Network Scalability under Quality of Information Requirements

***Abstract*—Practical network scalability, known as *symptotics*, has obvious applications in designing ad hoc networks. In any practical implementation, knowing the limitations on scalability, the capabilities of deliverable QoI, and the impact of QoI requirements are crucial to designing an operational, effective network. To obtain upper limits on network scalability and, more importantly, to understand how these limits are impacted by QoI functions and requirements, we apply QoI awareness to the symptotics framework. We use two similarity-based image retrieval algorithms to motivate and exemplify the relationship between timely QoI requirements and network scalability. Results show that high QoI and strict timeliness requirements can have a large impact on scalability, which gives a much clearer understanding of the tradeoffs present in network design. We also introduce and show examples of *scalably feasible QoI regions* for special cases, showing clear limitations on QoI requirements that are able to be satisfied by these networks.**

I. Introduction

Symptotic analysis is a relatively new framework for characterizing practical network scalability instead of using common asymptotic analysis. Introduced in [1], symptotics is extremely useful for applications and network designers that are interested in determining the limits of a specific network implementation as well as how various factors affect these limits in terms of scalability. For example, imagine designing an emergency ad hoc network to quickly replace destroyed infrastructure after a natural disaster or to exchange information without the use of government-controlled infrastructure in a time of political unrest. Quantifying the traffic limitations, what topology allows for the largest feasible network size, or how load balancing will impact capacity are all crucial to successfully designing the most effective network.

Understanding the effects of Quality of Information (QoI) requirements on network scalability is also extremely important, because as seen in many recent fields of study like [2]–[7], the actual benefits of a network in terms of data utility may vary greatly from the standard metrics often used to describe Quality of Service. As we will discuss later, timeliness of data collection is one such quality that is crucial in many applications, but is not captured without explicit consideration. For this reason, we aim to begin understanding the connections between QoI requirements and practical network scalability in this work.

Specifically, we consider the practical effects of real networks, including protocol overhead, contention, and traffic loads, in conjunction with QoI to build a framework that provides upper limits on network sizes for defined QoI requirements. Utilizing QoI in this framework is important because the relationship between throughput and QoI is often nonlinear, a notion that will be supported in Section III. We use timely, similarity-based image collection to provide motivation and concrete applications with which we test our methodology and provide results. The model, though, is designed to be general so that any relationship between data rates and QoI can be inserted for scalability analysis.

The results in Section V show several key insights. First we show the maximum sizes to which the networks can scale for different requested levels of QoI. The QoI depends on timeliness and one of two metrics considered: completeness or diversity. These attributes are defined by the sum similarity of collected images resulting from Top-K queries for completeness, and sum dissimilarity of images collected using the greedy spanner algorithm. These definitions will be explained in more detail in Section III. With these results, we show that the network scalability considerably reduces with higher completeness and diversity requests, as well as with stringent timeliness requirements.

Finally, we identify the trade-off of a given QoI requirement resulting in both a minimum required network size to provide a required number of images and the maximum feasible network size able to support that amount of necessary traffic. We identify the region of QoI requests where the former does not exceed the latter, and, hence, the QoI request can be satisfied.

II. Related Work

We adopt the symptotic scalability framework from [1], which has been previously applied to content-agnostic static networks [8] and mobile networks [9]. Other works that characterize the capacity of wireless networks, like [10]–[12], do so differently by considering how networks scale asymptotically or by analyzing specific network instances instead of developing a general model for scalability.

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 [4], [13], routing [14], and scheduling/rate control [6], [15]–[17]. It is also the focus of a credibility-aware optimization technique in [18].

The work in [5] 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 in [5].

Additionally, [2] and [19] outline a framework called Operational Information Content Capacity, which describes the obtainable region of QoI, a notion similar to the *scalably feasible QoI region* in Section IV. 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.

Similarity-based image collection has previously been considered [20] and [21]. In [20], authors consider a DTN network where the objective is to collect the most diverse set of pictures at every node. Authors consider a picture prioritization and dropping mechanism in order to maximize the diversity, defined by dissimilarities of the collection of pictures. However, it does not consider attributes of timeliness, nor the consideration of transmission rates and network topology. [21] considers a smartphone application where different queries called top-K, spanner, and K-means clustering are defined. Each of these queries are based on image similarity metrics, and our use of Top-K and Spanner algorithms here was inspired by this paper. While timeliness is considered as an objective in this work, the effects of rates and network topologies are overlooked.

III. QoI Model

The difference between Quality of Information and the traditional notion of network performance, called Quality of Service (QoS), is an important distinction to be made. While QoI has some connection to metrics like throughput, delay, jitter, packet loss, etc., its meaning goes beyond objective measurements. The true value of data in terms of QoI relies on its utility within the context of its use. A simple example of QoI is relating it to its importance in assisting a user with making or improving the confidence in a decision. Here, depending on a number of factors, such as timeliness, freshness, completeness, uniqueness, etc., a packet sent across the network may have very different impacts on the receiving end. For that reason, these various (often contextual) qualities of data are examples of the metrics used to determine the utility or QoI of a piece of data. To use these metrics in functional ways, applications keep vectors of the chosen attributes for data items. When a scalar value is required, one can be determined by inputting the vector into a designed QoI function that is weighted to prioritize specific attributes as the end user desires.

Even though QoI is highly contextual and customizable to a specific network and its goals, our goal here is to show how it can be effectively used in network design. To that end, we introduce several metrics here that provide real examples of how QoI can be measured quantitatively as described above. Specifically, we will adopt a notion of *completeness* to use as QoI within this work, showing how it can be defined in the contrasting situations of users having an underlying knowledge of the information available or needing to acquire the scope of available knowledge with some level of confidence.

We will also identify three example applications and accompanying algorithms that motivate using completeness as an effective network goal. We 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 scalability analysis of Section IV can be used with any feasible QoI requirements.

A. Image Selection Algorithms

As a motivating example, we choose an ad hoc network in which nodes generate photographs that are to be exchanged or collected at one or more data sinks. This example covers surveillance missions of military tactical networks or social applications for civilians, one example of which could be smartphone users contributing to an image-sharing application.

The first application in this setting that we introduce is as follows: Given the set of all photographs available in the network, we would like to return the set of k that exhibits the most diversity, ideally providing a user with a good sampling of images available in the network. This result is known as the **Spanner** of the set of known photographs. Such a result would be useful in a surveillance mission or in a social setting in which users would like a quick idea of the current state of a large area or event.

A similar approach to achieving this goal of discovering a complete view of the environment the network is sensing is to use the second algorithm we consider, which we simply call **Clustering**. Here, all images are separated into k clusters based on their pairwise distances using any version of a k-means clustering algorithm. Then, the most central photograph 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 environment in which the network is operating.

The final application we introduce occurs when one already has an 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 observes 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 of the building from another source may have a better view of the person in question that can be used for identification or more context. In a social situation, a user may want more images of a particular event of interest like a parade, a concert, or a sporting event. Called **Top-K**, the algorithm used for this application will choose the k images with the most similarity, or *smallest distance*, from the given image.

We note that while metadata associated with photographs may be useful in obtaining similar goals, content-based retrieval can sometimes be more effective, or at the very least can be used in addition to metadata to improve accomplishing the set goals. For instance, all of the images could be tagged with location and time stamps allowing an application to filter for desired values to get matching or spanning images sets. Even a location and time stamp will not account for the direction that the photographer is facing or objects in between the camera and a desired object or person of interest. Content-based

processing of these images, though, can be applied to the set of photographs existing after any location or time filtering to improve the Spanner, Clustering, and Top-K algorithm results on these image pools.

B. Image Similarity

To implement these algorithms, we again use the same choices for measuring the similarity of two images as was shown to be effective in [21]. This similarity is based on qualities inherent to a photograph like lightness, contrast, and color. While many techniques have been studied to compare photographs using these qualities, we choose an image-processing technique called Color and Edge Directivity Descriptor (CEDD) [22]. With CEDD, each image is described by a vector of 144 different features describing color and spatial color distribution.

We achieve a scalar representation of similarity between two images by calculating the *Tanimoto Similarity*, T_s, between their CEDD feature vectors, another practice commonly used in the image processing community [23]. The Taminoto similarity metric is defined as follows: Given two images with feature vectors $\mathbf{a}$ and $\mathbf{b}$,

$$T_s(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}.\mathbf{b}}{\mathbf{a}.\mathbf{a} + \mathbf{b}.\mathbf{b} - \mathbf{a}.\mathbf{b}}, \tag{1}$$

where $\mathbf{a}.\mathbf{b}$ is the inner product of two vectors. Proper normalization keeps this metric in the $[0, 1]$ range. Naturally, to describe the dissimilarity, or distance, of two photographs, then, we simply use $T_d = 1 - T_s$.

C. Measuring QoI

As already discussed, *Quality of Information* is a very contextual term, so defining metrics to provide objective measurements of it is a challenge within itself. Here, we provide methods for measuring completeness of a collected set of images with respect to the pool of all available images. First, we will explain how the similarity and dissimilarity for varying numbers of collected photographs, k, can be used to describe the QoI of each algorithm. Then, we will also show the resulting QoI with respect to specific scenarios with known ground truths to give real-world examples of how QoI differs dramatically from QoS.

1) Unknown Information Space: For the Spanner algorithm, we employ a greedy algorithm similar to that in [21] to simplify implementation and to define a *Sum Dissimilarity* metric. Here, the algorithm first chooses the two images with the greatest distance 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 QoI 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 effectiveness. Here, a higher level of dissimilarity is providing a more complete view of the feature space itself.

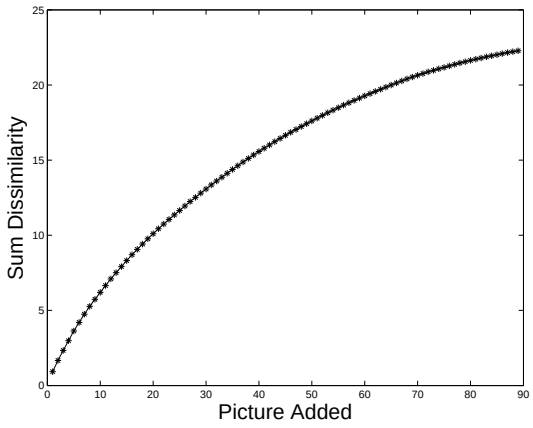


Fig. 1. Sum Dissimilarity for Spanners of Varying k

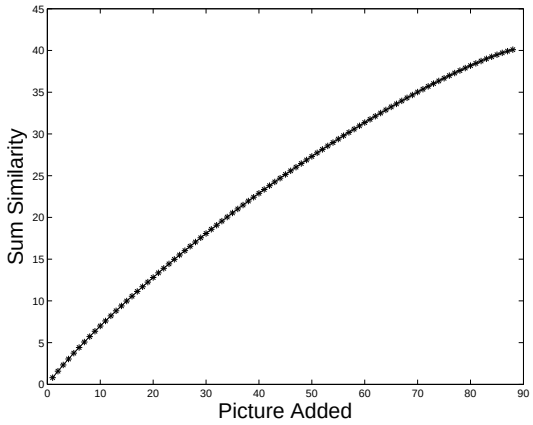


Fig. 2. Sum Similartiy for Top-K Results of Varying k

Conversely, the Top-k algorithm's goal is to provide images with features that are similar to the target image. Therefore, the Tanimoto Similarity of each successive image gives its effectiveness in providing a more complete view of the object or scene of interest. Naturally, then, the *Sum Similarity* of each successive image returned by the Top-k algorithm is a measure of the completeness being achieved.

Figures 2 and 1 show the average sum dissimilarity and similarity of images returned by the Spanner and Top-k algorithms, respectively, run on a set of images collected over the Penn State campus. These figures exhibit the diminishing returns of using similarity and dissimilarity metrics. This effect is important also because it visually shows how Quality of Information differs from throughput. As seen in these graphs, transmission of successive images is not linear in terms of gained completeness. Inversely, this relationship shows that obtaining a certain value of QoI or completeness may require a different number of images depending on the set available and their similarities. Specifically, we can denote the number of images required to achieve a level of completeness, S, as $N(S)$. This relationship will be useful later in determining feasible scalability.

2) Known Information Space: For our approach to measuring QoI, we can use a model in which each image belongs to one of n sets, Q_n, which can each represent a particular setting of interest. 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. In the case of a spanner query, the goal is to return images from different sets.

Using these two naturally occurring goals, we can measure the effectiveness of the algorithms, providing QoI values. For Top-K, the QoI value is the number of photographs returned that are in the same set as the target image. For the spanner algorithm, the QoI value can either be the number of sets covered by at least one of the returned images, or it can be the likelihood that all n sets will be covered by the returned images, as long as $k \geq n$.

To provide example values of these QoI metrics, experiments were run on photographs taken at 5 different settings around the Penn State campus. Each of the settings is of a pictorially different setting, e.g. a particular building, a downtown street, or a lawn setting, and over 20 images of each was taken. Then, for individual trials, sets of Q_n with 10 images in each set were randomly selected from this group of photographs and the Top-K and spanner algorithms were run over these 50 images with the target image being randomly selected in the case of Top-K.

Figures 4-3 show the average results of 1000 trials. In Figure 4, 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 usually requires collecting over twice that number of pictures.

This diminishing return is also evident in the spanner algorithm results. Figure 3 shows the number of sets represented by the algorithm output for increasing k. Here, if the goal is to achieve at least one image from each of the different settings as might be in a surveillance application, the spanner achieves it on average at $k \approx 17$. The same trend is evident in Figure **??** where the probability of covering all sets is plotted against k. Using this metric, the application can use the probability of achieving full coverage as the QoI metric. From this example application, if collecting at least one image from each set with 90% probability of success is acceptable, then only $k = 13$ images are necessary.

Comparing Figures 2-1 with Figures 4-**??**, we observe that completeness and diversity provide a sound indication of the number of sets covered for different numbers of images collected, encouraging our choice of focusing them as QoI attributes.

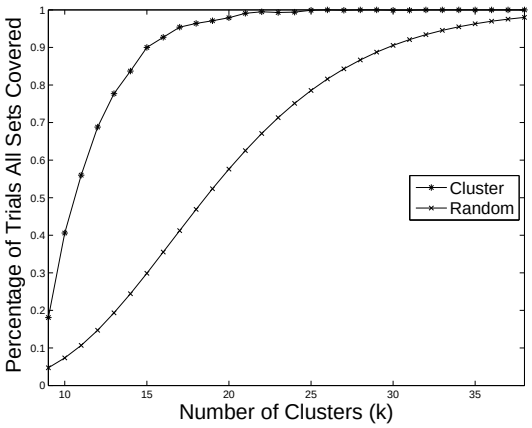


Fig. 3. Average Number of Sets Covered by Returned Images

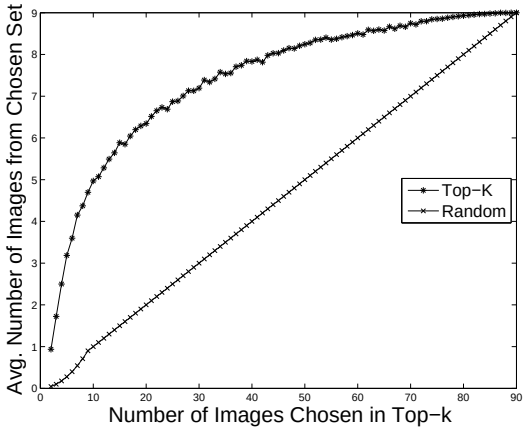


Fig. 4. Average Number of Images Selected from Same Set as Target Image

IV. QoI Scalability

As discussed in the previous sections, QoI is typically a highly non-linear function of the number of packets delivered at the destination. Given this behavior, simply delivering the highest possible rate is not necessarily the best option from a user QoI 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 instead desire to satisfy *QoI requirements*. In other words, given a certain QoI (e.g. completeness/diversity-timeliness pair) that is desired by the user of a network, what is the the number of nodes that the network can scale to? *And how sensitive is the scalability to the QoI that is desired?*

We investigate this question in the context of multihop wireless networks (such as mesh and sensor networks). In [1] an approximate upper bound on the 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:

$$R(m) = W(m) - \sum_j (1 + \gamma_j) D^j(m), \qquad (2)$$

where $R(m)$ is the *residual capacity* at a given node m, indicating the capacity remaining at a node after taking into account the load from all traffic sources from all nodes (Fig. 5). This is the difference between the *available capacity* $W(m)$

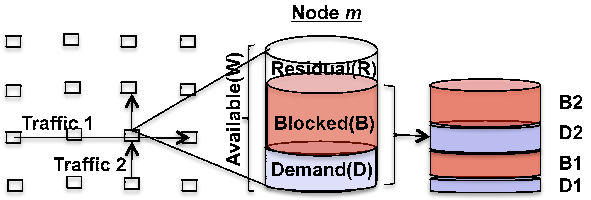


Fig. 5. Residual Capacities, Mesh Scenario

and the capacity $D^j(m)$ demanded by each source j (control overhead is regarded as one kind of source), where γ_j in the equation is the *contention factor*, which 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 [1] for details on the contention factor and derivation of Equation (2), and focus here on adapting the formulation to accommodate QoI.

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[1] 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 (2). 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

$$D^j = \xi(QRF_j(u_j)), \tag{3}$$

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

Combining with Equation (2), we have:

$$R(m) = W(m) - \sum_j (1+\gamma_j(m))\xi(QRF_j(u_j)(m)). \tag{4}$$

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.

For instance, given a Top-K query with QoI demand $\mathbf{q} = (C, T)$, we first determine the number K_{req} to provide completeness C from Figure 2. This results in a load in bits of $K_{req} * B$ where B is the nominal image size. Next, we obtain the required rate, which is $r_{req} \geq \frac{K_{req}B}{T}$. The QRF function for the spanner is also found similarly, where QoI demand $\mathbf{q} = (D, T)$ relates the requested diversity D to the number of images, again also defining the traffic requirement when timeliness requirements are considered.

We illustrate the relationship between QoI and scalability using a specific example. Consider a regular mesh network (also known as a "Manhattan grid") of N nodes. A continuous stream of traffic sent from each node to another node chosen uniformly at random.[2] Suppose further that we run the queries and request a specific QoI level $\mathbf{q}$. How many nodes can the network support (i.e., what is the upper bound on N) as a function of $\mathbf{q}$?

To determine this, we apply Equation (4) to the scenario, with $QRF(q)$ determined using the number of pictures required for the first QoI attribute, image size and timeliness.

Since the source and destination are chosen randomly, the scope of a flow is the average path length. In [24], 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 approximately $\xi(x) = \frac{2}{3}\cdot\sqrt{N}\cdot x$.

Finally, the contention factor for unicast traffic in a mesh network is $\gamma_1 = 7$ since the receiver plus three neighbors of each of sender and receiver have to defer on this transmission. Since we consider a stationary network, and are only looking for an approximate upper bound on the scalability, we ignore the routing overhead and assume the overhead due to MAC control messages is negligible.

We can then substitute these expressions into Equation (4) and note that the maximum scalability occurs when *R(m) = 0* for the node m at which $R(m)$ reaches zero first, also called the *bottleneck node*. Assuming this node m and dropping the per-node notation on R and W, resulting in the maximum node capacity of $W = (1+7)\cdot\frac{2}{3}\cdot\sqrt{N}\cdot QRF(q)$, which simplifies to a maximum network size of

$$N = [\frac{3\cdot W}{16\cdot QRF(q)}]^2. \tag{5}$$

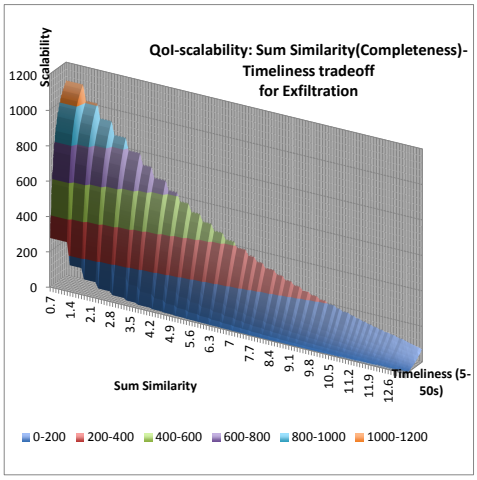
We note that given the coarseness of the model and the abstraction of many details, the above is by no means intended to be an accurate predictor of N in a real network. However, since the main intent is to study how the scalability *changes* with respect to QoI (rather than focus on absolute values), such an approximate upper bound suffices.

V. Results

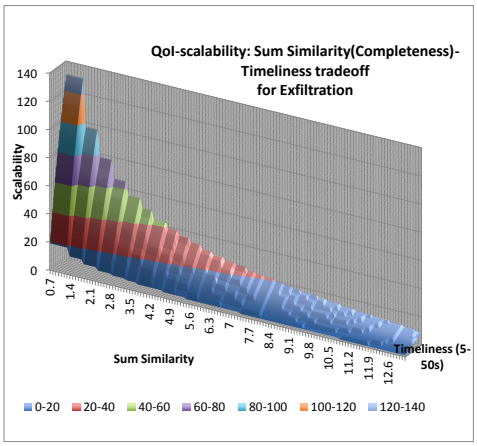
We consider a large sensor network where the sensors can do relaying but the network itself is unreachable. To exfiltrate the data, an Unmanned Aerial Vehicle (UAV) flies by over the field. The UAV (running the MediaScope MSCloud [21]) carries the queries, gets the feature vectors from the sensors, and determines which sensed data to upload. In order to avoid detection and being power constrained, the UAV can only hang around for a very limited time. The UAV arbitrarily roams over the sensor network, and periodically extracts information from one of the sensors, however the network does not have any prior knowledge on its trace, and which node it will request the information. Hence, from the network's perspective, the UAV can gather information from any of the nodes randomly. Without loss of generality, we assume the topology of the network is a mesh. For the unicast scenario, each node transmits its images to a destination node

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

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

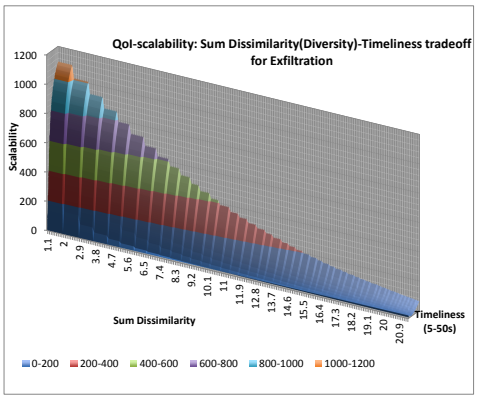


(a) Unicast Traffic

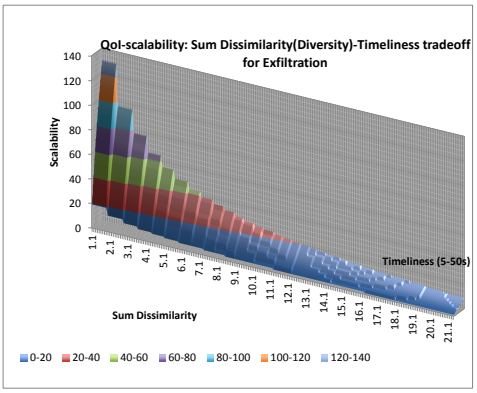


(b) Flooding

Fig. 6. Top-K: Sum Similarity vs. Scalability vs. Timeliness



(a) Unicast Traffic



(b) Flooding

Fig. 7. Spanner: Sum Dissimilarity vs. Scalability vs. Timeliness

randomly. The QoI of the information extracted depends on timeliness and completeness if a top-K query was issued, and completeness if a spanner query was issued instead.

We consider $W = 2Mbps$, both unicast and flooding traffic, TDMA as the medium access control protocol, and an image size of 2 Mbytes. In Figures 6(a) to 7(b), we demonstrate network scalability as a function of QoI requirements for different traffic properties in a mesh setting.

Clearly, there is a remarkable difference in the scalability depending upon the set QoI requirements. The fact that QoI makes a difference is not surprising, but the *magnitude* of the impact 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. For instance, it can potentially indicate when it makes sense to reduce QoI a bit and possibly gain significantly in scalability (e.g. from QoI=(10,5) to QoI=(10,10) in Figure 6(a)) and when such reductions will only give a marginal increase in scalability (e.g. from QoI=(3,40) to QoI=(3,45) in Figure 6(a)).

A. *Scalably Feasible QoI Regions*

Let us consider the special case where each node has at most one image. Note that even for such a setting, from fig. 4-5 the scalability of particularly flooding might be low. This leads to the following observation: To achieve a certain level of desired QoI q, which can be defined as (C, T) for Top-K queries and (D, T) for spanner queries, the completeness/diversity attribute necessitates a number $K_{req}(q)$ images to be collected. When each node can contribute with at most one picture, this implies a minimum network size of $K_{req}(q)$ that is necessary for the QoI level. On the other hand, the same QoI pair also results in a maximum network size $S(q)$ from the scalability framework. When $S(q) < K_{req}(q)$, it is not possible to provide QoI level q. Hence, we state that the QoI level q is infeasible, or *scalably infeasible*.

This phenomenon defines the concept of *scalably feasible QoI regions*, which define the set of QoI pairs that can be supported, given a given traffic structure. This region is given by a set of (completeness, timeliness) pairs for Top-K, and (diversity, timeliness) pairs for spanner queries. We demonstrate the scalably-feasible QoI regions in Figure 8-fig:spanScalR for flooding traffic.

While with the parameters in Fig. 4-5, no such problem is

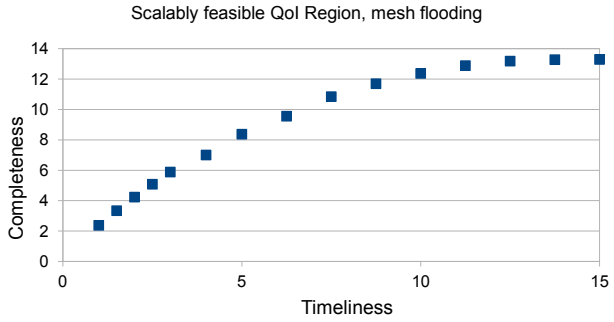


Fig. 8. Feasible Scalability Region of Spanner Algorithm, Flooding traffic

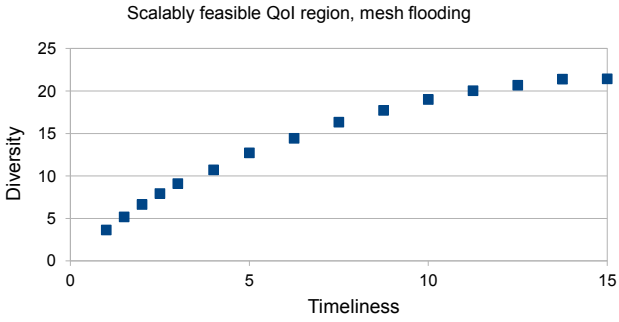


Fig. 9. Feasible Scalability Region of Spanner Algorithm, flooding traffic

observed with unicast traffic unless timeliness is very low, for cases where the available bandwith is scarce, we have the scalability-wise feasability issue for unicast as well. We demonstrate results when the bandwidth is reduced to 500 Kbps for unicast in Figures 8-11.

For both scenarios, these regions clearly demonstrate the tradeoff between the completeness/diversity that can be obtained and the timeliness that can be tolerated when system resources are scarce.

VI. Conclusion

Wrap it up with the highlights/takeaways. Maybe also include future work.

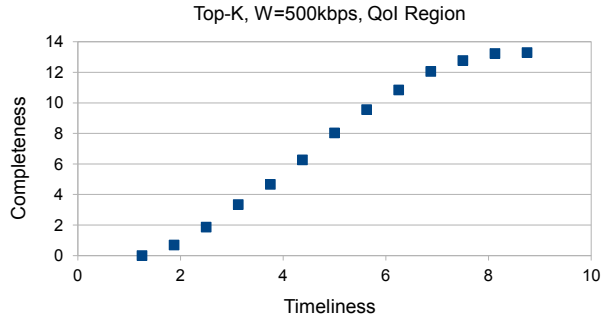


Fig. 10. Feasible Scalability Region of Top-K Algorithm, W=Unicast traffic, 500 Kbps

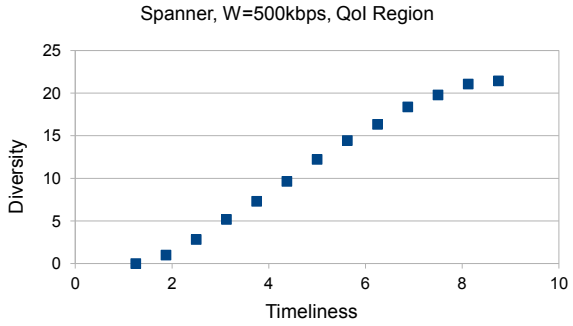


Fig. 11. Feasible Scalability Region of Spanner Algorithm, Unicast traffic, W=500 Kbps

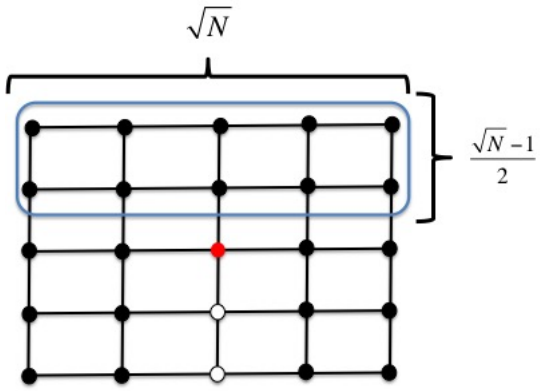


Fig. 12. Possible destinations for source nodes in set A

Appendix A
Proof of Transit Factor for Manhattan Grid Network

We outline a simple proof for determining the transit factor of the center node in a Manhattan grid topology of N nodes using "Row-First, Column-Second" routing, not including traffic originating or ending at the center node.

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 Figure 12. We will call these nodes set A. 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 white-colored nodes. We define the probability of a node in set A choosing one of these destinations from all possible destinations as

$$P_A = \frac{\frac{\sqrt{N}-1}{2}}{N-2} \tag{6}$$

Now, we can count the total number of nodes for which this probability holds. From the figure, we can quantify the number

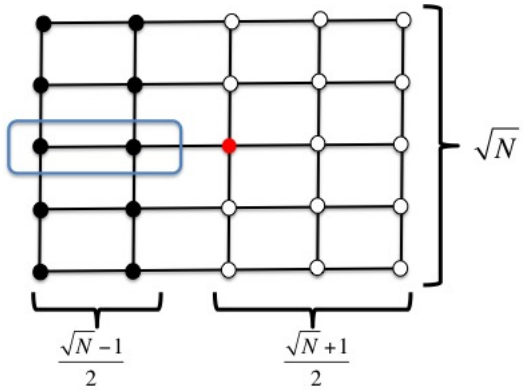


Fig. 13. Possible destinations for source nodes in set B

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

$$N_A = \sqrt{N} * (\sqrt{N} - 1) \tag{7}$$

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:

$$E[TF_A] = \frac{\frac{\sqrt{N}-1}{2}}{N-2} * \sqrt{N} * (\sqrt{N} - 1) \tag{8}$$

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, shown as the circled nodes in Figure 13. 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 and count the number of nodes in set B:

$$P_B = \frac{\frac{\sqrt{N}+1}{2} * m - 1}{N-2} \tag{9}$$

$$N_B = 2 * (\frac{\sqrt{N}-1}{2}) \tag{10}$$

The resulting expected transit factor for the center node attributed by nodes in set B is

$$E[TF_B] = \frac{\frac{\sqrt{N}+1}{2} * m - 1}{N-2} * 2 * (\frac{\sqrt{N}-1}{2}) \tag{11}$$

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

$$E[TF] = \frac{\sqrt{N}(N-2)+1}{N-2} \tag{12}$$

which is $O(\sqrt{N})$ for large N.

APPENDIX B
EXPLANATION OF RANDOM VARIABLES FOR MEASURING QOI

First, we explain the expected number of images that are from the same set as the target image in the Top-k algorithm. We define the following:

- n = total number of images (all sets)
- S = number of sets
- S_k = set of target image
- k = number of images collected
- N_S = number of images in each set (assumed to be the same here for simplicity)
- x = number of images returned $\in S_k$

If $k \leq N_S$, then

$$p(X = x|k) = \frac{\binom{k}{x} * \binom{n-k}{k-x}}{\binom{n}{k}}, \forall x \leq k \tag{13}$$

Otherwise, $p(X = x|k) = 0$. Comments: x must be less than k because one cannot choose more items from the target set than are chosen overall. The probability expression describes the possible combinations of choosing x from the target set and $k - x$ from the $n - N_{S_k}$ remaining images.

If $N_S < k < n - N_S$, then we consider the possible combination choosing x images from the target set and $k - x$ images from the remaining $n - N_S$ images, resulting in the following expression:

$p(X = x|k) = N_S choose x * n - N_S choose k - x / n choose k$

Finally, when $k > n - N_S$, then $k - (n - N_S + x)$ images must be from the target set by the pigeonhole principle, so the $p(X = x) = 0$ for all $k > n - N_S + x$. Otherwise, the same expression as directly above is true.

For cluster and spanner, we want to determine the probability that we will cover each of the S sets with at least one of the k chosen images if we had chosen them randomly. We will call X_i the random variable that represents the number of images from set i in the results. We use the following expression:

$$P(X_i > 0, \forall i) = (1 - P(X_i = 0))^S \tag{14}$$

where X_i is given by a multivariate hypergeometric distribution, which gives us the following:

$$P(X_i = 0) = \frac{\binom{n-N_s}{k}}{\binom{n}{k}} \tag{15}$$

REFERENCES

[1] 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 *MILITARY COMMUNICATIONS CONFERENCE, 2010-MILCOM 2010*, pages 493–498. IEEE, 2010.

[2] A. Bar-Noy, G. Cirincione, R. Govindan, S. Krishnamurthy, T.F. LaPorta, 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.

[3] S. Supittayapornpong and M. J. Neely. Quality of information maximization in two-hop wireless networks. In *Communications (ICC), 2012 IEEE International Conference on*, pages 4919 –4925, june 2012.

[4] F.H. Bijarbooneh, P. Flener, E. Ngai, and J. Pearson. Optimising quality of information in data collection for mobile sensor networks. In *Quality of Service (IWQoS), 2013 IEEE/ACM 21st International Symposium on*, pages 1–10, 2013.

[5] 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.

[6] 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.

[7] Rahul Urgaonkar, Ertugrul Necdet Ciftcioglu, Aylin Yener, and M. J. Neely. Quality of information aware scheduling in task processing networks. In *WiOpt*, pages 401–406. IEEE, 2011.

[8] 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 Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pages 31–38. ACM, 2012.

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

[10] 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.

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

[12] J. Jun and M.L. Sichitiu. The nominal capacity of wireless mesh networks. *Wireless Communications, IEEE*, 10(5):8–14, 2003.

[13] 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 *International Conference on Distributed Computing in Sensor Systems, DCOSS*, Marina Del Rey, CA, May 2014.

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

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

[16] 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.

[17] E. N. Ciftcioglu, A. Michaloliakos, A. Yener, K. Psounis, and T. F. La Porta. Power Allocation with Quality-of- Information Outages. In *Proc. IEEE WCNC*, Istanbul, Turkey, April 2014.

[18] B. Liu, P. Terlecky, A. Bar-Noy, R. Govindan, M. J. Neely, and D. Rawitz. Optimizing Information Credibility in Social Swarming Applications. *IEEE Transactions on Parallel and Distributed Systems*, 23:1147–1158, 2012.

[19] E. N. Ciftcioglu, A. Yener, R. Govindan, and K. Psounis. Operational Information Content Sum Capacity: Formulation and Examples. In *Proc. Fusion 2011, submitted*, Chicago, IL, July 2011.

[20] H. Wang, M. Uddin, G. Qi, T. Huang, T. Abdelzaher, and G. Cao. Photonet: A similarity-aware image delivery service for situation awareness. In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, pages 135 –136, april 2011.

[21] 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.

[22] 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.

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

[24] J.A. Silvester and L. Kleinrock. On the capacity of multihop slotted aloha networks with regular structure. *Communications, IEEE Transactions on*, 31(8):974–982, 1983.