IPHV1I90001x
International Journal Of Advanced Research and Innovations Vol.1, Issue .9
ISSN Online: 2319 – 9253
Print: 2319 – 9245

A Study on Multipath systems with load balancing issues

A.Sharada[1] B.Sahadev[2] D.Kalpana[3] Prof. P.Pradeep Kumar[4]

1.[M.Tech(CS)], Dept.of CSE& IT , Vivekananda Institute of Technology and Science, Karimnagar.

2. Sr. Asst. Prof., Dept.of CSE& IT , Vivekananda Institute of Technology and Science, Karimnagar.

3. Asst.Prof, Dept.of CSE& IT , Vivekananda Institute of Technology and Science, Karimnagar.

4. Professor & Dean, Vivekananda Institute of Technology and Science, Karimnagar.

ABSTRACT

Load balancing is the crucial task for every network enabled server. We are trying to provide the solutions of load balancing in multipath switching systems. The key concept is that handling of traffic in multiple paths without disturbing the interflow packets and its orders. The existing systems were suffered with delay in accessing and hardware complexities. We focused on one more important aspect is that flow slicing of each flow to achieve the optimum utilization with balancing the load on a finer granularity.

Key words: Multi paths, routing, flow control, flow slices, load balancing pockets.

I. Introduction

Multipath routing is the routing technique of using multiple alternative paths through a network, which can yield a variety of benefits such as fault tolerance, increased bandwidth, or improvedsecurity. The multiple paths computed might be overlapped, edge-disjointed or node-disjointed with each other. Extensive research has been done on multipath routing techniques, but multipath routing is not yet widely deployed in practice. CMR (Concurrent Multipath Routing)[4] is often taken to mean simultaneous management and utilization of multiple available paths for the transmission of streams of data emanating from an application or multiple applications. In this form, each stream is assigned a separate path, uniquely to the extent supported by the number of paths available. If there are more streams than available paths, some streams will share paths. This provides better utilization of available bandwidth by Creating multiple active transmission queues. It also provides a measure of fault tolerance in that, should a path fail, only the traffic assigned to that path is affected, the other paths continuing to serve their stream flows; there is also, ideally, an alternative path immediately available upon which to continue or restart the interrupted stream.

This method provides better transmission performance and fault tolerance by providing:

- Simultaneous, parallel transport over multiple carriers.
- Load balancing over available assets.
- Avoidance of path discovery when re-assigning an interrupted stream.

Shortcomings of this method are:

- Some applications may be slower in offering traffic to the transport layer, thus starving paths assigned to them, causing under-utilization.

- Moving to the alternative path will incur a potentially disruptive period during which the connection is re-established.

A more powerful form of CMR (true CMR)[4] goes beyond merely presenting paths to applications to which they can bind. True CMR aggregates all available paths into a single, virtual path. All applications offer their packets to this virtual path, which is de-muxed at the Network Layer, the packets then being distributed to the actual paths via some method such as round-robin or weighted fair queuing. Should a link or relay node fail, thus invalidating one or more paths, succeeding packets are not directed to that (those) paths. The stream continues uninterrupted, transparently to the application. This method provides significant performance benefits over the former:

- By continually offering packets to all paths, the paths are more fully utilized.
- No matter how many nodes (and thus paths) fail, so long as at least one path constituting the virtual path is still available, all sessions remain connected. This means that no streams need to be restarted from the beginning and no re-connection penalty is incurred.

It is noted that true CMR can, by its nature, cause Out-Of-Order-Delivery (OOOD) of packets[5][8], which is severely debilitating for standard TCP. Standard TCP, however, has been exhaustively proven to be inappropriate for use in challenged wireless environments and must, in any case, be augmented by a facility, such as a TCP gateway, that is designed to meet the challenge. One such gateway tool is SCPS-TP, which, through its Selective Negative Acknowledgement (SNACK) capability, deals successfully with the OOOD problem.

Another important benefit of true CMR, desperately needed in wireless network communications, is its support for enhanced security. Simply put, for an exchange to be compromised, multiple of the routes it traverses must be compromised. The reader is referred to the references in the “To improve network security” section for discussion on this topic.

II. Related systems

Server Load Balancing:

Coyote Point introduced our first product, the Equalizer Load Balancer[13] in 1999 and we’ve been improving it ever since.

A server load balancer is a device that sits between application clients like web browsers and applications servers. You can deploy as many servers as you like behind a single network address. The load balancer’s job is to optimally distribute client requests (load) across a cluster of servers.

The benefits provided by load balancing technology are illustrated in the animation below. On the left we have users, accessing a web-based application over the Internet. On the right we have a server farm providing web, middleware and database services that typically make up website, application or other services.

Go ahead and play the video. You’ll see:

- Client-server traffic without load balancing. Clients send requests to a “server” and all those requests wind up on… Exactly one server.
- At high-traffic times a single server can be easily overloaded. At best, users experience slow response times. At worst the server can crash, resulting in timeouts and “site unavailable” messages for the clients.

ISSN Online: 2319 – 9253
Print: 2319 – 9245

- By introducing an Equalizer load balancer in front of the servers, response times improve, server failures are recoverable and server resources are optimized.

Equalizer's load balancing[1] feature set improves application performance by intelligently directing client requests to the servers that are best able to handle them. Load balancing decisions are made in real time, taking into account server performance values measured "on the wire". Equalizer continuously measures server latency and connection backlog, integrating these values with configurable health checks while building a comprehensive view of the server cluster. Intelligent load balancing can improve server performance by 20-50% by preventing servers from operating outside of their peak performance range.

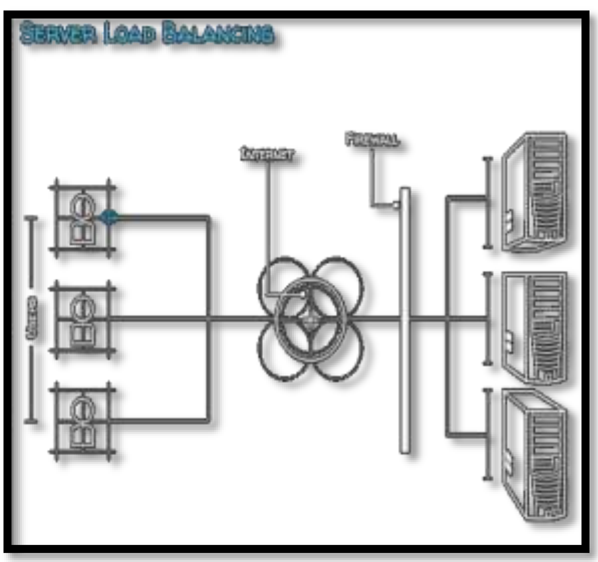


Fig 1. Load balacing

It's easy to load balance both virtual and physical servers. Equalizer provides a powerful, easy-to-use, web-based user interface as well as an enterprise-grade command line, to access the full load balancing toolkit. Features include:

- **Client-server persistence or affinity**. This important capability directs returning clients to the same server every time, and is often necessary for applications like ecommerce shopping carts where the client and server share state.
- **Layer-7 content switching rules**, which direct clients to specific pools of servers based on request parameters.
- **A broad range of server health checks** to ensure that application servers are fully functional and responsive. Servers returning errors are automatically removed from the load balancing pool, and configurable alerts notify administrators of the problem.

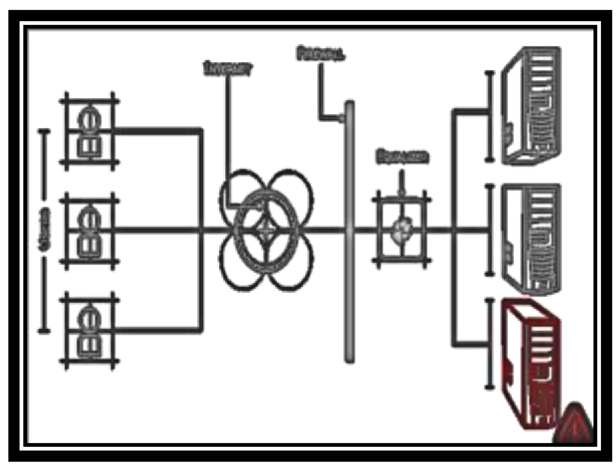


Fig 3. Load balancing with equalizer

III. Multipath routing in wireless networks

To improve performance or fault tolerance:

CMR (Concurrent Multipath Routing) is often taken[7][11] to mean simultaneous management and utilization of multiple available paths for the transmission of streams of data emanating from an application or multiple applications. In this form, each stream is assigned a separate path, uniquely to the extent supported by the number of paths available. If there are more streams than available paths, some streams will share paths. This provides better utilization of available bandwidth by creating multiple active transmission queues. It also provides a measure of fault tolerance in that, should a path fail, only the traffic assigned to that path is affected, the other paths continuing to serve their stream flows; there is also, ideally, an alternative path immediately available upon which to continue or restart the interrupted stream.

This method provides better transmission performance and fault tolerance by providing:

- Simultaneous, parallel transport over multiple carriers.
- Load balancing over available assets.
- Avoidance of path discovery when re-assigning an interrupted stream.

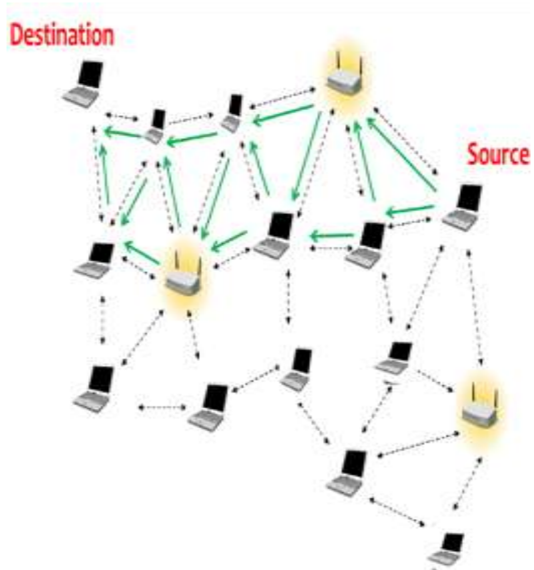
Shortcomings of this method are:

- Some applications may be slower in offering traffic to the transport layer, thus starving paths assigned to them, causing under-utilization.
- Moving to the alternative path will incur a potentially disruptive period during which the connection is re-established.

A more powerful form of CMR (true CMR) goes beyond merely presenting paths to applications to which they can bind. True CMR aggregates all available paths into a single, virtual path. All applications offer their packets to this virtual path, which is de-muxed at the Network Layer, the packets then being distributed to the actual paths via some method such as round-robin or weighted fair queuing. Should a link or relay node fail, thus invalidating one or more paths, succeeding packets are not directed to that (those) paths. The stream continues uninterrupted, transparently to the application. This method provides significant performance benefits over the former:

- By continually offering packets to all paths, the paths are more fully utilized.
- No matter how many nodes (and thus paths) fail, so long as at least one path constituting the virtual path is still available, all sessions remain connected. This means that no streams need to be restarted from the beginning and no re-connection penalty is incurred.

It is noted that true CMR can, by its nature, cause Out-Of-Order-Delivery (OOOD) of packets, which is severely debilitating for standard TCP. Standard TCP, however, has been exhaustively proven to be inappropriate for use in challenged wireless environments and must, in any case, be augmented by a facility, such as a TCP gateway, that is designed to meet the challenge. One such gateway tool is SCPS-TP, which, through its Selective Negative Acknowledgement (SNACK) capability, deals successfully with the OOOD problem..



International Journal Of Advanced Research and Innovations Vol.1, Issue .9
ISSN Online: 2319 – 9253
Print: 2319 – 9245

SYSTEM ARCHITECTURE:

IV. EXISTING SYSTEM:

Our major improvement over the existing works is to tailor the approach in the scenario by introducing the offline delay bound calculation, while the previous solutions either use an empirical slicing threshold or maintain flow context to facilitate the slicing. The traces here are collected at backbone links of one of the largest commercial backbones worldwide.

Disadvantage:

Our major improvement over the existing works is to tailor the FS approach in the MPS scenario by introducing the offline delay bound.

PROPOSED SYSTEM:

In this project a novel load-balancing scheme, namely, Flow Slice, based on the fact that the intraflow packet interval is often, larger than the. Due to three positive properties of flow slice, our scheme achieves good load-balancing uniformity with little hardware overhead and timing complexity. By calculating delay bounds at three popular, we show that when the slicing threshold is set to the smallest admissible value at, the FS scheme can achieve optimal performance while keeping the intraflow packet out-of-order probability negligible given an internal speedup up to two. Our results are also validated through trace-driven prototype simulations under traffic patterns.

Advantage:

It is immune to packet loss, while other solutions like the resequencer require additional loss detection Mechanisms.

Approaches to resolve Load Balancing issues:

1. **LOAD-BALANCING SCHEME,**
2. **MULTIPATH SWITCHING SYSTEM,**
3. **MULTISTAGE MULTIPLANE CLOS SWITCHES,**

LOAD-BALANCING SCHEME:

Interflow packet order is natively preserved besetting slicing threshold to the delay upper bound at .Any two packets in the same flow slice cannot be disordered as they are dispatched to the same switching path where processing is guaranteed; and two packets in the same flow but different flow slices will be in order at departure, as the earlier packet will have depart from before the latter packet arrives. Due to the fewer number of active flow slices, the only additional overhead in, the hash table, can be kept rather small, , and placed on-chip to provide ultrafast access speed. This table size depends only on system line rate and will stay unchanged even if scales to more than thousand external ports, thus guarantees system scalability.

MULTIPATH SWITCHING SYSTEM:

Through lay-aside Buffer Management module, all packets are virtually queued at the output according to the flow group and the priority class in a hierarchical manner. The output scheduler fetches packets to the output line using information provided by. Packets in the same flow will bevirtually buffered in the same queue and scheduled in discipline.

Hence, intraflow packet departure orders holdas their arriving orders at the multiplexer. Central-stage parallel switches adopt an output-queued model. By Theorem, we derive packet delay bound at firststage. We then study delay at second-stage switches. Define native packet delay at stage m of an be delay experienced at stage m on the condition that all the preceding stages immediately send all arrival packets out without delay.

MULTISTAGE MULTIPLANE CLOS SWITCHES:

We consider the Multistage Multi plane Close network based switch by Chao et a . It is constructed of five stages of switch modules with top-level architecture similar to a external input/output ports. The first and last stages close are composed of input demultiplexers and output multiplexers, respectively, having similar internal structures as those in PPS. Stages 2-4 of M2Clos are constructed by parallel switching planes; however, each plane is no longer formed by a basic switch, but by a three-stage Close Network to support large port count. Inside each Close Network, the first stage is composed by k identical Input Modules. Each IM is a packet switch, with each output link connected to a Central Module. Thus, there is a total of m identical in second stage of the Close networks.

CONCLUSION

We proposed a novel load-balancing scheme, namely, Flow Slice, based on the fact that the intra flow packet interval is often, say in 40-50 percent, larger than the delay upper bound at MPS. Due to three positive properties of flow slice, our scheme achieves good load-balancing uniformity with little hardware overhead and O(1) timing complexity. By calculating delay bounds at three popular MPSes, we show that when the slicing threshold is set to the smallest admissible value at 1-4 ms, the FS scheme can achieve optimal performance while keeping the intraflow packet out-of-order probability negligible (below 10_6), given an internal speedup up to two. Our results are also validated through trace-driven prototype simulations under highly busty traffic patterns.

References

[1] Cisco CRS-1, http://www.cisco.com/go/crs/, 2011.

[2] Real Traces from NLANR, http://pma.nlanr.net/, 2010.

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

[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] 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.

[6] N. Brownlee and K. Claffy, “Understanding Internet Traffic Streams: Dragonflies and Tortoises,” IEEE Comm. Magazine, vol. 40, no. 10, pp. 110-117, Oct. 2002.

[7] Z. Cao, Z. Wang, and E. Zegura, “Performance of Hashing-Based Schemes for Internet Load Balancing,” Proc. IEEE INFOCOM, pp. 332-341, 2000.

[8] L. Carter and M. Wegman, “Universal Classes of Hashing Functions,” J. Computer and System Sciences, vol. 18, no. 2, pp. 143-154, 1979

[9] 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.

[10] C.S. Chang, D.S. Lee, and Y.S. Jou, "Load Balanced Birkhoff-von Neumann Switches, Part I: One-Stage Buffering," Computer Comm., vol. 25, pp. 611-622, 2002.

[11] C.S. Chang, D.S. Lee, and Y.J. Shih, "Mailbox Switch: A Scalable Two-Stage Switch Architecture for Conflict Resolution of Ordered Packets," Proc. IEEE INFOCOM, 2004.

[12] C.S. Chang, D.S. Lee, and C.Y. Yue, "Providing Guaranteed Rate Services in the Load Balanced Birkhoff-von Neumann Switches," IEEE/ACM Trans. Networking, vol. 14, no. 3, pp. 644-656, June 2006.

[13]http://www.coyotepoint.com/solutions/server-load-balancing

[14] 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.

[15] G. Dittmann and A. Herkersdorf, "Network Processor Load Balancing for High-Speed Links," Proc. Int'l Symp. Performance Evaluation of Computer and Telecomm. Systems (SPECTS), 2002.