Research Inventy: International Journal Of Engineering And Science
Vol.3, Issue 3 (June 2013), PP 37-41
Issn(e): 2278-4721, Issn(p):2319-6483, Www.Researchinventy.Com

Topology Control across Mps Using Flow Slice in Wireless Networks

[1,]Gagana. K , [2],Asha
[1,]*M.Tech student, Dept. of CSE, Dr.AIT, Bangalore.*
[2],*Associate Professor, Dept of CSE, Dr.AIT, Bangalore.*

ABSTRACT: *Topology control is adjusting the network topology according to some criteria. The criteria can be scalability, network efficiency and much more. In order to have an efficient network, the packets in the network should be load balanced. In this paper a novel scheme is developed which balances the load on a finer granularity. Flow slice scheme ensures uniform load sharing, intra flow packet ordering and minimum hardware complexity. The theoretical analysis in this paper is also validated through discrete event simulations.*

KEYWORDS: *Load Balancing, traffic measurement, switching theory.*

I. INTRODUCTION

Wireless network refers to the technology that enables two or more computers to communicate using standard network protocols without network cabling. Multipath switching systems (MPS) are intensively used in state-of-the-art core routers to provide terabit or even peta bit switching capacity. One of the most intractable issues in designing MPS is how to load balance traffic across its multiple path while not disturbing the intra flow packet orders. Topology control is a technique used in distributed computing to alter the underlying network in order to reduce the cost. Topology control is consumed mostly by the wireless networks research community. The main aim of topology control in this domain is load balancing.There are several packet based solutions used in this regard. But all of them suffer from one or the other problem. In this paper we have used the concept of flow slice. The flow slice scheme achieves these three objectives simultaneously.

Uniform load sharing :Traffic dispatched to each path should be uniform. Specifically in MPS, traffic destined for each output should be spread evenly to avoid output contention, minimize average packet delay and maximize the throughput.

Intra flow packet ordering : Packets in the same flow should depart MPS as their arrival orders. This ordering is essential since out of order packets will degrade the performance of higher level protocols. For any two packets P_1 and P_2 in the same flow with arrival time $T(P_1)$, $T(P_2)$ and departure time $D(P_1)$, $D(P_2)$ the formula below should be guaranteed:
$D(P_1) < D(P_2)$ if $T(P_1) < T(P_2)$.

Low Timing and hardware complexity*:* The load balancing and additional re sequencing mechanisms at MPS should work fast enough to match the line rate, and should introduce limited hardware complexity. MPS is most likely to hold hundreds of external ports operating at ultra high speed. To provide such scalability, the timing/hardware complexity of O(1) is necessary.

The rest of this paper is organized as follows: Section II summarizes related work. Section III defines flow slice and proposes our load balancing scheme. Section IV explains the implementation of flow slice scheme. Section V presents the results of our work. Finally the conclusion of paper is in section VI.

II. RELATED WORK

A basic timestamp-based re sequencer was proposed by Turner [1] to deal with the cell out-of-order issue, when cells within a virtual circuit are allowed to take different paths in ATM switching systems. In each scheduling, the re sequencer implements a smart contention resolution mechanism to select the oldest cell and then compares its age with a predefined threshold equal to the system delay upper bound. If the selected cell exceeds this age threshold, it will be sent without disturbing cell orders since all previously arrived cells have left the system. Henrion improved the timestamp-based re sequencer by introducing time-wheel-like hardware [2] which does not need to compare cells' timestamps at output. It maintains an array of D pointer lists, where D is larger than delay upper bound at system measured by timeslots. Each pointer at the list stores location of a cell waiting for re sequencing. At timeslot t, the pointer list at slot t modulo D is linked to

the tail of the output cell list and removed from the array. After that, pointers of arrival cells at timeslot t are linked together and stored in this empty slot (t modulo D) of the array. This approach delays every cell by fixed D timeslots with O(1) complexity and strictly guarantees cell orders. Fixed threshold timestamp-based re sequencers work well for ATM switching systems where traffic is well defined and cell delay is moderate. However, in transition to an IP network, traffic becomes rather busty, introducing a higher age threshold on these re sequencers and equalizing cell delay to undesirable level. To cope with the traffic in the worst case, Turner proposed an adaptive re sequencer [3] that dynamically adjusts threshold according to a real-time delay bound. Trace-driven simulations show that the re sequencer performs well better than the fixed threshold one. Another kind of re sequencer uses sequence number instead of timestamp. Cells with the same priority from one input to one output are numbered sequentially at their arrivals. At the output, an expected sequence number is maintained for all the packets in the same flow. Only the cells with an expected sequence can be sent after reaching output, thus preserving cell order. Predefined path scheduling protocol known by both input and output, such as Round- Robin (RR), can be employed to avoid attaching a sequence number to each cell. Re sequencers using VIQ [4], [5], [6] can be categorized in this class. However, to recover from cell loss, more complicated mechanisms should be introduced, such as re sequencers developed by Chiussi et al. [7] in designing Switched Connection Inverse Multiplexing for ATM (SCIMA) and the solution named Rank [8], which maintains k-1 fields in each cell header to deal with cell loss. However, Rank is not scalable for MPS having more than a hundred internal paths.

III. FLOW SLICE

A flow slice is a sequence of packets in a flow, where every intra flow interval between two consecutive packets is greater than or equal to a slicing threshold Φ. Flow slices can be seen as mini flows created by cutting off every intra flow interval larger than Φ. Flow slice scheme can be described as follows.The flow slice scheme is applied to all the nodes other than the source node. When the source node receives the data it first identifies all the possible paths to reach the destination. Out of all the possible paths identified it only identifies such path which does not have shared link. Depending on the number of paths without shared link the data at the source will be divided and sent across those selected paths.Once each node receives the data it slices the data. The data sliced depends upon the bandwidth of the node and the slicing threshold value. At each node, the sliced data is kept within itself and the remaining data is sent to its neighbor. The same process is repeated until the data is reached to the destination. At the destination only some part of the data is reached. The remaining data which is present at each node as a result of slicing will reach the destination through buffering concept. The advantages of flow slice are:

1. It is immune to packet loss, while other solutions like the VIQ re sequencer require additional loss detection mechanisms.

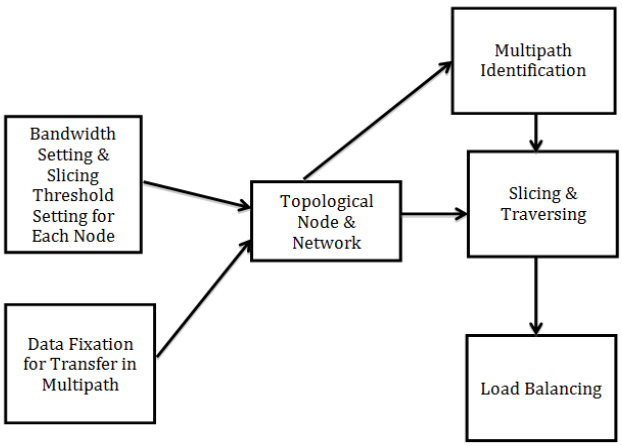

Figure 1: General design of flow slice concept

2. It maintains a hash table to record active flow-slice context, a redirection mechanism can be added to provide robustness to system failure. When some switching path stops working, the load balancer can simply redirect all the active flow slices going to this path at their next packet arrivals, still by FS.
3. It natively supports multicast. Multicast flows are treated in the same manner as unicast flows and still preserve packet orders.
4. N-FS supports load balancing across uneven switching paths by applying weighted round robin.

IV. IMPLEMENTATION

We have implemented this flow slice concept for a wireless network using NS 2 simulator. The topology used is as shown in the figure. However it can be changed to any topology. Initially the network will identify all its neighbor nodes using Euclidian distance. Once the neighbor nodes have been identified the nodes within a particular network will keep sending hello messages to its neighbor node. This hello message indicates that the node is active in the network. Suppose consider node 0 wants to send the 100TB of data to node 10. Initially the source node 0 will identify the multiple paths which do not have any shared link. In the example given below the paths are 0-1-3-6-10 and 0-2-4-7-10. Next the source will divide the data into parts depending on the number of paths. Here the data is divided into 50TB each at each path. Node 1 receives 50TB of data. It slices according to its bandwidth and the slicing threshold. Suppose if the threshold value is 5 TB then the data will be sliced greater than the 5 TB with respect to the bandwidth. Remaining data will be sent to the next neighboring node. Same thing repeats at the paths chosen and until part of data reaches destination. Once the destination has received part of its data then the remaining data can be easily transferred to the destination. The remaining data will be buffered at each node and the destination would have got all the information about how much data is present in each node. Finally the destination receives all the data.

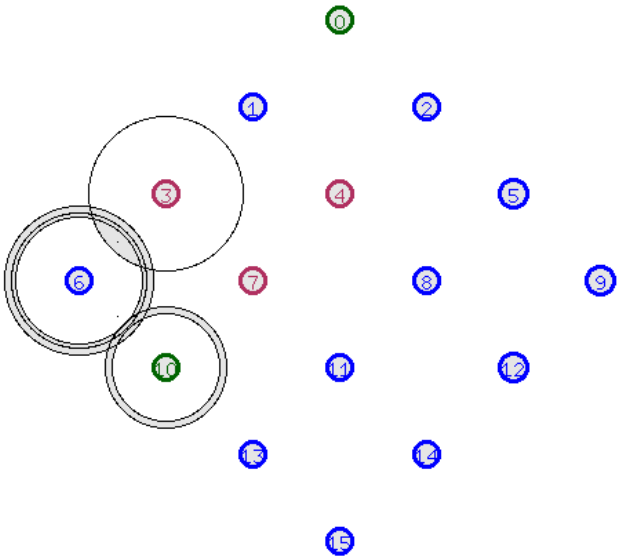

Figure 2: Flow slice implementation for the above topology

V. RESULTS

The concept of flow slice is being used in the wireless network for data delivery. Network simulator is used for simulation. The simulation results for the above topology is explained below.

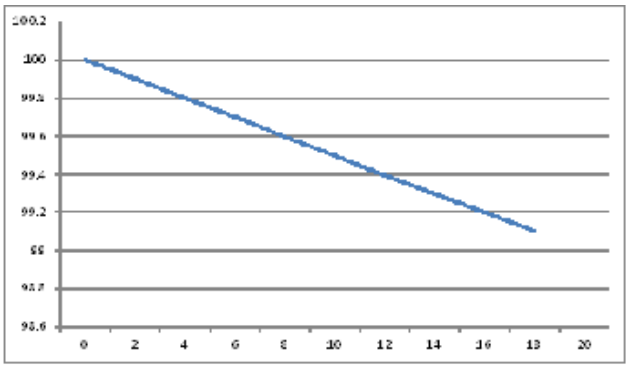

Figure 3: Variation in Energy

Initially all the nodes will be given certain amount of energy in order to make the transaction. In this experiment each node is given a energy of 100 joules. As the transaction takes place the energy gets down which is depicted in figure 3.

Packet loss occurs when one or more packets of data travelling across a computer network fail to reach their destination. Packet loss can be caused by a number of factors including signal degradation over the network medium due to multi-path fading, packet drop because of channel congestion

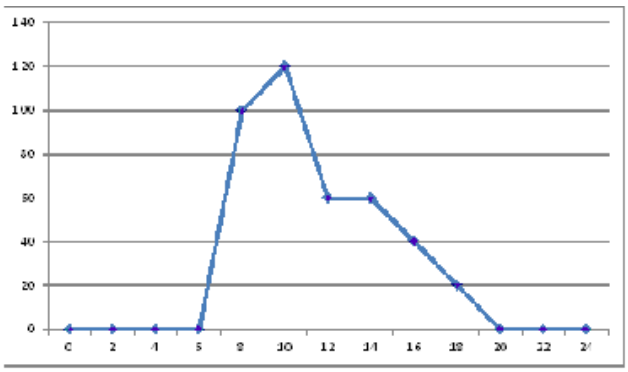

Figure 4: Packet Drop

corrupted packets rejected in-transit, faulty networking hardware, faulty network drivers. Figure 4 shows that at the initial time and end time of transaction there is no packet drop. Somewhere in the middle some packets have been dropped and gradually the packet drop has decreased.

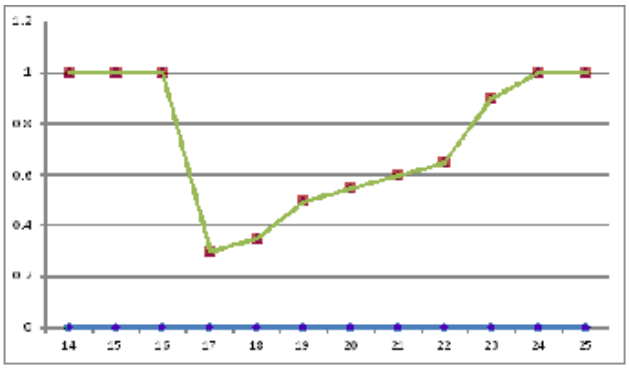

Figure 5: Packet delivery ratio

The ratio of the number of delivered data packet to the destination is called packet delivery ratio. This illustrates the level of delivered data to the destination. The greater value of packet delivery ratio means the better performance of the protocol. In figure 5 we can see that the packet delivery ratio is 1 most of the time.

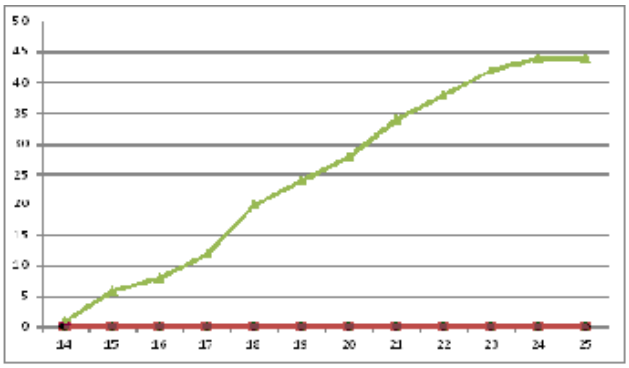

Figure 6: Throughput

As shown in the above figure through put is increasing indicating good signs in the delivery of data.

VI. CONCLUSION AND FUTURE WORK

Compared to the other approaches used for preventing the mis sequencing of packets flow slice scheme is better as it ensures uniform load sharing, intra flow packet ordering and low timing and hardware complexity. Here there is no need of using an additional re sequencing buffer at the output, hence reducing the hardware complexity We have proven FS to be effective under strictly admissible traffic, but it is also important to know how it behaves under extreme traffic. Our simulations with busty real traces shed some light on this issue and suggest that it still work well. Actually, if only the slicing threshold is larger than the delay variation bound at all switching paths, packet order will be undisturbed. Under busty input traffic, the delay at all switching paths may increase synchronously, leaving its delay variation bound nearly unchanged. The FS scheme is validated in switches without class-based queues. As QoS provisioning is also critical in switch designs, one of our future works will be studying FS performance under QoS conditions.

REFERENCES

[1] J.S. Turner, “Resequencing Cells in an ATM Switch,” Technical Report WUCS-91-21, Feb. 1991.

[2] M. Henrion, “Resequencing System for a Switching Node,” US Patent, 5,127,000, June 1992.

[3] J.S. Turner, “Resilient Cell Resequencing in Terabit Routers,” Technical Report WUCS-03-48, June 2003.

[4] A. Aslam and K. Christensen, “Parallel Packet Switching Using Multiplexors with Virtual Input Queues,” Proc. Ann. IEEE Conf. Local Computer Networks (LCN), pp. 270-277, 2002.

[5] C.S. Chang, D.S. Lee, and Y.S. Jou, “Load Balanced Birkhoff-von Neumann Switch, Part II: Multi-Stage Buffering,” Computer Comm., vol. 25, pp. 623-634, 2002.

[6] I. Keslassy and N. Mckeown, “Maintaining Packet Order in Two- Stage Switches,” Proc. IEEE INFOCOM, pp. 1032-1041, 2002.

[7] F.M. Chiussi, D.A. Khotimsky, and S. Krishnan, “Generalized Inverse Multiplexing of Switched ATM Connections,” Proc. IEEE Conf. Global Comm. (GLOBECOM), pp. 3134-3140, 1998.

[8] D.A. Khotimsky, “A Packet Resequencing Protocol for Fault-Tolerant Multipath Transmission with Non-Uniform Traffic Splitting,” Proc. IEEE Conf. Global Comm. (GLOBECOM), pp. 1283-1289, 1999.

[10] C. S. Chang, D. S. Lee, and Y. S. Jou, "Load Balanced Birkhoff-von Neumann Switches, Part I: One-Stage Buffering", Computer Communications, vol. 25, 2002.

[11] I. Keslassy, N. McKeown, "Maintaining packet order in two stage switches," Proceedings of the IEEE Infocom, June 2002.

[12] C. S. Chang, D. S. Lee, and C. Y. Yue, "Providing Guaranteed Rate Services in the Load Balanced Birkhoff-von Neumann Switches", IEEE INFOCOM, San Francisco, March 2003.

[13] Lei Shi, Bin Liu, Changhua Sun, Zhengyu Yin, “Load Balancing Multipath Switching System With Flow Slice,” IEEE Trans. Computers, vol.61, no.3, March 2012

[14] Vitesse Intelligent Switch Fabrics, http://www.vitesse.com, 2011.

[15] J. Bennett, C. Partridge, and N. Shectman, “Packet Reordering Is Not Pathological Network Behavior,” IEEE/ACM Trans. Networking, vol. 7, no. 6, pp. 789-798, Dec. 1999.