Efficient and Reliable Link State Information Dissemination

Nirwan Ansari, *Senior Member, IEEE*, Gang Cheng, *Student Member, IEEE*, and Ram N. Krishnan

***Abstract*—Distributing link state information may place a heavy burden on the network resource. In this letter, based on the tree-based reliable topology (TRT), we propose a simple but efficient and reliable scheme for disseminating link state information. We show that the computational complexity of computing the subnet topology over which link state information is distributed is the same as that of computing the minimum spanning tree.**

***Index Terms*—Link state update, minimum spanning tree, protection, quality-of-service (QoS) routing.**

I. INTRODUCTION

FINDING paths that satisfy the performance requirements of applications according to link state information in a network is known as the quality-of-service (QoS) routing problem and has been studied extensively. However, distributing link state information may introduce a significant protocol overhead on the network resource. Many existing link-state routing protocols recommend that link state information is disseminated by simply flooding or flooding-like approaches [1], [2]. These kinds of approaches ensure that all routers within a link state domain converge on the same link state information within a finite period of time; they are also robust, i.e., in the case of link failures and node failures, link state information is still reachable to all nodes as long as the network is connected. On the other hand, because of the poor scalability of flooding, a large update interval has to be adopted in order to reduce the protocol overhead on the network resource. For instance, a link disseminates its state information every 30 min in Open Shortest Path First (OSPF). Consequently, because of the highly dynamic nature of link state parameters, the link state information known to a node is often outdated. Hence, the effectiveness of the QoS routing algorithms may be degraded significantly.

To overcome the problems of disseminating link state information by flooding or flooding-like approaches, many tree based link state dissemination schemes have been proposed. Moy [3] has proposed to distribute link state information over a subset of the network topology. Specifically, link state information is distributed over a spanning tree, instead of flooding over the whole network. In order to reduce the communication cost for maintaining the trees, a protocol [4], called Topology Broadcast based on reverse path forwarding (TBRPF), uses the concept of reverse path forwarding (RPF) to broadcast link-state updates in the reverse direction along the spanning tree formed by the minimum-hop paths from all nodes to the source of the update. As reviewed above, it can be observed that the protocol overhead of the tree based approaches is far less than that of flooding. However, they suffer the reliability problem, i.e., in the case of a single link failure, the tree (subnet) is splitted into two parts and link state information is not reachable to some nodes anymore, even though the network is still connected. Hence, existing proposals have one or more of the drawbacks of poor scalability, poor reliability, and slow convergence.

In this letter, we propose a new scheme for distributing link state information that possesses the advantages of fast convergence, reliability, and scalability.

II. PROPOSED SCHEME

In this section, a reliable, fast convergent, and scalable link state information dissemination scheme is proposed. We first define the reliable link state information dissemination below:

Definition 1: Given a network topology $G(N,E)$, where N is the set of nodes and E is the set of links, a link state information dissemination scheme is reliable if for any link e, as long as $\{e\}$ is not a minimum edge cut of $G(N,E)$, link state information is still reachable to all nodes in the case that e is broken.

By the above definition, it can be observed that the scheme of simply flooding link state information over the whole network is reliable, while distributing link state information over a spanning tree is not when the network is not a tree. However, flooding link state information over a network introduces heavy burden on the network resource and is rather inefficient. Hence, one of the key issues of designing a reliable and efficient link state information dissemination scheme is to find an appropriate subnet such that the scheme is reliable and the protocol overhead is minimized. In this letter, we introduce a new term, reliable topology (RT), over which the scheme of simply flooding link state information is reliable.

Definition 2: Given a network topology $G(N,E)$, a subnet $G(N,\widetilde{E})$ is a RT if for any link $e \in \widetilde{E}$, $\{e\}$ is not a minimum edge cut of $G(N,\widetilde{E})$, as long as $\{e\}$ is not a minimum edge cut of $G(N,E)$.

Note that a RT must exist for any network topology by Definition 2. It can also be observed that a link state information dissemination scheme is reliable if the minimum edge cut of the subnet topology over which it floods link state information has at least two members (edges). However, such topology (the RT without a one-link minimum edge cut) does not exist if there exists a one-member minimum edge cut of G. On the other hand, if the network does not have a one-link minimum edge cut, any of its RTs does not have a one-link minimum edge cut either, i.e.,

---

Manuscript received August 20, 2003. The associate editor coordinating the review of this letter and approving it for publication was Prof. J. Choe. This work has been supported in part by the New Jersey Commission on Higher Education via the NJI-TOWER project, and by the New Jersey Commission on Science and Technology via NJWINS.

The authors are with the Advanced Networking Laboratory, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07012 USA (e-mail: ansari@njit.edu).

Digital Object Identifier 10.1109/LCOMM.2004.827446

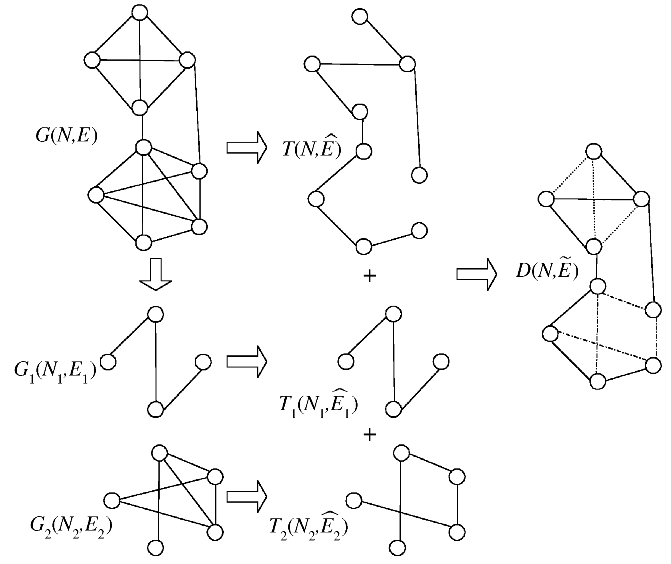

Fig. 1. An illustration of the construction procedure of a DST.

any link in a RT has at least one protection. Hence, the key point to provide reliable link state information dissemination is to find an appropriate subnet topology or a RT. An intuitive solution is the Hamiltonian Cycle [5], especially the least cost Hamiltonian Cycle. However, the major disadvantage of this approach is that finding a Hamiltonian Cycle in a given graph is NP-complete and would not be practical to use for fast convergence in real time networks. Therefore, alternatively, we provide an effective and efficient solution in this letter, TRT, which is proposed based on Theorem 1 (below) and built upon the combination of multiple spanning trees.

Definition 3—Tree-Based Reliable Topology (TRT): Given a connected network topology $G(N,E)$, and one of its spanning trees $T(N,\widehat{E})$, assume $G(N,E-\widehat{E})$ is the topology constructed by removing all the links in $\widehat{E}$ from $G(N,E)$, and consists of n $(n \geq 1)$ connected sub-networks, $G_1(N_1,E_1), G_2(N_2,E_2), \ldots, G_n(N_n,E_n)$. Further assume $T_1(N_1,\widehat{E}_1), T_2(N_2,\widehat{E}_2), \ldots, T_n(N_n,\widehat{E}_n)$ are the spanning trees of $G_1(N_1,E_1), G_2(N_2,E_2), \ldots, G_n(N_n,E_n)$, respectively. The topology $D(N,\widetilde{E})$ constructed by combining $T(N,\widehat{E}), T_1(N_1,\widehat{E}_1), T_2(N_2,\widehat{E}_2), \ldots,$ and $T_n(N_n,\widehat{E}_n)$ is referred to as a Tree-based Reliable Topology (TRT).

The construction procedure of TRT is illustrated by an example shown in Fig. 1. Given a network $G(N,E)$ as shown in Fig. 1, we can construct one of its TRTs, $D(N,\widetilde{E})$, by combining $T(N,\widehat{E}), T_1(N_1,\widehat{E}_1)$, and $T_2(N_2,\widehat{E}_2)$, where $T(N,\widehat{E})$ is one of the spanning trees of $G(N,E)$, and $T_1(N_1,\widehat{E}_1)$, and $T_2(N_2,\widehat{E}_2)$ are the spanning trees of $G_1(N_1,E_1)$ and $G_2(N_2,E_2)$, respectively, which are the remaining networks after removing the links in $T(N,\widehat{E})$ from $G(N,E)$. By Definition 3, it can be observed that if $n=1$, i.e., $G(N,E-\widehat{E})$ is still a connected network, TRT is actually constructed by combining the two spanning trees of $G(N,E)$.

Theorem 1: Any TRT, $D(N,\widetilde{E})$, is also an RT of the corresponding topology $G(N,E)$.

Proof: By contradiction. Assume $\exists e \in \widetilde{E}$ such that $\{e\}$ is a minimum edge cut of $D(N,\widetilde{E})$ while $\{e\}$ is not a minimum edge cut of $G(N,E)$. Since $\{e\}$ is a minimum edge cut of $D(N,\widetilde{E})$, further assume $D(N,\widetilde{E})$ is divided into two parts, $\widetilde{G}_1(\widetilde{N}_1,\widetilde{E}_1)$ and $\widetilde{G}_2(\widetilde{N}_2,\widetilde{E}_2)$, by removing e from $D(N,\widetilde{E})$. Moreover, since $\{e\}$ is not an edge cut of $G(N,E)$, assume a minimum edge cut of $G(N,E)$ that includes link e and divides nodes $\widetilde{N}_1$ from $\widetilde{N}_2$ is $\{e,e_1,e_2,\ldots,e_m\}$, as shown in Fig. 2. Note that $e_1,e_2,\ldots,e_m$ are not links of $D(N,\widetilde{E})$. Otherwise, $\{e\}$ cannot be a minimum edge cut of $D(N,\widetilde{E})$ because after removing $\{e\}$ from $D(N,\widetilde{E})$, $\widetilde{G}_1(\widetilde{N}_1,\widetilde{E}_1)$ and $\widetilde{G}_2(\widetilde{N}_2,\widetilde{E}_2)$ are both connected and there still exists at least a link connecting them, i.e., after removing link e from $D(N,\widetilde{E})$, it is still connected, thus contradicting the assumption. By the definition of TRT, assume $D(N,\widetilde{E})$ consists of $T(N,\widehat{E}), T_1(N_1,\widehat{E}_1), T_2(N_2,\widehat{E}_2), \ldots, T_n(N_n,\widehat{E}_n)$, where $T(N,\widehat{E})$ is a spanning tree of $G(N,E)$, and $T_1(N_1,\widehat{E}_1), T_2(N_2,\widehat{E}_2), \ldots, T_n(N_n,\widehat{E}_n)$ are spanning trees of $G_1(N_1,E_1), G_2(N_2,E_2), \ldots, G_n(N_n,E_n)$, respectively, which are the remaining connected networks by removing $\widehat{E}$ from $G(N,E)$. Assume e_1 is the link connecting node u and v. Since $e_1 \notin \widetilde{E}$ and $\widehat{E} \subset \widetilde{E}$, u and v are directly connected by e_1 after removing $T(N,\widehat{E})$ from $G(N,E)$, and thus belong to a single connected network, one of $G_k(N_k,E_k)$, $1 \leq k \leq n$. Hence, u is still reachable to v after removing $T(N,\widehat{E})$ from $D(N,\widetilde{E})$. Since $\{e\}$ is a minimum edge cut of $D(N,\widetilde{E})$ and $T(N,\widehat{E})$ is a spanning tree of $D(N,\widetilde{E})$, $e \in \widehat{E}$. Hence, by removing link e from $D(N,\widetilde{E})$, node u $(u \in \widetilde{N}_1)$ is still reachable to v $(v \in \widetilde{N}_2)$, which contradicts to the assumption that $\widetilde{G}_1(\widetilde{N}_1,\widetilde{E}_1)$ and $\widetilde{G}_2(\widetilde{N}_2,\widetilde{E}_2)$ are separated by removing link e from $D(N,\widetilde{E})$. Hence, any TRT is also a RT of the corresponding network. ■

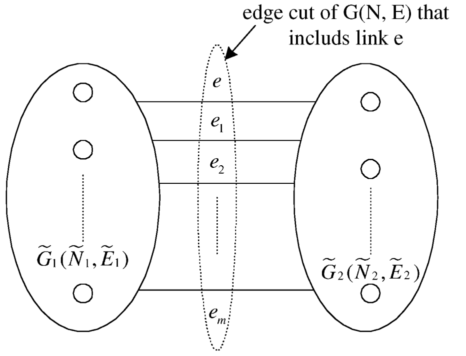

Fig. 2. A minimum edge cut of $G(N,E)$ that includes link e divides the network into two parts, $\widetilde{G}_1(\widetilde{N}_1,\widetilde{E}_1)$ and $\widetilde{G}_2(\widetilde{N}_2,\widetilde{E}_2)$.

By Theorem 1, we know that link state information can be reliably disseminated over a TRT, i.e., we can design a reliable link state information scheme by constructing a TRT over which link state information is distributed. Note that a TRT is the combination of several spanning trees. Hence, the remaining problem is to select appropriate spanning trees in order to minimize the protocol overhead and convergence time. In this letter, we simply deploy minimum spanning tree (MST) to construct TRT, i.e., in the process of constructing the TRT, all the spanning trees are the minimum spanning trees of the corresponding networks. As the result, we can guarantee that the convergence time of our proposed scheme is low. Furthermore, in order to evaluate the protocol overhead of link state information dissemination schemes, we define the number of link state advertisements (LSAs) upon each link state update as a performance

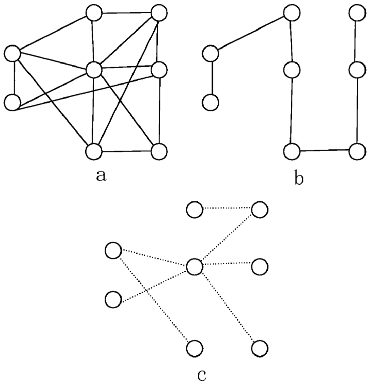

Fig. 3. A network and the spanning trees of its DST.

index. Note that a TRT consists of several spanning trees. Instead of treating a TRT as a single subnet but many independent trees with joint nodes, and LSAs are independently flooded over each of them, it can be proved that the number of LSAs is only twice that of the Moy's scheme [3], in which link state information is flooded over a spanning tree of the network. For instance, given a network (Fig. 3(a)) and one of its TRTs, which consists of two spanning trees (Fig. 3(b) and (c), respectively). In the case of a link state update, LSAs are flooded independently over two spanning trees (in this example, it can be viewed as the combination of two parallel tree-based link state dissemination schemes), instead of over the whole TRT. It can be observed that the number of LSAs is twice that of LSAs over a single spanning trees.

Computational Complexity: Note that by Prim's algorithm [6], the computational complexity of computing the minimum spanning tree in a network is $O(|E|\log|N|)$, where $|N|$ is the number of nodes and $|E|$ the number of links. From the definition of TRT, the computational complexity is the sum of the computational complexities of computing the spanning trees of $G(N,E)$, and $G_1(N_1,E_1), G_2(N_2,E_2),\ldots,G_n(N_n,E_n)$. Since $\sum_{i=1}^{n}|N_i| = |N|$ and $\sum_{i=1}^{n}|E_i| \leq |E|$, the computational complexity of computing the TRT that consists of minimum spanning trees is

$$\begin{aligned} &O(|E|\log|N|) + \sum_{i=1}^{n} O(|E_i|\log|N_i|) \\ &\leq O(|E|\log|N|) + \sum_{i=1}^{n} O(|E_i|\log|N|) \\ &= O(|E|\log|N|), \end{aligned} \tag{1}$$

where $|E_i|$ and $|N_i|$ are the numbers of links and nodes of $G_i(N_i,E_i)$, $i = 1,2,\ldots,n$, respectively. Hence, we can claim that the computational complexity of computing the TRT for our proposed link state update scheme is fairly low, and it can be deployed practically for real networks.

Note that although TRT is proposed for distributing link state information in this letter, it can be deployed for other purposes. For example, it can be used in WDM networks for providing link protection: in the case of any single link failure, TRT can guarantee an alternate route between the two corresponding nodes connected by the link as long as the network is still connected. Compared to the approach of using the Hamiltonian Cycle [7], the one using TRT has obviously the advantage of low computational complexity. Furthermore, since TRT is a fairly simple tree-based solution, it can be implemented in real networks by making only a slight modification to [3], while keeping the hardware intact.

III. CONCLUSIONS

In this letter, based on the TRT, we have proposed an efficient link state dissemination scheme. We show that our proposed scheme possesses the advantages of reliability, low protocol overhead, and fast convergence. By proving that the computational complexity of computing the TRT over which link state information is disseminated is compatible to that of computing the minimum spanning tree, we show that our proposed scheme is practical for a real network from the perspective of the computational complexity.

REFERENCES

[1] A. Zinin and M. Shand. Flooding optimizations in link-state routing protocols. presented at *IETF*. [Online] Available: draft-ietf-ospf-isis-flood-opt-01.txt
[2] J. Moy. OSPF version 2. presented at *IETF*. [Online] Available: RFC2328
[3] ——, Flooding over a subset topology. presented at *IETF*. [Online] Available: draft-ietf-ospf-subset-flood-00.txt
[4] B. G. Ogier, "A reliable, efficient topology broadcast protocol for dynamic networks," in *Proc. 1999 IEEE 18th Annu. Joint Conf. IEEE Computer and Communications Soc.*, vol. 1, 1999, pp. 178–186.
[5] K. Thulasiraman and M. N. S. Swamy, *Graphs: Theory and Algorithms*. New York: Wiley-Interscience, 1992.
[6] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge, MA: MIT Press, 1990.
[7] H. Huang and J. Copeland, "Hamiltonian cycle protection: A novel approach to mesh WDM optical network protection," in *Proc. 2001 IEEE Workshop on High Performance Switching and Routing*, May 2001, pp. 31–35.