SDC: An SCM-based Distributed Clustering Protocol

Yan Li[1], Li Lao[2], Jun-Hong Cui[1]
yan.li@uconn.edu, llao@cs.ucla.edu, jcui@engr.uconn.edu
[1] CSE, Univ. of Connecticut, [2] CS, Univ. of California, Los Angeles

1. INTRODUCTION

The procedure of partitioning a network topology into groups or clusters is usually referred to as graph clustering or network clustering. Network clustering has become an important technique widely used in networking research. For example, when we design a scalable routing protocol especially for less structured networks such as sensor networks and peer-to-peer networks, it is necessary to take node clustering features into account. Another example is network topology modelling. Since the Internet itself has a hierarchical structure, how to well characterize its clustering features is one critical step in topology modelling.

Network clustering can be done in both centralized and distributed ways. In our work, we are interested in the network clustering of large-scale distributed systems, such as peer-to-peer networks and sensor networks, where network and data management is totally decentralized and the global knowledge about the network is not available at individual nodes. Then the challenge is how to partition the network in an efficient and distributed manner, i.e., to design an effective distributed clustering protocol.

There are several characteristics of a good distributed clustering protocol. First of all, as a natural requirement of network clustering, nodes in the same clusters should be highly connected, and less connected between clusters. Secondly, the protocol should well control the cluster size, since big clusters are usually costly to maintain. Thirdly, the protocol should result in a minimum number of "orphan" nodes (i.e., nodes that form single-node clusters), because too many isolated nodes will violate the goal of clustering. Lastly, the protocol should take node dynamics into account, since the target networks (especially peer-to-peer networks) are highly dynamic with frequent entry and exit of nodes.

In the literature, MCL [2] is well accepted as an efficient and accurate network clustering algorithm. However, it works in a centralized fashion which can not be utilized in the network scenarios we are interested in. CDC [1], on the other hand, is a fully distributed algorithm. It forms clusters based on the node connectivity and effectively controls the cluster size. However, its performance heavily depends on the selection of cluster "originators" for which no good solution exists yet. Moreover, in the case of node exit, the whole algorithm has to be re-performed, resulting in high overhead.

With these problems in mind, we design a novel network clustering protocol called **SCM-based Distributed Clustering (SDC)**. SDC is a distributed protocol, where each node only needs to know its local information. The main idea of SDC is to dynamically adjust cluster formation based on **Scaled Coverage Measure (SCM)**, a practical clustering accuracy measure proposed by S. Dongen [2].

ICNP'05, Poster Session November 6–9, 2005, Boston, MA, USA.

Scaled Coverage Measure We assume the network topology in consideration $G = (V, E)$ is a connected, undirected graph. V is the set of nodes and E is the set of links, with $|V| = n$ and $|E| = m$. $\mathcal{C} = \{C_1, C_2, ..., C_l\}$ is a clustering of G. Given a node v_i, we have the following notations: **Nbr(**v_i**)** is the set of neighbors of v_i; **Clust**(v_i) is the set of nodes in the same cluster as v_i (excluding v_i); **FalsePos**$(v_i, \mathcal{C})$ is the set of nodes in the same cluster as v_i but not neighbors of v_i; **FalseNeg**$(v_i, \mathcal{C})$ is the set of neighbors of v_i but not in the same cluster as v_i. The *Scaled Coverage Measure*, $SCM(v_i, \mathcal{C})$, of node v_i is defined as:

$$1 - \frac{|\ FalsePos(v_i, \mathcal{C})\ | + |\ FalseNeg(v_i, \mathcal{C})\ |}{|\ Nbr(v_i) \cup Clust(v_i)\ |}. \quad (1)$$

The SCM of a graph G, $SCM(G)$, is defined as the average of the SCM values of all the nodes, i.e., $SCM(G) = (\sum_{v_i} SCM(v_i, \mathcal{C}))/n$, which lies in $[0, 1]$.

SCM can well reflect the clustering accuracy: the higher the SCM, the smaller the number of links between clusters and the higher the connectivity within clusters. Moreover, for graphs containing only isolated clusters/subgraphs that are themselves fully connected, their SCM is 1. Furthermore, the SCM of an orphan node is 0, which matches our goal of minimizing the number of orphan nodes. It should be noted that in some context, network clustering might be measured in different way, for example, only cluster size is a concern. Then designing a tailored network clustering algorithm for this scenario is beyond the scope of this work.

Based on the definition of SCM, the network clustering problem can be simplified as partitioning a network topology so as to maximize its SCM. Our proposed SDC protocol exactly follows this idea, adaptively forming clusters in an aggressive manner.

2. THE SDC PROTOCOL

The SDC protocol is performed in a distributed way. Each node only needs to maintain some basic information of its neighbors and the cluster it belongs to, such as the cluster id *clust_id* and the cluster size *clust_size*.

Given a network, each node is initialized as an orphan node with *clust_id* and *clust_size* (1 in this case). Then they start to exchange messages with their neighbors, conduct some simple computation, and form clusters in a greedy manner. After a number of rounds of communication, the clustering procedure becomes stable without further message exchange and the network is finally clustered.

In SDC, we define a set of **Clust_** type of messages. Suppose node v_i wants to be clustered with other nodes. The following key clustering messages may be involved.

◇ **Clust_Probe**. This message is sent by v_i to all of its neighbors to find out other clusters in its neighborhood. Each node which receives *Clust_Probe* will send its *clust_id* and *clust_size* back to v_i.

◇ **Clust_Request**. After v_i detects a cluster Cl, it issues a *Clust_Request* message, which is flooded in Cl and v_i's current cluster. This is a well-controlled flooding, since a node will forward this message only if it is in Cl or v_i's current cluster. For any node v_j in the cluster Cl, upon receiving *Clust_Reqest*, a very simple computation is performed to obtain $\Delta SCM(v_j)$, assuming node v_i joins Cl. For any node v_k in v_i's current cluster, it needs to compute $\Delta SCM(v_k)$ as if v_i leaves the cluster. To control the number of exchanged messages, a *TTL* is carried in *Clust_Request* messages. *TTL* is also used to control the cluster diameter.

◇ **Clust_Reply**. Upon receiving *Clust_Request* from v_i, node v_j sends back a *Clust_Reply* message carrying $\Delta SCM(v_j)$ and v_j's *clust_id* to v_i.

◇ **Clust_Reject**. Based on the *TTL* in *Clust_Request*, nodes in Cl can detect if the cluster diameter will exceed a predefined threshold due to the joining of node v_i. If this is the case, a *Clust_Reject* message will be sent back to v_i, and node v_i will not join Cl.

◇ **Clust_Update**. After node v_i receives the *Clust_Reply* from all the nodes in its own cluster and the neighbor cluster Cl (in the case that no *Clust_Reject* is received from Cl), it computes the overall gain ΔSCM based on the received information, assuming it leaves its original cluster and joins Cl. If $\Delta SCM > 0$, v_i should join Cl. Once v_i determines which cluster to join, a *Clust_Update* message containing v_i's node id and its original *clust_id* is flooded in its original cluster and the new cluster it will join. In this way, v_i and all the nodes receiving this message will update the *clust_size* and their own SCM.

After node v_i joins the new cluster, its neighbors in the original cluster are affected and should check whether they should join other clusters. The whole procedure will end if no node can join any cluster based on ΔSCM and the cluster diameter control.

A simple clustering example is shown in Fig. 1. In this example, *TTL* is set to 2. In Fig. 1.a, node 0 wants to be clustered with other nodes. After finding two neighbor clusters, A and B, it sends *Clust_Request*. At the same time, node 7 may also want to be clustered. Since node 0 is being processed, node 7 is "locked" (for which node 0 issues a Clust_Wait message) and it has to wait for a period of time. In clusters A and B, every node which receives *Clust_Request* computes the SCM gain and then sends *Clust_Reply* back to node 0 (Fig. 1.b). Node 0 then computes ΔSCM based on the received information and joins Cluster A (Fig. 1.c). Since node 4 is affected by node 0's joining action, it starts a new clustering procedure in a similar way as node 0 (Fig. 1.d).

Discussions. From the above explanation, we can see that SDC easily satisfies the four requirements of a good distributed network clustering algorithm: the choice of SCM and cluster size for clustering adaptation directly guarantees that SDC embraces the first three characteristics (high connection inside clusters, and less connection between clusters; well controlled cluster size; and minimized orphan nodes); member dynamics are also naturally handled since SDC itself is a dynamic procedure.

SDC does introduces some overhead when handling node dynamics. However, compared with CDC, this overhead is much smaller since only neighbors and/or those nodes in the same cluster are directly affected. In contrast, CDC has to re-do the complete clustering procedure for any node join or leave in order to maintain good clustering performance.

Preliminary Results. We conduct simulations to compare the performance of SDC with CDC for generated topologies and real Internet topologies with different network sizes. In our initial set of experiments, we demonstrate the clustering accuracy of SDC protocol. SCM is used as the accuracy measure. Fig. 2 shows the clustering accuracy of both protocols for generated random topologies. The SDC protocol outperforms CDC by about 50% for all network sizes. The improvement is even more significant for the real Internet topologies. For example, on a router-level real topology with 1620 nodes, the SDC protocol yields an accuracy value of 0.155 against 0.054 given by CDC.

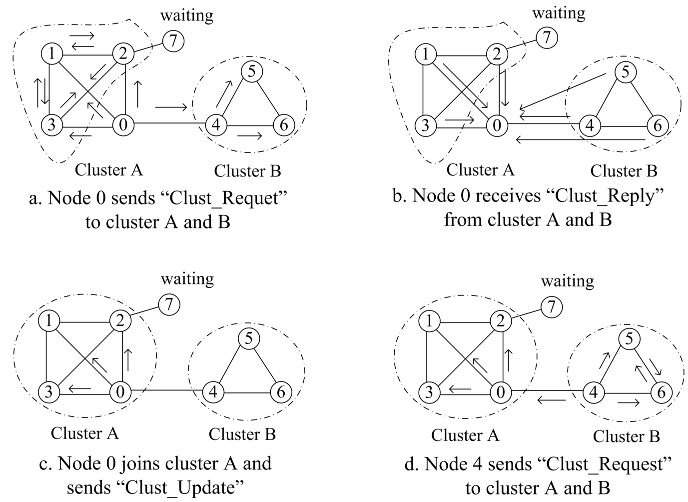

Figure 1: A simple example of SDC protocol ($TTL = 2$).

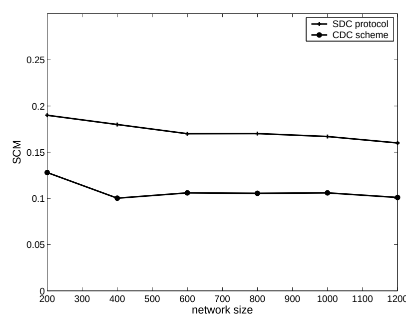

Figure 2: Clustering accuracy of SDC and CDC

3. SUMMARY AND ON-GOING WORK

In this paper, we have presented a distributed clustering protocol called SDC. It satisfies all the requirements of a good clustering algorithm: it considers node connectivity; it well-controls the cluster size; it minimizes the number of orphan nodes; and it can locally handle node dynamics with small overhead. Our preliminary results show its promising performance in clustering accuracy. More experiments are in progress to evaluate the proposed protocol in terms of time efficiency and message overhead for node dynamics.

4. REFERENCES

[1] L. Ramaswamy, B. Gedik, and L. Liu. A distributed approach to node clustering in decentralized peer-to-peer networks. *IEEE Transactions on Parallel and Distributed Systems*, 16(9), Sept. 2005.

[2] S. van Dongen. A new cluster algorithm for graphs. *Technical report INS-R9814, Centrum voor Wiskunde en Informatica (CWI), ISSN 1386-3681*, Dec. 1998.