Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 6, Issue. 2, February 2017, pg.01 – 07

SESSION LAYER WORKING BASED ON CLOUD ENVIRONMENTS

P.SURESH
Research Scholar, Department of Computer Science, H.H The Raja's College (Autonomous), India

Abstract

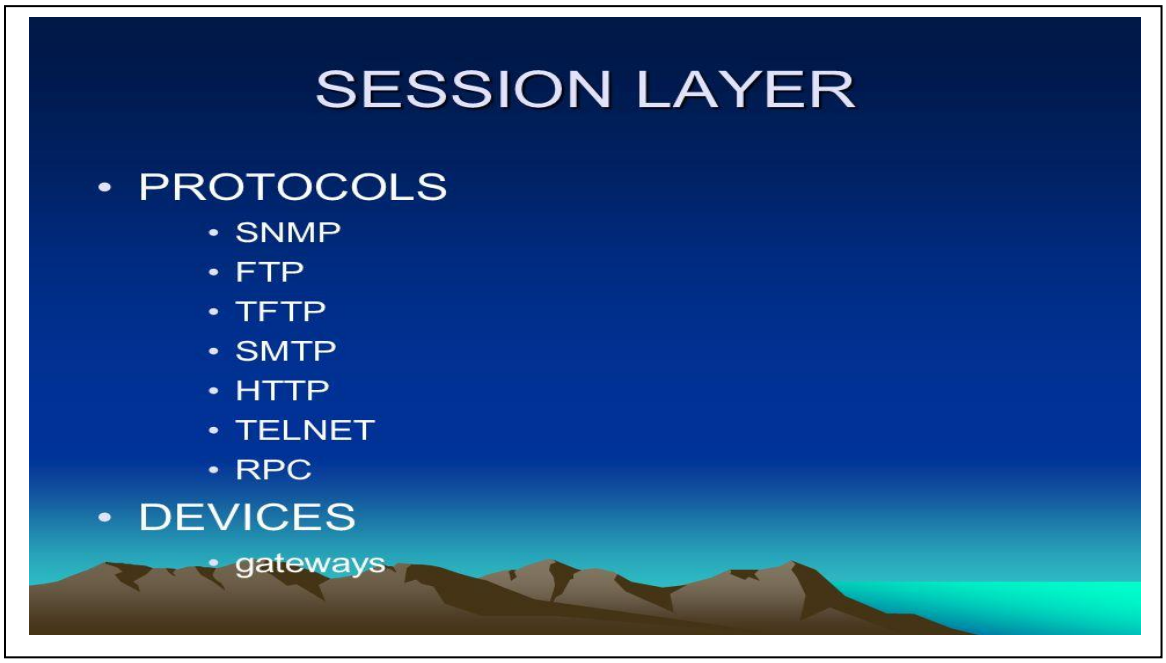
Clouds provide a powerful computing platform that enables individuals and organizations to perform variety levels of tasks such as: use of online storage space, adoption of business applications, development of customized computer software, and creation of a "realistic" network environment. In previous years, the number of people using cloud services has dramatically increased and lots of data has been stored in cloud computing environments. This paper Session-layer services for enhancing functionality and improving cloud network performance are gaining in importance in the Internet. Examples of such services include connection multiplexing, congestion state sharing, application-level routing, mobility/migration support, and encryption. The session layer provides the mechanism for opening, closing and managing a session between end-user application processes, i.e. a semi-permanent dialogue. Communication sessions consist of requests and responses that occur between applications.

Keywords: cloud computing, client and server, osi layer, protocol, session layer.

INTRODUCTION

Cloud computing has been involved in everyone's life. It delivers applications and storage spaces as services over the Internet for little to no cost. Most of us utilize cloud computing services on a daily basis. For example, we use web-based email systems (e.g. Yahoo and Google) to exchange messages with others; social networking sites (e.g. Face book , LinkedIn, MySpace, and Twitter) to share information and stay in contact with friends; on-demand subscription services (e.g. Netflix and Hulu) to watch TV shows and movies; cloud storages (e.g. Humyo, ZumoDrive, and Dropbox) to store music, videos, photos and documents online collaboration tools (e.g. Google docs) to work with people on the same document in real time and online backup tools (e.g. JungleDisk, Carbonite, and Mozy) to automatically back up our data to cloud servers. Cloud computing has also been involved in businesses; companies rent services from cloud computing service providers to reduce operational costs and improve cash flow. For example, the social news website, reddit, rents Amazon Elastic

Compute Cloud (EC2) for their digital bulletin board service. The digital photo sharing website, Smug Mug, rents Amazon S3 (Simple Storage Service) for their photo hosting service. The automaker, Mazda USA, rents Rack space for their marketing advertisements. The software company, HRLocker, rents Windows Azure for their human resources software service. Session layer deals with connections. It establishes, manages, and terminates sessions between two communicating nodes. This layer provides its services to the presentation layer. Session layer also synchronizes dialogue between the presentation layers of the two hosts and manages their data exchange. For example, web servers may have many users communicating with server at a given time. Therefore, keeping track of which user communicates on which path is important and session layer handle this responsibility accurately. This session layer design provides the user with the notion of a session, but more importantly, allows them to authenticate and open up the range of services available to them. This layer makes access control decisions based on who the user is, what groups or roles that user may be a part of and any number of additional policies the network administrator might wish to support. This provides the type of comprehensive access control modern networks need. Yet the Benets of a robust session layer extend beyond simplifying the lives of network administrators and reducing the complexity of security conjuration. The current Internet architecture forces each individual network application to write large amounts of sensitive code to provide security features, including authentication and encryption. Applications often simply omit some of these features, while the remainder provide a wide array of encryption and authorization solutions of varying quality in terms of usability or security. A session layer puts security features on-par with core networking concepts like congestion control. With a session layer in place, applications can take advantage of one united codebase to perform these types of sensitive operations. Support for new authentication mechanisms, new encryption technologies, or other new security features, can be added in one place and made immediately available to all applications running on the session layer. This layer different protocol use on cloud application. List of Session layer protocol and services.in this figure1 session layer protocol list.



- Authentication
- Permissions
- Session restoration (check pointing and recovery)

Authentication

Authentication is the act of establishing or confirming something (or someone) as authentic, that is, that claims made by or about the thing are true. This might involve confirming the identity of a person, the origins of an artifact, or assuring that a computer program is a trusted one.

Permissions or Access control

One familiar use of authentication and authorization is access control. A computer system supposed to be used only by those authorized must attempt to detect and exclude the unauthorized. Access to it is therefore usually controlled by insisting on an authentication procedure to establish with some established degree of confidence the identity of the user, thence granting those privileges as may be authorized to that identity. In some cases, ease of access is balanced against the strictness of access checks. For example, the credit card network does not require a personal identification number, and small transactions usually do not even require a signature. The security of the system is maintained by limiting distribution of credit card numbers, and by the threat of punishment for fraud.

Checkpoints

Session layer is responsible for creating several checkpoints, checkpoints are also treated as recovery points i.e. in case of failure the system rollback to its previous checkpoint configuration or action.

Session different protocol used some cloud application.

- ADSP AppleTalk Data Stream Protocol
- ASP AppleTalk Session Protocol
- H.245 Call Control Protocol for Multimedia Communications
- iSNS Internet Storage Name Service
- L2F Layer 2 Forwarding Protocol
- L2TP Layer 2 Tunneling Protocol
- NetBIOS, File Sharing and Name Resolution protocol - the basis of file sharing with
- Windows.
- NetBEUI, NetBIOS Enhanced User Interface
- NCP NetWare Core Protocol
- NFS Network File System
- PAP Password Authentication Protocol
- PPTP Point-to-Point Tunneling Protocol
- RPC Remote Procedure Call
- RTCP RTP Control Protocol
- SDP Sockets Direct Protocol
- SMB Server Message Block
- SIP Session Initiation Protocol
- SMPP Short Message Peer-to-Peer
- SOCKS "SOCKetS"
- SSH Secure SHell
- ZIP Zone Information Protocol {For AppleTalk}

This layer provides session management capabilities between hosts. For example, if some host needs a password verification for access and if credentials are provided then for that session password verification does not happen again. This layer can assist in synchronization, dialog control and critical operation management (e.g., an online bank transaction).The session layer offers provisions for efficient data transfer. Examples :- SQL, ASP(AppleTalk Session Protocol).

RELATED WORK

One of the challenging parts about explaining fived has been in comparing it to existing technologies. Fived 's design goals revolve around incorporating solutions to problems where we know the network has needs. The solutions fived implements often aren't particularly different than existing technologies. It isn't in the choice of encryption algorithm or the protocol that five's contributions are really understood. It is in the way this session layer enables the use of these technologies in a way that creates a coherent architecture between every application using fived. It is in the way that fived shifts the responsibility into the underlying layers and eliminates sections of security critical code required in many of today's applications. It is in the way fived strictly adheres to the end-to-end principles and eschews any requirement that the core networking hardware know about our protocol for it to succeed. It is in the way the session layer architecture enables application access to these technology in a uniform way across the deployment base. Which isn't to say no other projects have had these goals-is Service-oriented network designs, such as those seen in Planet lab [4] and GENI [7] often have similar design goals. Chandrasekhar's paper on a Service Oriented Internet [3] comes up with a strikingly similar design in some respects. In this paper, a session layer with a service-oriented architecture is fairly clearly proposed and outlined. The main difference between these works and those of fived is a difference in how these systems interact with legacy technology. Many of these designs fall under the category of ″clean slate″ networking architectures, where the goal of the research is to clean up the Internet and switch to a ″better″ architecture. Fived on the other hand, is what I like to call a ″dirty slate″ design. The goal of fived is to add the features into the existing network that seem to be missing. When there's a way to do it that seems to prod the network towards a cleaner architecture, fived takes the opportunity, but the guiding goal is to get the features into the network. The resulting systems turn out fairly different. While the SLaBS model described in this paper breaks new ground, the basic principles behind buffering and bursting are certainly not new concepts and have been proposed for a variety of network technologies. In the early 1980's, Amstutz [2] takes advantage of the busty nature of voice "talk spurts" and data messages in order to dedicate transmission channels only when needed, thereby improving efficiency in telecommunication switches. Protocol optimizations such as Nagle's algorithm [6] are guided by similar ideas at the transport layer. The Delay Tolerant Networking Research Group (DTNRG) [1],with its associated Bundle Protocol [6], attempts to solve the problem of message delivery and routing over challenging network environments, including very large delay transmission and potentially frequently disconnected network paths. Recently, a "session layer" for DTN [7] has been proposed that will allow receiver-driven applications to manage relationships between individual "bundles" of data. SLaBS shares common themes with the bundling approach through slabbing, but we do so at a much coarser granularity while targeting high-performance network environments. Mills et al. set the stage for optical burst switching (OBS) with an architecture called Highball [8] and associated scheduling algorithms [9]. Their early work outlines architectural considerations for a wide area network that reserves access ahead of time (or "just-intime") for bursts of data staged at the edge of the network. This proposed architecture did not include buffering within the network as is the case with SLaBS, but the possibility of staging data burstsin node controllers was considered. The proposed reservation-timedivision multiple access protocol (R-TDMA) for use in configuring crossbar switches ahead of the data burst parallels our proposed slabscheduling approaches between session layer gateways. More recently, OBS advances [10] manage bursty Internet traffic and guide the design of the next generation Optical Internet with IPover WDM. Control and management techniques for OBS have been proposed in recent years however, these approaches are geared towards specific optical switching hardware whereas SLaBS applies similar concepts via a general session-layer protocol for use over existing heterogeneous networks.

METHODOLOGY USED

SESSION LAYER work based on cloud explain how SLIM may be used to describe common communication patterns. SLIM does not guarantee the order of messages between participants in broadcast, survey and publish-subscribe patterns. Note that there is a possibility of two participants writing to the flows in aforementioned pattern sat the same time. If the application chooses to use message-oriented flows, the payloads would have de-fined boundaries. In this case, the messages may not arrive in the same order that they were generated. The session-layer does not define message boundaries in case of stream-oriented flows, as some applications may need them while others may not. The application is expected to define message boundaries if it needs to determine the source of the messages, or choose to use communication patterns such as client-server, peer-to-peer or pipeline. The session layer guarantees correctness of data exchanges between participants in that writes from different sources will not be interleaved.

Potential for New Paradigms

The use of the control channel along with support for multi-party communication has the potential for redefining communication paradigms. Consider for example, the use of load balancers. Instead of having the load balancer distribute incoming connections to replicated services (based on some criteria), a service may direct a client to authenticate itself with a dedicated service, using the public-key infrastructure. The (dedicated) authentication service can then confirm the authentication and from there onwards, the client and the service may use symmetric keys. This entire conversation would be part of the same session, where the service invites the authentication service, which then completes the assigned task and leaves. The session and flow management as well as negotiation of configuration primitives fall along the control path, where as the read and write primitives fall along the data path. Thus, once communication is setup and configured, the management and negotiation services are not involved in the communication process. This suggests that any overhead that may be observed is related to connection management and not data transfer. Since the setup cost of a SLIM session with a flow, involves instantiating two flows (one for session control and the other for a data flow), the initial cost is twice that of a TCP connection setup. All subsequent flow instantiations cost the same as setting up a TCP connection.

A transaction- oriented transport protocol should efficiently handle the following cases:

1.Transactions in which both the request and response messages t in a single packet. The response can serve as an acknowledgment, and the client handles the case of lost packets by retransmitting the original request.2. Large multi-packet request and response messages, where the data does not necessarily t in a single packet. F or instance, some systems use RPC to fetch pages of a file from a file server .A single-packet request would specify the le name, the starting position of the data desired, and the number of bytes to be read. The response may consist of several pages (e.g. 8K bytes) of data.

EXPERIMENTAL RESULT

Session layer work different application on cloud platform HOST enables a session server to provide services for many hosts, virtual or physical. Depending on the host selected, different services can be enabled. When a client issues service groupname. Assuming the session's privilege level is client and the name the client request exists, the daemon issues anarmative response and associates the session with the requested hostname. This allows the session layer to do virtual hosting at a network level. This allows organizations to centralize sessions into a small set of session servers which act much like load balancers do in existing networks.(eg figure2.)

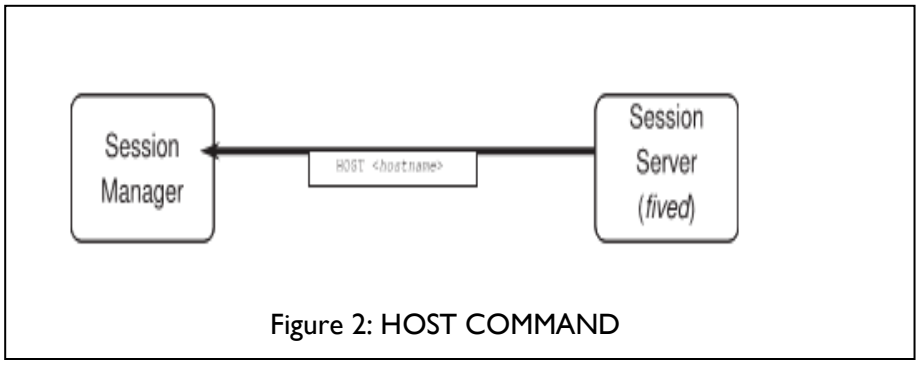
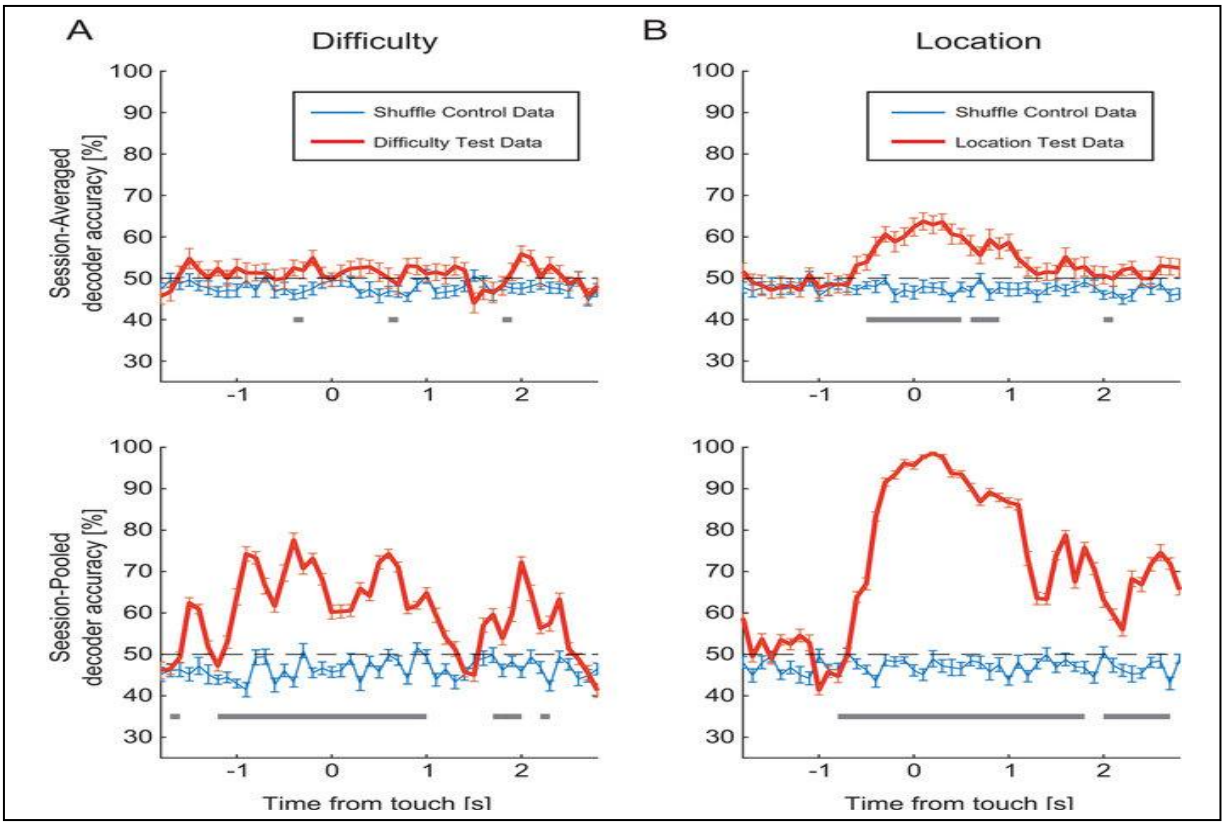


Figure 2: HOST COMMAND

And otherwise session layer DETACHABLE allows a client to disconnect from a session without destroying its state. If allowed by the server, DETACHABLE is a mechanism to request the server maintain a session's state while a client disconnects from the server for a time. This is almost a network equivalent of the UNIX screen command. The DETACHABLE service provides the user with some sort of secret. This secret could be a cryptographic certificate, a password, ASCII art or any piece of data appropriate for the security requirements of the session. When the client disconnects from the session server, the session's state persists. Data from services which remain open will be queued. The amount of time a session's state is preserved and the amount of trac it is willing to queue is up to the administrator of the session server. ATTACH allows a client to resume a previously detached session. The user provides the secret issued by a previous invocation of the DETACHABLE service along with the number of bytes they've received since the session began. After verifying the secret, the user will be allowed to resume their session. However, since the user is likely to want to start a TLS session before providing the secret to the ATTACH service

to prevent man in the middle attacks, resumed sessions will use this newTLS session, if one exists, instead of resuming an old one. (This also provides a re-key mechanism for long-standing sessions.)It is not required to break a session connection before using ATTACH on a DETACHABLE session. Instead, a user can attach another layer 4 connection to their existing layer 5 session. This allows different quality of service properties or connection bonding. DETACHABLE and ATTACH canal so be used on one specific connection, which allows users to gracefully roam networks or even physical machines. This layer service provides a lightweight distributed identity system. PROVEAUTH allows a user to use their session to prove their identity to another system. Where AUTH creates a system of authentication for the session layer, PROVEAUTH allows a user to prove that identity elsewhere. This allows users to use an identity from one entity to authenticate with another. Measuring 500 data points shows the Layer 5 Time To First Byte is larger than the Layer 4Time To First Byte, showing the expected performance degradation caused by needing to interact with a session server before being able to start an application protocol. While the difference is highly statistically significant, there is also overlap between the standard deviation of each dataset as shown on the graph. Which means many individual uses of the session layer will not be significantly distinguishable from ordinary network jitter. The story gets better when you look at time to completion. The gap between Layer 4 metrics and Layer 5 metrics narrows as more bytes are transferred across the connection, which shows the dominating performance impact of the session layer is in the initial establishment of the session. While this doesn't show up as much with a short transfer of a thousand bytes of data, the performance gap at a million bytes of data is substantially lower. For an implementation of ved entirely in user space with no kernel components and several remaining optimization opportunities, this is not a particularly bad performance picture. It seems likely that the performance of ved may be manageable. Of course, this isn't the whole Performance picture of ved. The opportunities of a session layer allow us exibility application protocols otherwise don't have. In this figure 3 (a,b) session pooled data and session averaged data in cloud application difficulty, location.



CONCLUSION AND FUTURE WORK

In this paper we presented a Session Layer, which covers several limitations of TCP for distributed applications. We propose an extensible session-layer modern networking libraries and provides the means for different cloud application. Session layer implementation SLIM enables mobility, multi-party communication, and dynamic reconfiguration of the network . We present how this layer enables variety of communication patterns. In the future, we plan to keep improving some different multi cloud application a sensible of performance in large servers.

REFERENCES

[1] V.G cerf,s.c Burleigh,A.J Hooke,L.Torerson R.C .Durst, K.L Scott K.fall and H.S weiss" RFC4828:Dely – Tolerant network architecture",http://www.ietf.org/rfc/rfc4388.txt,April 2007.

[2] S.Amstutz"Burst Switching An Introduction",Communication Magazine ,IEEE,Vol,21,No.8'983.

[3] H.song,B-w.kim and B.mukherjee," Long-reach optical access networks. A Survey of Research challenges, demonstration, and band width assignment mechanisms" communication survey and tutorials, IEEE Vol,12,no 1.2010.

[4] "GRID FTP" http://www.globus.org/datagrid/gridftp.html.

[5] "NET:netem," http://www.gridmpi.org/pspacer/index.enjsp.

[6] R.wang.X. Wu T.Wang, and T.Taleb"Experimental evaluation of delay tolerant networking (dtn) protocol for long-delay cisluncr communication"in GLOBECOM;09':PROCEEDINGS of the 28th IEEE -2009.

[7] M.Demmer and k.fall" The design and implementation of a session layer for delay –tolerant networks," comput,vol-32.16pp1724-2009.

[8] D,L mills, C.G Boncelet J.G elics.P.A Schragger. And A.W Jackson" HIGHBALL: A HIGH SPEED , reserved access "wide are networks"tech rep 90-91- sept 1990.

[9] P.Sehragger"scheduling algorithm for burst reservations on wide area high speed networks" INFOCOM ,1991 PP.589-596.

[10] C.Y.M QIAO," optical BURST Switching (obs) a new paradigm for on optical internet," Journal high speed network .vol 8.pp-69-84,1999.

