私 立 東 海 大 學

資 訊 工 程 與 科 學 研 究 所

碩 士 論 文

指 導 教 授：呂 芳 懌 博士

以行動代理人及串流控制傳輸協定建置網格服務平台

A Grid Service Platform Using Mobile Agents and Stream Control Transmission Protocol

研 究 生：王 泰 翔

中 華 民 國 九 十 六 年 七 月 五 日

Abstract

Due to a huge amount of scientific, commercial and industrial requirements, Grid computing infrastructures have been quickly developed and constructed recently to fulfill users' computational needs. However, most current Grid systems connect nodes with hard-wired links. Seldom adopt wireless and mobile devices as their communication media and client facilities, respectively, to serve users. If we can integrate a Grid system with wireless networks, devices, techniques and resources to form a wireless Grid infrastructure where end users can deploy different kinds of wired and wireless terminal devices to access Grid resources, Grid systems will be soon popularly and widely used in different domains to serve many more people in the world. In this paper, we propose a wireless Grid resource platform, named Reliable Wireless Grid Service Platform (RWGSP), which integrates mobile agent and wireless techniques with a Grid system to provide users with a wireless accessible mobile agent environment in which the SCTP protocol is employed to increase reliability of system communication and solve Head-of Line problem.

Keywords: Grid, mobile agent system, wireless Grid platform, sendbox, SIP, SCTP

摘要

近年來，工商業及學術研究領域對於網格運算資源的需求日漸增加，其技術架構與規模正因應使用者的需求而快速的發展與佈建中。然而，大多數的網格系統僅能利用有線網路相互連接，極少有配備無線網路作為通訊媒介，亦無法進一步服務使用手持行動裝置之使用者。如果能將網格系統、無線網路與裝置、無線通訊技術與資源等進行整合，形成一無線網格系統，如此使用者便可利用各式裝置及有線、無線網路存取網格資源，進而使網格技術更加普及，並能更廣泛的應用在各領域中，服務更多的使用者。在此論文中，我們提出一套可靠的無線網格服務平台（Reliable Wireless Grid Service Platform，RWGSP），其將行動代理人平台及無線通訊設備安裝於各網格節點上，使用者將可藉由行動代理人，透過有線或無線網路進行服務或資源之存取，同時藉由整合串流控制傳輸協定，使用者將獲得更穩定的通訊品質，並能解決排頭擁塞（Head-of-Line，HOL）問題。

誌謝

轉眼間就在東海過了七年，經過多年的磨鍊後，不論是在專業技能或處世態度上，皆獲得了許多的成長。首先特別要感謝的，是從專題製作到碩士班五年來，對我照顧有加的指導教授呂芳懌教授，不論是課業上、研究上或是待人接物上，您都讓我獲益良多，修改論文報告時的嚴謹與精益求精、參加比賽時的分析建言、出國參訪時的照顧、待人處世的經驗分享…在此雖難以一一陳述，但學生都銘記在心，祝福您未來在學術研究上能過關斬將、步步高升。其次要感謝家人的支持，在求學路上給我莫大的鼓舞，並在我想要延畢一年以完成更多責任與夢想時，仍然贊成、包容我的選擇。也感謝女友及三隻寶貝貓咪在日常中的陪伴，在我忙於寫作、比賽時，帶來輕鬆與歡笑，替我分攤壓力。

資工系的楊朝棟教授、林正基老師，在課業及生活上給予學生許多幫助，口試委員林宜隆教授、蕭瑞祥教授及林熙禎教授對於學生碩士論文的建言，亦讓學生的論文更加正確充實。資料庫實驗室的惟傑、劾宏、子逸及各位學長姐在研究方向上給我的建言，世彥、鎮嘉、偉翔在比賽時的合作，又新、國財、文晉等學弟妹在我研究路途上的相互激勵與扶持，都是我最寶貴的回憶，也祝福大家未來都能在事業或學業上獲得理想的結果。

Table of Content

Abstract I
摘要 II
誌謝 III
Table of Content IV
List of Figures VI
List of Tables VIII
Chapter 1: Introduction 1
Chapter 2: Background and Related Work 4
2.1. Grid computing approaches 4
2.2. Wireless Grid approaches 4
2.3. Mobile Agent approaches 4
2.4. SCTP approaches 5
Chapter 3: System Architecture 6
3.1. Multiple Network Interfaces 8
3.2. Wired/Wireless Connectivity 8
3.3. Traversal through NAT 9
3.4. Multi-Homing 9
3.5. Multi-Streaming 10
3.6. GMAP 11
3.7. Sendbox Maintenance 12
3.8. Security 12
Chapter 4: Inter-agent Communication Protocols 13
4.1. Agent Management and Tracking 13
4.2. Agent States 13
4.3. Distributed Sendbox Scheme 14
4.3.1. Direct Mode 14
4.3.2. Forward Mode 15
4.4. Algorithms 16

4.5. Utilization Analysis 18
4.6. Communication Cost Analysis 18
Chapter 5: Service and Resource Broking 20
5.1. Resource Broker 20
5.2. Service types 21
5.2.1. Instant Response Service 21
5.2.2. Subscribe-and-Notify Service 22
5.3. Resource Selection 23
5.3.1. Communication Scoring 23
5.3.2. Resource Scoring 24
5.4. Transferring Optimization 26
Chapter 6: Experiments and Discussion 28
6.1. Inter-Agent Communication 28
6.2. Single-Homed and Multi-Homed Services 31
6.3. Multi-Streamed Service Delivery 36
6.4. Service Delivery in Grid Environment 36
Chapter 7: Conclusion and Future Work 40
References 41

List of Figures

Fig. 3-1 The RWGSP architecture and its network topology 7
Fig. 3-2 Network-layered structure of a Grid node in the RWGSP..8
Fig. 3-3 The SCTP packet format in which a chunk belongs to an independent data stream. 11
Fig. 3-4 The connection establishing diagram of SCTP 12
Fig. 4-1 Agent state diagram 14
Fig. 4-2 A scenario of the direct mode communication 15
Fig. 4-3 A scenario of the forward mode communication... 16
Fig. 4-4 Timing Sequence of an Agent.. 18
Fig. 5-1 The GSRB architecture and its components 21
Fig. 5-2 The Instant Response Service architecture and its processing scenario.................... 22
Fig. 5-3 The Subscribe-and-Notify Service architecture and its processing scenario 23
Fig. 5-4 Queuing behavior of the TCP socket connections... 27
Fig. 5-5 Queuing behavior of a SCTP association... 27
Fig. 6-1 Direct mode utilization rates 29
Fig. 6-2 Communication time of delivering service requests and data in different modes..... 29
Fig. 6-3 Comparison of communication time between our schemes and other service/data delivery schemes... 30
Fig. 6-4 Comparison of channel utilization rate between our scheme and other service/data delivery schemes... 31
Fig. 6-5 Network topology of our testbed in which a router has 4 interfaces and a client has 2 (The interfaces of wired testbed are 100MB Ethernet LAN interfaces, whereas those of wireless testbed are 802.11g WLAN interfaces) 32
Fig. 6-6 Utilization rates of an alternate path or alternate paths given different dropping rates of primary path.. 33
Fig. 6-7 Throughput of service/data delivery from a Grid node to a wired client 34
Fig. 6-8 Throughput of service/data delivery from a Grid node to a wireless client.............. 35
Fig. 6-9 Comparison of communication time and throughput of single-streamed and dual-streamed service delivery in a wireless environment 36

Fig. 6-10 Snapshot of C_Score and R_Score tables in a GSRB during node evaluation process. After querying Available Service Index, R_Score and C_Score of serving nodes are computed by using formulas (6) and (7), respectively, and finally GSRB chooses Node 2 which has the highest T_Score as the serving node by deploying formulas (8) and (9).. 38
Fig. 6-11 The time required to deliver a service file of 1MB from a serving node to a client in a multi-node wireless environment... 39

List of Tables

Table 5-1 An example of a C_Score table 23
Table 5-2 An example of a R_Score table 25
Table 5-3 Score table for CPU level 25
Table 5-4 Score table for current available memory size 25
Table 5-5 Score table for current length of a waiting queue 25
Table 5-6 Score table for current average CPU utilization rate 26
Table 6-1 Standard deviations (SDs) of communication time in wired and wireless environments. 35

Chapter 1: Introduction

Recently, many commercial and industrial wireless communication standards have been developed and proposed. By the transmission distance and application domain, they can be classified into several types, including Wireless Wide Area Network, Wireless Metro Area Network, Wireless Local Area Network (WLAN), Wireless Personal Area Network and so on. Among which WLANs have been widely established around us in recent years because prices of peripheral and access points (APs) are significantly reducing. They are now speedily entering users' life, education and entertainment to make users' life much more convenient and efficient. Although computing capability and storage capacity of mobile devices have become more powerful and larger, respectively, than before. The limited resolution of screen size, unacceptable power consumption and less multi-tasking functionality make them very difficult to replace PC, laptop and work station in the near future.

Besides, due to a huge amount of computational requirements, Grid computing infrastructures have been quickly developed and constructed. However, most Grid systems are used in scientific computing or to provide with a huge storage to users only, not very helpful to ordinary people. Also, they connect nodes with hard-wired links. Seldom adopt wireless technology and mobile devices as their communication media and client facilities, respectively, to serve users. In fact, if we can integrate wireless networks, devices, techniques and resources with a Grid system to form a wireless accessible infrastructure that can effectively bridge users and the Grid system, end users can then exploit different kinds of wired and wireless terminal devices to pervasively access Grid resources, and Grid systems will also soon be popularly and widely deployed by different domains and applications to serve many more people in the world, thus making Grid become much more practical and useful and providing users with a convenient network environment, through which we can ubiquitously access the internet to search and retrieve required information to enrich our everyday life. Mobile computing will be soon more powerful and pervaded than before. However, communication stability, cost and session continuity will be the major concerns, i.e., "online" time should be reduced, and messages should be successfully and reliably sent and received.

Additionally, a mobile agent is a software program which moves among nodes within a network based on designed logic to perform predefined task. It finally returns results to users. Autonomy and mobility are its key features. Many mobile agent systems have been developed and applied to a variety of domains, such as information retrieval and percolation [1], e-commerce [2], mobile communication [3] and distributed computing [4][5].

Basically, communication is one of the key issues in a mobile agent system, especially in a wireless environment. An agent has to communicate with others before it can properly cooperate with them to finish their given tasks. Hence, a reliable communication protocol is required. However, “reliable” often induces a sophisticate validation process or mechanism which often degrades the performance of a mobile agent system, and even makes mobile agents unable to accomplish desired mission. Another shortcoming is that TCP, one of the most popular transport layer protocols, can establish only one socket to connect two endpoints (a pair of IP and port number). When a connection fails (e.g., IP changed or port closed), the socket would be disrupted, even if several backup connections are available.

However, although two or more network interfaces can be installed in a mobile device, e.g., 802.11a/b/g and Ethernet in our laptop or 802.11b/g and GPRS/UMTS in our smartphone, the connection is unable to seamlessly switch among interfaces until the Stream Control Transmission Protocol (SCTP) was available. The SCTP was defined by Internet Engineering Task Force (IETF) Signaling Transport (SIGTRAN) working group in 2000 [6], as a transport layer protocol working analogously to TCP or UDP. The purpose, similar to TCP, is to provide a multi-phase handshaking and in-sequence packet delivery to increase system reliability. Moreover, it enhances the communication reliability by supporting the multi-homing and multi-streaming features.

In this article, we propose a wireless Grid service system, named Reliable Wireless Grid Service Platform (RWGSP), which integrates mobile agent and wireless techniques with a Grid platform to provide users with a wireless-end-device accessible Grid environment so that users can access Grid resources through wired and/or wireless networks. RWGSP not only deploys a Session Initiate Protocol [7] based (SIP-based) communication mechanism to establish a wireless Grid environment, but also employs SCTP to enhance itself to become a hard-wired and multi-homed Grid infrastructure so that its communication reliability and efficiency can be significantly improved and an ubiquitously pervasive computing environment can be achieved.

Besides, we bring up a transfer mechanism base on mobile agent technique, which delivery all information in RWGSP in synchronous or asynchronous way. A distributed sendbox scheme is developed on a each Grid node to temporarily buffers and appropriately resends messages for agents so that messages can be safely delivered to a receiver agent, even the agent migrates frequently. Each node in RWGSP has a sendbox to.

The rest of this article is organized as follows. Section 2 introduces the relevant background and related work. Section 3 describes our system architecture. Section 4 shows our distributed sendbox communication protocol and prerequisites. The grid service resource broker and experimental results of are presented in section 5 and section 6, respectively. Section 7 concludes this article and addresses our feature work.

Chapter 2: Background and Related Work

2.1. Grid computing approaches

Grid computing is becoming an interesting and important new field of research, mainly because it offers a large variety of applications, which, aggregating distributed resources and technologies to form a dynamic and distributed virtual organization over LANs or WANs, is frequently employed to process difficult and complex tasks and solve large-scale and complicated problems [7] aiming to enable dynamic selection, sharing and aggregation of distributed autonomous resources.

The Grid metaphor strongly illustrates the relation to and the dependency on a highly interconnected networking infrastructure [9]. The main components of a grid infrastructure are a security component, resource management services, information services and data management services. In grid computing, the term resource management refers to the operations used to control how capabilities provided by grid resources, and services are made available to other entities such as users, applications or services [10]. Grid computing has integrated with variety domains now, i.e., intrusion detection [11], to solve the problem with efficiency.

2.2. Wireless Grid approaches

Wireless Grid is a new but popular concept in recent years. McKnight et al. [12] proposed a resource sharing model for small and nomadic devices with which they simultaneously recorded a series of mono and mixed them to stereo sounds. However, they did not discuss communication performance and stability.

Srinivasan [13] proposed an assumption that wireless devices were primary integration devices. All peripherals could use short distance wireless communication (e.g., bluetooth, ultra wide band, Zigbee) to form a personal area network in order to share resources, such as monitor, disk, or input devices. However, the ultra wide band is still in draft, and bluetooth is widespreaded but too slow to transfer a huge amount of information.

2.3. Mobile Agent approaches

MEFS [14] proposed chasing message register and over-speed agent waiting mechanism. A message was considered as a chasing message after C times of failed delivery. While the migration frequency of an agent was over a threshold V, it stoped moving and checked if any chasing message was stored in the chasing message register. Some problems on the other hand were raised, e.g., the configuration of system parameter V.

Resending-based protocol [15] could provide reliable communication by using sliding-window for error control. When some messages were lost, sender resent them. After several trails, the sender requested the receiver’s new location from the server, and sent messages again. However, there was no upper limit of resending times. For frequently migrating agents, this protocol suffered from communication overhead.

Voyager [16] and Epidaure [17] raised a forwarding-pointer protocol on agent’s migration path. Before moving to the next node N, agent A left a forwarding pointer pointing to N on the node it currently resided. Messages were sent to the agent homeserver which forwarded them along A’s migration path based on the forwarding pointers. However, this approach could not solve the message chasing problem when the receiver changed its location very frequently.

Other schemes, such as Mailbox-based scheme [18], ARP [19], forwarding-based MAS [16-19] and resending-based approaches [14][15], all had their own problems and drawbacks yet to be solved.

2.4. SCTP approaches

Due to the advancement of reliability and performance, SCTP outperformed other protocols and communication schemes in several domains, e.g. parallel computing, handoff enhancement and multimedia.

A SCTP-based Grid web service, which inherited all the web service architecture from TCP-based one, was proposed by Otgonchimeg et al. [20]. Authors compared its performance with that of TCP. Kamal et al. [21][22] added the multi-stream functionality into traditional MPI (Message Passing Interface) program, which showed the advantage of multi-streamed transferring, but lacked multi-home support. Ahmed et al. [23] introduced a SCTP-based video transmission scheme, which detects buffer status of video stream to trigger connection switching, without affecting network performance. MA et al. [24] compared the performance of TCP and SCTP while handing over between UMTS and WLAN. Wang [25] and Camarillo et al. [26] employed SCTP to transport SIP traffic, but both lacked throughput analysis of a real environment.

Chapter 3: System Architecture

Figure 3-1 illustrates the RWGSP architecture in which Grid Mobile Agent Proxy (GMAP) is the management center taking charge of agent login, registration, dispatch, and recall. STUN (Simple Traversal of UDP through NATs) server, as a Network Address Translation (NAT) traversal solution of SIP, makes packets routable when users are now inside a NAT environment.

SIP is a popular application-level, text-based signaling protocol for creating, modifying, and terminating peer-to-peer communication session, especially for mobile networks. It provides with feasible methods, such as subscribe, notify, invite and so on, and has been invoked by several applications, e.g., VoIP or Instant Message systems, to connect end users.

Sendbox, as one of a storage subsystem of RWGSP, temporarily stores messages, that are currently unable to be delivered to their destinations, for agents. A sendbox forwards the message when receivers are ready. This can solve the message chasing problem raised by Zhou et al [14]. Besides, users can establish a wired or wireless connection, and register their SIP to GMAP through Grid mobile agent which is a mobile agent implemented on Grid. After the registration, users are authorized to access node status, and request Grid services (e.g., check node's status and send computational job) and/or other SIP-compatible services (e.g., voice over IP and video on demand) also through Grid mobile agent. All the accessed contents are sent to users in synchronously or asynchronously way.

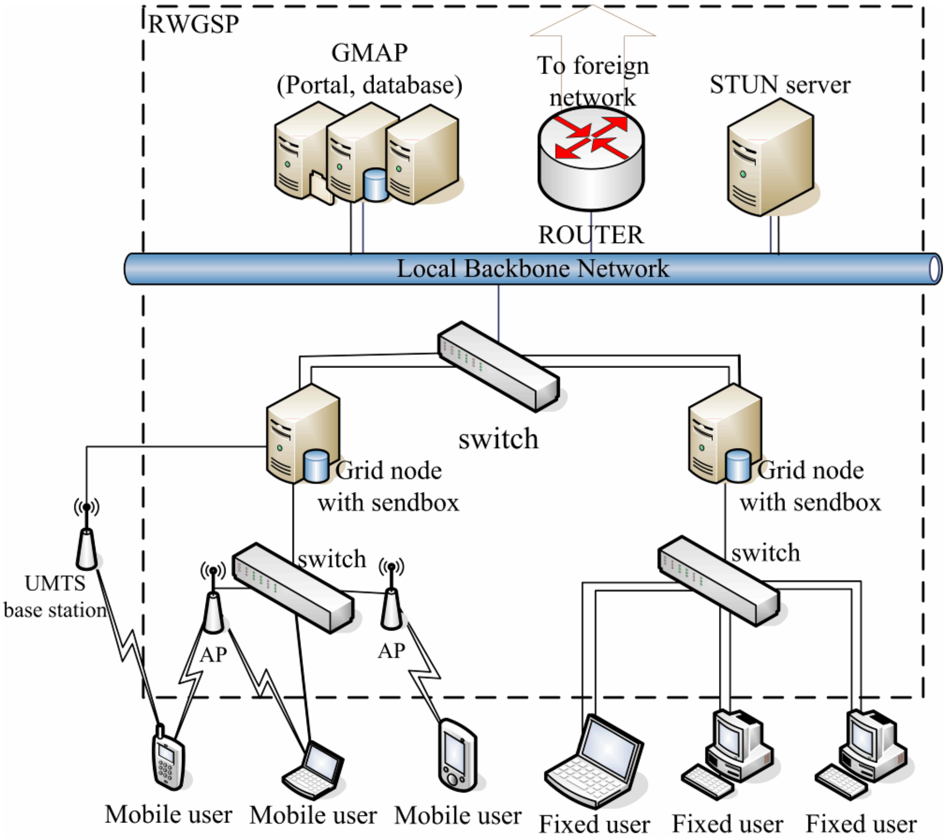


Fig. 3-1 The RWGSP architecture and its network topology

Figure 3-2 shows the layered structure of Grid nodes in RWGSP, in which network-access and Internet layers enable the multi-interfaces and multi-IPs facilities, respectively. In transport layer, the SCTP supports the multi-homing and multi-streaming features. Sendbox, mobile agents, path selection and command parallelization are also given to establish a reliable communication mechanism and a complete wireless Grid service platform. Furthermore, an arrow means a lower-level functions or features support higher-level mechanisms.

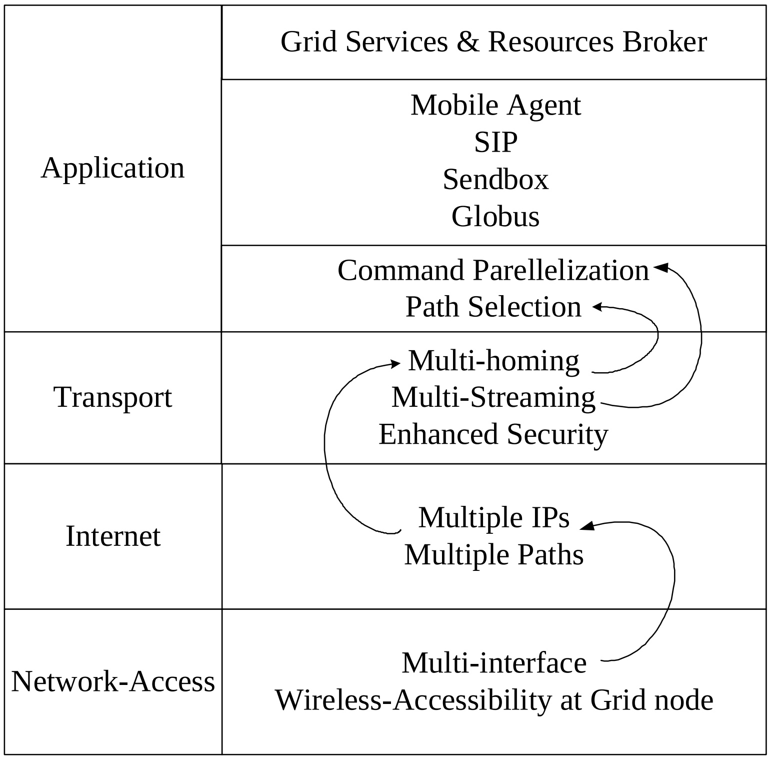


Fig. 3-2 Network-layered structure of a Grid node in the RWGSP

3.1. Multiple Network Interfaces

To raise the stability and availability of a Grid system, multiple network interfaces are created for each Grid node so that a node holds more than one routable IPs. Each IP establishes a path to connect itself and another node, i.e., a multi-link connection connecting two Grid nodes. Each node is able to switch its traffic (outbound and inbound) among the paths dynamically. A mobile user can also create multiple interfaces to communicate with the RWGSP if necessary. For example, the user keeps a WLAN session while establishing another session through LAN interfaces, both connecting himself/herself and RWGSP. When the user moves away and disconnects the hard-wired LAN connection, the wireless connection and interface will take over for the disconnected immediately without any reconfiguration, e.g., reset network setting or restart underlying application.

3.2. Wired/Wireless Connectivity

To achieve a high usability of RWGSP and the goal of pervasive computing, both RWGSP's wired and wireless connections must be reliably maintained. We propose an architecture that uses Grid nodes as local gateways to improve network connectivity and hence accessibility.

We connect those wireless APs having no routing function to their nearby Grid nodes, which provide the DHCP (Dynamic Host Configuration Protocol) service, through direct wired links. Using this type of APs, we can not only significantly reduce system construction cost, but also efficiently manage IP assignment of multiple APs without modifying the APs' firmware. Hence the resource usage of a Grid node can be markedly maximized.

Similarly, when a user access the Internet by directly connecting his/her terminal device to a switch or hub, the device will also receive a temporary IP from DHCP server so that it can communicate with others.

3.3. Traversal through NAT

RWGSP uses the STUN protocol [27] to make packets "routable", especially when they are tunneled by an NAT. The process is as follows. A Grid mobile agent (or simply a mobile agent), e.g., A, inside a NAT sends a query packet to the STUN server, which being outside the NAT retrieves current value of the "source port number" field from the packet, and replies with the port number to the agent, which will send the number to another Grid mobile agent, e.g., B. Thus, with the number, B can directly communicate with A as in a normal situation.

3.4. Multi-Homing

In SCTP, there are two ways to achieve multi-homing. One is providing a single interface that has multiple IP addresses, and the other is supporting multiple interfaces, each with an IP address and a transmission port. With multiple interfaces, the multi-homing protocol will as stated above switch task of transferring packets to one of the other interfaces/paths automatically when current path can not work properly resulted from malfunction of an IP, a router or a line. Often a multiple-interface node outperforms a single-interface node in reliability due to existing alternative paths and interfaces [28]. With SCTP protocol, all nodes in RWGSP are equipped with more than one wired and wireless interfaces to connect themselves with backbone and mobile facilities, respectively.

Further more, the quality of a path in RWGSP is determined by a quality decision algorithm which, located at the bottom of application layer, is implemented based on the RTT values which are reported by heartbeat packets and saved in SCTP-enabled nodes or user devices. An application chooses the path having the shortest RTT value as the primary path. Others are backup paths. When the primary path is congested or suspended, one of the backup paths will take over for the primary automatically and immediately.

3.5. Multi-Streaming

TCP transports a byte-stream at any moment, i.e., packets must be delivered in order, requiring that a byte should safely arrive at its destination before the latter bytes can be sent, even if the latter bytes have been ready for a while. That is so called the Head-of-Line (HOL) problem. SCTP in contrast supports multiple message-streams and conserves message boundaries. The term multi-streaming refers to the capability of SCTP that can transmit several independent streams of messages in parallel, acting like to bundling several TCP-connections into one SCTP-association which operates on messages instead of on bytes [29].

Using the TCP-based job dispatching and resource broking, HOL may occur in a crowded network environment. The waiting jobs belonging to different users or applications will be stuck by the unsendable messages in queue's top. Such will decrease the performance of a Grid platform since while waiting for an ACK issued by receiver, the sender can do nothing except receiving packets. However, by adaptation of SCTP, users can make an unordered communication by assigning a stream id to each independent job. Figure 3-3 shows the format of a SCTP packet in which several chunks share the same port information. Data chunks in a packet may belong to the same or different sources (applications, users, contents, etc.), and may be given the same or different stream identifications (stream-ids) depending on upper layer application's id-assignment policy [21]. After that, each id has an independent buffer. Jobs which transfer packets to the same node but belong to different tasks will be processed in parallel to ease up the HOL congestion.

Bits	Bits 0 - 7	8 - 15	16 - 23	24 - 31
+0	Source port		Destination port	
32	Verification tag			
64	Checksum			
96	Chunk 1 type	Chunk 1 flags	Chunk 1 length	
128	Chunk 1 data			
...	...			
...	Chunk N type	Chunk N flags	Chunk N length	
...	Chunk N data			

Fig. 3-3 The SCTP packet format in which a chunk belongs to an independent data stream

3.6. GMAP

In our scheme, before moving to another site S, a mobile agent must un-register to GMAP. After arriving at S, it must register to GMAP which can then keep track the agent's position and status in order to minimize the probability of communication failure. The registration information includes the agent's ID, S's IP, arrival time, and task (such as, querying node status or requesting services offered).

The GMAP creates a database, named registration information Databased, to keep mobile agents' information, including agent's SIP, status (online or offline), location (staying at a host or moving), port number (must be provided while the device is under a NAT environment), registration time, device type (laptop, PDA and smart phone, etc). The database like others also provides query interface so that users can conveniently access the contents.

A Grid mobile agent, e.g., A, before sending packets to another mobile agent, e.g., B, should query B's location from GMAP. The querying information consists of B's ID, query code (such as the code for querying B's location), and query time. The replied message issued by GMAP comprises B's ID, B's state (such as, waiting or moving) and query answer (e.g., B's current location). GMAP is also responsible for commanding Grid mobile agents to do something, e.g., order a mobile agent to access some types of documents or stop accessing when an error occurs.

3.7. Sendbox Maintenance

A Grid node stores messages that can not be sent currently in a sendbox, in which the information concerning a message, including sender's SIP, receiver's SIP, time of arrival, current life time, max life time, message priority, and contents, should be recorded. If current life time of a message reaches the max life time set by the sender or system administrator, sendbox monitor, which is a rule-based sub-system that manages sendbox, will delete the message to avoid the sendbox fully filled caused by receiver being offline for a long time. "If the free space of the sendbox is less than 30%, then sendbox monitor will delete the oldest message first based on the current life time field, otherwise, delete the messages of which current life time > 3hr." is a rule example.

3.8. Security

A DoS/DDoS attack can be issued by exploiting the defects of the TCP 3-way handshaking mechanism. SCTP protects its system against this type of attacks through a four-way handshake and the introduction of cookie mechanism. In SCTP, a client initiates a connection by sending an INIT packet to its server which responds with an INIT-ACK, that includes a cookie. A cookie is a unique context identifying a currently established connection. The client then replies with a COOKIE-ECHO, which contains the cookie sent by the server, indicating that the client is ready. Figure 3-4 illustrates the procedure. At this point, the server allocates resources for the connection and acknowledges the client by sending a COOKIE-ACK. Such can effectively protect RWGSP from being attacked by this type of flooding attacks.

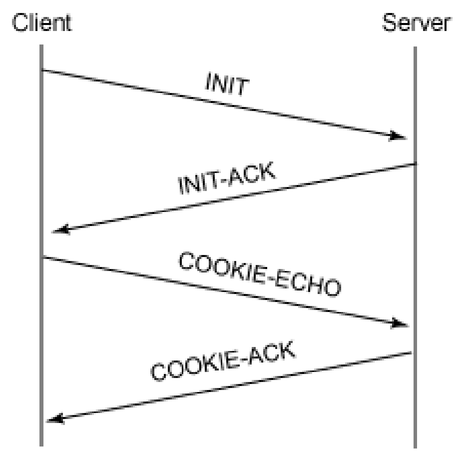


Fig. 3-4 The connection establishing diagram of SCTP

Chapter 4: Inter-agent Communication Protocols

Agents in RWGSP have two communication modes, direct and forward. Generally, direct mode handles all the communication between agents. If receiver agent is moving all the time, the message can not be safely delivered, RWGSP switches to forward mode to asynchronously forward messages.

A practically reliable protocol should effectively track the journey of an agent, and ensure the messages can be reliably delivered to their destinations, no matter how often and to which the receiver agent migrates. In the following we briefly describe the prerequisite functions.

4.1. Agent Management and Tracking

GMAP tracks all Grid mobile agents. Its tracking mechanism has two operational modes: real-time and non-real-time. To ensure that GMAP is able to trace an agent's steps, the former stipulates that after each migration, an agent as stated above should register to GMAP. It can then receive messages forthwith. Before migration, an agent submits a *Unregister* request to GMAP. As an agent moves frequently due to demands, registration process may be an obstacle in speeding up its movement. Our compromised solution is that non-real-time tracking allows an agent to register once per several movements. As a result, a frequently moving mobile agent may obtain a great level of freedom. Due to the constitution of SIP, each agent has an unique SIP URI (SIP Uniform Resource Indicators), formatted by agent_id<agent_id@hostname>, to identify the agent's name and current location where agent_id is the unique agent name (ID) that remains unchanged during its life, and hostname of SIP URI is name of the host that the agent currently resides.

4.2. Agent States

An agent has three states, as shown in Fig. 4-1. Normally an agent is in the active state, in which it may perform its requested operation or be free to migrate to any other host. When an active agent attempts to send requests to or receives requests from server or agent, it switches to the waiting state to continue its communication session. After sending or receiving responses, it switches from waiting to active. As sending a unregister request to and acknowledged from GMAP, agent turns into moving state. After reaching a node, and registering to GMAP again, it switches state from moving to active.

In addition, we have to set a feasible threshold for the waiting limit of the waiting-state agent in order to avoid indefinite postponement which occurs when a message can not arrive at its destination on time due to some possible failures encountered in the lower layer transmission during a communication. We redo the previous action if the waiting timer expires. Other state transitions are described in the following sections.

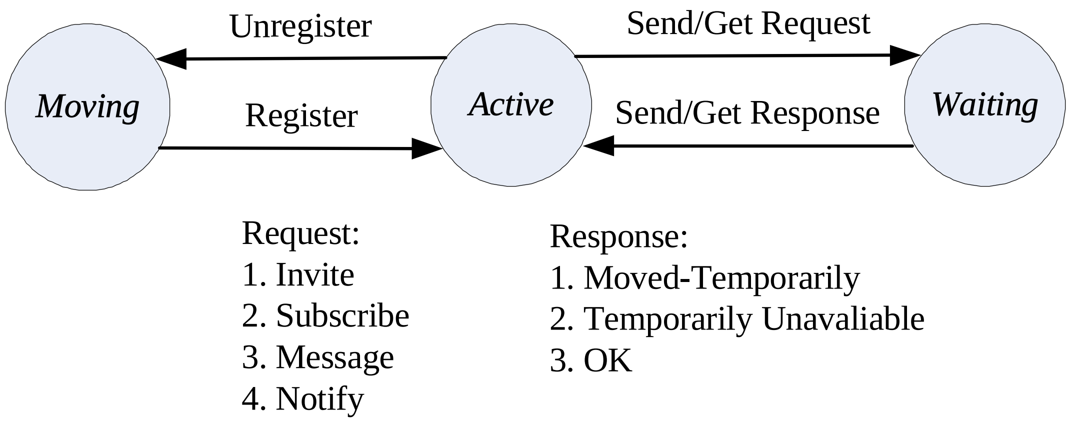


Fig. 4-1 Agent state diagram

4.3. Distributed Sendbox Scheme

In the distributed sendbox scheme, if recipient agent is staying on a node, the sendbox could send messages to it directly, Direct mode communication so-called. Otherwise, Forward communication mode is used.

4.3.1. Direct Mode

As illustrated in Fig. 4-2, sender S, staying at Grid node X, and receiver R, residing at Grid node Y, have registered to GMAP. Each of S and R is in either active or waiting state. With a desire to send a message to R, S first initiates an *Invite* request to GMAP for inquiring R's location.

After checking R's current state and location, GMAP replies with R's SIP URI and sends a *Notify* request to R asking R waiting for messages. Now the two interested agents reach their synchronization. Later, S sends a message directly to R with the informed address and waits for the response. After R receives the message and delivers an *OK* response to S, the communication resumes to its original situation, i.e, both S and R switch to their *Active* state. Once receiving a *Notify* request, R stays and waits until the message arrives. This occurrence ensures that message can be reliably delivered.

However, if agents do not move frequently, direct mode is capable of handling most interagent communications. An exception is that R is now migrating (moving state). This will incur unavailability of receiver's location. Especially as R migrates very frequently, message chasing may occur. The forward mode is then invoked.

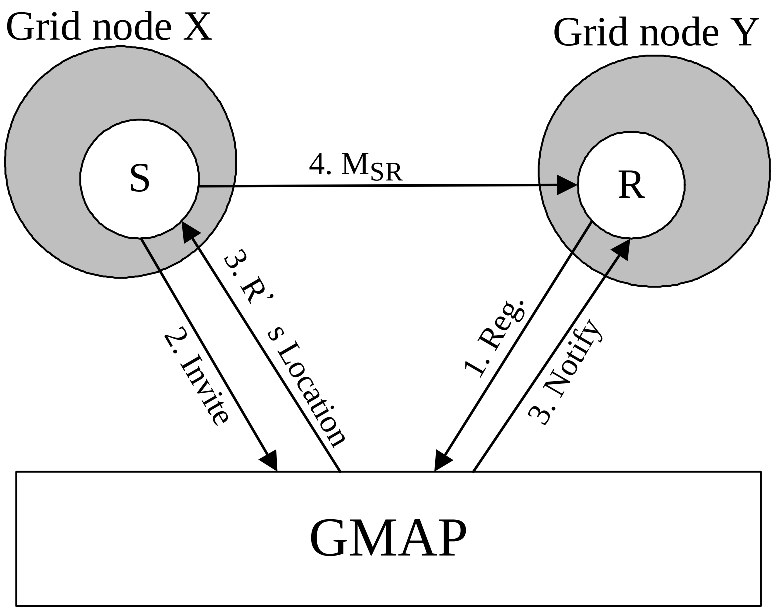


Fig. 4-2 A scenario of the direct mode communication

4.3.2. Forward Mode

As shown in Fig. 4-3, while R is migrating, S receives *Temporarily Unavailable* response after initiating an *Invite* to GMAP. S then sends message M_{SR} to Grid node X's sendbox. Accordingly, GMAP creates an event E_{SR} in its event table and replies with an *OK* to S. An event table at least comprises three primary fields, Sender's ID, X's location and receiver ID. Whenever getting a response *OK*, S returns to *Active* state and keeps performing some other task or migrates among the network. After R submits a *Register* request, GMAP checks if there is an event related to R. Assume E_{SR} does exist, GMAP clears E_{SR}, notifies R to stay and sends *Notify* with R's location to X which retrieves M_{SR} and forwards it to R. Even R migrates quite often in the network, messages can be delivered once R registers. Message chasing problem is then solved with a minor constraint, i.e, registering to GMAP. If no E_{SR} exists, R can keep migrating.

With the assistance of sendbox, agents do not need to carry unsent messages on their journey. Sendboxes distributively save and forward messages. S confirms existence of E_{SR} in GMAP to check if M_{SR} has been forwarded. In an extreme case, the sender and the receiver stay alternately, a series of communication will rely on the forward mode. Even two agents never stay on nodes at the same time, they can also reliably communicate.

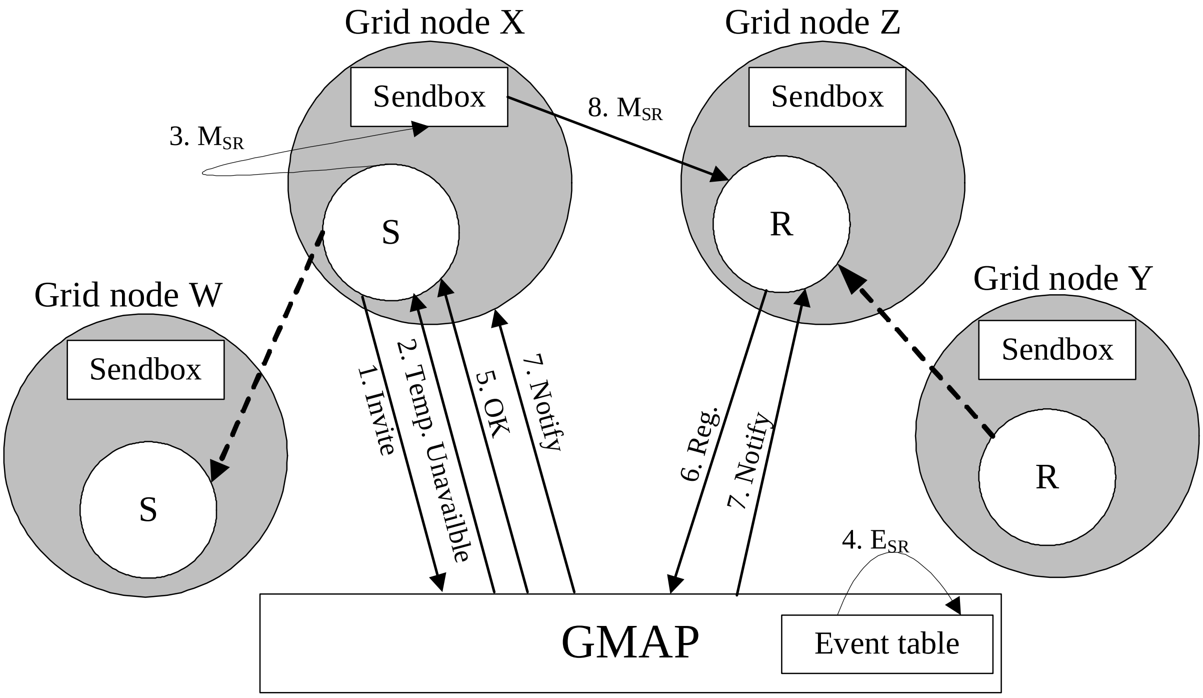


Fig. 4-3 A scenario of the forward mode communication

4.4. Algorithms

The algorithms for Grid mobile agent, GMAP and Grid nodes are summarized as follows.

1. Grid Mobile Agent

Case 1, Send()

GMA S requests for sending M_{SR} to mobile agent R: S sends *Invite* to GMAP, waits for a response from GMAP, State(S) =Waiting.

Case 2, Receive()

A mobile agent receives a message *m* from (S|GMAP|Grid node):

(1)If m=*Moved Temporarily* with R's SIP URI, /*the receiving mobile agent is sender S

Sends M_{SR} to R, State(S) =Waiting.

(2)If m=*Temporarily Unavailable*,/*the receiving mobile agent is sender S

Sends M_{SR} to Grid node's sendbox, sends *Subscribe* to GMAP.

(3)If $m = M_{SR}$ from (S|Grid node), /*the receiving mobile agent is receiver R

Replies with *OK* to (S|Grid node), State(R) = Active.

(4)If *m* =*Notify* from GMAP, /*the receiving mobile agent is receiver R

Waits for receiving M_{SR}, initiates a waiting timer, State(R) = Waiting.

(5) If m = *OK*, /*the receiving mobile agent x may be S or R

State(X) =Active.

Case 3, Migration

(1)If agent A reaches a platform, i.e., a Grid node, sends a *Register* to GMAP, then receives OK, State(A) = Active.

(2) If A is ready to migrate, sends *Unregister* to GMAP, State(A) = Moving.

2. GMAP

Case 1, Receives an *Invite* from S for requesting R's SIP URI:

(1)If R has registered, GMAP replies with *Moved Temporarily* attached by R's SIP URI to S, sends *Notify* to R.

(2)If R is unregistered, GMAP replies with *Temporarily Unavailable*.

Case 2, Receives *Subscribe* from S:

Creates an event E_{SR} with three fields, S's ID, R's ID and the Grid node's location, in its event table.

Case 3, Receives *Register* from agent A:

Checks if some events E_{XA} exist or not.

(1)If yes, for each E_{XA} sends *Notify* to A and the Grid node x, clears E_{XA}.

(2)If not, updates A's SIP URI, changes A's state(A) to *active*.

Case 4, Receives *Unregister* from agent A:

State (A) = moving.

3. Grid Nodes

Case 1, Receives a *Notify* with R's SIP URI from GMAP:

(1)Sends M_{SR} to R, waits for receiving *OK* from R.

(2)After receiving *OK* from R, removes M_{SR}.

Case 2, Receives M_{SR} from a local mobile agent:

Saves M_{SR} in its sendbox.

In the following, we analyze the two inter-agent communication modes and present their communication costs.

4.5. Utilization Analysis

There are two major factors affecting the two modes' utilization probabilities, *UP(Dir)* and *UP(Fwd)*, including migration time λ and the period of time R stays on a node, T_{stay}. The timing sequence of an agent is shown in Fig. 4-5. If an agent moves frequently, it spends much more time to migrate, and *UP(Fwd)* will be higher. While network is congested or receiver R moves more frequently, an agent takes longer migration time to migrate and to negotiate with GMAP. This occurrence also increases *UP(Fwd)*. *UP(Dir)* and *UP(Fwd)* of an agent are defined as:

$$UP(Dir) = \frac{\sum_{i=0}^{k} T_{stay}}{\sum_{i=0}^{k} (T_{stay} + l)} \tag{1}$$

$$UP(Fwd) = 1 - UP(Dir) \tag{2}$$

where k is the number of nodes an agent has passed through along its journey, and the term l includes costs of unregistering to GMAP, mobile code transportation and registering to GMAP.

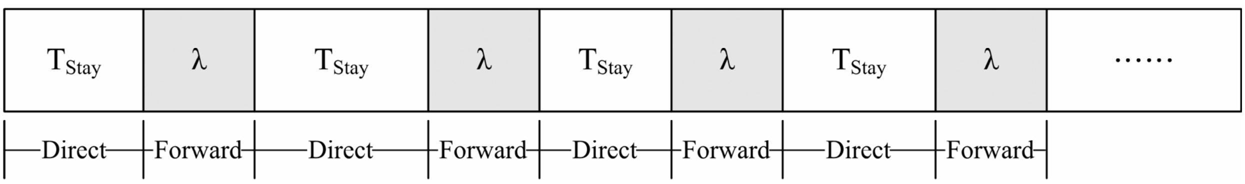


Fig. 4-4 Timing Sequence of an Agent

4.6. Communication Cost Analysis

The components of communication costs are shown as follows:

T_{inv}: the duration from the moment an agent sends an Invite to the moment it receives either Moved Temporarily or Temporarily Unavailable from GMAP

T_{msg}: the duration from the moment an agent sends a message to receiver to the moment it receives an OK response

T_{not}: the time needed to deliver a Notify message to an agent from GMAP

T_{sub}: the duration from the moment an agent sends a Subscribe message to GMAP to the moment it receives an OK response

T_{mig}: the duration from the moment from an agent leaves a Grid node to the moment it reaches another Grid node

The parameters include all hardware computation time and network transmission time between platforms.

The communication cost of direct model is:

$$Cost(Dir) = T_{inv} + T_{msg} \tag{3}$$

In forward mode, the communication cost is:

$$Cost(Fwd) = T_{inv} + T_{sub} + T_{mig}(R) + T_{not} + T_{msg} \tag{4}$$

The average cost of delivering a message from a sender to a receiver is:

$$Avg(\cos t) = T_{inv} + UP(Dir) * T_{msg}(Dir) + UP(Fwd) * (T_{sub} + T_{mig} + T_{not} + T_{msg}(Fwd)) \tag{5}$$

Chapter 5: Service and Resource Broking

To intermediate Grid services and resources for users, all Grid nodes and user devices each are installed with a GSRB subsystem not only to enable users to search and send their requests, but also to enhance the quality of service.

5.1. Resource Broker

In RWGSP, users are classified into ordinary users and Grid Service Providers. All services that Grid Service Providers provide must be indexed by GSRB before they can be conveniently searched or used by users. Desiring to register to GMAP, users can deploy mobile agent to connect to GMAP through wired/wireless network.

The GSRB, as shown in Fig. 5-1, consists of Service Repository, Request Repository, Available Service Index and Temporary Response Holder. Service Repository, a service container of a node, stores local service descriptions, contents and results for mobile agents. Request Repository keeps the information concerning users' requests. Available Service Index is an index of service providers and contents to speed up users' searching and browsing. Temporary Response Holder is implemented by using sendbox to temporarily hold service information.

After a Grid node is booted up and starts receiving service items provided by Grid service provides, GSRB stores the items in Service Repository, indexes the service content automatically with several key sentences extracted from service descriptions [30] and saves the indexes to Available Service Index. Users can query and store indexes into their devices, and deploy Resource Selector to accordingly select the most suitable services and Grid nodes that can effectively serve the services selected where Resource Selector is a subsystem that evaluates capability of all nodes of RWGSP, quality of links that connect underlying user and a node, and service types of user requests (e.g., an Excel or a SPSS file) before it selects serving nodes. Available Service Index in each Grid node exchanges its index with others automatically and periodically to keep its data and information up to date. The goals of broking appropriate Grid nodes to serve user requests are to increase the utilization of Grid nodes and their resources, and the performance of serving requests in RWGSP.

Users can send requests to GSRB periodically or occasionally to query or access information. Basically, to clearly identify services required, a request has to specify its service providers, service name, responding frequency (e.g., instantly or hourly) and sustaining period of the request (e.g., one hour or one day). After requesting a service, a user may go offline temporarily before the service information arrives. The GSRB will store the response into its Temporary Information Holder. When the user goes online again, the Holder delivers the response, e.g., playing a video program or sending the states of a node, to the user directly.

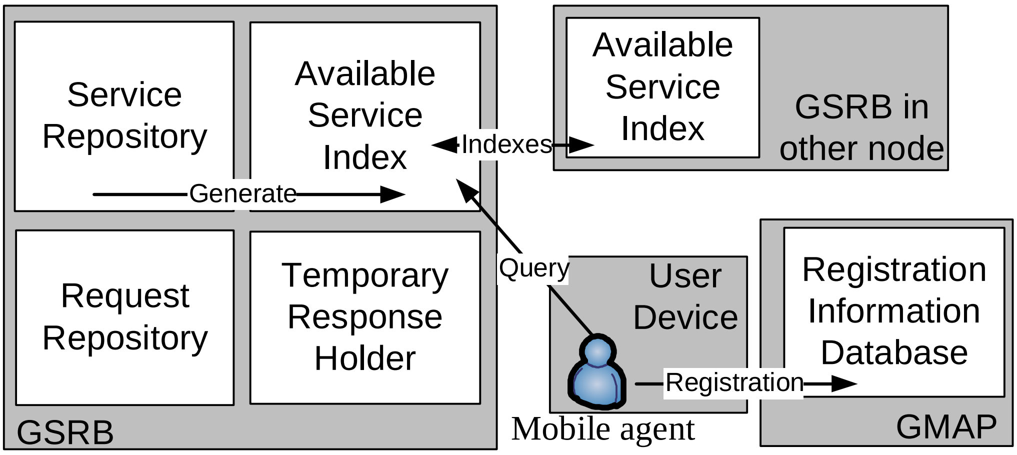


Fig. 5-1 The GSRB architecture and its components

5.2. Service types

To improve quality of service, we propose two main information exchanging mechanisms: Instant Response Service and Subscribe-and-Notify Service. The former is operated in an interactive mode. The later delivers service information indirectly.

5.2.1. Instant Response Service

Instant Response Service is the basic service of RWGSP. Wanting to query a Grid node x's status or to request its services, as shown in Fig. 5-2, a user, after registering to GMAP, dispatches a mobile agent to the node first. Request Repository stores the request M_{req} and forwards M_{req} to Service Repository, which, after the node finishes serving M_{req}, sends the response M_{res} to the user's device also through a mobile agent. Instant Response Service is implemented by using direct mode mentioned above. However, if the user device goes offline suddenly, GSRB will buffer the remaining messages in Temporary Response Holder and switch to Subscribe-and-Notify mode.

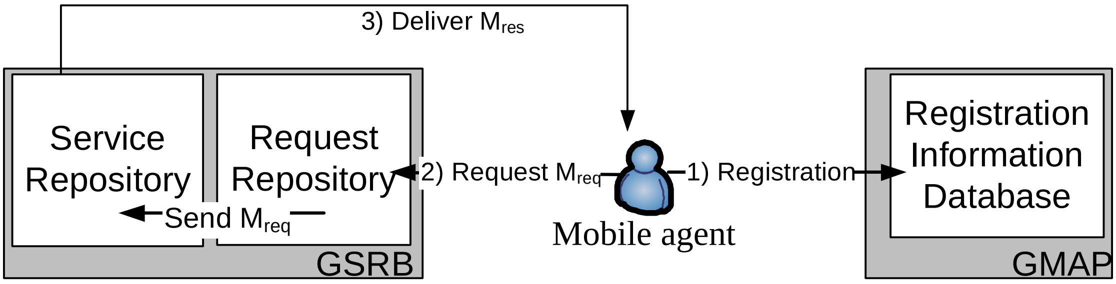


Fig. 5-2 The Instant Response Service architecture and its processing scenario

5.2.2. Subscribe-and-Notify Service

GSRB allows a mobile user to go offline after sending his/her request since service may be currently unavailable. It is inconvenient for the user to keep online and browsing the system periodically or frequently to wait for receiving the corresponding response. All the operations, including users' registration, request, notification, and communication, are performed by a mobile agent which is generated when the service is available. The agent first checks the Request Repository to determine which user issued the request. After that, it sends response to the Temporary Response Holder, and waits for the user's registration which will be reported by GMAP.

On the other hand, if the service is currently available, but the user, after sending a subscription or a request due to some reason may move or go offline immediately. When the user goes online again, two mobile agents will be generated. One on GMAP notifies the Temporary Response Holder to send/resend the corresponding response. The other on the user's device receives the response from GSRB, and shows the response on the screen. Then, he/she will never lose his/her desired information, even the user is busy or moving frequently, or the receiving device is temporarily offline. Of course, being notified by the agent on GMAP, if Temporary Response Holder discovers that the corresponding service is still unavailable, it notifies the user by sending an "unavailable" message. Fig. 5-3 illustrated the architecture and its processing scenario.

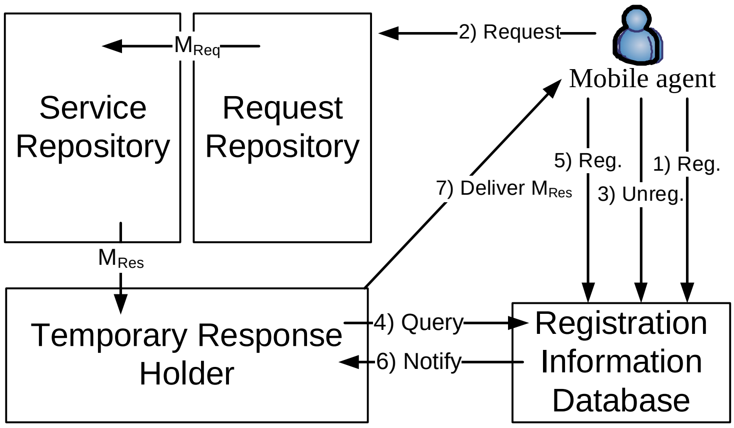


Fig. 5-3 The Subscribe-and-Notify Service architecture and its processing scenario

5.3. Resource Selection

To broke Grid resources for user requests, the Resource Selector has to consider several aspects, including users' requirements on user side, and processor capability, memory and secondary storage capacities and network bandwidth on resource side. However, a multi-homed network brings up an extra aspect – communication quality.

5.3.1. Communication Scoring

GSRB defines a table, named Communication-Score table (C_Score table in short), as shown in Table 5-1, for evaluating communication quality of a link. The table has eight fields.

(1). Node_ID: the reference number (a unique ID) of a candidate Grid node.

(2). Path_ID: the reference number of an available path connecting two Grid nodes.

(3). Source_IP: sender IP of an available path.

(4). Dest_IP: receiver IP of an available path.

(5). RTT: round trip time of a path.

(6). I_Type: interface type of a path, e.g., FE means a 10/100 Fast Ethernet path, and UMTS stands for a 3G mobile network link.

(7). I_Weight: weight of an interface of a path, ranging from 1 to 10.

(8). Throughput: the most recent throughput of a path where null means "have not transferred before".

Table 5-1 An example of a C_Score table

Node_ID	Path_ID	Source_IP	Dest_IP	RTT	I_Type	I_Weight	Throughtput
0	0	140.128.101.1	140.111.57.1	5ms	FE	6	48mbps
0	1	140.128.101.1	140.111.58.1	9ms	FE	6	null
0	2	220.131.99.6	140.111.57.1	timeout	UMTS	3	null
0	3	220.131.99.6	140.111.58.1	17ms	UMTS	3	1mbps
1	0	140.128.101.1	140.110.122.41	3ms	FE	6	null
1	1	140.128.101.1	140.110.123.67	4ms	FE	6	null
1	2	220.131.99.6	140.110.122.41	22ms	UMTS	3	null
1	3	220.131.99.6	140.110.123.67	25ms	UMTS	3	null
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

The scores in a C_Score table can be updated and referred to by upper layer applications dynamically. For example, the throughput can be updated by an application when this application finishes its transmission. By measuring all paths' average RTTs, the application can determine which path is the most suitable one as one of the aspects for building a connection to serve a request. Let C_Score(x) be the communication quality of a link that connects a node X and its opposite site, which is either a Grid node or a user.

$$C\_Score(x) = 0.1 * Avg.(\sum_{i=1}^{n} \frac{I\_Weight_{xi} * throughput_{xi}}{RTT_{xi}}) \tag{6}$$

where n is number of paths that connect x and its opposite site, and $I\_Weigcht_{xi}$, $throughtput_{xi}$ and RTT_{xi} are respectively I_Weight, throughput and RTT of i-th path.

5.3.2. Resource Scoring

A resource score (R_Score in short), recorded in a table named Resource-Score table (R_Score table in short), as shown in Table 5-2, is generated for each Grid node by deploying formula (7).

$$R\_Score(x) = (freq_x / 1G) * NoCores_x + \sum_{k=1}^{p} (SC_{xk}) \tag{7}$$

where p is the number of concerned features (excluding CPU frequency and number of cores) of a node, R_Score(x) and SC_{xk} are the current scores of node x and feature k ($1 \le k \le p$), respectively, freq the frequency of CPU, and NoCores is number of cores that has been equipped in node x, e.g., NoCores=2 when two single-cored CPUs or one dual-cored CPU is installed, whereas NoCores =4 when there are two dual-cored CPUs or four single-cored CPUs, instead. In this study, p=4 for a Grid node. There are CPU level (CPU), current available memory size (CAMS), current length of a waiting queue (CWQL) and current average CPUs utilization rate (CACUR). Tables 5-3 to 5-6 [37] respectively show their score tables from which a feature score, SC, can be retrieved and R_Score can then be calculated. The score tables should be updated periodically to reflect the real trend of the hardware evolution. Besides, a node with a higher R_Score should be one that performs better in computation.

Table 5-2 An example of a R_Score table

Node_ID	CPU	Freq	NoCores	CAMS	CWQL	CACUR
0	4	2.8Ghz	2	4	4	5
1	4	2.4Ghz	1	5	3	4
2	3	1.8Ghz	1	3	3	4
3	1	733Mhz	1	1	2	2
4	2	400Mhz	1	1	3	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 5-3 Score table for CPU level

CPU level (*CPU*)	Score
Intel Core 2 Duo or Intel Xeon	**5**
Intel Pentium IV or AMD Athlon K8	**4**
Pentium III or AMD Athlon K7	**3**
Pentium II or AMD Athlon K6	**2**
Earlier model	**1**

Table 5-4 Score table for current available memory size

Current available memory size (CA*MS*)	Score
$2048MB \le CAMS$	**5**
$1024MB \le CAMS \le 2048MB$	**4**
$512MB \le CAMS < 1024MB$	**3**
$256MB \le CAMS < 512MB$	**2**
$CAMS < 256MB$	**1**

Table 5-5 Score table for current length of a waiting queue

Current waiting queue length (*CWQL*)	Score
$0 \le CWQL \le 25$	**5**
$25 \le CWQL < 50$	**4**

$50 \le CWQL < 75$	**3**
$75 \le CWQL < 100$	**2**
$CWQL < 100$	**1**

Table 5-6 Score table for current average CPU utilization rate

Current average CPU utilization rate (*CACUR*)	**Score**
$0\% \le CACUR < 20\%$	**5**
$20\% \le CACUR < 40\%$	**4**
$40\% \le CACUR < 60\%$	**3**
$60\% \le CACUR < 80\%$	**2**
$80\% \le CACUR < 100\%$	**1**

The total score T_Score of a wired-node is calculated by formula (8), and that of a wireless node is computed by formula (9) in which communication quality is a major concern.

$$T\_Score_{wired}(node_X) = W_{pwd} * C\_Score(node_X) + W_{rwd} * R\_Score(node_X) \quad (8)$$

$$T\_Score_{wireless}(node_X) = W_{pwl} * C\_Score(node_X)^2 + W_{rwl} * R\_Score(node_X) \quad (9)$$

where W_{pwd} and W_{rwd} (W_{pwl} and W_{rwl}) are respectively weights of path and resources if hard-wired (wireless) connection is used.

5.4. Transferring Optimization

In a TCP-based environment, a socket connection shares the same outgoing queue with other sessions of the same socket. As stated above, follow-up packets in the queue may be blocked by its former packet, particularity when communication quality is poor and the former can not be smoothly delivered. Fig. 5-4 gives an example, in which we assume that all control commands are privilege commands for maintaining Grid nodes, and DATA-A and DATA-B are two independent data streams generated by different processes but sent to the same destination node. If a data packet, e.g., DATA_A1, is dropped due to network congestion. Such will block following packets until the packet is resent successfully. In such a situation, control commands 2 and 3 may be delayed, causing control problems and troubles for underlying platform.

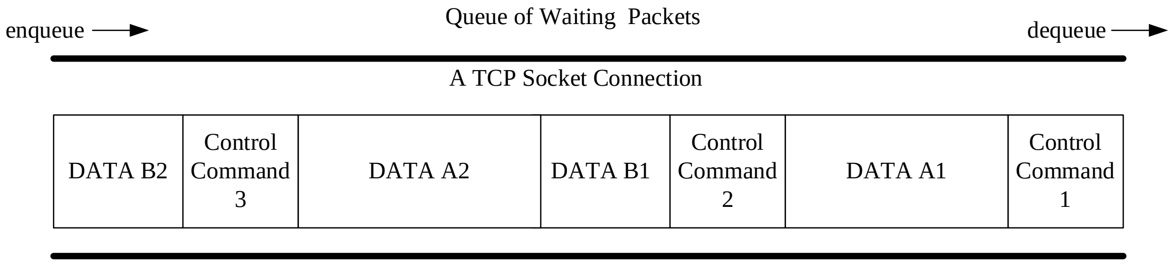


Fig. 5-4 Queuing behavior of the TCP socket connections

However, a SCTP-based environment decomposes messages into several streams based on the messages' characteristics, e.g., messages are generated by an application, different applications or different users. Packets of a stream are inserted into an independent queue to wait for being transmitted. Fig. 5-5 depicts behaviors of the SCTP waiting queues. When a packet, e.g., DATA_B1, is dropped, the HOL effect will be only on those behind the packet in Stream_id 3. The other two queues still work normally.

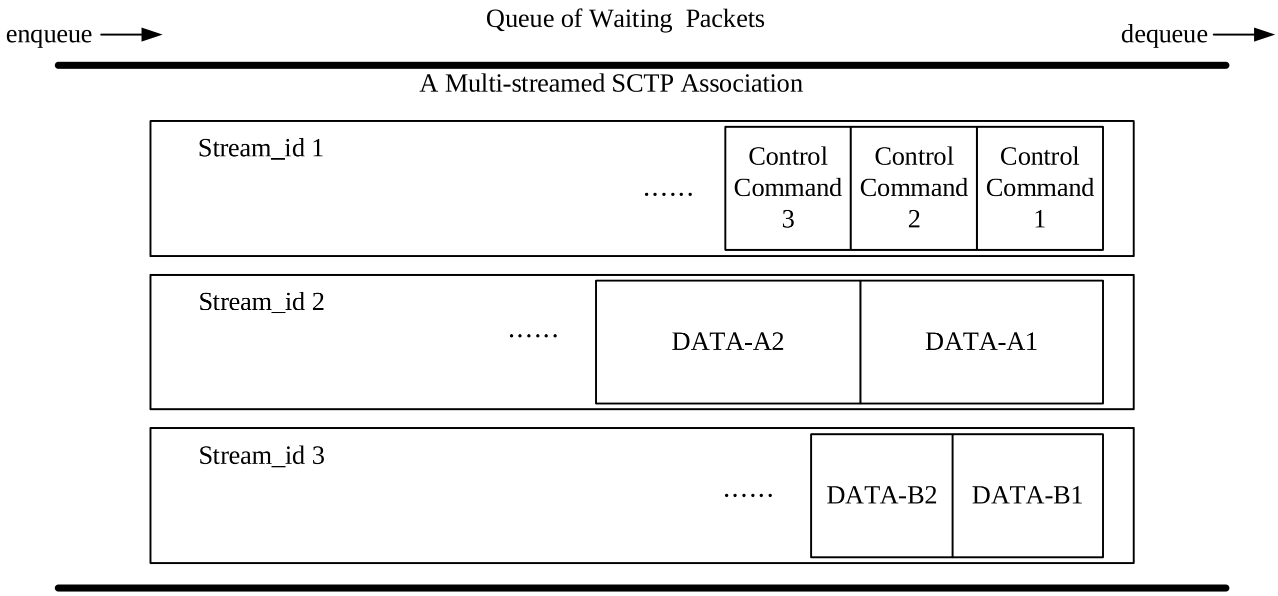


Fig. 5-5 Queuing behavior of a SCTP association

Chapter 6: Experiments and Discussion

The RWGSP prototype and our agent communication protocols are developed with Java. The test-bed of the RWGSP system consists of: 1). 8 Grid nodes that are connected by links of 100MBit/s Ethernet interfaces and managed by Globus Tool Kit 4.0 as the mobile agent network, and each node is equipped with dual 100Mbit/s Ethernet LAN interfaces to support multi-homed requirements; 2). 802.11g (i.e., 54 Mbps) WLAN APs to serve wireless network users; 3). three laptops as the wireless clients, each of which is equipped with a 802.11g WLAN interface and an Ethernet LAN interface.

6.1. Inter-Agent Communication

In the following experiments, a receiver roams in the RWGSP environment, visiting Grid nodes randomly, staying at a node for a given time T_{stay}, and then migrating to other node to simulate the users' migration. A sender sends a message M_{SR} to the receiver periodically, once every g_{msg} time interval. A user requests, M_{req} (e.g., a balance sheet file for statistic calculation), is sent from a mobile device to a Grid node via wired or wireless links. The size of a mobile agent and M_{SR} are 20KB and 10 KB, respectively.

The first experiment deals with utilization rate of direct mode *UP(Dir)* given different g_{msg} and T_{stay}. The results are shown in Fig. 6-1, in which λ (recall, migration time) ranges from 50ms to 130ms, avg(λ)=90ms, and from which we can conclude that 1). the more frequently a receiver moves (a shorter T_{stay}), the higher *UP(Fwd)* (i.e., 1-*UP(Dir)*) it will be; 2). A larger g_{msg} results in a higher *UP(Dir)* since the smaller a g_{msg} is, the higher the probability that when the sender is ready to send a request, receiver has already unregistered. However, the affection of g_{msg} is not significant.

The second experiment concerns the communication costs of the direct and forward modes between two Grid nodes against different T_{stay}. The results are shown in Figs. 6-2(a) and 6-2(b). The communication time of direct mode varies between 14ms and 41ms, while that of forward mode ranges from 49ms to 133ms because using forward mode the sendbox has to buffer and send messages, consequently generating extra data access and communication costs. Interaction between the two nodes is also much more complicate than using direct mode. Furthermore, different g_{msgs} do not significantly affect communication time.

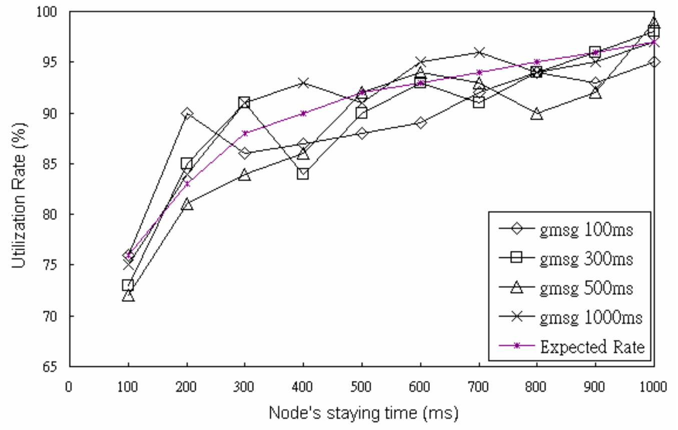
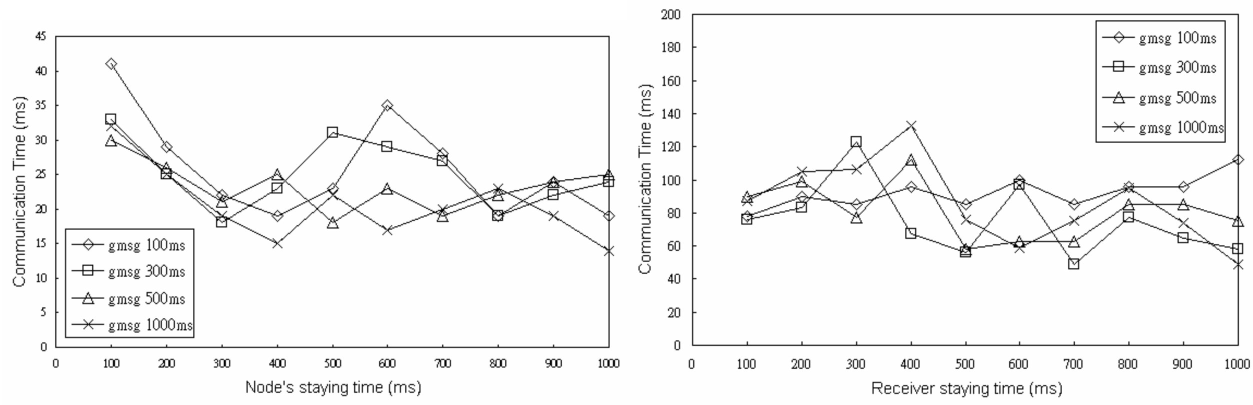


Fig. 6-1 Direct mode utilization rates

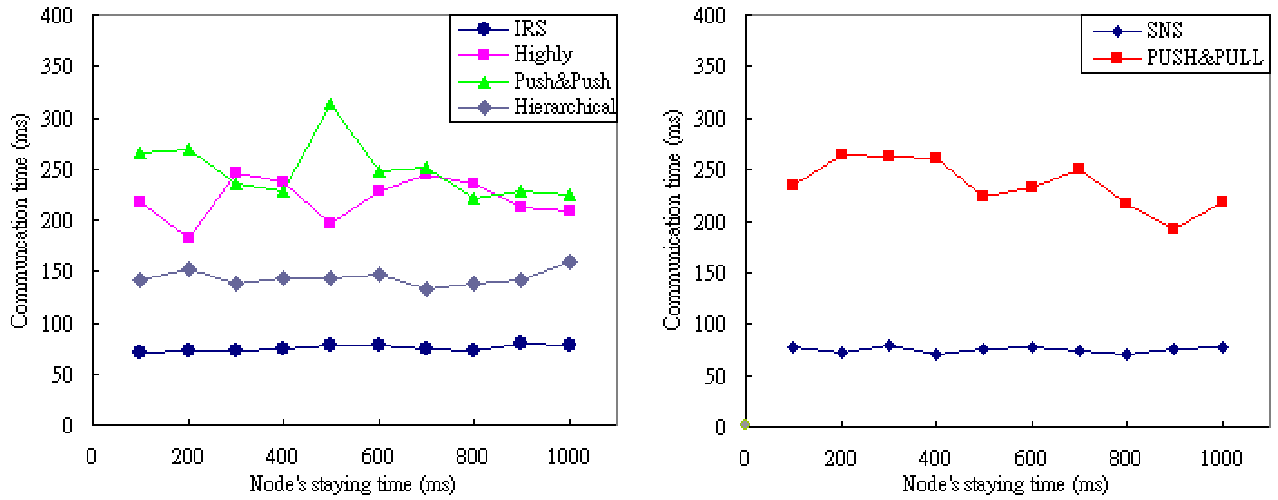


(a) direct mode (b) forward mode

Fig. 6-2 Communication time of delivering service requests and data in different modes

Figs. 6-3(a) and 6-3(b) show comparison of communication time of our Instant Response Service and Subscribe-and-Notify Service schemes with other synchronous [32][36] and asynchronous schemes [15][18], respectively. For a synchronous delivery, as shown in Fig. 6-3(a), the Instant Response Service scheme outperforms others due to the former involving no relay stations. Relay-based schemes (e.g., Push&Push and Highly Mobile Agents) send their service contents via a relay station, thus increasing packet traveling time and causing a bottleneck while relay station’s loading is high. Hierarchical scheme navigates task agents with a navigator agent. This increases time required for them to migrate from node to node, resulting in additional transferring costs.

For an asynchronous delivery (see Fig. 6-3(b)), the Subscribe-and-Notify Service scheme stores the delivered contents into sendbox temporarily and forwards them to receivers sometimes later, without requiring the sender keeping online and staying at a node. On the contrary, the Push&Pull scheme resends the contents to receiver only when sender stays at a node, causing re-delivery delay and immovability of mobile agents.

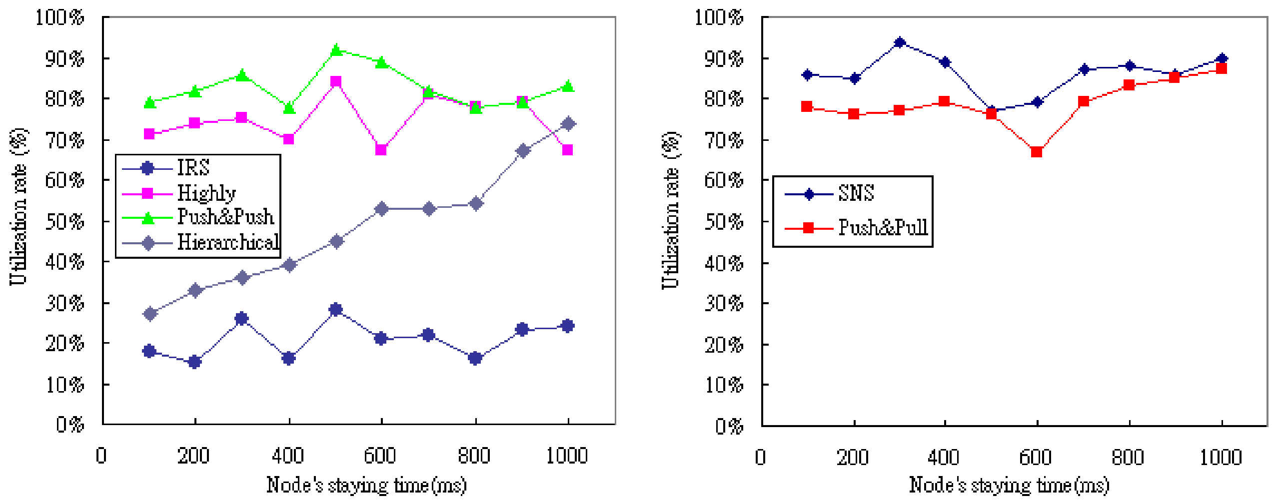


(a) synchronous schemes (IRS: Instant Response Service)

(b) asynchronous schemes (SNS: Subscribe-and-Notify Service)

Fig. 6-3 Comparison of communication time between our schemes and other service/data delivery schemes

Figs. 6-4(a) and 6-4(b) show comparison of channel utilization rates of our Instant Response Service and Subscribe-and-Notify Service schemes with other synchronous and asynchronous schemes [18][19], respectively. Compared with other synchronous schemes, the Instant Response Service outperforms the relay-based schemes addressed due to the same reason mentioned above. The more interaction between two nodes for delivering a message, the more bandwidth they consumes. In Hierarchical scheme [36], a navigator agent navigates more task agents given a longer T_{stay}. This occurence increases the GSRB's analytical time to track the agents' location, also raising its channel utilization rate. On asynchronous schemes, channel utilization rate of our Subscribe-and-Notify Service scheme is slightly higher than that of Push&Pull since sending messages that are temporarily stored in sendboxes needs to establish a channel for a faster re-delivery. The more interactive activities, the more bandwidth a scheme consumes. However, the establishment consequently solves the message chasing problem[14].



(a) synchronous schemes (IRS: Instant Response Service)

(b) asynchronous schemes (SNS: Subscribe-and-Notify Service)

Fig. 6-4 Comparison of channel utilization rate between our scheme and other service/data delivery schemes

6.2. Single-Homed and Multi-Homed Services

In the following experiment, we evaluate performance of service delivery in wired and wireless LAN environments using TCP-based and multi-homed SCTP-based approaches. Clients and servers in the environments are all equipped with 1.4GHz CPU, 512MB RAM, ubuntu linux server 7.04 and a SCTP protocol stack implemented on linux named lksctp [33]. The network topology of our wired testbed is illustrated in Fig. 6-5, in which a router has 4 100MB-based LAN interfaces, and a client has 2 such interfaces.

The network topology of our wireless testbed is same as that of the wired one, except that all 100MB-based LAN interfaces are substituted by 802.11g 54MBps wireless LAN interfaces. User requests M_{req} issued by a wired client are individually 1MB, 10MB, 50MB and 100MB in length, but for a wireless client, user requests are each 1MB, 5MB, 10MB and 25MB in length. Dropping rates of user request delivery range from 0% to 10% which are emulated by the NIST WAN Emulator [31].

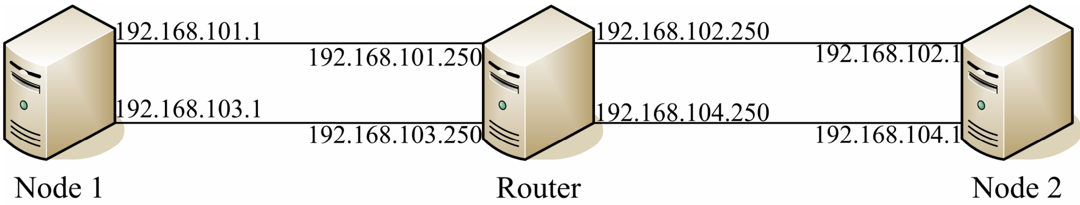


Fig. 6-5 Network topology of our testbed in which a router has 4 interfaces and a client has 2 (The interfaces of wired testbed are 100MB Ethernet LAN interfaces, whereas those of wireless testbed are 802.11g WLAN interfaces)

Utilization rate of a primary path is defined as that in formula (10).

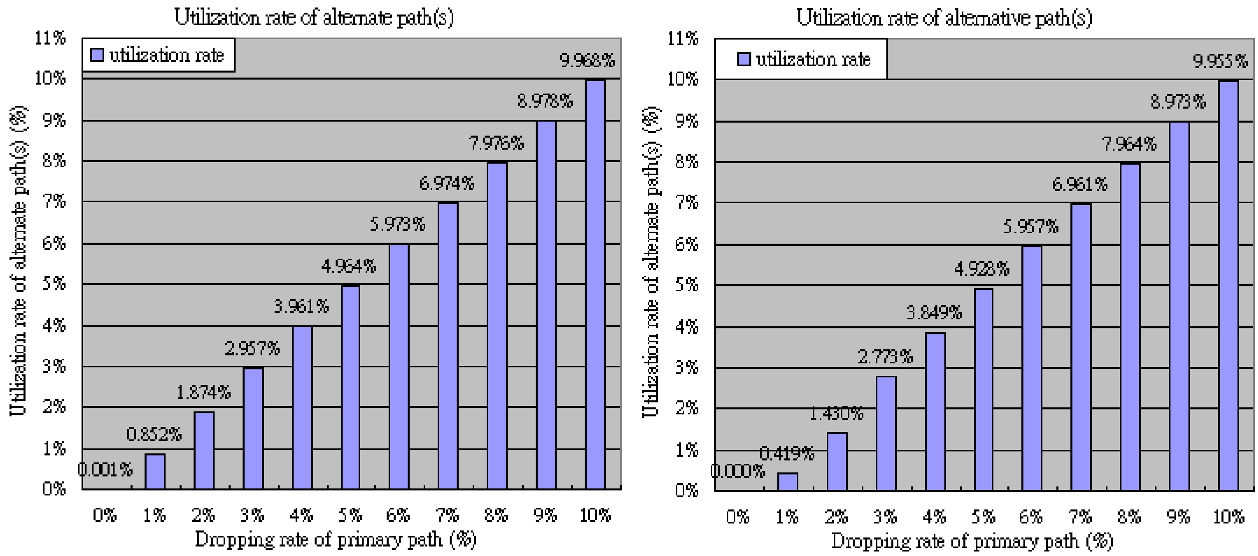
$$U_{primary-path} = \frac{bytes\_transferred\_through\_primary\_path}{number\_of\_bytes\_generated} \qquad (10)$$

and utilization rate of an alternate path is

$$U_{alternate-paths} = 1 - U_{primary-path} \qquad (11)$$

where we assume that the bandwidths of the primary path and alternate path are the same.

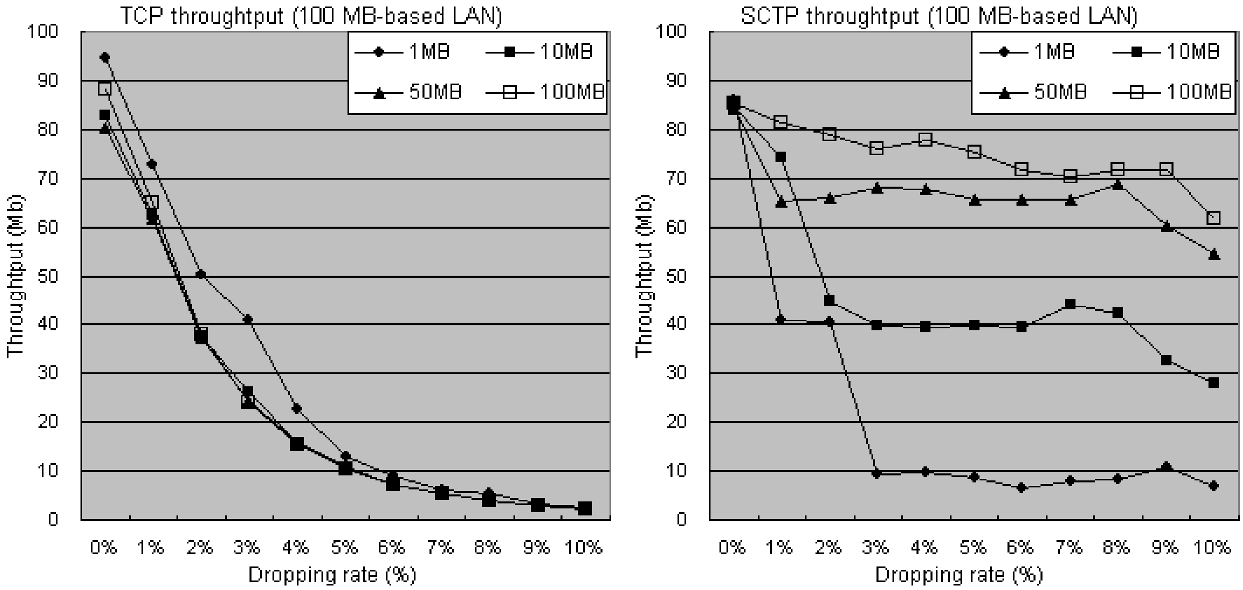
Fig. 6-6 depicts that using SCTP utilization rates of alternate paths are almost equal to the dropping rates of the primary path. The difference between the two rates is resulted from overhead packets, i.e., ACKs. For example, the ideal situation for delivering 410000000 bytes content in a wireless environment with 4% dropping rate by deploying SCTP is that 393600000 bytes are transmitted through primary path and 16400000 bytes via alternate one. Actually, 413159556 and 16538612 bytes are transferred through primary path and alternate path, respectively. That is overhead size is 19618168 bytes and $U_{alternate\text{-}path}$=3.849%. The total sizes of overhead transferred through primary and alternate paths are 19559556 bytes and 138612 bytes, respectively. That is 99.3% instead of 96% of overhead go through the primary path. Hence, the utilization rates of alternate paths are all slightly lower than the utilization rates of primary path.



(a) a wired environment (b) a wireless environment

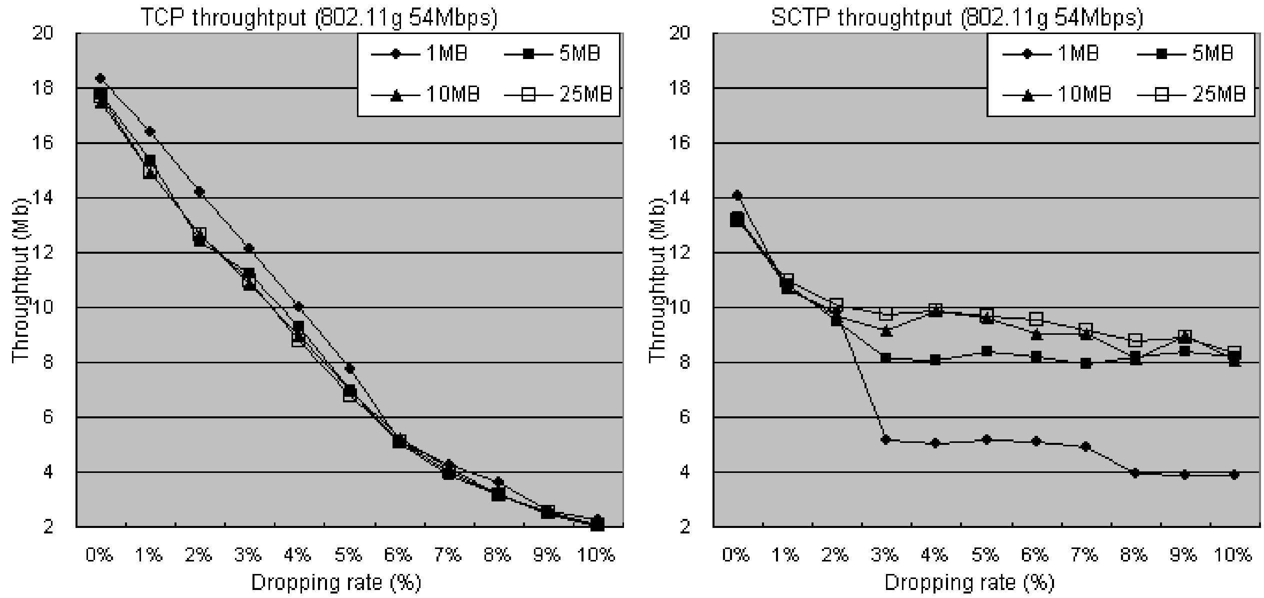
Fig. 6-6 Utilization rates of an alternate path or alternate paths given different dropping rates of primary path

Fig. 6-7 (Fig. 6-8) shows throughput of TCP (SCTP) against dropping rates when a wired (wireless) LAN environment is deployed, where throughput is defined as the amount of successfully delivered contents from Grid node to our mobile client. From these two figures, we can conclude that 1). due to different slow-start mechanisms, sizes of TCP-based service contents transferred have less influence on throughput than those of delivered SCTP-based service contents. The slow-start mechanisms [34] of TCP increase size of a congestion window when receiving an ACK packet, whereas that of SCTP increases its window size based on number of bytes received by opposite site. The number is recorded in a Selective ACK (SACK in short) packet replied. Furthermore, SCTP triggers SACKs by delayed ACK mechanism, which generates an SACK after 4 later packets are received [7]. Basically, it is possible that several SACKs are sent back aggregately after a small file is completely delivered. The window size is then not increased as receiving these delayed SACKs. A slow-start effect is clearly illustrated in Figs. 6-7(b) and 6-8(b). That's also why SCTP throughput slightly falls behind TCP throughput both in drop-free wired and wireless environments [35] (see dropping rate = 0% in Figs 6-7(a), 6-7(b), 6-8(a) and 6-8(b); 2). The decreasing rate of SCTP throughput is more slowly than that of TCP throughput, especially when files are large enough owing to two reasons. The first is that a large file can avoid the slow-start effect. The other is that SCTP resends dropped packets through alternate path immediately rather than via temporarily or continuously malfunctioned primary path. Also, unlike that in a drop-free environment [35], decreasing size of service contents that flow through primary path will slightly decrease the maximum transmission speed of primary path since it also decreasing the probability of increasing window size; 3). The decreasing rates of SCTP throughput on different content sizes in a wireless environment are not as significant as that in a wired one since maximum throughput of a wireless environment is lower than that of a wired one, i.e., SCTP can achieve its full transmission speed earlier and easier than TCP; 4). The difference of SCTP throughputs on large file sizes (e.g., lager than 100MB in wired or 10MB in wireless to avoid slow-start effect) is not very significant if parameters (e.g., dropping rate and physical bandwidth) are themselves the same.



(a) TCP (b) SCTP

Fig. 6-7 Throughput of service/data delivery from a Grid node to a wired client



(a) TCP (b) SCTP

Fig. 6-8 Throughput of service/data delivery from a Grid node to a wireless client

Furthermore, Table 6-1 shows standard deviations of communication costs obtained from a twenty-fold experiment, in which files of 1MB, 10MB, 50MB and 100MB are transmitted in a wired environment, whereas files of 1MB, 5MB, 10MB and 25MB are delivered in a wireless environment. Now, we can conclude that SCTP-based delivery is more stable than TCP-based delivery since SCTP's standard deviations are much smaller than those of TCP, especially when slow-start effect dose not occur.

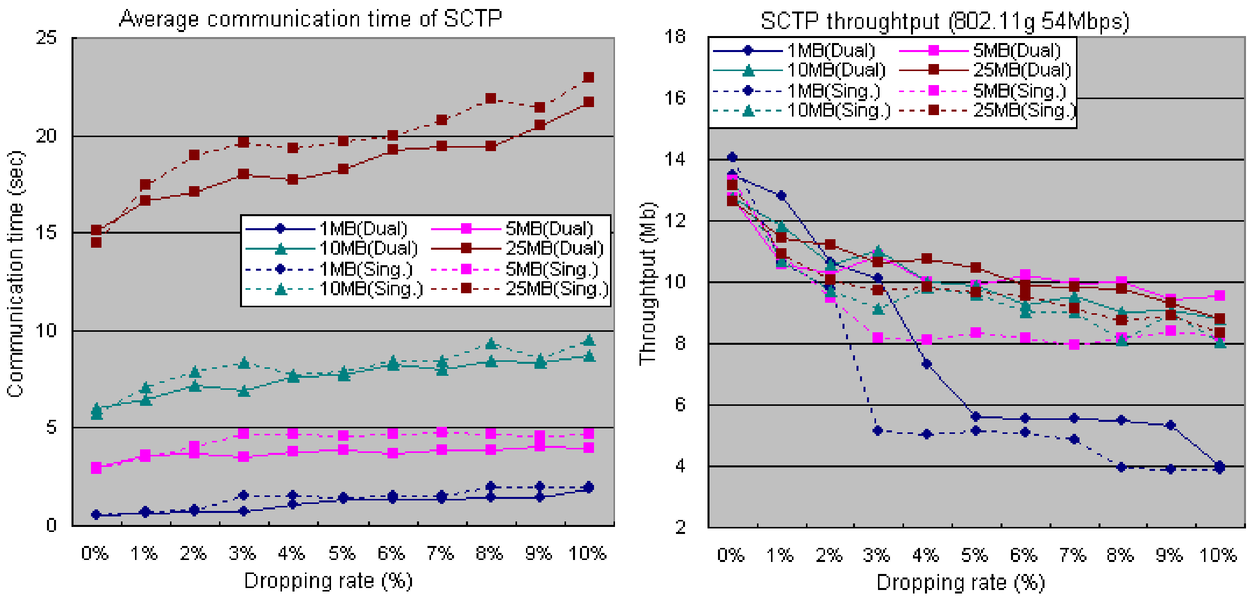
Table 6-1 Standard deviations (SDs) of communication time in wired and wireless environments.

	wired			wireless			
Protocol / Size of delivered content	TCP(ms)	SCTP(ms)	TCP/SCTP	TCP(ms)	SCTP(ms)	TCP/SCTP	Protocol / Size of delivered content
1MB	0.421297	0.370887	1.135917	0.400939	0.383644	1.045080	1MB
10MB	1.461428	0.440808	3.315339	1.108191	0.533513	2.077158	5MB
50MB	2.123679	0.730778	2.906052	1.406420	0.679977	2.068335	10MB

100MB	2.062215	0.810162	2.545435	1.920785	0.689895	2.784170	25MB
Average	1.517155	0.588159	2.579498	1.209084	0.649257	1.862258	Average

6.3. Multi-Streamed Service Delivery

To study communication time and throughput of a multi-streamed service delivery, we developed an application also based on the lksctp library to deliver service content with single-stream and dual-stream. A stream is given a unique ID. Fig. 6-9 shows that in a congested environment multi-streamed/dual-streamed delivery using different outgoing queues has better throughput and shorter communication time than those of a single-streamed also due to the HOL effect.



(a) communication time (b) throughput

Fig. 6-9 Comparison of communication time and throughput of single-streamed and dual-streamed service delivery in a wireless environment

6.4. Service Delivery in Grid Environment

This experiment evaluated the performance of transferring service requests in environments of 1, 2, 4 and 8 Grid nodes. Nodes are homogeneous and connected to a router via 802.11g wireless LAN interfaces. Each node is installed a GSRB module. Each time a client sent its user request, e.g., service A (Svc_A in short) which is a file of 1MB for statistic calculation, through mobile agents to a GSRB on a Grid node. The GSRB then queries the Available Service Index to access information of all serving nodes, evaluates these nodes by using formulas (6) to (9) and redirects the requests to the most suitable one. The node after serving the request returns service results (a file afer statistic calculation which is also 1MB in length) to the client via single or multiple paths. In this experiment, a total of 500 clients are involved. Fig. 10 shows a snapshot of C-Score and R-Score table established by a GSRB for resource broking. The weight variables W_{pwd}, W_{rwd}, W_{pwl} and W_{rwl} in formula (8) and (9) are set to 0.2, 0.8, 0.5 and 0.5, respectively, to calculate T_Scores for all nodes.

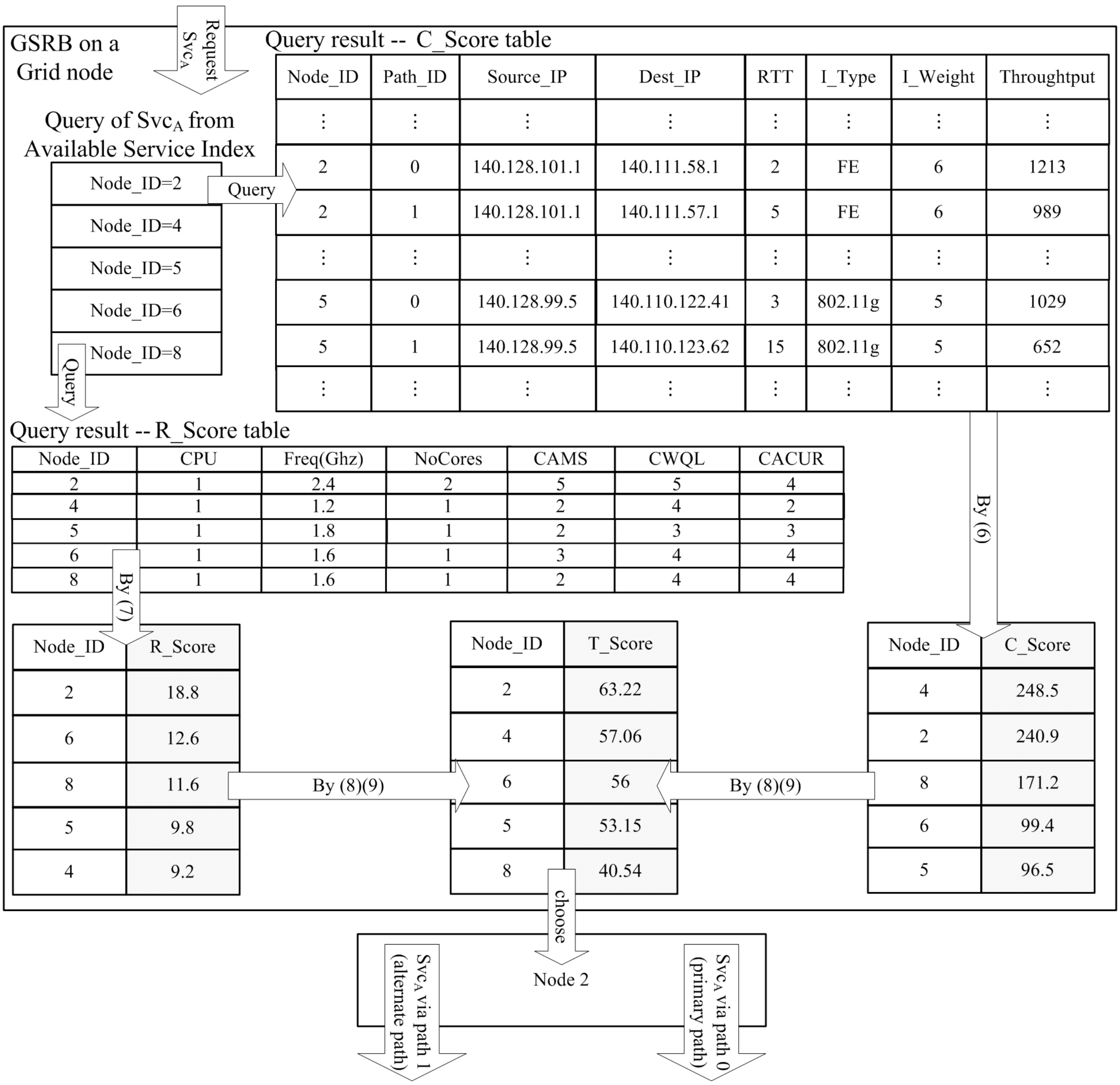
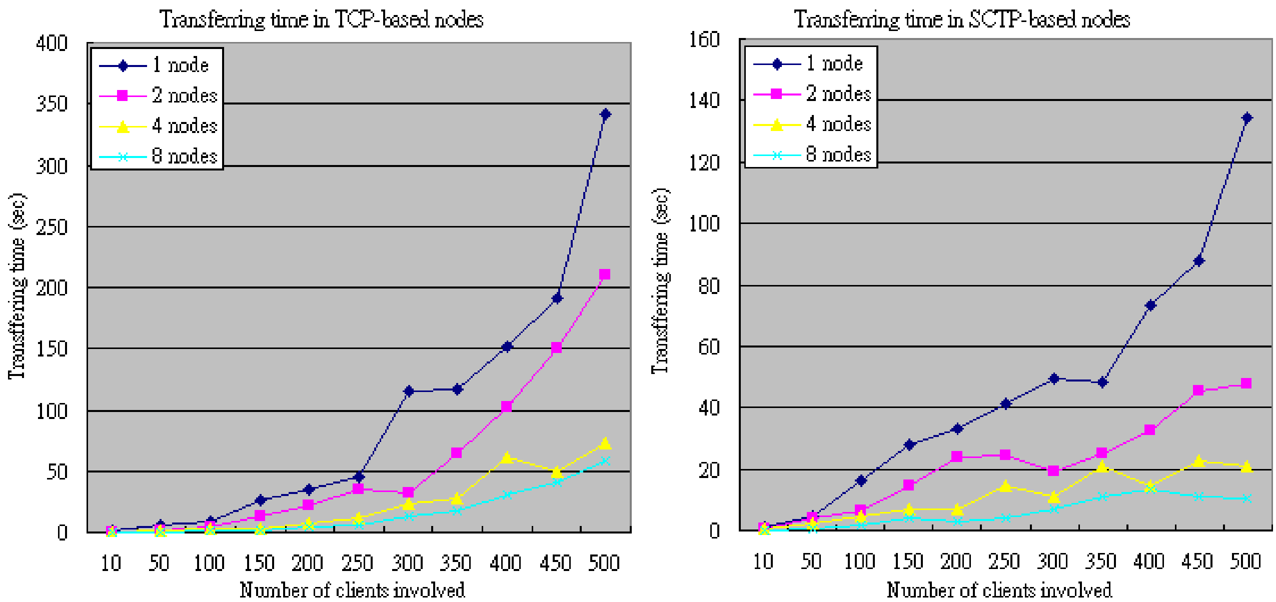


Fig. 6-10 Snapshot of C_Score and R_Score tables in a GSRB during node evaluation process. After querying Available Service Index, R_Score and C_Score of serving nodes are computed by using formulas (6) and (7), respectively, and finally GSRB chooses Node 2 which has the highest T_Score as the serving node by deploying formulas (8) and (9).

The experiment results of transferring the 1MB file from a serving node to a client are shown in Fig. 6-11, in which x-axis represents the amount of clients involved, and y-axis is the time required to serve all clients, including statistic calculation and file transferring. However, in a homogeneous environment, costs of statistic calculation are almost the same. Hence, we omit this portion. Only file transferring time is addressed. Figs. 6.11(a) and 6.11(b) illustrate that 1). the more nodes a Grid environment has, the better service performance and faster transferring time it will be; 2). SCTP-enabled nodes outperformed the TCP-based since users' requests are distributively served by Grid nodes and service results are also distributively sent back to clients through different/parallel links. But SCTP requires at least one alternate path.



(a) TCP-based (b) SCTP-based

Fig. 6-11 The time required to deliver a service file of 1MB from a serving node to a client in a multi-node wireless environment

Chapter 7: Conclusion and Future Work

In this study, we develop a service platform RWGSP, which integrates WLAN, Grid computation, SCTP and mobile agents, to provide users with a more ubiquitous, robust and reliable environment, not only to make Gird resources wirelessly accessible, but also to maximize the resource availability for users. The two key-features of the SCTP, i.e., multi-homing and multi-streaming, support RWGSP with a more stable and reliable communication. GMAP handles all users' states and requests through mobile agents, whereas GSRB deals with service and resource management and brokes suitable resources to serve clients/users. Moreover, we integrate a distributed sendbox scheme with Grid platform so that this platform can deliver agent messages based on SIP to solve communication failure and message chasing problems. While receiver agents are stationary, direct mode is used to handle inter-agent communications. Forward mode is invoked as recipients are unregistered, and messages are forwarded indirectly and distributively by sendboxes. The two modes could eventually delivery all messages to their receivers and were successfully integrated into the GSRB. Instance Response Service and Subscribe-and-Notify Service are implemented based on direct mode and forward mode, respectively.

Besides, we implemented a SCTP-based message delivery testbed to verify the performance gained in a congested environment, and compared the performance of our SIP-based agent mechanism with others existing protocols.

Moreover, we plan to port different feasible SCTP-compatible applications, especially those of mobile computing domain, like location-based service, multimedia and scientific computing to RWGSP and GSRB, and analyze and derive the reliability, cost and performance model of RWGSP in the near future.

References

[1] B. Talwar, P. Venkataram and S. Samuel, “A Method for Finding Resources Using Mobile Agents,” In Proc. of International Conference on Information, Communications and Signal Processing, 2005, pp. 138-142.

[2] C. Badica, M. Ganzha and M. Paprzycki, “Mobile Agents in a Multi-Agent E-commerce System,” In Proc. of International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, 2005, CD-ROM.

[3] Y.H. Li, C.R. Dow, K.H. Chang and K.H. Chen, “Design and Implementation of an Open Sourced Location Service System for the Aglets Mobile Agent System,” In Proc. of International Conference on Integration of Knowledge Intensive Multi-Agent Systems, 2007, pp. 85-90.

[4] C.W. Thompson, “Agents, Grids and Middleware,” IEEE Internet Computing, Sep./Oct. 2004, pp. 97-99.

[5] S. Ilarri, E. Mena and A. Illarramendi, “Location-Dependent Queries in Mobile Contexts: Distributed Processing using Mobile Agents,” IEEE Transactions on Mobile Computing, 2006, pp. 1029-1043.

[6] Internet Engineering Task Force (IETF), “RFC 2960 (Stream Control Transmission Protocol)”, 2000, http://www.faqs.org/rfcs/rfc2960.html.

[7] Internet Engineering Task Force (IETF),”RFC3261 (SIP: Session Initiation Protocol),” 1999, http://www.cs.columbia.edu/sip/drafts.html

[8] I. Foster and C. Kesselman, The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann: San Francisco (CA), 1999.

[9] D. Minoli, A Networking Approach to Grid Computing, John Wiley & Sons: Hoboken, New Jersey, 2005.

[10] H.M. Lee, K.S. Chung, S.H. Chin, J.H. Lee, D.W. Lee, S. Park and H.C. Yu, “A Resource Management and Fault Tolerance Services in Grid Computing,” Journal of Parallel and Distributed Computing, vol. 65, issue 11, November 2005, pp. 1301-1470.

[11] F.Y. Leu, J.C. Lin, M.C. Li and C.T. Yang, "A Performance-Based Grid Intrusion Detection System," In Proc. of IEEE International Conference on Computer Software and Applications, 2005, pp. 525-530.

[12] S.H. Srinivasan, "Pervasive Wireless Grid Architecture," In Proc. of the International Conference on Wireless On-demand Network Systems and Services, 2005, pp. 83-88.

[13] L.W. McKnight, J. Howison and S. Bradner, "Guest Editors' Introduction: Wireless Grids--Distributed Resource Sharing by Mobile, Nomadic, and Fixed Devices," IEEE Internet Computing, vol. 8, 2004, pp. 24-31.

[14] J.Y. Zhou, Z.Y. Jia and D.X. Chen, "Designing Reliable Communication Protocols for Mobile Agents," In Proc. of the International Conference on Distributed Computing Systems Workshops, 2003, pp. 484-487.

[15] M. Ranganathan, M. Beddnarek and D. Montgomery, "A Reliable Message Delivery Protocol for Mobile Agents," In Proc. of the Joint Symposium on Agent Systems and Applications/, Mobile Agents, LNCS 1882, Springer-Verlag, 2000, pp. 206-220.

[16] ObjectSpace Inc., "Objectspace voyager", http://www.objectspace.com/voyager, 1997.

[17] J. Desbiens, M. Lavoie and F. Renaud, "Communication and Tracking Infrastructure of a Mobile Agent System," In Proc. of the International Conference on System Science, vol.7, 1998, pp. 54-63.

[18] X.Y. Feng, et al., "An Efficient Mailbox-Based Algorithm for Message Delivery in Mobile Agent Systems," In Proc. of the IEEE International Conference on Mobile Agents, LNCS 2240, Springer-Verlag, 2001, pp. 135-151.

[19] J.N. Cao, et al., "Design of Adaptive and Reliable Mobile Agent Communication Protocols," In Proc. of International Conference on Distributed Computing Systems, 2002, pp. 471-472.

[20] O. Buyanjargal, J.T. Kim, S.Y. Lee and Y.M. Kwon, "Performance Improvement of Grid Web Services based on Multi Homing Transport Layer," In Proc. of International Conference on Computer and Information Technology, 2006.

[21] H. Kamal, B. Penoff and A. Wagner, "SCTP versus TCP for MPI," In Proc. of ACM/IEEE Conference on Supercomputing, 2005.

[22] H. Kamal, B. Penoff, M. Tsai, E. Vong and A. Wagner, "Using SCTP to Hide Latency in MPI Programs," In Proc. of International Conference on Parallel and Distributed Processing, 2006.

[23] A.A. El Al, M. Lee and T. Saadawi, "Supporting Real-Time Video in SCTP Networks," In Proc. of International Conference on Military Communications, 2006, pp. 1-7.

[24] L. Ma, F. Yu, V.C.M. Leung and T. Randhawa, "A New Method to Support UMTS/WLAN Vertical Handover using SCTP," Wireless Communication, vol. 11, issue 4, 2004, pp. 44-51.

[25] X.L. Wang and V.C.M. Leung, "Applying PR-SCTP to Transport SIP Traffic," In Proc. of International Conference on Global Telecommunications, 2005.

[26] G. Camarillo, R. Kantola and H. Schulzrinne, "Evaluation of Transport Protocols for the Session Initiation Protocol," IEEE Network, vol. 17, no. 5, Sept, 2003, pp 40-46.

[27] Internet Engineering Task Force (IETF), "RFC 3489 (STUN: Simple Traversal of UDP through NATs)", 2003, http://www.faqs.org/rfcs/rfc3489.html.

[28] A.L. Jr. Caro, J.R. Iyengar, P.D. Amer, S. Ladha, G.J. II. Heinz, and K.C. Shah, "SCTP: A Proposed Standard for Robust Internet Data Transport," Computer, 2003, pp. 56-63.

[29] M.T. Jones, "Better Networking with SCTP," 2006, http://www.ibm.com/developerworks/linux/library/l-sctp/.

[30] F.Y. Leu and M.C. Hsu, "A Study on Ontology-based Document Summarization System for Chinese Stock News," Thesis, Department of Computer Science and Information engineering, Tung-Hai University, 2004 (in Chinese).

[31] NIST Internetworking Technology Group, "NIST Net Home Page," 1998-2007, http://www-x.antd.nist.gov/nistnet/.

[32] A.L. Murphy and G.P. Picco, "Reliable Communication for Highly Mobile Agents," In Proc. of International Conference on Agent Systems and Architecture/Mobile Agents (ASA/MA), 1999, pp. 141-150.

[33] "LKSCTP," 2006, http://lksctp.sourceforge.net/.

[34] Internet Engineering Task Force (IETF), "RFC 2001 (TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms)", 1997, http://www.faqs.org/rfcs/rfc2001.html.

[35] R. Fracchia, C. Casetti, C.F. Chiasserini and M. Meo, "WiSE: Best-path Selection in Wireless Multihoming Environments," IEEE Transactions on Mobile Computing, vol. PP, issue 99, 2007 , pp. 1-1.

[36] I. Satoh, "Building Reusable Mobile Agents for Network Management," IEEE Transactions on Systems, Man and Cybernetics, vol. 33, issue 3, 2003, pp. 350-357.

[37] F.Y. Leu, M.C. Li, J.C. Lin and C.T. Yang, "Detection Workload in a Dynamic Grid-based Intrusion Detection Environment," To be appeared in Journal of Parallel and Distributed Computing.