QoI Symptotics

I. Introduction

Here we will introduce the topic (and possibly include related works?).

Similarity-based image collection has previously been considered [PN,MS]. In [PN], authors cosider a DTN network where the objective is to collect the most diverse set of ictures at every node. Authors consider a picure 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. [MS] has also considered a smartphone application where different queries are defined as top-K, spanner, and K-means clustering based on image similarity metrics. While timeliness is also considered as an objective, the effect of rates and network topology is overlooked.

QoI based scheduling has been considered from various angles. Start with event detection Bisdikian, Charbiwala etc, then OICC, Time varying queued, credibility-aware, QoI outage, Freshness-based for multiple sensors. Also DCOSS work (coverage).

In this work, we consider the practical effects of real networks as protocol overhead, contention, rates in conjunction with similarity based timely image collection. We adopt the symptotic scalabilty framework[], which has been previously applied to content-agnostic static networks[] and mobile networks[].

Our results first provides maximum sizes that the networks can scale, given requested levels of QoI. The QoI depends on timeliness and one of either completeness, or diversity. These attributes are defined by sum similarity of collected images resulting from Top-K queries for completeness, and sum dissimilarity of images collected using the greedy spanner algorithm. We observe that the network scalability considerably reduces with higher completeness and diversity requests, as well as stringent timeliness requirements.

Finally, we identify the trade-off such that a given QoI requirement results in both a minumum required network size, and a maximum feasible network size. We identify the region of QoI requests where the former does not exceed the latter, hence the QoI request can be satisfied.

II. Symptotics Framework

Here, we provide an overview of the network scalability framework.

III. QoI Examples

The utility of data that is extracted from a sensor or military tactical network is often highly dependent on the context of the data with respect to aspects like the network's goals and the other data being collected from the rest of the network. While this effect is often very qualitative by nature, we introduce metrics here that provide real examples of how QoI can be measured quantitatively. These measures will then be used as examples in the QoI-aware network scalability in Section IV.

As an example scenario, we choose a mobile network in which nodes generate photographs that are to exchanged or collected at one or more data sinks. This example covers surveillance missions of military tactical networks or civilian/social scenarios, one example of which could be smartphone users contributing to an image-sharing application.

Qualities like lightness, contrast, and color are all inherent to a photograph, and many techniques have been studied to compare photographs. We use one such technique called Color and Edge Directivity Descriptor (CEDD) [1]. Using the CEDD vector of each image, we can compare any two images. To achieve a scalar representation of similarity, from comparing two vectors, we use *Tanimoto similarity* [**?**], T_s. Naturally, to describe the dissimilarity, or distance, of two photographs, then, we simply use $T_d = 1 - T_s$.

A. Scenarios

Here, we will provide two example scenarios to motivate the algorithms used and results generated in this work. First, imagine a scenario in which you have one photo of an important event, like a criminal entering a building. If there are devices in the area also generating images, we would like to collect more images that can provide more information and context. For example, more pictures of the criminal entering the building might give a better idea of whether the criminal was accompanied by others or give a higher likelihood of finding the criminal's identity using facial recognition.

Even if we have location and time stamps of photos in the area, collecting all of the photographs taken in the correct place in a particular time period may be inefficient or even impossible depending on the network's capabilities. Additionally, even two cameras at almost the exact same location can take two photographs that are facing in opposite directions, leading to completely different content. Therefore, being able to find and acquire those photographs that have content most like the initial image will allow users to quickly obtain high quality information for the given context.

The second scenario would be one in which the goal is to provide surveillance of as much of an area as possible. In this case, photographs of different content are desired to create as complete of a view of the information in all available images.

B. Image Selection Algorithms

With the Tanimoto similarities and distances between all of the images, a chosen algorithm can be applied to select images to provide a desired level of quality of information. We present two such possibilities here along with examples and scenarios.

The first application of selecting images based on content occurs when a user would like to find more images that are similar to one already obtained. 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 user may have a better view of the person in question that can be used for identification or more context. Called *Top-k*, this algorithm choose the k images with the smallest distance from the given image.

The second application we introduce is called a *spanner*, and it is based on an opposing scenario. Instead of choosing matching images, the goal of a spanner is to select the k images that exhibit the most joint dissimilarity. This algorithm would be useful in a surveillance mission or other setting in which a user would like to get a snapshot of all areas of which photographs were taken that is as complete as possible. In order to choose images with the most dissimilarity, though, we must first define what that means. While the dissimilarity between two images is determined by the Tanimoto distance, creating a measure of the distance between more than two images requires consideration.

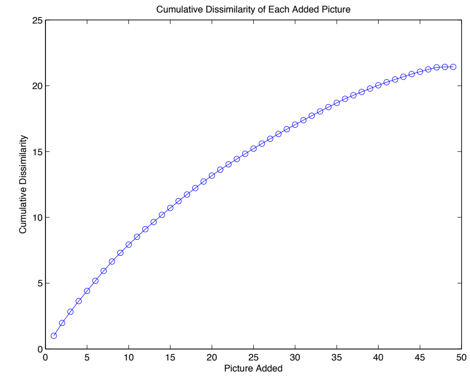

Fig. 1. Cumulative Distances for Spanners of Varying k - Regular

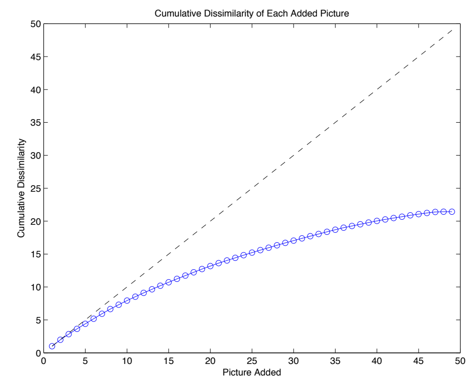

Fig. 2. Cumulative Distances for Spanners of Varying k - Square Axes

C. QoI vs. Throughput

With both the Top-k and Spanner algorithms, initial choices exhibit higher degrees of similarity and dissimilarity, respectively, that naturally decrease. Therefore, if we establish a measure of overall QoI being obtained as k is increased, we witness an effect of diminishing returns. For the Top-k algorithm, we define *Sum Similarity*, which is the sum of the Figures 1 and 2 show 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 these graphs clearly show, 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.

If necessary, here, we could also include the toy examples of which pictures are actually selected from the top-k and spanner algorithms, showing that they are useful tools.

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. *For example, an image recognition application will perform nearly the same for a range of compression ratios, suddenly experiencing a spike at some particular ratio.* Given this, simply delivering the highest possible rate is not necessarily the best option from a user 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 the maximum *QoI*. 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 [**?**] 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:

$$res = avail - \sum_j (1 + \gamma_j) D_j, \qquad (1)$$

where *res* is the *residual capacity* at a given node, indicating the capacity remaining at a node after taking into account the load from all traffic sources from all nodes (Fig. **??**). This is the difference between the *available capacity* and the capacity D^j demanded by each source j (control overhead is regarded as one kind of source), where γ_j in the equation is the *contention factor* that is a rough inverse measure of the spatial reuse, and indicates the number of nodes that have to defer on a transmission. We refer the reader to [**?**] for details on the contention factor and derivation of Equation (1), 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 (1). 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

$$D_j = \xi(QRF_j(u_j)), \qquad (2)$$

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

Combining with Equation (1), we have:

$$res = avail - \sum_j (1 + \gamma_j) \xi(QRF_j(u_j)). \qquad (3)$$

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 *graph sumsim*. This results in a load in bits using the nominal image size B as $K_{req} * B$. Next, we obtain the required rate by $r_{req} \geq \frac{K_{req} B}{T}$. The QRF function for the spanner is also fond similarly, with QoI demand $\mathbf{q} = (D, T)$ relates the requested diversity D to the number of images, hence traffic requirement.

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 use the multi-application scenario given in Figure **??**, and that we require a particular application QoI (score) q. How many nodes can the network support (i.e., what is the upper bound on N) as a function of q?

[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.

To determine this, we apply Equation (3) to the scenario. $QRF(q) = f^{-1}(q)$ where f is the function represented by Figure **??** that maps rates to achievable QoI.

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

Finally, the contention factor for unicast traffic in a mesh network is $\gamma_1 = 7$ 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.

Substituting in Equation (3), and noting that maximum scalability is when $res = 0$, we have $A = (1+7) \cdot \frac{2}{3} \cdot \sqrt{N} \cdot f^{-1}(q)$, which simplifies to

$$N = [\frac{3 \cdot A}{16 \cdot f^{-1}(q)}]^2. \qquad (4)$$

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.

Put the symptotics scalability equations here, including the scaling factor of the QoI function, and then include discussion.

$$\eta W = \sum_j (1 + \gamma_j) \xi(Q)(1 + T_j) \qquad (5)$$

V. Results

Figures 7 and 8 are some example figures that we can use to show the feasible scalability when considering QoI.

Clearly, there is a remarkable difference in the scalability depending upon the QoI.That QoI makes a difference is not suprising, but the *magnitude* of difference is surprising, along with the fact that there are some critical thresholding points. Our preliminary work shows that scalability analysis with QoI awareness has the potential to open up new tradeoff points with significant potential benefits in scalability. 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 **??**) and when such reductions will only give a marginal increase in scalability (e.g. from QoI=(3,40) to QoI=(3,45) in Figure 3).

A. Scalably Feasible QoI Regions

For the special case where each node possesses one image, we have observed the following dilemma. 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

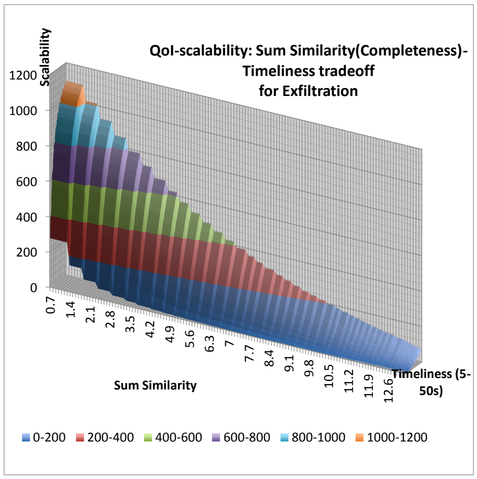

Fig. 3. Sum Similarity vs. Scalability vs. Timeliness, Unicast Traffic

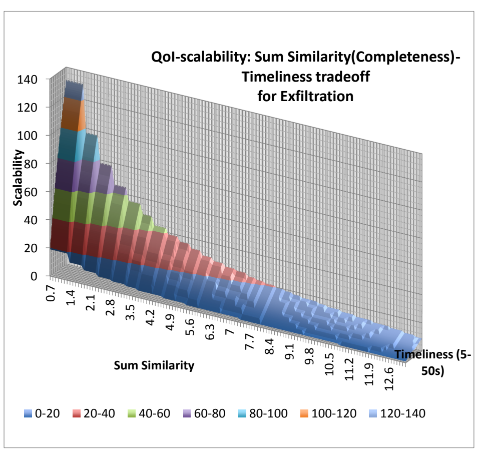

Fig. 4. Sum Similarity vs. Scalability vs. Timeliness, Flooding

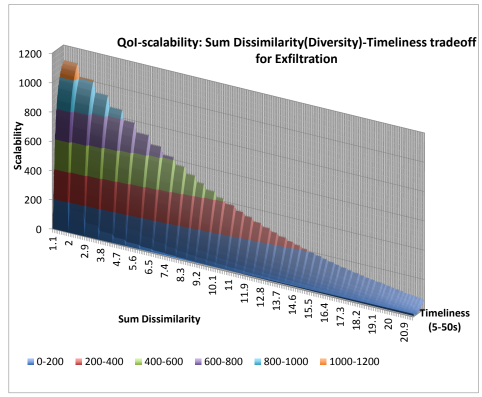

Fig. 5. Sum Dissimilarity vs. Scalability vs. Timeliness, Unicast Traffic

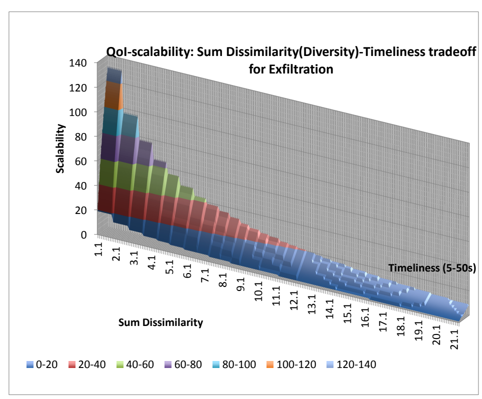

Fig. 6. Sum Dissimilarity vs. Scalability vs. Timeliness, Flooding

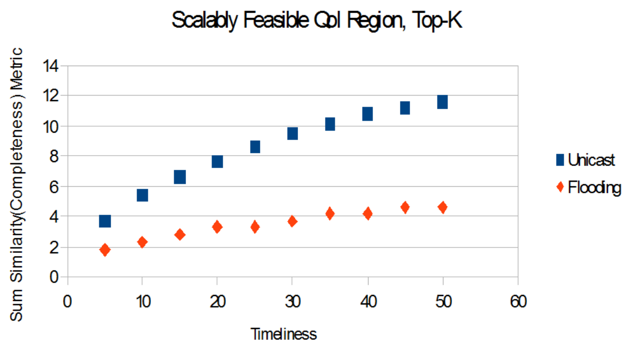

Fig. 7. Feasible Scalability Region of Top-K Algorithm

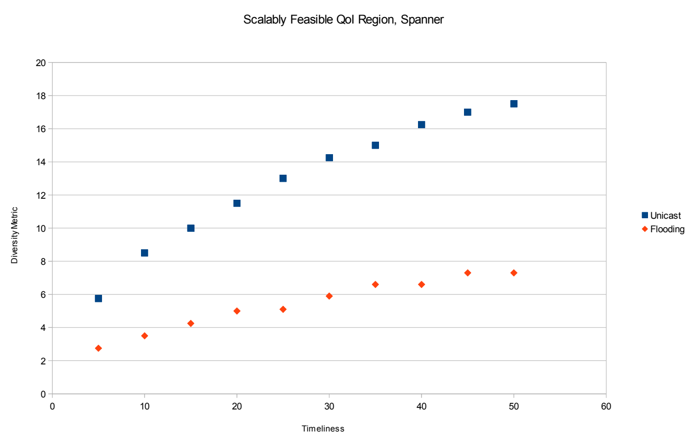

Fig. 8. Feasible Scalability Region of Spanner Algorithm

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 Figures 7-8.

As expected, the feasible QoI region is smaller for flooding compared with unicast. Moreover, these regions clearly demonstrate the tradeoff between the completeness/diversity that can be obtained and the timeliness that can be tolerated.

VI. Conclusion

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

References

[1] Savvas A Chatzichristofis and Yiannis 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.