QoI-aware Analysis of Wireless Networks

Tara Bahadori*, Ahmed Bahjat†, Thomas LaPorta†, Kostas Psounis* and Ram Ramanathan‡
* University of Southern California
† Pennsylvania State University
‡ Raytheon BBN Technologies

***Abstract*—**
The design and analysis of (multi-hop) wireless networks has traditionally focused on end-to-end throughput. In real-world networks however, what matters more is the *Quality of Information* (QoI) of the delivered packets to the application in question. The effect of QoI on network capacity and scalability is not well understood.

We present a simple framework for a QoI-aware approach to analyzing wireless networks. We introduce the notion of a *QoI-rate function* for an application, and apply it in two directions. First, we give a number of examples of such a function, show how a combined function can be obtained when multiple applications are used for a problem, and show how operational capacity regions can be derived with multiple QoI flows. Next, we consider the impact of the desired QoI on the scalability of networks. Using an example of a regular WiFi mesh network and an optical character recognition application, we derive the scalability as a function of QoI. Our key findings are: 1) real-world application QoI typically has a complex, non-linear relationship to rate; 2) QoI-awareness can significantly alter the estimate of operational capacity and scalability of a MANET.

I. Introduction

The doctrine of network-centric warfare relies heavily on the Mobile Ad Hoc Network (MANET) architecture. Analyzing and understanding the capacity and scalability of military MANETs and, more general, tactical wireless networks, is key to the long term success of network-centric operations. Traditional analytical modeling approaches to MANETs have almost always targeted user *throughput* as the metric of interest. However, increased throughput does not always correlate with better user experience – what matters more is the *Quality of Information* (QoI) in the delivered packets in the context of the application in question.

QoI[1] is a composite, multi-dimensional metric that characterizes the information delivered to the application. We discuss QoI in more detail in Section III, including the concepts of QoI functions and QoI-to-Rate translations. Using these definitions, we examine two broad questions: 1) how does QoI, or QoI-to-rate function compose over multiple applications, and what is the operational information carrying capacity in a multi-application QoI setting; and 2) how is the scalability of a network impacted by QoI considerations, specifically, in a regular WiFi mesh network?

Using examples drawn from diverse applications including speech recognition, motion detection, face recognition, fingerprint recognition and optical character recognition we illustrate how QoI-aware network analysis can yield insights. In particular, we show that operational capacity regions can be constructed for multiple flows with different QoI, and show that the desired QoI can have a dramatic impact on the scalability of the network.

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053.

II. Related Work

To date, QoI has been applied in different ways to military networks [10], [5], often when characterizing data coming from a sensor network [1], [2]. More generally, QoI has been used to assist in data retrieval or to facilitate sharing stored data [9]. Unlike these approaches, ours incorporates information quality directly into the design of a tactical network, borrowing from military doctrine that specifies information quality criteria to be used during operations [5]. Tactical applications may specify a desired QoI, and the network seeks to deliver this QoI efficiently.

The capacity and scalability of wireless networks has been extensively studied from a theoretical viewpoint both asymptotically, see, for example, [4], [3], and non-asymptotically, e.g., [6]. More recently, in [11], a formulation for in-practice scalability is given. However, these works do not consider the Quality of Information (QoI) in their formulation. Our work is unique in that it considers capacity regions and scalability in the context of QoI and presents a preliminary framework for handling them.

III. QoI: Application Perspective

QoI will be valued differently by different applications. This reflects the fact that different applications require information of different quality to work properly. Different applications may also require information in different modalities.

Consider the following example. A commander requests the answer to the following question: was there a person on the side of the road at a certain time? The answer to this question may be fulfilled with a relatively low resolution image taken at the proper time. A second question may be: who was the person on the side of the road at a certain time? This question may be answered by using a facial recognition application on a photo of sufficient resolution taken at the proper time. Finally, consider the question: where did the person on the side of the road go when they left? This may require a series of images or a video clip. Each of these queries requires information of different quality.

To support the specification of QoI by users, the labeling of information, both stored and that being retrieved in real time, and to allow the optimization of network control to deliver the maximum operational information content capacity we provide several definitions.

QoI is a composite, multi-dimensional metric that captures the trade-offs of several components to characterize the information ultimately delivered to the application. QoI is composed of both intrinsic and contextual metrics. Intrinsic metrics are those that are valued independently of the use of the information. For example, the freshness of information, i.e., its age, is a function of when the information was generated and will have the same value regardless of the use of the information. Contextual metrics are a function of the use of the information. For example, completeness depends on the use of information. If a photo is being used to count people in a room, it is only complete if it contains all the people in the room; if its use is to determine if at least one person is in the room, then it is complete if it shows enough of the room to see a single person.

Requested QoI is defined as the QoI requested by a user when issuing a query. *Delivered QoI* represents the QoI delivered to the user, either by retrieval of information in real-time or by retrieving information from a database. Requested QoI will be fully specified, including both intrinsic and contextual attributes, because the requestor has knowledge of the application. If the information source and delivery nodes also have knowledge of the QoI requirements and application, then the delivered QoI may also be fully specified. Information that is stored will have only intrinsic attributes specified because it may be used by unknown applications. However, it could be tagged with descriptors that allow contextual attributes to be derived.

In general, QoI may be represented as a vector or a function. Vectors are appropriate to use when the values of the metrics in the vector represent a fixed value. For example, the delivered QoI will be a vector because the values of the metrics are known. Likewise, requested QoI may be a vector if the metrics are specified in terms of minimum acceptable values. The QoI values for stored information will also be a vector, but the vector may not be fully populated as discussed above.

QoI functions allow a requestor of information to define the relationships and trade-offs between QoI metrics. For example, the QoI may degrade as precision of information decreases, but increase as the delay in retrieving the information, called timeliness, decreases. Consider the case of an image. By compressing the image, timeliness is improved but precision is degraded. A QoI function allows the specification to quantify this tradeoff. When using a QoI function, requested QoI may be in the form of QoI(metrics) $>= Q$. This, in effect, defines a surface of acceptable QoI. In our examples, for simplicity, we use a QoI function to determine a scalar.

To set network controls and determine network scalability, we must translate QoI requirements into allocated network resources. In this paper we consider transmission resources. To accommodate resource assignment, we translate QoI requirements into a required network rate using a *QoI-Rate function* (QRF). The QRF cannot capture all QoI metrics. For example, it cannot capture the freshness or completeness of information because transmission rate does not impact these metrics. But, it can be used to translate precision (which is related to file size) and timeliness (which is related to latency).

To evaluate the impact of QoI on applications, we define a variable called the application score (*App-score*). The App-score is used to provide a normalized subjective measure of the quality of information given to an application for a given set of QoI metric values.

IV. Multi-application QRF functions

In the previous sections we defined the QoI-to-Rate Function (QRF) for scenarios where a single task/application will be served by a single node. In this section we extend this definition to the cases in which multiple tasks are serviced by multiple nodes in a network.

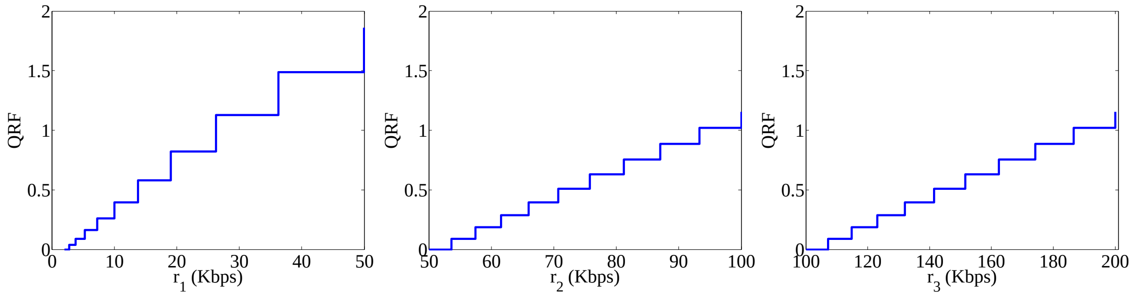
First, let's consider the scenario in which a single wireless node is set to send data of different tasks to a base station, that is, we have a single user serving different communication tasks. The goal is to divide the available bandwidth/rate between the various tasks in a way which maximizes the total QoI. In the rest of this section, each task/application corresponds to a scalar QoI value, and this value is associated with a corresponding required rate by the QRF mapping function as mentioned earlier.

Suppose T tasks/applications are assigned to a node, where each task i, $1 \leq i \leq T$, has a QRF function $Q_i(r)$. The multi-application QRF mapping function, $Q(r)$, is defined as follows:

$$\begin{array}{rc} & Q(r) = \max_{\mathbf{r}} \sum_{i=1}^{T} Q_i(r_i) \\ \text{subject to:} & \\ & \sum_{i=1}^{T} r_i \leq r, \end{array} \tag{1}$$

where $\mathbf{r} = [r_1, \cdots, r_T]$ is the vector of rates allocated to different tasks and r is the total available rate. The function $Q(r)$ finds the maximum QoI that can be achieved subject to the total available rate r. Note that achieving this rate allocation in an operational system requires scheduling algorithms that are beyond the scope of this paper.

As an example for the definition above, consider the QRF mapping functions for the three tasks shown in Figure 1. As shown in the plot, the first application is useful as long as the provided rate is a few Kbps, whereas the other two applications require higher rates to start being useful and yield positive QRF values. The multi-application QRF mapping functions for these three applications is shown in Figure 2. This example highlights a few properties of the multi-application QRF mapping functions: (i) Clearly, the resulting rate allocation depends on the individual QRFs and the QoI of each application given the rate it gets. For example, in Figure 1, for r close to 100 (to be exact for $r = 102.22$ Kbps), the optimal rate assignment turns out to be $[r_1, r_2, r_3] = [36.24, 65.98, 0]$, since the first application does not yield higher QRF values if the assigned rate is further increased from 36.24, and the third application does not give any positive QRF values for rates less than 110, thus the remaining rate (65.98=102.22-36.24) is assigned to the second application. (ii) The multi-application QRF mapping function is less smooth than the individual QRF mapping functions because combining discontinuous functions tends to produce even less continuous functions.



(a) (b) (c)

Fig. 1: The QRF mapping functions for three tasks.

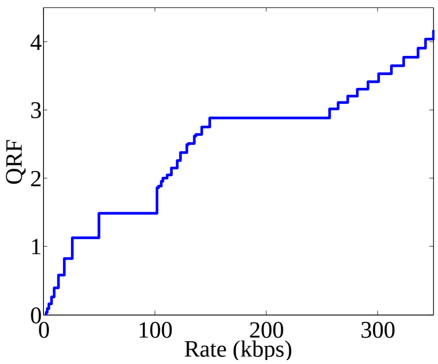


Fig. 2: The multi-application QRF mapping function for the QRF mapping functions shown in Figure 1.

A. Real world applications

Next, we provide some examples utilizing real world applications. We consider five applications that are of interest to military networks: speech recognition, face recognition, motion detection, fingerprint recognition and optical character recognition (OCR).

First we need to find the QRF mapping function for each application. A general rule to compute the QRF value of an application is the following: $\mathrm{Q(rate)} = \mathrm{accuracy(rate)} \times \text{App-score}$, where App-score is a subjective metric decided by the user or network operator to distinguish between different applications, as mentioned in the previous section. Quite often, for data types such as images the literature provides accuracy(size) curves instead of the desirable accuracy(rate) ones. In order to convert the curves from the literature to what we need, we use the timeliness requirement and define the concept of an *equivalent rate* as follows:

$$\textit{Equivalent Rate} \equiv \frac{\text{Media Size}}{\text{Max Tolerable Delay}}. \qquad (2)$$

Table I lists the values of App-score and the Max Tolerable Delay for the five applications that we consider. Note that these values are arbitrary in the sense that they are subjective. Using the parameters in Table I together with accuracy(size)/accuracy(rate) curves from the literature[1] we find the curves shown in Figure 3.

	Image size	Max delay	App-score
Speech Recognition	–	–	2
Motion Detection	–	–	1.5
Face Recognition	5 Kb	20 mSec	2.2
Fingerprint Recognition	2 Kb	100 mSec	0.2
OCR	20 Kb	100 mSec	3

TABLE I: Parameter values used.

To make our example more specific, we define two sets of multi-application tasks: (T1) {motion detection, speech recognition, face recognition} and (T2) {fingerprint recognition, OCR, face detection}. The idea here is there is a node which is using concurrently all three applications, and different nodes may be using different applications as well as different number of applications. The corresponding multi-application QRF mapping functions are shown in Figures 4 and 5, respectively. The arrows in the figures show approximately when each application begins to receive service. In general, if the rates at which applications begin to be useful are distant enough from each other in the corresponding QRF mapping functions, we expect to have in the resulting multi-application QRF function as many "regions" as the number of applications.

V. OICC REGION OF THE NETWORK

In order to demonstrate how one may use the QRF functions to compute the achievable QoI that a network may yield while taking into consideration multiple nodes which may depend to each other, for example, due to wireless interference, we consider a simple network configuration. Note that we use the term OICC region, which stands for Operational Information Content Capacity region, for this quantity. This quantity is reminiscent of the well known concept of capacity region, or achievable rate region, which is used in traditional network theory to express the vector of rates that can be supported by the network, see, for example, [4], [6]. The fundamental difference here is that we are not merely interested in the rates

[1]The papers we used from the literature are the following: for speech recognition [7], for face recognition [14], for motion detection [15], and for fingerprint recognition [8]. For the curve for Optical character recognition we have used results from experiments with the open source tool Tesseract (http://code.google.com/p/tesseract-ocr/).

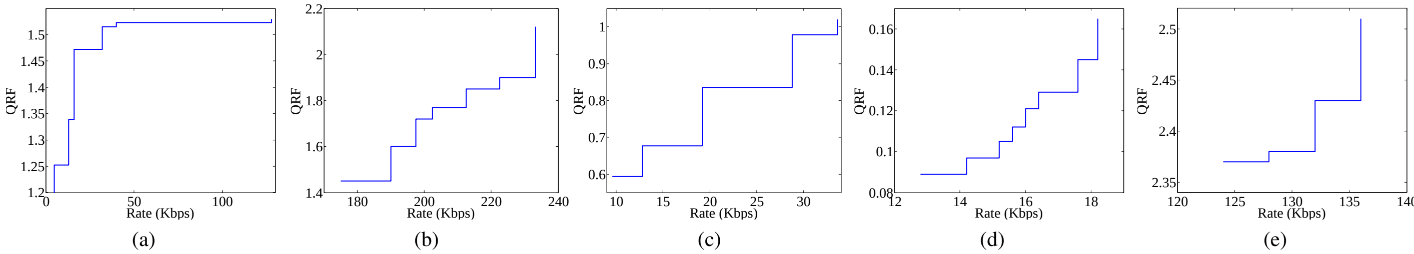


Fig. 3: The QRF mapping function for the individual tasks. (a) Speech Recognition, (b) Face Detection, (c) Motion Detection, (d) Fingerprint Recognition, (e) OCR.

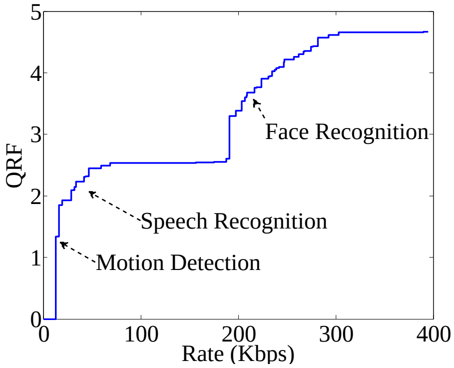


Fig. 4: The multi-application QRF mapping function for the first task set, T1.

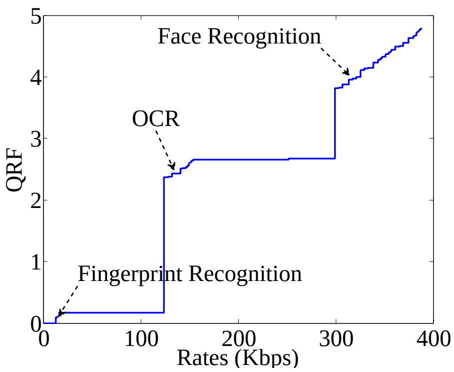


Fig. 5: The multi-application QRF mapping function for the second task set, T2.

that can be supported. Instead, we care about the information content that can be supported, or, put it differently, in the quality of the information that can be transferred.

Suppose we have two mobile nodes and a stationary base-station to which the mobile nodes send the data of their assigned tasks. The task sets (T1) and (T2) defined above are assigned to the first and second node, respectively. Using a centralized scheduler in the base-station (e.g. TDMA), this network can achieve the rate region shown in Figure 6. Note that this two-dimensional plot shows all rate vectors that are achievable in this network. For example, the first user maybe sending at a rate equal to 600 Kbps but then the second user does not have any rate, or both users may send at 300 Kbps each, but the network cannot support both users sending say at 400 Kbps.

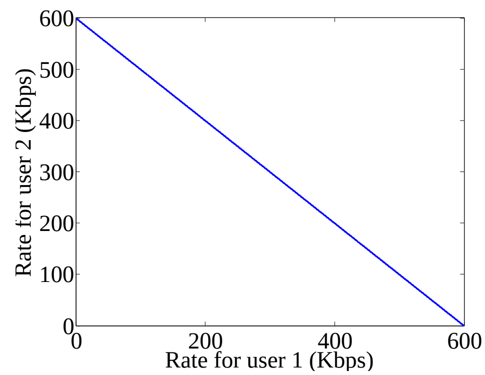


Fig. 6: The achievable rate region for the base-station configuration.

The OICC region (achievable QoI region) corresponding to the base-station configuration is shown in Figure 7. This region is computed by considering all possible achievable rate vectors and computing the QoI obtained from each achievable rate vector. Putting all achievable QoI vectors together yields the plot shown in Figure 7.

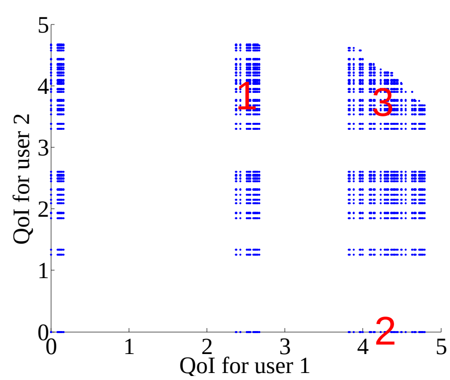


Fig. 7: The OICC region in the base-station configuration.

Figure 7 highlights some properties of the OICC region of the network under study which we discuss below. Note that the concept of an OICC region is quite complex and general, and it is beyond the scope of this paper to provide a full exposition of the concept.

- The region is composed of $(n_1+1)\times(n_2+1)$ discrete regions, where n_1 and n_2 are the number of distinguishable applications available to the first and the second user respectively. For example, in figure 7, region 1 corresponds to the speech recognition task of the first user and the face recognition task of the second user. As another example, region 2 corresponds to the case in which the second user does not achieve any QoI because the user gets no rate or too small rate, whereas the first user does achieve some QoI.
- Region 3 in the figure clearly shows the effect of the rate region on the shape of the OICC region (achievable QoI region). The upper right corner is constrained because the corresponding rate vectors are not achievable.

VI. QoI and Scalability

As discussed in the previous sections, QoI is typically a highly non-linear function of the number of packets delivered at the destination. For example, an image recognition application will perform nearly the same for a range of compression ratios, suddenly experiencing a spike at some particular ratio. Given this, simply delivering the highest possible rate is not necessarily the best option from a user viewpoint. More generally, we would like to know the capacity of a network (and relatedly, the scalability achievable) if we wanted *not* the maximum *throughput* (as traditional asymptotic analyses targets), but the maximum *QoI*. In other words, given a certain QoI requirement, what is the scalability?

In [11] the in-practice scalability of a network was considered in terms of the "residual capacity", that is the difference between the available and used capacities at a node. Assuming homogeneity, a coarse-grained model was developed based on the simple observation that the network can support the offered flows if and only if the residual capacity at every node in the network is positive. A generic expression was derived there:

$$\mathsf{R} = \mathsf{A} - \sum_j (1+\gamma_j)\cdot \mathsf{D}^{\mathsf{j}}, \tag{3}$$

where R is the *residual capacity* at a given node, indicating the capacity remaining at a node after taking into account the load from all traffic sources from all nodes. This is the difference between the *available capacity* A, and the capacity D^j demanded by each source j (control overhead is regarded as one kind of source), and γ_j is the "contention factor" that is a rough inverse measure of the spatial reuse, and indicates the number of nodes that have to defer on a transmission. We refer the reader to [11] for details on the contention factor and derivation of Equation (3), and focus here on applying the expression to a QoI setting.

Suppose that a network has M types of flows. Each flow j has its own QoI-rate function $QRF_j(u_j)$ where u_j is one of the possible QoI values[2] for the application corresponding to flow j. Let $\xi(s)$ be a function that maps a source rate s to its average contribution to a node's demanded capacity per Equation (3). ξ depends upon a number of factors such as the average length of the flow, whether it is unicast or multicast etc. and is instantiated in the context of the particular network. With these, the demanded capacity from flow j is

$$\mathsf{D}^{\mathsf{j}} = \xi(QRF_j(u_j)), \tag{4}$$

where u_j is the desired QoI of application using flow j.

Combining with Equation (3), we have:

$$\mathsf{R} = \mathsf{A} - \sum_j (1+\gamma_j)\xi(QRF_j(u_j)). \tag{5}$$

We are given application QoI requirements, and QRF can be obtained by empirical studies or given as part of the application profile. ξ needs to be calculated on a case by case basis given the topology and the traffic profile.

We illustrate our approach using a specific example. Consider a regular mesh network (also known as a "manhattan grid") of N nodes using 802.11g. Suppose there is an Optical Character Recognition (OCR) request from node i to node j once every 5 seconds with i and j being chosen at random[3]. Suppose further that we require a particular application QoI (score) q. How many nodes can the network support (i.e., what is the upper bound on N) for each value of q?

Figure 8 shows our experimental results for the application "score" achieved by an Optical Character Recognition (OCR) application as a function of the number of 1 kb packets received. It clearly shows a "step-function" behavior – as the number of packets increases, the recognition score is constant, and then at some threshold packet number, the score jumps to a new level. There is some overlap because at some range of number of packets, the application score may be one of two levels. For example, between approximately 1100 and 1500 packets, the application score may be 99 or 100. The vertical dashed lines show the number of packet thresholds above which a particular score is guaranteed. For example, above 1100 packets, a score of 99 is guaranteed. Conversely, if we need a score of 99+, we need at least 1100 packets.

We now apply Equation (5). We take the QoI to be the scalar application score. Then $QRF(q) = f(q)/5$ where f is the function represented by Figure 8, that is, for each application score q, the projection of the vertical line to the x-axis gives the value of the number of packets $f(q)$ required per recognition. Since there is one recognition stream every 5 seconds, this value is divided by 5.

Since the source and destination are chosen randomly, the scope of a flow is the average path length. In [12], the average path length for a regular mesh of N nodes is shown to be $\frac{2}{3}\cdot\sqrt{N}$. Thus, the used capacity per node in an N node network each node generating x bps is $\xi(x) = \frac{2}{3}\cdot\sqrt{N}\cdot x$ approximately.

Finally, the contention factor for unicast traffic in a mesh network is $\gamma_1 = 7$ (see Figure 9) – the receiver plus six other nodes have to defer on this transmission. Since we consider a

[2]Per flow QoI allows flexibility to have the range of values appropriate for the application for that flow.

[3]This is not intended to model any particular operational scenario, only an example to illustrate our model in a simple manner.

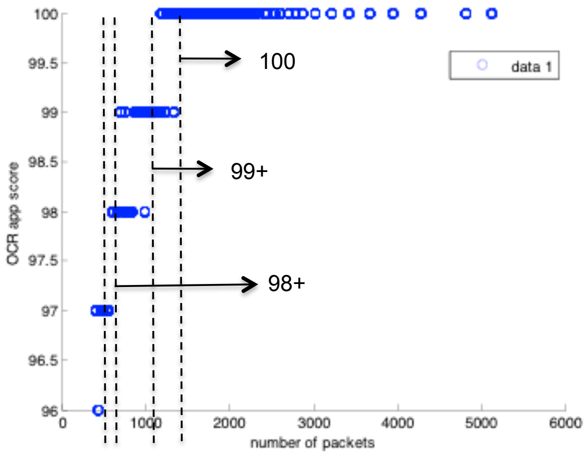


Fig. 8: The application score of an OCR application.

stationary network, we ignore the routing overhead and assume the overhead due to MAC control messages is negligible.

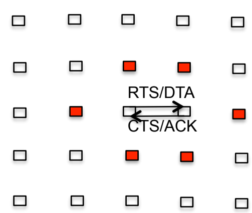


Fig. 9: The unicast contention factor with 802.11g MAC.

Substituting in Equation (5), and noting that maximum scalability is when R = 0, we have

$$A = (1+7) \cdot \frac{2}{3} \cdot \sqrt{N} \cdot \frac{f(q)}{5}, \quad (6)$$

which simplifies to

$$N = [\frac{15 \cdot A}{16 \cdot f(q)}]^2. \quad (7)$$

Using A = 22 Mbps as the effective data rate for 802.11g [13], and using q=97, 98, 99, 100 and taking f from Table II, the scalability of the network for various QoI's (application scores) are shown in the table below.

App score(q)	Packets (f(q))	Scalability
100	1500	189 nodes
99	1100	351 nodes
98	600	1181 nodes
97	500	1701 nodes

TABLE II: Scalability of a regular mesh as a function of the desired QoI (application score).

Clearly, there is a remarkable difference in the scalability depending upon the QoI desired. For instance, if we can afford a 98% application score instead of 99%, we can get a 330% increase in scalability! That QoI makes a difference is not suprising, but the *magnitude* of difference is surprising, along with the fact that there are some critical thresholding points. Our preliminary work shows that scalability analysis with QoI awareness has the potential to open up new tradeoff points with significant potential benefits in scalability.

VII. Concluding Remarks

We have presented the first QoI-aware analysis of wireless networks. Using the concept of QoI-to-rate (QRF) function, extended to multiple applications, we have characterized the operational information content capacity (OICC) of multiple QoI flows, analogous to traditional capacity regions. Additionally, using the framework in [11], we have analyzed, very approximately, the in-practice scalability of an OCR application in a regular mesh network with 802.11g radios. Our results show that there can be a significant increase in scalability if we can afford slightly less QoI.

Our work is preliminary, but points to considerable potential for this research area. In future work, we hope to further formalize the notions introduced here, consider a large number of flows, and examine the OICC and the scalability for realistic topologies with multiple flows having different desired QoI.

References

[1] C. Bisdikian, D. Verma, L. Kaplan, M. Srivastava, and D. Thornley. Defining quality of information and metadata for sensor-originating information. In *4th USMA Network Science Workshop*, 2009.

[2] M. Burgess, W. Gray, and N.J. Fiddian. Establishing a taxonomy of quality for use in information filtering. In *Proceedings of the 19th British National Conference on Databases: Advances in Databases*, 2002.

[3] M. Grossglauser and D. Tse. Mobility increases the capacity of ad-hoc wireless networks. *IEEE/ACM Transactions on Networking*, 10:477–486, 2002.

[4] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE TRANSACTIONS ON INFORMATION THEORY*, 46(2), March 2000.

[5] Department of the Army Headquarters. Field manual 6-0: Mission command: Command and control of army forces, 2003.

[6] Apoorva Jindal and Konstantinos Psounis. The achievable rate region of 802.11-scheduled multi-hop networks. *IEEE/ACM Transactions on Networking*, 17(4), August 2009.

[7] B.T. Lilly and K.K. Paliwal. Effect of speech coders on speech recognition performance. In *Signal Processing and Its Applications, 1996. ISSPA 96., Fourth International Symposium on*, volume 1, pages 333 –336, August 1996.

[8] A. Mascher-kampfer, Herbert Stgner, and Andreas Uhl. Comparison of compression algorithms impact on fingerprint and face recognition accuracy. In *in Visual Computing and Image Processing VCIP 07, Proceedings of SPIE*, 2007.

[9] P. Missuer, S. Embury, M. Greenwood, A. Preece, and B. Jin. Quality views: Capturing and exploiting the user perspective on data quality. In *Proc. of ACM VLDB*, 2006.

[10] Department of Defense. Joint publication 6-0: Joint communications system, 2006.

[11] R. Ramanathan, R. Allan, P. Basu, J. Feinberg, G. Jakllari, V. Kawadia, S. Loos, J. Redi, C. Santivanez, and J. Freebersyser. Scalability of mobile ad hoc networks: Theory vs practice. In *MILCOM 2010*, pages 493 –498, 31 2010-nov. 3 2010.

[12] J. Silvester and L. Kleinrock. On the capacity of multihop slotted aloha networks with regular structure. *IEEE Transactions on Communications*, 31:974–982, 1983.

[13] wndw.net. Wireless networking in the developing world, 2nd ed., 2007.

[14] Yanjun Yan, R. Muraleedharan, Xiang Ye, and L.A. Osadciw. Contourlet based image compression for wireless communication in face recognition system. In *Communications, 2008. ICC '08. IEEE International Conference on*, pages 505 –509, May 2008.

[15] Jar-Ferr Yang, Shih-Cheng Chang, and Chin-Yun Chen. Computation reduction for motion search in low rate video coders. *Circuits and Systems for Video Technology, IEEE Transactions on*, 12(10):948 – 951, October 2002.