed from [7], and we omit its proof. □

Lemma 2. *With T-interval L-hop cluster head connectivity and T-interval stable hierarchy, for any token t known by node u at the beginning of any phase i, at least* $\lfloor (T-k)/L \rfloor$ *cluster head nodes will newly learn t in the end of the phase i.*

Proof. Let $N_i^d(u)$ denote the set of nodes at distance at most d from u in phase i. By Lemma 1, for any node $v \in N_i^{T-k}(u)$, either v has collected t or v has k different tokens in round T. Since there are totally k tokens, all nodes in $N_i^{T-k}(u)$ have known token t at the end of phase i. Because the clusters are L-hop, and the cluster-based hierarchy has T-interval L-hop cluster head connectivity, at least $\lfloor (T-k)/L \rfloor$ cluster head nodes are in the set $N_i^{T-k}(u)$. Therefore, at least $\lfloor (T-k)/L \rfloor$ cluster head nodes newly learn token in phase i, unless all cluster head nodes have got t. The lemma holds. □

Theorem1. *If T≥ k+α*L, each node will contain k tokens after executing M≥θ/α+1 phases in Algorithm 1, where θ is the upper bound number of nodes that can be cluster head, and α is a coefficient (it can be any positive integer).*

Proof. By Lemma 2, for any token t known by v at the beginning of phase i, at least α cluster head nodes newly learn token t in the end of phase i. Since there are at most θ cluster heads, after θ/α phases, all the cluster head nodes should have collected all the k tokens. Because the cluster-based hierarchy is T-interval stable (Definition 4), any token t known by cluster head u at the beginning of any phase i will be delivered to all nodes in the same cluster at the end of phase i. Then, after $\theta/\alpha+1$ phases, all nodes must have collected the k tokens. □

Remark 1. Algorithm 1 can be improved if cluster head set is ∞-interval stable, (Definition 2), i.e. keep unchanged during the algorithm execution. Modifications to Algorithm 1 are as follows.

The cluster member nodes need to send all tokens in its TA set out in the first phase. Later, even if a member node changes its cluster, it still does not need to send tokens collected. Obviously, the communication cost is reduced by reducing the sending at cluster members. Moreover, since $|V_h^i|$ should be less than θ in the cases of changing cluster head set, the algorithm can terminate earlier.

With the changes above, Algorithm 1 can complete dissemination of k tokens in $|V_h^i|/\alpha+1$ phases, with $(k+\alpha*L)$-interval L-hop cluster head connectivity (Definition 7)..

B. The Algorithm for (1, L)-HiNet

As shown in the correctness proof, the algorithm presented in the Section 4.1 can disseminate k tokens with guaranteed delivery only if $T \geq k+\alpha*L$. In this subsection, we consider environment with weaker stability in connectivity and design mechanism to disseminate k tokens correctly. The price to pay is larger communication cost caused by including previously known tokens in each packet.

Algorithm 2. k-token dissemination in (1, L)-HiNet

```
For cluster member:
i=0
sent TA to its cluster head
receive S1, S2, …,St from neighbors
TA←TA ∪ { S1} ∪ { S2}… ∪ { St }
for i=1,…,M-1 do
   if cluster head is changed then
      sent TA to its cluster head
   receive S1, S2, …,St from neighbors
   TA←TA ∪ { S1} ∪ { S2}… ∪ { St }
return TA

For cluster head and gateway:
for i=0,…,M-1 do
   broadcast TA to neighbors
   receive S1, S2, …,St from neighbors
   TA←TA ∪ { S1} ∪ { S2}… ∪ { St }
return TA
```

Fig. 5. Pseudo code of Algorithm in (1, L)-HiNet

Same as in Algorithm 1, the procedure of disseminating k tokens is divided into M rounds and the value of M will be discussed later. Each node needs to maintain TA, the set of tokens that it has collected. TA is updated in each round by adding tokens newly collected.

A cluster head/gateway needs to broadcast all tokens in TA in each round. A cluster member sends all tokens in TA to its cluster head in the beginning round. Later, it simply waits for tokens from its cluster head, and will not send out tokens unless its cluster head is changed. That is, if a member node changes its cluster head, it needs to sends all tokens in its TA to the new cluster head. Obviously, a member node sends tokens to a cluster head only once.

The correctness of the algorithm is proved as below.

Theorem 2*. With $M \geq n-1$, at the end of the algorithm execution each node contains the k tokens.*

Proof. For a token t collected by any cluster member node u, if its cluster head c does not have t, u must be newly affiliated with the current cluster head in the current round. Then, u will send all its tokens, including t, to cluster head c.

Since the network is 1-interval connected, for any token x, if x has not been known by all nodes, there will be at least one node that newly learns x in each round. Then, x will be known by all nodes in at most n-1 round. The theorem holds. □

Theorem 3*. If the network has (α*L)-interval cluster head connectivity (Definition 5), each node will contain k tokens after executing $M\geq\theta/\alpha+1$ rounds, where θ is the upper bound number of nodes that can be cluster head.*

Proof. The proof is similar to that of Theorem 1. We do not provide it here. □

Theorem 4*. If the network has L-interval stable hierarchy (Definition 4), each node will contain k tokens after executing $M\geq\theta*L+1$ rounds.*

Proof. For a token t collected by any cluster member node u, if its cluster head c does not have t, u must be newly affiliated with the current cluster head in the current round. Then, u will send all its tokens, including t, to cluster head c.

Since the network has L-interval stable hierarchy and L-hop cluster head connectivity, for any token x known by any cluster head node, if x has not been known by all cluster head nodes, there will be at least one cluster head node that newly learns x in per L rounds. Thus, for any token t , there will be at least one cluster head node that newly learns t in per L rounds, After $\theta*L$ rounds all cluster head nodes have all the k tokens. Then, after $\theta*L+1$ rounds, all the nodes must have collected all the k tokens. □

V. Performance Comparison and Analysis

In this section, we analyze the performance of the proposed algorithms in terms of time cost and communication cost, and compare them with the algorithms proposed in [7]. We choose [7] because it is also based on the dynamic model of T-interval. We consider the two scenarios of dynamics adopted in Section 4 respectively, i.e. (k+α*L, L)-HiNet and (1, L)-HiNet.

The number of nodes in the network is n_0. The upper bound number of nodes that can be cluster head is θ. The average number of cluster member nodes in one round is n_m. The average number of re-affiliations a cluster member conduct is n_r. In our analysis, time cost is represented by the number of rounds, and communication cost is represented by the total number of tokens sent. The time cost of each algorithm has been calculated in the correctness proof.

Table 1 Notations Used in Performance Analysis

Notations	Descriptions
n_0	The total number of nodes in the network.
θ	The upper bound number of nodes that can be cluster head.
n_m	The average number of cluster member nodes in one round.
n_r	The average number of re-affiliations a cluster member conducts.
k	The number of tokens to be disseminated.
α	A coefficient. It can be any positive integer.

The communication cost is calculated as follows. In the algorithm in [7] with (*k*+*α***L*)-interval connectivity, each node needs to broadcast in each phase. Since there are totally *k* tokens, each node broadcasts at most *k* times in one phase. Then, we can get the overall communication cost $\lceil n_0 / 2\alpha \rceil n_0 k$. In our algorithms, a cluster head/gateway node needs to broadcast in each phase, but a cluster member node broadcasts only when it re-affiliates to a new cluster head. Our communication cost is $(\lceil \theta / \alpha \rceil + 1)(n_0 - n_m)k + n_m n_r k$. The analysis under 1-interval connected model is similar.

Table 2 shows the results of both time cost and communication cost under different dynamics models. Since n_r should be much less than n_0, the communication cost of our algorithm is much less than that of [7]. This is obviously the benefit of hierarchical design.

Table 2 Performance of Different Algorithms

Network Models	Spending Time (rounds)	Communication Cost (total size of packets)
(k+α**L*)-interval connected [7]	$\lceil n_0 / (\alpha l) \rceil (k + al)$	$\lceil n_0 / 2\alpha \rceil n_0 k$
(k+α**L*, *L*)-HiNet	$(\lceil \theta / \alpha \rceil + 1)(k + al)$	$(\lceil \theta / \alpha \rceil + 1)(n_0 - n_m)k + n_m n_r k$
1-interval connected [7]	$n_0 - 1$	$(n_0 - 1) n_0 k$
(1,*L*)-HiNet	$n_0 - 1$	$(n_0 - 1)(n_0 - n_m)k + n_m n_r k$

To show the advantage of our design more clearly, we calculate information dissemination costs with an example network setup (in Table 3). The number of nodes in the network is 100. The upper bound number of nodes that can be cluster head is 30. The average number of cluster members in one round is 40. The average number of re-affiliations a cluster member conduct is set to be three and ten in (*T*, *L*)-HiNet and (1, *L*)-HiNet, respective. This is because in a (1, *L*)-HiNet, the dynamics is higher and consequently re-affiliations should occur more times. The number of tokens is set to be eight. The values of α and L are set to be five and two respectively.

As shown in Table 3, the communication cost of our algorithms is much less than that algorithms in [7]. At the same time, the time cost of our algorithms is the same as or even smaller than that of the algorithms in [7]. Although these are just results from one example setting, they still validate the advantage of our design to some extent.

Table 3 Numerical Results of Performance Analysis

Models of Dynamic Networks	Spending Time (rounds)	Communication Cost (total size of packets)
(k+α*L)-interval connected [7]	180	8000
(k+α*L, L)-HiNet	126	4320
1-interval connected [7]	99	79200
(1,*L*)-HiNet	99	51680

VI. Conclusion and Future Work

We study information dissemination in dynamic networks, with the consideration of cluster-based hierarchy to reduce communication cost. Our work includes model definition, algorithm design, correctness proof and performance analysis. Based on the notion of *T*-interval connectivity proposed by Kuhn, Lynch and Oshman [7], we define a new dynamic network model, named (*T*, *L*)-HiNet, which defines the stability of cluster-based hierarchy on top of dynamic topology. Based on (*T*, *L*)-HiNet, we design hierarchical information dissemination algorithms. With help of clusters, information from cluster members is disseminated along connections among cluster head, and consequently fewer packets need to be transmitted. Performance analysis shows that, compared with the algorithms designed in [7], our solution can disseminate information with much less communications cost while the time cost is similar or even smaller, and the benefit can be as much as 50%.

To the best of our knowledge, this is the first work that considers clusters in dynamic networks, so much more efforts are needed to extend and improve it. One interesting direction is to extend other flat dynamic network models, e.g. dynamic diameter, edge-Markovian dynamic graph, should also be extended with clusters. For the hierarchy itself, how to handle multi-hop clusters should be an interesting issue.

Acknowledgment

This research is partially supported by Guangdong Natural Science Foundation (No. S2012010010670), Pearl River Nova Program of Guangzhou (No. 2011J2200088.

References

[1] D. Kempe, J. Kleinberg, A. Kumar, "Connectivity and inference problems for temporal networks", STOC '00 Proceedings of the thirty-second annual ACM symposium on Theory of computing, 2000, pp.504-513.

[2] P. Grindrod , D.J. Higham,"Evolving graphs: dynamical models, inverse problems andpropagation", In Proceedings of the Royal Society of London Series A, 466(2115), 2010, pp.753–770.

[3] F. Kuhn, R. Oshman, "Dynamic Networks: Models and Algorithms", ACM Special Interest Group on Algorithms and Computation Theory News, 2011, pp.82–96.

[4] J. Leskovec, J. Kleinberg, C. Faloutsos, “Graph evolution: Densification and shrinking diameters”, ACM Transactions on Knowledge Discovery from Data, 2007, pp.1-41.

[5] R. O’Dell, R. Wattenhofer, “Information dissemination in highly dynamic graphs”, In Proceedings of 9th Joint Workshop on Foundations of Mobile Computing, 2005, pp.104–110.

[6] J. Hromkoviˇc, R. Klasing, B. Monien, and R. Peine, “Dissemination of information in interconnection networks (broadcasting & gossiping)”, Combinatorial Network Theory, 1996, pp.125–212.

[7] F. Kuhn, N. Lynch, R. Oshman, “Distributed computation in dynamic networks”, In Proceedings 42nd ACM Symp. on Theory of Computing, 2010, pp.513–522.

[8] B. Haeupler, D. Karger, “Faster Information Dissemination in Dynamic Networks via Network Coding”, In Proceedings of ACM Symp. on Principles of Distributed Computing, 2011, pp.381–390.

[9] S. Hedetniemi, S. Hedetniemi, A. Liestman, “A survey of gossiping and broadcasting in communication networks”, Networks, 18, 1988, pp.319–349,

[10] H. Baumann, P. Crescenzi, P. Fraigniaud, “Parsimonious flooding in dynamic graphs”,In Proceedings of 8th ACM Symp. on Principles of Distributed Computing, 2009, pp.260–269.

[11] D. M. Topkis. “Concurrent broadcast for information dissemination”. IEEE Transactions on Software Engineering, SE-11(10), 1985, pp.1107-1112.

[12] B. Han, W. Jia, “Clustering wireless ad hoc networks with weakly connected dominating set”, Journal on Parallel Distribted Computing, 67, 2007, pp.727–737.

[13] Y.Z. Chen, A.L. Liestman, “Maintaining weakly-connected dominating sets for clustering ad hoc networks”, Ad Hoc Networks, 3(5), 2005, pp.629–642.

[14] R. Friedman, A. Kogan, “Deterministic Dominating Set Construction in Networks with Bounded Degree”, Distributed Computing and Networking, 6522, 2011, pp. 65-76.

[15] N. Wang, J. Yu, Y. Qi, “Nodes neighborhood relation-based constructing algorithm for minimum connected dominating set”, Comput. Eng. 36 (13), 2010, pp.105–107.

[16] N.C. Seh, M. G. Qiang, A. D. O. Brain, “On the Properties of One-Dimensional Infrastructure-Based Wireless Multi-Hop Networks”, IEEE Transactions on Wireless Communications, 11 (7), 2012, pp. 2606-2615.

[17] J. Wu, M. Cardei, F. Dai, S. Yang, “Extended dominating set and its applications in ad hoc networks using cooperative communication”, IEEE Transactions Parallel Distrib. Syst. 17 (8), 2006, pp.851–864.

[18] A. Clementi, C. Macci, A. Monti, F. Pasquale, R. Silvestri, “Flooding time in edge-markovian dynamic graphs”, In Proceedings of 27th ACM Symp. on Principles of Distributed Computing, 2008, pp.213–222.

[19] C. Avin, M. Kouck´y, and Z. Lotker, “How to explore a fast-changing world (cover time of a simple random walk on evolving graphs)”, In Proceedings of 35th Coll. on Automata, Languages and Programming, 2008, pp.121–132..

[20] A. Casteigts, P. Flocchini, W. Quattrociocchi, N. Santoro, “Time-Varying Graphs and Dynamic Networks”, International Journal of Parallel, Emergent and Distributed Systems, 27, 2012, pp.387– 408.

[21] D. Mosk-Aoyama and D. Shah, “Computing separable functions via gossip”, In Proceedings of 25th ACM Symp. on Principles of Distributed Computing, 2006, pp.113–122.

[22] D. Kempe, A. Dobra, and J. Gehrke, “Gossip-based computation of aggregate information”, In Proceedings of 44th Symp. on Foundations of Computer Science, 2003, pp.482–491.

[23] B. Pittel, “On spreading a rumor”, SIAM J. Appl. Math. 47(1), Feb., 1987 pp. 213–223.

[24] M. Jelasity, R. Guerraoui, A. Kermarrec, M. Steem, “The peer sampling service: Experimental evaluation of unstructured gossip-based implementations”, Middleware '04 Proceedings of the 5th ACM/IFIP/ USENIX international conference on Middleware, 2004, pp.79–98.

[25] A. Clementi, A. Monti, and R. Silvestri, “Flooding over Manhattan.”, In Proceedings of 29th ACM Symp. on Principles of Distributed Computing’10.

[26] A. Pettarin, A. Pietracaprina, G. Pucci, and E. Upfal,”Tight Bounds on Information Dissemination in Sparse Mobile Networks”, In Proceedings of 30th ACM Symp. on Principles of Distributed Computing, 2011, pp.355–362.