ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

A Distributed Data Sharing and Co-Operative Caching for Peer-To-Peer Networks in Internet of Things

[1]S. Sivakumar and [2]S. Anbu

[1]*Research Scholar, Department of Computer Science and Engineering, St.Peter's University, Avadi, Chennai, Tamilnadu, India.*
[2]*Professor, Department of Computer Science and Engineering, St.Peter's College of Engineering and Technology, Avadi, Chennai, Tamilnadu, India.*

ARTICLE INFO

Article history:
Received 23 June 2015
Accepted 25 August 2015
Available online 2 September 2015

Keywords:
Internet of Things, Peer-to-Peer Networks, Caching, Load balancing, Network Traffic Management.

ABSTRACT

Distributed networks comprises of overlaid Peer-to-Peer networks. There are two kinds of topologies are applied for maintaining the Peer-to-Peer networks, They are: (i). Structured P2P network, where the peers are connected in a regular topology. (ii). Unstructured P2P networks, where the peers are connected arbitrarily. In the structured P2P networks, the networks can provide better efficiency and more accurate services but maintenance of the regular topology needs more efforts, time and more cost. However, unstructured P2P networks are tremendously strong to the recurrent peer joining and leaving but it is commonly achieved at the expense of efficiency. The main objective of this paper is to design a P2P system which is dynamically overlaid, distributed data sharing and load balancing to improve the efficiency of traffic management of IOT-[Internet of Things]. Also a cooperative caching method is applied to P2P system to increase the performance.

© 2015 AENSI Publisher All rights reserved.

To Cite This Article: S. Sivakumar and S. Anbu., A Distributed Data Sharing and Co-operative Caching for Peer to Peer Networks in Internet of Things. ***Aust. J. Basic & Appl. Sci.,*** *9(27): 287-290, 2015*

INTRODUCTION

Peer-to-Peer (aka) P2P is a network model which adopts a different architecture other than that defined in conventional client-server network architecture (Vu, Q.H., *et al*, 2010). P2P Networks do not have a common server for every system connected in the network whereas it acts both client as well as server in the network. The system connected in the network is called as peer. In this type of network the peer also raises a request to other peers connected to the network and also responds to the request s raised by other peers in the system. When the peer rises a request it acts as a client and when responding to a request it acts as a server. The peers are able to do both the processes simultaneously. Whereas in conventional network architecture a client can act only as a client and can only request the server, the server can only respond to the request from the client. In P2P networks the performance of the network is dependent on the amount of peers connected since the network is decentralized and do not have a dedicated server. With less number of peers connected the increase in requests could blow the network performance due to scarcity of peers acting as servers. A P2P network will be more efficient whenever there is more number of peers connected to the network.

The peers connected in the network align themselves into groups in order to communicate themselves. They transfer data between themselves and also allocate bandwidth for the transactions. In a P2P network a peer can upload and download data simultaneously. A peer can also leave or join the network at any time. Another distinct characteristic of P2P network is the transparency among groups. The operations of a group are visible even to the end-users of the network. P2P networks also possess a good fault tolerance rate. Whenever there is a failure of a peer or a peer suddenly absconded from the network the P2P network continues to function normally by using resources from other peers. The best way to understand a P2P network is studying the nature of a torrent system where a system which downloads a file through torrent will also act as a server and uploads and downloads data simultaneously. Whenever there is failure of a peer the client system downloads the parts from other peers where it is available. When comparing to the conventional client-server architecture the P2P networks are highly fault tolerant since they are decentralized and have no dedicated server. There are two kinds of problems can meet in network: (1) dynamic P2P traffic management is hard by ISP

Corresponding Author: S. Sivakumar, Research Scholar, Department of Computer Science and Engineering, St.Peter's University, Avadi, Chennai, Tamilnadu, India.
E-mail: drssivakumar74@gmail.com

(Aggarwal, V., 2007; Xie, H.Y., 2008) and (2) In P2P network the traffic often cross the boundaries more times (Aggarwal, V., 2004; Karagiannis, T., 2005). To find out a best solution in P2P traffic management a distributed file sharing method is proposed here.

Traffic In P2P:

The peers in a P2P network are connected as a group where they exchange data between themselves without any common server. The ability of peers to act as both client and servers simultaneously is considered a very innovative characteristic in networking field. Also the ability of nodes to find the required content from the network makes the peers in P2P network the most utilized. Once a peer is connected to the network the resources available in the node are fully utilized by the other peers. The peers basically find the required data in the network by using the centralized indexing service which is maintained throughout the network for the parts of files available in peers. In general the applications use P2P networking architecture such as torrents use a large number of peers for downloading a file even though there is availability of file parts from few peers in order to increase the bandwidth so that they can finish the process as soon as they can. The process of increasing the reach of certain peers in the network causes dissimilarity between the peers. Since the P2P application is decentralized and dependent on the peer availability the problem cannot be addressed by a single solution. The technical advancements in P2P networking pave way for many future applications of internet such as Internet of Things (IOT) which is discussed below.

Internet Of Things:

IOT – Internet of Things is a concept in which each and every system in the world is connected to each other over an IP address. IOT concept is closing towards reality with invention of applications such as cloud computing and Peer-to-Peer networking. The concept of interconnection of systems all over the world will be very useful for applications such as data analysis for the service providers, data transfer etc. The term IOT was coined in 1999 by a member of Radio Frequency Identification development community. By the recent innovative inventions there are so many visions about IOT in recent times. The “things” mentioned in the term may denote anything that can be studied about an individual person or system connected to the network such as their location, Time, etc. According to CISCO the number of devices connected to the network was around 12.5 billion in 2010 and is expected to be 50 billion in 2020.

Background Study:

According to (Krishna P. Gummadi,) the peers being able to share the most indexing functionality are the major factor of performance in a peer to peer network since the resources are only available in peers. There are three ways used by a peer in a P2P network to find required resource in a network. The conventional method is using a centralized indexing service which has details of resources available in all the peers connected to the network. This method is explained by an example in (Napster. http://www.napster.com). The next method used by the peers is broadcasting the request all over the network; this method is used in Gnutella. The third method is a hybrid method in which the centralized indexing is done by a special node in the network which contains index of resources in every node. This method is used in Kaaza. The most important factor in P2P networking is the traffic in the network. There are several works in the past that discussed about reducing traffic in the P2P network. According to the P2P traffic is responsible for over half of the overall traffic in ISP networks. Also some research states that the P2P traffic constitutes 80-90 percent of local traffic and 40-60 percent of backbone traffic as in (Jorma Mellin, 2004). The interesting fact about these analytics is that only a less number of users use P2P networking architecture. Therefore the small number of users uses a large bandwidth in the overall network. According to (Jorma Mellin, 2003), about 5 percent of subscribers use 50 to 60 percent of bandwidth available in an ISP network. This causes inequality among service provided to the customer.

Since most of the P2P applications are used to transfer illegal or copyrighted data between each other, the blocking of all the P2P is an easy solution for avoiding P2P traffic. But the problem is not all the data transfer is illegal by nature. Data such as open source applications, copyright free data can be transferred through P2P networks since they are shareable. Even the software developers use these P2P based application platform to release their applications. Recently a latest release of Redhat Linux is released in Bit Torrent. Bit Torrent is a torrent application which uses P2P architecture for communication. Since the port used by a P2P application cannot be identified the blocking of a peer is much complex than in other type of networks. Some applications use ports reserved for other services like HTTP (80) and IP sec port. According to (Alex Goldman, 2004) since they use these ports reserved to other applications they cannot be easily blocked. The Layer 7 Inspection is found commonly in most of the methods proposed for reducing traffic in P2P networks. This method identifies many P2P applications using the header information which is extracted from the payload section. By using Layer 7 inspection the traffic can be reduced and used bandwidth usage can be reduced. According to this method tracks every conversation of a P2P application by which the traffic can be controlled at any instant.

Proposed Approach:

From the above discussion about the existing methodologies about the P2P traffic management in internet or network of network, it is clear that still there is a need of a complete solution to manage the traffic in internet by concentrating on P2P traffic. It makes the present internet as the smart internet which is scalable and reliable in terms of traffic growth. The traffic depends on the number of people using internet, location of the objects, and time information about the objects and conditions of the objects. These things are the real entities involved in internet of things. Our paper concentrates on delivering things in distributed manner parallel as well as distributedly in fast manner. The entire work consists of

- sharing the data as distributed
- Managing the cache

IOT consists of different suite of technologies supporting internet of things. It serves to illustrate how various technologies relate to each other and to communicate the scalability, modularity and configuration of IOT deployments in different scenarios.

Distributed Data Sharing:

It is considered that a network of network called as transit network is comprises of numerous sub-networks. Each sub-network is connected to the core network at least by one node in the transit network. The transit network is called as T-NET, which is a structured P2P network and the peers are arranged in a ring topology. The peers in the T-NET are called as T-Peers assigned with a unique ID [PID]. Each peers are connected two other peers in both of its sides. The sub-networks comprised in to T-NET is termed as S-NET and the peers are called as S-Peers. T-NET is structured and S-NET is Un-structured one. In the basic structure of a P2P network the peers are connected to each other in a decentralized manner. The connected peers are able to act both as server as well as client in the network. Each of the S-NET is connected to T-NET by a T-Peer. The topological formation of the proposed approach is shown in Figure-1.

Whenever the S-NET is used for service accommodation the T-NET will cooperate for providing efficient and accurate service. The main advantage in this approach is the peers in the total network can communicate directly to any T-NET or S-NET. Also the second advantage is the peers can join or leave dynamically.

When a query comes from any S-NET to T-NET the request is forwarded the centralized servers in all the S-NET and T-NET, and any S-NET can provides an appropriate answer for the query. If the query is for a file or data it can be passed to the central location from there any peers can download the answer data and it is shown in Figure-1.

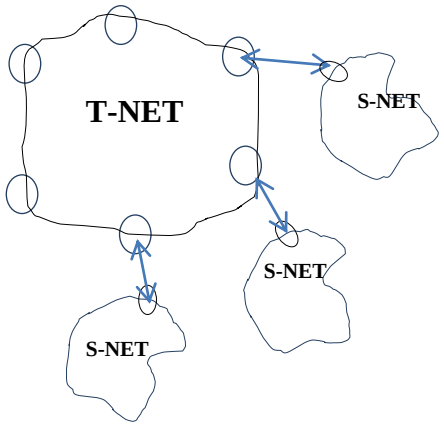
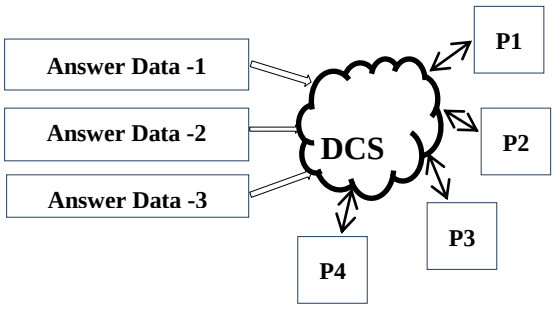


Fig. 1: Proposed Distributed P2P Network for IOT.



DCS: Distributed Central Server; **P1, P2, P3 and P4**: Peers

Fig. 2: Distributed Data Sharing.

The distributed data sharing system effectively reduces the topology of the T-NET and S-NET for reducing the complexity. Whenever a query enters into the network, the answer data will be transferred into the central server from there all the peers can download the data speedily and efficiently without any data loss. This is due to the new peers can join in the corner of the networks dynamically. A new peer entry increases the size of the network and distance among the peers and then with the BS. The new entered peer details are updated immediately in the network.

In this paper, it focused that the P2P system is the distributed data sharing. Each data is represented by a data-ID, peers are represented by a P-ID are unique to the entire network. So it is easy to identify the files and peers very speedily. Same request based peers are formed into an overlay network and this network can download the data which is very close to the centralized server. The overlaid network can be formed by computing the distance among the peers. Peers which are closer are formed as overlay network for sharing the same file speedily. Peers can join with intimation and leave without any intimation.

Cooperative Caching:

Another feature used in this paper is cooperative caching. Here the caches are deployed in the S-NET as well as in the T-NET. This cooperative caching is suitable for large size of internet service providers with more access points and exit point. Several numbers of ISPs are connected through more number of points of presence are internally connected with the high-speed optical links. All the peers can get the POPs from the ISPs [2]. Generally caching applied in P2P network to improve the performance in terms of reducing burden of P2P traffic. The cache software not allows modifying the software in P2P client system.

The main objective of caching applied in P2P traffic is to reduce the load and the operational cost. In this paper cooperative caching is applied for reducing the traffic by increasing the data hit rate. The hit rate is defined as the number of objects served from the cache to the total number of objects transferred. Our cooperative caching always maintains the internal links among the peers in the network. Simultaneously a smaller number of traffic can be allowed to among fewer ISP links which can able to reduce the overload and congestion. The entire functionality of the proposed approach is given in the form of algorithm to implement and verify in various algorithms.

Proposed Algorithm ( )

{

Initialization:

$G: T-NET;\ Ni: ith\ S-NET;\ k: number\ of\ S-NETs$

Procedure:

1. G = {{ N_1, N_2, N_3, … , N_k}, { P1, P2, P3, …., Pm}}
2. k is the number of S-NET
3. for I = 1 to k
4. join(Ni, G) // Network **Ni** is joining into the whole network **G**
5. end i
6. for J=1 to m
7. for I = 1 to k
8. join(Pj, Ni) // $Peer_i$ is joining into the network Ni
9. end j
10. for I = 1 to n
11. leave(Pi, Ni) // $Peer_i$ can leave from network i
12. end i
13. If (load >δ) then
14. $Peer_i \leftrightarrow Peer_j$ // calls the cache between peer-i and peer-j
15. end if
16. For I = 1 to R // number of requests
17. If (REQi == REQi+1) then count++; // peers sending same request
18. create overlay network for count number of peers
19. end if
20. response-DATA -> upload(common server) // data distributed
21. Pi(DATA) <- download(common server)
22. end i

}

The above proposed algorithm describes the overall functionality of the paper. The T-NET is represented by G and the S-NETs are represented by N. Each peer is represented by Pi. Peers can dynamically join in the network and can leave from the network. When the request comes from various peers the requests are analyzed for same requests. The numbers of peers sending request for same peers are connected into an overlay network [16] for sharing the distributed data among themselves to increase the downloading time. The overlay network is a sub network formed by interconnecting the nearest peers sending same request. This can download the data from the centralized server for improving the efficiency in terms of reduced downloading time and increased speed.

Simulation and Result:

The proposed algorithm is implemented in JAVA and experimentally verified by adding some number of nodes [as peers] in the T-NET. At the same time a few number of S-NET is also created and deployed in the network by interconnecting one peer in S-NET with one peer in T-NET. In out experiment, the nodes are treated as computers inter connected using a Real-Wire WiFi internet through a modem automatically. Some of the nodes are interconnected in LAN cables and connected into WiFi modem manually. Systems connected through

LAN cables are treated as peers in S-NET and the systems connected in WiFi are treated as peers in T-NET. In our experiment the number of peers deployed in the network is 100, 200, 300, 400 and 500 in various rounds. The following figures show the experimental results of the proposed approach.

One of the functionality of our proposed approach is verified in each step such as peer joining, peer leaving, request sending, centralized data sharing and time taken to process the requests. The number of peers joining and leaving in S-NET or in T-NET is un-defined and dynamic.

Figure-3 depicts the number of peers joining is increasing in terms of number of peers deployed in the network. But the number of peers leaving is zigzag. From this it is clear that the number of peers needs the internet facility is increasing rapidly.

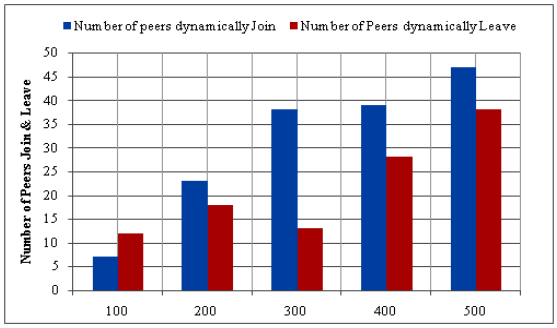


Fig. 3: Number of Peers Join and Leave Dynamically.

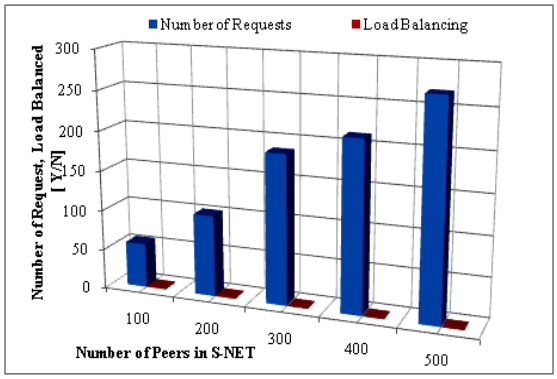


Fig. 4: Number of Request vs. Load Balancing.

The number of request to the server is increases according to the peers. When the request size increases, it is essential to fulfill the requests by load balancing or not? Figure-5 shows the number of requests and the requests can be balanced or not. From figure-5, it is clear that if the number of requests greater then load balancing is not effective and it is result less. If the numbers of request are lesser than the 50% of the number of nodes deployed, then the requests can be balanced. Another efficiency of the proposed approach is completion time for each request.

It is well known that a web request takes a few seconds to few minutes to complete the request. In our experiment, the time taken to complete the number of requests is between 1hr 30 minutes into 3 hours 10 minutes for 56 requests into 218 requests respectively. Figure-5 depicts the number of requests and time taken for completing the requests.

The efficiency can also be verified by investigating the CPU utilization when a peer connected in S-NET as well as in T-NET. Number of requests is proportionally increases according to the number of peers whereas the CPU utilization is increases according to the number of peers. Since the peers connected in network should be cooperate with other systems and the data transmission anomaly utilizes the other intermediate systems in the network. Whether the peers can process the requests or not, due to participation a minimum amount of CPU is utilized and it increased according to the number of peers. From Figure-3 to Figure-6, the efficiency of the proposed approach is proved that our distributed data sharing and load balancing with cooperative caching can provide better efficiency in terms of CPU utilization, request compilation time, load balancing and peers joining and leaving from the network.

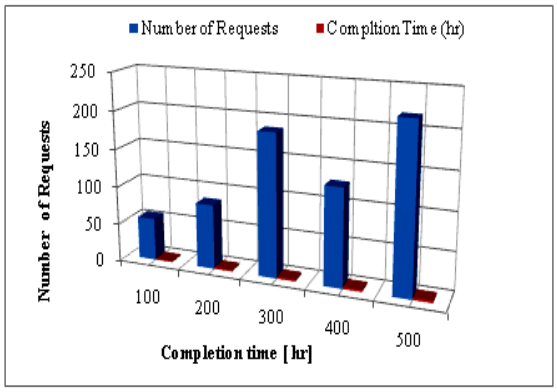


Fig. 5: Number of Request vs. Completion Time.

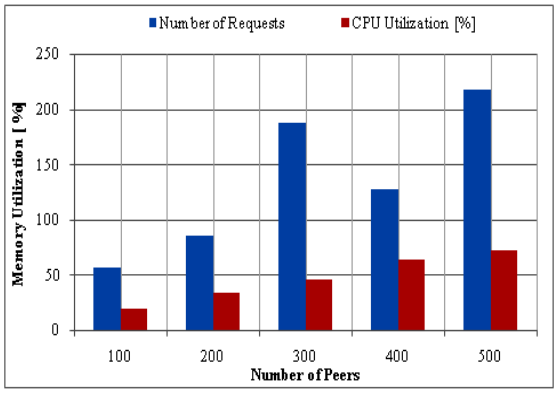


Fig. 6: Number of Request vs. CPU utilization.

Conclusion:

In this paper, a distributed data sharing method is proposed to both structured as well as to unstructured P2P network. This P2P makes a two-tier hierarchy for improving the efficiency in terms of accurate fast service. The data sharing and query reply is taken care distributedly. The T-NET is considered as the top tier where it takes care about all the S-NET requests. The S-NET is considered as the bottom tier which is can be accommodated with approximate S-NET with flexibility. This paper provides an efficient way of accommodating new peers in S-NET and new S-NET into T-NET. This distributed P2P network provides a good file sharing methodology and fast cooperation among the networks. The three methods such as distributed, load balancing, file sharing, cooperative caching are utilized in this paper for providing better quality of traffic management in Internet.

REFERENCES

"Allot Communications P2P Control White paper on Allot traffic management solution", http://www.allot.com/pages/P2P_WhitePaper_040922.pdf , accessed on March17, 2005.

Aggarwal, V., A. Feldmann and C. Scheideler, 2007. "Can ISPs and P2P users Co-operate for Improved Performance?", ACM SIGCOMM Computer Communication Review, 37(3).

Aggarwal, V., S. Bender, A. Feldmann and A. Wichmann, 2004. "Methodology for Estimating Network Distances of Gnutella Neighbors", in GI Jahrestagung - Informatik.

Alex Goldman, 2004. "Examining Layer Seven", Article in ISP-Planet, July 16. http://ispplanet.com/equipment/2004/streamsight_510.html.

Anttoni Halme, 2005. "Peer-to-peer Traffic: Impact on ISPs and Evaluation of Traffic Management Tools", HUT T-110.551 Seminar on Internetworking.

Clip2, "The Gnutella protocol specification v.0.4, March2001", http://www.clip2.com/GnutellaProtocol04.pdf.

Jorma Mellin, 2003. "Peer-to-peer and ISPs", Presentation at Tekes seminar Broadband at Home, Autum.

Jorma Mellin, 2004. "Peer-to-Peer Networking–Phenomenon and Impacts to Carriers", Presentation at Telecom Forum, Autumn.

Karagiannis, T., P. Rodriguez and K. Papagiannaki, 2005. "Should ISPs fear Peer-Assisted Content Distribution?", in IMC.

Krishna P. Gummadi, Richard J. Dunn, Stefan Saroiu, Steven D. Gribble, Henry M. Levy and John Zahorjan, "Measurement, Modeling and Analysis of

a Peer-to-Peer File-Sharing Workload", In Proceedings of the19th ACM Symposium on Operating Systems Principles,(SOSP-19).

Napster. http://www.napster.com.

Nathaniel Leibowitz, Aviv Bergman, Roy Ben-Shaul, Aviv Shavit, "Are File Swapping Networks Cacheable? Characterizing P2P Traffic", In Expand Networks, TelAviv, Israel.

Nathaniel Leibowitz, Matei Ripeanu and Adam Wierzbicki, "Deconstructing the Kazaa Network", In Proceedings of the 3rd IEEE Workshop on Internet Applications, (WIAPP'03).

Sherlia Shi, 2002. "Design of Overlay Networks for Internet Multicast", Doctor of Science Dissertation at Sever Institute of Washington University, Saint Louis.

Stefan Saroiu, Krishna P. Gummadi, Richard J. Dunn, Steven D. Gribble and Henry M. Levy, "An Analysis of Internet Content Delivery Systems", In Proceedings of the 5th Symposium on Operating Systems Design and Implementation , (OSDI).

Stefan Saroiu, Krishna P. Gummadi, Steven D. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems", At the Department of Computer Science& Engineering, University of Washington.

Vu, Q.H., *et al*, 2010. "Architecture of Peer-to-Peer Systems", Peer-to-Peer Computing, DOI 10.1007/978-3-642-03514-2-2, pringer-Verlag Berlin Heidelberg.

Xie, H.Y., Y.R. Yang, A. Krishnamurthy, Y.B. Liu and A. Silberschatz, 2008. "P4P: Provider Portal for Applications", ACM SIGCOMM.