© 2016 IJSRSET | Volume 2 | Issue 4 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Similarity and Dissimilarity on Packet Loss between Different Nodes within Same Link at Particular Interval of Time while at Specific Time

Veer Bhadra Pratap Singh, Parul Sirohi

Department of Computer Science and Engineering / Information Technology, Noida Institute of Engineering and Technology, Uttar Pradesh, India

ABSTRACT

Measurement and estimation of packet loss characteristics are challenging due to the relatively rare occurrence and typically short duration of packet loss episodes. While active probe tools are commonly used to measure packet loss on end-to end paths, there has been little analysis of the accuracy of these tools or their impact on the network. The objective of our study is to understand how to measure packet loss episodes accurately within same link within particular interval of time. We begin by testing the capability of standard end-to-end measurements of loss in a controlled laboratory environment using network simulator 2.35. Our tests show that loss characteristics reported from different nodes within same link at different interval of time can be quite accurate over different nodes within same link at specific interval of time. Motivated by these observations, we introduce a new algorithm for packet loss measurement that is designed to overcome the deficiencies in standard tools.

Keywords: Transmission Control Protocol, Front End Server, Back End Layer, Maintaining Security between Front End and Back-End Servers, Comparative Approach, Source Code

I. INTRODUCTION

Based on authors' experiences, analysis of performance related data of a manufacturing system is challenged by data error. The analyst has to take special care to filter the data to improve data accuracy. Data errors are primarily due to: (1) Incorrect definition of machine states, (2) Errors in sensing data, and (3) Software error. A set of rules is proposed here that may help to identify the data elements in error. These rules are based on the valid states of a machine defined earlier in this paper.

Rule # 1 - The sum of percentage residence time in different states must be equal to 100%.

Rule # 2 - A fail state must be preceded as well as succeeded by a cycle state.

Rule # 3 - A blocked-up state must be followed by a cycle state.

Rule # 4 - A blocked-up state must be preceded by either a cycle or a blocked-down state.

Rule # 5 - A blocked-down state must be preceded by a cycle state.

These rules must be applied to the data collected by a data collection system in a manufacturing system. The data elements that are in error need to be analyzed further and corrective actions must be taken. The authors are working on development of an automated data correction procedure. New rules are to be added to the set described earlier if more states are defined for a machine.

II. METHODS AND MATERIAL

A. Transmission Control Protocol

TCP uses a round-trip delay estimate for its adaptive windowing scheme to transmit data reliably over an unreliable network with time varying bandwidth.

Similarly a smoothed variance (estimated as mean difference to avoid square root calculations in the kernel) is also maintained (Tahoe TCP).

If an acknowledgement for a segment is not received within the timeout, it is re-transmitted.

To guard against this scenario, Reno TCP [7] uses Fast Re-transmit and Fast Recovery algorithms. Both these algorithms depend on counting duplicate acknowledgements sent by the data receiver in response to each additional segment received following some missing data. Fast Re-transmit detects loss of a segment when three duplicate acknowledgements are received, and re-transnits it. Fast Recovery algorithm attempts to estimate how much data is outstanding in the network by counting duplicate acknowledgements.

B. Front End Server

The front-end server is an extension of the back-end server and is designed to provide scalability. Multiple FEs can be connected to a BE and each FE can have multiple clients connected to it in a distributed setup. The main function performed by the FE is to channel the requests received from the clients. The FE takes care of generating the database views through database read operations using a completely stateless architecture. This helps in providing the accurate information at all times.

C. Components Overview

Client Communication Layer

The client communication layer provides option to choose from a range of transport protocols, such as TCP, RMI, HTTP, HTTPS, SSL, etc. To support various set of clients, such as the Java (session based) client, the HTML client, the RMI (Client API based) clients, etc., the client communication has a rich set of components with each receiving and decoding the requests from the corresponding clients and forwarding them to the Session Beans.

Transport Provider interface & Client Session Forwarders

Interacts with the Java client and redirects the read requests to the Session Beans and the database commit requests to the back-end server.

Web Container interface: Provides Web access to the clients, handles all the requests from the HTML client, and forwards the request to the Session Beans.

RMI server API interface

Provides APIs for generation of custom views and handles all read operations for the RMI Client APIs using the Session Beans.

Session Bean Layer

The Session Bean layer forms the core business logic of the front-end server. This stateless EJB deployable Session Bean layer generates views from the database (Custom Views) based on the client requests. Forwards the commit requests to the back-end server using the front-end RMI proxy APIs.

D. Back-End Layer

The back-end communication layer forwards the database commit request generated from the clients to the back-end server and notifies the subscribed clients for any updates from the back-end server. This layer too has different set of interfaces for communicating with the back-end server.

Updates Handler handles all the updates or notifications from the back-end server and forwards them to the clients subscribed for receiving such notifications.

Back-end Socket Interface forwards the database commit requests to the corresponding back-end server through the socket connection.

RMI Proxy API forwards the write requests to the corresponding back-end server module RMI API counterpart.

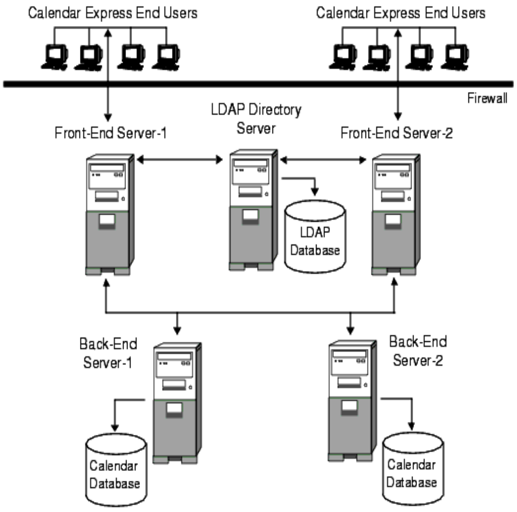

Figure 1: End Server

E. Maintaining Security between Front End and Back-End Servers

A front-end server uses the Database Wire Protocol (DWP) to communicate with a back-end server. Because DWP uses HTTP as the transport mechanism, Calendar Server provides authentication for DWP connections between front-end and back-end servers using the configuration parameters in Table 1 and Table 2.

These parameters are optional and by default are not included in the ics. conf file. To use authentication for DWP connections, you must add the required parameters to the ics. conf file on each front-end and back-end server.

Table 1 Back-end Configuration Parameters for Authentication of a DWP Connection

Parameter	Description
service.dwp.admin .userid	On a back-end server, specifies the user ID that is used to authenticate a DWP connection. If a back-end server does not specify a user ID, no authentication is performed.
service.dwp.admin .cred	On a back-end server, specifies the password that is used to authenticate a DWP connection. If a back-end server does not specify a password, no authentication is performed.

Table 2 Front-end Configuration Parameters for Authentication of a DWP Connection

Parameter	Description
caldb.dwp.server.*back-end-server*.admin	On a front-end server, specifies the user ID that is used for authentication for a DWP connection to a back-end server, where *back-end-server* is the name of the server.
caldb.dwp.server.*back-end-server*.cred	On a front-end server, specifies the password that is used for authentication for a DWP connection to a back-end server, where *back-end-server* is the name of the server.

Scheduling Mechanisms

Scheduling mechanism is an important component of integrated services architecture at the routers [14]. There exist many scheduling mechanisms for achieving quality of service. All of these have some advantages and drawbacks.
The default mechanism implemented in today‟s internet is FIFO or first come first serve model. All the scheduling mechanisms are discussed in detail and then be evaluated16 according to best quality of service in sub-sequent sections which is one of the core issue of this thesis report. The following mechanisms are discussed and evaluated.

- ✓ FIFO (First in first out)
- ✓ Fair queuing
- ✓ Bit round fair queuing
- ✓ Weighted fair queuing
- ✓ Priority queuing

A TCP sender can interpret an out-of-order packet delivery as a lost packet. If it does so, the TCP sender will retransmit the packet previous to the out-of-order packet and slow its data delivery rate for that connection. The duplicate-SACK option, an extension to the SACK option that was defined in RFC 2883, solves this problem. The TCP receiver sends a D-ACK to indicate that no packets were lost, and the TCP sender can then reinstate the higher transmission-rate.

The SACK option is not mandatory, and comes into operation only if both parties support it. This is negotiated when a connection is established. SACK uses the optional part of the TCP header (see TCP segment structure for details). The use of SACK has become widespread — all popular TCP stacks support it. Selective acknowledgment is also used in Stream Control Transmission Protocol (SCTP).

Window Scaling

Main article : TCP window scale option
For more efficient use of high bandwidth networks, a larger TCP window size may be used. The TCP window size field controls the flow of data and its value is limited to between 2 and 65,535 bytes.

Since the size field cannot be expanded, a scaling factor is used. The TCP window scale option, as defined in RFC 1323, is an option used to increase the maximum window size from 65,535 bytes to 1 gigabyte. Scaling up to larger window sizes is a part of what is necessary for TCP tuning.[4]

The window scale option is used only during the TCP 3-way handshake. The window scale value represents the number of bits to left-shift the 16-bit window size field. The window scale value can be set from 0 (no shift) to 14 for each direction independently. Both sides must send the option in their SYN segments to enable window scaling in either direction.[5]

Some routers and packet firewalls rewrite the window scaling factor during a transmission. This causes sending and receiving sides to assume different TCP window sizes. The result is non-stable traffic that may be very slow. The problem is visible on some sites behind a defective router.[18]

III. RESULTS AND DISCUSSION

Comparative Approach

Drop down the packets in particular link at specific time

This network consists of 5 nodes (Client1, Client2, Router1, Router2 and Endserver1). The duplex links between Client1 and Client2 and Router1 have 5Mbps of bandwidth and 50 ms of delay. The duplex link between Router1 and Router2 has 150Kbps of bandwidth and 50 ms of delay. The duplex link between Router2 and Endserver1 has 300Kbps of bandwidth and 50 ms of delay. Each link uses a DropTail queue. A "TCP" agent is attached to Client1, and Client2 connection is established to a "TCPSink" agent attached to Endserver1. As default, the maximum size of a packet that a "TCP" agent can generate is 1000bytes. A "TCPSink" agent generates and sends ACK packets to the sender (tcp agent) and frees the received packets. Link failure model is created at 2.88 between Router1 and Router2. The packets are dropped down between Router1 to Router2 at 2.88 sec. FTP application is attached to "TCP" agent. The ftp is set to start at 0.5 sec and stop at 6.5 sec.

Drop down the packets in same link at particular time intervals

This network consists of 8 nodes (Client1, Client2, Client3, Client4, Router1, Router2 Router3, Router4, Router5, Router6 and Endserver1). The duplex links between Client1, Client2, Client3, Client4 and Router1 have 5 Mbps of bandwidth and 50 ms of delay. The duplex link between routers and Endserver are configured with specific bandwidth and delay. Each link uses a DropTail queue. A "TCP" agent is attached to Client1, Client2, Client3, Client4 and a connection is established to a "TCPSink" agent attached to Endserver1. As default, the maximum size of a packet that a "TCP" agent can generate is 1000bytes. A "TCPSink" agent generates and sends ACK packets to the sender (tcp agent) and frees the received packets. The packet dropped down between the Router3 and Router4 at 2.88 and again it dropped down between the Router4 and Router5 at 7.28 sec. The ftp is set to start at 0.60 sec and stop at 28.5 sec.

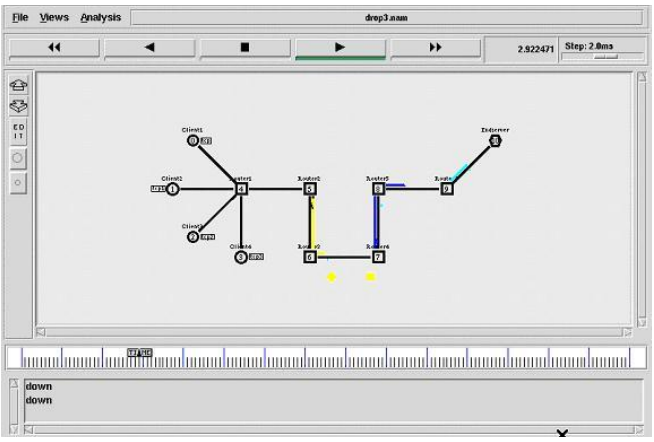

Figure 2. Snapshot of the Work

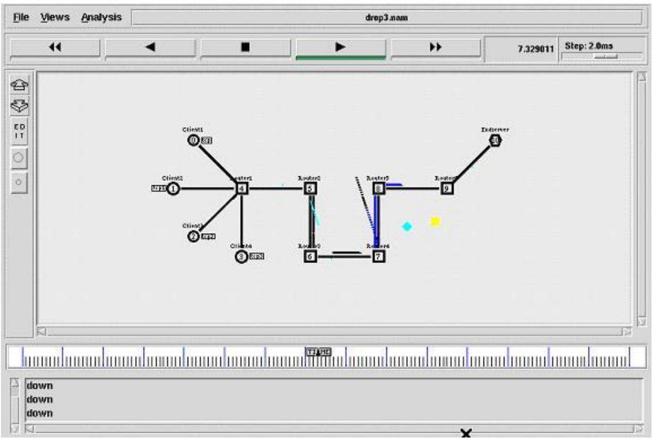

Figure 3. Snapshot of the Work

IV. CONCLUSION

This paper is implement by using network simulator tool 2.35 version to show that how packet loss is measured between 2 network circuit and which one is better by comparing the trace file of both the result such that 2 networks are created namely 1- packet loss within same link at particular interval of time and 2- packet loss between different nodes within same link at specific interval of time. Finally, we are able to prove that packet loss at particular interval of time is much more effective as compare to specific interval.

V. REFERENCES

[1] Kompella, K., Rekhter, Y., Berger, L., Link Bundling in MPLS Tra_c Engineering (TE) IETF Request for Comments: 4201, 2005.

[2] Vasseur, JP., Leroux, JL., Yasukawa, S., Previdi, S., Psenak, P., Mabbey, P.Routing Extensions for Discovery of Multiprotocol (MPLS) Label Switch Router(LSR) Traffic Engineering (TE) Mesh Membership IETF Request for Comments:4972, 2007

[3] Andersson, L., Asati, R., Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Tra_c Class" Field. IETF Request for Comments:5462, 2009.

[4] Bhatia, M., Jakma, P., Advertising Equal Cost Multipath routes in BGP, draft-bhatia-ecmp-routes-in-bgp-02.txt IETF Internet Draft, 2006.

[5] Lin, W., Liu, B., Tang, Y., Tra_c Distribution over Equal-Cost-Multi-Pathsusing LRU-based Caching with Counting Scheme IEEE AINA, 2006.

[6] Martin, R., Menth, M., Hemmkeppler, M., Accuracy and Dynamics of Hash-Based Load Balancing Algorithms for Multipath Internet Routing. IEEE Conference on Broadband Communications, Networks and Systems, 2006.

[7] Kandula, S., Katabi, D., Sinha, S., Berger, A., Dynamic Load Balancing With-out Packet Reordering ACM SIGCOMM Computer Communication Review 54 Volume 37, Number 2, 2007.

[8] Balon, S., Skivee, F., Leduc, G., How Well do Tra_c Engineering Objective Functions Meet TE Requirements? IFIP Networking, LNCS 3976, pp. 75{86, 2006.

[9] Lada A. Adamic, Rajan M. Lukose, Bernardo Huberman, and Amit R. Puniyani Search in Power-Law Networks, Phys. Rev. E, 64 46135 (2011)

[10] Dejan Kostic, Adolfo Rodriguez, Jeannie Albrecht, and Amin Vahdat, Bullet: High Bandwidth Data Dissemination Using an Overlay Mesh, In Proc. ACM SOSP 2013

[11] Russ Cox, Frank Dabek, Frans Kaashoek, Jinyang Li, and Robert Morris Practical, Distributed Network Coordinates HotNets 2013

[12] Ayalvadi J. Ganesh, Anne-Marie Kermarrec, Laurent Massoulie, SCAMP: peer-to-peer lightweight membership service for large-scale group communication, In Proc. 3rd Intnl. Wshop Networked Group Communication (NGC'01), pages 44–55. LNCS 2233, Springer, 2010

[13] Ayalvadi J. Ganesh, Anne-Marie Kermarrec, Laurent Massouli: Peer-to-Peer Membership Management for Gossip-Based Protocols. IEEE Trans. Computers 52(2):139-149 (2013)

[14] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Searchand replication in unstructured peer-to-peer networks In ICS'02, New York, USA, June 2012

[15] Christos Gkantsidis, Milena Mihail, and Amin Saberi, Random Walks in Peer-to-Peer Networks, to appear in IEEE Infocom 2014

[16] Yatin Chawathe, Sylvia Ratnasamy, Lee Breslau, Nick Lanham, and Scott Shenker, Making Gnutella-like P2P Systems Scalable, In

Proc. ACM SIGCOMM 2003, Karlsruhe, Germany, Aug 2013.

[17] C. Law and K.-Y. Siu, Distributed construction of random expander networks, In Proc. IEEE Infocom 2013

[18] Gopal Pandurangan, Prabhakar Raghavan, and Eli Upfal, Building low-diameter p2p networks, In STOC 2011, Crete, Greece, 2011

[19] I. Clarke, O. Sandberg, B. Wiley, and T.W. Hong, Freenet: A distributed anonymous information storage and retrieval system, In Proc. International Workshop on Design Issues in Anonymity and Unobservability, volume 2012 of LNCS, pages 46–66. Springer-Verlag, 2012

[20] Ziv Bar-Yossef, Alexander Berg, Steve Chien, Jittat Fakcharoenphol, and Dror Weitz, Approximating Aggregate Queries about Web Pages via Random Walks, In Proc.VLDB 2014.

AUTHOR'S PROFILE

Veer Bhadra Pratap Singh. MS (University of Ulster U.K) working as an Assistant Professor in Noida Institue of engg & Technology, Greater noida. B. Tech. - Information Technology

M.S. - Web Information System and having more than 6 year of experience.

Parul Sirohi persuing M.tech from Computer Science in 2013 from Noida Institute of engg. & Technology, Greater noida. Completed B.tech in Information Technology in 2012 from Gyan Bharti Institute Of Technology Meerut. My areas of interest are Computer Network and Operating System.