Scalability and Satisfiability of Quality-of-Information in Wireless Networks

Scott T. Rager* Ertugrul N. Ciftcioglu† Ram Ramanathan‡ Thomas F. La Porta* Ramesh Govindan§
*The Pennsylvania State University, University Park, PA 16802
†IBM Research, Yorktown Heights, NY 10598
‡Raytheon BBN Technologies, Cambridge, MA 02138
§University of Southern California, Los Angeles, CA 90089
Email: rager@psu.edu, enciftci@us.ibm.com , ramanath@bbn.com, tlp@cse.psu.edu, ramesh@usc.edu

***Abstract*—Quality of Information (QoI) provides a context-dependent measure of the utility that a network delivers to its users by incorporating non-traditional information attributes. Quickly and easily predicting performance and limitations of a network using QoI metrics is a valuable tool for network design and deployment. Even more useful is an understanding of how network components like topology, bandwidth, protocols, etc. impact these limitations. In this paper, we develop a QoI-based framework that can be used to provide this understanding of limitations and impact by modeling the various contributors to delay in the network, including channel rate and contention, competing traffic flows, and multi-hop propagation effects, and relating them to QoI requirements, especially completeness and timeliness. Analysis shows that large tradeoffs exist between different network parameters, such as QoI requirements, topology, and network size. Simulation results also provide evidence that the developed framework can estimate network limits with high accuracy. Finally, this work also introduces and shows an example of *scalably feasible QoI regions*, which provide upper bounds on QoI requirements that can be supported for certain network applications.**

I. Introduction

Traditionally, approaches to studying network scalability and performance limitations have been focused on finding theoretical limits on throughput and delay. Additionally, much of this work focuses on one network protocol or parameter at a time. While this research is useful for determining upper bounds on performance for specific cases, it lacks several considerations that we address in this work.

First, only considering delay or network throughput does not capture the impact of these metrics on the data's usefulness. In the same manner, simply using throughput as a metric implies that all delivered data is equally useful. As we show in Section III, this assumption is unrealistic for many applications. There, we use image selection as a concrete example to exemplify this point. For these reasons, we adopt *Quality of Information* (QoI), which can include a number of information attributes (many of which are context-dependent), such as completeness, diversity, credibility, creation time, and timeliness, as our measure of network performance. Specifically, in this work, we focus on satisfiability of completeness and timeliness, explained in detail in Section III.

The second contribution differentiating this work from previous analysis is that we focus on providing a framework that can be adjusted to determine scalability and QoI satisfiability for any instance of a network. The wide applicability and easy reuse of this framework make it easy to compare protocols, topologies, traffic models, etc., without creating extensive simulation or experimentation testbeds, which is a much more difficult task.

Finally, in contrast to discovering theoretical, asymptotic limits, this framework seeks to quickly obtain an accurate estimate of a network's abilities. While the goal of this paper is not in optimizing network performance, we show in Section VI that this model can also be used to quickly and easily understand the impact of parameters and design choices. This ability provides a secondary benefit to network designers of allowing them to compare networks and identify tradeoffs.

As an example, imagine given the task of deploying a wireless sensor network for a particular task or application. Given a proposed network with a defined signature describing its size, parameters, protocols, and expected traffic, what is the level of QoI requirements it can support? Alternatively, consider the converse: Given a certain QoI that is desired by users of a network, what is the maximum number of nodes that the network can support? Which has a bigger impact on this scalability, the imposed data requirements, or the strict timeliness requirements?

Currently, no framework exists that provides a methodology to predict scalability and performance with respect to QoI requirements. In this paper, our contribution is to provide such a framework, which we do in Section IV. In Section V, we prove its effectiveness by comparing it to simulations performed in the ns3 network simulation package. We provide examples of how it is also useful in network design in Section VI. Finally, we also take the concept one step further in Section VII with the introduction of a *scalably feasible QoI region*, which describes the maximum QoI capacity of a particular network scenario.

Research was sponsored by the U.S. Army Research Laboratory under the Network Science Collaborative Technology Alliance, Agreement Number W911NF-09-2-0053.

II. Related Work

The capacity and scalability model derived in this work is inspired by the symptotic scalability framework outlined in [1], which has been previously applied to content-agnostic static networks [2] and mobile networks [3]. Other works characterize the capacity of wireless networks, like [4], [5], but all do so differently by considering how networks scale asymptotically or by analyzing specific network instances instead of developing a general model. Experimental techniques, like Response Surface Methodology [6], for example, may be applied to solve the problem we do, but these require complex test beds instead of a compact mathematical framework.

A large number of works provide definitions for Quality of Information and frameworks utilizing it. We will address only the most relevant ones here. Primarily, QoI has been used in scheduling and has been considered from a number of various angles, including control choices of data selection [7], routing [8], and scheduling/rate control [9], [10].

The work in [11] evaluates the impact of varying QoI requirements on usage of network resources, which is certainly related to this paper. Our focus is on a broader scale than this work, though, by modeling an entire network instead of a single node as the authors do in [11].

Additionally, [12] outlines a framework called Operational Information Content Capacity, which describes the obtainable region of QoI, a notion similar to the *scalably feasible QoI region* developed here. These approaches use a general network model, though, and do not provide any method for determining the possible size of the network or impact of various network design choices like medium access protocols.

In Section III, we use similarity-based image collection as an example of an application that is best evaluated using QoI. This application has previously been considered in [13] and [14]. Our scope is greater than that of [13], which does not consider attributes of timeliness, nor the consideration of transmission rates and network topology. [14] considers a smartphone application where different queries called Top-K, Spanner, and K-means Clustering are defined. We use these same similarity-based image selection algorithms, but we provide new methods of quantifying QoI from them.

III. QoI Model

QoI is a metric that can be defined for an application to give a more meaningful value to information by capturing attributes such as timeliness, age/freshness, completeness, accuracy, precision, etc. For example, information that contributes to a decision-making process may only be useful if it arrives before the decision must be made, or it may have varying usefulness based on how similar or dissimilar it is to other data already collected.

The specific details of which attributes are considered and how they contribute to QoI is application-dependent. The chosen QoI metrics are stored as a vector associated with a data item. There are two ways to define QoI using that vector. In one, a QoI function is defined that accepts the vector of metrics and provides a scalar value as an output like in [11], [15] and many others. In the second, as explained in [16], a vector of minimum values for each QoI metric can be specified, and data can be evaluated based on whether it satisfies all of the QoI requirements or not. We choose the latter method of determining QoI and focus on establishing the edges of QoI satisfiability for the vector of metrics, which defines the boundaries of maximum achievable QoI regions in the metric space.

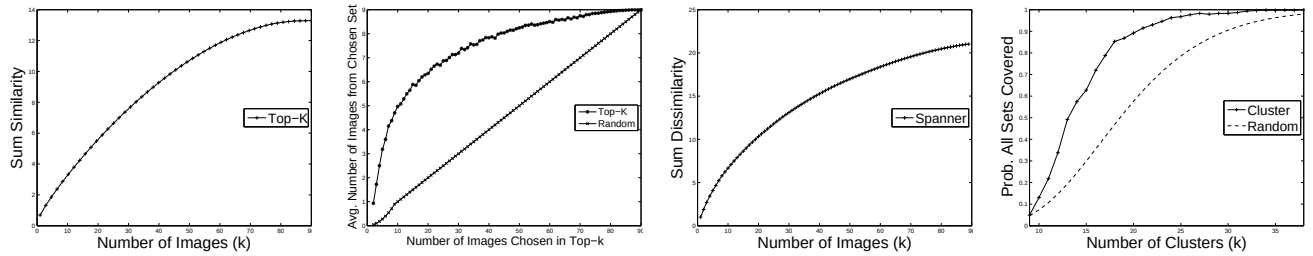
We choose to use two QoI attributes, one that is time-based and one that is information-content-based. The first attribute, whose importance has already been described, is timeliness, T, of data. For the second attribute, we present a notion of *completeness*, C, which we show can be defined multiple ways, depending on the context. Together, a QoI requirement of $\mathbf{q} = \{C, T\}$ specifies a quantity of data that must be delivered as well as a deadline by which it must arrive to be useful. While timeliness is not necessarily a new concept, completeness is less well defined. Therefore, we identify three image selection algorithms, one that selects images based on similarity and two that select images based on diversity, and show how they can be evaluated with completeness.

A. Image Selection Algorithms

As a motivating example, we choose a network in which nodes store photographs that are to be exchanged or collected at one or more data sinks. This example covers surveillance missions of military tactical networks or camera sensor networks. Without adopting any one specific application, we present two query algorithms from [14], one that helps return images similar to a target, and one that helps return a diverse set of images. We also present ways to measure the completeness of each, providing a quantified value of a QoI metric.

Each of these queries require measuring the similarity or dissimilarity of two images. To get a similarity measurement, we use the same choice as was shown to be effective in [14]. This similarity is based on a technique called Color and Edge Directivity Descriptor (CEDD) [17], which uses qualities inherent to a photograph like lightness, contrast, and color. The similarity between two images can then be given as a scalar by calculating the *Tanimoto Similarity* [18] between their CEDD vectors. The dissimilarity value is simply defined as 1 minus the similarity.

1) Selecting Similar Images: The first type of query we introduce occurs when a user already has one image of a particular area or object of interest and would like to obtain similar images to get a more complete view of that specific scene or object. For example, if a user has a picture of an unknown suspicious person entering a building, but the person is not identifiable from that image, it would be useful to collect more images that are similar to that one with the possibility that another picture may have a better view of the person in question that can be used for identification or more context. Called **Top-K**, the query algorithm used



(a) Top-K: Sum Similarity (b) Top-K: Expected Sets Covered (c) Spanner: Sum Dissimilarity (d) Clustering: Prob. all Sets Covered

Fig. 1. Completeness metrics for the three image selection algorithms. Each exhibits a diminishing return as more images are added.

for this application will choose the k images with the most similarity with respect to the target image.

Considering the goal of the Top-K query, we can evaluate the completeness of the result in one of two ways. First, we can use the similarity of the images as a value representing each image's effectiveness in providing a more complete view of the target object or scene. If we sum the similarity of all k images returned by the algorithm, we get a representation of completeness, which we naturally call *Sum Similarity*. While this measure of completeness is abstract, it can be refined in an actual implementation through testing and evaluating. This definition of completeness is useful, though, because it can be applied without any predetermined knowledge of the environment or pool of images.

Often, though, we can partition the environment in which the network operates into a number, n, of distinct settings or areas. In those cases, we can utilize a second method of quantifying completeness. Assume that each image belongs to one of these n sets, related to the setting it depicts. Naturally, then, when executing a Top-K query, the goal is for the algorithm to return images from the same set as the target image. Completeness can then be given by the fraction of images returned that are in the same set as the target image.

2) Selecting Diverse Images: In contrast, given the set of all photographs available in the network, we might want to return the set of k images that exhibits the most diversity, ideally providing a user with a good sampling, or *complete view*, of available images. For instance, such a result would be quite useful in a surveillance mission. We present two query algorithms that can be used to achieve this goal.

One query that provides diverse images is known as the **Spanner** of the set of known photographs. For the Spanner algorithm, we employ a greedy algorithm similar to that in [14] to simplify implementation and to define a *Sum Dissimilarity* metric. Here, the algorithm first chooses the two images with the greatest dissimilarity between them from all available images. Then, each successive image is chosen to be the one with the greatest minimum distance between it and all images already chosen, until k images are selected. This minimum distance between the image being selected and the images in the collected set is the value added to the running cumulative completeness metric of *Sum Dissimilarity*. Since the Spanner algorithm's goal is to provide images at the edges of the available feature space, the Sum Dissimilarity represents a measure of its completeness because a higher level of dissimilarity is providing a more complete view of the feature space.

The other query that can achieve a complete view over all images is **Clustering**. In the Clustering algorithm, all images are separated into a k clusters based on their pairwise distances using any version of a k-means clustering algorithm, where k is given by the user. Then, the most central image from each cluster is returned. Here, assuming that the photographs of the same settings or objects of interest exhibit similar characteristics, Clustering should provide a complete view of the network's environment.

Both Spanner and Clustering algorithms can also be evaluated using the model in which the environment is split into n sets. With this model, we can define completeness as either the number of sets represented by at least one of the k images returned or the probability of all n sets being represented by at least one image when k are returned. Here, though, we only show results for the second definition.

3) Experimental Results: To provide example values of these completeness metric definitions, experiments applying each query algorithm were run on a set of pictures taken at $n = 9$ different settings around the Penn State campus. Each of these 9 settings is of a pictorially different area, e.g. a particular building, a downtown street, or a lawn setting, and over 20 images of each was taken. Then, for individual trials, 10 images from each set were randomly selected to create an image pool of 90 pictures. The three algorithms were run over these 90 images, with the target image being randomly selected in the case of Top-K. Results for each of the different completeness metrics were averaged over $1,000$ trials are shown in Figures 1(a) to 1(d).

Figure 1(a) shows the average sum similarity of images returned by the Top-K algorithm. Figure 1(b) provides the second definition of completeness for the Top-K algorithm, the number of images matching the set that the target image was randomly chosen from. Completeness results when dissimilarity is the objective are shown in Figures 1(c) and 1(d). Specifically, Figure 1(c) depicts the average Sum Dissimilarity returned by the Spanner algorithm, and Figure 1(d) represents the empirical probability of all 9 sets being represented in the k returned images. For reference, we also include expected values for the metrics in Figures 1(b) and 1(d) if the images were selected from the entire image pool at random, i.e., without regard for image similarity or dissimilarity.

All of these figures exhibit the diminishing returns of

completeness as more images are collected. This effect is important because it visually shows how QoI differs from throughput. As seen in these graphs, transmission of successive images does not result in a linear gain in completeness. For example, in Figure 1(b), it is evident that a value of only $k \approx 10$ is needed to collect 5 images matching the target content, while collecting an additional 2 from the same set usually requires collecting over twice that number of pictures.

As a second example, Figure 1(d) shows that jumping from $k = 10$ to $k = 20$, the likelihood of capturing at least one image of every setting grows substantially from just over 10% to approximately 90%. To approach probabilities close to gaining that final 10%, however, requires a jump to $k \approx 30$.

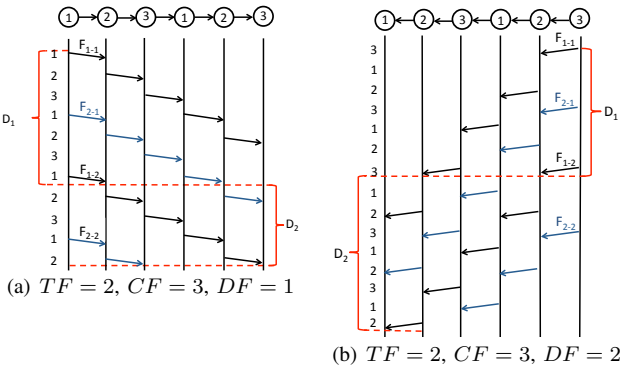
The relationship between the number of images and completeness in each of these graphs also shows that obtaining a certain value of QoI or completeness requires a different number of images depending on the set available and their similarities. We can denote the number of images required to achieve a level of completeness, C, as $k_{req} = Q(C)$. This relationship will be useful later in determining capacity and scalability limits.

B. Further Discussion of QoI

We have defined and provided examples for a number of ways that completeness can be defined and used to obtain a concrete data requirement from a contextual QoI requirement. Throughout the rest of the paper, we mainly use sum similarity as the completeness metric, but any of the definitions of completeness used here, or any other QoI requirement that can be translated into a data requirement, for that matter, can be used in the following formulation in Section IV.

Also, note that QoI and its usage in understanding networks is not exclusive to these metrics and applications. On the contrary, the model used in the capacity and scalability analysis of Section IV is meant to be an in-depth example of this concept. Modifications to account for different data size requirements should be quite straightforward, and extensions to other time-based metrics should be possible with careful extensions to the framework.

Finally, while metadata associated with photographs may be useful in obtaining similar goals to those given in this section, relying on such information is problematic because metadata is not guaranteed to be available, and it is not as universally applicable as content-based retrieval. For example, tags describing the image contents would require users to participate by entering this information, which is a time-consuming and unreliable step that users are likely to ignore. Location and time stamps may be automatically applied by the device allowing an application to filter images accordingly, but these tags often do not account for factors such as the direction of the camera or obstructed views. Content-based processing of these images, though, can be applied to any set of images.



(a) $TF = 2$, $CF = 3$, $DF = 1$

(b) $TF = 2$, $CF = 3$, $DF = 2$

Fig. 2. Example of line network using TDMA highlighting source of delays. (Node labels are TDMA slot assignments.)

IV. QoI Scalability

Given the nonlinear returns of completeness and importance of timeliness outlined in the previous section, we contend that simply establishing the highest average supportable rate should not be the only goal of a QoI-aware network. With this knowledge, we set the goal of determining the capacity of a network (and relatedly, the scalability achievable) with respect to *QoI requirements*, instead of the maximum throughput.

A. QoI Satisfiability Framework

In order to establish the framework, we examine an arbitrary flow, F_1, in the network that has a QoI requirement of $\mathbf{q} = \{C, T\}$, where C is the minimum required completeness metric of choice, such as sum similarity as explained above, and T is the required timeliness. This flow will have a data size requirement, which is given by a chosen QoI function $Q(C)$ like those discussed in Section III. Using the example applications from III, for example, $Q(C)$ can return the number of images, k_{req}, required to achieve the requested completeness C according to established relations like those in Section III. Assuming each image has an average size of I_S, then we can also use B to describe the total number of bits required by the flow, $B = k_{req} * I_S$. To match realistic network implications, we assume this data will be transmitted in a series of packets with size P_S bits each. The number of packets per flow, then, is simply $P_N = \lceil B/P_S \rceil$. We assume that each node in the network can transmit at W bits per second when it is allocated media access.

Our goal is to establish the limits at which this arbitrary flow on average can no longer be completed with the QoI requirements satisfied. We build and explain our model for achieving this goal by working through an example TDMA line network, a portion of which is shown in Figures 2(a) and 2(b) (We discuss addressing non-TDMA networks in Section VIII). In this network, we assume a simple 3-slot TDMA scheme, which allows each node equal time access to the medium and removes any potential interference or hidden terminal issues. Each node in the figure is labeled with its allocated slot. For simplicity, we will assume that one slot is appropriately sized to transmit a single packet, i.e.,

$T_{slot} = P_S/W$ and that packets use static routes calculated beforehand such that the overhead is not a consideration here.

Now, two factors, D_1 and D_2, contribute to the total delay of completing F_1. The first contributor, D_1, is the end to end delay incurred by sending the B bits across the entire path. To quantify this delay, we must consider several factors beyond just the available bandwidth and number of packets. First, each node can only utilize its allotted channel time, creating a Channel Factor of $CF = N_{frame}/N_i$, where N_i is the number of slots allocated to node i and N_{frame} is the total number of slots in each frame. In this case, $N_i = 1, \forall i$ and $N_{frame} = 3$. The second factor to consider for D_1 is the fraction of allocated slots that are utilized by the node to serve flows other than F_1 that are either originating in or being routed through nodes on the path of flow F_1. We call the total number of flows competing at node i the Traffic Factor, TF_i, of that node. For any flow, the maximum contributor to delay is the node along the path with the maximum TF_i, which we will just call TF here. Incorporating these considerations into a calculation for D_1, we achieve the following expression

$$D_1 = T_{slot} \cdot P_N \cdot CF \cdot TF \tag{1}$$

Figure 2(a) depicts the delay of D_1 in a simple case of only two flows, F_1 and F_2, being present, in which case $TF = 2$. In this example we use flows that consist of only 2 (F_{i-j} is packet j in flow i) packets to portray the delay. In most real applications, P_N will be much larger, making D_1 a good approximation of this delay component.

The second delay that exists is due to multi-hop propagation of packets. This delay is simply the time for a single packet to traverse the path length. Note that this delay is not necessarily just the path length multiplied by T_{slot}, because of possible queuing delays and/or ordering constraints. We show here how ordering constraints impact this TDMA network. A node cannot forward a packet from the flow until it receives that packet from the previous hop. In our line network example, when the direction of the flow matches the nodes' schedule of slots $1-2-3-1-2-3$, as in Figure 2(a), each successive node receives a packet on the time slot before it is scheduled, resulting in no extra delay. For a flow in the opposite direction, though, where nodes are scheduled $3-2-1-3-2-1$, as in Figure 2(b), the first slot 1 is not utilized, because the first node scheduled in time slot 1 has not yet received a packet in the flow. Every other slot is wasted, on average, for the initialization of the flow, resulting in approximately twice the delay. We will use a term that we call the *Delay Factor*, or DF, to account for this effect where it exists.

The multi-hop propagation delay, then, is modeled by

$$D_2 = T_{slot} \cdot DF \cdot (PL - 1) \tag{2}$$

where PL is the average path length.

We note several points about this delay factor. First, in a loaded network, the nodes can and will serve other flows while awaiting the arrival of packets in this flow of focus. That utilized bandwidth does not, however, preclude this DF impact on delay for the flow of interest, F_1. Any node cannot serve F_1 until a packet from that flow has been received. Second, this delay is only accounted for once per flow because all other packets are pipelined. All other packets' delay is captured by the end to end delay, D_1. This effect is best illustrated by examining the difference between D_2 in Figures 2(a) and 2(b). Here, we see the multi-hop propagation requires twice the number of slots because every other slot is unused in F_1's propagation.

To calculate a DF for an entire network, we can calculate a DF for each possible sample path and find the average of these values. For example, in the case of the line network with a 3-slot schedule, $DF = 1$ in the direction for Figure 2(a) and $DF = 2$ in the opposite direction shown in Figure 2(b), so the average DF used to approximate average delay in the network is $DF_{line-avg} = 1.5$.

By combining the two components of delay, we can give a relation for a network that will successfully achieve an average flow's data and timeliness requirements:

$$D_1 + D_2 \leq T$$

$$T_{slot} \cdot P_N \cdot CF \cdot TF + T_{slot} \cdot DF \cdot (PL - 1) \leq T$$

Recalling that the time of a slot is determined by the size of a packet, P_S, and available channel rate, W, in the relation $T_{slot} = P_S/W$, we can substitute to get

$$\frac{P_S}{W} \cdot P_N \cdot CF \cdot TF + \frac{P_S}{W} \cdot DF \cdot (PL - 1) \leq T$$

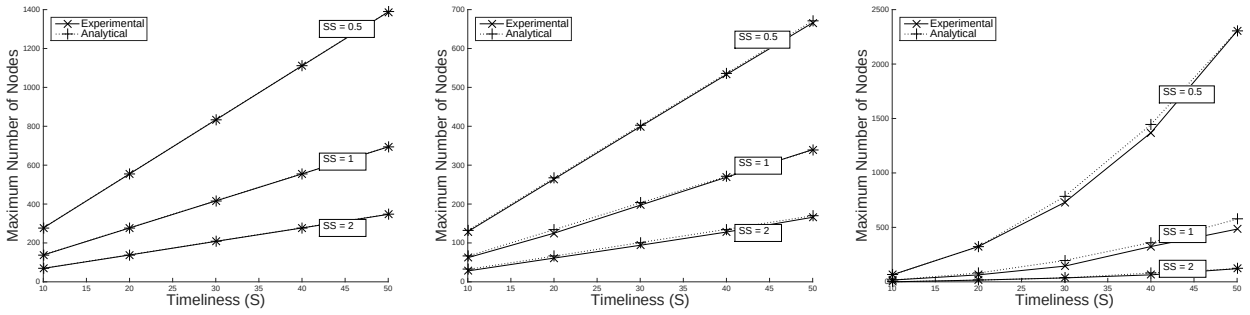
Finally, substituting the total number of bits required for a flow $P_S \cdot P_N = k_{req} \cdot I_S$ (where k_{req} is given by a function of required QoI), and rearranging this inequality, we can obtain a cleaner view of each parameter's impact on network limits:

$$W \cdot T - k_{req} \cdot I_S \cdot CF \cdot TF - P_S \cdot DF \cdot (PL - 1) \geq 0 \tag{3}$$

Every network will have its own set of parameters that can be substituted into this general formula, providing a tool to approximate limitations and sensitivity to changes in specific parameters. To show this usefulness concretely, we provide derived values for a TDMA-based wireless network with several basic topologies.

B. Example of Applying Framework

We illustrate the application of network signatures to the relationship in (3) using an N-node TDMA network with three different topologies: clique, line, and grid, also known as a "Manhattan grid." (Discussion of other network control protocols and topologies are addressed in Section VIII.) We adopt a traffic model that uses Top-K queries as an example application. We assume that all nodes have a set of collected images that are used to respond to Top-K queries. Each node produces a query with a target image and target QoI, $\mathbf{q} = \{C, T\}$, describing the required completeness (here, we



(a) Clique Network, $I_S = 9MB$ (b) Line Network, $I_S = 12MB$ (c) Grid Network, $I_S = 48MB$

Fig. 3. Empirical results match analytical results closely for all performed tests. Results for each topology and a variety of sum similarity (SS) and timeliness requirements are provided.

use sum similarity) and timeliness, and sends it to another node chosen at random. The queried node will respond with the number of images, k_{req}, required to achieve the target sum similarity. Values for k_{req} are taken from the empirical relation in Figure 1(a)[1].

	CF	TF	DF	PL
Clique	N-1	1	1	1
Line	3	$\frac{(N-1)^2}{2(N-2)}$	1.5	$\frac{N}{4}$
Grid	5	$\sqrt{N}$	2.5	$\frac{2}{3}\sqrt{N}$

TABLE I
CF, TF, DF, AND PL VALUES FOR EXAMPLE TOPOLOGIES

Since our goal is to determine the point at which an average flow is no longer sustainable, we derive and use average values for TF, CF, DF, and PL for the network. In the case of TF, we use the value for the node with the largest expected TF_i since flows that are routed through this node are expected to experience that largest delay and are likely to be the first that fail to meet their timeliness requirements. Values for this example are shown in Table I, and a derivation of TF for a grid network is included in Appendix A. Details about deriving the other values are explained in detail in [1]. The following equations can be used to determine QoI and network size limitations, which will be exemplified in the next two sections:

Clique:

$$W \cdot T - I_S \cdot k_{req} \cdot (N-1) \geq 0 \quad (4)$$

Line:

$$W \cdot T - 3 \cdot I_S \cdot k_{req} \cdot \frac{(N-1)^2}{N-2} - 1.5 \cdot P_S \cdot (\frac{N}{4} - 1) \geq 0 \quad (5)$$

Grid:

$$W \cdot T - 5 \cdot I_S \cdot k_{req} \cdot \sqrt{N} - 2.5 \cdot P_S \cdot (\frac{2}{3}\sqrt{N} - 1) \geq 0 \quad (6)$$

V. VALIDATION

To show how effective estimates using this framework can be, we simulated the network topologies and traffic described above in Section IV-B in the ns3 network simulator, comparing empirical results to those generated analytically. We present a subset of these comparisons to provide evidence of the effectiveness of the methodology. All results generated, however, exhibit very similar trends of proximity between empirical and the analytical values.

We use a channel rate of $W = 2Mbps$, packet sizes of $P_s = 1500$ bytes, and image sizes of 9, 12, and 48 Mbytes. As above, the correlation between Sum Similarity and k_{req} is taken from the actual observed relation in Figure 1(a). All values of parameters (SS, T, I_S, etc.) were chosen to test a variety of network sizes and QoI requirements while remaining within realistic network sizes, both with respect to real-world deployments and simulations with feasible runtimes.

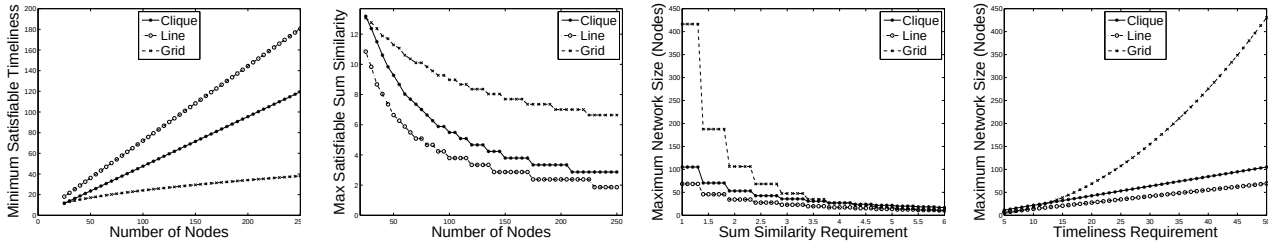
Figure 3 shows the maximum scalability projected by solving inequalities (4)-(6) and the maximum scalability observed in our experiments. In our simulations, we defined *scalable* as a network in which all of the requested queries were entirely fulfilled within the specified timeliness requirement. Small differences arise due to average values being used for CF, TF, DF, and PL, but, as the graphs show, in all of these scenarios, our network size estimates are very close to those realized in practical simulations.

VI. IMPACT ON NETWORK DESIGN

Now that we have established a model for QoI satisfiability and scalability and shown its accuracy in predicting estimated limits, we show how it can also provide quick, but useful intuition about the impact of various network parameters. In this section, we approximate where appropriate like in (8)-(9) below since the insights we seek here are more easily gained with simpler expressions, and are not greatly affected for large values. In applying the framework in practice, though, numerical solvers can be used, removing the need for approximation.

Throughout this section, channel rate and packet size values are again fixed at $W = 2$ Mbps and $P_S = 1,500$ Bytes. Also, we again choose sum similarity as the illustrative metric of completeness, but any other definition of completeness could be swapped in here. Specific values of SS, T, and N are varied to show various effects, so their values are explicitly listed in each figure and/or description.

[1]This application is not necessarily intended to model a known operational scenario, only a generic example to illustrate our model in a simple manner.



(a) Max Timeliness vs. Network Size (Sum Sim. = 5.0) (b) Max Sum Sim. vs. Network Size (Timeliness = 50) (c) Max Network Size vs. Sum Sim. (Timeliness = 10) (d) Max Network Size vs. Timeliness (Sum Sim. = 5.0)

Fig. 4. Varying design parameters provide immediate limitations as well as evident trends and comparisons.

1) Timeliness: Since $\mathbf{q}$ is a composite requirement of both timeliness and sum similarity, the value representing completeness in this application, we can explore the limits and impact of network parameters on each separately. First, we will substitute appropriate values for each topology, and then solve the inequality for T. The result for each is as follows:

Clique:

$$T \geq \frac{B}{W} \cdot (N-1) + \frac{P_S}{W} \tag{7}$$

Line:

$$T \geq 1.5 \cdot \frac{B}{W} \cdot \frac{(N-1)^2}{N-2} + 1.5 \cdot \frac{P_S}{W} \cdot (\frac{N}{4} - 1) \tag{8}$$

$$\approx 1.5 \cdot \frac{B}{W} \cdot (N-1) + 1.5 \cdot \frac{P_S}{W} \cdot (\frac{N}{4} - 1) \tag{9}$$

Grid:

$$T \geq 5 \cdot \frac{\sqrt{N}}{W} \cdot (B + \frac{P_S}{3}) \tag{10}$$

Using these inequalities, not only can network designers gain estimates of maximum timeliness constraints satisfiable in various network scenarios, but they can also quickly compare the dependency on specific parameters. By quick examination of these relationships, one can see that timeliness has a $O(N)$ relationship with respect to network size in clique and line topologies, but $O(\sqrt{N})$ in a grid topology. Figure 4(a) reinforces this lesson. The graph shows minimum T values for increasing values of N with all other values fixed, showing that a grid network can serve tighter timeliness constraints than other topologies in larger networks.

While it may have been intuitive that a grid network should be able to serve flows with lower timeliness constraints than a line network since it will have shorter average path lengths, what is not intuitive is *how much* lower of a timeliness constraint is satisfiable. For example, we see here that a grid network can scale up to over 250 nodes while supporting flows with $T = 40$. A line network, on the other hand, can only scale to only approximately 20% of that network size for the same timeliness value. In either case, it is intuitive to know how a clique network's timeliness satisfiability should respond to either a grid or line network since its operation is quite different. With the inequalities in (7)-(10), though, it is quickly and easily understood.

2) Sum Similarity (Completeness): In the exact same manner, we can evaluate maximum sum similarity values, SS, achievable by a network. To do so, we define the inverse of the function $Q(SS) = k_{req}$ defined above. Here, $Q'(k_{req}) = SS$ provides the sum similarity corresponding to the number of images served in each flow. The relationship for sum similarity for each topology is described with the following relations:

Clique:

$$SS \leq Q'(\frac{W \cdot T - P_S}{I_s \cdot (N-1)}) \tag{11}$$

Line:

$$SS \leq Q'(\frac{W \cdot T - 1.5 \cdot P_S \cdot (\frac{N}{4} - 1)}{3 \cdot I_s \cdot \frac{(N-1)^2}{2(N-2)}}) \tag{12}$$

Grid:

$$SS \leq Q'(\frac{W \cdot T - 2.5 \cdot P_S \cdot (\frac{2}{3} \cdot \sqrt{N} - 1)}{5 \cdot I_s \cdot \sqrt{N}}) \tag{13}$$

Figure 4(b) visualizes these sum similarity limits for network instances of different numbers of nodes while timeliness is fixed. Here, again, limits and trends are quickly evident. In addition, the nonlinearity of completeness requirements are manifested here as the data requirements for the highest completeness requirements are only sustainable for small networks, but as the network size increases, the impact on achievable completeness is not as dramatic.

To highlight this example, we can refer back to Figure 1(b) and see that in order to obtain 7 images matching the target image on average, we should collect 28 images, which we see from 1(a) equates to a sum similarity of approximately 6.5. If we require a timeliness of 50 seconds and our network topology has 50 nodes that are in a line, a sum similarity of 6.5 is the maximum completeness the network can support. If these same nodes are arranged in a grid network, however, we see that the network still has the capability to support more traffic, closer to a sum similarity of 11.5, which is twice as many images on average.

3) Scalability: Finally, given a topology and application with predetermined requirements of $\mathbf{q}$, we may be simply interested in how large our network can grow before its capacity to deliver the desired QoI is no longer possible. The proper relationships to answer this question are below in (14)-(16), and Figure 4(c) gives maximum scalability

examples when either component of $\mathbf{q}$ is varied. Once again, the graphs show the major impact completeness and timeliness requirements can have on network size.

Clique:

$$N \leq \frac{W \cdot T - P_S}{B} - 1 \tag{14}$$

Line:

$$N \leq \frac{W \cdot T + \frac{3}{2}P_S}{\frac{3}{2}B + \frac{3}{8}P_S} \tag{15}$$

Grid:

$$N \leq (\frac{W \cdot T}{5 \cdot B + \frac{5}{3} \cdot P_S})^2 \tag{16}$$

VII. Scalably Feasible QoI Regions

Now let us consider a special case where nodes collect a total of k_{req} images, but each image must come from a different node, perhaps because each node only has one image or maybe by design to provide increased credibility through corroboration, another contextual metric of interest in QoI-aware networks. This scenario allows us to illustrate an interesting observation.

Consider a set of QoI requirements that include completeness and timeliness, $\mathbf{q} = \{C, T\}$. As previously noted, a certain number of images are required to achieve this desired level of completeness, $Q(C) = k_{req}$. Unlike the model used in Section VI, however, each node can contribute only one image to k_{req}, which implies a minimum network size of $N \geq k_{req}$ in order to achieve the completeness outlined in the QoI requirements. On the other hand, applying the framework of Section IV to this network, we can determine the maximum network size, which we will call N_{max} here for distinction.

These two facts are important, because when $N_{max} < k_{req}$, then it is not possible to provide the QoI level $\mathbf{q}$. Hence, we say that this set of QoI requirements is infeasible, or *scalably infeasible*. This phenomenon defines the concept of a *Scalably Feasible QoI Region*, which refers to the region in which all sets of QoI pairs can be supported with the given network signature.

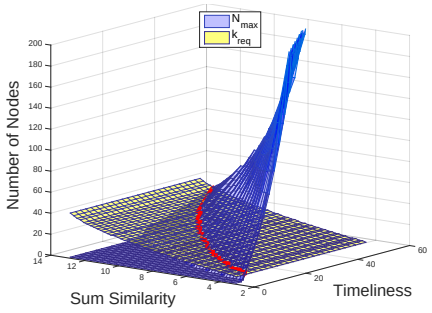


Fig. 5. The blue plane represents maximum scalability, and the yellow plane represents minimum required images. Therefore, all Sum Similarity, Timeliness pairs to the right of the red line are within the scalably feasible QoI region.

Figure 5 provides a visual representation of this region for a grid network that institutes the given traffic model assuming traffic is routed over randomly chosen shortest-path routes. Here, N_{max}, calculated for $\mathbf{q}$ pairs from $\{2.5, 1\}$ to $\{13.0, 50\}$, is shown in the graph by the blue surface. On the same graph is the number of images required, k_{req}, for each Sum Similarity requirement, shown with the yellow surface on the graph. The intersection of these two surfaces, displayed with a red line, provides the edge of the scalably feasible QoI region. In this example, all sum QoI pairs to the right of this line, i.e. the region where $N_{max} > k_{req}$, are scalably feasible.

In general, regardless of how many images a node has, it is possible to analyze the trade-off between different QoI attributes for a fixed value of maximum network size, N_{max}. Specifically, by fixing N in (4)-(6), one can obtain T and k_{req} (and, hence, C) resulting in the set of supportable $\{C, T\}$ pairs defining a feasible region for QoI. This region can be visualized by intersecting the blue scalability curve with a flat surface fixed at N_{max} instead of the yellow surface in Figure 5. The scenario given in this section goes one step further and also takes into account the network size actually required to generate/support a given QoI requirement (using the non-flat yellow surface derived from experimental results in Figure 5).

VIII. Discussion

We note that although TDMA is primarily used in this work, the same approach can be taken to derive QoI-based relations in networks that use other MAC layer protocols. In these cases, the appropriate Delay Factor would need to be derived for each protocol. To examine an 802.11 network, for example, the DF would capture queuing delays and could be determined by extending a delay model such as in [19], for example.

Similarly, although we only address the regular topologies of clique, line, and grid networks here, we believe the framework can be applied to more complex, irregular network topologies. In fact, some of the simple models presented here may already be quite useful for some of the topologies addressed. As shown in [1], for example, a dense random network may be closely approximated by a clique network, or random networks' capabilities can be approximately bounded by the limits of clique and grid networks, since these networks can be viewed of as examples of dense and sparse random networks, respectively.

Another approach to more complex topologies is to extract expressions for DF, TF, etc. empirically from simple simulations when deriving closed-form expressions is impossible or infeasible. This approach would provide the benefits of the framework presented here without being as complicated as a packet-based simulation or testbed that must implement all layers of the network. We are currently pursuing realistic examples and validation of this concept for more complex networks, including random and social network topologies among others.

IX. Conclusion

This work provides several contributions to the field of QoI-aware wireless networks. First, we motivated the use of completeness and timeliness as QoI attributes, providing

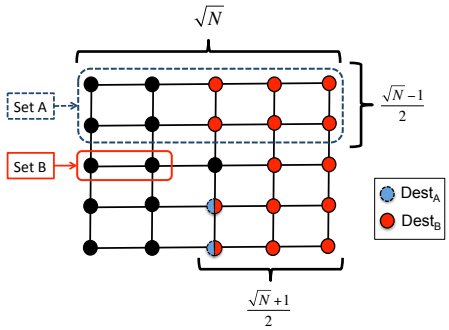


Fig. 6. Sources and destinations used in proving TF for grid networks

example applications and several different ways to measure completeness in these applications. Next, we developed a framework that can be used to predict QoI and network size limits for a specific network and then validated the framework's accuracy by comparing analytical results with simulations performed in the ns3 network simulator. Examples of the impact of different network parameters were shown, providing concrete examples of the framework's usefulness in real-world applications. In addition, the concept of scalably feasible QoI regions was introduced. In addition to future work already mentioned in Section VIII, we plan to expand this framework to include consideration of more complex network control actions, such as caching and/or data compression or fusion, which are all of interest in QoI-aware networking.

Appendix A
Proof of Traffic Factor for Grid Network

We outline a simple proof for determining the traffic factor of the center node in a grid topology of N nodes using "Row-First, Column-Second" routing, not including traffic originating or ending at the center node. For simplicity, we only go through the proof for when $\sqrt{N}$ is odd. The construction for even values of $\sqrt{N}$ follows from the same logic.

Assume that each node is the source of exactly one flow at all times and that the destination of this flow is uniformly chosen from all other $N-1$ nodes in the network. Node i, then, has a $\frac{1}{N-2}$ chance of choosing each other node that is not the center of the grid. For each source node, we can determine the number of destinations that route through the center. We separate nodes into two categories for this counting.

The first set of nodes we consider are those circled in set A in Figure 6. Through manual inspection, one can deduce that the only destination nodes in the figure that result in a path that is relayed by the center node are the two bottom nodes in the center column in the figure, marked with blue. We define the probability of a node in set A choosing one of these destinations from all possible destinations as $P_A = \frac{\frac{\sqrt{N}-1}{2}}{N-2}$.

Now, we can count the total number of nodes for which this probability holds. From the figure, we can quantify the number of circled nodes, but we must also consider the reverse, i.e. imagine the figure rotated vertically, so the total number of nodes falling into set A is actually $N_A = \sqrt{N} \cdot (\sqrt{N} - 1)$. Then, the expected number of paths being forwarded by the center node at any given time by nodes in set A is simply the product of P_A and N_A:

$$E[TF_A] = \frac{\frac{\sqrt{N}-1}{2}}{N-2} \cdot \sqrt{N} \cdot (\sqrt{N} - 1) \tag{17}$$

Next, we consider the nodes not in set A. These nodes are in the same row as the center node, and we will call them set B, also shown in Figure 6. Here, all destinations on the "opposite" side of the center as well as those in the same column of the center require being routed through the center node when originating from any nodes in set B. Just as above, we can relate the probability of choosing one of these destinations as $P_B = \frac{\frac{\sqrt{N}+1}{2} \cdot \sqrt{N} - 1}{N-2}$.

Using the number of nodes in set B, $N_B = \sqrt{N}$, the resulting expected traffic factor for the center node attributed by nodes in set B is

$$E[TF_B] = \frac{\frac{\sqrt{N}+1}{2} \cdot \sqrt{N} - 1}{N-2} \cdot 2 \cdot (\frac{\sqrt{N}-1}{2}) \tag{18}$$

Since sets A and B account for all non-center nodes in the network, the overall expected traffic factor is just the sum of $E[TF_A]$ and $E[TF_B]$, which simplifies to

$$E[TF] = \frac{\sqrt{N}(N-2)+1}{N-2} \tag{19}$$

which is effectively $\sqrt{N}$ for large N.

References

[1] R. Ramanathan, Ciftcioglu E., A. Samanta, Urgaonkar R., and T.L. Porta. An approximate model for analyzing real-world wireless network scalability. Technical report, Raytheon BBN Technologies, 2015. http://www.ir.bbn.com/~ramanath/pdf/symptotics-techreport.pdf.

[2] R. Ramanathan, A. Samanta, and T. La Porta. Symptotics: A framework for analyzing the scalability of real-world wireless networks. In *Proceedings of the 9th ACM symposium on PE-WASUN*, pages 31–38. ACM, 2012.

[3] E.N. Ciftcioglu, R. Ramanathan, and T.F. La Porta. Scalability analysis of tactical mobility patterns. In *MILCOM 2013*, pages 1888–1893. IEEE, 2013.

[4] J. Li, C. Blake, D. SJ De Couto, Hu Imm Lee, and R. Morris. Capacity of ad hoc wireless networks. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 61–69. ACM, 2001.

[5] P. Gupta and P.R. Kumar. The capacity of wireless networks. *Information Theory, IEEE Transactions on*, 46(2):388–404, 2000.

[6] A. I Khuri and S. Mukhopadhyay. Response surface methodology. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(2):128–149, 2010.

[7] S. Rager, E. Ciftcioglu, T. F. La Porta, A. Leung, W. Dron, R. Ramanathan, and J. Hancock. Data selection for maximum coverage for sensor networks with cost constraints. In *DCOSS*, Marina Del Rey, CA, May 2014.

[8] H. Tan, M. Chan, W. Xiao, P. Kong, and C. Tham. Information quality aware routing in event-driven sensor networks. In *IEEE INFOCOM, 2010*, pages 1–9, 2010.

[9] Z. M. Charbiwala, S. Zahedi Y. Kim, Y.H. Cho, and M. B. Srivastava. Toward Quality of Information Aware Rate Control forSensor Networks. In *Fourth International Workshop on Feedback Control Implemenation and Design in Computing Systems and Networks*, April 2009.

[10] E.N. Ciftcioglu, A. Yener, and M.J. Neely. Maximizing quality of information from multiple sensor devices: The exploration vs exploitation tradeoff. *Selected Topics in Signal Processing, IEEE Journal of*, 7(5):883–894, Oct 2013.

[11] J. Edwards, A. Bahjat, Y. Jiang, T. Cook, and T.F. La Porta. Quality of information-aware mobile applications. *Pervasive and Mobile Computing*, 11:216–228, 2014.

[12] E. N. Ciftcioglu, A. Michaloliakos, A. Yener, K. Psounis, T. F. La Porta, and R. Govindan. Operational information content sum capacity: From theory to practice. *Computer Networks*, 75:1–17, 2014.

[13] M.Y.S. Uddin, H. Wang, F. Saremi, G.-J. Qi, T. Abdelzaher, and T. Huang. Photonet: a similarity-aware picture delivery service for situation awareness. In *Real-Time Systems Symposium (RTSS), 2011 IEEE 32nd*, pages 317–326. IEEE, 2011.

[14] Y. Jiang, X. Xu, P. Terlecky, T. Abdelzaher, A. Bar-Noy, and R. Govindan. Mediascope: selective on-demand media retrieval from mobile devices. In *Proceedings of the 12th international conference on Information processing in sensor networks*, pages 289–300. ACM, 2013.

[15] E.N. Ciftcioglu and A. Yener. Quality-of-information aware transmission policies with time-varying links. In *2011 MILCOM*, pages 230 –235, nov. 2011.

[16] A. Bar-Noy, G. Cirincione, R. Govindan, S. Krishnamurthy, T.F. La-Porta, P. Mohapatra, M. Neely, and A. Yener. Quality-of-information aware networking for tactical military networks. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pages 2–7. IEEE, 2011.

[17] S.A. Chatzichristofis and Y.S. Boutalis. Cedd: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval. In *Computer Vision Systems*, pages 312–322. Springer, 2008.

[18] T.T. Tanimoto. An elementary mathematical theory of classification and prediction. *International Business Machines Corporation*, 1958.

[19] N. Gupta, P.R. Kumar, et al. A performance analysis of the 802.11 wireless lan medium access control. *Communications in Information & Systems*, 3(4):479–304, 2003.