Performance Comparison and Analysis of two QoS schemes: SWAN and Diffserv

Harpreet Arora Lloyd Greenwald Usha Rao John Novatnack
{*harpreet@drexel.edu, lgreenwa@mcs.drexel.edu, uvr22@drexel.edu, john@plan.mcs.drexel.edu*}
Department of Computer Science
Drexel University, Philadelphia, PA 19104

I. INTRODUCTION

This paper compares and analyzes the performance of two QoS schemes for MANETs: SWAN and DiffServ. The comparison is performed both qualitatively and quantitatively (by simulations). We also suggest a hybrid QoS model which combines the merits of the two models to produce an effective QoS model for MANETs.

Quality of Service in an ad hoc environment has been widely recognized as a challenging problem [1]. Characteristics of ad hoc networks such as mobility, dynamics of the environment and uncertainty of resource availability makes provisioning of QoS guarantees difficult. Two broad approaches for QoS in networks have been suggested [8]: a *'stateful'* approach, requiring all nodes to create and maintain state information for each flow passing through them and a *'stateless'* approach, where the nodes differentiate traffic according to the class they belong to without maintaining any state information. Both the approaches have their merits and drawbacks. The stateful approach can assure bandwidth and delay bounds, but requires complex signalling mechanism to setup, update, maintain and remove per flow state information. Maintaining state information for each flow also makes this approach non-scaleable. Several examples of stateful schemes can be found in literature [2]. The stateless approach on the other hand, can provide only probabilistic guarantees, but the implementation is easy and scaleable. There is no need for per flow state information to be maintained at each node. The nodes only need to provide differential treatment to the packets based on information in their header. Examples of schemes based on stateless approach are SWAN [7] and Differentiated Services (DiffServ) [9]. We believe that stateless schemes are more suitable to an adhoc environment because of their inherent merits like simplicity, scaleability and ability to adapt to dynamic conditions. A comprehensive survey of the available schemes in this category is important to understand their applicability and success in achieving the desired results. This paper compares the performance of SWAN and DiffServ quantitatively and qualitatively. Quantitative analysis is performed by simulations using Network Simulator (NS-2) [11] under varied mobility conditions and scenarios. The results of simulations and general understanding of the working of the two models are used for their qualitative comparison.

SWAN uses admission control for real time traffic and varies the rate of TCP traffic based on feedback from MAC layer to maintain delay and bandwidth bounds for real time traffic. DiffServ, on the other hand, relies on policing, marking and scheduling mechanisms and RED mechanism [12] for congestion control implemented at the queue at MAC layer for providing QoS guarantees to traffic of different classes. The quantitative comparisons are performed using three routing algorithms: AODV [3] and DSR [13] which are distance vector based reactive algorithms and OLSR [4] which is a link-state based proactive routing algorithm. Throughput and delay graphs of flows under the two schemes suggest that SWAN outperforms DiffServ in terms of throughput and end-to-end packet delays. Detailed analysis of this performance difference is performed in sections IV and V. We also observe that neither of the schemes is able to provide absolute guarantee bounds to the traffic. The assurances are only probabilistic and dependent on network conditions and traffic patterns. SWAN is able to maintain high throughput and low delay for the admitted UDP flows, but treats all real-time traffic equally. Some UDP flows may be dropped during congestion. DiffServ, on the other hand, has the capability to support multiple classes but suffers from the drawback of using static parameters without taking into account the dynamics of the network. We propose a hybrid QoS model, which combines the merits of the two schemes. We believe that MAC feedback used by SWAN can be combined with traffic differentiation mechanism of DiffServ to produce an effective QoS model for MANETs.

The paper is divided as follows: the following section presents overview of SWAN and DiffServ, section III defines the simulation environment used for analysis, section IV compares and analyzes the performance of the two models.In section V we explain the need for multiple classes of traffic and point out the inability of SWAN in supporting more than 2 classes. In section VII we propose a hybrid QoS model that combines the merits of the two models. Finally we end with some discussion in section VIII

II. SWAN and DiffServ: An Overview

A. SWAN

SWAN (Service differentiation in wireless ad hoc networks) model for QoS in MANETs has been designed by the SWAN group at University of Columbia [6]. SWAN considers TCP traffic as best-effort traffic and UDP traffic as real-time traffic requiring QoS assurances. It tries to maintain delay and bandwidth requirements of UDP real time traffic by admission control of UDP traffic and rate control of TCP and UDP traffic.

Each node has an admission controller which estimates the availability of resources in the local neighborhood. To admit a new real time flow, the admission controller located at the source node sends a probing request packet towards the destination. This packet returns to the source with the value of bottleneck bandwidth along the path. If this value is greater than the requirements of the flow, the flow is admitted. To allow some flexibility and bandwidth sharing between real-time and best-effort traffic, the admission threshold is set below the maximum available resource limit. No state for any flow is maintained at any node and no resources are reserved. Packets of admitted UDP flows are marked as real-time packets, otherwise they are considered as best-effort packets. To regulate the traffic and to maintain the bandwidth and delay requirements of the admitted real-time traffic, rate of the TCP best-effort traffic is controlled using MAC delay measurements as feedback. A classifier and a shaper situated between the IP layer and the MAC layer differentiate the best effort traffic from real time traffic, such that the real time traffic passes directly to the MAC layer for transmission, while the best effort traffic is shaped and delayed according to rate set by the rate controller.

Dynamics of the network may however cause violations of the admitted real-time traffic, degrading the quality of some of the flows. To accomodate that, SWAN uses explicit congestion notification (ECN) to regulate UDP traffic. ECN bits in IP header of packets are set indicating the source to re-initiate the flow starting from admission control. A failed admission control causes the source to drop its flow. SWAN can thus provide only soft QoS assurances. An admitted real time flow is not guaranteed to meet its requirements for the entire duration of the connection.

B. DiffServ

Differentiated services (DiffServ) architecture [9] has been proposed as a QoS model for the wired internet. The model aims at moving the complexity to the edge of the network and simplying the core as much as possible, resulting in a system which is scaleable to an arbitrary extent. Traffic is divided into classes according to priorities and resource requirements. Customers communicate their requirements by forming service level agreements (SLAs) with the service providers. Routers at the edge of the network perform admission control and classify, shape and police the traffic so that it confirms to the SLAs. Classification is done by marking the IP header with a 4-bit DiffServ code point (DSCP). Within the core of the network, packets are queued and scheduled for transmission at each node according to the class they belong to. End-to-end QoS is built by aggregating the per-hop behavior at each node along the path.

This model, however, is not readily applicable to the wireless paradigm. Peer-to-peer nodes in an ad hoc network, for example, cannot be statically classified as edge and core nodes. Each node may assume either functionality depending upon its position in the network. Therefore, we modified the wired implementation of DiffServ in NS-2 to make it applicable to wireless networks. Each node incorporates the functionality of an edge node and a core node. Marking of the flows is performed at the source node while policing and shaping is done at all nodes along the path. The nodes implement Random Early Detection with priority (RIO) based queues [12]. Packets belonging to different classes are queued separately. The length of the queues and packet dropping probabilities during congestion are dependent on the class they belong to. Packets of higher priority are scheduled for transmission before the packets of lower priority are. The parameters of the queue are configured statically. No feedback of any kind is used to change any parameters. The parameters are tuned for best results after a series of trial-and-error experiments.

III. Simulation Environment

We used Network Simulator (NS-2) [11] with its wireless extensions developed at CMU for our analysis. The simulation environment consists of 51 nodes moving in a 300 meters X 1500 meters area. The movement of the nodes is defined by the random-waypoint model. Two installations of NS-2 simulator are used. DiffServ code for wired networks, which is available as a part of NS-2 package, was modified to make it suitable for wireless networks. The modified code was implemented on one instance of NS-2. SWAN code for NS-2 was downloaded from the SWAN webpage [6] and installed on another instance of NS-2. Four nodes were randomly selected to act as TCP sources, generating best-effort traffic and five nodes were selected to source UDP real-time traffic. TCP sources model FTP and web traffic and UDP sources model voice and video traffic. The length of file transfers generated by the TCP sources and the length of silent period between transfers is a Pareto distribution with a mean of 10 seconds. More details of simulation environment are presented in the paper. The scenario files used are the same ones used by SWAN group for their analysis [7]

For DiffServ, the traffic is divided into two classes: real time UDP traffic belonging to the higher priority class and TCP traffic belonging to lower priority class. The queue lengths and parameters are adjusted to yield high throughput of UDP traffic.

IV. Comparisons and Analysis

Quality of service can be defined as the ability of a network element to provide some level of assurance for consistent network data delivery. In this section, we analyze and compare the level of assurance that SWAN and DiffServ can provide to a flow demanding QoS. The analysis is done by measuring the average throughput available to UDP real-time flows and the average end-to-end delay experienced by the packets of these flows. We also measure the goodput of TCP flows as an indication of the amount of bandwidth share available to the best effort traffic. Graphs showing average throughput of a voice connection and the goodput available to a web connection are shown in Fig 1. For these graphs, DSR was used as a routing protocol. Focussing on the throughput graph,we can see that SWAN clearly outperforms DiffServ. Similar results are observed for other real-time flows and with other routing protocols (AODV and OLSR). These results are not produced here, but will be presented in the paper. SWAN uses rate control of TCP and UDP traffic to maintain manageable levels of congestion in the network. DiffServ, on the other hand, relies on static configuration of the parameters of the RED queue and schedular to provide service differentiation. This broadly explains the superior performance of SWAN over DiffServ. More details of this distinction are provided in the next section.

Although SWAN peforms relatively better, the throughput is not only much lower than 100%, but also varies with mobility. This can be attributed to the difficulty in capturing the dynamics of the environment in an ad hoc network. SWAN relies only on feedback from the MAC layer as a measure of congestion in the network. However, other factors such as the average speed of the nodes may be important in determining the effective resource availability [14]. Absence of reservations and source based admission control also causes some flows to be dropped, affecting their throughput.

Fig 2 compares the average end-to-end delays experienced by packets of web and video flows in SWAN and DiffServ. SWAN is able to maintain a delay of less than 200 ms which is essential for a voice conversation. Absence of rate control allows DiffServ to build up congestion in the network causing packets to wait in the queues for longer periods of time. SWAN mechanisms ensure that if the delays increase beyond a certain threshold, either the rate of TCP flows is cut down or some of the UDP flows are dropped, thereby maintaining low delays for the remaining flows.

(In our paper, we will provide details of each of these mechanisms, backed with graphs and figures)

V. Explaining the Performance Differences

The Differentiated Services framework is suggested as an end-to-end QoS solution for wired networks. Users sign service level agreement (SLA) with the service provider. Providers engineer the traffic of the network to meet the SLAs. Traffic Engineering in such networks is relatively easy due to their static nature. Bandwidth and other resources are unlikely to change by large amounts frequently. Admission control at the edge nodes also ensures that the capacity of the network is not exceeded to violate QoS of the admitted flows. It may not be possible to define SLAs in a dynamically formed peer-to-peer ad hoc network. Further, frequently changing bandwidth and traffic conditions make traffic engineering a challenging task. It is imperative to use some kind of feedback as a measure of the conditions of the network to dynamically regulate the traffic of the network. Since the differentiated services architecture does not define any scheme for taking corrective action on-the-fly, a direct application of this model to a wireless network results in a deterioted performance as seen in the previous section. Detailed comparison of the performance difference of DiffServ in wired networks and their wireless counterparts is presented in the paper.

SWAN engineers traffic by observing packet delay at the MAC layer and considering it as a measure of the amount of congestion in the network. It uses this to dynamically regulate the rate of both TCP and UDP traffic. On detecting

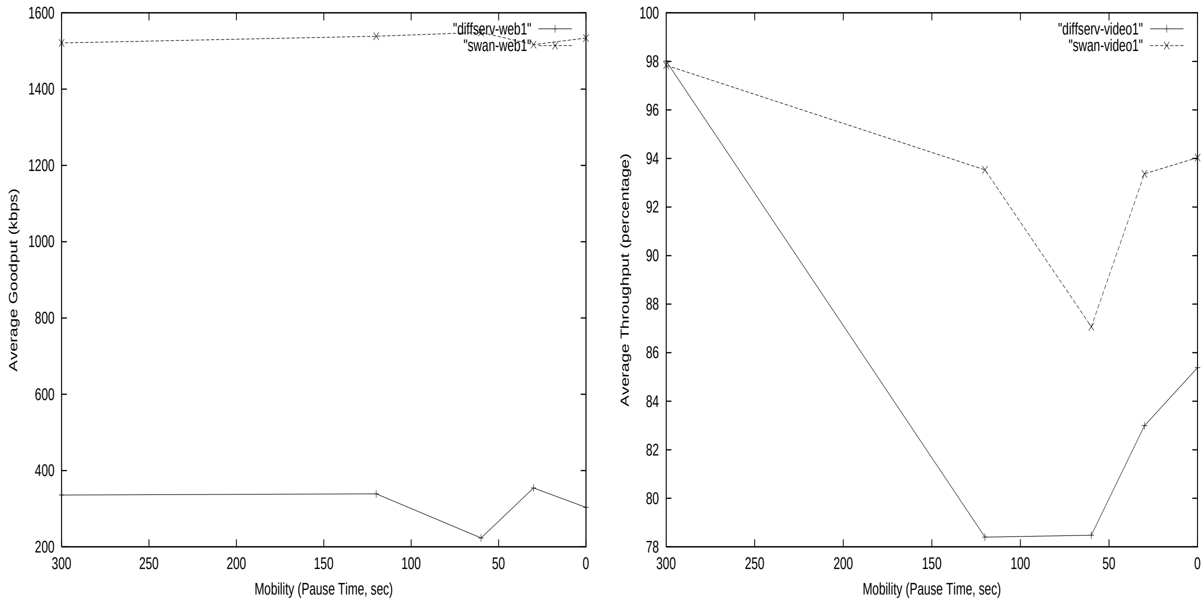

Fig. 1. (left) Average goodput of a TCP flow measured in kbps v/s mobility. Mobility is increased by decreasing the average pause time of the nodes. (right)Average throughput of a UDP flow v/s mobility

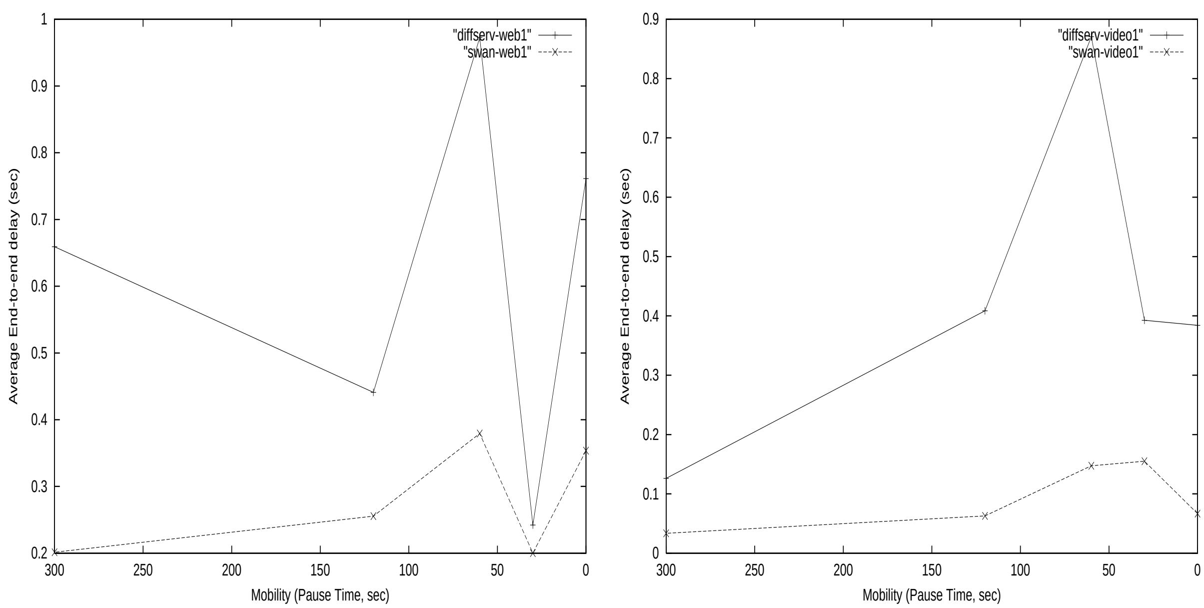

Fig. 2. Average end-to-end delay experienced by packets of a TCP flow (left) and a UDP flow (right)

high levels of congestion, it allows some real-time flows to be dropped or rerouted, thereby maintaining the QoS of the remaining flows. The success of this technique is visible in the results presented in the previous section. More details of this mechanism and its comparison with DiffServ are presented in the paper.

VI. TRAFFIC DIFFERENTIATION

Traffic differentiation is the primary goal of any QoS scheme. The conventional internet treats all traffic as best-effort. However, modern applications such as VoIP, video conferencing, streaming multimedia demand certain guarantee of resources for the entire duration of connection for their proper functioning. Further, some connections may be more important than others and may require to be treated with higher priority. Traffic can be broadly differentiated into classes according to two criteria:

a. **Resource requirements:** Traffic belonging to different classes may have different bandwidth and delay requirements. For example, voice conversations need stringent delay requirements and a low steady bandwidth for the duration of the call. A live video telecast on the other hand, has high bandwidth requirements, though small amount of jitter can be tolerated by prebuffering. Also, in times of congestion, flows belonging to such connections can be more readily dropped or rerouted.

b. **Priority:** Some flows in a network may be more important than others. For example, a voice conversation between a director and a manager may need higher priority treatment than a voice conversation between two salesmen. Hence in times of congestion, the latter flow should be dropped prior to the former flow.

SWAN differentiates traffic into 2 classes, best effort and real time traffic. It treats all UDP traffic with equal priority and each one of them has an equi-probable chance of getting dropped, or rerouted during congestion. This may not be sufficient for a complete QoS solution for the network. DiffServ on the other hand, allows traffic to be classified according to the above criteria. Packets belonging to lower priority classes may be selectively dropped prior to dropping packets of higher priority classes. Also, a flow consuming large amount of bandwidth could be dropped instead of dropping many flows consuming low bandwidth. This allows flexibilty in handling network dynamics. In [10], the author analyzes the ability of DiffServ model to provide traffic differentiation in an ad hoc network. Diffserv enqueues packets belonging to different classes in separate queues and uses a schedular to schedular the packets out of the queues for transmission according to their priorities. SWAN enqueues all real time packets into a single fifo queue, without any distinction. Thus, SWAN cannot be called as a complete QoS solution. In the next section, we propose a hybrid QoS model combining the merits of DiffServ and SWAN.

VII. A Hybrid QoS Model

From sections V and VI, we can conclude that neither SWAN nor DiffServ is effective as a complete QoS solution for MANETs. The merits of the two models can be combined to produce an effective QoS model for MANETs. This model can use feedback from the network to dynamically regulate its parameters as done in SWAN and support multiple classes of traffic as supported by DiffServ. The classes are differentiated according to their priorities and resource requirements. The model can have an admission control unit, a feedback mechanism, a RED queue will multiple physical queues for different classes of traffic, rate controllers for each class of traffic and a schedular to schedule traffic for transmission. Different rate regulators will control the parameters of the RED queue such that traffic belonging to lower priority classes is regulated more aggressively as compared to traffic belonging to higher priority classes. We plan to design this model and analyze it by simulations in our ongoing work. Detailed mechanisms and block diagrams of the model will be shown in the paper.

VIII. Discussion

In this paper we compare the performance of two stateless QoS models for ad hoc networks: SWAN and DiffServ. Simulation analysis shows that SWAN performs better than DiffServ in maintaining bandwidth and delay requirements. We analyze the differences by understanding the working of the mechanisms of the two models. Although SWAN outperforms DiffServ, it does not present a complete QoS solution for MANETs. We propose a hybrid model combining the merits of SWAN and DiffServ. The hybrid model will be able to provide different levels of bandwidth and delay assurances for different classes of traffic. In our ongoing work, we plan to write NS-2 code for this model and make it available for use to the research community.

References

[1] S. Chakrabarti and A. Mishra, "QoS issues in ad hoc wireless networks," IEEE Communications Magazine, vol. 39, issue 2, pp. 142-148, February 2001.

[2] S.-B. Lee, G.-S. Ahn, X. Zhang, and A.T. Campbell, "INSIGNIA:An IP-Based Quality of Service Framework for Mobile Ad Hoc Networks," J. Parallel and Distributed Computing, special issue on wireless and mobile computing and communications, vol. 60, no. 4, pp. 374-406, Apr. 2000.2002.

[3] C.E. Perkins and E.M. Royer, "Ad-Hoc On Demand Distance Vector Routing," Proc. IEEE Workshop Mobile Computing Systems and Applications, Feb. 1999.

[4] Cedric Adjih, Thomas Clausen et al., "Optimized Link State Routing Protocol," Internet Draft, IETF MANET working group, March 2003

[5] S. Floyd, “TCP and Explicit Congestion Notification,” ACM Computer Comm. Rev., vol. 24, no. 5, Oct. 1994.
[6] SWAN Project: http://comet.columbia.edu/swan, 2002.
[7] G.-S. Ahn, A.T. Campbell, A. Veres, and L.-H. Sun, “SWAN: Service Differentiation in Stateless Wireless Ad Hoc Networks,” Proc. IEEE INFOCOM 02, 2002.
[8] I. Mahadevan and K. M. Sivalingam, “Quality of service architectures for wireless networks: IntServ and DiffServ models,”Workshop on Mobile Computing at the International Symposium on Parallel Architectures, Algorithms and Networks pp 420-425, June 1999.
[9] K. Nichols, V. Jacobson and L. Zhang, “A Two-bit Differentiated Services Architecture for the Internet,” RFC-2638, July-1999
[10] H. Arora and H. Sethu, “A simulation study of the feasibility of differentiated services architecture for QoS in mobile ad hoc networks,” Applied Telecommunications Symposium, San Diego, Apr. 2002.
[11] http://www.isi.edu/nsnam/ns
[12] S. Floyd and V. Jacobson , “Random early detection gateways for congestion avoidance,” IEEE/ACM Transactions on Networking, Volume: 1 Issue: 4 , Aug 1993 pp. 397-413
[13] D. Johnson, D. Maltz and Yih-Chun Hu, “Dynamic Source Routing Protocol for Mobile Ad Hoc Networks,” Internet Draft, IETF MANET working group, Feb 2003
[14] H. Arora and H. Sethu, “A Simulation Study of the Impact of Mobility on Performance in Mobile Ad Hoc Networks,” Applied Telecommunications Symposium, San Diego, Apr. 2002.