NETSAQ: Network State Adaptive QoS Provisioning for MANETs

Shafique Ahmad Chaudhry, Faysal Adeem Siddiqui, Ali Hammad Akbar and Ki-Hyung Kim[1]

Division of Information and Computer Engineering, Ajou University, Korea
{shafique, faysal, hammad, kkim86}@ ajou.ac.kr

Abstract. The provision of ubiquitous services in Mobile Ad-hoc Networks (MANETs) is a great challenge, considering the bandwidth, mobility, and computational-resources constraints exhibited by these networks. Incorporation of modern delay-sensitive applications has made the task even harder. The traditional Quality of Service (QoS) provisioning techniques are not applicable on MANETs because such networks are highly dynamic in nature. The available QoS provisioning algorithms are either not efficient or are embedded into routing protocols adding a high computation and communication load. In this paper, we propose a Network State Adaptive QoS provision algorithm (NETSAQ) that works with many underlying routing protocol. It ensures the QoS provisioning according to the high level policy. NETSAQ is simple to implement yet minimizes the degradation of the best effort traffic at a considerable level. Our simulation results show that NETSAQ adapts well in MANET environments where multiple services are contending for limited resources.

Keywords: Mobile ad-hoc networks (MANETs), QoS, Network state Adaptive, QoS routing, Policy-based QoS provisioning

1 Introduction

The role of Mobile Ad-hoc networks (MANETs) [1], in realization of a ubiquitous world, is manifold. MANET devices can serve as user interface, provide a distributed data storage, or act as a mobile infrastructure access point for other nodes. The MANETs are infrastructure-less networks that are envisioned to be spontaneously created whenever two or more nodes come in close proximity to each other. MANETs are characterized by dynamic topologies, limited processing and storage capacity, and bandwidth constrained wavering capacity links.

The described inherent characteristics of MANETs implicate newer requirements and technical challenges for the management of such networks. The incorporation of modern real time services, like transfer of audio and video concomitant to delay-agnostic services has further increased the management complexity. On one hand,

[1] Corresponding author

MANET nodes have limited resources to share amongst contending services, on the other hand such services need and expect high priority on the network in order to meet higher user expectations about their reliability and quality.

In order to meet QoS requirements, network managers can attempt to negotiate, reserve and hard-set capacity for certain types of services (hard QoS), or just prioritize data without reserving any “capacity setting” (soft QoS). Hard QoS can not be provided in ad hoc networks due to their dynamic nature. Therefore soft QoS [2] is provided in ad hoc networks using the QoS routing protocols and the IP Differentiated Services (DiffServ) framework. Mechanisms such as Integrated Services (IntServ) that emphasize on flow reservation cannot be implemented per se, in ad hoc networks because of the resource limitations and dynamic network conditions. DiffServ that provides aggregated classes of services may be a possible solution but necessitates some adaptation in order to be applicable in a completely dynamic topology.

Many QoS routing protocols have been proposed for wireless networks. One of the commonalities of these studies is that these routing protocols reserve the resources dynamically from source to destination before the application flow starts. If route breaks due to mobility, same procedure is repeated before a new route is established. Traditional MANET routing protocols do repair the broken path automatically but QoS violations can still occur even if the path is not broken due to mobility and interference etc. Very few QoS routing protocols for MANETs have considered this phenomenon such as [3]-[7]. The QoS routing protocols for MANETs are actually modifications of existing routing protocols to support QoS functionality. As a result, these QoS routing protocols utilize a lot of MANETs’ resources and are heavier than traditional protocols [8].

In this paper we propose Network State Adaptive QoS provisioning algorithm (NETSAQ) that provides soft QoS without making resource reservations. It implements the high-level policies for QoS provisioning. NETSAQ is independent of routing protocols and can integrate with most of the routing protocols. On one hand it provides the high level control in form of policies and on the other hand it is independent of underlying routing protocol. Our simulation results show that its performance is a compromise between QoS routing and normal (best-effort) routing. NETSAQ avoids the degradation of best effort traffic, which is observed in many QoS routing schemes, while still providing minimum QoS guarantees for applications.

2 Related Work

Ensuring QoS through routing is relatively a new issue when it comes to MANETs. QoS routing protocols aim to search for routes with sufficient resources to satisfy initiating applications. The QoS routing protocols work closely with the resource management module in order to set up paths through a list of mobile nodes to meet the end-to-end service requirements in terms of bandwidth, delay, jitter, and loss, etc. The computational and communication cost of QoS routing is known to be fairly high and it has raised the questions whether or not should it be tackled in MANETs.

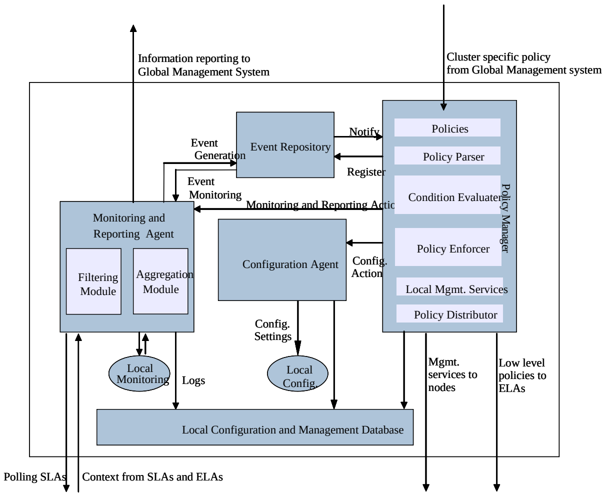
Many proposals have been made for QoS provisioning in MANETs. The In band Signaling (INSIGNIA) [3], Stateless Wireless Ad hoc Networks (SWAN) [4], Core Extraction Distributed Ad Hoc Routing (CEDAR) [5], Adaptive Source Routing (QoS-ASR) [6], and Quality of Service for Ad hoc Optimized Link State Routing Protocol (QOLSR) [7] are examples of QoS-routing proposals. We have summarized various parameters to have an insight into the important features provided by these protocols. Table 1 describes the distinguishing features of these protocols.

Table 1. Comparison with QoS routing protocols

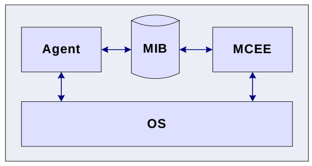
	QOLSR	**QoS-ASR**	**INSIGNIA**	**SWAN**	**NETSAQ**
QoS type	Fixed	Fixed	Dynamic (degrade to best effort)	Fixed	Dynamic (degrade to min QoS)
QoS parameters	Delay, bandwidth	Delay, bandwidth, packet loss	Bandwidth	Bandwidth	Delay, bandwidth, packet loss
Admission Control	Yes	Yes	No	Yes	Yes
Resource Reservation	Priori soft and posteriori hard reservation	Priori hard reservation	Posteriori soft reservation	Priori hard reservation	Posteriori hard reservation
Complexity	High	Highest	Low	High	Low
Application level support	No	No	Yes	No	Yes
Nodes doing rerouting due to interference	Source and intermediate nodes	Source	Source and intermediate nodes	Source	Source
Nodes measuring QoS	Source, destination and intermediate nodes	Source, destination and intermediate nodes	Source, destination and intermediate nodes	Source, destination and intermediate nodes	Source and destination
QoS violation rerouting	Path break and path degradation	Path break and path degradation	Path break	Path break and path degradation	Path break and path degradation
Best effort traffic drop probability	High	Highest	Lowest (admitted)	Low (rate decreased gradually)	Low (Adaptive QoS)

3 System Model and Assumptions

The system model for NETSAQ can best be implemented by an autonomous policy based network management architecture that provides a multi-tier management [9]. In this work we have proposed an architecture for cluster based MANETs. Each cluster head is equipped with a Domain Policy Manager (DPM). The DPM is responsible to monitor the overall status of the MANET; making, updating and distributing the network wide policies for the cluster and correspondingly deploy policies on the MANET nodes. The DPM's architecture is shown in fig.2 (a). For simple MANET nodes, which may exhibit great degree of heterogeneity, are assigned an extended local agent (ELA) or a simple local agent (SLA). These local agents are responsible to implement the policies on the nodes they are deployed at. The components of an ELA are shown in fig. 2(b).



(a) DPM architecture for a MANET cluster head



(b) ELA for normal MANET node

Legend

MCEE : Mobile code execution environment
MIB : Management information base
SLA : Simple local agent
ELA : Extended local agent
DPM : Domain policy manager

Fig. 1. Management components for cluster head and normal MANET node

Many QoS policies are defined for various services. Each policy can be associated with specific types of applications or users or even to a specific user using specific application.

In order to provide QoS to specific application, NETSAQ algorithm is executed and dynamic QoS is provided to the specific traffic according to the policy. The application specifies a minimum acceptable and maximum allowed QoS values called b_{min} and b_{max} respectively. During adverse network conditions, at least b_{min} is satisfied but if the network state is good then a higher level of QoS is provided. Thus QoS depends upon the network state and is automatically adjusted. It is assumed that the nodes exchange information like packet loss ratio, delay and available bandwidth when building and maintaining their routing tables. This can be done through *hello messages* which most of the protocols exchange for making and updating their routing tables.

3.1 Node Mobility and Interference Scenario

Consider a network that provides services to users that belong to different groups such as special and ordinary groups. Now a MANET special user starts a video conferencing application. The QoS of the application can be violated due to: a) *mobility,* whenever the source, the destination or any intermediate node moves, the established QoS path can be broken, b) *interference or contention,* whenever the number of nodes around the QoS path increase considerably, interference increases and contention against wireless channels escalates. In both the situations there is a possibility of QoS degradation even if the resources were reserved beforehand. In either case, the alternative path is needed which can facilitate the required QoS provisioning. This phenomenon is inherent to MANETs, and cannot be eliminated.

4 Network State Adaptive QoS Provisioning Algorithm

In a network, that renders services to multiple user applications, some applications may spell their QoS. We assume that, any application that starts has certain QoS requirements with minimum and maximum QoS bandwidth constraints b_{min} kbps and b_{max} kbps. In order to admit this application into the network, the average data rate of the network is checked. A bandwidth threshold ***s***, based on network media access delay ***D*** and Packet Loss Ratio (***PLR***) is calculated for the application at the initiating node of the application. The resources are then marked along the path based on ***s***, ***i***, and if such bandwidth is not available then we use b_{min}. Once all the links along the path, from source to destination are covered, then we start the application flow. After the flow is established, resources can be reserved on intermediate nodes according to threshold ***s***. This, posteriori reservation, is different from reserving the resources before starting the flow (priori reservation). Posteriori reservation eliminates the complexity and delay which is inherent to priori reservation and provides better response time to the application user. The application flow is re-routed if the end-to-end bandwidth falls below the reserved resources for more than a specific time. The algorithm is depicted in fig. 4.

Lets ***i*** kbps be the initial averaged data rate over ***t*** sec

Set a threshold ***s*** kbps using network media access delay ***D*** and packet loss ratio ***PLR***

$$s = max\ [\ \{1- 0.5\ (w_1\ D + w_2\ PLR)\}\ b_{max}\ ,\ b_{min}\ ]$$

(Where w_1 and ***w2*** are the weights associated to ***D and PLR***)

Reserve the bandwidth along the path on every node equal to ***s*** kbps

If ***s*** kbps bandwidth can't be reserved, reserve ***i*** kbps

If even ***i*** kbps bandwidth is not available, reserve b_{min} kbps

Reroute if the end-to-end QoS falls below the reserved resources for more than ***t*** sec

Calculate value of ***s*** before rerouting

New route is found with resources greater than or equal to ***s*** kbps

If no such path exists, find a path with resources greater than or equal to ***i*** kbps (only if ***s > i***)

In the worse case, find a path with resources greater than or equal to b_{min} kbps

Fig. 2. NETSAQ algorithm

If the established path is broken due to link failure or mobility of any node(s), rerouting is done at the node where the path is broken. If this intermediate node can't find the QoS path, the next upstream node tries to find the path. If no QoS path can be established from the upstream node as well, source is notified to broadcast a new route request with b_{min} kbps and b_{max} kbps QoS constraints.

If the data rate for the application decreases than the reserved bandwidth for more than *t* sec, the destination will notify the source. It means all the intermediate nodes do not need to continuously monitor the data rate violations. Also, in case of interference, the source has a greater possibility to find new disjoint routes. The routing agent at the source will then be invoked to find a new route with data rate > ***s*** kbps or > ***i*** kbps or > $\boldsymbol{b_{min}}$ kbps as it would be described in the policy.

MANET routing protocols may use link level acknowledgement (ACK) messages and a timeout period for link level connectivity information. In an area with dense population of nodes, hidden terminal problem can become quite significant. Due to hidden terminal problem and high contention, some nodes will not receive the link layer ACK packets from the neighboring nodes. When the timeout period expires, a node declares the link as broken, discards all ongoing communication packets and generates a route error message [10]. This causes the throughput to drop drastically. The communication resumes when a new path is found or the same path is eventually re-discovered. This instability problem is caused by fast declaration of link failures which is rooted at the link layer. The breaking and rediscovery of the path result in the drastic throughput oscillations. In order to avoid this problem, we extend the solution proposed in [11]. This solution uses a "don't-break before-you-can-make" strategy. This strategy is based on modifying the routing algorithm so that the routing agent continues to use the previous route for transmissions before a new route can be found. When the new route is found or the same route is eventually re-discovered, all nodes discard the previous route and switch to the new one (or the same one) for transmissions. An example for the explanation is shown is fig.3.

Let b_{min} = 50 kbps, b_{max} = 100 kbps, t = 1 min, w_1D = 0.2, w_2PLR = 0.3 and i=65 kbps

Then threshold s:

s = max [{1- 0.5 (w1 D + w2 PLR)} b_{max}, b_{min}]

s = max [{1- 0.5 (0.2 + 0.3)} 100, 50] = 75 kbps

Reserve the resources along the path equal to 75 kbps (*s*)

Else reserve the resources equal to 65 Kbps (*i*)

Otherwise reserve the resources equal to 50 Kbps (b_{min})

If resources are reserved equal to 75 kbps

  when destination node receives data rate < 75 kbps for 1 min

  It will notify the source

  Source will establish a new route with data rate > *s* kbps

  If no such route is available, a route with data rate > *i* or minimum bandwidth b_{min} condition will be set up

If resources are reserved equal to 65 kbps

  when destination node receives data rate < 65 kbps for 1 min

  It will notify the source

  Source will establish a new route with data rate > *s* kbps

  If no such route is available, a route with data rate > *i* or minimum bandwidth b_{min} condition will be set up

If resources are reserved equal to 50 kbps

  when destination node receives data rate < 50 kbps for 1 min

  It will notify the source

  Source will establish a new route with data rate > *s* kbps

  If no such route is available, a route with data rate > *i* or minimum bandwidth b_{min} condition will be set up

Fig. 3. A numerical example for NETSAQ algorithm working

5 Performance Evaluation

The simulations is this section evaluate the suitability of the algorithm to support adaptive flows in a MANET under various conditions such as traffic, mobility, and channel characteristics. In particular, we evaluated system wide adaptation dynamics and the impact of threshold based rerouting mechanisms and mobility on end-to-end sessions.

We used OPNET simulator and the simulation environment consists of 20 Ad-hoc nodes in an area of 500m x 500m. Each mobile node has a transmission range of 300m and shares a 5.5 Mbps air interface between neighboring mobile nodes. The nodes follow the standard random waypoint mobility model with the maximum speed of 10m/s used. The underlying routing protocol is Dynamic Source Routing (DSR). We have used various application flows having different bandwidth requirements ranging from 90 Kbps to 320 Kbps. An arbitrary number of best effort flows are randomly generated to introduce different loading conditions distributed randomly throughout the network. These loading flows are dynamic in nature which helps analyze the adaptive behavior of NETSAQ.

We measure per-session and aggregate network conditions for a number of experiments that analyze adaptation, threshold based rerouting, and nodes mobility. We observe throughput, delay, packet loss, rerouting frequency and degradation, as the measures of system dynamics during the course.

5.1 Adaptive Flows

We measure the performance of two adaptive flows with User Datagram Protocol (UDP) and Transmission Control Protocol (TCP). Fig. 4 and 5 show flows with $\boldsymbol{b_{max}}$ values as 160 and 320 while $\boldsymbol{b_{min}}$ values are 90 and 160 respectively.

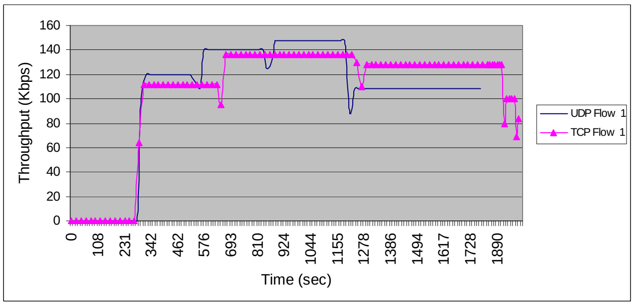


Fig. 4. Adaptive flow 1

Variable traffic load was generated at random nodes that caused the flows to be rerouted depending on the dynamic network conditions. Momentary QoS violations occurred due to rerouting but overall the flow kept its minimum bandwidth guarantee. We observed that the flow fluctuated between its maximum and minimum bandwidth constraints. Especially the flows under TCP fall below minimum bandwidth constraint at times. This is due to the TCP rate adjustment because of large network delay and packet loss.

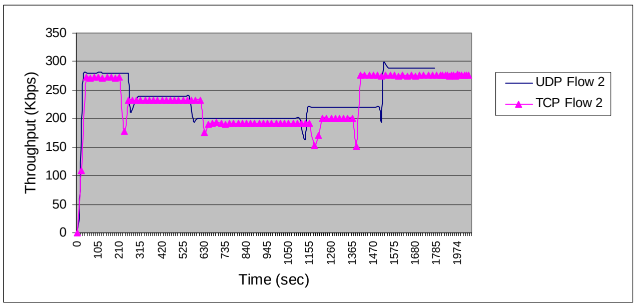


Fig. 5. Adaptive flow 2

5.2 Increasing Number of Nodes

The simulation environment consists of 5, 10 and 15 ad hoc nodes in an area of 200m x 200m. Each mobile node has a transmission range of 100m and shares 1Mbps air interface between neighboring mobile nodes within that transmission range. Two video streams were introduced in the network with different QoS requirements. Stream 1 has minimum QoS requirement as 150 Kbps and maximum QoS requirement as 300 Kbps. Similarly, stream 2 has minimum QoS requirement as 40 Kbps and maximum QoS requirement as 100 Kbps. As we compare Fig. 6 and Fig. 7 it is obvious that NETSAQ adapts to the increasing number of nodes very well. As we increase the number of nodes from without any QoS mechanism, the interference and contention increase considerably and degrade the performances of both the streams.

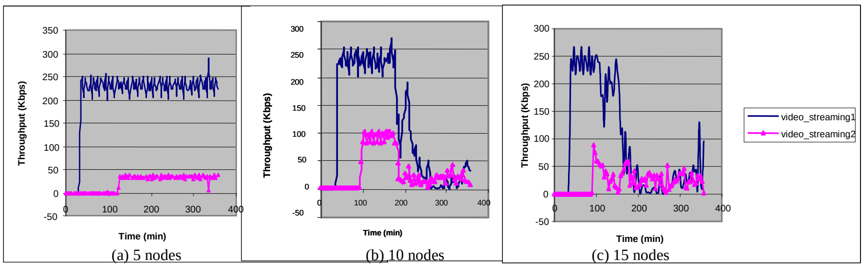


Fig. 6. Increasing number of nodes with no QoS

The Fig. 7 shows NETSAQ adapts to the changing conditions and finds a route with a slightly decreased QoS value within the minimum and maximum QoS constraints for both the streams. NETSAQ reroutes both the streams to a lower QoS levels to cope up with the degrading network conditions, but when the loading traffic increases and saturates the network towards the end, only best effort QoS can be provided.

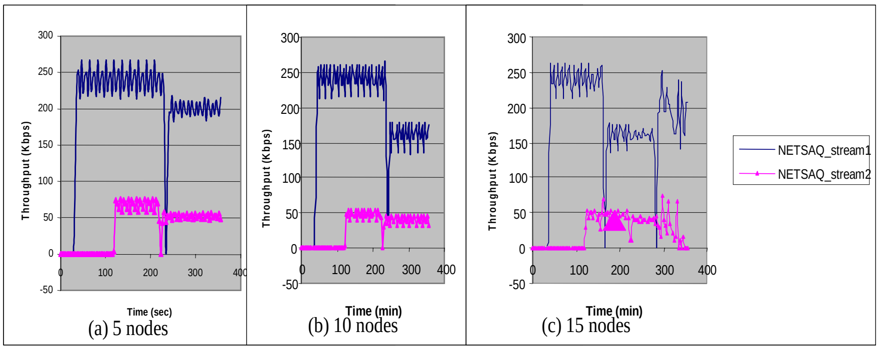


Fig. 7. Increasing number of nodes with NETSAQ

5.3 Changing Transmission Rate

We evaluated the performance under increasing transmission rates (by changing the air interface and increasing stream rates as well. The simulation environment consists of 15 ad hoc nodes in an area of 200m x 200m. Each mobile node has a transmission range of 100m. The transmission rate or air interface is changed from 1 to 2 and 5.5 Mbps. The comparison between Fig. 8 and Fig. 9 allows us to observe the way NETSAQ adapts to transmission rate variations.

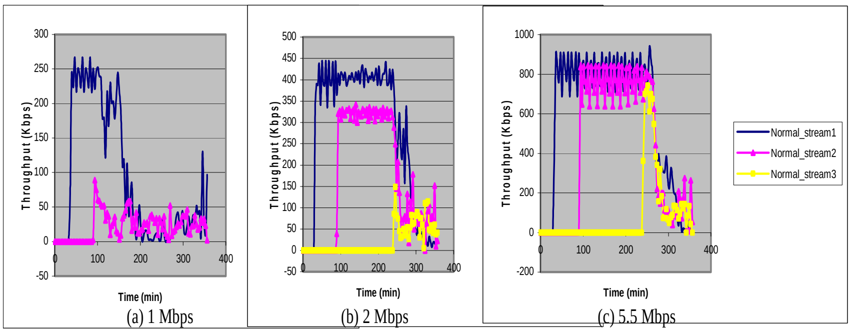


Fig. 8. Different transmission and streaming rates with no QoS

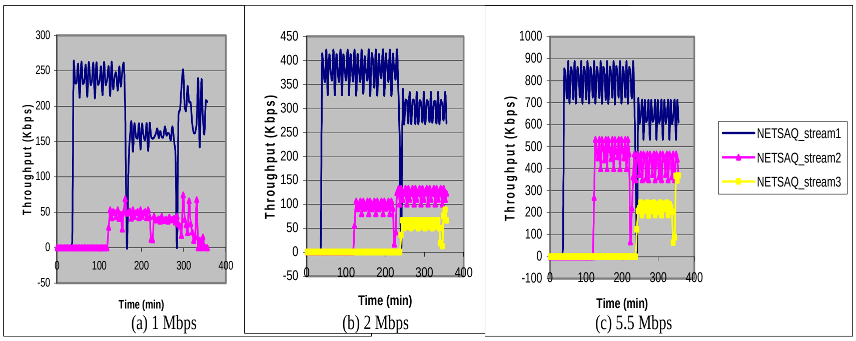


Fig. 9. Different transmission and streaming rates with NETSAQ

6 Conclusion

In this paper we have proposed NETSAQ, a network state adaptive QoS algorithm for MANETs. NETSAQ implements the QoS provisioning policies using any underlying routing protocol. It eliminates the bulky computation and communication for QoS in routing algorithm. It utilizes lesser MANET resources as compared to existing QoS routing protocols. It is simple yet minimizes the degradation of best effort traffic, which is observed in many QoS routing schemes.

References

1. Frodigh, M., Johansson, P., Larsson, P.: Wireless ad hoc networking: the art of networking without a network. *Ericsson Review*, No.4, pp. 248-263, 2000.
2. Veres, A., Campbell, A., Barry, A., Sun, L. H.: Supporting Service Differentiation in Wireless Packet Networks Using Distributed Control. *IEEE JSAC*, vol. 19, no. 10, Oct. 2001.
3. Lee, S., Ahn, G., Zhang, X., Campbell, A.: INSIGNIA: *IETF Internet Draft*, draft-ietfmanet-insignia-01.txt, work in progress, Nov. 1999.
4. Ahn, G., Campbell, A., Veres, A., Sun, V.: SWAN, *IETF Internet Draft*, draft-ahn-swanmanet-00.txt, work in progress, Oct. 2002.
5. Sivakumar, R., Sinha, P., Bharghavan, V.: Core Extraction Distributed Ad Hoc Routing (CEDAR) Specification. *IETF Internet Draft*, draftietf-manet-cedar-spec-00.txt, work in progress, 1998.
6. Labiod H., Quidelleur, A.: QoS-ASR: An Adaptive Source Routing Protocol with QoS Support in Multihop Mobile Wireless Networks. *IEEE VTC'02*, pp. 1978- 1982. 2002.
7. Badis, H., Al-Agha, K.: Quality of Service for Ad hoc Optimized Link State Routing Protocol (QOLSR). *IETF Internet Draft*, work in progress draft-badis-manet qolsr-01.txt, Apr. 2005
8. Ge, Y., Kunz, T., Lamont, L.: Proactive QoS Routing in Ad Hoc Networks. *ADHOC-NOW'03*, pp. 60-71, 2003.
9. Chaudhry, S.A., Akbar, A.H., Siddiqui F.A., Sik Y.W.: Autonomic Network Management for u-Zone Networks. *UbiCNS' 05*, Jeju, Korea, June 9-10, 2005
10. Charles E. Perkins and Elizabeth M. Royer, "The Ad hoc On-Demand Distance Vector Protocol" In Charles E. Perkins, editor, *Ad hoc Networking*, pages 173–219. Addison-Wesley, 2000.
11. Chung-Ng, P. Liew, K.: Re-routing Instability in IEEE 802.11 Multi-hop Ad-hoc Networks. *WLN'04*, Nov. 2004.