Designing & Challenges With Best Performing Path Implementation

1Padmashree Kunthe* and 2R.C. Soni

1Research Scholar, Electronics & Communication Engineering Department,

CMJ University, Shillong, Meghalaya, INDIA.

2Director/Principal, Administrator Dept., IAMR, Ghaziabad, Uttar Pradesh, INDIA.

*Corresponding Author : gocetinstitute01@gmail.com

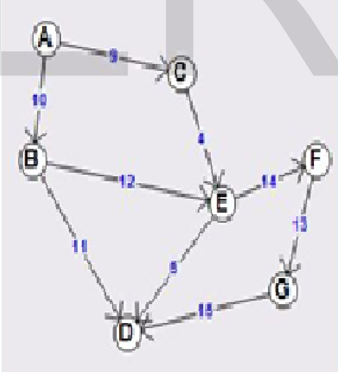
ABSTRACT:- The performance and reliability of the Internet depend, in large part, on the operation of the underlying routing protocols. Today's IP routing protocols computer paths based on the network topology and configuration parameters, without regard to the current traffic load on the routers and links. The responsibility for adapting the paths to the prevailing traffic falls to the network operators and management systems. This chapter discusses the modeling and computational challenges of optimizing the tunable parameters, starting with conventional intradomain routing protocols that compute shortest paths as the sum of configurable link weights. Then, we consider the problem of optimizing the interdomain routing policies that control the flow of traffic from one network to another. Optimization based on local search has proven quite effective in grappling with the complexity of the routing protocols and the diversity of the performance objectives, and tools based on local search are in wide use in today's large IP networks.

INTRODUCTION:-

This paper represents a generalization of trees called *networks*. In a tree, every node must have exactly one parent node (except for the tree's root node, which has no parent).

In a network, any node may be connected by links (also called edges) to any number of other nodes. Depending on the problem, links may be *directed* (pointing in one direction) or *undirected* (you can move across a link in either direction) but there's no notion of a parent node and no root.

In most representations, a link also has a *cost* (sometimes called *weight*, *distance*, or *length*) that gives the cost for traveling across the link. For example, in a street network a link's cost might be the time it takes you to drive across it. In a power network, a link's cost might be the electrical loss when current travels across the link. In a highway network, a link's cost might be the expense of building the corresponding stretch of highway (often more than two million dollars per mile, per lane).



The Figure shows small directed network.
Where Each link has given cost.

Networks can represent many real-world objects, such as the streets in a city, water pipes, power lines, computer networks, storm drains, sewer lines, railroad lines, airline connections, and so forth. In such networks, finding the shortest paths can give you the best route to drive from one point to another, the route through a computer network that uses the fewest resources on intermediate computers, the airline flight with the fewest connections, and so forth.

There are also several more tenuous applications that use shortest-path calculations. If you set up a network properly, shortest paths can represent

IJSER

the best sequence of moves to solve problems such as Rubik's Cube or the smallest number of transformations required to convert one piece of text into another (which gives a measure of how similar the pieces are). Shortest-path calculations are also required to perform calculations such as solving the traveling salesperson problem (finding the most efficient order for visiting a series of points and returning to the starting point).

Finding the shortest path from one node to another in a network actually involves more than finding a single path. When you look at a street map, you may be able to intuitively guess a near-optimal path without looking at all the dead ends, cul-de-sacs, side streets, and other clutter that obviously won't play a part in the final path.

Unfortunately, shortest-path algorithms cannot make similar leaps of intuition. There are a couple of good reasons for this. First, in some networks it may be very hard to tell which links are obviously not useful.

IJSER

Second—and perhaps more important: These algorithms just aren't that smart. You could add code to try to identify parts of the network that obviously won't be part of the final solution but that would complicate the algorithm and might actually slow overall performance. These algorithms contain very tight loops that are executed many, many, MANY times very quickly. Adding special tests to help find the shortest paths makes the algorithm more intelligent, but usually slows the loops down and gives you a net loss.

Best Path Design Goals :-

1) Receiver driven multihoming : receiver chooses the network interface(s).
2) Source driven multihoming: source chooses interface(s) before address binding.

3) Network driven multihoming: Network makes a decision dynamically to select the network interface(s).

4) Transport protocol to support multipath with multihoming with improved performance.

IDEAS & CHALLENGES:-

Ideas:

1) Receiver interface manager scans link qualities and selects the network interface(s) as per certain policies.

2) The receiver publishes all the available network interface(s) and sender chooses among those before address binding.

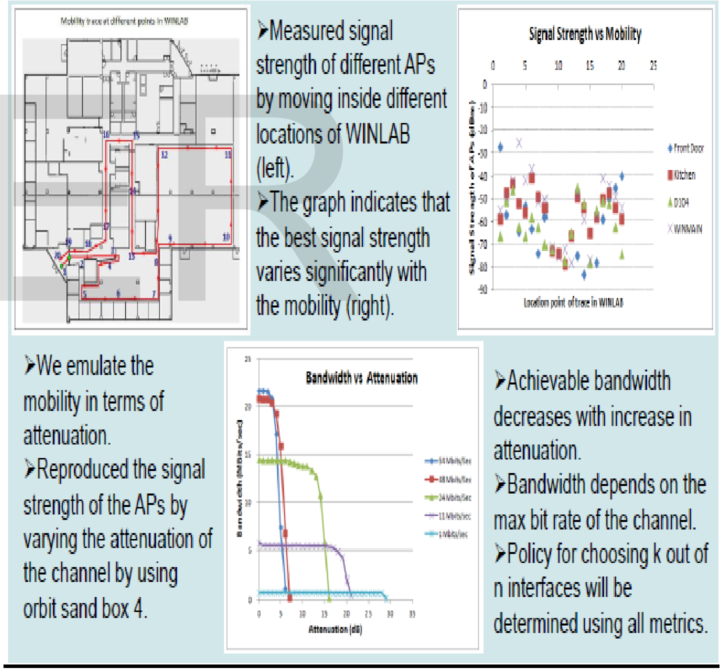
3) The router has more information about the path quality; it can choose the interface(s) dynamically.

Challenges:

1) Efficient policies are required to get a performance enhancement.

2) Additional cost will be incurred at the protocol stack for analyzing the link quality metrics.

EMULATION OF LINK QUALITY VARIATION:-



➢Measured signal strength of different APs by moving inside different locations of WINLAB (left).

➢The graph indicates that the best signal strength varies significantly with the mobility (right).

➢We emulate the mobility in terms of attenuation.

➢Reproduced the signal strength of the APs by varying the attenuation of the channel by using orbit sand box 4.

➢Achievable bandwidth decreases with increase in attenuation.

➢Bandwidth depends on the max bit rate of the channel.

➢Policy for choosing k out of n interfaces will be determined using all metrics.

CONCLUSION:-

1) The best performance policy with higher mobility (attenuation) gives a consistent throughput with multiple APs where as for a single AP it drops to nil.
2) The Max Throughput policy will not give the sum of individual bit rates of the channels, as it will incur additional overhead.
3) The upper limit of the achievable bit rate is the value supported by the device

4) Multihoming is more useful in the scenario where the device can be connected to different networks with non-overlapping multipath transfer.

REFERENCES :

1. Cisco Press Blogs
2. Wikipedia.org
3. Cisco.com Support Forum
4. Apnic.net
5. *computer.howstuffworks.com*
6. Contents Available on internet.

IJSER