A FRAMEWORK FOR AN AGGREGATED QUALITY OF SERVICE IN MOBILE AD HOC NETWORKS

Ash Mohammad Abbas, Khaled Mohd Abdullah Al Soufy
Department of Computer Engineering
Zakir Husain College of Engineering and Technology
Aligarh Muslim University
Aligarh – 202002, India
am.abbas@amu.ac.in

ABSTRACT

Providing quality of service in an ad hoc network is a challenging task. In this paper, we discuss a framework for user perceived quality of service in mobile ad hoc networks. In our framework, we try to aggregate the impact of various quality of service parameters. Our framework is flexible and has a provision of providing dynamic quality of service. Further, an application may adapt from the required quality of service to that which can readily be provided by the network under a stressful environment. Our framework may adapt to the QoS desired by a source based on user satisfaction.

Keywords: User perceived quality of service, quality of service aggregation, dynamic and adaptive, ad hoc networks.

1 INTRODUCTION

An ad hoc network is a cooperative engagement of a collection of mobile devices without the required intervention of a centralized infrastructure or a centralized access point. In the absence of centralized infrastructure, an ad hoc network may provide a cost-effective and a cheaper way of communication. Applications of an ad hoc network include battlefield communications, disaster recovery missions, convention centers, online classrooms, online conferences, etc.

In an ad hoc network, there are no separate routers. As a result, the devices need to forward packets of one another towards their ultimate destinations. The devices possess limited radio transmission ranges, therefore, routes between any two hosts are often multihop. The devices are often operated through batteries whose power depletion may cause the device failure and/or associated link failures. Further, nodes may move about randomly and thus the topology of the network varies dynamically.

There can be applications of an ad hoc network, where users expect a given level of quality of service (QoS) to be provided by the network. These applications may include multimedia streaming, exchanging geographical maps, etc. However, the requirements and the expectations of users about the level of the QoS to be provided by an ad hoc network may not be as high as those in case of a wired network or a wireless network that possesses a centralized infrastructure.

Provision of QoS in an ad hoc network is a challenging problem. The challenge is posed by the characteristics of ad hoc networks. In an ad hoc network, one cannot have a solution that either relies on extensive amount of computations or consumes a significant amount of power and energy because the resources of the devices used to form such a network are scarce. Note that an ad hoc network is an on the fly network and should be self organizing in nature. Frequent node and link failures together with mobility of nodes give rise to a highly dynamic topology of the network. The dynamically varying topology of the network makes it difficult to provide any hard QoS guarantees. However, as the users are aware that they are part of an ad hoc network, therefore, instead of expecting hard QoS guarantees, users may expect a *soft* QoS.

In situations, when an end-user can tolerate variations in the QoS, the user should have a flexibility to change the specifications of QoS parameters depending upon the extent of satisfaction with the QoS provided by the network. However, not much work is done in this area. In [6], a framework for QoS aware service location is presented in the context of an ad hoc web server system. Therein, the authors assign the priorities in integers to QoS parameters. In [11], an adaptive QoS routing protocol is presented by rerouting the packets that faced QoS violations.

The rest of this paper is organized as follows. In Section 2, we discuss the framework for an aggregated quality of service. In Section 3, we discuss how to adapt quality of service based on user feedback. In Section 4, we describe methods of QoS aggregation. Section 5 contains results and

Table 1: Symbols used for QoS parameters in AQS.

Parameter	Weight	Min/Max Value	Tolerance Limit (%)
End-to-End Delay	w_D	D_{max}	δD
Delay Jitter	w_J	J_{max}	δJ
Bandwidth	w_B	B_{min}	δB
Packet Delivery Ratio	w_R	R_{min}	δR
Route Lifetime	w_L	L_{min}	δL

Table 2: Example 1 – Values of QoS parameters.

Parameter	Weight	Min	Max	Tolerance Limit (%)
End-to-End Delay	0.40	3.0	-	1.0
Delay Jitter	0.20	0.2	-	1.0
Bandwidth	0.25	2.0	-	3.0
Packet Delivery Ratio	0.1	-	0.80	2.0
Route Lifetime	0.05	-	10	2.0

discussions. Finally, the last section is for conclusion and future directions.

2 A FRAMEWORK FOR AN AGGREGATED QOS

In this section, we describe a framework for an aggregated QoS. We call our framework as an aggregated QoS (AQS) framework because it aggregates the effect of many QoS parameters or metrics. In our framework, we consider a set of QoS parameters such as *end-to-end delay, delay jitter, bandwidth, packet delivery ratio, route lifetime*[1].

Our aggregation mechanism consists of assigning importance or weights to each of the parameters discussed in the previous subsection, and then computing a factor of aggregation. Let us first consider assignment of importance or weights[2]. To each of these parameters, we assign a weight $w_i, 0 \leq w_i \leq 1$, in such a fashion so that

$$\sum_{i=1}^{n} w_i = 1 \qquad (1)$$

where w_i represents the relative importance (or the weight) of parameter *i*. If there are *n* parameters, and each parameter *i* is assigned a weight $1/n$ then all QoS parameters are equally important. If $w_i > w_j, i \neq j$, then parameter *i* is said to be relatively more important than parameter *j*.

The notation used to represent information related to QoS parameters is as follows. We use the following symbols to represent QoS parameters− *D: end-to-end delays*, *J: delay jitter*, *B: bandwidth*, *R: packet delivery ratio*, *L: route lifetime*. For each of these parameters, the symbol *w* is used to represent relative importance or weight, prefix Δ is used to represent tolerance limit, and subscripts *max/min* are used to represent either the maximum or minimum value of the parameter. This notation is summarized in Table 1.

In what follows, we define aggregated QoS to incorporate the effect of the parameters mentioned above.

Definition 1: Let there be *n* QoS parameters $P_1, P_2, ..., P_n$. Let $P_k, 1 \leq k \leq n$ for *bandwidth* be defined as follows.

$$P_{BW} = \frac{FileSize}{BW} \qquad (2)$$

where, *FileSize* denotes the size of file that is sent using the particular bandwidth.

Let $P_k, 1 \leq k \leq n$ for *packet delivery ratio* be defined as follows.

$$P_{PDR} = R\Delta \qquad (3)$$

where, Δ is the duration of time for which the particular *packet delivery ratio*[3] is desired.

Having defined the constituent parameters in time units, we now define a parameter called Weighted Aggregate QoS (WAQ) as follows.

$$WAQ = \sum_i \left\{ \left(\left| P_i^{min} \right| + \left| TL_i \right| \right) w_i \right\} \\ + \sum_j \left\{ \left(\left| P_j^{max} \right| + \left| TL_j \right| \right) w_j \right\} \qquad (4)$$

where $1 \leq i, j \leq n, i \neq j$, and P_i^{min} is the value of *i*th parameter whose minimum value is specified, and P_j^{max} is the value of *j*th parameter whose maximum values is specified. Further, TL_i and TL_j are values of the corresponding tolerance limits for *i*th and *j*th parameters.

Note that the unit of aggregation parameter, WAQ, is time. Further, it will have positive values and its values may come out to be greater than 1 depending the values of its constituent QoS parameters. In what follows, we define an aggregation factor, whose value lies between 0 and 1.

---

[1] There can be long list of QoS parameters, however, we consider only the parameters defined above.

[2] Note that it is a different issue that who assigns the weights to the parameters and shall be discussed later in this paper.

---

[3] In this way, we have converted all parameters to a single unit i.e. time. This is done so that we are able to aggregate the effect of different parameters on the QoS.
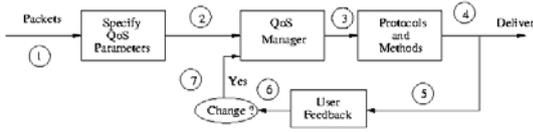
Figure 1: Adapting QoS through user feedback.

Definition 2: Let the QoS parameters, $P_i^{\min}$, $P_j^{\max}$, and their tolerance limits, TL_i, and TL_j, be defined as in *Definition* 1. We define a factor that we call Weighted Aggregate QoS Factor (WAQF) as follows.

$$WAQF = \frac{WAQ}{\sum_i \left(\left| P_i^{\min} \right| + \left| TL_i \right| \right) + \sum_j \left(\left| P_j^{\max} \right| + \left| TL_j \right| \right)}$$

(5)

where, WAQ is given by (4) as part of *Definition* 1. It is worth mentioning that WAQF has no unit as it is simply a ratio bearing the units of time in both the numerator as well as the denominator of the expression defining it.

In what follows, we discuss an example incorporating different weights and the values of the QoS parameters mentioned above.

Example 1: Let the weights, min/max values, and tolerance limits assigned by the source for different QoS parameters be as given in Table 2.

In *Example 1*, the end user has given the highest relative importance by assigning a weight 0.40 to the end-to-end delay. The maximum value of the delay is specified to be 3.0 milliseconds, however, the user may accept upto a tolerance limit 1.0%. This means that user may accept a value of the end-to-end delay upto 3.01 milliseconds. The second parameter is delay jitter. For that the user has specified a weight of 0.20, the maximum value to be 0.2 milliseconds, and a tolerance limit of 1.0%. This means that the user can accept the value of delay jitter upto 0.202 milliseconds. The third parameter is bandwidth reserved for the flow. The user has specified a weight of 0.25, the minimum value of bandwidth to be 2.0 Mbps, and a tolerance limit of 3.0%. In other words, the user may accept the bandwidth upto 1.96 Mbps. The fourth parameter, packet delivery ratio is assigned a weight of 0.10, the minimum value 80% and a tolerance limit of 2.0%. This means that the user may accept the packet delivery ratio upto 78%. The last parameter is route lifetime[4] that is assigned a weight of 0.05, the minimum value 10 milliseconds, and a tolerance limit of 2.0%. In other words, the user may accept the value of the route failure time upto 9.8 milliseconds.

---

[4] There can be a debate whether one should consider the minimum value or the maximum value of route failure time. The user would prefer to specify the minimum value of route failure time, so that there are no route failures during packet transmissions. However, from the point of view of network, one would like to maximize the value of route failure time.

Table 3: Example 2 – Values of QoS parameters.

Parameter	Weight	Min	Max	Tolerance Limit (%)
End-to-End Delay	0.30	2.0	-	1.0
Delay Jitter	0.10	0.3	-	1.0
Bandwidth	0.45	2.0	-	3.0
Packet Delivery Ratio	0.1	-	0.80	2.0
Route Lifetime	0.05	-	10	2.0

Table 4: Default values of QoS parameters.

Parameter	Value	Tolerance Limit (%)
End-to-End Delay	5.00	2.0
Delay Jitter	0.01	2.0
Bandwidth	10.0	2.0
Packet Delivery Ratio	0.90	2.0
Route Lifetime	10.0	2.0

Table 5: Sets of weights assigned to QoS parameters.

Set No.	Set of Weights
S1	0.6 0.1 0.1 0.1 0.1
S2	0.1 0.6 0.1 0.1 0.1
S3	0.1 0.1 0.6 0.1 0.1
S4	0.1 0.1 0.1 0.6 0.1
S5	0.1 0.1 0.1 0.1 0.6

Table 6: The QoS parameter WAQ and the factor WAQF as a function of bandwidth.

Bandwidth	WAQ	WAQF
1	3.40964	0.2
2	3.30760	0.2
3	3.27359	0.2
4	3.25658	0.2
5	3.24638	0.2
6	3.23957	0.2
7	3.23471	0.2
8	3.23107	0.2
9	3.22824	0.2
10	3.22597	0.2

Table 7: The QoS parameter WAQ and the factor WAQF as a function of FileSize.

FileSize	WAQ	WAQF
1	3.22957	0.2
2	3.24638	0.2
3	3.26678	0.2
4	3.28719	0.2
5	3.30760	0.2
6	3.32801	0.2
7	3.34842	0.2
8	3.36883	0.2
9	3.38923	0.2
10	3.40964	0.2

Table 8: The QoS parameter WAQ and the factor WAQF as a function of Δ.

Δ	WAQ	WAQF
1	3.22597	0.2
2	3.40957	0.2
3	3.59317	0.2
4	3.77677	0.2
5	3.96037	0.2
6	4.14397	0.2
7	4.32757	0.2
8	4.51170	0.2
9	4.69477	0.2
10	4.87837	0.2

Table 9: The QoS parameter WAQ and the factor WAQF as a function of sets of weights.

Set of Weights	WAQ	WAQF
S1	4.06298	0.251892
S2	1.61788	0.100304
S3	1.66400	0.103163
S4	2.07198	0.128457
S5	6.71298	0.416184

Let the file size be 1 Mb, and the maximum bandwidth be $B_{max} = X$ Mbps. As a result, $1/X$ seconds is consumed in sending a file of the specified size over the specified bandwidth. For a variation or tolerance limit of Δ in a given bandwidth X, the value of the parameter P_k for bandwidth is

$$P_{BW} = \frac{1}{X - (X \times \Delta)}. \qquad (6)$$

Therefore, for a tolerance limit of 3.0% in the value of bandwidth which is 2.0 Mbps (as shown in Table 2 for *Example* 1), the value of P_{BW} is

$\dfrac{1}{2 - (2 \times 0.03)} = 0.515463917$. Further, in case of packet delivery ratio which is given by (3), assume that Δ be 1 time unit, so that P_{PDR} is affected by the packet delivery ratio only and that is R. For *Example* 1, the value of aggregation parameter, WAQ, comes out to be 1.948065979. The value of denominator is 14.69946392. The value of the aggregation factor, WAQF, comes out to be 0.132526328.

Note that in *Example 1*, the end-to-end delay is assigned the highest weight or priority. This is an example of delay-sensitive application. Depending upon the weight assigned to a QoS parameter, there can be other types of applications as well. For that consider another example with different weights and different values of corresponding QoS parameters.

Example 2: Let the weights, min/max values, and tolerance limits assigned by the source for different QoS parameters be as given in Table 3.

In *Example* 2, the bandwidth is assigned the highest relative importance. This is an example of bandwidth-sensitive application. The next relatively important parameter is end-to-end delay. Other parameters may not be so important for a particular application, therefore, those are assigned relatively low weights. For rest of the parameters, we assume a similar scenario as in *Example 1*.

In *Example 2*, $P_{BW} = 0.515463917$. The value of aggregation parameter, WAQ, comes out to be 1.447258763. The value of denominator is 15.233. The value of the aggregation factor, WAQF, comes out to be 0.095008124. Note that the aggregation factor is 28.31% smaller as compared to that in *Example 1*. The reason is that in case of WAQ, the contribution of end-to-end delay component has become half of that in *Example 1*, and the contribution of delay jitter has also been decreased. The total decrease of these two parameters is approximately 0.6. The contribution due to bandwidth has increased by a value of approximately 0.103. As a result, the net effect is a decrease by a value of approximately 0.5 in the value of WAQ. The aggregation factor, WAQF, has changed accordingly.

In what follows, we discuss a framework for adapting QoS through user feedback.

3 ADAPTIVE QOS THROUGH USER FEED-BACK

Fig. 1 shows a framework for adapting QoS using user feedback. The steps in our framework are as follows.

- The source or the user specifies the values of different QoS parameters with their minimum/maximum values and tolerance limits.
- The QoS parameters alongwith their respective values are given to the *QoS Manager.*
- If *QoS Manager* receives QoS parameters for a packet of the flow for the first time, it sorts the parameters according to their relative importance or weights. After, that the *QoS Manager* calls a protocol or a method to take care of the parameter that is relatively the most important. After that it takes measures to take care of the next relatively important parameter (if possible), and so on.
- The packet is then delivered to the destination according to its QoS specifications and the source is informed accordingly.
- If the source or the user is satisfied with the QoS of the packet delivered, the next packet is sent to the destination.
- If the source is not satisfied with the QoS of the packet delivered, it informs the *QoS Manager* about the change it wishes to have

in the QoS. The *QoS Manager* tries to adjust the QoS parameters accordingly.

Note that the source specifies the values of QoS parameters, their minimum and/or maximum values, relative importance, and tolerance limit for each parameter. As mentioned above, *QoS Manager* sorts the parameters according to their relative importance. The *QoS Manager* calls appropriate methods or protocols for providing the QoS. The fact that which protocol has to be called first depends upon the relative importance of the parameters. Depending upon the relative importance of QoS parameters, different methods are required to be called.

Further, the functionality of *QoS Manager* [5] resides at every node in the network. As mentioned earlier, we confine to wireless networks using 802.11 and the nodes in that are operating in ad hoc mode. However, the same can be extended with some modifications to other types of wireless networks as well. There might be a question that in ad hoc networks, nodes have limited resources so why do we have QoS with user feedback. It should be noted that provision of user feedback in our approach should require little more computations. In general, the energy and power consumption during transmissions is significantly larger than that of computations. We believe that the provision of user feedback shall not consume a significant amount of energy rather it will add few more computations and would be feasible with current technological trends.

In what follows, we describe what are the methods and protocols that may need to be called for a QoS specification.

4 METHODS FOR AGGREGATED QOS

In this section, we present methods and protocols that the *QoS Manager* needs to call for providing the desired QoS. Note that the input to the *QoS Manager* is a set of parameters with their respective values, tolerance limits, and relative importance. The first and the foremost task that the *QoS Manager* needs to perform is sorting of the QoS parameters according to their relative importance or weight. Another set of input is the user feedback, in case when some form of QoS has been provided to the flow but the user is not satisfied with the QoS. Once the QoS parameters have been sorted, appropriate methods and/or protocols are required to be called, to provide the given level of QoS.

Algorithm 1 describes what are the actions that are taken by the *QoS Manager*. When a sources needs to send packets of a flow, it sends a qosEnquiry packet to the *QoS Manager* along with the specification of QoS parameters. The *QoS*

Manager extracts the values of QoS parameters, tolerance limit, and relative importance. If the sum of all relative importance or weight is greater than 1, the QoS specifications are referred back to the source. Otherwise, the *QoS Manager* marks whether a parameter is "insignificant" or "significant".

We assume that if the weight assigned to a particular parameter is less than $1/(2k)$, where k is the number of QoS parameters, then the parameter is insignificant, otherwise it is significant. If a parameter is insignificant, the *QoS Manager* need not bother about it. For all significant parameters, the weights of the parameters are arranged in descending order. The first parameter in this order is the most significant parameter. An appropriate protocol is called to provide QoS for the most significant parameter so obtained. A qosReply packet is sent to the source by the *QoS Manager*. Upon receiving a qosReply packet, the source sends a TestQoS packet. The TestQoS packet is delivered to the destination and a flag named statusP_i is set to be "done" for the QoS parameter, P_i. The source, if satisfied sends the extent upto that he/she is satisfied. If the satisfaction of the source falls below 50%, the source shall specify his/her desired QoS parameters again, and the above process shall continue. When the user is satisfied by the QoS provided to TestQoS packets, he/she will start sending actual packets.
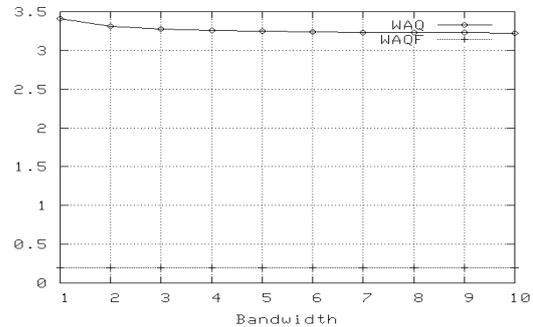
Figure 2: The QoS parameter WAQ and the factor WAQF as functions of bandwidth.

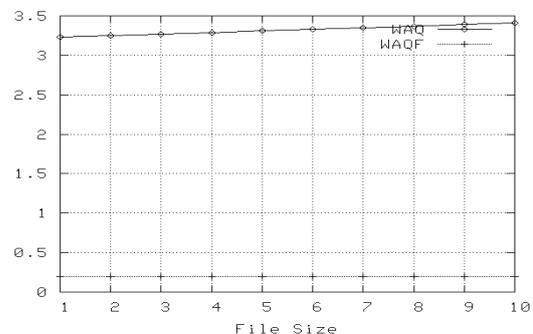**Figure 3:** The QoS parameter WAQ and the factor WAQF as functions of *FileSize*.

---

[5] The component *QoS Manager* is not only manages the QoS, however, it is also responsible for other functions like resource reservation, call admission control, and negotiating the QoS.


Algorithm 1: Marking of significant parameters by *QoS Manager*

Let $i \in$ enum P: {Delay, Jitter, Bandwidth, Delivery Ratio, Lifetime}, $1 \le i \le k, k = |P|$, where |.| denotes the cardinality. For parameter i, let w_i : weight, V_i : value, δ_i : tolerance limit.

1: **if** UserSatisfaction$P_i \le 0.5$ **then**

2:        status$P_i = $ "notdone"

3:        Get w_i , V_i , δ_i for all i such that status$P_i = $ "notdone"

4:        **if** $\sum_i w_i > 1$ **then**

5:            Refer back the QoS parameters to the source

6:        **else if** $0 \le w_i \le \dfrac{1}{2k}$ **then**

7:            Mark P_i : "insignificant"

8:        **else**

9:            Mark P_i : "significant"

10:        **end if**

11:        For all P_i that are marked "significant", arrange w_i in descending order

12:        For $\max(w_i)$, call a protocol to provide the QoS for parameter P_i

13:        Deliver the packet to destination

14:        set staus$P_i = $ "done"

15: **end if**
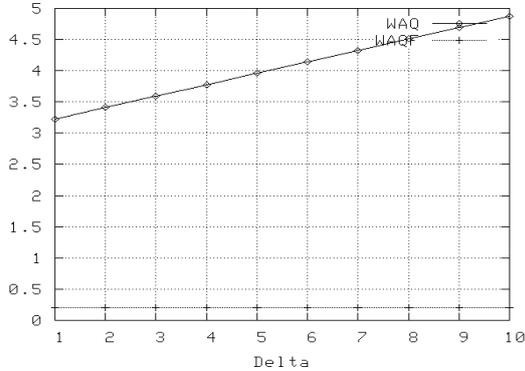
Figure 4: The QoS parameter WAQ and the factor WAQF as functions of Δ .
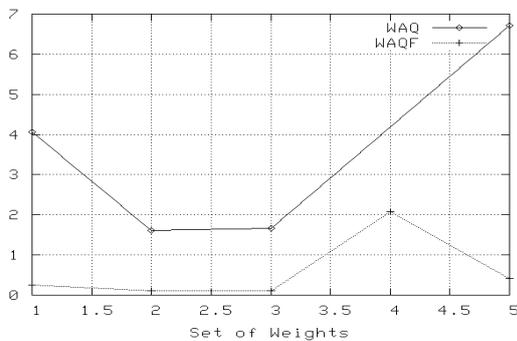
Figure 5: The QoS parameter WAQ and the factor WAQF versus sets of weights.

There is an issue about how long should one be allowed so as not to waste time in setting up of the desired QoS. This depends upon how many attempts are being made for a QoS parameter. We limit these attempt to 3 irrespective of the user satisfaction. After third attempt, we assumed that the most significant parameter is taken care of.

In what follows, we discuss some empirical results.

5 RESULTS AND DISCUSSION

Let us first discuss some results pertaining to the effect of aggregation of QoS parameters.

5.1 Effect of Aggregation

We have defined the weighted aggregation QoS parameter, WAQ, and weighted aggregation QoS factor, WAQF, in Section 2. For the results in this subsection, the values of different QoS parameters and their tolerance limits are shown in Table 4. The default weight assigned to each QoS parameter is equal and is 0.20. The default value of *FileSize* is 1 bandwidth unit (e.g. 1 Mbps, if the bandwidth is expressed in Mbps). The default value of time duration for which a desired packet delivery ratio is needed, Δ is 1 time unit.

Table 6 shows the empirical values of the QoS aggregation parameter, WAQ, and that of the QoS aggregation factor, WAQF, as functions of bandwidth. It is observed from Table 6 that the QoS aggregation parameter, WAQ, decreases with the increase in the bandwidth. The decrease in WAQ is not linear (see Figure 2). The values of the QoS factor remain constant. The reason is that for equal weights assigned to individual QoS parameters, the amount of increase in numerator and the denominator of the QoS aggregation factor, WAQF (as defined by (5)), is almost the same. Hence, WAQF remains the same and is equal to the weight assigned to each individual QoS parameter.

Table 7 shows the empirical values of the QoS aggregation parameter, WAQ, and that of the QoS aggregation factor, WAQF, as functions of *FileSize*. It is observed from Table 7 that the QoS aggregation parameter, WAQ, increases with the increase in the *FileSize*. The decrease in WAQ is linear (see Figure 3).

Table 8 shows the empirical values of the QoS aggregation parameter, WAQ, and that of the QoS aggregation factor, WAQF, as functions of Δ. We mentioned earlier that Δ is the time duration for which the specified packet delivery ratio is required. It is observed from Table 8 that the QoS aggregation parameter, WAQ, increases with the increase in Δ. The decrease in WAQ is linear (see Figure 4). The values of the QoS factor, WAQF, remain constant for the reason mentioned above.

The sets of weights assigned to the QoS parameters in the following order <*end-to-end delay, delay jitter, bandwidth, packet delivery ratio, route lifetime*> are shown in Table 5. It is observed that the value of the QoS aggregation parameter, WAQ, and that of aggregation factor, WAQF, are the largest for the set of weights, **S6**, and is the smallest for the set of weights, **S2** (see Figure 5). The reason being that in case of **S6**, the contribution of the largest valued QoS parameter i.e. *route lifetime* is multiplied by the largest weight among **S6**. However, the situation in case of **S2** is just reverse of that in **S6**. Note that the next largest values of WAQ and WAQF are for the set of weights, **S2**. In that case the contribution of the next large valued parameter i.e. *end-to-end delay* is multiplied by the largest weight among the set of weights **S1**.

5.2 Effect of User Feedback

Recall that when the significant parameters are found. The *QoS Manager* selects and calls appropriate protocols depending upon the QoS parameters. A qosReply packet is sent to the source by the *QoS Manager*. Upon receiving a qosReply packet, the source sends a TestQoS packet. The TestQoS packet is delivered to the destination and a flag named statusP_i is set to "done" for the QoS

parameter, P_i. The source, if satisfied sends the extent upto which he/she is satisfied. If the satisfaction of the source falls below 50%, the source shall specify his/her desired QoS parameters again, and the above process shall continue. When the user is satisfied by the QoS provided to TestQoS packets, he/she will start sending actual packets.

The TestQoS packets are simply overheads [6]. The number of these packets depends upon the extent of user satisfaction [7]. However, since there is a cost associated with user perceived QoS, therefore, the number of attempts made by the user for a desired level of QoS is restricted to 3. The user may select any one level of QoS that suits to his/her needs out of that provided by these attempts.

Note that we mentioned it earlier that nodes are operating in ad hoc mode and the type of network we are interested in, is supposed to use 802.11 standards. A consequence of having user feedback is that some of the energy is consumed by the *TestQoS* packets which would have been used for some other useful task. However, we would like to remind that these packets are very small in size and that we have limited these packets to only a few (say 3). Although, there is some additional energy consumption, however, that is the price to be paid to have user feedback about the QoS. However, we believe that it would not be too large to be afforded.

5.3 Probability of User Satisfaction

In order to evaluate the probability of user satisfaction, let us assume a simple scenario in which p_s is the probability that the user is satisfied after an attempt has been made, i.e. after sending a TestQoS packet. The probability that the user is not satisfied after sending a TestQoS packet will then be $1 - p_s$. Let us assume that each attempt is made independently. Then, the probability that the user is satisfied in k such attempts out of n attempts have been made, will then be governed by

$$P_k^{US} = \binom{n}{k} p_s^k \left(1 - p_s\right)^{n-k}. \qquad (7)$$

The above equation is an expression of the binomial distribution for Bernoulli trials. The probability getting exactly one success (i.e. user is

---

[6] There will be overheads for a protocol that is used to get the desired level of QoS. However, those overheads will depend upon the specific protocol.

[7] Although, the level of user satisfaction is not a measurable quantity, however, depending upon the feedback received by letting them to answer a set of questions, one may be able to get a feel of the level of user satisfaction. Either the number of questions successfully answered by an end-user, or by some other measure, may be taken as the user satisfaction. Therefore, let us assume that a user satisfaction of 50% means that 50% of the questions have successfully answered by the end-user.

satisfied in exactly one of the attempts) is given by

$$P_1^{US} = \binom{3}{1} p_s^1 (1 - p_s)^2. \qquad (8)$$

We want that the user to be satisfied in at least one of the attempt out of three attempts have been made. The probability that the user is satisfied in atleast one of the three attempts is given by

$$\psi = \binom{3}{k} p_s^k (1 - p_s)^{3-k}. \qquad (9)$$

For $p_s = 0.5$, $\psi = 0.875$. It means that if the probability of success (i.e. the probability that the user is satisfied after an attempt) is assumed to be 0.5, then the probability of satisfaction of the user in at least one of these attempts is 0.875. As mentioned earlier, the user may select the QoS that fits best to his/her needs, if he wishes to do so.

Note that, in this paper, wherever we referred to the aggregation of QoS parameters, we mean the aggregation of only those parameters whose combined effect may be computed in a reasonable amount of time (i.e. polynomial time). The parameters that we considered in this paper are only for the purpose of example. One has to see which parameters can be computationally combined before actually aggregating their effect. In case, one wishes to see the effect of the parameters that cannot be combined computationally, one may use a technique called QoS filtering. In that, if one wishes to seek QoS based on parameters $(A_1, A_2, ..., A_n)$. One should first seek the QoS based on one of these parameters, say A_1, and then on the resulting set of A_1, one should seek QoS based on A_2, and so on. Further, which parameter should be considered first or what should be the order of parameters in the QoS filtering will depend upon what is the relative importance of the parameters considered.

6 CONCLUSION

Providing QoS in a mobile ad hoc network is a challenging task due to inherent characteristics of such a network. In this paper, we proposed a framework for provision of QoS in a mobile ad hoc network. Our contributions are as follows.

- We proposed that one can aggregate the effect of QoS parameters depending upon the importance or weights assigned to each parameter.
- In our framework, we tried to incorporate the level of user satisfaction about the QoS provided by the network.
- We proposed that there can be a trade-off

between the QoS expected by the end-user and the QoS that may be provided by the network. However, we left it on to the end-user to decide about the trade-off depending upon his/her requirements.
- We discussed overheads incurred in adapting the QoS to the level of expectance of the user.

In summary, we discussed a framework for an aggregated and dynamic QoS based on user satisfaction. Further validation of the framework forms our future work.

7 REFERENCES

[1] S.C. Lo, G. Lee, W.T. Chen, J.C. Liu, "Architecture for Mobility and QoS Support in All-IP Wireless Networks", *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 22, no. 4, May 2004.

[2] B. Li, L. Li, B. Li, K.M. Sivalingam, X.R. Cao, "Call Admission Control for Voice/Data Integrated Cellular Networks: Performance Analysis and Comparative Study", *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 22, no. 4, May 2004.

[3] N. Passas, E. Zervas, G. Hortopan, and L. Merakos, "A Flow Rejection Algorithm for QoS Maintenance in a Variable Bandwidth Wireless IP Environment", *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 22, no. 4, May 2004.

[4] Dutta, W. Chen, O. Altintas, H. Schulzrinne, "Mobility Approaches for All-IP Wireless Networks", *Proceedings of 6th World Multi Conference on Systematics, Cybernetics and Informatics (SCI)*, July 2002.

[5] M. Ghaderi, R. Boutaba, "Towards All-IP Wireless Networks: Architectures and Resource Management Mechanism", *InderScience International Journal on Wireless and Mobile Computing (IJWMC)*, 2005.

[6] J. Liu, V. Issarny, "QoS-Aware Service Location in Mobile Ad hoc Networks", *Proceedings of IEEE International Conference on Mobile Data Management (MDM)*, pp. 224-235, 2004.

[7] H. Zhai, X. Chen, Y. Fang, "How Well Can the IEEE 802.11 Wireless LAN Support Quality of Service?", *IEEE Transactions on Wireless Communications*, vol. 4, no. 6, November 2005.

[8] C.S.R. Murthy, B.S. Manoj, Ad Hoc Wireless Networks: Architectures and Protocols, Pearson Education, New Delhi, 2005.

[9] H.J. Chao, X. Guo, Quality of Service Control in High-Speed Networks, John Wiley, New York, 2002.

[10] Zanella, D. Miorandi, S. Pupolin, P. Raimondi, "On Providing Soft-QoS in Wireless Ad hoc Networks", *Proceedings of International*

Symposium on Wireless Personal Multimedia Computing (WPMC), October 2003.

[11] V. Kone, S. Nandi, "QoS Constrained Adaptive Routing Protocol for Mobile Ad hoc Networks", *Proceedings of 9th IEEE International Conference on Information Technology (ICIT)*, pp. 40-45, December 2006.

[12] M. Mirhakkak, N. Schult, D. Thomson, "Dynamic Quality-of-Service for Mobile Ad hoc Networks", T*echnical Report, Mitre Corporation*, http://www.mitrecorporation.net/work/tech_papers/tech_papers_00/thomson\mp_adhoc/thomson_adhoc.pdf, 2000.

[13] B. Li, "QoS-Aware Adaptive Services in Mobile Ad hoc Networks", *Proceedings of IEEE International Workshop on Quality of Sevice (IWQoS)*, pp. 251-268, 2001.

[14] A.M. Abbas, K.A.M. Soufi, "LANM: Lifetime Aware Node-Disjoint Multipath Routing for Mobile Ad hoc Networks", *Proccedings of IET International Conference on Information and Communication Technology in Electrical Sciences (ICTES)*, 2007.

[15] S. Nelakudity, Z.L. Zhang, R.P. Tsang, D.H.C. Du, "Adaptive Proportional Routing: A Localized QoS Routing Approach", *IEEE/ACM Transactions on Networking*, vol. 10, no. 6, December 2002.

[16] Y.S. Chen, Y.C. Tseng, J.P. Sheu, P.H. Quo, "An On-demand, Link-State, Multipath QoS Routing in A Wireless Mobile Ad hoc Network", *Elsevier Journal on Computer Communications*, 2003.

[17] S. Wu, K.Y.M. Wong, B. Li, "A Dynamic Call Admission Policy With Precision QoS Guarantee Using Stochastic Control for Mobile Wireless Networks", *IEEE Transaction on Networking*, vol. 10, no. 2, pp. 257-271, April 2002.