Scientific Journal of Impact Factor (SJIF): 4.72

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 4, Issue 3, March -2017

APPLICATION LAYER WORK BASED ON CLOUD COMPUTING

P.SURESH

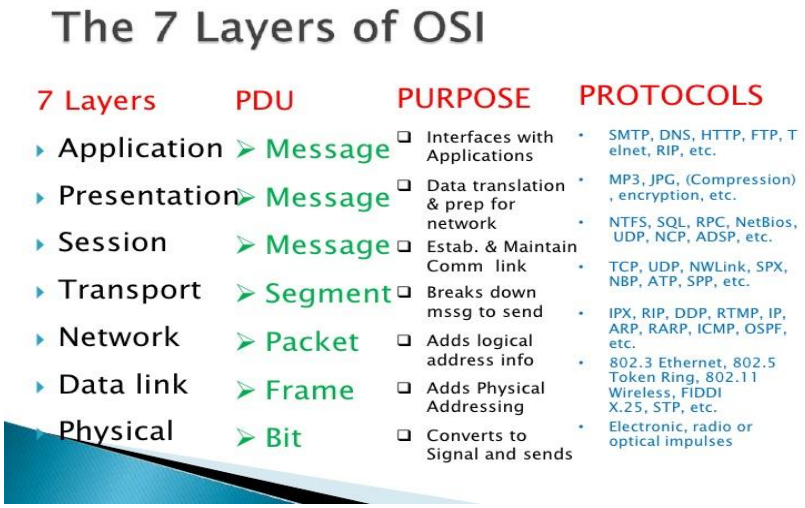
Research Scholar, Department Of Computer Science, H.H The Rajah's College (Autonomous) India

Abstract - *The cloud computing is growing rapidly for it offers on-demand computing power and capacity. The power of cloud enables dynamic scalability of applications facing various business requirements. The world experiences the Internet through the use of the World Wide Web, e-mail, and file-sharing programs. These applications, as well as others, provide the human interface to the underlying network, allowing you to send and receive information with relative ease. Most of the applications are intuitive; they can be accessed and used without the need to know how they work. As you continue to study the world of networking, it becomes more important to know how an application is able to format, transmit, and interpret messages that are sent and received across the network. Visualizing the mechanisms that enable communication across the network is made easier if you use the layered framework of the Open System Interconnection (OSI) model. The OSI model is a seven-layer model, THIS PAPER designed to help explain the flow of information from SEVENTH layer ON APPLICATION LAYER work to some help us* ***PROTOCOLS, DNS*** *and* ***API*** *(APPLICATION PROGRAM INTERFACE). There are many different types of APIs for operating systems, applications or websites.*

keywords: *API(Application program interface), cloud computing, (DNS)domain name service, (OSI) seven layer, Protocol.*

INTRODUCTION

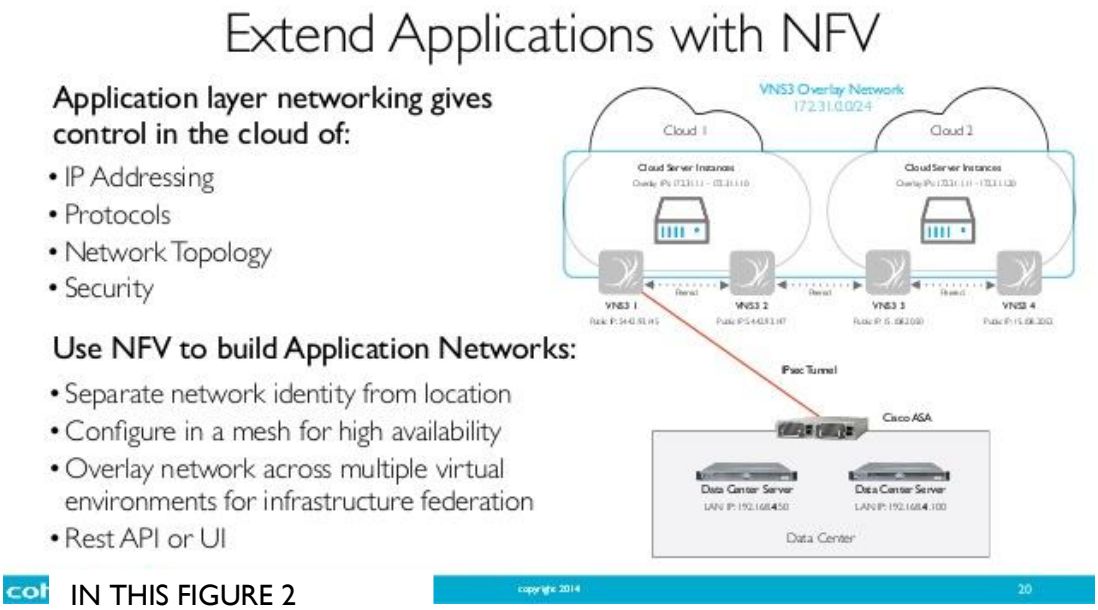
The Phrase "cloud computing"[1] describes it as a system platform or kind of software application. Cloud computing is an on-demand and cost saving computing with scalability, high-availability, and reduced management. Amazon's Elastic Compute Cloud (EC2) is an example of IaaS (Infrastructure as a Service)[2] platform. It offers basic infrastructure component such as CPUs, memory, and storage. Google App Engine is an example of PaaS (Platform as a Service) platform. In Cloud Computing platform Cloud server is a physical server. Based on IaaS and PaaS platforms, a lot of time and money have been saved for start-up companies, such as foursquare and dropbox. Cloud Computing is a Technology that uses the internet and central remote servers to maintain applications and data. .The Open System Interconnection (OSI) reference model is a principle of internetworking that you must understand to appreciate the way Cisco devices operate. The OSI reference model is a seven-layer architectural model developed by the International Organization for Standardization (ISO) and the International Telecommunications Union-Telecommunications (ITU-T). It is used universally to help individuals understand network functionality. The OSI reference model adds structure to the many complexities involved in the development of communications software. The development of communications software involves many tasks, including dealing with multiple types of applications, transmission strategies, and physical network properties. Without structure, communications software might be difficult to write, change, and support.The OSI reference model is divided into seven distinct layers. Each layer performs a specific, distinct task that helps communications systems operate. The layer operates according to a set of rules, which is called a protocol. In addition to following the rules of the protocol, each layer provides a set of services to the other layers in the model. The seven layers of the OSI reference model are the application, presentation, session, transport, network, data link, and physical layers. In the following sections, we briefly review each layer, starting with the application layer. Eg figure(1).



IN THIS FIGURE 1

The Application Layer

The application layer provides the interface to the communications system, which the user sees. Many common applications are used today in an internetwork environment, such as web browsers,FileTransferProtocol(FTP)clients,andelectronicmail.Anexampleofapplicationlayercommunication is a web browser downloading a document from a web server. The web browser and server are peer applications on the application layer that communicate directly with each other for the retrieval of the document. They are unaware of the six lower layers of the OSI reference model, which are working to produce the necessary communications. This layer networking gives control on cloud applications of figure (2).



IN THIS FIGURE 2

The Presentation Layer

The presentation layer deals with the syntax of data as it is being transferred between two communicating applications. The presentation layer provides a mechanism to convey the desired presentation of data between applications. Many people infer that the look and feel of the environment of a computer desktop, such as the way all the applications look and interact

uniformly on a computer by Apple Computer, Inc., is an example of a presentation layer. In fact, this is not a presentation layer, but a series of applications using a common programmer's interface. One common presentation layer in use today is Abstract Syntax Notation One (ASN.1), which is used by protocols such as the Simple Network Management Protocol (SNMP) to represent the structure of objects in network management databases.

The Session Layer

The session layer allows two applications to synchronize their communications and exchange data. This layer breaks the communication between two systems into dialogue units and provides major and minor synchronization points during that communication. For example, a large distributed database transaction between multiple systems might use session layer protocols to ensure that the transaction is progressing at the same rate on each system.

Transport Layer

The transport layer, Layer 4, is responsible for the transfer of data between two session layer entities. Multiple classes of transport layer protocols exist, from those that provide basic transfer mechanisms (such as unreliable services) to those that ensure that the sequence of data arriving at the destination is in the proper order, that multiplex multiple streams of data, that provide a flow control mechanism, and that ensure reliability. As you will see in the next section, some network layer protocols, called connectionless protocols, do not guarantee that the data arrives at the destination in the order in which it was sent by the source. Some transport layers handle this by sequencing the data properly before handing it to the session layer. Multiplexing of data means that the transport layer can simultaneously handle multiple streams of data (which could be from different applications) between two systems. Flow control is a mechanism that the transport layer can use to regulate the amount of data sent from the source to the destination. Transport layer protocols often add reliability to a session by having the destination system send acknowledgments back to the source system as it receives data. In this text, we discuss the three commonly used transport protocols: the Transmission Control Protocol (TCP) that is used on the Internet, Novell's Streams Packet Exchange (SPX), and Apple's AppleTalk Transport Protocol (ATP).

The Network Layer

The network layer, which routes data from one system to another, provides addressing for use on the internetwork. The Internet Protocol (IP) defines the global addressing for the Internet; Novell defines proprietary addressing for the Internetwork Packet Exchange (IPX), its client/server architecture; and Apple's AppleTalk uses the Datagram Delivery Protocol (DDP) and proprietary addressing for communicating between its machines on the network layer. In later chapters, we explore the specifics of each of these types of network layer addresses. Network layer protocols route data from the source to the destination and fall into one of two classes, connection-oriented or connectionless. Connection-oriented network layers route data in a manner similar to using a telephone. They begin communicating by placing a call or establishing a route from the source to the destination. They send data down the given route sequentially and then end the call or close the communication. Connectionless network protocols, which send data that has complete addressing information in each packet, operate like the postal system. Each letter, or packet, has a source and a destination address. Each intermediate post office, or network device, reads this addressing and makes a separate decision on how to route the data. The letter, or data, continues from one intermediate device to another until it reaches the destination. Connectionless network protocols do not guarantee that packets arrive at the destination in the same order in which they were sent. Transport protocols are responsible for the sequencing of the data into the proper order for connectionless network protocols.

The Data Link Layer

Layer 2, the data link layer, provides the connection from the physical network to the network layer, thereby enabling the reliable flow of data across the network. Ethernet, Fast Ethernet, Token Ring, Frame Relay, and Asynchronous Transfer Mode (ATM) are all Layer 2 protocols that are commonly used today. As you will see throughout this text, data link layer addressing is different from network layer addressing. Data link layer addresses are unique to each data link logical segment, while network layer addressing is used throughout the internetwork.

The Physical Layer

The first layer of the OSI reference model is the physical layer. The physical layer is concerned with the physical, electrical, and mechanical interfaces between two systems. The physical layer defines the properties of the network medium, such as fiber, twisted-pair copper, coaxial copper, satellite, and so on. Standard network interface types found on the physical layer include V.35, RS-232C, RJ-11, RJ-45, AUI, and BNC connectors.

API APPLICATION PROGRAM INTERFACE

As cloud computing continues to gain momentum, system administrators are looking for more ways to integrate with their cloud model. There are now more direct use cases for cloud computing, which require greater levels of customization. The ability to enhance the cloud experience and have cross-cloud compatibility has helped form the Cloud **API (Application Program Interface)** environment. Now, administrators can integrate applications and other workloads into the cloud using these APIs. Understanding the cloud API model isn't always easy. There are many ways to integrate into an infrastructure, and each methodology has its own underlying components. To get a better understanding of cloud computing and how APIs fit into the process, it's important to break down the conversation at a high level. There are four major areas where cloud computing will need to integrate with another platform (or even another cloud provider).PaaS APIs (Service-level): Also known as Platform-as-a-Service, these service APIs are designed to provide access and functionality for a cloud environment. This means integration with databases, messaging systems, portals, and even storage components. SaaS APIs (Application-level): These APIs are also referred to as Software-as-a-Service APIs. Their goal is to help connect the application-layer with the cloud and underlying IT infrastructure. So, CRM and ERP applications are examples of where application APIs can be used to create a cloud application extension for your environment. IaaS APIs (Infrastructure-level): Commonly referred to as Infrastructure-as-a-Service, these APIs help control specific cloud resources and their distribution. For example, the rapid provisioning or de-provisioning of cloud resources is something that an infrastructure API can help with. Furthermore, network configurations and workload (VM) management can also be an area where these APIs are used. Cloud provider and cross-platform APIs: Many environments today don't use only one cloud provider or even platform. Now, there is a need for greater cross-platform compatibility. More providers are offering generic HTTP and HTTPS API integration to allow their customers greater cloud versatility. Furthermore, cross-platform APIs allow cloud tenants the ability to access resources not just from their primary cloud provider, but from others as well. This can save a lot of time and development energy since organizations can now access the resources and workloads of different cloud providers and platforms.

Domain Name System (DNS)

DNS is a hierarchical system, based on a distributed database, that uses a hierarchy of Name Servers to resolve Internet host names into the corresponding IP addresses required for packet routing by issuing a DNS query to a name server. Name servers are usually Unix machines running the Berkeley Internet Name Domain (BIND) software. On many Unix-based machines using the sockets-API, get host by name() is the library routine that an application calls in order to issue a DNS query. Resource record: Associated with each host on the Internet, includes IP address, domain name, domain name server, etc. When resolving a host name, DNS returns the associated resource record o f the host. Internet domain names are divided into generic top-level domains (edu,com,gov, mil) which include all US domains and country domains.

Domain Name Meaning

au Australia, in India, cl Chile, fr France, us United States, za South Africa, uk United Kingdom
jp Japan, es Spain, de Germany, ca Canada, ee Estonia, hk Hong Kong.

APPLICATION PROTOCOLS

Application layer one or more then protocols to implement on networking. In thi figure (3) list of protocol.

.DHCP • DHCPv6 • DNS • FTP • HTTP • IMAP • IRC • LDAP • MGCP • NNTP • BGP • NTP • POP • RPC • RTP	• RTSP • RIP • SIP • SMTP • SNMP • SOCKS • SSH • TELNET • TLS/SSL • XMPP • (MORE)
IN THIS FIGURE:3	

The Internet Protocol (IP) is the principal communications protocol in the Internet protocol suite for relaying data grams across network boundaries. This function of routing enables internetworking, and essentially establishes the Internet. IP is the primary protocol in the Internet layer of the Internet protocol suite and has the task of delivering packets from the source host to the destination host solely based on the IP addresses. For this purpose, IPdefines data gram structures that encapsulate the data to be delivered. It also defines addressing methods that are used to label the datagram source and destination. Historically, IP was the connectionless datagram service in the original Transmission Control Program introduced byVint Cerf and Bob Kahn in 1974, the other being the connection-oriented Transmission Control Protocol (TCP). TheInternet protocol suite is therefore often referred to as TCP/IP.

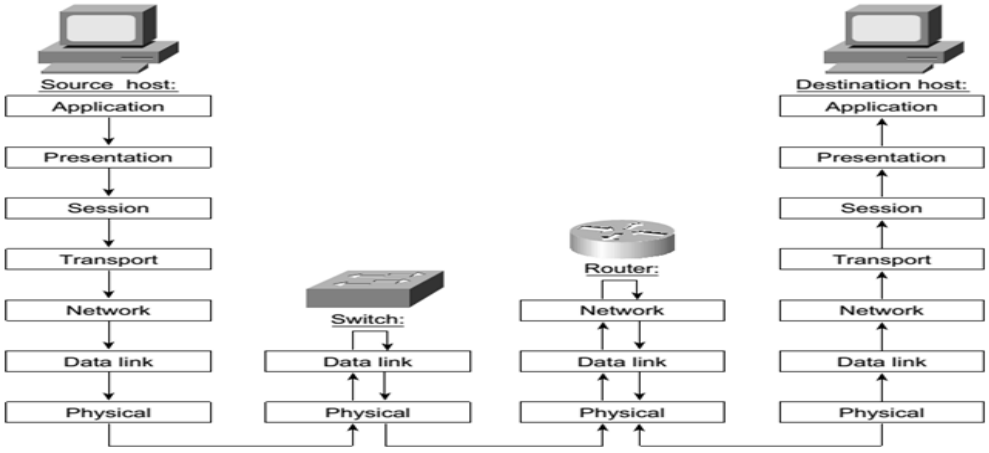
RELATED WORK

In [6], Jung June Lee et al proposed that the Constrained Application Protocol (CoAP) was developed t o support the communication between resource constrained nodes via low-power links. As an Internet protocol, CoAP needs congestion control primarily to stabilize the networking operation. They introduced a new round trip time basedadaptive congestion control scheme, which improves CoAP by utilizing the retransmission count information in estimating the retransmission timeout. An experiment is conducted based on Californium CoAP framework and real devices. The results exposed that the proposed scheme significantly improves CoAP in terms of throughput and rate of successful transaction.Nan Chen et al [7], proposed that as the number of mobile devices per user increases, the need to connect/combine them grows. Current approaches focus on the use of cloud-hosted backend services which allow file and app-state synchronization but fail in providing true resource sharing among mobile devices. To enable true resource/service sharing, the mobile devices of a single user should be combined into a cloud of cooperating mobile devices. Instead of accessing the resources/services of an individual device, a user should be able to seamlessly access the combined resources/services of his/her device cloud. Enabling seamless access to the resources/services hosted on different mobile devices is therefore a key challenge. Exposing the resources/services of each mobile devices within the user's device cloud via Restful micro-services, is one possible approach. The authors were focussed on the use of the IoT CoAP as an application layer protocol. To minimize the energy costs of communication, it was necessary to replace CoAP's standard transport protocol (UDP) with BLE 4.1. They demonstrated their performance of the CoAP protocol using BLE 4.1 on Android Lollipop.

METHODOLOGY USED

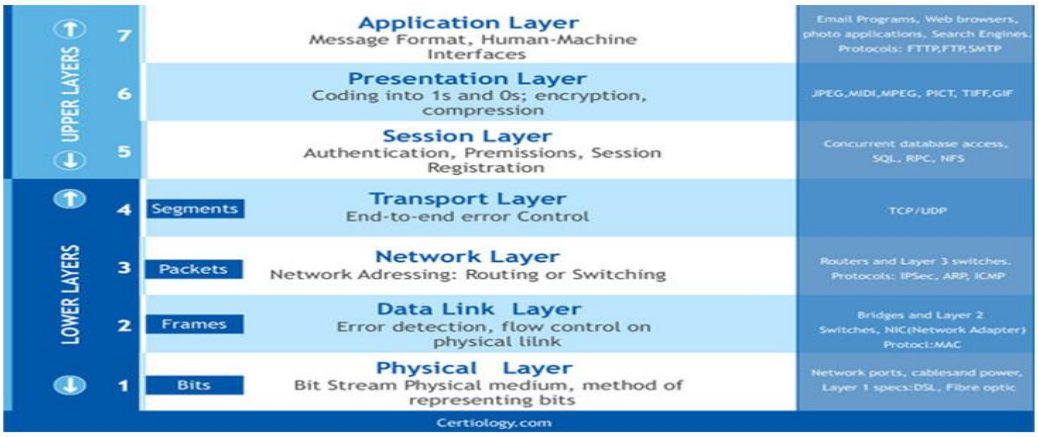
These seven layers all work together to provide a communications system. The communication occurs when a protocol on one system, which is located at a given layer of the model, communicates directly with its corresponding layer on another system. The application layer of a source system logically communicates with the application layer of the destination system.

The presentation layer of the source system passes data to the presentation layer of the destination system. This communication occurs at each of the seven layers of the model. This logical communication between corresponding layers of the protocol stack does not involve many different physical connections between the two communications systems. The information each protocol wants to send is encapsulated in the layer of protocol information beneath it. The encapsulation process produces a set of data called a packet from figure(4).



IN THISFIGURE 4

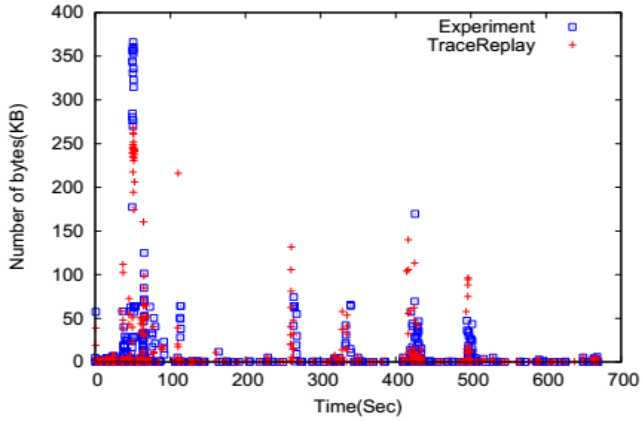
Both the source and destination devices use application layer protocols during a communication session. For the communications to be successful, the application layer protocols implemented on the source and destination host must match. Protocols perform the following tasks: Establish consistent rules for exchanging data between applications and services load on the participating devices. Specify how data inside the messages is structured and the types of messages that are sent between source and destination. These messages can be requests for services, acknowledgments, data messages, status messages, or error messages. Define message dialogues, ensuring that a message being sent is met by the expected response and that the correct services are invoked when data transfer occurs. Many different types of applications communicate across data networks. Therefore, application layer services must implement multiple protocols to provide the desired range of communication experiences. Each protocol has a specific purpose and contains the characteristics required to meet that purpose. The right protocol details in each layer must be followed so that the functions at one layer interface properly with the services in the lower layer. Applications and services can also use multiple protocols in the course of a single conversation. One protocol might specify how to establish the network connection, and another might describe the process for the data transfer when the message is passed to the next lower layer. In this figure(5) explain about some protocols implement application layer.



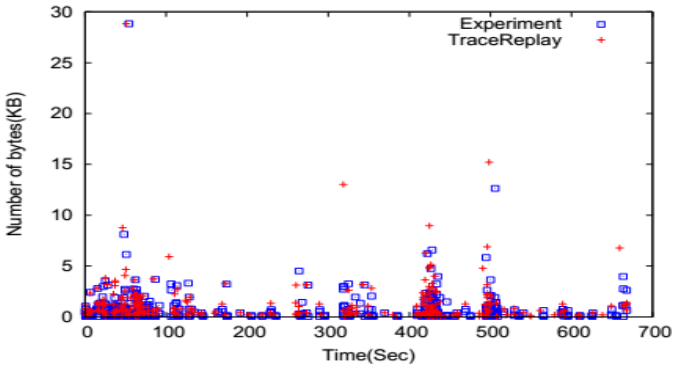
IN THIS FIGURE:5

EXPERIMENTAL RESULT

Our first experiment consists of a user downloading video lectures from a course website hosted on a local web server in our university campus. The user opens his web browser, authenticates himself, browses the website for content, and then downloads a 7 MB lecture video. Other applications like email are running in the background, and user also browses other sites for short periods of time while the download is ongoing. A network trace was collected during this entire activity of the user. This trace represents typical traffic that WiFi networks see in classrooms. The network trace consisted of mainly HTTP traffic, with a total of 56 TCP connections to various remote hosts. We now setup a simulation with a single WiFi client connected to an AP, and replay the trace on the client and the server. The bandwidth and delay of the point-to-point link from the AP to the server are set to 100 Mbps and 5ms respectively, because the actual server in the experiment was connected over the LAN to the AP. We now compare the traffic generated in the simulation by Trace Replay to the traffic seen in the actual trace, to validate the correctness of Trace Replay. Figures 6 and 7 show the upload and download traffic (aggregated over 0.1 sec intervals) generated by the client in the experiment and with Trace Replay. We find that the traffic generated in simulation matches very closely to that in the experimental trace. Further, this realistic traffic was generated in simulation with very little effort, i.e., without the user having to set any parameters for any models. run simulations comparing Trace Replay with the Bulk **Send application layer** model in ns-6, with varying number of clients. For the Trace Replay simulations, we replay the single trace collected above at all clients, with a suitable randomization parameter. For simulations with Bulk Send, we generate traffic with two Bulk Send connections, one for the upload and the other for download. \The download times generated from Trace Replay are found to be closer to what the course instructor observed(for comparable class sizes in real life), and also match results on TCP performance in large classrooms in our previous research [6].

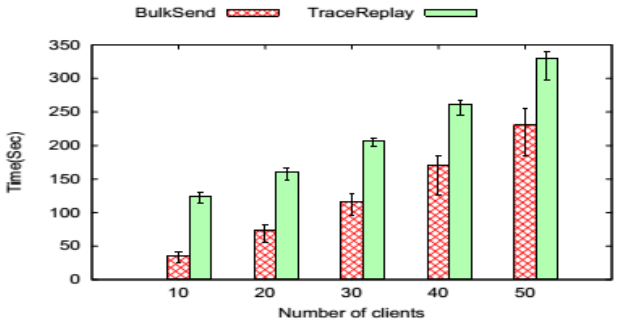


IN THIS FIGURE:6



IN THIS FIGURE FIGURE:7

Why does Trace Replay lead to higher download times as compared to Bulk send? One reason is that Trace Replay accurately captures user think times and other **application layer** delays. It also delays sending packets to preserve causal ordering of packets, much like real applications. The most important reason, however, is that the traffic generated by Trace Replay introduces significantly higher contention on the wireless channel as compared to the traffic from Bulk Send. With Bulk Send, the small amount of upload.



IN THIS FIGURE:8

Figure 8. Total download time (min, max, avg) when fetching content over HTTP from local server, for Trace Replay and Bulk Send traffic proceeds independently from the large download, and lasts for a shorter duration, unlike in the real trace. Asa result, for most part of the simulation, the contention on the wireless channel was very low. With Trace Replay, however, the upload traffic overlaps significantly with the download (much like in the actual experiment), and adds a constant "chattiness" on the wireless channel. This increases contention, leading to more wireless collisions and losses, resulting in higher download times.

CONCULSION AND FUTUREWORK

In this paper we have presented the taxonomy of performance issues of cloud Application, this layer at the time of integrating end devices of IoT. The proposed thin server architecture promotes a Web like application layer with a common programming model for constrained networked embedded devices. The application layer contains the higher-level protocols used by most applications for network communication. In this layer protocols include the one more then protocol working of network. Data coded according to application layer protocols are then encapsulated into one or(occasionally) more transport layer protocols (such as TCP or UDP), which in turn use lower layer protocols to effect actual data transfer. The TCP/IP model's application layer corresponds to the fifth, sixth, and seventh layers of the Open Systems Interconnection (OSI) model, which are (5) session layer, (6)presentation layer, and (7)application layer. The functions associated with the cloud application this layer protocols in both the OSI and the TCP/IP models enable the human network to interface with the underlying data network. When you open a web browser or an instant message window, an application is started, and the program is put into the device memory, where it is executed. Each executing program loaded on a device is referred to as a process. This paper future work multi cloud application working of application layer.

REFERENCES

1. M.Armbrust A.fox,R.Griffith etal.,"A view of cloud computing",communication of the ACM ,vol,3.no:4pp.50-68,2010.
2. S.Bhardwaj,l.jain " cloud computing:a study of infrastructure as a service (IAAS),"international journal of engineering and information technology,vol 2,no.1,pp;60-63,2010.
3. W.LU J.jackson, and R.Barga,"Azureblast, a case study of developing science application on cloud "pp.413-420.
4. Netgear prosafe xsm7224S references manual.
5. RFC 1726 section 6.2.

6. Jung June Lee:sung min chung :Byungjun lee; kyung tae kim;hee youg youn (2016) round trip time based adaptive congestion control with coap for sensor network. International conference on distributed computing in a sensor system(DCOSS) p.p 113-115.
7. Collaboration & mophile cloud computing using coap to enable resources sharing between cloud of mophile devices,ieee conference on collaboration and internet computing (CIC) p.p 119-124.
8. Bellovin, s.″ guidelines for mandating the use ip sec″, work in progress ,IETF, October 2003.
9. Bernardo, D.V And hong,D.B,," A Conceptual approach against next generation security threats: security a high speed network protocol –UDT''proc.IEEE 2nd ICFN 2010,Shanya china.
10. BernardoD.V And hoang,D.B"Network security consideration for a new generation protocol UDT..PROC,,IEEE THE 2nd ICCIST conferences 2009.beijing china.

