A Qos-Aware Context Construction And Discovery For Mobile Context Services In Next Generation Networks

GJCST Computing Classification
C.1.3 ,C.2.2, C.2,m

[1]R.Nandhagopal and [2]C.Chandrasekar

Abstract- **The significant factors that contribute to the success of the mobile service delivery are the Quality of Service (QoS) and the mobile connectivity. The mobile service infrastructure must be QoS-aware in addition to context-aware. In this paper, we propose to develop a QoS-aware mobile service infrastructure for context construction and discovery. The choice of connectivity and adaptation of application protocol parameters can be intelligently made on the basis of the available information about these offered-QoS. We implement a QoS-predictions server as a part of the QoS and context aware service infrastructure to attain the objectives. In addition, we develop a QoS aware Context Construction protocol. As per the requirements of a specific mobile host mentioned in the context definition, for a set of mobile hosts which are parts of the context defined, a cost effective routing tree is constructed and maintained dynamically. In the given context, only the hosts are employed to carry out Context-sensitive operations through a cooperative effort. By simulation results, we show that our QoS-aware architecture attains ore throughput with low delay in acquiring the service.**

I. INTRODUCTION

A. Providing Services In Mobile Devices

Networks, the developing field we can also say it as blooming field with a boom. As we know the network may be a fixed or wireless, both these networks network together aiming high speed of wired networks and the wider coverage of wireless networks. The development of hardware and protocols of wireless networks pushes higher-level applications and lift the services in integrated networks to a hike mainly when wireless LAN (WLAN) technology becomes increasingly popular for providing IP connectivity and 3G is undergoing deployment stage.With regards to computing power, periphery and storage space, mobile devices are restricted. Hence, they depend on the external sources, just to acquire information or functionality. Static connectivity to resources of particular type won‗t sound reasonable to the mobile devices which operate in changing and dynamic environments. Hence there are possibilities for suitable resource connection at runtime, dynamically and automatically, which are transparent to the user. This can be done by service oriented computing. This approach provides functionalities as stand-alone services. These stand alone services are illustrated by a service offer, it is then published and are automatically discovered and selected by comparing the service request with the existing offers [1].

IT industry usually switches from making products to providing services to provide more flexibility and adaptability. The participants of a service are service provider and consumer. There will be a request from the customer for the service to be provided. If the interface of the service is public, an entity can request for a service. A single entity can provide service to rest of the entities by combining the numerous services. The architecture which relies on the services is known as Service Oriented Architecture (SOA).

The following tasks are performed by the service industry, presently [2]

Service description-Description of the service from the service developer point of view and from the service consumer point of view.

Service discovery-Discovery of new services, matching the user‗s needs.

Service monitoring-Monitoring service behavior, in order to ensure that it matches service specification

Service composition-composition a new service on the basis of several existing services in order to create a new service

B. Mobile Context Services

The context is the information which may be clearly or absolutely obtained. It can be used to distinguish certain aspects involved in a specific application or network service of an entity. A person, a place, a router, a 3G network gateway, a physical link, or a virtual object such as IP sec tunnel, SNMP agent anything in the list may be called as an entity. In order to respond to the highly changing computational environments a context-aware service should be more flexible and self-directed.

The traditional services are delivered by the mobile devices such as mobile phones or PDAs. When there is a change in the mobile user‗s requirements, the mobile services can be modified unambiguously. A context-aware mobile service adapts the current situation of the user. The main intent of the context-aware service is to provide right service at the right time to the user. It is an independent service such that it does not require too much of interaction with a computing device, to help the user.

---

[1] *R.Nandhagopal -Assistant Professor & Head, Department of Computer Applications, Annai Mathammal Sheela Engineering College, Namakkal-637013, Tamil Nadu, India. (Email - nandhagopal67@gmail.com)*
[2] *C.Chandrasekar-* Associate Professor, Department of Computer Science, Periyar University, Salem – 636006, Tamil Nadu, India.
(Email -ccsekar@gmail.com)

There are various types of devices which may range from a wall display to mobile phones. They must be provided with mobile and context-aware services. There are also many variety in their resources like screen size, memory etc. We must also know the services that adapt the host of the resources. The services that are provided should know the user‗s current context and should adjust themselves when the context changes.

C. Challenges Of Mobile Context Services

Suppose a user on his travel needs to send his work through GPRS mobile to his workplace. Of course it is impossible due to the lack of bandwidth of the mobile network. So the user will be in need of a service which can detect a faster wireless network and dynamically change the connection to that network.

In heterogeneous networking environment different providers provide different wireless ok and wired networks. Usually mobile service works in a similar kind of environment. Therefore, a part of the ‗end-to-end‗ communication path between a mobile user and an application server is the responsibility of every network.

Quality of Service (QoS) and the mobile connectivity are the two main significant factors that pay for the success of service delivery. The success can be measured by satisfying the QoS of the user. The current mobile services, will assume that the required-QoS to be static, without bothering the changes in user‗s context, during the service delivery [3].

D. Proposed Work

We propose to develop a mobile service infrastructure with the following objectives:

1. The mobile service infrastructure must be QoS-aware in addition to context-aware. (i.e.) aware of the user‗s required-QoS and the QoS offered by the various networks in user‗s context (location, time, etc).
2. The choice of connectivity and adaptation of application protocol parameters can be intelligently made on the basis of the available information about these offered-QoS.

II. EXISTING WORK

Iris Braun, et al,. have proposed an approach for context and QoS aware discovery namely ConQo. Their approach is the enhancement of semantic service discovery and selection by taking quality of service and contextual information into the account [4].

Shudong Chen, et al,. have presented a context-aware resource management middleware namely VICSDA, for service oriented applications which aims to handle the inherent dynamics of services and the network [5].

Pravin Pawar, et al,. have presented a Mobile Service platform middleware which enables the resource constrained, handheld mobile devices to offer the hosted Nomadic Mobile services in the Internet based on the principles of service Oriented Architecture [6].

Kun Yang and Alex Galis, proposed an all-policy based context-aware service method for next generation networks (NGN).In their approach a thorough consistency was expected to be achieved where policies are well planned to cover from context representation through services down to the underlying networks [7].

Katarzyna Wac, focused on a QoS and context-aware service infrastructure for supporting the development of mobile applications in a heterogeneous network environment. Also they argued that the use of context information helps to better capture the user's required QoS and also improves the delivered QoS [8].

Youngkon Lee [9] has presented the design principle for incorporating quality management on Web service registry developed in UDDI specification and Web service quality management system (WSQMS). They adopted the WSQDL (Web Service Quality Description Language), for representing Web service quality information which is published by WSQM technical committee in OASIS. For quality data and to modify the necessary data structure of the registry, they have also presented the scheme to compose the classification scheme.

Licia Capra, Stefanos Zachariadis and Cecilia Mascolo [10] have presented Q-CAD, a resource discovery framework. In order to discover and select the best satisfying resources which the user requires, it takes the current execution context and quality-of service (QoS} requirements into account which enables persistent computing applications. Only those suitable to the current execution context of the application will be considered and hence the available resources are screened initially. Against the QoS needs of the application the shortlisted resources are evaluated and a binding is established to the best available.

Jose Antonio Parejo, Pablo Fernandez, Antonio Ruiz Cort´es [11] have addressed the optimal QoS-aware selection in composite web services. They have proposed metaheuristic based algorithms such as hybrid genetic algorithm and tabu search.

Rashid J. Al-Ali¤, Omer F. Rana and David W. Walker [12] have extended the service abstraction in the Open Grid Services Architecture for Quality of Service (QoS) properties. Advance or on-demand reservation of resources varies in type and implementation, and independently controlled and monitored. Based on the particular QoS properties, their focus is on the application layer where a given service may indicate the QoS properties it can offer, or where a service may search for other services.

III. SYSTEM REQUIREMENTS

A. Requirements On A Qos Aware Mobile Service Infrastructure

In a context-aware service infrastructure, in order to implement QoS-awareness, the infrastructure should satisfy the following set of requirements

i. The context-aware mapping of user‘s QoS requirements into the requirements on end-to end offered-QoS

ii. Context-aware service infrastructure should support

a) User‘s QoS requirements specification

b) The service delivery adaptation to the end-to-end offered-QoS, to satisfy the user‘s required-QoS.

c) QoS monitoring and real-time measurements of end-to-end offered-QoS and record those measurements to the QoS-predictions server.

In heterogeneous networking environment, the QoS-predictions server is responsible for the predictions of the end-to-end QoS offered.

The QoS-predictions server has these functional requirements

i. The server should obtain and combine the results of real-time end-to-end offered-QoS measurements performed by the mobile users. The quality of the obtained information is estimated by the server.

ii. The obtained data should be transformed into a meaningful form by the server for gathering further QoS information.

iii. The QoS-predictions server must provide generic predictions of end-to-end offered QoS to mobile users. This can be done with the estimation of the quality of the prediction

B. Protocol Fundamentals

In QACCD, QPS specifies the operating context. The details of the hosts in the context are not necessary to be known earlier. Hence, the computation of the context can operate in a purely distributed fashion. In which the requests are responded by sending the reply in the same path in which it came. QACCD is also on-demand process in which, when a QPS send a request then only a shortest path tree is built. Along with this message, the context specification and the information necessary for its computation are embedded.

The Components of a request

Request, req

Req_QPS_id	*QoS-predictions server's id*
Req_seq_num	*application sequence number of req*
Req_forwarder	*sender of this copy of req*
Req_QPS_dist	*distance from the QPS to Req_forwarder*
Req_client_dist	*distance from the QPS to the requesting host*
Req_bound	*cost function bound*
Req_cost	*cost function*
Req_data	*application level data associated with this request*

i. Protocol State Information

Client_id	*host's unique identifier*
Seq_num	*application sequence number*
Dist	*distance from QPS*
Parent	*host's parent in the tree*
Parent_dist	*parent's distance (or cost) from QPS*
bound	*cost function bound*
Cost	*cost function*
P	*set of connected neighbors*
P_s	*a subset of P containing the connected neighbors*

The list of all connected neighbors of a client is included in set P. From the client, each neighbor has a link to it and the weight of that link is stored in P. It is referred to as Wp for some p Є P. If a client receives a request from any nodes in p then it would give it a cost distp < bound. Thus it does not use as its shortest path and it also remembers p‘s cost in P. In finding a new shortest path quickly and to replace a invalid path, this information is much useful.

ii. Context Building

In a request, the protocol maintains no global state instead all the information which are essential arrives in it for computing a context. The context specification and the request are bundled in a request from the application and forwarded to all its neighbors. The Req_bound and Req_cost together define the context. When a request arrives at a server, it includes these and also the cost to this server.

Before sending the request to a node the sending node calculates the destination node‘s cost hence it is guaranteed to be within the context‘s bound. The messages are sent to the neighbors that fall within that bound. Except for the request that offer a lower cost path, successive copies of the same request are ignored.

The following algorithm is used by a server to build the context tree, when a request is received.

Algorithm

```
1. If a request req is received then
   1.1 if req. Seq_num = Seq_num + 1 then
         1.1.1 Cost = Req_cost,
               bound = Req_bound
         1.1.2 remove P
         1.1.3 dist = Req_client_dist,
               Parent = Req_forwarder,
               Parent_dist = Req_QPS_dist
         1.1.4 for each p
               1.1.4.1 if Cost(dist,wp) < bound then
                          send the request to p
                          Req_client_dist = Cost(dist,wp)
                          Req_forwarder = Client_id
                          update $P_s$
               1.1.4.2 End if
         1.1.5 End for
         1.1.6. process the data message of the request
         1.1.7 Seq_num = Req_seq_num
1.2. else if Req_client_dist < dist then
         1.2.1 dist := Req_client_dist,
         Parent := Req_forwarder,
         Parent_dist := Req_QPS_dist)
         1.2.2 repeat from 1.1.4 to 1.1.5
1.3 end if
2.dist Req_forwarder:= Req_QPS_dist
```

In QACCD, when a shorter cost path is found, the cost of the new path, the new parent, and the new parent's cost are all stored in QPS. The request is broadcasted to non-parent neighbors also. The distance of these non-parent neighbors will maintain them inside the context.

Let the non-parent neighbor be np, QPS applies the cost function to its own distance and the weight of the link to np. QPS broadcasts the request to np if the cost of np is less than the bound. This shorter path may allow additional downstream hence a node should broadcast a request with a lower cost even if its application has already processed it from a previous parent. QPS adds the information about the parent to the set P after receiving any request.

When QPS receives a request which is not seen before, the application automatically processes it not considering whether or not it arrived on the currently stored shortest path.

It is possible that the path via the parent may not exists hence QPS need not wait for more additional copies of a request to arrive only from its parent. When QPS receives a new request and if the path does still exist and is still the shortest path, then the request will ultimately come along that path. This may cause the cost to be updated and the effects to be passed to the children. The QPS host sends the data portion of the request to the application for processing.

a. QoS-Predictions Server (QPS)

Consider there are n nodes in which n1 nodes are clients and n2 nodes are servers. Assume that each server contains m services of which k of them with same service. The QoS-predictions server (QPS) maintains the following structures which contain the information of all the clients, servers and the services which are running currently.

The client structure includes

i. client_id
ii. bw (bandwidth)
iii. power
iv. speed
v. capacity

The server structure includes

i. server_Id
ii. servno (Service number)
iii. req_bw (Required Bandwidth)
iv. req_power (Required Power)
v. req_speed (Required Speed)

The service structure includes

i. servno (Service number)
ii. desc (Description)
iii. lifetime

A service request from the client will be of the following form

serv_req (client_id, servno)

Algorithm

```
1. Client C1 sends serv_req (client_id, servno) to QPS
2.  For each server {Si}, (i=1, 2…N2)
      2.1 If (servno = Si.servno) Then
              3.1.1 Add Si into the set M.
      2.2 End If
3.  End For
4.  If M <> NULL Then
4.1 For each server {Sk}, (k=1,2…M)
      4.1.1 If (C1.bw >= Sk.req_bw) and
            (C1.power >= Sk.req_power) and
                  (C1.speed >= Sk.req_speed) Then
            4.1.1.1 return the server Sk as the QOS aware server
      4.1.2 End If
    4.2 End For
5.  End If
6.  QPS forward the req (client_id, servno) to Sk
7  Sk processes the request and sends reply to C1.
```

The client sends a service request with its id and service number to QPS. Then QPS searches the service number in the servers. QPS then add the matching servers in a set M. Then QPS find the QOS aware server with matching QOS constraints bandwidth, power and speed. Then QPS forwards the request to QOS aware server. Finally QOS aware server process the service request and sends reply to client.

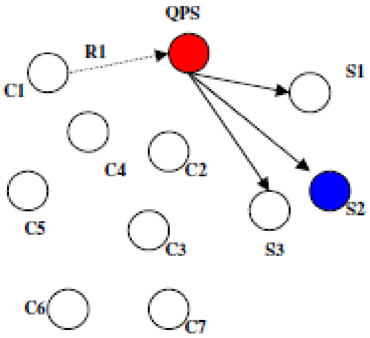

Fig 1.a

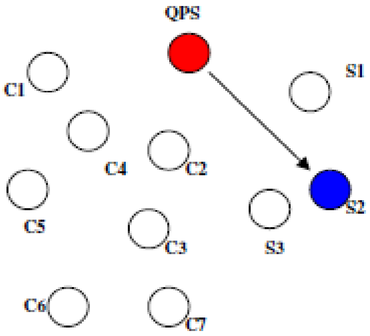

Fig 1.b

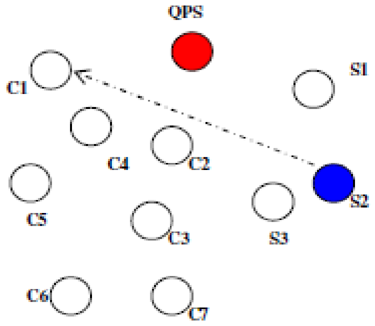

Fig 1.c

In diagram 1.a, the client c1 sends a service request to QPS. QPS forwards the request to all the servers (s1, s2, s3). In diagram 1.b, the service request will be forwarded to QOS aware server s2. In diagram 1.c, the server s2 send reply to client c1.

IV. SIMULATION RESULTS

A. Simulation Model And Parameters

We use NS2 to simulate our proposed protocol. In our simulation, the channel capacity of mobile hosts is set to the same value: 2 Mbps. We use the distributed coordination function (DCF) of IEEE 802.11 for wireless LANs as the MAC layer protocol. It has the functionality to notify the network layer about link breakage

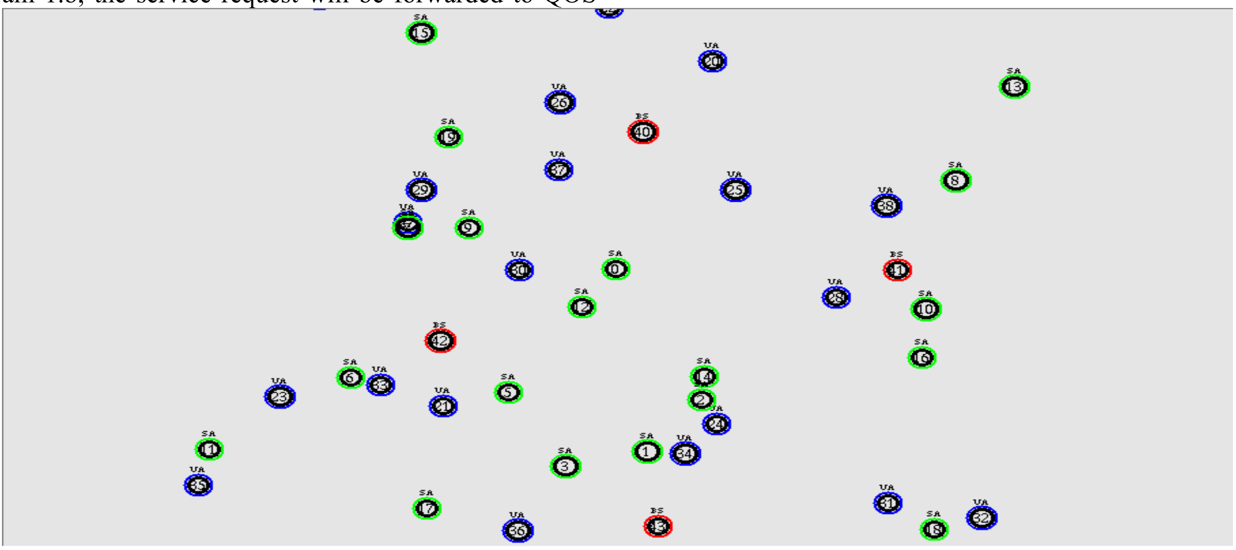

Fig 2: Network Topology

Figure 1 gives the sample network topology used in our simulation. In our simulation, 40 mobile nodes move in a 1000 meter x 1000 meter region for 100 seconds simulation time. Among the total 40 nodes, we treat 20 nodes as clients and 20 nodes as servers. There are 4 base stations to handle the queries of the clients.

We assume each node moves independently with the same average speed. All nodes have the same transmission range of 250 meters. In our simulation, the speed is 10 m/s. We have taken the Service Location Protocol (SLP) for service discovery. A SLP service agent is attached to the servers for providing the services and SLP user agent is attached to the clients for requesting the service. We have used the network abstraction model for the routing process.In our simulation, 4 clients send service requests to the server through the base station. Our simulation settings and parameters are summarized in table 1

Table1: Simulation Parameters

No. of Clients	20
No. of Servers	20
Base stations	4
Area Size	1000 X 1000
Mac	802.11
Radio Range	250m
Simulation Time	100 sec
Service Discovery Protocol	SLP
Server Application	SLPsa
Client Application	SLPua
Speed	10m/s
clients	1,2,3 and 4
Routing Protocol	NETABS

B. Performance Metrics

We compare our QoS-aware service infrastructure with a non-QoS aware service infrastructure. We evaluate mainly the performance according to the following metrics.

Average Delay: It is measured as the average delay occurred for each client while getting the requested service.

Average Throughput: It is measured as the throughput for each client in terms of Mb/sec.

C. Results

i. Based On Rate

In our first experiment we vary the requested traffic rate as 100 to 500 kb and measure the throughput and delay for each client.

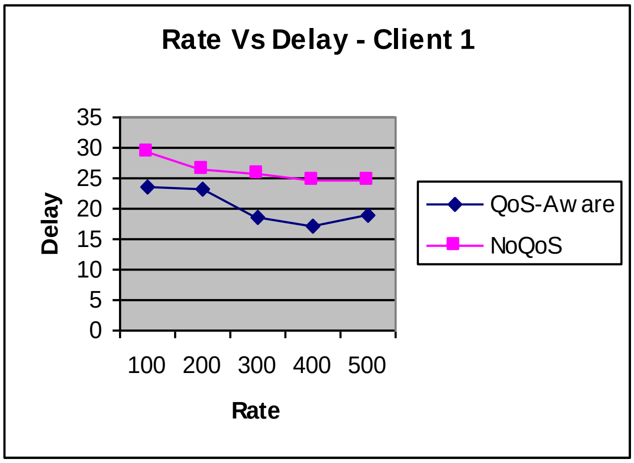

Fig 3: Rate Vs Delay – Client 1

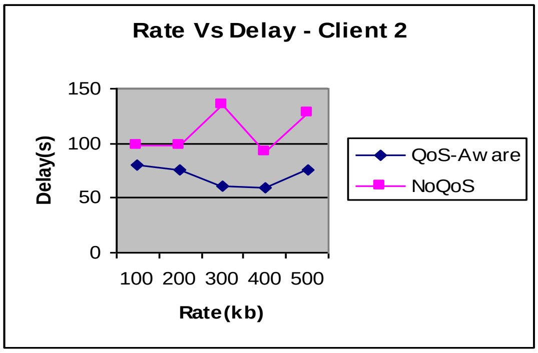

Fig 4: Rate Vs Delay – Client 2

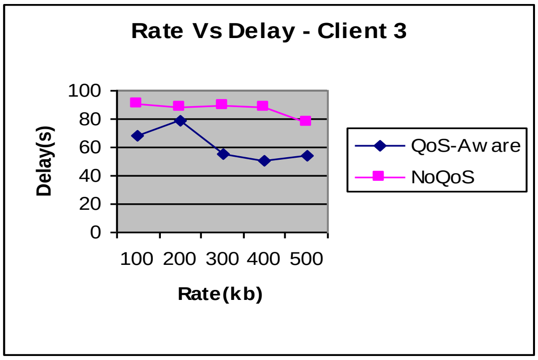

Fig 5: Rate Vs Delay – Client 3

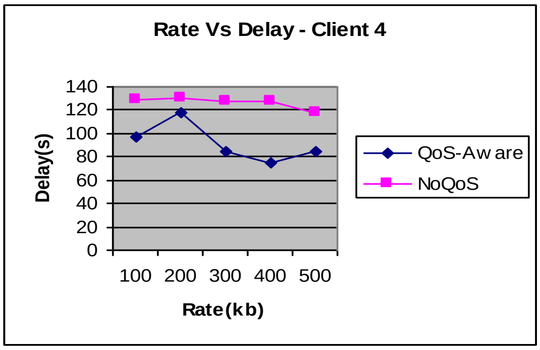

Fig 6: Rate Vs Delay – Client 4

Figures 3 to 6 show the delay occurred for the clients 1 to 4 using the QoS-Aware and NonQos-Aware schemes respectively. As we can see from the figure, the delay is significantly less for QoS-Aware, when compared to NonQos-Aware.

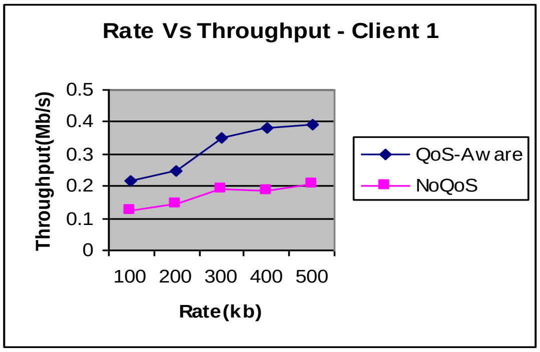

Fig 7: Rate Vs Throughput – Client 1

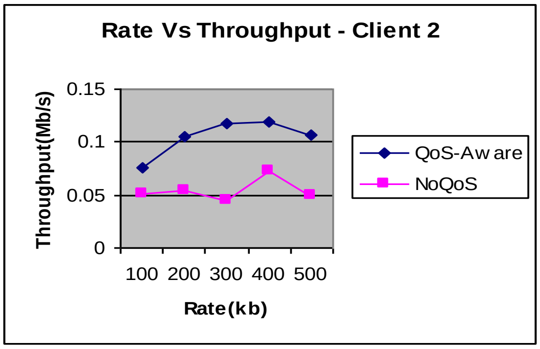

Fig 8: Rate Vs Throughput – Client 2

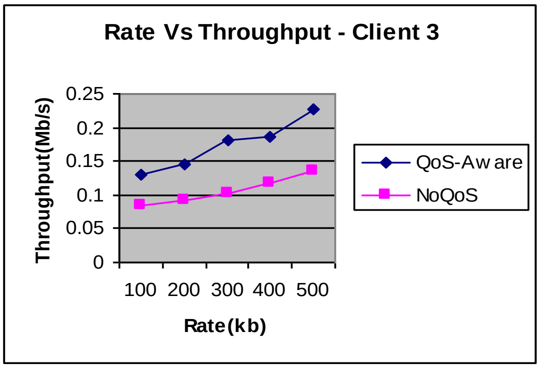

Fig 9: Rate Vs Throughput – Client 3

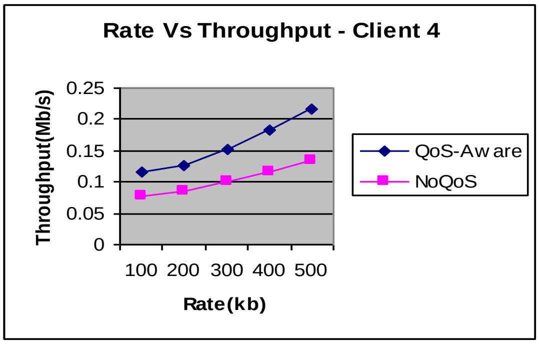

Fig 10: Rate Vs Throughput – Client 4

Figures 7 to 10 show the throughput achieved by the clients 1 to 4 using the QoS-Aware and NonQos-Aware schemes respectively. As we can see from the figure, the throughput is significantly high for QoS-Aware, when compared to NonQos-Aware.

ii. Packet Size

In our second experiment we vary the requested query size as 100 to 500 bytes and measure the throughput and delay for each client

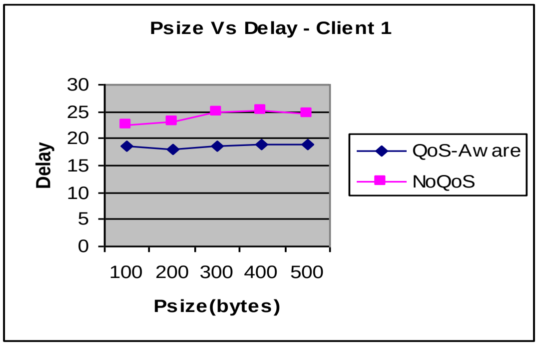

Fig 11: Psize Vs Delay – Client 1

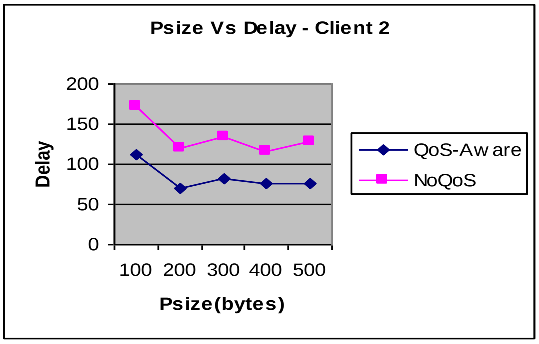

Fig 12: Psize Vs Delay – Client 2

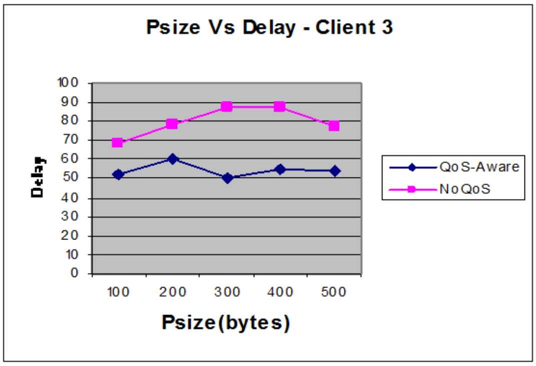

Fig 13: Psize Vs Delay – Client 3

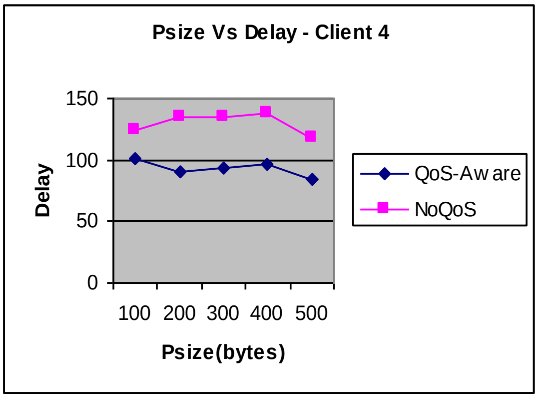

Fig 14: Psize Vs Delay – Client 4

Figures 11 to 14 show the delay occurred for the clients 1 to 4 using the QoS-Aware and NonQos-Aware schemes respectively. As we can see from the figure, the delay is significantly less for QoS-Aware, when compared to NonQos-Aware.

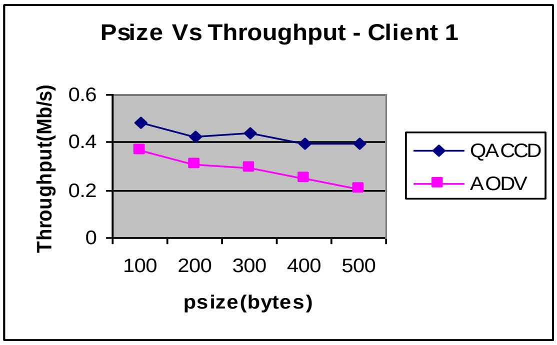

Fig 15: Psize Vs Throughput – Client 1

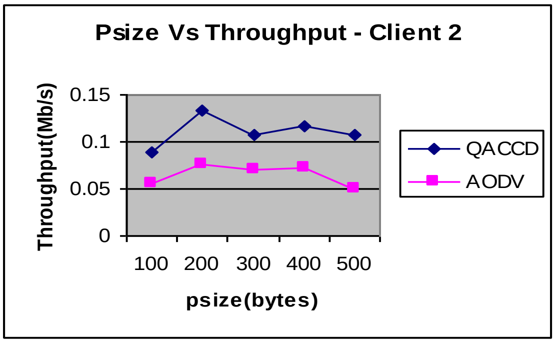

Fig 16: Psize Vs Throughput – Client 2

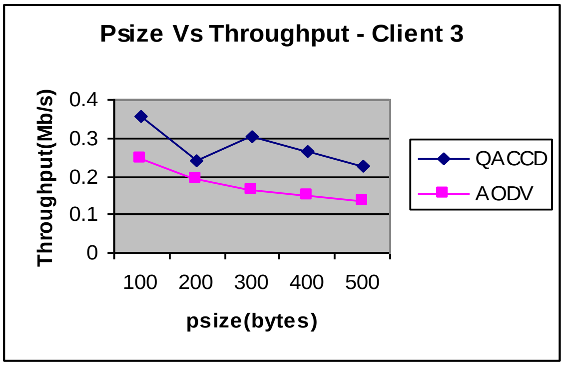

Fig 17: Psize Vs Throughput – Client 3

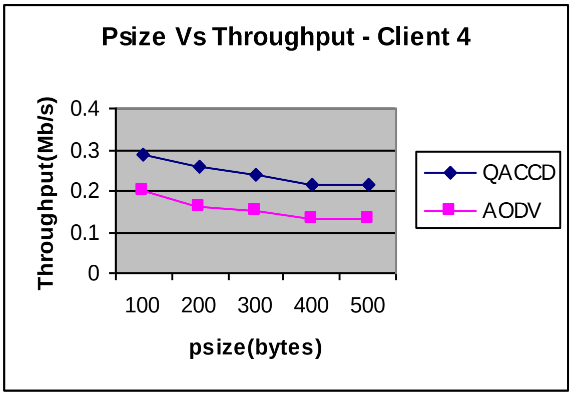

Fig 18: Psize Vs Throughput – Client 4

Figures 15 to 18 show the throughput achieved by the clients 1 to 4 using the QoS-Aware and NonQoS-Aware schemes respectively. As we can see from the figure, the throughput is significantly high for QoS-Aware, when compared to NonQos-Aware.

V. CONCLUSION

In this paper, we have proposed a mobile service infrastructure which is a QoS-aware in addition to context-aware. The choice of connectivity and adaptation of application protocol parameters can be intelligently made on the basis of the available information about these offered-QoS. We have implemented *QoS-predictions Scheme* as a part of the QoS and context aware service infrastructure to attain the objectives. On the basis of historical end-to-end offered-QoS, generic predictions on the offered-QoS are provided by this scheme. In addition, we have developed a QoS aware Context Construction protocol. As per the requirements of a specific mobile host mentioned in the context definition, for a set of mobile hosts which are parts of the context defined, a cost effective routing tree is constructed and maintained dynamically. In the given context, only the hosts are employed to carry out Context-sensitive operations through a cooperative effort. The source node only needs to broadcast the message over the nodes in the routing tree in order to send a message to only the members of the context. By simulation results, we have shown that our proposed architecture has less service delay and attains more throughput.

VI. REFERENCES

1) Friederike Klan,” Context-aware service discovery, selection and usage”, in proc. of 18th GI-Workshop on the Foundations of Databases, pp: 95-99, June 2006
2) Julien Pauty, Davy Preuveneers, Peter Rigole, Yolande Berbers, "Research Challenges in Mobile and Context-Aware Service Development", in proc. of workshop on Research Challenges in Mobile and Context-Aware Service Development,pp:141-148,April 2006
3) Katarzyna Wac, Aart van Halteren, and Dimitri Konstantas,"QoS-Predictions Service: Infrastructural Support for Proactive QoS- and Context-Aware Mobile Services", in proc. of Workshop on Context Aware Mobile Systems, vol.4278, pp: 1924-1933, November 2006
4) Iris Braun, Anja Strunk, Gergana Stoyanova, Bastian Buder,"ConQo – A Context- And QoS-Aware Service Discovery”, in Proceedings of WWW/Internet, 2008
5) Shudong Chen, Johan Lukkien, Richard Verhoeven, Pim Vullers, Goran Petrovic,"Context-Aware Resource Management for End-to-End QoS Provision in Service Oriented Applications”, in proc. of workshop on Service Discovery and Composition in Ubiquitous and Pervasive Environments,pp:1-6, Nov.30-Dec. 4 2008
6) Pravin Pawar, Bert-Jan van Beijnum, ―Cotext-Aware Mobile Service Platform for Vertical Handover Support to Nomadic Mobile Services”, in proc Of ACM SIGMOBILE 9th Workshop on Mobile Computing Systems and Applications (HotMobile 2008), February 2008
7) Kun Yang, Alex Galis, Chris Todd,” Policy-driven Mobile Agents for Context-aware Service in Next Generation Networks", in proceedings of International Workshop on Mobile agents for Telecommunication Applications, vol.2881, pp:111-120, October 2003
8) Katarzyna Wac,"Towards QoS-awareness of Context-aware Mobile Applications and Services", in proc. of OTM Workshops, ph.D Symposium, 2005
9) Youngkon Lee,” QoS Management for SOA by Synchronizing Quality Context in UDDI”, International Journal of u- and e- Service, Science and Technology, June 2009
10) Licia Capra, Stefanos Zachariadis and Cecilia Mascolo,” Q-CAD: QoS and Context Aware Discovery Framework for Mobile Systems”, IEEE, 2005
11) Jose Antonio Parejo, Pablo Fernandez, Antonio Ruiz Cort´es,” QoS-Aware Services composition using Tabu Search and Hybrid Genetic Algorithms”, SISTEDES, 2008
12) Rashid J. Al-Ali¤, Omer F. Rana and David W. Walker,” G-QoSM: Grid Service Discovery Using QoS Properties”
13) Le-Hung Vu, Fabio Porto, Karl Aberer, Manfred Hauswirth,” An Extensible and Personalized Approach to QoS-enabled Service Discovery”
14) Dongyan Xu, Klara Nahrstedt, Duangdao Wichadakul,” QoS-Aware Discovery of Wide-Area Distributed Services”