This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE SECON 2005 proceedings.

Variable-Resolution Information Dissemination

Arpita Ghosh
Stanford University
CA 94305
arpitag@stanford.edu

Dan Greene Qingfeng Huang Juan Liu
Palo Alto Research Center
CA 94304
greene,qhuang,juan.liu@parc.com

Abstract

We consider the problem of information dissemination in wireless ad-hoc networks. Capacity constraints and the varying needs of applications lead to the need for delivering data at varying resolutions as a function of distance. We introduce a primitive, which we call visibility, to quantify the variable-resolution requirement. We design and analyze new variable resolution multicast algorithms based on simple probabilistic schemes to achieve a required visibility. We also examine how multi-path effects influence visibility and propagation algorithms. Finally, we consider the problem of designing the visibility function to maximize an overall utility function (specified by applications) over the network. We derive a condition relating optimal utility and visibility, which can be used to derive the optimal visibility for a class of utility functions.

1 Introduction

There are three basic information dissemination paradigms from the perspective of information flow patterns: one-to-one, one-to-many, and many-to-one. In this paper we introduce variable-resolution information dissemination (VRID) as a new one-to-many information dissemination paradigm. VRID allows users not only to express and specify the global constraints (such as the multicast IP address the group is subscribed to in IP multicast, or the set of geographical locations the recipients are in, as in geocast[7], or the set of spatial-temporal locale the recipients are supposed to be in as in mobicast[6]) for the potential recipient set, but also to specify the relative delivery probabilities among the set of potential recipients. In other words, VRID multicast no longer implicitly treats all potential recipients in the same way as conventional multicast paradigms do. VRID gives the users an additional dimension of control in multicast information dissemination, and is useful in developing a set of emerging applications on wireless ad hoc networks. With VRID one would be able to, for instance, deliver information with higher resolution (in space and time) to nodes closer-by and with lower resolution to nodes further away, with one call to the VRID service. In a sense, IP multicast, geocast, and mobicast are special cases of VRID with an expected delivery probability of 1 for the set of expected recipients.

The motivation of the generalized concept of variable-resolution information dissemination is based on the following observations on many ad-hoc network applications.

- Information has well-defined spatial value in many real-world applications. For instance, in a vehicle-to-vehicle collision warning application, the location and velocity information of a specific vehicle is useful for its neighboring vehicles, is less useful to vehicles further away, and is of no value to vehicles on the other side of the globe. Furthermore, neighboring vehicles may demand high accuracy from each other's the location and velocity data, and may be able to tolerate less accurate data from vehicles further away in the neighborhood.

- Information granularity requirements differ from application to application. For instance, an exit-aware lane-switching advisory system might require location and velocity of all vehicles within 500 meters, while a peer-to-peer cooperative traffic monitoring system might only need location and velocity information from a vehicle at every half kilometers interval at very low frequency. When there are multiple applications running on the same vehicle requiring the same type of information but with different granularity requirements, the resource usage can be further optimized by marshaling the demands and providing a protocol to serve the needs of all the applications rather than asking each application to run their information dissemination and data-collection protocols independently.

- A service allowing utility-level control for information dissemination will greatly simplify the programming task for many ad-hoc network applications. We will show that variable-resolution specification may be derived from an information utility function. Lower-level information dissemination in turn can be optimized for a given utility.

The contributions of this paper include: (1) the introduction and analysis of the visibility concept and the variable-resolution information dissemination idea; (2) analysis relating global visibility requirements with local packet forwarding strategies; (3) simple protocols achieving target visibility and a study of the effect of multi-path; (4) a study of the impact of conventional wireless ad hoc network broadcast/multicast heuristics on achieving target visibility; (5) linking information visibility requirements with higher-level information utility considerations, and examining how a utility model impacts the visibility specification for variable-resolution information dissemination.

The paper is organized as follows. Section 2 introduces a visibility primitive for VRID, and discusses some of the issues in defining such a primitive. Sections 3 and 4 present several algorithms for efficiently implementing the visibility primitive of section 2. Section 5 studies how higher-level applications can use a system based on the visibility primitive to support their information propagation needs. This involves developing a mathematical connection between an application's utility for information and the visibility primitive.

1.1 Related Work

Efficient information dissemination is of interest to many networked application domains spanning from distributed information systems[10], sensor networks[5, 14, 8, 4] and vehicular networks[12, 3, 16, 11]. Wireless ad hoc networks are often the technology substrate for applications in these areas. Past research in these areas has generated many novel protocols, insights and heuristics for cost-effective information dissemination in wireless ad hoc network. For instance, the issue of proximity based data dissemination is discussed in [10], where the architecture allows subscription to information based on proximity; different applications can subscribe to information within different ranges. In [11], the idea of layered data dissemination is discussed, once again motivated by a proximity-based need for information in different applications. The idea in [11] is to transmit packets with layers of data which are successively peeled off as information travels further. [16] discusses a broadcast protocol for dissemination of travel and traffic information with adaptive inter-broadcast times, reducing collisions and decreasing average error of information. [5] proposes to use meta-data negotiations to eliminate the transmission of redundant data throughout the network. [8] proposed a structured routing-searching paths for cost-effective information dissemination and discovery. Utility has been applied to selectively collecting sensor data in energy-constrained networks in [2].

However, as mentioned earlier, most of the previous work have been done under a conventional notion of uniform and deterministic delivery, i.e., a node is either a group member of a specific information delivery session, or not a member. Very little work has been done in the direction of non-uniform and variable information delivery. An earlier notion of non-uniform information delivery was proposed in [15]. Our work extends this work by introducing a novel notion of information visibility and linking it with a general definition of variable resolution information dissemination. We also present more rigorous mathematical analysis connecting visibility, propagation protocols, and multi-path effects, and examine the relationship between visibility and utility.

2 The visibility function

In this paper, we focus on the scenario of disseminating information to multiple recipients throughout the network. This differs from typical point-to-point communication in the sense that the goal is not to tranport information to any specific node, but rather to spread information so that all nodes can be aware to some extent about what is happening in the network.

In many applications, the requirement on information resolution varies with distance, with short-range nodes requiring more accurate information. This is due to the fact that closer range information is more relevant to immediate decisions. For example, a collision avoidance application would require that alarm information (accidents or sudden braking) propagates to a relatively small distance with a strong guarantee. Further out, this alarm information is less important, since vehicles at longer distance have plenty of time to respond. Similarly, a driver may often want information such as road conditions, weather updates, and the traffic situation in a local region around the vehicle. In such applications, a variable-resolution representation of the world, with resolution decaying with distance, is sufficient. Compared to broadcasting, variable-resolution information dissemination also significantly reduces consumption of communication bandwidth, which is a scarce resource in ad hoc wireless networks.

To achieve variable-resolution information dissemination, we first introduce the concept of a *visibility* *function*. It specifies to what extent the information originating from a point is 'visible' at a given distance from that point. For example, imagine a network along a 1-D line $x \in \mathbf{R}$. The visibility function $v(x)$ takes value in the interval $[0, 1]$, and specifies the 'percentage' of information from the source (assumed to be at $x = 0$) that reaches location x. The visibility can be similarly defined for higher dimensions. For clarity of exposition we will assume in this paper that v is a spherically symmetric function; the techniques described can be readily extended to asymmetric dissemination.

Now we discuss what we mean by 'percentage of information'. It is often hard to quantify information content, and even harder to define percentage, since both are related to application semantics. Depending on application requirements, information can be interpreted in various ways. One interpretation could be in the compression sense, *i.e.*, the entropy of the source is $H(0)$, but only $H(x) = H(0)v(x)$ bits are used to encode the information to be sent to x. Another interpretation is the reconstruction accuracy sense: suppose $\epsilon(0)$ is the error in reconstruction when all packets reach the destination, measured by some metric, and $\epsilon(x)$ is the reconstruction error at location x, then $v(x) = \epsilon(0)/\epsilon(x)$. However these definitions depend on specific compression or reconstruction schemes.

Here, we will not discuss definitions of visibility that are application dependent, but rather deal with a more general definition, (which we will be able to use to analyze the application-specific visibility for different applications). In this paper, we only focus on the issue of how to spread information. To disentangle ourselves from application-specific algorithms, we use a much simpler definition, which inherently assumes that all packets are equally important to applications[1]:

> The visibility function $v(x)$ is defined as the *percentage of packets* that reach a destination at distance x.

3 Algorithms for visibility

In this section we discuss algorithms for achieving a specified visibility function using probabilistic dropping schemes. In most of this paper, the applications considered require us to mostly use push style algorithms; thus we focus on the algorithms for push propagation (the algorithms for pull are similar).

In push propagation, there is one source node, and all other nodes are destination nodes; the source disseminates information to all other nodes in the network without any explicit requesting of information. There are two ways to probabilistically achieve a given visibility function with push.

1. **$\mathcal{A}_1$- Adjusting forwarding probabilities:** Given a visibility $v(x)$, messages sent out from the source are probabilistically dropped by nodes en route to achieve the visibility function. Note that since we use probabilistic dropping en-route, we can necessarily only deal with visibilities $v(x)$ that are decreasing[2]. The question in this case is to quantify how, *i.e.*, according to what probability distribution, are the nodes to drop the messages from the source. (For example, in the one-dimensional case, if all nodes on the route were to forward a message with a constant probability p, the only achievable visibilities would be exponential functions $v(x) = p^x$, and not any arbitrary decreasing function: in general, we need to be able to find the forwarding probabilities $p(x)$ that implement any given visibility $v(x)$). In what follows, we are considering a one-dimensional situation, with the source at $x = 0$ [3].

   First consider the simplest case, where v is a decreasing staircase function, *i.e.*, $v(x) = v_i,\ x_i \in [l_i, l_{i+1})$. (Note that making $\sup(l_i - l_{i+1})$ arbitrarily small allows us to approximate any decreasing function v.) In this case, the forwarding probability from node i to j is 1 if $x_i, x_j \in [l_k, l_{k+1}]$ for some k, and is v_{k+1}/v_k if $x_i \in [l_k, l_{k+1}]$ and $x_j \in [l_{k+1}, l_{k+2}]$. The overall distribution is then exactly the same as though the source tossed a coin with bias v_i to decide whether or not to send a message to nodes in region i and then directly communicated the message to these nodes [4].

   Now consider a smooth decreasing function v. Suppose nodes were placed unit distance apart on the x-axis, that is, node i is placed at i, and each node forwards messages with probability p_i. Then the visibility function

$$v(n) = \prod_{i}^{n-1} p_i. \quad (1)$$

[1] In compression applications, this is the same as assuming all packets are equal in reducing uncertainty; in estimation applications, this assumes a straight-forward down-sampling based estimation approach, as opposed to data filtering approaches.

[2] This is not a very limiting requirement: Most of the real visibilities we will be interested in will have this feature; or at worst will peak close to the source and then decay from that peak on. In the latter case, we just approximate this visibility by the closest decreasing function, *i.e.*, with $\tilde{v}(x) = v_{\max}$ until the peak, and $\tilde{v}(x) = v(x)$ there onwards.

[3] The one-dimensional case is not a bad approximation to the situation on a single roadway; a similar idea can be used in higher dimensions, although the effect of multipath needs to be considered, we discuss this later.

[4] From the practical point of view, it is easier to know the location of the sender of a received message than the receipient of a sent message; we discuss this briefly later.

For completeness, we want to derive a formula for a continuum of nodes on the x axis (the results for a discrete placement of nodes should be a special case of this formula). We define the continuous analog of the p_i, $p(x)$, by

$$\log v(x) = \int_0^x \log(p(x))dx \quad (2)$$

$$\Rightarrow p(x) = \exp((\log v(x))') \quad (3)$$

since $\log v$ is differentiable given the assumptions on v. Observe that if $p(x)$ is constant over $[i, i+1)$, then (2) reduces to (1).

Now suppose we have a placement of nodes i at locations x_i on the x-axis, with the source at the origin. (If we assume the point-to-point abstraction for communication, we have no multi-path effects: thus in the one-dimensional situation, a message from a node is heard by (and only by) the neighbor adjacent to it on its left.) To achieve the visibility $v(x)$, a node i at distance x_i from the origin, with the recipient of its message located at $x_j > x_i$, flips a coin with bias

$$v(x_j)/v(x_i) = \exp(\int_{x_i}^{x_j} \log(p(x))dx). \quad (4)$$

Most of the time, in real applications, this knowledge might be hard to obtain: it is easier to know, in addition to one's coordinates, the coordinates of the node x_i *from* which one received the message. Let the coordinate of the receiver be x_j. Node j can toss a coin with probability $v(x_j)/v(x_i)$, to obtain an upper bound on the visibility function with an error factor of p, where p is the visibility at the node closest to the origin. (That is, if the actual visibility should be $v(x)$ at some x, this scheme will cause it to be $v(x)/p$.)

This algorithm does not generalize quite as well to two dimensions. The effect of multipath in one-dimension can be modelled under somewhat restrictive assumptions, and a correction factor introduced; this is discussed in detail in the following subsection. However, in two-dimensions, the effect of multi-path is not quite as easy to analyze, since there are a large number of paths by which a message from a source can reach any given node.

The algorithm we discuss next has significant advantages over $\mathcal{A}_1$ in that it generalizes readily to multiple dimensions, and also lends itself more easily to policing node behaviour.

2. **$\mathcal{A}_2$- Choosing distance to propagate:**

Here, the visibility function $v(x)$ is achieved as follows: each time, the source selects a distance x_c uptil which the update will travel (the message is dropped with probability 1 by the first node with $\|x\| > x_c$, where $\|x\|$ is the distance of the node from the origin). The problem then is to determine the distribution $p_d(x)$ according to which this distance must be picked, in order to achieve the visibility $v(x)$. If a node at distance x receives a message, all nodes with distance less equal x receive it too; thus we can only achieve decreasing visibility functions v.

Since all nodes at a distance less or equal to x_c receive the message when x_c is chosen (with probability $p_d(x_c)$), the probability that a node at distance x gets the message is $\int_x^\infty p_d(x)dx$. Therefore, we must have

$$1 - v(x) = \int_0^x p_d(x)dx. \quad (5)$$

Suppose v is decreasing and differentiable, then setting $p_d(x) = -v'(x)$ satisfies the above. Therefore, we can choose the distance up to which to send according to the distribution $p_d(x) = -v'(x)$. If v is a staircase function, the solution is easy again: choose d_i, the distance at end of interval i, with probability $v_i - v_{i+1}$.

This algorithm scales obviously to n-dimensions, and is not affected by multipath in terms of the delivered visibility. Each node simply ignores multiple copies of the same message obtained via multipath. If the message is to travel to a distance x and completely die beyond x, this still happens despite multipath, with not much forwarding overhead, since repeated messages are not forwarded.

3.1 Effect of multipath

Here we briefly discuss the effect of multipath on the algorithms for implementing visibility in the push case, and make a case for using algorithm $\mathcal{A}_2$ for implementing push. In 1-D, for $\mathcal{A}_1$, we are able to approximately correct for multipath with some assumptions on the nature of the propagation. In 2-D, an analysis of the effect of multipath is extremely hard, and thus corrections need to be empirical. On the other hand, algorithm $\mathcal{A}_2$ is unaffected by multipath in terms of delivering a given visibility.

Consider the following situation in one-dimension: each node forwards a message with the same probability p, *i.e.*, $v(x) = p^x$, with message propagation outward from the source (located at the origin). We make the simplifying assumption that every node's transmission is heard by its m immediate neighbours to its right; the number on the left does not matter. This assumption corresponds to placing the nodes on Z^+, with the source at 0, and assuming a communication radius of m.

First we consider the protocol where each of the m recipients of the message forwards the message with probability p, independent of the actions of the other m nodes. The fact that there needs to be a mechanism ensuring non-simultaneous transmission to avoid losses due to collision is abstracted away. Let the probability that node j (*i.e.*, the node located at j) receives the message from the source be denoted by p_j. We can write a recursion for p_j in the presence of multipath m (the p_j are the attained visibilities):

$$1 - p_j = \prod_{i=1}^{m}(1 - p_{j-i}p). \tag{6}$$

Figure 1 shows the actual visibility attained in the presence of multipath, compared to the desired visibility. It is clear that even with a small multipath of 4, the visibility achieved is much higher than required, leading to a large amount of wasted communication bandwidth.

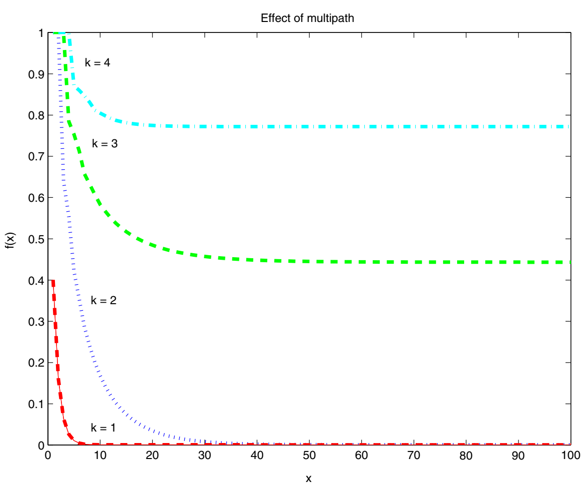

Figure 1. Achieved and desired visibility in the presence of multipath.

As it stands, the recursion (6) is a mth order, non-linear recursion; we can show that an approximation can be made to (6) bringing it to a m-th order, linear recursion, as

$$p_j = p(\sum_{i=1}^{m} p_{j-i}). \tag{7}$$

Suppose there is a solution of the form $p_j = x^j$, then it must satisfy $x^j = p(\sum_{i=1}^{m} x^{j-i})$, *i.e.*

$$x^m = p(x^{m-1} + \ \dots \ + 1). \tag{8}$$

To achieve $f(n) = p_0^n$ in the presence of k-multipath, we want the forwarding probability to be such that $x = p_0$, *i.e.*,

$$p_f = \frac{p_0^m}{p_0^{m-1} + \ \dots \ + 1}. \tag{9}$$

If each node forwards the message with probability p_f instead of p_0, then the actual visibility achieved is approximately p_0^n. Simulations indicate that this solution to the approximate recursion is indeed a good approximation to the actual solution of the recursion (6), solved with p_0 replaced by p_f from (9). Figure 2 shows one such plot. In this one-dimensional case, we are also able to analyze the more realistic protocol where the transmission from a node suppresses that from other nodes within range of that transmission, and correct for the effect of multipath.

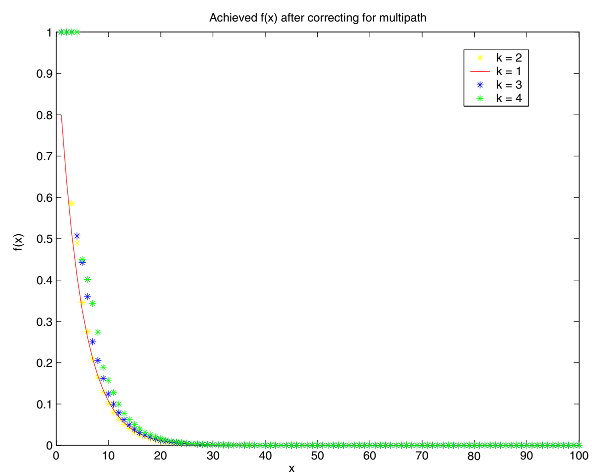

Figure 2. Achieved visibility after correcting for the presence of multipath.

However, none of these analyses extend to 2-dimensions, and it is hard to derive a correction factor that does not overestimate too much the required visibility nor underestimate it anywhere. Thus, in a situation with multipath, it is advantageous to use algorithm $\mathcal{A}_2$ (which is unaffected by multipath) for achieving a given visibility function.

4 Integrating Visibility with Propagation Algorithms

The previous section we compared predetermined dropping ($\mathcal{A}_2$) with in-transit dropping of information ($\mathcal{A}_1$), and concluded that predetermined dropping was particularly simple to implement and had many desirable properties. In this section we consider the details

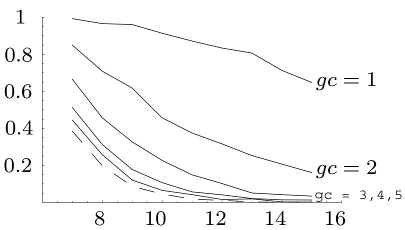

Figure 3. Failure rate for a gossip algorithm at various densities.

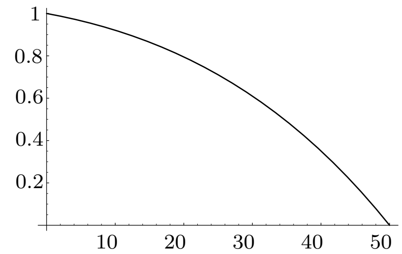

Figure 4. A linear visibility function adjusted in advance for in-transit failure

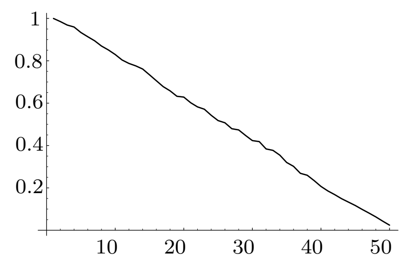

Figure 5. Actual visibility observed after precorrection.

of implementing $\mathcal{A}_2$. In particular, we find that a predetermined approach, that can accommodate some accidental dropping of information in-transit, has excellent traffic characteristics.

The broadcast storm problem (Ni et al [13]), can occur when algorithms flood information in an ad-hoc network: congestion causes nodes to delay propagating an update, and they eventually transmit updates long after they are useful to their neighboring nodes. The broadcast storm problem has been studied and numerous solutions have been proposed. In this paper we will use the elegant gossip-style algorithms of [13] to illustrate the issues of implementing visibility. In these algorithms nodes receiving new updates do not always retransmit the update, but rather listen to neighboring broadcasts; if the update is overheard a predetermined number of times (the "gossip count") then the update is not propagated. Figure 3 shows simulation results of gossip-style algorithms on an linear example typical of vehicle roadways. For these simulations the nodes were distributed uniformly in a 1 unit by 50 unit linear strip, according to various densities shown on the x-axis. A time-step simulation was used, nodes were able to transmit 1 unit distance, and (to model carrier-sense conflict avoidance) a random covering was chosen so that transmitting nodes could not hear other transmissions (although hidden terminal conflicts could occur). Each line shows a different gossip count, and the broadcast storm (an infinite gossip count) is shown as a dashed line.

The simple and robust gossip-style algorithm is an attractive choice for propagating information with a visibility function. It's not the only choice, and we will see later that some improvements are possible, however, we will use the gossip-style algorithm to illustrate the integration of visibility with propagation algorithms. Other propagation algorithms can be similarly modified.

At a density of 10, Figure 3 shows a gossip algorithm ($gc = 2$) with a loss of 50.5% over the length of 50 units. Assuming a constant $p(x)$ in (3) gives $p(x) = 0.986$ and a uniformly decaying visibility: $v_f(x) = e^{-0.01406x}$. This visibility is the result of accidental loss; to obtain a desired target visibility, for example, to obtain a linearly decaying $v_t(x) = 1 - x/50$, we can use a "pre-corrected" $v_c(x) = v_t(x)/v_f(x)$. An example is shown in figure 4, for the linear target, and simulation results are shown in figure 5. This is a very efficient way to achieve linear visibility: the gossip algorithm at density 10, $gc = 2$ has only 36% of the nodes transmitting the update. But this technique has limitations. First, at low gossip counts and high in-transit loss rates, not all target visibilities can be realized efficiently, since the pre-corrected visibility must still be less than 1. In fact, it is keeping the pre-corrected visibility less than 1 that sets the limit on possible communication savings with this pre-correction technique. Second, at low gossip counts the failure rate is sensitive to density, so the correction may be inaccurate. Figure 6, shows the results of the same approach as figure 5, but with density varying sinusoidally between 7 and 13 rather than being fixed at 10.

For the linear strip simulation model we are using here, the most significant contribution to propagation failure is a gap-jump failure. This occurs when there is a gap between neighbors that is larger than the communication range—none of the nodes across the gap will receive the update. With the gossip algorithm, a gap jump-failure is more likely because all nodes that have received an update on one side of the gap, will not necessarily transmit the update, in particular, the nearest nodes to the gap may not transmit the update when they have already heard other nearby nodes transmit-

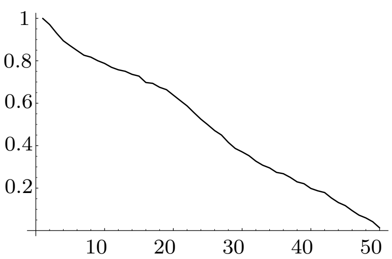

Figure 6. Consequences of variation in density.

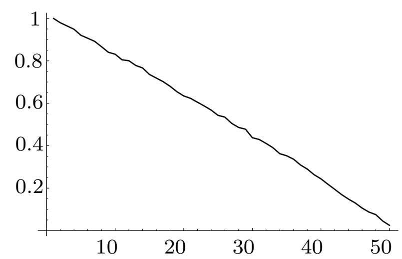

Figure 7. Results of varying gc to compensate for variations in density.

ting the update. Consider a single gap jump problem: k points are drawn uniformly in the interval $[-1, 0]$ and a gap Z is drawn in the interval $[0, \infty]$ according to an exponential distribution $\lambda e^{-\lambda z}$. If X_r, the rightmost point from $[-1, 0]$,is within communication range of Z, that is $Z - X_r \leq 1$ then we say that the gap is jumped. This problem can be analyzed to show a probability of gap-jump failure of:

$$\lambda^{-k} k(Gamma[k] - Gamma[k, \lambda]). \qquad (10)$$

For a single gap, equation (10) relates gap-jump failure to density λ and gossip count $k - 1$. The failures of figure 3 are more complex than this simple gap jump problem, resulting from propagation along an entire linear strip that is slightly more than 1-dimensional, with multiple potential gap-jump failures and non-uniform distribution of points with updates. However, because we will eventually pre-correct the visibility function to compensate for actual in-transit loss, as we did in figure 4, we only need a model that captures the salient features of the problem—in this case a model that relates density and gossip count to failure rate.

In particular, to deal with the variable density problem we can have nodes sample their local density and then adjust their gossip count to obtain a more stable failure rate. Figure 7 shows the results of using the following formula to set the gossip count gc depending on the local density d:

$$gc(d) = 0.92e^{(10.17/d)}, \qquad (11)$$

and applying it to the variable density problem of figure 6. The visibility function was pre-correcting as in figure 4, for an observed per-unit forwarding probability of 0.986. The density d is computed from the number of nodes within immediate communication range (while density estimation could be extended several hops, one hop is simple to compute and works reasonably well). The fractional portion of $gc(d)$ is interpreted probabilistically, so that a node will always use an integer gossip count of either the floor or ceiling of $gc(d)$. The constants in equation 11 were obtained by fitting the form of equation (11) to the gap-jump failure model to equalize the failure rate across density.

While gc can be varied to make the system more robust against variations in density, perhaps equally important, it also can be varied to reduce traffic. Since traffic is roughly linearly dependent on gossip count, we would like gc to be "well spent," that is, gc should be high for updates that still have large distances to propagate, where the potential loss from dropping these updates would be greater. More precisely, gc can be adjusted dynamically based on the remaining distance an update has to propagate, to equalize the marginal reduction in failure rate per gc. We performed a simulation experiment on the constant density example ($d = 10$) above, using the following formula for gc based on r (the remaining distance to propagate):

$$gc(r) = 0.235 - 0.704 \log(1/r) \qquad (12)$$

We found that the dynamically adjusted gc reduced traffic by 15%, while slightly improving the in-transit loss. As before, we could pre-correct the visibility function to remove the effects of in-transit loss. The constants in equation (12) were obtained by fitting the form of equation (12) to a simplified variant of gap-jump failure model, to achieve constant marginal benefit for gossip count.

5 Visibility and utility

So far we have discussed the visibility primitive, and algorithms and implementations given a visibility function. In this we consider the question of where a visibility function comes from. Clearly, the visibility will depend on the application it is being used for: it is clear that the shape of the visibility curve is very different, for example, for an emergency alert application and a route planning application. Here, we quantify this relationship by modelling the information need of an application in terms of a *utility* function.

With an application, we associate a utility function U, where $U = U(r, v)$ is a function of distance from source r, as well as the visibility v at which information is received. The dependence on r reflects the fact that the value of information received is a function of distance from the source of the message; the dependence on v reflects the fact that the visibility affects the quality of reconstruction, and therefore the value of the reconstructed information.

In what follows, we will discuss how to design the visibility function $v(r)$ to maximize the *overall utility* over the network, where the overall utility is the integral over the entire network of the utility at each distance r.

5.1 Capacity constraints

With no constraint on the visibility function, the overall utility would be maximized by having a constant visibility of 1, *i.e.*, $v(r) = 1$ for all r. This is because for any r, the maximum utility is for $v = 1$: one cannot do a worse reconstruction with more information than with less. Therefore, to make a meaningful problem, we must first outline the constraints on v, which arise from capacity limits in the network.

Suppose we have a circularly symmetric visibility function in two dimensions $v(r, \theta) = v(r)$. Suppose that the nodes are distributed with some density $D(r)$, which again for simplicity we assume to be radially symmetric. Assume that all nodes are generating updates at the rate of 1 update a second. Then, the total number of messages reaching the origin per second is $\int v(r)D(r) \cdot 2\pi r dr$. This number must be less than the capacity at the origin; therefore,

$$\int v(r)D(r) \cdot 2\pi r dr \leq C. \tag{13}$$

For example, if density is constant, and $v(r) \neq 0$ for $r < \infty$, then this indicates that $v(r)$ must be $o(1/r^2)$ in order to be able to satisfy the capacity constraint[5].

For simplicity, we will henceforth assume we are dealing with a constant density D and radially symmetric functions U and v. So, we can now state the utility maximization problem as follows:

$$\begin{array}{rc} \text{maximize} & \int_r U(r,v)2\pi r dr \\ \text{subject to} & 2\pi D \int_r v(r) r dr \leq C, \\ & 0 \leq v(r) \leq 1, \\ & \dot{v}(r) \leq 0. \end{array} \tag{14}$$

The last two constraints come from the fact that the visibility is a decreasing function of distance taking values between 0 and 1.

Consider a relaxation of (14), obtained by removing the last two constraints; if the solution of the relaxation satisfies these constraints, then we have found the solution to the original problem. For this relaxed problem, we can derive a very simple necessary condition for a visibility $v(r)$ to be optimal for (14).

A necessary condition for $v(r)$ to be optimal for the relaxed problem is that the Lagrangian have a stationary point at v ([9]). Thus, we are looking for a stationary point of

$$\int_r U(v,r)2\pi r + \lambda v(r) 2\pi r dr.$$

Using the Euler-Lagrange equations, we require that

$$\frac{\partial U(r,v)}{\partial v} 2\pi r + \lambda 2\pi r = 0. \tag{15}$$

Therefore, we must have

$$\frac{\partial U(r,v)}{\partial v} = -\lambda, \tag{16}$$

that is, $\frac{\partial U(r,v)}{\partial v}$ is constant through the network.

An intuitive explanation for this equation is as follows: for an optimal $v(r)$ that maximizes overall utility, it should not be possible to move a small amount of visibility from some location r_1 to some other location r_2 that results in a change in overall utility, *i.e.*, $\frac{\partial U(r,v)}{\partial v}$ must be the same throughout the network.

Note that (16) is a only a necessary condition for the relaxed problem. However, under certain circumstances, we can use this to extract the optimal visibility function $v(r)$.

Suppose that $U(r, v)$ is separable in r and v as $f(r)g(v)$. That is, the effect of visibility on quality of reconstruction is modeled in $g(v)$, and the effect of a decrease in value of the reconstructed information is modelled in $f(r)$. In this case, (16) can be written as

$$f(r)\frac{\partial g}{\partial v} = k. \tag{17}$$

Let $G(v) = \frac{\partial g}{\partial v}$. If G is an invertible function, then we can derive the optimal visibility from this condition as

$$v(r) = G^{-1}(\frac{k}{f(r)}), \tag{18}$$

where the necessary condition is now sufficient to give us an optimal $v(r)$, since there is only one solution to the above equation. If further this v is a decreasing function lying between 0 and 1, then we have actually solved the original optimization problem (14).

Furthermore, finding the visibility function satisfying (14) can be simplified when $g(v)$ is a concave function of v. The concave assumption is very plausible, and commonly encountered in practice: it is equivalent to

[5]Note that the presence of multiple sources is what gives rise to a capacity constraint of this form: if only one source were transmitting, we would only have the constraint that the rate at which it generates updates is less than the capacity C.

diminishing returns, which is satisfied by many practical utilities. One can easily show that for concave $g(v)$, the visibility function (18) is decreasing in r. Therefore, the last constraint in (14) (i.e., the monotonicity constraint) is automatically satisfied. To satisfy the remaining constraint that $0 \leq v \leq 1$, one can adjust the free parameter k to find a feasible solution.

If the utility $U(v,r)$ is concave in v, even if it is non-separable, the problem (14) can also be solved numerically. Divide the range r into intervals of length Δr, and assign a (discrete) variable v_i for the ith interval[6]. The value of $U(v,r)$ in the ith interval can be approximated by the value at the midpoint r_i. The problem then becomes

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{N} c_i U(v_i, r_i) \\ \text{subject to} & \sum_{i=1}^{N} d_i v_i \leq C, \\ & 0 \leq v_i \leq 1, \quad i = 1, \ldots, N \\ & v_{i+1} - v_i \leq 0, \quad i = 1, \ldots, N-1, \end{array} \tag{19}$$

where N is the total number of intervals, and $c_i = 2\pi r_i \Delta r$, $d_i = r_i \Delta r$ are constants. Since U is concave, the objective function is a sum of concave functions, and is therefore concave in the v_i. The constraints are linear in v_i, and therefore we have a concave maximization problem, which can be solved efficiently [1].

Thus, we can now move from the requirements of an application to a visibility function, by constructing a utility function that reflects the need of the application as a function of distance, and by analyzing how fidelity in the application changes with visibility.

Example: As an example, we study the problem of detecting rare events as an instance of deriving a visibility function from utilities. Suppose we want to detect the occurence of a rare event. For example, we might be interested in using the differential slip of a vehicle as an indicator of icy road conditions. If there is a large number of high reported slips, it is more likely than not that the road is icy, but a few reports are a weaker indication. The number of reports scales with visibility, and thus visibility must be accounted for in the decision problem.

In the appendix, we model the rare event problem, and express the $g(v)$ portion of the utility as a function of visibility, which is used to derive a visibility for utility maximization. The results for a sample instance are shown in Figure 8. Using the dependence $g(v)$ of the utility on the visibility, we can now numerically obtain a visibility function that maximizes the utility for the network once we know $f(r)$, the decay of the value of information with distance. For $f(r) = 1/r^3$, and $g(v)$ as in Figure 8, the obtained visibility is shown in Figure 9.

[6]For practical problems, the range r is bounded as well, albeit by some large value.

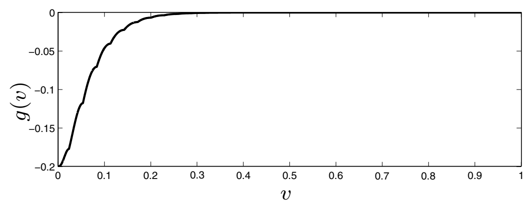

Figure 8. An example of utility $g(v)$ as a function of visibility for rare-event detection.

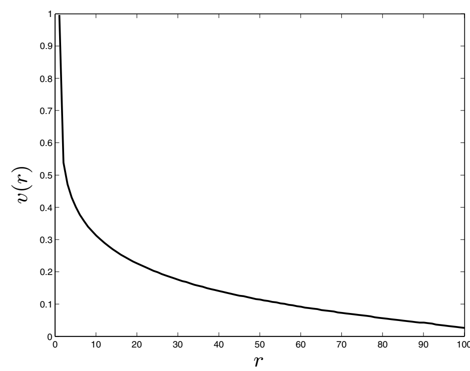

Figure 9. Optimal visibility for rare event detection with $f(r) = \frac{1}{r^3}$.

6 Discussion

We have developed the concept of visibility to interface between higher-level applications and lower-level information propagation algorithms. As an interface concept, it has many desirable properties: it is simple to describe, and because of its simplicity it is possible to mathematically link it to relevant quantities on either side of the interface, *i.e.*, we can analytically relate visibility to quantities such as utility and information loss.

In this paper we have not discussed filtering. Since visibility is a form of downsampling, filtering before downsampling could be useful to avoid aliasing artifacts when a signal is reconstructed. Here we have assumed that the data does not contain high frequency artifacts, an assumption that is reasonable for many vehicle applications. However, we are continuing to research combining filtering with visibility.

As an illustration of the potential value of filtering, and its interaction with utility, we note that the rare events example of the previous section can be modified as follows: in addition to propagating individual rare events, sensors can average rare events that they have received from neighboring nodes. Because these aggregated events are averages they will have lower variance and higher utility in the decision problem. They can also be propagated with a visibility function, derived as above, with much higher dropping and consequently lower traffic rates. However, this aggregation introduces additional delay, and creates some additional issues in reconstruction. It is likely that a combination of direct propagation of rare events to nearby nodes, and delayed propagation of aggregated events to distant nodes is the best approach to this application. In both instances, however, the visibility function is governed by the analysis of the previous section.

References

[1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2003. Available at `http://www.stanford.edu/~boyd/cvxbook.html`.

[2] J. W. Byers and G. Nasser. Utility-based decision-making in wireless sensor networks. In *MobiHoc*, pages 143–144, 2000.

[3] H. Hartenstein, B. Bochow, A. Ebner, M. Lott, M. Radimirsch, and D. Vollmer. Position-aware ad hoc wireless networks for inter-vehicle communications: The fleetnet project. In *Proc. ACM Symposium on Mobile ad-hoc networking and computing*, pages 259–262, 2001.

[4] J. Heidemann, F. Silva, and D. Estrin. Matching data dissemination algorithms to application requirements. In *Proceedings of the ACM SenSys Conference*, pages 218–229, November 2003.

[5] W. R. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 174–185, New York, NY, USA, 1999. ACM Press.

[6] Q. Huang, C. Lu, and G.-C. Roman. Mobicast: Just-in-time multicast for sensor networks under spatiotemporal constraints. In *Proc. of the 2nd International Workshop on Information Processing in Sensor Networks*, pages 442–457, Palo Alto, CA, USA, April 2003.

[7] Y. Ko and N. Vaidya. Geocasting in mobile ad hoc networks: Location-based multicast algorithms. In *Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, pages 101–110, New Orleans, LA, February 1999.

[8] X. Liu, Q. Huang, and Y. Zhang. Combs, needles, haystacks: balancing push and pull for discovery in large-scale sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 122–133, New York, NY, USA, 2004. ACM Press.

[9] D. Luenberger. *Optimization by Vector Space Methods*. John Wiley and Sons, Inc., 1969.

[10] R. Meier and V. Cahill. Exploiting proximity in event-based middleware for collaborative mobile applications. In *Proc. Distributed Applications and Interoperable Systems: 4th IFIP WG6.1 International Conference*, pages 285–296, 2003.

[11] L. B. Michael. Adaptive layered data structure for inter-vehicle communication in ad-hoc communication networks. In *Proc. 8th World Congress on Intelligent Transportation*, 2001.

[12] R. Morris, J. Jannotti, F. Kaashoek, J. Li, and D. Decouto. Carnet: a scalable ad hoc wireless network system. In *Proc. 9th workshop on ACM SIGOPS European Workshop*, pages 61–65, 2000.

[13] S. Ni, Y. Tseng, Y. Chen, and J. Sheu. The broadcast storm problem in a mobile ad-hoc network. In *Proc. ACM MobiCom*, 1999.

[14] C.-C. Shen, C. Srisathapornphat, and C. Jaikaeo. Sensor Information Networking Architecture and Applications. *IEEE Personel Communication Magazine*, 8(4):52–59, August 2001.

[15] S. Tilak, A. Murphy, and W. Heinzelman. Non-uniform information dissemination for sensor networks. In *Proceedings of the International Conference on Network Protocols*, November 2003.

[16] L. Wischhof, A. Ebner, H. Rohling, M. Lott, and R. Halfmann. Adaptive broadcast for travel and traffic information distribution based on inter-vehicle communication. In *Proc. Intelligent Vehicle Symposium*. IEEE, 2003.

Appendix: Modeling rare events

Denote the event of the underlying phenomenon's occurence by R. Suppose there are n nodes in the region on interest. Let X_i, $i = 1 \dots, n$ be the event that node i senses R. Let $p(X_i|R) = p_1$ be the conditional probability of the nodes registering the event r given that it happens. Let $p(X_i|\bar{R}) = p_2$ be the noise probability of an event being wrongly detected (ideally, p_1 is much larger than p_2). Let p_r denote the probability of the rare event R.

Suppose that the receiver uses threshold c to determine whether the event occured or not. Denote by Y the event of deciding that R occured. The receiver associates utilities u_1, u_2, u_3 and u_4 with the events $Y|R$, $Y|\bar{R}$, $\bar{Y}|R$, and $\bar{Y}|\bar{R}$, where by $\bar{A}$ we mean the complement of event A. The utility achieved for a given value of the threshold c,

$$\begin{aligned} U(v,c) &= u_1 P(Y|R)p_r + u_2 P(Y|\bar{R})(1-p_r) + \\ & \quad u_3 P(\bar{Y}|R) + u_4 P(\bar{Y}|\bar{R}) \\ &= u_1 p_r + u_4(1-p_r) + p_r(u_3-u_1)P(\bar{Y}|R) + \\ & \quad (1-p_r)(u_2-u_4)P(Y|\bar{R}). \end{aligned}$$

Suppose initially that $v = 1$, then $P(\bar{Y}|R) = \sum_{i=0}^{c-1}\binom{n}{i}p_1^i(1-p_1)^{n-i}$, and $P(Y|\bar{R}) = \sum_{i=c}^{n}\binom{n}{i}p_2^i(1-p_2)^{n-i}$.

Consider what happens when we have a visibility $v < 1$. Now the analysis depends on the protocols used. One option is to say that on average, with a visibility of v, nv messages reach the receiver: in this case, we replace n by nv. Alternately, messages could be sent only if the node detected the presence of the rare event: in this case, we replace p_1 and p_2 with $p_1 v$ and $p_2 v$.

Now consider the question of choosing the optimal threshold. Let $u_{11} = p_r(u_3 - u_1)$ and $u_{22} = (1-p_r)(u_2 - u_4)$. To find c that maximizes $U(v, c)$, we first find the (non-integer) i that solves

$$\begin{aligned} u_1 1(n_i) p_1^i (1-p_1)^{n-i} &= u_{22}(n_i) p_2^i (1-p_2)^{n-i} \\ \Rightarrow [\frac{p_1(1-p_2)}{p_2(1-p_1)}]^i &= \frac{u_{22}(1-p_2)^n}{u_{11}(1-p_1)^n} \\ \Rightarrow i &= \frac{\log(\frac{u_{22}(1-p_2)^n}{u_{11}(1-p_1)^n})}{\log(\frac{p_1(1-p_2)}{p_2(1-p_1)})}, \end{aligned} \tag{20}$$

which is the intersection of the corresponding continuous binomials. We can then appropriately round i (*i.e.*, by choosing the larger of $U(v, \lfloor i \rfloor)$ and $U(v, \lceil i \rceil)$) to obtain the optimal c.