A Meshed Tree Algorithm for Loop Avoidance in Switched Networks

Nirmala Shenoy
Networking Security and Systems Administration Department
Rochester Institute of Technology,
Rochester, NY, USA
nxsvks@rit.edu

***Abstract*—Loop free forwarding and routing is a continuing challenge in networks that have link and path redundancy. Solutions to overcome looping in bridged or switched networks are addressed by special protocols at layer 2, which block ports in the bridges to build a logical forwarding spanning tree. In this paper a meshed tree algorithm that aids in building and maintaining multiple overlapped tree branches from a single root node without blocking any ports is presented. Its potential use in bridged networks for loop avoidance is discussed. Some of its salient features are compared with spanning tree-based protocols and TRILL (Transparent Interconnection of Lots of Links) on RBridges (router bridges), another solution proposed for resolving loops in bridged networks.**

Keywords- Loop Avoidance; Switched Networks; Meshed Trees.

I. INTRODUCTION

Link redundancy is introduced in bridged (switched) networks to provide backup paths in the event of failure of an active link. This results in a physical network topology that has loops. The physical loops in turn cause broadcast storms when forwarding broadcast packets. Implementing a loop free logical topology over the physical topology is one way to avoid broadcast storms. The first of such logical loop free forwarding solutions called the Spanning Tree Algorithm (STA) was proposed by Radia Perlman [1]. Spanning tree in bridged networks was constructed by blocking some of the bridge ports. Based on STA the Spanning Tree Protocol (STP) was developed, and is an the specification for this protocol are available IEEE standard [1]. Rapid Spanning Tree protocol (RSTP) was then developed to overcome the high convergence times during topology changes in the basic STP. TRILL (transparent interconnection of lots of links) on Rbridges (router bridges) was subsequently proposed by the same researcher to overcome the disadvantages of STA based loop avoidance at the cost of some overhead and implementation complexity by adopting the IS-IS (intermediate system to intermediate system) routing protocol, where IS-IS related messages are encapsulated in special frames by the Rbridges. This is currently an *ietf* (Internet Engineering task Force) draft [4].

The premise of the above solutions is that a single logical tree from a root node that operationally eliminates physical loops is necessary to resolve the conflicting requirements of physical link redundancy and loop free forwarding. In the event of a link failure the tree has to be recomputed. While spanning tree is a single tree constructed from a single elected root node, with Dijkstra algorithm a tree is constructed at every node, assuming itself to be a root node, thus every node has a tree that it can use to forward. Dijkstra algorithm requires the connectivity information about all segments in the networks to compute the tree, while in the case of spanning tree, nodes join the tree branches based on the information they receive from their neighbors.

In this paper, we introduce a *meshed tree* (MT) algorithm that allows creation and maintenance of *multiple* overlapping tree branches from *one* root node. The multiple branches mesh at nodes, and in the event of failure of a link (or branch) the node can immediately fall back on another branch without the necessity for renewed tree resolution. This eliminates intermittent inconsistent topologies, which ensue during tree reconstruction.

Fig. 1 is provided to illustrate the difference between normal and meshed trees constructed over a given physical network topology. The circles are the nodes and the dotted lines are the physical links. Picture (a) shows a normal logical tree (thick line), which can be created either using the spanning tree or the Dijkstra algorithm. Picture (b) shows three tree branches (two originating from the root and the third from another node) that mesh at the nodes. The meshed tree branches thus formed have a single root node

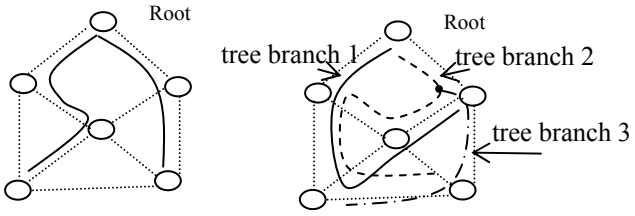

Figure 1 Single vs meshed trees

which is the principle of meshed trees. As each node in the network is on multiple tree branches, packets to the nodes can be forwarded using any branch.

MT Rationale: It is necessary to have a logical tree that spans all the nodes for forwarding broadcast packets. Let us call this the primary tree. But, that should not preclude the construction of multiple tree branches simultaneously or the overlapping of the tree branches, if achieved without loop formations. Tree branches other than those belonging to the primary tree will thus take over packet forwarding seamlessly in the event of link failure in the primary tree.

 ISBN: 978-1-61208-186-1

Meshed trees are implemented through a simple numbering scheme that will be used to assign virtual IDs (VID) to a node (in this case bridge) in the network. A VID in a compact manner defines a tree branch and hence a logical packet forwarding path from the root node to the node with the VID. A node can acquire several VIDs as it is allowed to join multiple tree branches. Meshed trees thus leverage the redundancy in meshed topologies to set up several loop free logical forwarding paths without blocking bridge ports.

Meshed tree creation and maintenance requires simple processing while enabling an easy transition from STA based protocols as described in this article. The goal in this paper is to provide the operational details of the MT algorithm and describe its features in comparison with existing loop resolution algorithms used in bridged networks. One optimization that is possible because of the unblocked bridge ports is also presented.

The rest of the paper is organized as follows. Section II discusses related work. In Section III, we present the operational details of the MT algorithm (MTA), as applicable to bridged networks. Section IV provides details on one optimization possible with MTA for forwarding unicast packets. Section V compares the performance and features of the MTA based solution against both STA based and the TRILL protocol. Section VI provides conclusions.

II. RELATED WORK

In this section, we focus on the two primary techniques adopted for loop resolution in bridged networks. The first of these is based on ST algorithm and the second is the TRILL on RBridges. The presentation in this section focuses on some distinct features of these techniques, without describing operational details as they are publicly available.

A. Protocols on Spanning Tree Algorithm

The STP is based on the STA. To avoid loops in the network while maintaining access to all the network segments, the bridges collectively elect a root bridge and then compute a spanning tree from the root bridge. In STP, each bridge first assumes that it is the root and announces its bridgeID. The information is used by the bridges to elect the root bridge. The bridgeID besides carrying the uniqueID of the bridge which is its MAC (medium access control) address also has a priority field to override the lowest MAC address bridge from being elected as the root bridge. Once a root bridge is elected, the other bridges then resolve their connection to the root bridge, by listening to messages from their neighbors to form a spanning tree.

STP has high convergence times subsequent to topology changes. To reduce the convergence times the *Rapid Spanning Tree* protocol (RSTP) was proposed [2]. RSTP is a refinement of the STP and therefore shares most of its basic operation characteristics, with some notable differences. The differences are; the detection of root bridge failure is done in 1 'hello' time; response to Bridge Protocol Data Units (BPDUs) sent from the direction of the root bridge; allowing RSTP bridges to 'propose' their spanning tree information on their designated ports; allowing the receiving RSTP bridge to determine if the root information is superior, and set all other ports to 'discarding' and send an 'agreement' to the first bridge; whereupon the first bridge, can rapidly transition that port to forwarding state bypassing the traditional listening/learning states, and thus allowing faster convergence; maintain backup details regarding the discarding status of ports to avoid timeouts if the current forwarding ports were to fail.

Advantages: STA based implementation is simple as the spanning tree is executed with the exchange of BPDUs at layer 2, where a BPDU carries the 'tree formation' information in multicast Ethernet frames.

Disadvantages: Several disadvantages of STA based protocols are noted in [2]. Traffic is concentrated on the spanning tree path, and all traffic follows that path even when other more direct paths are available, resulting in inefficient use of the link topology and reduction in aggregate bandwidth and causing traffic to take circuitous paths. Spanning tree is dependent on the way a set of bridges is interconnected. Small changes in this topology can cause large changes in the spanning tree. Changes in the spanning tree take time to propagate and converge especially for non-RSTP protocols. Though 802.1Q support for multiple spanning trees helped, it also required additional configuration. The number of trees is limited, and the defects apply within each tree regardless [3].

B. TRILL Protocolon RBridges

TRILL on Rbridges overcomes the shortcomings of the STP as it combines the functionality of layer 3 by using the IS-IS routing protocol [4] at layer 2 to compute pair-wise optimal paths between two Rbridges based on a link state algorithm. The computed pair-wise optimal paths will be used for forwarding the frames at layer 2. The solution is transparent to layer 3 protocols. IS-IS allows for the inclusion of information such as layer 2 addresses of reachable end nodes. Inconsistencies and loop formations during topology change are overcome by the 'hop count' used in TRILL frame headers for inter-bridge forwarding.

C. Operation of TRILL Protocolon

- *Election* of a Designated Rbridge (DR), which is the only bridge allowed to learn the membership of end nodes on that link, and to forward traffic destined to that link.
- The egress Rbridge from a link, usually the DR, *encapsulates* the frame with an additional header that contains, at the minimum, a hop count, and preferably also a destination Rbridge identifier. Frames in transit are distinguished from originating frames, since they contain the encapsulation header.
- Rbridges additionally calculate a spanning tree-based on the link state database used by IS-IS for purposes of delivering layer 2 multicast frames, and frames to unknown

 ISBN: 978-1-61208-186-1

destinations. Frames to be handled by the spanning tree use an encapsulation header with a destination 'Rbridge ID=0'.

- Use of End station address distribution (ESADI) protocol by the RBridge to distribute addresses of end nodes on its link to enable all Rbridges to know which Rbridge is the appropriate destination Rbridge for an end node.

Advantages over 802-style bridging [4]: Frames travel via an optimal path. As transit frames are routed, with a header that contains a hop count, temporary loops will not result in frame proliferation, and will quickly be discarded on the hop count reaching 0. Routing changes can be made instantaneously and safely based on local information

Loop Avoidance: An appointed forwarder for a link is responsible for loop avoidance [4]. It inhibits itself for a configurable time from 30 to zero seconds, which defaults to 30 second, after it sees a root bridge change on the link. An inhibited appointed forwarder for a port drops any native frames it receives and does not transmit any native frames in the LAN for which it is appointed. The forwarder will inhibit itself, as described above, if, within the past five Hello times, it has received a Hello in which the sender asserts that it is appointed as the forwarder. Optionally, they may not de-encapsulate a frame from ingress RBridge say 'RBm' unless it has RBm's Link State PDUs and the root bridge on the link it is about to forward onto is not listed in RBm's list of root bridges for that LAN. This is known as "de-encapsulation check" or "root bridge collision check".

III. THE MESHED TREE ALGORITHM

The 'meshed tree' algorithm allows construction of logically 'meshed trees' from a single root node in distributed fashion and with local information [12-14]. In the discussion presented in this article the election of a root bridge is not included as the focus is on the loop resolution / avoidance capability of MT algorithms. However a process similar to that adopted by STA can be used to elect a root bridge or a bridge can be designated to be a root bridge.

Bridge ID: For the operation of the MT algorithm bridgeIDs are necessary. These have to be unique only within the bridged network. Hence, a simple MAC address derivative can be used. This would be useful to keep the tree VIDs

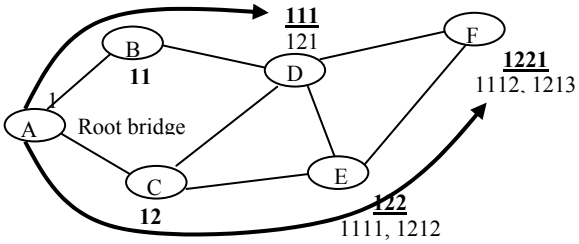

Figure 2: Meshed Tree Creation

simple, as the first value in the tree VID is the root bridgeID. One approach is to allocate a single digit ID to the root bridge once it is elected or designated. This however calls for a process of resolution if the root bridge fails which would be out of scope for this article.

Operation: The creation of a single meshed tree using VIDs is described first. In Fig. 2 a meshed tree originating at bridge 'A' is shown. Let bridge A have VID = 1. After being elected the root bridge, at regular intervals, bridge 'A' will announce its VID in a BPDU packet. Bridges B and C listen to the advertised VID and request to join as branches of bridge A. Bridge A allocates B a VID=11and C a VID=12 (by appending single digit value to its ID - the rationale for using a single digit is provided at the multiple digits can be used) and thus bridges B and C have now joined in tree originating from bridge A. Bridges B and C now advertise their VIDs.

Multiple VIDs from different parents: D hears advertisement from B and C and decides to join the branches from both B and C, while E hears only from C and decides to join the branch from C. D gets assigned a VID of 111 from B and 121 from C, while E gets assigned 122 by C.

Multiple VIDs under same parent: When E hears D announcing its VIDs, it can request a VID under each of D's

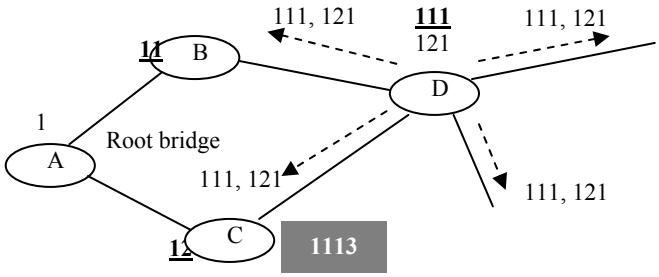

Figure 3 Loop Avoidance

VIDs and thus acquires VIDs 1111 and 1212.

To complete the explanation, F acquires VIDs 1221 from E and 1112 and 1213 from D. (*Note that D could also have acquired VIDs under E's other VIDs.)* Though the tree branches can mesh to the maximum limited only by the actual physical connections, they can however be controlled by limiting the number of VIDs that each bridge can acquire. In Fig. 2, only partial meshing is shown for clarity purposes. The 'meshed tree' thus formed by the VIDs provides a simple yet robust scheme to set up several redundant logical paths for packet forwarding without blocking ports [5-7].

Loop Avoidance in the algorithm is explained with Figure 3, which captures a partial topology from Figure 2. Assume C hears the VID 121 and 111 advertised by bridge D. It will not request to join the tree branch under the VID 121, as 'C' sees its VID sequence '12', in the advertised VID 121 and thus avoid loop formation.

Primary VID Tree: This is the tree that will be used for forwarding broadcast packets. Except for the root bridge, each of the other bridges will maintain one VID as the primary VID under the meshed trees and other VIDs as backup to be used in the failure of the primary VID. In

Figure 2, the VIDs that are underlined and in bold are the primary VIDs. The criteria to determine a primary VID may be predefined i.e. it could be based on link costs or on hops, as shown in the examples. The thick arrows identify the primary VID tree originating from bridge A.

Broadcast Packets: For forwarding broadcast packets or packets to unknown destinations the bridges should associate the VIDs to the ports through which they were acquired, so when using a VID, they are aware of the port on which the packet should be forwarded. This information is omitted in the figure for picture clarity. For simplicity and without loss of generality port 1 has been assumed to be the port from which the primary VID was heard by the bridges.

The rule for forwarding broadcasts packets by non-root bridges is: if received from the port of primary VID, then send out on all ports that have a VID derived from the primary VID. However, if the broadcast packet is received from any other port send out on port with the primary VID.

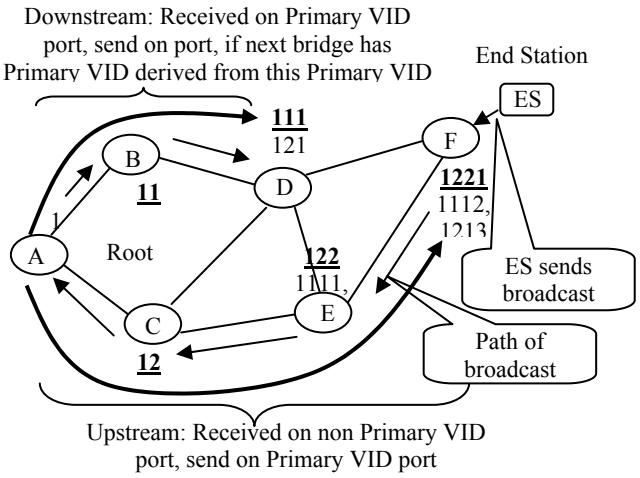

Figure 4: Broadcast Packet Forwarding

We will illustrate this with an example using Fig. 4. When bridge F receives a broadcast packet on one of its other ports let us say the one in which end station ES is connected it will send the packet out of port 1, its primary VID port. E receives the packet and will forward to its port 1, C will receive the packet and forward on its port 1 to A. This is the process that will be adopted when forwarding a broadcast packet upstream along the Primary VID tree.

The root bridge receives the packet on one of its ports, and will send it out on the other ports – in this case there is only one other port. The broadcast packet is then picked up by bridge B on its primary VID port. The broadcast packet now has to be forwarded downstream on the primary VID tree. B will send the packet out on all ports, if there are bridges on those ports that have a VID derived from its primary VID. To forward packets destined to end stations an optimized approach is discussed in Section IV. A general case is also explained in this section.

Link Failures: Let us assume that link CE failed as shown in Fig. 5. Bridge E will detect this and invalidate its VID 122, and fallback on VID 1111 as the primary VID. E may announce to bridges that have VIDs derived from 122 about the failure of VID 122 based on which bridge F invalidates VID 1221 to fallback on VID 1112 as the primary VID. The new primary VID tree is shown by the thick arrows. Frame forwarding will continue as usual except that E will use the path via D to forward to other bridges as none of the ports are blocked. On the revival of link CE the VID 122 may be restored.

The time for a bridge to identify a link failure is limited by the time it takes the protocol to recognize that the link is down. The bridge that recognizes this will immediately fallback to its backup VID and propagate the information only to those bridges that have a VID derived from the failed VID. The tree will thus get pruned if necessary but new tree reconstruction will not be necessary.

Other Features: Each single digit following the root bridge VID indicates a hop from the root bridge. End nodes connected to the bridge ports are not included in this count. Based on the root bridge VID other bridges will acquire their VIDs, which they then use for packet forwarding, thus the first digit in a VID carries the 'root' bridge VID.

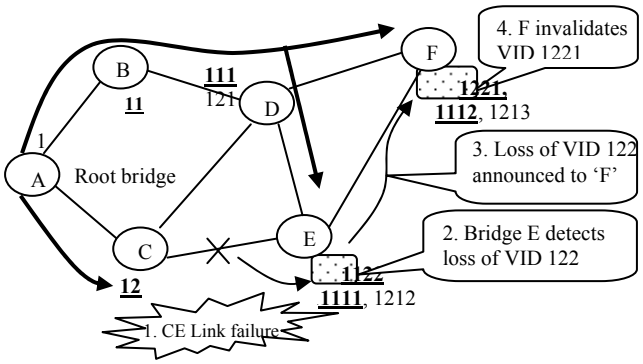

Figure 5: Primary VID Tree after Failure of Link CE

Construction of meshed trees requires bridges to advertise their VIDs at regular intervals. As the VID carries the branch information, bridges that hear the advertised VIDs can decide to join any or all of branches advertised. The joining decision can be based on criteria like shortest hops, best cost, and diversity in branch among others. Thus there is a local tree joining process for nodes, which makes the meshed trees to be constructed with local information.

IV. OPTIMIZED FORWARDING

Bridges can learn of hosts connected to them from source address in the frames they forward. Optimized forwarding of unicast packet with the MT algorithm is described in this section. In the MT approach as ports are not blocked, each bridge can advertise it's SAT (Source Address Table) of hosts directly connected, to neighboring bridges. Each receiving bridge can then populate their SAT with this information. This will require a bridge to use its

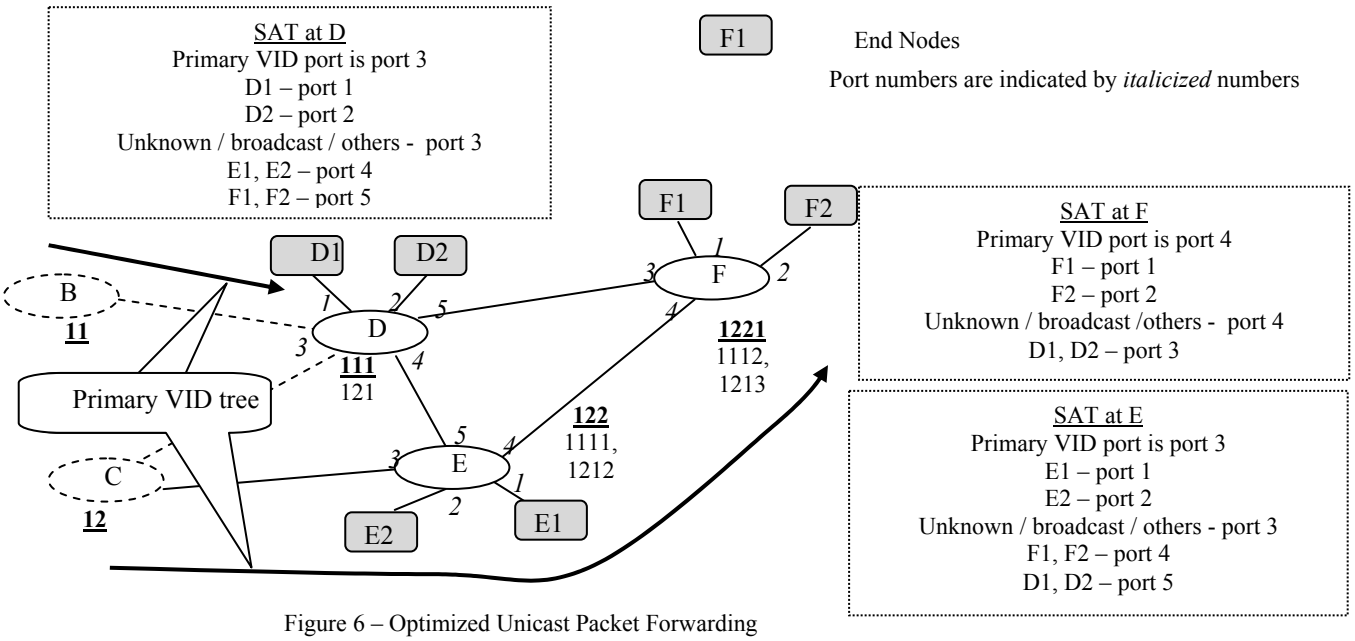

Figure 6 – Optimized Unicast Packet Forwarding

primary VID tree for broadcast frames or frames to unknown destinations. Frames destined to known MAC addresses can be forwarded on the most suitable port

In Fig. 6, we illustrate this with only a partial topology of the bridged network discussed thus far. Bridge F learns of end nodes F1 and F2 connected to it. Similarly bridge E learns of E1 and E2 and bridge D learns of D1 and D2. The local SAT information is advertised by the bridges on all their ports to neighboring bridges as these ports are not blocked. Bridge F has learned of hosts D1 and D2 which can be accessed on port 3. If it receives frames addressed to these hosts it will forward to port 3. However if it receives frames to hosts connected to bridges B and C the packets would be forwarded along the primary VID tree towards the root. Along the path if a bridge knows about the end nodes through its SAT, the packets would be directly forwarded to that port without going through the root bridge.

V. CONCLUSIONS

Loop free forwarding in networks with redundant paths have been addressed on the premise that a logical single tree topology originating from a root is essential. This resulted in the spanning tree algorithm, which faced high convergence delays later resolved by RSTP. Several disadvantages outlined in this article persisted, resulting in introduction of a more complex solution using the IS-IS on a protocol above layer 2. This article describes a simple solution that can replace STA algorithm at layer 2, without its disadvantages, but at the same time avoid the complex implementation requirements of TRILL on Rbridges. While the documents on STP and TRILL on Rbridges provide detailed specifications, it was possible only to highlight certain feature of meshed trees, to emphasize its loop free forwarding capabilities, without the complexity of the solutions being investigated to replace STP.

The current state of the work is as described above, where different implementation and optimization approaches for the meshed tree algorithm have been investigated. It is planned to model these details and compare for performance with Spanning tree and Rapid Spanning tree implementation in switched networks of varying topologies using Opnet simulation tool [7].

REFERENCES

[1] LAN/MAN Standards Committee of the IEEE Computer Society, ed. (1998). *ANSI/IEEE Std 802.1D, 1998 Edition, Part 3: Media Access Control (MAC) Bridges*. IEEE

[2] Wodjek W., "Rapid Spanning Tree Protocol: A new solution from old technology", http://www.compactpci-systems.com/articles, March 2003.

[3] Perlman R., Eastlake D., Dutt G. D. and Gai, A. G., "Rbridges: Base Protocol Specification", http://www.ietf.org/internet-drafts/draft-ietf-trill-rbridge-protocol-16.txt, March 3, 2010

[4] Touch J., Perlman R., "Transparent Interconnection of Lots of Links (TRILL): Problem and Applicability Statement", RFC 5556.

[5] Shenoy N., Pan Y., Narayan D., Ross D. and Lutzer C. (2005), "Route robustness of a multi-meshed tree routing scheme for internet MANETs", Proceeding of IEEE Globecom 2005. 28 Nov – 2nd Dec. 2005 St Louis, pp. 3346-3351.

[6] Book chapter on "Multi Hop routing and load balancing in Mobile Ad hoc networks", Encyclopedia on Ad Hoc and Ubiquitous Computing, Chapter editor Nirmala Shenoy and Sumita Mishra, Published by World Scientific Book Company, 2008.

[7] www.opnet.com

 ISBN: 978-1-61208-186-1