Broker's Communication for Service Oriented Network Architecture

AP Manu, Bhawana Rudra, Vipin Kumar and OP Vyas

Indian Institute of Information Technology, Allahabad, India
{apmanu, bhawanarudra7, vipinyadav1987, dropvyas}gmail.com

Abstract

The diversity, heterogeneity and innovation in network application domain prompted a new era of Internet. Many technologies such as ATM, Frame Relay, X.25, etc., were developed to provide alternative for communication in networked systems, but could not succeed to get evolved as one of the alternate / replacement technology for the Internet. The Internet adopted a method of patchwork approach to cope with the needs of evolution and revolution of technology growth with acceptable cost and speed. The Internet is not designed for any specific application purposes; rather it is for generic and evolves purposes. The Future Network is expected to hosts much more than today's applications in an efficient manner, but researcher predicted rigidity as one of the failure factors for Current Internet (CI). In this paper the authors' emphasis on the development of the improved Future Network using with broker's communication which works based on SOA while accounting and addressing the issues of CI failure factors.

Keywords: *SOA, Orchestration, Future Network, SONATE, FNA*

1. Introduction

In 60's, DARPA supported for successful development of network communication architecture and in 90's, same original network architecture was reused with modification to share web based content across network and researchers believed that this initiated a new era of Internet [22]. Many limitations and bottlenecks are being chronicled by researchers worldwide and efforts for improvement in technology were also being done to catch up with the dynamically evolving Internet. Some of the highlighted limitations for Internet were chronicled by David D. Clark, et al., Tim Moors, Subharthi Paul, et al., & AKARI architecture design project to name the significant ones in their successive research papers [8, 17, 20, 11]. Most of the researchers in last four decades have criticized the Internet in its limitations to meet rapidly changing requirements. David D. Clark vociferous in his large number of research papers in highlighting the *tussle in cyber space* [2, 4, 7, 8, 24]. Although the architecture of the internet is based on a number of principles including self-describing, datagram packet the end-to-end argument, diversity in technology, and global addressing [8], but David D. Clark along with J. H. Saltzer highlighted end-to-end arguments amongst the most influential of all the communication protocol design goals. However their arguments were challenged by the emergence of many developments like firewalls, caches, active networks, network address translator, multicast, network quality of service.

It is also believed by many researchers that commercialization of the Internet has given birth to any diverse interest groups in the Internet ecosystem. These diverse and contrasting set of requirements for Internet has been resulted into the so called *Tussle in Cyberspace* [8] which describes the ongoing contention among parties with conflicting interests. It is opined by some researchers that the Internet is shaped by this controlled tussle. Tim Moors [17] has also argued that the end-to-end arguments are insufficiently compelling to outweigh other criteria for some functionalities such as routing and congestion control etc., therefore the end-to-end argument should be adopted intelligently with *a great deal of information about system implementation.*

Although M Handley has chronicled many significant limitations in terms of convergence, architectural problems, and other pitfalls arising out of these issues. M Handley has given the Internet historical perspective highlighting the era of 1970-1993 as a history of change and 1993 - till publication of his paper (2006) as the *era of failures and stagnation* in his article *Why the Internet Only Just Works*. As Internet does not design for any specific purpose so the Future Network is expected to host much more than today's applications in an efficient manner. At the same time researcher predicted rigidity of Internet architecture as one of the significant failure factors for meeting these demands. In this regard, AKARI has done significant work in identifying the strengths and limitations of the current Internet and has highlighted that the architectural expectations can be stipulated as capabilities for *integrating existing communication technologies and satisfying all user communication requests*. The AKARI group has observed that the main factors contributing to the success of Internet Protocol are;

- *Aggregate lower layer technologies in the network layer, so that emerging communication technologies could be easily converged to minimize the effects on upper layers.*
- *The functions of network layer are held at a minimal level to support for new application request flexibly.*

It was also agreed by many researchers that the current Internet design principles also conforms to the KISS principle (Keep It Simple, Stupid) [13], and for the Internet, this means that, the network layer is kept as simple as possible and that services or applications are implemented at end hosts or edge nodes. Having designed its architecture using these principles, the Internet could adopt new technology while accommodating various services over the years [5]. Leading Internet researchers agreed that the simplicity of the network carries the tendency to optimize or functionally maximize one technology without considering its consistency with other technologies or other layer functions [11].

The Internet is getting aged at the same time deviating from the purpose of its original creation; as a result of this severe problem like rigidity, layering, etc., are being faced [16]. The concept of layering was introduced to give an abstract view to designer, but J. Crowcroft, et al., clearly showed in their paper that layering is harmful [9]. In the work of Stefan Leue and Philippe A. Oechslin clearly showed that the rigidity can worsen the communication performance [15]. One of the ideal ways of solving these problems could be a modification of existing protocol or changing protocol interaction or introducing a new protocol to satisfy requirements. The G-Lab initiated a project called SONATE.

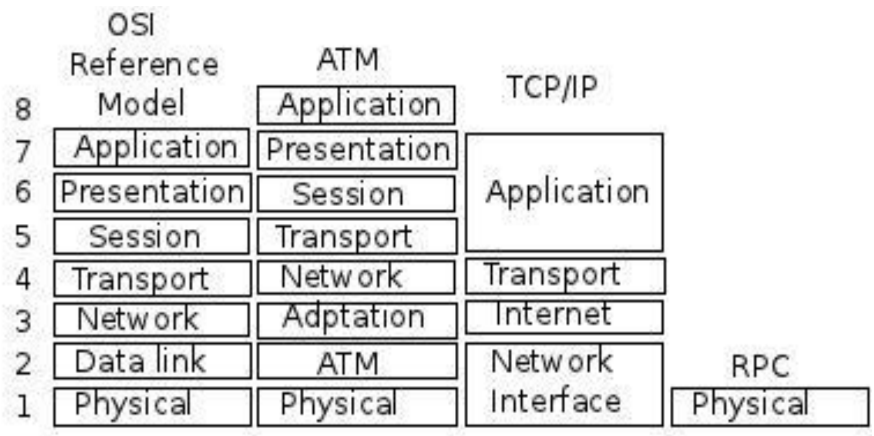

Figure 1. Layers of Different Architectures

Funding agencies such as the European Union, USA, China, Korea, Australia, Japan, India and others showed interest towards the research of networking architecture which may or may not be based on TCP/IP. Many projects from different agencies are already being started at the initiative level to cope up with the need of new technology, such as G-Lab, GENI, Internet2, National LambdaRail, FIND, AKARI, JGN2plus, FIRE, SOA4ALL to name a few.

2. Critical Review of Internet

Due to the complexity of network, computer network software's follows a hierarchy of protocols and to create a kind of communication service abstraction follows P2P communication. There is a debate in the network community that *what is the right number of layers?* since from many years [18]; the ISO community claims as seven layers, TCP/IP community argues as five, that of OS research community has implemented by just single layer (RPC protocol), X.25 reference as three, G.hn has got three etc., (Figure 1). In fact, layering results in inefficient network code and poor performance no matter how good the OS is [9]. Some of the current Internet hindrances are as follows;

- TCP/IP was originally designed and implemented for WAN's, even though it is usable on LAN. But it is not optimized for this domain. In TCP checksum is used for E2E reliability check, in spite of using CRC check per-packet basis.
- IP uses *Time-To-Live* which is relevant only in the context of WAN environment. It also supports packet fragmentation; inter network routing and reassembly features which are not useful in a LAN environment.
- TTL has the maximum number of seconds that a packet can exist within an Internet [12]. However because the field allows only integers (with values from 0 through 255), and because every node accepting a packet must take some non-negative, non-zero amount of time to process it, the TTL behaves almost exactly like a hop counter. As an 8-bit field, the maximum possible TTL is 255 [21]. Requirements for Internet hosts communication layers, suggests that the default value for new packets should be set to at least big enough for the Internet *diameter* that is the longest possible path. A reasonable value is about twice the diameter, to allow for continued Internet growth. Current figure suggests a default of 64, and which is unchanged since 1994. RFC 1122 requires that this

value be configurable on all Internet hosts so that the value may be changed as and when necessary.

- Internet will do the best what it can, but there is no standard specification of how good that is. Packet delivery may go out of order or get delayed or even drop out based on the conditions. This approach burden application designer to cope with variations and with the intention of poor service is better than no service at all [7].
- TCP/IP round-trip latency is poor, which affects the communication performance.
- The assumption of the TCP/IP model is that it communicates between autonomous machines which cooperate minimally. But machines on LAN frequently a share common file system, administrative service and user base.
- The TCP/IP protocol suite and Socket separate interfaces and protocols stack into multiple layers and the transitions can be costly in terms of effort put to program and amount of time. This focuses on message throughput rather than latency of protocol. There are only a small number of protocols (TCP/IP and UDP/IP) and interfaces (System V Transport Layer Interface & Berkeley Sockets) are in widespread usage so the generality of forming the multi-layer concept is questionable.
- The implementation of TCP/IP is complex with the memory management mechanism which reduces the system performances.
- The outer header identifies contained inner protocol header. Thus in advance, the outer protocol must be defined by a set of possible inner protocols which is a hard-coded selection. Because of this during development phase developer has to know the pool of protocols exist in the system, which makes new protocols induction almost impossible.
- The TCP/IP protocol suite was developed under the sponsorship of DARPA; despite there are a number of serious security flaws inherent in it. Some of the securities based flaws are authentication attacks, sequence number spoofing, Source address spoofing, and routing attacks, trivial attacks etc [1].
- TCP/IP assumes a fairly simple and predictable notion of the E2E communication, that is, availability of minimum one permanently functional path between source & destination with a small E2E delay and packet loss, but this does not always holds well in a dynamically varying mobile environment. Because TCP/IP has been designed for wired networks, but not for wireless network so handling data link layer in wireless media requires a different approach.
- Implementation of QoS, QoC, mobility and other sub-services come in the form of accessories and these were not considered as design issues at the time of TCP/IP architectural design.
- Simultaneous use of IP security with Quality of Service, Mobile IP and Multicast is completely infeasible in TCP/IP paradigm, and also failed in providing Simplicity, Efficiency and Trust (SET).

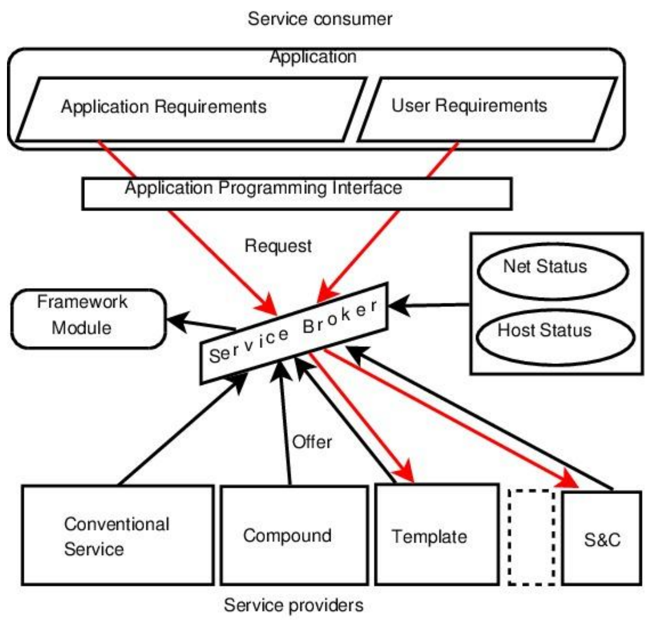

Figure 2. Architecture of Service Oriented Network

- The functionality of locating circuit switching technology of MPLS lies above link layer and below the network layer. As a result of this internet got speed up, however it duplicates many roles of network and link layer. In the same way SIP reserves resources between network providers. However if the transport layer reserves the resources then this protocol will be unnecessary. Hence this shows that there is a limitation in layer mechanism.
- The reduction in the price of the memory increased the buffer size considerably to large. A buffering provides a space for packets to wait during a transmission while helping minimization of data loss. But a problem which needs more attention is *bufferbloat* [6] and it is due to the excess of buffering inside a network resulted in defeat of the TCP's congestion avoidance mechanisms. Transmission of more packets even after reaching a choke point starts and lengthens the queue to reduce drop. The queue length drain after flood takes time and it is observed as slow response. Some of the applications such as VoIP, network gaming, chat programs etc., are working with latency constraints becomes unusable. With the increase in demand the problem likely to worsen due to bufferbloat.
- Apart from the above mentioned issues other factors which are also held equally responsible for the development of new architectures are multitude of application growth, protocols need to perform congestion control, loss recovery, no universally approved model for traffic engineering, busty nature of traffic, and so on so forth.

2.1 Design Approaches

Clean Slate, Evolutionary, Re-engineering, SOA, Economical, etc., are some of the approaches used to address the issues of a fundamental limitation of the current Internet. Based on these approaches many architectures got evolved such as the RBA [3], ANA [14], RNA [25], SILO [10], SONATE [23] etc. One of the outcomes of the Clean Slate approach is SONATE.

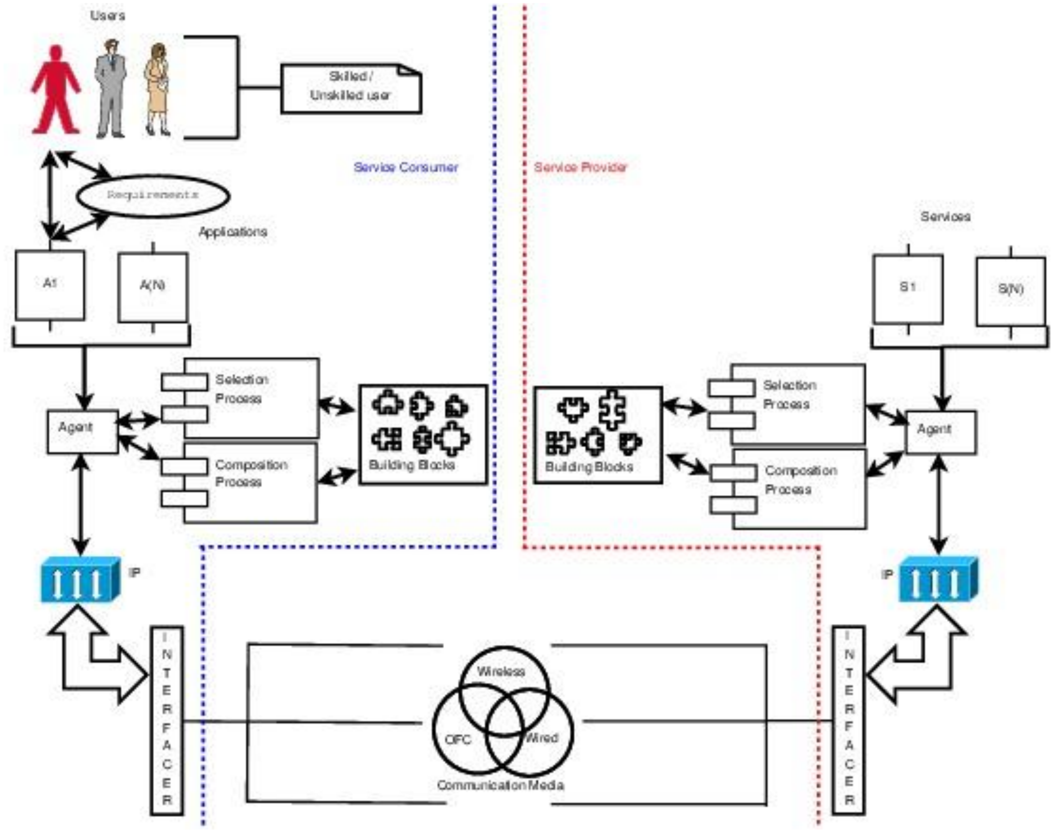

Figure 3. Flexible Network Architecture Model

Due to the gigantic size of the Internet and the practical implementation difficulty at global level hinders the developer to suggest any major changes unless and otherwise that is very essential. The roots of the Internet had been deepened over past four decades, so there is no easy practical way out for changing all available protocols or type of interaction or their functionalities. The mechanism of future Internet architecture demands for easy build up of protocols, version's of protocols, easy to add, remove or modify, automatic communication, so on and so forth. To overcome this problem SONATE provides full support of the existing Internet, treating as a backward compatibility and also allows for any upcoming changes with its flexibility.

3. SONATE

SONATE is an acronym of Service Oriented Network Architecture, which uses a paradigm of SOA to overcome some of the flaws of the current Internet. SOA paradigm utilizes services as fundamental elements for developing application / solution [19]. A service can be either *atomic or composite*. An *atomic service* is a service of unbreakable functionality. On the other hand *composite service* consists of a number of atomic services to constitute a complex Service. Services are the central design elements of SOA. The paradigm of SOA prominent in addressing the rigidity issues by providing the flexibility, which made to use in the development of flexible network architecture.

3.1 Architecture of SONATE

A fine grained protocol (micro-protocol) is known as Building Block (BB). These BBs are loosely coupled and highly cohesive in nature. Any type of services can be developed with the help of BBs such as encryption, decryption, compression, or even it may be TCP/IP. SONATE is a G-lab project, developed based on these principles. The fundamental rules of SOA are the Service Providers, the Service Consumers and the Service Brokers / Agents and it is used in the development of SONATE architecture (Figure 2).

A complete operational model of SONATE begins with the users (Figure 3). These users are categorized into skilled and unskilled, based on their technicality. According to the requirements of the user suitable applications are selected (A1 . . . A(N)) and these applications are interactive in nature. The user can fine tune the application, if capable enough to do so, or by default it follows as that of normal applications. The broker gathers need and requirements from user point of view, and at the same time it also gathers other information like its availability of BBs, network status etc. As an example, consider a requested service S(N) be made up of 3 sub services (s1, s2, & s3) and that are available at different nodes. A broker requests for a selection process to know its availability in its repository, if it is not available then that request is sent to other connected nodes.

BB for Communication: There are various BBs available for the development of services. Some of the essential BBs used for the communication by the agents are as follows:

Application BB: It is a single ported BB that provides the facility for executing workflow with other applications. The *App port* helps in providing connectivity between the application and the workflow.

Transmission BB: It performs sending of data by making it into small permissible sized chunks. The *Up port* receives data to perform chunks and *Down port* does the reverse of *Up port*.

Network BB: This is a part of SONATE framework, which connects its *data* and *addr ports* to other BBs.

3.2 Broker's Communication Model

A broker needs to perform at least following three types of functionality for the successful communication in a distributed environment.

Workflow Generator - A very complex service can be made by the combination of simple services. A definite combination pattern / sequence are called workflow. The workflow description specifies the definite interaction pattern required for getting a specific service of a consumer. Each node should have the capability for generating workflow.

The creation of Workflow is done by service Selection & Composition algorithm. The application requirements and network constraints are used for the dynamic creation of Workflow. The dynamic information such as how to combine? What they do? And etc., are addressed with the help of these BB descriptions.

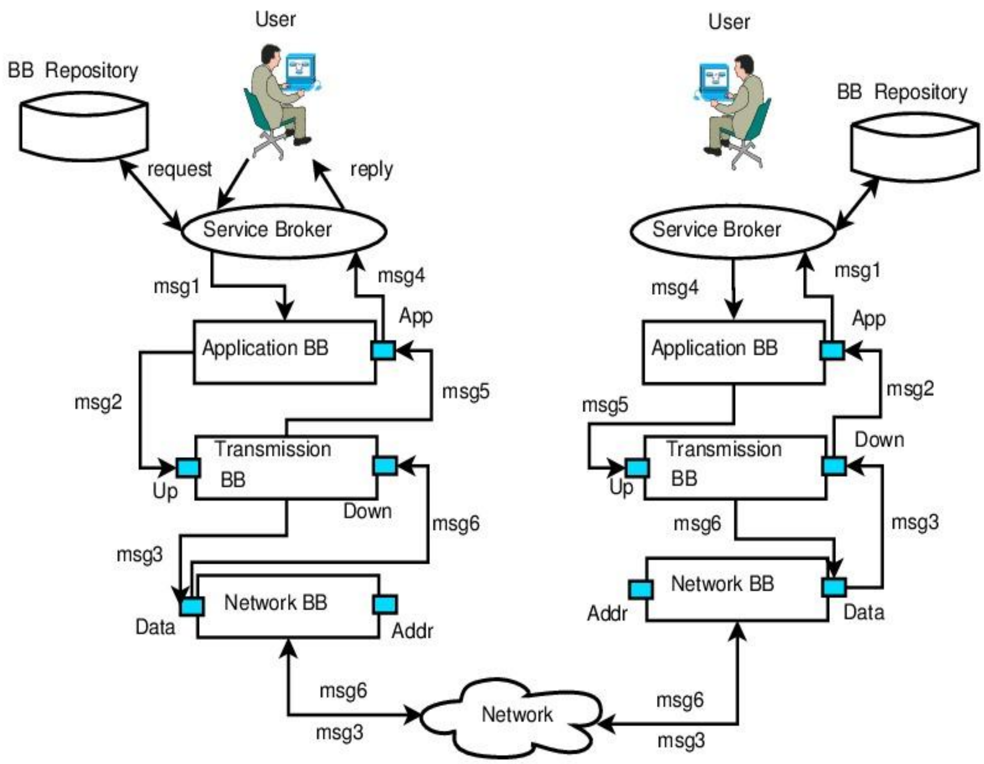

Figure 4. Broker's Communication Model

Message Translator - There are some messages which need to be composed of more than one message and which is done using a mesageList (with the use of TLV formats). A BB can create / add messages or reads / removes messages from a received mesageList to make compatible. Transformation of messages is done with the help of special BB, that is, an application BB which bridges the gap between application and the workflow.

Message Transmission - The intra and inter message transmission involves Application BB and Network BBs respectively.

3.2.1 Message Formats for Communication: Following abbreviations and its corresponding information / messages are used in explaining inter and intra communication of SONATE system.

msg1 & msg4: A XML file name in string format.
msg2 & msg5: It is *msg1* built in mesageList format.
msg3 & msg6: A tagged message with file name and fragmented data to be transferred.

3.3 Communication Broker's Operation

After getting a request from the user, the agent creates a list consisting of unavailable BBs and which will be in XML format. This information is sent to *Application BB* as *msg1* (Figure 4). The Application BB sends *msg2* to up port of Transmission BB after the translation. *Transmission BB* sends *msg3* to Data port of *Network BB* after making a data chunk. So received *msg3* by *Network BB* will be communicated to the other *Network BB* of the network with the same message format. At the receiving side *Network BB* sends *msg3* to down port of *Transmission BB.* The data extracted from *Transmission BB* will be copied in the form of XML file and the above procedure is repeated until all the packets get over. And the above mentioned procedure is repeated for a return message with appropriate message formats.

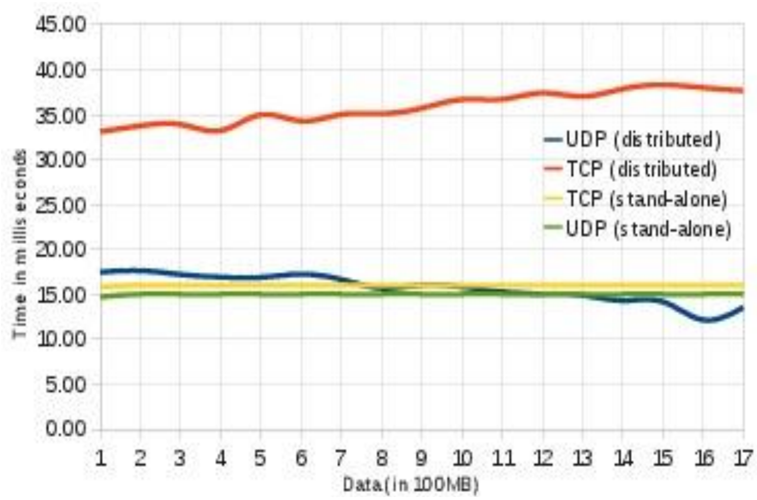

Figure 5. Comparison of Data Transfer Time for Different Cases

3.4 Results and Discussion

The required building blocks for broker communication such as Application BB, Transmission BB, Network BB, etc., are designed and developed in the lab. These building blocks are incorporated into the existing SONATE framework to test the feasibility of the broker's communication concept practically by considering a real implementation scenario.

Two broad scenarios are considered for the conduction of experiments such as stand-alone and distributed systems. Each of these scenarios is tested for TCP and UDP cases. Based on sending and receiving the data different experimentations are conducted successfully. The verification and validation of the data are done for different types of application data started with a simple & small text file, to very big files like an audio clip, video clip, image files, compressed files etc.

The experiment is conducted by considering all suitable and necessary conditions for the successful delivery of data and an inherent delay of some millisecond is purposefully introduced during packet transmission to avoid the congestion problem. The experimental result shows that in the case of TCP and UDP the time taken for stand-alone system is almost constant (Figure 5). Due to the process of necessary acknowledgment the time taken for the TCP (distributed) system is considerably larger than that of UDP.

3.5 Module Implemented Hitherto

Cyclical Redundancy Checking (CRC): The CRC is a technique used for checking errors. It calculates a numeric value to detect errors in data transmitted. Sender calculates a Frame Check Sequence (FCS) and appends it to outgoing message. Receiver recalculates the FCS, and compares the value with the received FCS. If there is any difference then the receiver re-sends the same.

Retransmission: This module provides the retransmission functionality for SONATE framework. It will be triggered if the system request for the retransmission functionality.

Multicast: This module provides the multicast functionality for SONATE framework, where Service primitives of multicast are *connected, disconnect, join, leave, send and receive*. *Connect* forms the group members, *join* and *leave* used dynamically for joining a group and leaving a group respectively. The *disconnect* is used to terminate the group, *send* is used to send messages similarly *receive* is to receive messages. When an application requests SONATE framework for transfer of packets over the network a sender multicast agent (SMA) is created at the sender side. This SMA forms a spanning tree for MAs using multicast IDs of nearest multicast agent and algorithm. This ties all MAs logically with spanning tree structure. Now MA is ready to transfer data/packet to its PMA (Parent MA) as well as CMAs (child MAs). Group multicast may contain multiple senders and multiple receivers which sends or receives data between each other. Network independent, application level multicast schemes over service based network architecture can be deployed.

Multiple Workflows: In this each coming messages is mapped to a list to get a *flowID* and *connectionID*. These IDs are used to communicate through Mux and Dmux in order to receive/send the information. It uses *Up port* and *Down port* during communication where the *Up port* receives a message list from workflows and appends *flowID* correspond to *connectionID*. *Down port* receives message list form network and retrieves the *flowID* form the message list to communicate with the network.

These working modules are successfully implemented and tested for SONATE framework and lot more such type of service providing modules are required to be developed in the future.

4. Conclusion

The problems associated with the current Internet is thoroughly analyzed and provided a new flexible structure to the Internet. The new improved technology serves as a service. And a service is a fine grained loosely coupled and highly cohesive BBs. The flexibility for long term and a short term services are provided with the concept of SOA. The self organizing nature of BBs made the system flexible and can develop the services of simplex to very complex types in a distributed environment to support dynamic, distributed, and heterogeneous conditions. A broker's communication model for Service Oriented Network is developed, tested and found working satisfactorily for different cases.

Acknowledgements

Wholeheartedly author acknowledges German-Lab (G-LAB) for kind cooperation and extended support.

References

[1] S. M. Bellovin, "Security Problems in the TCP/IP Protocol Suite", ACM SIGCOMM Computer Communication Review archive, vol. 19, no. 2, **(1989)** April, pp. 32-48.
[2] M. S. Blumenthal and D. D. Clark, "Rethinking the Design of the Internet: the End-to-End Arguments vs. the Brave New World", ACM Trans. Internet Technol., vol. 1, no. 1, **(2001)** August, 70-109, http://doi.acm.org/10.1145/ 383034.383037.

[3] R. Braden, T. Faber and M. Handley, "From Protocol Stack to Protocol Heap: Role-Based Architecture", SIGCOMM Comput. Commun. Rev., vol. 33, no. 1, **(2003)** January, 17- 22, http://doi.acm.org/ 10.1145/774763.774765.
[4] I. Brown, D. D. Clark and D. Trossen, "Should Specific Values be Embedded in the Internet Architecture?", In Re-architecting the Internet Workshop, ReARCH'10, **(2010)**, pp. 10:1-10:6, New York, NY, USA, ACM.
[5] R. Bush and D. Meyer, "Some Internet Architectural Guidelines and Philosophy", RFC 3439 (Informational), **(2002)** December, http://www.ietf.org/rfc/rfc3439.txt.
[6] V. Cerf, V. Jacobson, N. Weaver and J. Gettys, "Butter Bloat: What's Wrong with the Internet?", ACM Queue, vol. 9, no. 12, **(2012)**, pp. 10.
[7] D. Clark, "The Design Philosophy of the DARPA Internet Protocols", SIGCOMM Comput. Commun. Rev., vol. 18, no. 4, **(1988)**, August, pp. 106 – 114, http://doi.acm.org/10.1145/52325.52336.
[8] D. D. Clark, J. Wroclawski, K. R. Sollins and R. Braden, "Tussle in Cyberspace: Defining Tomorrow's Internet", SIGCOMM Comput. Commun. Rev., vol. 32, no. 4, **(2002)** August, pp. 347-356, http://doi.acm.org/10.1145/964725.633059.
[9] J. Crowcroft, I. Wakeman, Z. Wang and D. Sirovica, "Is Layering, Harmful? (Remote Procedure Call)", Network, IEEE, vol. 6, no. 1, **(1992)** January, pp. 20 -24.
[10] R. Dutta, G. N. Rouskas, I. Baldine, A. Bragg and D. Stevenson, "The SILO Architecture for Services Integration, controL, and Optimization for the Future Internet", In IEEE International Conference on Communications, **(2007)** June.
[11] AKARI-Online, http://akari-project.nict.go.jp/eng/concept-design/AKARI_fulltext_e_preliminary_ver2.pdf. **(2008)**.
[12] Internet Engineering Task Force, RFC 791 Internet Protocol - DARPA Internet Programm, Protocol Specification, **(1981)** September, http://tools.ietf.org/html/rfc791.
[13] D. S. Isenberg, "The Rise of the Stupid Network", Computer Telephony, **(1997)**, pp. 16-26.
[14] C. Jelger, C. Tschudin, S. Schmid and G. Leduc, "Basic Abstractions for an Autonomic Network Architecture", In IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, **(2007)**, pp. 1-6, IEEE.
[15] S. Leue and P. A. Oechslin, "On Parallelizing and Optimizing the Implementation of Communication Protocols", IEEE/ACM Trans. Netw., vol. 4, no. 1, **(1996)** February, pp. 55-70, http://dx.doi.org/10.1109/90.503762.
[16] A. P. Manu, B. Rudra, B. Reuther and O. P. Vyas, "Design and Implementation Issues of Flexible Network Architecture", In International Conference on Computational Intelligence and Communication Networks (CICN), **(2011)** October, pp. 283-288.
[17] T. Moors, "A Critical Review of End-to-End Arguments in System Design", In Communications, 2002, ICC 2002, IEEE International Conference on, vol. 2, **(2002)**, pp. 1214-1219.
[18] S. W. O'Malley and L. L. Peterson, "A Dynamic Network Architecture", ACM Trans. Computer Syst., vol. 10, no. 2, **(2008)** May, pp. 594-606.
[19] M. P. Papazoglou, P. Traverso, S. Dustdar and F. Leymann, "Service-Oriented Computing: State of the Art and Research Challenges", IEEE Computer Society, vol. 40, no. 11, **(2007)** November, pp. 38-45.
[20] S. Paul, J. Pan and R. Jain, "Architectures for the Future Networks and the Next Generation Internet: A Survey", Computer Communications, vol. 34, no. 1, **(2011)**, pp. 2-42.
[21] V. Paxson, M. Allman, J. Chu and M. Sargent, "Computing TCP's Retransmission Timer", RFC 6298 (Proposed Standard), **(2011)** June, http://www.ietf.org/rfc/rfc6298.txt.
[22] J. Postel, "Transmission Control Protocol", RFC 793 (Standard), **(1981)** September, Updated by RFCs 1122, 3168, 6093, 6528.
[23] B. Reuther and D. Henrici, "A Model for Service-oriented Communication Systems", J. Syst. Archit., vol. 54, no. 6, **(2008)** June, pp. 594-606, http://dx.doi.org/10.1016/j.sysarc.2007.12.001.
[24] J. H. Saltzer, D. P. Reed and D. D. Clark, "End-to-End Arguments in System Design", ACM Trans. Comput. Syst., vol. 2, no. 4, **(1984)** November, pp. 277-288, http://doi.acm.org/10.1145/357401.357402.
[25] L. Vlker, D. Martin, C. Werle, M. Zitterbart and I. E. Khayat, "Selecting Concurrent Network Architectures at Runtime", In Proceeding of IEEE International Conference on Communication 2009, **(2009)**, pp. 2124-2128.