



ADAPTIVE SERVICE DELIVERY IN PEER-TO-PEER NETWORKS

PATEL R.B.¹ AND VISHAL GARG²

¹Dept. of Computer Science and Engineering, Deenbandhu Chhotu Ram University of Science & Tech., Murthal (Sonapat)-131039, India.

²Dept. of Computer Engineering, M.M. Engineering College, Mullana-133203, Haryana, India.

*Corresponding Author: Email- ¹patel_r_b@yahoo.com, ²vishalgarg_vg@yahoo.com

Received: January 12, 2012; Accepted: February 15, 2012

Abstract- This article presents a common solution to Peer-to-Peer (P2P) network problems and distributed computing with the help of "Neighbor Assisted Distributed and Scalable Environment" (NADSE). NADSE supports both device and code mobility. In this article mainly we focus on the Adaptive Service Delivery Protocol (ASDP) based resource management technique. How service delivery in Peer-to-Peer (P2P) is speedup when using the NADSE service provider node in large network. Results show that performance of the NADSE network is better in all the network conditions.

Keywords- Cluster Head (CH), P2P, NADSE, Mobile Agent

Citation: Patel R.B. and Vishal Garg (2012) Adaptive Service Delivery in Peer-To-Peer Networks. Journal of Information Systems and Communication, ISSN: 0976-8742 & E-ISSN: 0976-8750, Volume 3, Issue 1, pp.-225-229.

Copyright: Copyright©2012 Patel R.B. and Vishal Garg. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Introduction

Peer-to-peer (P2P) is a computing model in which peer nodes collaboratively perform a computing task [8]. These peers can serve as both clients and servers and eliminate the need for a centralized node. More simply, a P2P network links the resources of all the nodes on that network and allows the resources to be shared in a manner that eliminates the need for a central host [14, 15].

The claim for P2P architecture is that enables true distributed computing, creating networks of computing resources. Hosts that have traditionally been used as clients can act as both clients and servers. P2P allows systems to have temporary associations with one other for a while, and then separate. Besides, nodes in P2P systems are autonomous in the sense that: (i) they can join the system anytime, (ii) they can leave without any prior warning, and (iii) they can take routing decision locally in an ad hoc manner. Unlike the conventional centralized systems, P2P systems offer scalability and fault-tolerance [2]. It is a feasible approach to implement global-scale systems such as the Grid [3, 15].

But the existing P2P systems take more computing time for finalizing the task. For the limitation of processing time of a request we

need a system which must support device and computation mobility as per need of the application.

In this article mainly we focus on the Adaptive Service Delivery Protocol (ASDP) based resource management technique. When "Neighbor Assisted Distributed and Scalable Environment (NADSE) [13] bridges two networks in that situation it maintains services available in the network along with route information to the node maintaining available services [17-20]. How service delivery in Peer-to-Peer (P2P) is speedup when using the NADSE service provider node in large network. Results show that performance of the NADSE network is better in all the network conditions.

Rest of the article is organized as follows. Related work is explored in Section II. System model is presented in Section III. Section IV gives the brief architecture of NADSE. Section V gives a look on ASDP. Section VI discusses on performance study of ASDP. Finally article is concluded in Section VII.

Related works

Napster [7] adopts central database approach for song titles, but it has inherent reliability and scalability problems that make it vul-

nerable when there are attacks on the database. Another approach, at the other end of the spectrum, is for the consumer S to broadcast a message to all its neighbors with a request for F. When a node receives such a request, it checks its local database. If it has F, it responds with the item. Otherwise, it forwards the request to its neighbors, which execute the same protocol. Proceeding in this manner will ensure that a requested resource is always be found when it exists. However, this solution has some critical limitations such as large overhead produced and the looping problem.

Gnutella [4] is a system approach for distributed data management that is based on this idea with some mechanisms to avoid request loops. It uses the time-to-live (TTL) flags in the request message to limit the broadcast scope of the message. However, this scoped broadcast approach does not scale either because of the bandwidth consumed by broadcast messages and the computing cycles consumed by the many nodes that must handle these messages. In fact, the day after Napster was shutdown, reports indicated that the Gnutella network collapsed under its own load, created when a large number of users migrated to Gnutella for sharing MP3 music files. To reduce the cost of broadcast messages, several other studies have been proposed in the literature to support intelligently forwarding and directed bread first search. Many variants of the well-known depth first search (DFS) have been also proposed. Many of the current popular systems, such as KaZaA [9], which are all based on the FastTrack [10] platform, adopt DFS concept. However, the disadvantage of these approaches, which are considered as hierarchical, is that the nodes higher in the tree take a larger fraction of the load than the leaf nodes, and therefore require more expensive hardware and more careful management. The failure or removal of the tree root or a node sufficiently high in the hierarchy can be catastrophic for the stability of the system.

The current available solutions for structured and unstructured P2P have advantages and severe limitations with regard to performance issues. A better P2P performance can be achieved with a more flexible and intuitive architecture which should be well-researched. Thus, we need to design an adequate P2P system which should meet requirements/challenges for fulfilling need of a successful resource management solution.

System Model

We are required to develop a computing/communication P2P systems that fulfills most of the above the challenges. This system should facilitate to improve the performance and incorporate new ideas. And also it should implement a structured P2P concept which must enable efficient resource management in P2P systems even during high rate of network whips. Developed protocol/algorithm should bridge two networks with the help of NADSE and should maintain services available in the network along with route information to the node maintaining available services.

System Architecture

A Neighbor Assisted Distributed and Scalable Environment (NADSE) system is shown in Figure 1. NADSE supports both device and code mobility. A mobile device will be the member of a cluster. A mobile node [16] in a cluster will work like cluster head (CH) and maintains information about other members of the cluster

in the form of database [1]. When a mobile node wants to search some information it requests to CH for members information (viz. IP address, identification certificate, etc.). If the CH is not aware about availability of the type of services a mobile node is interested and presence of the same in the cluster then it guides the same to the mobile node. Then mobile node uses dynamic and scalable pedestal (DSP) for P2P systems [5, 12] and creates a mobile agent to perform its desired task in the present cluster. Further, if the requested task is not completed with members of the present cluster mobile node may move to next cluster or it may take help of the CH and multicast its mobile agent to the CHs in the network. After completion of the task final result reaches to the mobile node which was its launching station. In NADSE system it is assumed that P2P network consist any number of MHs/Fixed Host (FH). MH/FH will often be disconnected for prolonged period of time due to the low power of battery or unreachable of signal but they will also frequently reallocate between different P2P networks at different time.

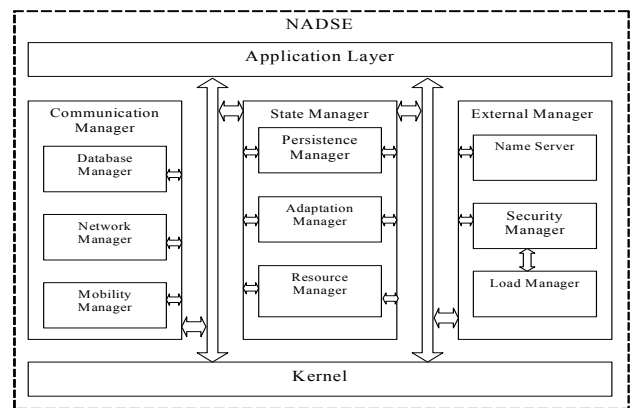


Fig. 1- Architecture of NADSE

NADSE serves users at anytime, anywhere. It is the integration of independent modules, and the synergy provided by a well established, unified baseline architecture that promotes the development of efficient, secure and fault tolerance P2P computing applications. NADSE consists of the manager modules and a Kernel. The Kernel is the basic utility module, which lies below the manager modules and is responsible for driving the NADSE, by ensuring proper co-ordination between the various managers and making them work in tandem. It uses secure communication pipe to transfer requests to/from other NADSEs. Various manager modules help to perform functions, viz. forwarding/receiving packets, communication, mobility, adaptation, fault tolerance, load balancing, etc.

When any MH/FH wants to communicate with some other MH/FH for the fulfilling the requirement for some required resources (viz., data file) which may be available on that MH/FH. To make this type of communication user friendly User Interface (UI) is provided which run on a MH/FH and enables the mobile applications to communicate with other MH/FH in the network. UI is platform independent and adapts variety of MH/FHs displays. To establish the communication link among the MH/FHs and BHs (running kernel of the NADSE at the Cluster head (CH) of a cluster) UI calls the Communication Manager (CM) which manages the data exchange between the different components of the NADSE. Usually, the MH/

FHs use different types of network connections to access the necessary information. The major task of CM is to decide which network component will perform desired task efficiently at lower cost. CM also supports the adaptivity of the system in case of bandwidth fluctuation.

Mobility Manager (MM) performs the Location Management (LM) of the MH/FHs. LM enables the system to track the locations of MH/FHs between consecutive communications. It includes two major tasks. The first is location registration/location update, where the MH/FH periodically informs the CH to update relevant location databases with its up to date location information. The second is call delivery, where the system determines the current location of the MH/FH based on the information available at the CH databases when a communication for the MH/FH is initiated. The design of LM techniques has the following objectives: (1) Reduction of latency of service delivery. (2) Quality of service (QoS) guarantees in different CHs. (3) When the service areas of heterogeneous wireless networks are fully overlapped: (a) through which CHs a MH/FH should perform location registrations. (b) In which CHs and how the up-to-date user location information should be stored. (c) How the exact location of a MH/FH would be determined within a specific time constraint. The LM keeps MH/FH connection active when it moves from one CH to another. When a mobile node is moving out of the serving area of a CH will enter another overlaying CH shortly. If a MH/FH is connected to a particular CH and wants chooses to be handed-off to the underlying CH for its future service needs may do it.

The overall network load among different nodes, a Resource Manager (RM) manages the resources of the system (viz., system memory used by MH/FH, files, etc.), and also controls the admission of mobile nodes that subscribe the services. Main duties include: authentication, Type of Services (ToS) and QoS negotiation, service subscription and un-subscription. To simplify the functionalities RM is subdivided into two another modules Resource Allocator (RA) and Replication Controller (RC). RA allocates and controls the resources for newly subscribed services. Resources are allocated fairly among mobile nodes, at the same time fulfilling individual mobile nodes requirement. RC keeps the global state of the distributed resources consistent among all local resources based on a given coherence strategy. We have assumed number of mobile MH/FHs, and the coordination among all the applications are managed by a Security Manager (SM), which collaborates with Name Server (NS) because names are symbolic ways of referring to objects across a network. Names are detached from their corresponding objects: one may possess a name without having immediate access to any object of that name. To enable mobility and disconnected operation, all objects across a wireless network should be denoted by unique names. NS manages the process of naming and identification. During this process there is very much risk of security and inconsistency of the data due disconnection and security. Security, trust and privacy are addressed from the very beginning of system design and on all levels such as hardware, operating system, protocols, and architecture. SM does the duties of (1) Protecting CH against unauthorized modifications, (2) Program validation/verification (what an uploaded/downloaded piece of software really does), Trust modeling, (3) How fragments of information can be efficiently shared in a controlled manner, Key/certificate management, and

(4) Implications of ad-hoc communities (what can be done without trusted servers). Persistence Manager (PM) is implemented for recovering the loss of information during disconnection while performing the mobile computations. It also provides fault tolerance to the communication channels on failure. The execution environment for both public architecture services (one set of services that are shared among all MHs) and application specific architecture services (each MH has its own set of services) is provided by the core module of the architecture, which is known as kernel with the help of different managers running just above.

For system fault tolerance and load balancing of the network NADSE contains mainly three agents and two servers. For balancing the load over the network these agents work together as a multiagent system [14].

Adaptive Service Delivery Protocol (ASDP)

When link a failure can occur in a network. Network Partitioning is a disastrous consequence of such communication link failures. The resource gets trapped inside one of these partitions. If all the resource requesters RPEs, the resource container SPE and the services served (SS) are in the same partition, the SS can still be served to resource requester. However, if the SPE or some RPEs are in different partitions, it is impossible for the SS to reach the remaining RPEs on its route until the link is recovered. This protocol uses two messages for insuring the delivery of services. These are- registration message and deregistration message

Suppose the SS is at RPE_u and its next service requester is RPE_v as shown in Figure 2. When the edge (RPE_u , RPE_v) fails, three cases arise depending on the position of the SS when the failure occurs.

1. Communication Link failure occurs before the SS starts traveling to RPE_v , so it cannot proceed and waits at RPE_u until the edge (RPE_u , RPE_v) recovers.
2. Communication Link failure occurs while the SS is M_{dereg}^v on the way to RPE_v . The SS is now not available to requester.
3. Communication Link failure occurs after SS has arrived at RPE_v . The messages exchanged between RPE_u and RPE_v will not reach their destinations.

We discuss each of these cases in detail below.

Case 1: In this case, the SS stops advancing and is caught in RPE_u . Instead of waiting for recovery, it can travel to another unvisited PE, say $RPE_{v'}$ in its route {Figure 2}. The decision on whether to travel to $RPE_{v'}$ or wait in RPE_u for the communication link to recover, is based on the number of trials in de-

protecting the availability of RPE_v . If this number is beyond a predefined threshold, the SS gives up traveling to RPE_v and instead travels to $RPE_{v'}$. The value of the threshold is application dependent.

If the edge (RPE_u, RPE_v) is not a cut-edge, the SS can finally travel to RPE_v without the recovery of (RPE_u, RPE_v) by the above mechanism. However, the SS may need to know the topology or routing information of the network in order to make an appropriate choice of $RPE_{v'}$. If the information is

available, the process of choosing $RPE_{v'}$ can be more efficient and an alternative route can be determined for the SS to travel to RPE_v ultimately. Unfortunately, the routing information of the whole network is usually not easy to get. More importantly, the routing information may change after the SS received it. Nevertheless, if there is no service server PE in the same partition, the SS can only wait for the recovery of the cut edge.

Case 2: When the communication link failure occurs as the SS is traveling to service requester, the SS is assumed to be lost. Since

the SS fails to travel to RPE_v , there will be no message M_{reg}^v sent to the RPE_u . Eventually, after the communication link is recovered, the SS will be recovered in RPE_v . In this scenario, one possible design is to allow RPE_u to recover the SS in another PE, say $RPE_{v'}$. Such an option can increase the

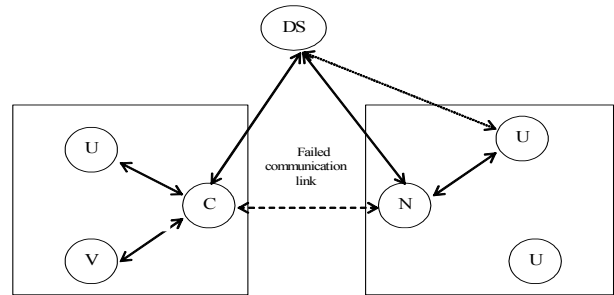
efficiency of the protocol. However, as RPE_u cannot guarantee whether the SS has survived the communication link failure or not, it must wait for communication link recovery if it is in the cut edge that does not contain SPE.

Case 3: In this scenario, the mechanism discussed in for Case 2 still works, but it may become less efficient. When the SS is in

RPE_v , registration message M_{reg}^v and deregistration message M_{dereg}^v will be sent to RPE_u . However, since the communication link is broken, the messages cannot reach

but can reach indirectly via domain server (DS). Instead of waiting for the successful message transmissions, the SS keeps on ad-

vancing. When the SS resides in a PE, it leaves indirect messages there for the PE. The SS stops traveling until it either reaches the RPE or is terminated by an PE failure. On the other hand, RPE_u keeps on trying to send messages to RPE_v which recovers the pending messages.



C- Current PE routing SS, U-helper PE to route the service, V-Service received, N- Service recipient

Fig. 2- Rerouting the SS on communication link failure to RPE

Implementation And Performance Study

For testing NADSE we have used total 25 nodes (2.2 Core 2 Due processor, 1 GB RAM, 160 GB HDD, Windows-XP, Java SDK 1.5), one server (for Internet communication), 1 access point (2700 DLink), 6 routers (2 No. CISCO 2851, 2 No. CISCO 2811, and 2 No. CISCO 1841) dividing 24 nodes into six networks of categories (Class A, Class B and Class C). 25th node used to take the services of the global network.

Here in this setup this node (25th) used to move the mobile agent on the infrastructured network, i.e. wired network. Complete setup is wireless. Further it is also considered that same node may be considered for multiple times for increasing the number of nodes in the system or mobile agent may visit infrastructure network depending on the availability of the services. This node also maintains list of available services and route to destination where services are available.

We have tested the ASDP in NADSE for distributing searching algorithm service to the requester when NADSE services are active in NADSE network. Figure 3 presents the time consumed in the completion of task is gradually increasing as density of nodes in the network is increasing. It is found that NADSE based network is superior in term of overall performance, means always service delivery is done. Reason behind the better performance is mobile service which moves to the service requester.

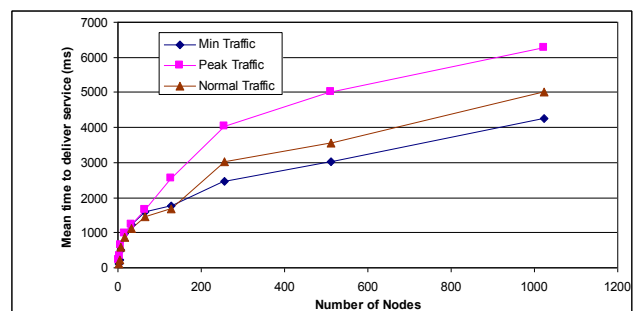


Fig. 3- When Location of Service is not known and may be available in the same domain failure of service providers is very frequent

Figure 4 presents the time consumed in the completion of task (distributing an information file of size 512 KB) when ASDP is active in NADSE network failure is very frequent. Service delivery is being done same network. From the result it is clear that service delivery time is gradually increasing as density of nodes increasing in the network. It is found that NADSE based network is superior in term of overall performance, means always service delivery is done. Reason behind the better performance is mobile service which moves to the service requester.

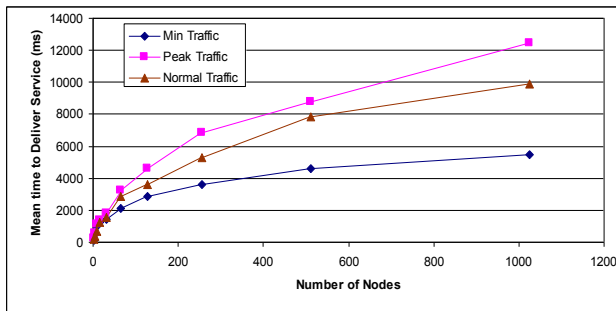


Fig. 4- When service is available on any nodes in the global network and failure of service providers is very frequent

Conclusion and Future Work

In this article we have ASDP based service discovery and delivery in NADSE P2P network. ASDP works like a pedestal to NADSE and it will work any P2P system for speeding the turn around time of a task. Further NADSE is based on the cluster and network is divided into small subnetworks. This division of the network does not increase more traffic when parallel computation of a task is done. Only traffic increased in a cluster which is being very small network and not affected by the network traffic. In future we will see other performance issues related to P2P networks.

References

- [1] Kai Guo and Zhijing Liu (2008) *IEEE Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, 352-355.
- [2] Nikta Dayhim, Amir Masoud Rahmani, Sepideh Nazemi Gelyan, Golbarg Zarrinzad (2008) *IEEE Third International Conference on Convergence and Hybrid Information Technology*, 166-171.
- [3] Qian Zhang, Yu Sun, Zheng Liu, Xia Zhang and Xuezhi Wen (2005) *IEEE 3rd Annual Communication Networks and Services Research Conference*, 1-6.
- [4] Ripeanu M. and Foster I.T. (2002) *1st International Workshop on Peer-to-Peer Systems*, 85-93.
- [5] Patel R. B. and Garg K. (2004) *WSEAS Transaction on Computers*, 1(3), 57-64.
- [6] Clarke I., Sandberg O. and Wiley B. (2000) *Workshop on Design Issues in Anonymity and Unobservability*.
- [7] Waldman M., Ad R. And Lf C. (2000) *9th USENIX Security Symposium*.
- [8] Theotokis S.A., Spinellis D. (2004) *ACM Computing Surveys*, 36(4), 335-371.
- [9] Kazaa (2003) *The Kazaa web site*. <http://www.kazaa.com>.
- [10] FastTrack Accessed on-line (2003) *The FastTrack web site*. <http://www.fasttrack.nu>.

- [11] Patel R.B., Vishal Garg (2009) *International Conference on Innovative Computing, Information and Communication Technology*, 323-329.
- [12] Patel R.B., Vishal Garg *PhD Thesis, IIT Roorkee, India*.
- [13] Patel R.B., Vishal Garg (2009) *International Conference on Recent Trends in Computing and Communications*, 147-152.
- [14] Patel R.B., Vishal Garg (2011) *2nd International Conference On Methods and Models in Science and Technology*, 1414, 159-164.
- [15] Trunfio P., Talia D., Papadakis H., Fragopoulou P., Mordacchini M., Pennanen M., Popov K., Vlassov V., Haridi S. (2007) *Future Generation Computer Systems*, 23, 864-878.
- [16] Wei-Shinn Ku, Roger Zimmermann, Chi-Ngai Wan and Haojun Wang (2008) *IEEE 22nd International Conference on Data Engineering*, 1.
- [17] Wassef Louati and Djamel Zeghlache (2007) *IEEE Wireless Communication and Networking Conference*, 2288-2293.
- [18] Heng Tao Shen, Yanfeng Shu and Bei Yu (2004) *IEEE Transactions on Knowledge and Data Engineering*, 16(7), 813- 826.
- [19] Bugra Gedik, and Ling Liu (2005) *IEEE Transactions On Computers*, 54(6), 767-782.
- [20] Ming Xu, Shuigeng Zhou, Jihong Guan and Xiaohua Hu (2010) *Journal of Network and Computer Application*, 33(2) 115-127.