

Redundancy Avoiding Algorithm in Network

Sanjeev Gangwar, Santosh Kumar Yadav, and Ashok Kumar Yadav

Abstract—Computer connects to the internet with single path, does not more reliable because many problems occur such as cable cutting device fail etc. For finding more reliable, we need two or more paths between switches. Multiple path between two system is insure that more reliable. But occurs a problem that is redundancy problem and creates a cycle. According to switch concept data move between two systems continuously, for this activity switch work very slow or shutdown. We can remove this problem using a bridge protocol data unit concept. In which easily finds the cycle in a network or not. If a cycle between switch devices so need to remove the cycle. We use new spanning tree avoiding algorithms which break the cycle and improve the efficiency of the network.

Index Terms—Root bridge, root port, redundant connection between switches, BPDU.

I. INTRODUCTION

The computer network communication devices and media perform a big role to deliver the data from source to destination. Two computers use many types of communication device is work efficiently and properly.

Computer network connected with single device and single path. It may occur some problems such as device or path fails then network will be stopped or fail. So we have established two or more path from source to destination. If one path fails then use alternate path. But multiple paths make a cycle and create a problem such as shutdown the network. For removing this cycle we use redundancy avoiding algorithm.

A spanning tree is a subset of Graph G , which has all the vertices covered with minimum possible number of edges. Spanning tree does not have cycles and it cannot be disconnected. By this definition we insure that every connected & undirected Graph G has at least one spanning tree. A disconnected graph does not have any spanning tree, as it cannot span to all its vertices.

We found three spanning trees off one complete graph. A complete undirected graph can have maximum $n(n-1)$ number of spanning trees, where n is number of nodes.

Another way we use redundancy avoiding algorithm which is different from spanning tree avoiding algorithm, and take minimum time to avoid the redundancy in a network.

II. PROPERTIES OF SPANNING TREE

General properties of spanning tree, we now understand

Manuscript received March 26, 2018; revised June 8, 2018.

Sanjeev Gangwar, Santosh Kumar Yadav, and Ashok Kumar Yadav are with the Department of Computer Application, VBS Purvanchal University Jaunpur (U.P.), India (e-mail: gangwar.sanjeev@gmail.com, santosh.yadav08@gmail.com, ashok231988@gmail.com).

that one graph can have more than one spanning trees. The few properties of spanning tree are given below for a connected graph $G (V, E)$. A connected graph G can have more than one spanning tree. All possible spanning trees of graph G have same number of edges and vertices. Spanning tree does not have any cycle or loops, remove one edge from spanning tree will make the graph disconnected i.e. Spanning tree is minimally connected. Adding one edge to a spanning tree will create a circuit or loop i.e. spanning tree is maximally acyclic. Mathematical properties of spanning tree Spanning tree has $n-1$ edges, where n is number of nodes or vertices from a complete graph, by removing maximum $e-n+1$ edges, we can construct a spanning tree. A complete graph can have maximum $n(n-1)$ number of spanning trees. So we can conclude here that spanning trees are subset of a connected graph G .

III. APPLICATION OF SPANNING TREE

Spanning tree is basically used to find minimum paths to connect all nodes in a graph. Common application of spanning trees is computer network routing protocol, cluster analysis. Minimum spanning tree (MST) in a weighted graph, a minimum spanning tree is a spanning tree that has minimum weight that all other spanning trees of the same graph. For spanning tree we shall learn about two most important spanning tree algorithms, Kruskal's algorithm and Prim's algorithm both are greedy algorithms.

IV. NEED OF IDEAL NETWORK OF SPANNING TREE

Computer are connected with a single path between two switches, there is no problem, problem occur when the single path is break, all connection is break. Then we make another link between two switches. Reliability increase but connection was redundant that is a problem which is slow down the switches .then we need of ideal network of spanning tree [1].

V. IDEAL NETWORK OF SPANNING TREE

If you have 50, 100 switches then not need of spanning tree concept. If 100 or 200 switches are available then spanning tree concept is needed.

In ideal network design user has more option to access the server, here two paths is available for every user primary path and redundant path. If primary connection is fail then redundant path make a connection [2].

VI. SOLUTION FOR THE SPANNING TREE NETWORK

We use the redundancy concept which makes reliable

network. We use access layer approach, core layer approach and distributed layer approach as shown in Fig. 1.

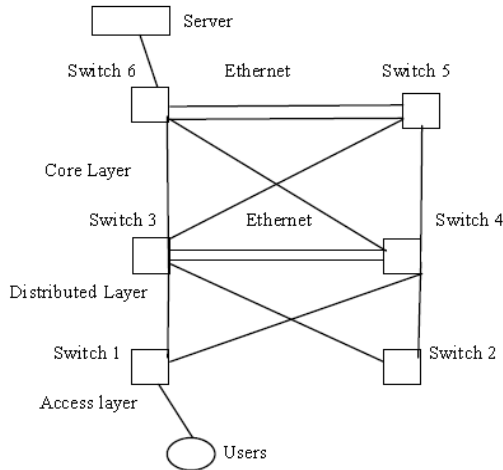


Fig. 1. Simple ideal network.

Layer Approach: By help of this layer approach we easily find that where a problem is occur.

Access Layer: In this layer users are access the ideal network.

Distributed Layer Approach: This is a middle layer approach, at this layer many switches are available. This layer is access to core layer here redundancy connection are available.

Core Layer: This is a top layer approach and server types.

VII. TYPES OF CONNECTIONS

Primary Connection: Direct connection of systems is called as primary connection.

Redundancy Connection: Optional connection or secondary connection is called as Redundancy connection.

Over Subscription: according to the switch subscription limited for users. If link is 100 mbps according to Cisco 1/20 means $100/20=5\text{mbps}$.if every link(48 port) send 100 mbps data then from one switch data used by 48gig that is problem for upper layer for this reason switch recommended 1:20 on access layer and 1:4 on distributed layer.

VIII. REDUNDANT TOPOLOGY

When we have small network then our work is happen on single path, but single path is not more reliable, if single path may cut or fail any problem in this condition reliability go down. We make one or more connection, so multi connection is called redundancy connection as shown in Fig. 2.

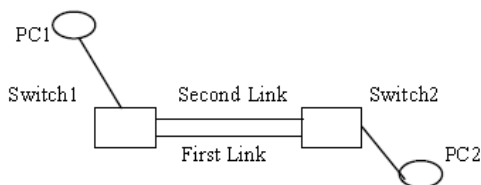


Fig. 2. Redundant connections.

Double connection between two are more switch, this type

of connection is called redundant, two or more connection is create a problem between two switches, that redundant problem solve by spanning tree algorithms [3].

IX. PROBLEM IN REDUNDANT PATH

In this mechanism PC1 send data to PC2 by switch1, forward the data to all port, left only PC1 and data come near the switch2, switch2 send the data to all does not know the about the PC2 that send the all joined node left only coming node PC2.for this switch2 again send the data to switch1 for this make a loop as shown in Fig. 3.

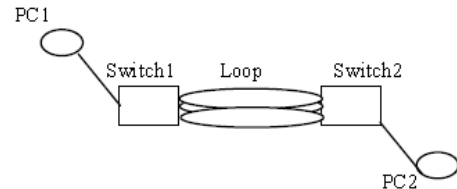


Fig. 3. Problem in redundant path.

X. AVOID REDUNDANCY

We close the path where is a redundant path, close the path between two or more switches, this technique is called spanning tree.

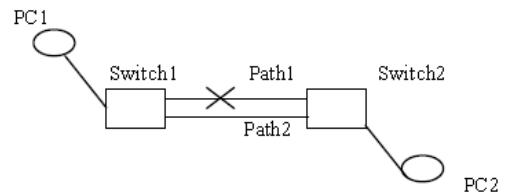


Fig. 4. Avoid redundancy.

In the Fig. 4 PC1 want to send data to PC2, we break the second path after then switch1 send data path1 on switch2 and switch2 broadcast to all nodes, here second path is break then no able to go switch1 then no make cycle [4], [5].

XI. REDUNDANT RING TOPOLOGY WITH ALL DUAL ATTACHED DEVICES

In highly critical applications, it may be necessary to dual attach all devices (outstations and PCs) to the network to provide a very high level of redundancy. This provides maximum reliability but also generates maximum cost and complexity.

XII. SPANNING TREE ALGORITHMS

- 1) Prevent loop (802.1D)
- 2) BPDU (Bridge Protocol Data Unit)
- 3) Election for Root Bridge
- 4) Block redundant link
- 5) Prevent loops (802.1D)

For high reliability we make a number of connections between two switches. For this reason switch is slow down. We use a spanning tree technique. In which we close all path only one path is available for communication.

In BPDU technique, if it is find the loop in the network. Then BPDU start from where we make a self signature at that node and move to another node and make a signature, and start to end. If BPDU find a self signature then BPDU is ensure that here is a loop in the network and announce that here is a loop between two switches or a network as shown in Fig.5.

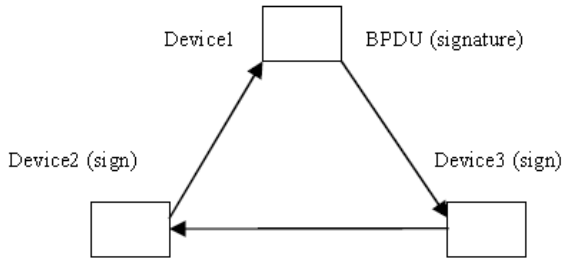


Fig. 5. Bridge protocol data unit.

XIII. ELECTION FOR ROOT BRIDGE

Root Bridge: In all switches which have small MAC address that switch is become a root bridge. Another if you want make root bridge on any of switch then we change the priority value on that switch a make address have three part first part is priority part, second is vender value part and third is incremented value part as shown in Fig. 6.

32768 1111.1111.1111.1111
 32768 2222.2222.2222.2222

Here bridge id is same MAC id is different so small address is first address so becoming root bridge. If you have new switch and want to make a root bridge then change the bridge id we do the smaller to all id's as look like.

32765 1111.1111.1111.1111

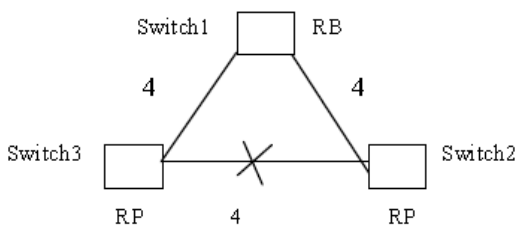


Fig. 6. Shortest path selection in network.

Root port: Spanning tree find the cost of the path value. If the D link id 100 mbps then the value is 4 switch3 go to switch 1 valve is 4 and switch3 to switch1 via switch2 the value is 8 then block the greatest valuable link, after that switch3 become the root port.

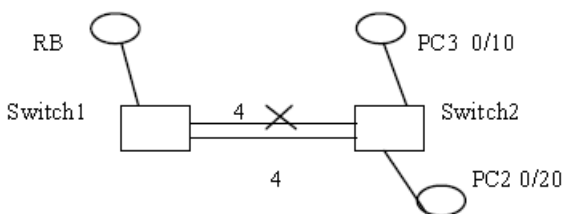


Fig. 7. Path selection in network where link Length is same.

In the Fig. 7 if both path values are same and MAC address is same then in this condition decides that which is port forward or which port is blocked by rule small port number is always forward and bigger port number is block.

XIV. SPANNING TREE AVOIDING ALGORITHMS

First time algorithms send a BPDU packet on the network take 20 second, after go to listing process take 15 second after that process think that which node is forwarding or block one side on forward node take 15 second, in the total forwarding process take 50 second as shown in Fig. 8.

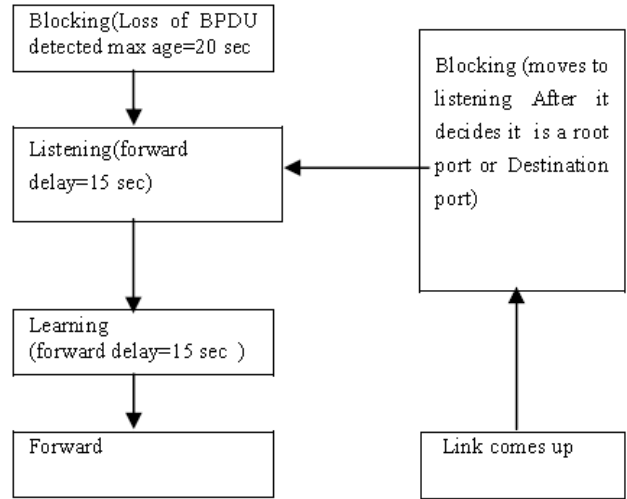


Fig. 8. Spanning tree avoiding algorithms.

In starting we plug the wire in switch every node in block state for this node status take 50 second we have discussed already .suppose that switch1 is a starting node and send BPDU to on a node and take 20 second there is no response because is a computer not a switch, after go to listening state take 15 second, and go to next state that is learning state take 15 second .After completed all state then go to forward state otherwise switch1 forward the BPDU to switch2 and switch2 forward the BPDU to switch1 means loop in between switches so we apply spanning tree algorithms [6], [7].

XV. CONCLUSION

All network are connected by multipath definitely multipath is more reliable. But in multipath some it slowdown or stop the network device. We can remove this problem by using spanning tree avoiding algorithm. This algorithms work very efficiently and remove the redundant path in a network, it remove the overhead from the switches and provide the smallest valuable path from source to destination in network.

REFERENCES

[1] Y. Bendavid, S. F. Wamba, and L. A. Lefebvre, "Proof of concept of an RFID-enabled supply chain in a B2B ecommerce environment," in *Proc. the 8th International Conference on Electronic Commerce (ICEC'06)*, 2006, pp. 564-568.
 [2] C. H. Hsu, Y. M. Chen, and C. T. Yang, "A layered optimization approach for redundant reader elimination in wireless RFID networks," in *Proc. IEEE Asia-Pacific Services Computing Conference*, 2007, pp. 138-145.

- [3] D. Y. Kim, B. J. Jang, H. G. Yoon, J. S. Park, and J. G. Yook, "Effects of reader interference on the RFID interrogation range," in *Proc. the 37th European Microwave Conference*, 2007, pp. 728-731.
- [4] B. Carburnar, M. K. Ramanathan, M. Koyuturk, C. Hoffmann, and A. Grama, "Redundant-reader elimination in RFID Systems," in *Proc. IEEE Secon Second IEEE Communications Society Conference on Sensor & Ad Hoc Communications & Networks*, 2005, pp. 176-184.
- [5] S. E. Sarma, "Towards the five-cent tag," Technical Report MIT-AUTOID-WH-006, MIT AUTO ID Center, 2001.
- [6] M. Jo and C. G. Lim, "RFID tag detection on a water content using a back-propagation learning machine," *KSII Transactions on Internet and Information Systems*, vol. 1, no. 1, pp. 19-32, 2007.
- [7] D. W. Engels, "The reader collision problem", White Paper MIT-AUTOID-WH-007, MIT AUTO ID Center, 2001.



Sanjeev Gangwar is an assistant professor in the Department of Computer Application, VBS Purvanchal University Jaunpur (U.P.). He obtained his MCA degree from MJP Rohilkhand University Bareilly and M.Phil in computer science. He has more than 10 years teaching experience in different organization. He is member of different reputed journals, such as International Association of

Engineers (IANEG) (Id: 116368). He has got published ten papers in reputed national/international journals. His research interests are in the field of mobile Ad hoc networks.



Santosh Kumar Yadav is a lecturer in the Department of Information Technology, VBS Purvanchal University Jaunpur (U.P.). He has a master of technology in computer science and engineering from Uttar Pradesh Technical University Lucknow. He has more than eight years teaching experience in different organizations. He has published two papers in national journals. His current research interests are computer network,

network security and mobile Ad hoc networks.



Ashok Kumar Yadav is a lecturer in the Department of Information Technology, VBS Purvanchal University Jaunpur (U.P.). He has a master of technology in computer science and engineering from Uttar Pradesh Technical University Lucknow. He has got international certification in Oracle 9i (OCA). He has more than four years' teaching experience in different

organizations. He has published two papers in international journals. His current research interests are computer network, mobile Ad hoc networks and parallel algorithm.