Measurement-Based Design of Roadside Content Delivery Systems

Vinod Kone, Haitao Zheng, Antony Rowstron, Greg O'Shea, and Ben Y. Zhao

Abstract—With today's ubiquity of thin computing devices, mobile users are accustomed to having rich location-aware information at their fingertips, such as restaurant menus, shopping mall maps, movie showtimes, and trailers. However, delivering rich content is challenging, particularly for highly mobile users in vehicles. Technologies such as cellular-3G provide limited bandwidth at significant costs. In contrast, providers can cheaply and easily deploy a small number of WiFi infostations that quickly deliver large content to vehicles passing by for future offline browsing. While several projects have proposed systems for disseminating content via roadside infostations, most use simplified models and simulations to guide their design for scalability. Many suspect that scalability with increasing vehicle density is the major challenge for infostations, but few if any have studied the performance of these systems via real measurements. Intuitively, per-vehicle throughput for unicast infostations degrades with the number of vehicles near the infostation, while broadcast infostations are unreliable, and lack rate adaptation. In this work, we collect over 200 h of detailed highway measurements with a fleet of WiFi-enabled vehicles. We use analysis of these results to explore the design space of WiFi infostations, in order to determine whether unicast or broadcast should be used to build high-throughput infostations that scale with device density. Our measurement results demonstrate the limitations of both approaches. Our insights lead to Starfish, a high-bandwidth and scalable infostation system that incorporates device-to-device data scavenging, where nearby vehicles share data received from the infostation. Data scavenging increases dissemination throughput by a factor of 2-6, allowing both broadcast and unicast throughput to scale with device density.

Index Terms—Wireless, wide-area networks, communication/networking and information technology, computer systems organization, performance of systems, computer systems organization

◆

1 INTRODUCTION

Recent years have seen a rapid growth of smartphones and tablet PCs, making thin computing devices nearly ubiquitous at work, at home, and on the road. Using the WiFi, GPS, and cellular 3G interfaces built into these devices, mobile users are now accustomed to having location-aware information at their fingertips. Location-aware information services such as digital versions of restaurant menus, shopping-mall maps, transportation schedules, grocery store circulars, and movie trailers can be delivered via mobile applications [1], or through dedicated WiFi hotspots [2], [3].

Delivering content at high bandwidth, however, remains a significant challenge for highly mobile users in vehicles, e.g., cars, buses, and trains. Existing technologies such as satellite-based broadcasting and FM-radios are widely deployed, but provide low bandwidth links. Cellular-3G provides higher bandwidth, but only with significant monthly costs. More importantly, cellular operators, overwhelmed by data usage on their networks, are implementing rate restrictions and WiFi offloading to discourage mobile clients from using their mobile data services [4], [5]. Thus, we believe WiFi infostations [6] are the ideal alternative for deploying location-aware information services to mobile users. With the ubiquity and high-bandwidth of WiFi devices, content providers such as restaurants and tourism offices can easily deploy a small number of WiFi infostations that quickly deliver content to users passing by. Individual infostations could be placed at key locations such as freeway exits into a city, and at street corners near the places of interest.

While sufficient for low-mobility users, WiFi infostations still face two challenges in delivering high-quality content to highly mobile users, e.g., users in moving vehicles. First, given the short time a vehicle is in range of the infostation, users do not have time to interact with roadside infostations to choose the content they desire. Recent studies [7], [8] have also shown that unlike 3G, WiFi communications cannot support continuous access to vehicles. As a result, a WiFi device must proactively fetch all available data from the infostation, so that the user can browse information of interest offline. Therefore, vehicles need to download a large common content (like a tourist package) as they pass by the infostation.

The second major challenge is *rate scalability*: the rate of data dissemination must be high and scale as the number of receivers increases. This means the system must work well when a small fraction of vehicles carry equipped devices, e.g., initial deployment, but also scale up as device density increases. If infostations use traditional unicast, then the bandwidth offered by an infostation is effectively divided by the number of vehicles[1] in range. Ironically, this means

- *V. Kone, H. Zheng, and B.Y. Zhao are with the Department of Computer Science, University of California, Santa Barbara, Santa Barbara, CA 93106. E-mail: {vinod, htzheng, ravenben}@cs.ucsb.edu.*
- *A. Rowstron and G. O'Shea are with Microsoft Research, 7 J J Thomson Avenue, Cambridge CB3 0FB, United Kingdom. E-mail: {antr, gregos}@microsoft.com.*

Manuscript received 3 Apr. 2011; revised 31 Jan. 2012; accepted 23 Mar. 2012; published online 5 Apr. 2012.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2011-04-0178.
Digital Object Identifier no. 10.1109/TMC.2012.90.

---

[1] We use the terms devices, vehicles, and receivers interchangeably.

Published by the IEEE CS, CASS, ComSoc, IES, & SPS

as the system becomes more popular and the density of vehicles using it increases, the average amount of data that a vehicle receives from the infostation drops. Therefore, either the size of the content distributed needs to be reduced as the system becomes more popular, which is clearly nonideal, or more infostations need to be deployed, which is shown to be costly and can limit scalability in high-density environments [9]. In contrast, broadcast-based infostations have the benefit that the data dissemination rate is not a function of the number of devices in range. However, using broadcasts introduces its own challenges: 802.11 broadcasts are unreliable [10], [11], [12], because they do not use acknowledgments or retransmissions; they also use a static data rate, and cannot leverage dynamic rate adaptation. Both of these limit dissemination throughput.

Multiple projects have proposed techniques to improve data dissemination using application-level encoding techniques such as network coding [13], [14] and BitTorrent-like protocols [15], [16]. However, all of these systems are either designed on top of the wireless transport layer, or assume reliability through retransmissions at the wireless MAC layer. Few studies have examined the issue of vehicle density on real systems, or how well coding techniques perform in high mobility, high-loss environments.

In our work, we reexamine the design of infostation-based data dissemination systems using insights from a large experimental measurement. We perform detailed measurements on a public highway using a fleet of four vehicles traveling at up to 96 km/h (60 mph), each equipped with a WiFi-enabled laptop and communicating with a roadside infostation. Using data from nearly 200 h of on-road driving, we measure the impact of vehicular density on both unicast- and broadcast-based systems. Our measurements show that both broadcast and unicast have their limitations. For broadcast, the high loss rates of the wireless channel severely degrade the dissemination throughput. Application level encoding at the infostation improves reliability, but the coding overhead and lack of rate adaptation limit the total achievable throughput. For unicast, per-vehicle throughput degrades due to time sharing as vehicle density increases.

Remarkably, our measurements also show that data reception across multiple vehicles exhibit both *temporal diversity*, where different vehicles have the opportunity to receive different data, and *spatial diversity*, which produces uncorrelated loss across vehicles receiving the same data. To get either broadcast- or unicast-based infostation to scale and disseminate content at high bandwidth, we propose to augment infostation bandwidth with communication bandwidth between the devices. We propose *data scavenging*, where immediately after passing through an infostation, vehicles cooperate and share information with others nearby. Intuitively, data scavenging exploits natural diversity produced by vehicle mobility in order to achieve throughput improvements similar to network coding. It exploits both receiver spatial diversity and temporal diversity to compensate for packet losses without retransmissions, and also increases the infostation's achievable dissemination throughput. Scavenging requires no hardware modification to vehicle devices (or infostations), and is ideal for location-aware content delivery, where neighboring vehicles seek to obtain a common content package and can effectively help each other.

We integrate the proposed data scavenging technique into both broadcast and unicast infostations. Our infostation-based content distribution system, *Starfish*, has a high data dissemination rate that scales as the number of devices grows. We implement a prototype on our testbed and evaluate its performance. Our four-vehicle experiments show that using data scavenging increases the data dissemination throughput by a factor of 2 to 6. We also use QualNet simulations to understand the performance of Starfish at larger scales, for example, with different vehicle densities and at different speeds.

We make four key findings.

1. At higher vehicle densities, data dissemination across vehicles exhibit both spatial and temporal diversity.
2. Data scavenging between vehicles is feasible, and can provide a scalable dissemination service when combined with either efficient broadcast or unicast.
3. For broadcasting infostations, scavenging improves dissemination reliability without sacrificing throughput; for infostations using unicast, data scavenging allows dissemination to scale with number of vehicles.
4. Unicast with scavenging outperforms broadcast with scavenging in most scenarios, except when both vehicle density and speed are high.

Roadmap. First, we motivate our problem and describe the design of broadcast and unicast infostations in Section 2. Then in Section 3, we describe our vehicular testbed, analyze our testbed measurements, and make key observations to improve infostation throughput. We then propose the design of our data scavenging protocol in Section 4, and evaluate our Starfish prototype in Section 5. Section 6 describes simulations where we study the impact of several system parameters. Finally, we discuss related work in Section 7 and conclude in Section 8.

2 DISSEMINATING LOCATION-AWARE CONTENT WITH STARFISH

The goal of our work is to design an efficient and scalable delivery system to distribute location-aware content for mobile users on the move. In this section, we describe *Starfish*, an infostation-based content delivery system for highly mobile users. We begin by defining the problem of delivering location-aware content, and then describe designs for Starfish based on roadside infostations using either wireless unicast or wireless broadcast.

2.1 Location-Aware Content Delivery

We are primarily interested in scalable ways to deliver location-aware content to mobile users. We define location-aware content as local information of interest to travelers on the go. Consider for example, a family driving on a highway, where there are several services (e.g., entertainment, food, gas) at the next 10-20 exits. Ideally, they would like to take an exit which has the best combination of services and prices. Now imagine, a road-side infostation is

TABLE 1
Sample File and Content Sizes of Popular Data

Description	File Size	Content Size
CaminoReal Market	62 KB(Map)	4.3 MB
Olive Garden	110 KB (Menu)	8.5 MB
McDonald's	270 KB (Menu)	10.4 MB
menupages.com	120 KB (Menu)	16.9 MB
traffic.com	452 KB (Audio)	10.6 MB

deployed a few miles ahead of the first exit. As the vehicle passes by this infostation, they can download brochures describing local entertainment options and information such as gas stations and shopping malls. Once they download all this information, they could peruse it and decide on their best option. Similarly, in an urban scenario imagine traveling around the city where several infostations are deployed, each providing latest information about services oferred in an area. As users drive by the infostation, their tabletPC receives digital maps of all stores, along with hours, sales, and available parking spots. They could also download graphical menus of restaurants with colorful photos and nutritional data. Again, once they download this data they could decide which of these services they would be interested in.

These types of content are typically organized as a collection or hierarchical list of files, each provide details on an item or location of interest. For example, restaurant menus are lists of descriptions of dishes, while traffic reports are collections of reports each describing a nearby intersections or crossroads. While users normally would interactively browse the list and view specific files, this is not possible for highly mobile users, whose vehicle might be in range of an infostation for only 4-12 s (at highway speeds). Without time to interact with content, user devices must proactively fetch and cache the entire collection so users can browse them at their leisure offline. We found several examples of these data collections, and list them with their total size and size of single files in Table 1. While individual objects are around 100 KB, entire data collections are much larger.

Given the high rates of mobility, the aim of our system is not to guarantee perfect delivery of the entire collection of content (all files), but to maximize the number of complete files delivered to each vehicle to optimize the offline browsing experience. This is in contrast to prior work [17], [18], [19], [20], which focused on the performance of using 802.11b/g unicast to connect devices to fixed access points for *bidirectional* network access, e.g., online web access.

Finally, while some of these types of content are available today via the web, customizing and updating content for a specific location can be cost prohibitive for the provider, and locating such content can be challenging for the user. A local delivery mechanism such as a road-side infostation can be easily maintained by a local tourist office, and can be deployed as needed without huge infrastructure costs.

We explore the design of this system in the context of *Starfish*, an infostation-based content distribution system for location-aware content. Starfish infostations use commodity 802.11 a/b/g hardware, and can be mounted on the side of a street, highway, or freeway. As we later show, a critical challenge for the design of these infostations is that they should work well both when the density of devices is low, e.g., initial adoption, and when the density is high, e.g., when the system becomes popular. Achieving this property is difficult. In the rest of this section, we discuss the design of Starfish infostations and consider unicast or broadcast-based approaches, and their implications on scalability and reliability.

2.2 Starfish with Unicast Infostations

Infostations using unicast can leverage MAC mechanisms like rate adaptation and packet retransmissions to achieve reliable data delivery. In our context, we need an infostation design that maximizes the benefits of rate adaption to reliably transmit the most number of files. Since the infostation can only send to one receiver at a time, it should transmit to the in-range device with which it can establish the best connection. If only a single device is in range, dissemination performance will be determined by the infostation's ability to retransmit lost packets and adapt transmission rate to minimize packet transmission time. But when multiple vehicles are in infostation range, *scheduling* of the packets among multiple vehicles greatly affects the dissemination throughput. Scheduling involves both selecting the next packet to send and the vehicle to send that packet to. As previously observed [21], naively unicasting packets using round-robin to all receivers brings down the throughput of the system to that of the farthest receiver. Clearly, better performance requires an improved scheduling algorithm.

A good scheduling algorithm should consider varying link quality at receivers. If the goal is to maximize the average files delivered per-vehicle (without considering fairness), the infostation should transmit to the in-range device with the best connection. To do so, each starfish infostation maintains an up-to-date *scheduling list* that contains an entry for every device in its radio range. To maintain this list, we use a simple hello protocol where the infostation broadcasts hello packets every t ms. Each device that receives the hello packet unicasts a response to the infostation with the received signal strength embedded inside. The infostation parses hello responses and ranks responding devices in the scheduling list by the embedded signal strength. Whenever it begins to transmit a new file, the infostation examines the devices with the best received signal strength, and makes a change in it's transmit destination if and only if the new device has a considerably better signal strength than the currently scheduled device, e.g., beyond a threshold of $\tau = 2$ dB. Otherwise, it continues to transmit to the current device.

This scheduling design addresses two important issues. First, by making scheduling decisions only at file boundaries, it ensures that all the packets of a file are sent to the same device, and maximizes the probability a given file arrives correctly at its destination. Second, the use of threshold τ increases stability in transmissions so that the rate adaptation has time to ramp up to the maximum sustainable data rate. Otherwise, small fluctuations in signal strength would produce oscillations the choice of devices, significantly limiting the effectiveness of rate adaptation. Overall, this design helps to maximize infostation dissemination throughput using the signal strength as a measure of link quality. As we will show in Section 3, as device density increases, the fraction of time the infostation transmits at the highest data rates also increases, leading to higher throughput.

We note that one can augment this scheduling design to consider fairness among receivers, e.g., by using proportional-fair scheduling that considers both signal strength and average dissemination rate at each receiver.

2.3 Starfish with Broadcast Infostations

Using broadcasts for infostation dissemination has the benefit that per-vehicle throughput is not a function of the number of devices in range. This makes the infostation inherently more scalable. However, for broadcasts, the 802.11 MAC family does not use acknowledgments, packet retransmissions, or rate adaptation mechanisms. As we later confirm using measurements, this produces significant packet losses in our highly mobile environment. In addition, no rate adaptation means that infostation transmissions will be limited to a basic parameterized rate.

The Starfish broadcast infostation design is simple. The infostation iterates over all files, broadcasting all files in order. Any device in range can receive the packet, which could be corrupted due to random propagation effects. Thus, the challenge is to increase resilience to packet losses.

Starfish can leverage several existing techniques to improve resilience in broadcast data delivery, ranging from physical layer techniques like smart antennas to application layer solutions such as forward error correction and source coding [22], [23]. Many of these, however, come at the cost of high complexity and overhead. In this paper, we use a simple but well-known erasure encoding scheme. This simple method divides each file into blocks of m packets, and encodes each block independently into n packets. From our measurements, we observe that packet loss is often bursty (Section 3). Thus rather than broadcasting all packets in each block contiguously, the infostation interleaves encoded packets of multiple blocks.

We choose to augment Starfish broadcast infostations with this simple encoding because of two reasons. First, such simple encoding leads to minimum complexity, allowing us to make a fair comparison between unicast and broadcast infostations. Second, while the above mentioned solutions can improve the resilience, the coding overhead and lack of rate adaptation still limit achievable throughput of broadcast infostations. This motivates us to search for mechanisms beyond traditional infostation-driven methods.

3 INFOSTATION EXPERIMENTS

We performed detailed experiments to evaluate both of our infostation designs, and describe our results here. For the broadcast infostation, we study the loss characteristics of the data dissemination, and motivate the need for application level encoding. Yet even with near-optimal encoding parameters, throughput in a broadcast infostation system is still quite low due to the redundancy required to overcome data loss. Finally, we study the impact of broadcast rate selection. For the unicast infostation, we show the impact of our proposed scheduling algorithm, and quantify scalability limitations as vehicle density increases. In both cases, our metric of interest is the number of complete files successfully received per vehicle.

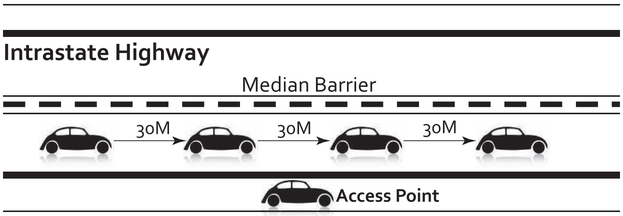


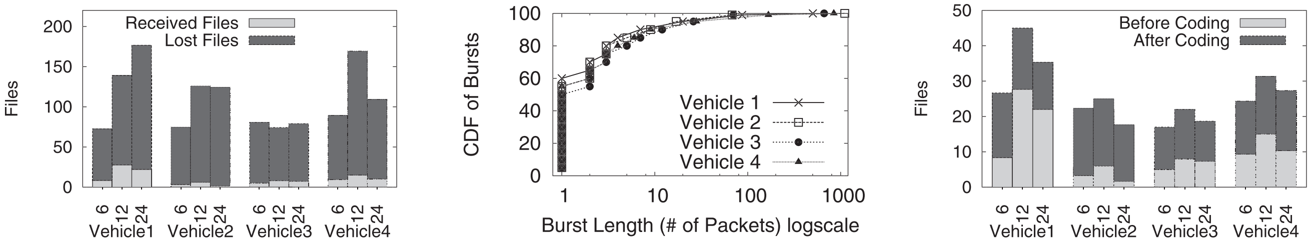
Fig. 1. Our measurement route is a 1.95 mile stretch of highway 217 in Santa Barbara, California, with one static roadside infostation and four vehicle mounted devices.

3.1 Vehicular Testbed

Our testbed consists of five laptops each equipped with a commodity CB9-GP-EXT 802.11 a/b/g WiFi PCMCIA card, which uses an Atheros 5213 chipset, and a rubber duck external omnidirectional antenna mounted on vehicle roof. Each laptop has a GPS unit that records the vehicle's longitude and latitude every second. One device is attached to a parked vehicle acting as the road-side infostation. The others are fixed to a small fleet of four moving vehicles.

In all our experiments throughout the paper, we configure the WiFi cards of the infostation and vehicles to operate in ad hoc mode with a common BSSID and fixed 802.11a channel. We use 802.11a because it is closest to the frequency band (5.9 GHz) allocated for Intelligent Transport Systems in USA and Europe. Our laptops use a modified Atheros reference driver [24]. In broadcast mode, the modified driver allows us to change the broadcast data rate beyond the 802.11a default rate of 6 Mbps. In our experiments, we use broadcast data rates of 6, 12, and 24 Mbps to explore the impact of data rate on transmission reliability and throughput. We chose these specific rates because they use different modulation schemes: BPSK, QPSK, and 16-QAM, respectively. In unicast mode, we use the unmodified proprietary data rate adaptation algorithm in the Atheros reference driver. The modified driver allows the application to directly inject raw Ethernet frames below the TCP/IP stack, and we do not use IP in any of the experiments. This removes issues avoiding the need to configure the IP stack when a receiver comes into range. We argue that bypassing TCP/IP is the right approach for our scenario for the following reasons. First, it has already been shown in previous literature [18] that using the TCP/IP stack is detrimental in VANETs, where short windows of connectivity are prevalent. This is because, making a TCP connection eats up time that could be used toward maximizing data transfer. Second, Starfish's scavenging four uses 802.11's MAC layer unicasts which are fast, reliable, and efficient for our settings. Finally, our experiments assume that the infostation has a large content to disseminate, which is decomposed into many files of 100 KB each. We choose 100 KB because it is representative of file sizes in Table 1. We use 1,500 byte packets with an application payload of 1,464 bytes. Thus, each file is split into 71 packets.

We illustrate the experimental layout of our testbed in Fig. 1. All our experiments are done on a 1.95 mile stretch of highway connecting the UCSB campus west gate to the inter-state highway. A median concrete barrier 1 foot thick and 3 feet high separates the traffic traveling in opposite



(a) File-level Performance without Encoding (b) Burstiness of Packet Loss (c) File-level Performance with Encoding

Fig. 2. Starfish broadcast evaluation. (a) Losses are high for all three broadcast rates. (b) Packet losses are bursty, motivating the need for interleaved coding. (c) The number of files received with and without application level encoding. Encoding does improve the file reception but at a heavy overhead.

directions. The vehicles are on the same side of the road as the infostation. We tried to ensure that, when the vehicles are in the range of the infostation, they maintained a speed of approximately 96 km/h which is the speed limit for the highway. Since the highway is public, external vehicles occasionally entered our experiments. We tried to maintain a distance of approximately 30 m between each vehicle. We completed close to 200 h of driving on the highway, including iterative runs to develop and tune the mechanism of our infostation designs for improved performance. Each experiment is run three times. We show the mean results unless otherwise noted, with error bars showing a single standard deviation.

3.2 Broadcast Evaluation

We first examine the base performance of 802.11 broadcast without application level encoding. Fig. 2a plots the number of complete files received and lost files for each vehicle at different broadcast rates. We count a file as lost if it was transmitted between the first and last received files from the infostation, and was not completely received. We see that 70-90 percent of the files are lost. To isolate the impact of file sizes, we also examine the average raw bytes received by each vehicle, where 11-51 percent of the raw bytes are lost, indicating that losses are distributed evenly across files (results omitted for brevity).

To understand the losses in more detail, Fig. 2b shows the CDF of loss burst lengths across all vehicles for a sample run. Burst length is defined as the number of contiguous packets that were not received. The median burst length is 1, and the 90th percentile is only 9, implying short bursts of packet loss. We see from our traces that long bursts occur during the entry and exit phase of the connection, while short error bursts are distributed across the entire contact window. Since the 802.11 MAC retransmits lost unicast packets up to 16 times, it is not surprising that this loss behavior was unobserved by previous work on unicast systems [18], [20].

Clearly, without reliability mechanisms, 802.11 broadcast provides extremely low throughput across all three broadcast rates. We note, however, the lost files show that potentially achievable throughput is quite high. This motivates us to improve broadcast reliability using application level encoding. The observed short error bursts also suggest block interleaving based encoding.

Impact of application level encoding. We use trace-driven simulations to quantify the impact of different encoding parameters on dissemination throughput. Using gathered packet-level traces, we evaluate the well-known erasure coding scheme (discussed in Section 2.3) with different coding rate (0.1-1) and interleaving depth (1-512). Our results show that the configuration of 0.6 coding rate ($m = 6$, $n = 10$) and depth 8 interleaving works best for all three broadcast rates (we omit detailed results for brevity). Using this encoding configuration, Fig. 2c compares the number of complete files received with and without encoding. We see that encoding does increase reliability and throughput, but comes at the heavy cost of increased overhead.

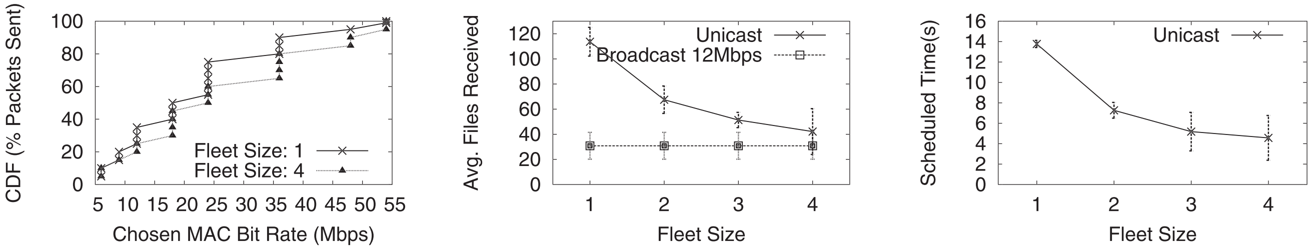
We only evaluated the erasure coding because it is widely used with provable effectiveness. More importantly, all encoding schemes have a fundamental tradeoff between reliability and overhead, which still limits achievable throughput. This motivates us to look for additional mechanisms to improve broadcast reliability without sacrificing throughput.

Impact of broadcast data rate. Without rate adaptation, broadcast infostation must choose a data rate. Results in Fig. 2c show that 12 Mbps outperforms the other two consistently. This is due to two reasons. First, higher broadcast rate leads to shorter contact time with the infostation. In our tests, the average contact times are 11, 9.5, and 4.5 s for 6, 12, and 24 Mbps, respectively. Second, different broadcast rates also map to different packet loss rates—30, 38, and 42 percent, respectively. Accounting for observed contact times and packet losses, 12 Mbps generates the highest effective throughput in our tests. We note, however, even with the optimal broadcast rate, the throughput is still quite limited due to lack of both reliability and rate adaptation.

3.3 Unicast Evaluation

Our evaluation of the unicast infostation design begins with the effectiveness of its transmission scheduling algorithm. Fig. 3a plots the CDF of the rates picked by the infostation for fleet sizes 1 and 4. Curves for fleet sizes 2 and 3 fall between 1 and 4, and are omitted here for clarity. We see that at higher fleet size, the infostation can achieve higher unicast rates. This is because, as density increases, the probability that at least one vehicle is very close to the infostation increases. Hence, our design schedules the nearest device with the highest signal strength, resulting in higher data rates from rate adaptation.

Next, we study the scalability issues of unicast infostation. Fig. 3b shows the average number of files received per vehicle as we vary the fleet size. For comparison, we also plot the broadcast result with 12 Mbps and application



(a) Rates picked by the Infostation (b) Files Received Per-vehicle (c) Scheduled Time Per-vehicle

Fig. 3. Starfish unicast evaluation. (a) Scheduling ensures infostation picks high rates as fleet size increases. (b) and (c) The per-vehicle throughput decreases with vehicles because the scheduled time decreases.

coding. We see that, as the fleet size increases, unicast throughput decreases because the amount of time a vehicle is scheduled to receive data decreases (Fig. 3c). Standard deviations for scheduled times are high, because edge effects increase the scheduled times of vehicles at the edge of the fleet. Finally, the throughput of broadcast infostations is independent of fleet size, but is lower than the unicast infostations for all fleet sizes (1-4).

In summary, our experiments show that although unicast's retransmissions and rate adaptation provide reliability and good throughput, per-device throughput drops significantly when vehicle density increases because an infostation's bandwidth is shared by all vehicles in range. Clearly, better mechanisms are needed to address unicast's scalability limitations.

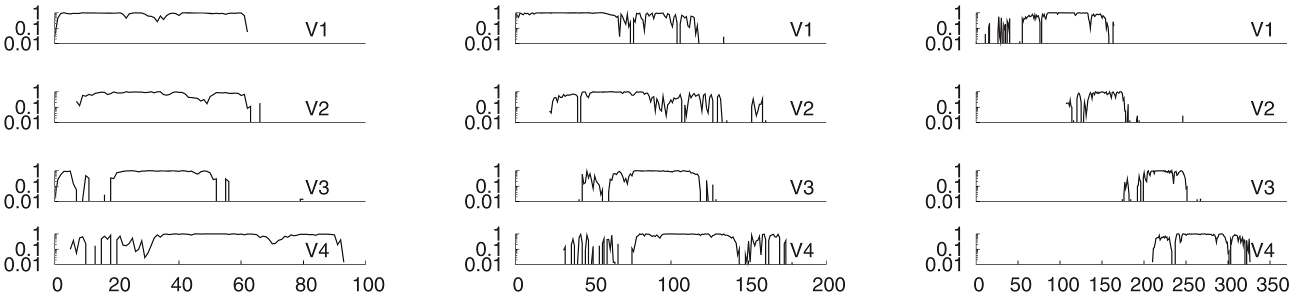
4 STARFISH DATA SCAVENGING

Our measurements show that both broadcast and unicast infostations have their own limitations. Even with advanced infostation design, per vehicle throughput is fundamentally limited by the transmission capacity, and by the number of vehicles in range (in unicast). To address these limitations, we propose to augment infostation bandwidth using bandwidth offered by the devices. We propose *device data scavenging*, where nearby vehicles share data among themselves shortly after passing through an infostation, boosting dissemination throughput. Next, we present the detailed motivation, design, and feasibility study of our proposed solution.

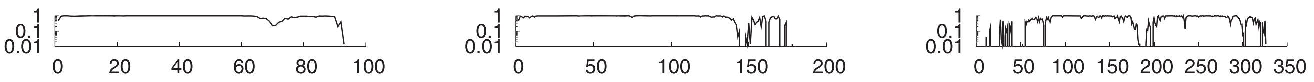
4.1 A Case for Data Scavenging

Our proposal was motivated by the observed packet reception patterns across multiple receivers in a broadcast infostation system. Figs. 4a, 4b, and 4c plot for each broadcast rate the per-device file-level reception rates, i.e., how much of each transmitted file is received. A value of 0.1 means that 10 percent of the file is received. We see that at 6 Mbps, the contact windows of the vehicles show significant overlap, and this overlap decreases at higher data rates. In addition, devices display different loss patterns, which is expected from independent transmission paths to devices.

To explore how this diversity can help improve system throughput, we plot the file-level reception rates when we combine the packets across all the four receivers (Figs. 4d, 4e, and 4f). We make several crucial observations. For broadcasting infostations, significant overlap across neighboring vehicles' contact windows at lower rates means correlated loss is much lower. By exchanging packets, neighboring vehicles can exploit this overlap to recover lost data and improve throughput. At higher broadcast rates, while there is very little overlap, neighboring vehicles can still improve throughput by exchanging packets because they will receive completely new files from each other.



(a) 6Mbps (b) 12Mbps (c) 24Mbps



(d) 6Mbps after combining packets (e) 12Mbps after combining packets (f) 24Mbps after combining packets

Fig. 4. Motivation behind data scavenging. (a)-(c) Per-receiver file level packet reception in a sample run for the three broadcast data rates. (d)-(f) Exchanging packets with neighboring receivers increases both reliability and throughput. In all the figures, x-axis represents the file number of the files transmitted by the infostation.

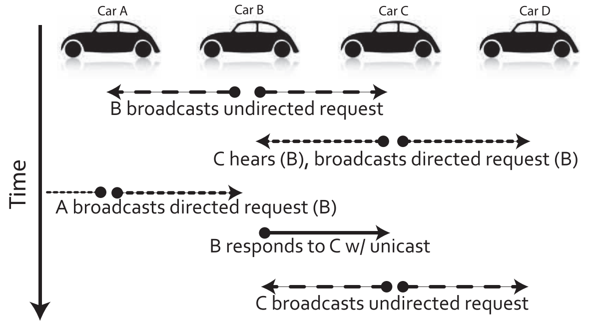


Fig. 5. The starfish scavenging protocol.

Similarly, vehicles in a unicast-based infostation system can also exploit this diversity to improve throughput, by making the scheduling algorithm unicast *different* packets to different vehicles, ensuring no received data overlaps between vehicular neighbors.

We conclude that exploiting receiver diversity between vehicles can significantly improve throughput of both broadcast and unicast infostations. Furthermore, the decrease in correlated loss means the broadcast infostation can reduce the overhead of using high redundancy for reliability. More importantly, a mechanism for exchanging packets between neighboring vehicles can be effective regardless of whether the infostation selects unicast or broadcast.

4.2 Scavenging Design

Motivated by the above observations, we propose *data scavenging*, where nearby vehicles exploit local network connectivity to share data shortly after passing through the transmission range of an infostation. By delaying scavenging until devices exit infostation transmission range, devices can exploit receiver diversity without interfering the infostation's normal operation. Finally, we assume that vehicles do not tamper with packets before forwarding them in the scavenging phase. This can be achieved using existing works on secure vehicular communications [25], [26].

Scavenging enjoys several desirable properties. First, scavenging increases the throughput of the devices by exploiting both *spatial diversity* and *temporal diversity*. It exploits spatial diversity by recovering uncorrelated packet loss across devices. A device that only loses a few packets of a file can decode the file once it obtains those lost packets from other devices. This primarily applies to broadcast infostations. In addition, scavenging exploits temporal diversity by allowing a device to receive a file transmitted when it was out of range. This effectively extends the transmission range of the infostation and is applicable to both unicast and broadcast infostations. Second, scavenging is also fair across devices because devices in a scavenging fleet will receive the same set of files. Finally, scavenging's design and performance is independent of the file size because it is done at packet level rather than file level.

Scavenging protocol. We now describe the data scavenging protocol used in Starfish. Fig. 5 illustrates the scavenging protocol across four vehicles. Data scavenging begins when a device estimates that it is out of range of the infostation. The device periodically broadcasts metadata about both complete files it has received, as well as lost packets it requires to complete other files. Initial beacons sent by a device are *undirected requests* that include its MAC address, a list of completed files, a list of missing data block identifiers, encoding information and a version number. If the file list is long and this information cannot be encoded in a single packet of 2,304 bytes, then the device generates and broadcasts multiple request packets.

A device that receives the undirected request compares the information provided with its locally received files. If a device thinks it can benefit by data from another device, it generates a *directed request*, which is just an undirected request augmented by the MAC address of the device it needs data from. After a random jitter period the packet is broadcast. If, during the jitter period, the device receives another directed hello packet which includes the same destination MAC address then the *directed* request packet is converted to a *undirected* request before broadcast. When the target device receives a directed request with its own MAC address, it responds by unicasting the missing packets back to the requester. The unicast transmission ensures the recovery data is received reliably.

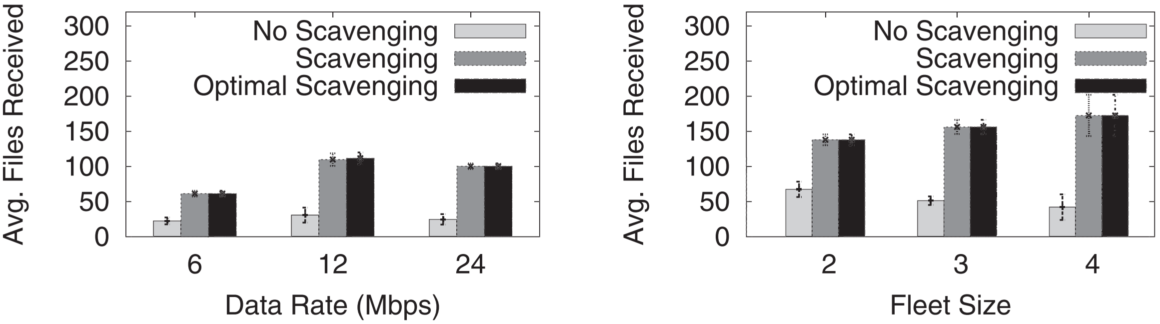
We consider two devices to be "synchronizing" if one of them is sending recovery packets to the other. Once two devices have started synchronization, they cache any broadcast requests received from other nodes, but do not respond or generate new requests. Once synchronization is completed, they review cached requests and attempt to initiate synchronization with other nodes. When a node generates a request (directed or undirected) but receives no response, it simply repeats an undirected request every 500 ms.

While evaluating our protocol on the highway testbed, we found that selecting the right device to synchronize with can dramatically shorten the total time required to synchronize data across a group of vehicles. Thus each Starfish device maintains a *neighbor cache* that stores recent requests. Entries in the cache are ordered by *packetCount* × *signal strength*, where packetCount is the total number of packets that can be recovered from the neighbor. We choose this ranking metric to promote sync sessions that can recover more packets at higher transmission rates, thus reducing the number of sync sessions and their durations. Our experiments confirm that this metric significantly outperforms that using just *signal strength*.

Infostation support. Infostation and data scavenging can be jointly optimized to maximize overall data dissemination. Broadcast infostations can use less coding overhead because scavenging recovers most lost packets. Unicast infostations will unicast different files to different vehicles to minimize overlap. With the SNR-based scheduling, this allows infostations to push a maximum amount of content to a fleet of vehicles, who then use scavenging to receive the entire content at each vehicle.

4.3 Feasibility Study

Data scavenging is most effective when the density of receiver devices around the infostation reaches a minimum threshold. While our experimental testbed consisted of four mobile receivers, we want to determine whether device density in real environments is sufficient to support data



(a) Broadcast

(b) Unicast

Fig. 6. Scavenging increases the number of files received for both broadcast and unicast infostations. Starfish's scavenging almost always reaches optimal scavenging performance. Scavenging for unicast increases with fleet size.

scavenging. To do so, we analyze three vehicular traffic traces gathered on a 640 m segment of the Southbound US Highway 101 in Los Angeles [27]. The traces were each 15 minutes long, and were gathered on June 15, 2005 between 7:50-8:35 AM with relatively high vehicle density. Each trace is a sequence of locations for each vehicle updated once every 100 ms. Average vehicle speeds in the traces are 42, 32, and 29 km/h, respectively. We also consider low-density scenarios, with a vehicle spacing of 61 m, as reported by ElBatt et al. [28].

From our 802.11a measurements, the maximum contact range for a 6 Mbps transmission is 282 m. Analyzing the US101 traffic trace with the highest average vehicle speed shows that at any given time, there are on average 56 vehicles within contact range of an infostation. Therefore, even if only 30 percent of the vehicles had participated, there would always be an average of $15+$ participating vehicles within an infostation's range. Similar conclusions hold for other data rates. Our testbed measurements show a range of 233 m at 12 Mbps and 105 m at 24 Mbps, translating into 46 vehicles in range at 12 Mbps, and 21 vehicles at 24 Mbps. These results for high-density scenarios are supported by other US101 traces and Interstate Highway 80 traces in San Francisco [27]. Also, for low-density scenarios, with a spacing of 61 m [28], our results show that four vehicles can be expected within the range of infostation. Finally, and perhaps more importantly, we note that Starfish's scavenging does not need all the vehicles to be within the range of each other. Since scavenging is a 1-hop gossip based protocol, each vehicle can receive all the packets it needs, as long as the group of vehicles involved are connected. In other words, there exists a multihop path between any two vehicles involved in the scavenging.

TABLE 2
Raw Data (MB) Received Per-Vehicle Before and After Scavenging

	Broadcast (Rate Mbps)			Unicast (Scheduler)	
Options	6	12	24	SNR	RR
Before	5.1	7.1	5.7	4.5	2.2
After	9.4	16.3	18.9	18.1	8.9

For unicast, we compare two schedulers, SNR and RR. For broadcast, we compare three broadcast rates.

5 DATA SCAVENGING EXPERIMENTS

We incorporate device-to-device data scavenging in our Starfish prototype and evaluate its performance in our highway testbed. We perform three test runs at each data rate for the broadcast infostation scenario, and three test runs for different fleet sizes (between 1 and 4) for the unicast infostation scenario. Broadcast infostations use a coding rate of 0.9 (rather than 0.6) because scavenging recovers most lost packets.

Scavenging effectiveness. To understand how well our data scavenging protocol performs in practice, we compare it against an optimal scavenging scheme. The optimal scavenging scheme assumes each device can obtain *any* packet received by other devices in the fleet, thus providing an upper bound on the scavenging performance. Fig. 6 compares, for broadcast and unicast, the number of files received by Starfish devices compared to optimal scavenging. For reference, we also include the result of files that a vehicle receives directly from the infostation assuming no scavenging. We label this as *No Scavenging*.[2] We can see that Starfish's data scavenging is highly effective and reaches the optimal recovery in almost all cases. For both unicast and broadcast scenarios, vehicles in the Starfish system receive significantly more files after scavenging than they directly receive from the infostation. For broadcast with four vehicles, the improvement is about a factor of 3-4, and for unicast, the improvement factor is 2-6.

We also observe that for unicast scenarios, throughput benefits of scavenging increase as the number of vehicles in range increases. This is because of two reasons. First, as explained earlier, as fleet size increases the infostation selects higher rates to unicast (Fig. 3a). Thus it can push more data into the fleet in the same contact window. Second, as fleet size increases, the total amount of time the infostation spends unicasting data to the fleet increases. This results in more files being pushed into the fleet of receivers, thus further increasing throughput.

To isolate the impact of file size, we show in Table 2 the raw bytes received with and without scavenging for four vehicles. We also evaluate scavenging with two unicast scheduling algorithms, signal strength (SNR) and round

---

[2] For *No Scavenging*, we use a coding rate of 0.6 for broadcasting infostations.

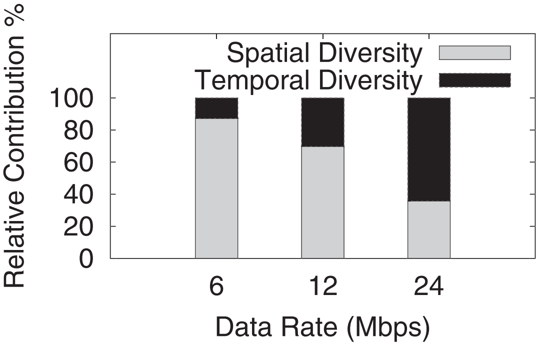


Fig. 7. Diversity in broadcast: contribution from spatial diversity decreases and that from temporal diversity increases with broadcast data rate, because of reduction in contact overlap between neighboring vehicles.

robin (RR). The results again confirm that data scavenging improves dissemination throughput by up to a factor of 4.

Spatial versus temporal diversity. To understand how scavenging benefits from both spatial and temporal diversity, we plot the percentage of new files decoded due to temporal and spatial diversities for the broadcast scenario. From Fig. 7, we see that as expected, spatial diversity dominates at lower rates, e.g., 6 Mbps, because of significant overlap between contact windows of neighboring vehicles. At higher data rates, temporal diversity begins to dominate as overlap between neighboring vehicles decreases.

Scavenging time. A key metric is the amount of time required for vehicles to perform data scavenging. Scavenging times should be short enough so that the fleet of receivers does not disperse before scavenging completes. We measure "scavenging time" as the duration between when a receiver device first sends a directed request and when it receives its last data packet from a neighbor.

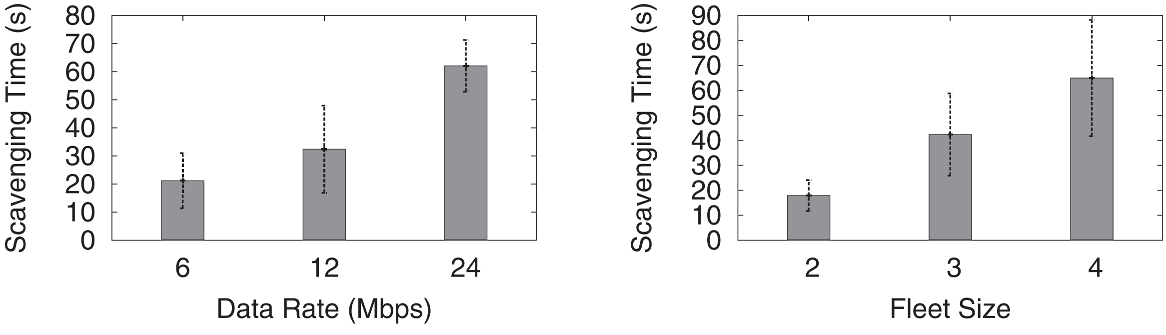
Fig. 8a plots average scavenging times of each vehicle at each broadcast rate. As expected, scavenging time increases with the broadcast rate. We also make several other observations. First, Starfish at 6 Mbps has a low scavenging time (≈ 18 s), because vehicles only need to scavenge a relatively small number of files due to the larger overlap in their contact windows, and because of the lower transmission rate. Second, although the number of absolute files scavenged is similar for 12 and 24 Mbps, Starfish with 24 Mbps has a longer scavenging time. This is because dissemination at 24 Mbps experiences higher temporal diversity and lower spatial diversity (see Fig. 7). So a vehicle needs to potentially synchronize with more vehicles to get its files, and different vehicles block while waiting for their neighbors to finish their synchronization sessions. The impact of these factors is further amplified by contentions among multiple concurrent unicast sessions in recovery, leading to longer times. Fig. 8b shows that scavenging time for unicast increases with the number of vehicles in range. This is because as the fleet size increases, unicast infostations transmit more files to the fleet, and vehicles need more time to scavenge all files from neighbors.

Overall, scavenging time in both scenarios depends on how many total files are sent to the group of vehicles, and the level of temporal diversity. Our field tests show that at 96 km/h, data scavenging for small vehicular groups (four vehicles) can complete in roughly 1 minute (1.6 km traveled). This means that data scavenging can be highly effective even for relatively short-lived vehicular clusters.

6 LARGE-SCALE EVALUATION

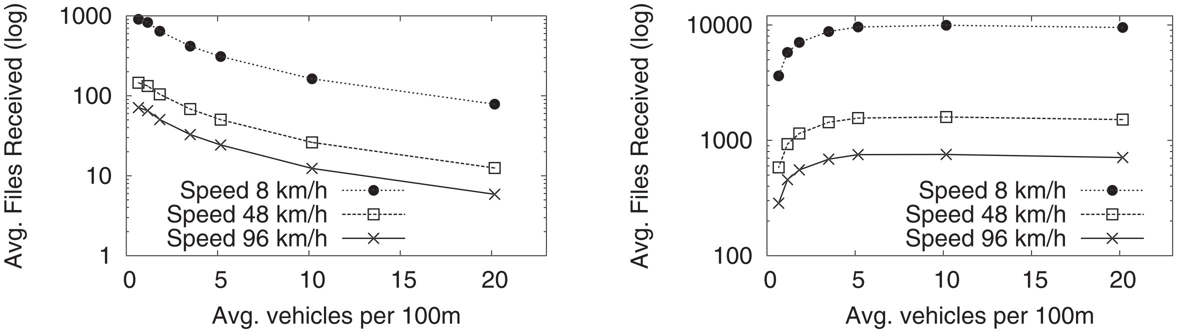
Our highway experiments show that Starfish with data scavenging is feasible, and can greatly improve disseminating throughput for both broadcast and unicast infostations. But several key questions remain: How does Starfish perform as density and speed vary? If a service provider wants to deploy infostations, should they use broadcast or unicast? Which approach performs better and under what conditions? Answering these questions requires large-scale experiments, so we use QualNet simulations to analyze Starfish performance under various system parameters.

Preliminaries. We implemented both Starfish unicast and broadcast infostations in Qualnet. To model channel characteristics observed in our outdoor experiments, we calibrate propagation parameters in Qualnet using our experimental traces. While replicating the exact wireless channel of the experiments is difficult, we found that a 2-ray propagation model with log-normal shadowing (mean = 2.4) and Ricean fading (K-factor = 50) provides transmission range and loss rates at 6 Mbps similar to our testbed results. The model deviates slightly at higher data rates, but with negligible impact and without affecting our conclusions. Our simulations predict scavenging results using packet traces recorded by individual vehicles, and assume that scavenging always



(a) Broadcast (b) Unicast

Fig. 8. For broadcasts, scavenging time increases at higher data rates because of increased temporal diversity. For unicasts, scavenging time increases with fleet size as more files are transmitted by the infostation to the group.



(a) Before Scavenging (b) After Scavenging

Fig. 9. Before data scavenging, unicast throughput decreases with density because the scheduled time per vehicle decreases. After scavenging, throughput increases with density because the infostation unicasts at higher rates.

completes in time after they receive data from the infostation. Thus, our results here focus on the optimal performance possible for data scavenging.

We configure our simulations to investigate changes in performance across a wide range of vehicle density and speed parameters. For our simulations, we use a single lane road 2 km long, 3.5 m wide. An infostation lies on the side of the road at the half-way point. A "fleet" of evenly spaced vehicles drive by the infostation at a uniform constant speed and receive common files. Varying fleet sizes and using nonuniform vehicle spacing produced similar results, which we omit for brevity. The distance between the first and last vehicles (length of the fleet) is 600 m. In our experiments, we vary spacing between vehicles from 5 m to 200 m, and their speed from 8 km/h to 96 km/h. With a fixed fleet length, lowering intervehicle spacing effectively increases the number of vehicles in the fleet. For simplicity, we assume all vehicles in the fleet participate in data scavenging, and any two consecutive vehicles are within transmission range of each other. Finally, we define vehicle density as the number of vehicles per 100 m.

We report *before scavenging* and *after scavenging* values. Before scavenging is the number of complete files a vehicle receives *directly* from the infostation. After scavenging includes those plus all files scavenged from other vehicles. All results are per-vehicle values averaged over the number of vehicles in the fleet of each experiment.

6.1 Unicast Infostations

For unicast infostations, we implemented the device scheduling mechanism (from Section 2.2) on top of ARF [29] rate adaptation in QualNet. We modified some ARF parameters to improve unicast performance in the vehicular environment: the number of successful acks required to step up the transmission rate to 4, the number of unsuccessful acks to step down the transmission rate to 2, and the rate adjustment timer to 1 ms. While we cannot match the proprietary rate adaption scheme in our testbed Atheros driver, we believe that our high-level observations and conclusions are still valid. This is because, though absolute performance numbers might vary with different rate adapation schemes, we argue that the relative performance of unicast and broadcast would still follow the patterns observed in our experiments. We leave the comparison of different rate adaptation schemes to future work.

Before scavenging. We examine baseline performance of unicast infostations with a hello message interval of 200 ms. Fig. 9a shows that as density increases, average per-vehicle throughput decreases as infostation bandwidth is shared by more vehicles. Assuming signal strength decreases monotonically with distance, we can roughly estimate the amount of time a vehicle gets scheduled (by having the best signal strength in the fleet) as *spacing/speed*. This is confirmed by our experiments, and explains the trend in Fig. 9a where per-vehicle throughput scales inversely with vehicle spacing and speed. We will show later how the hello interval also impacts unicast performance.

After scavenging. Fig. 9b shows that dissemination throughput after scavenging increases significantly with vehicle density, similar to those observed in our testbed experiments. As explained in Section 5, this is because as vehicle density increases, the infostation benefits from more multiuser diversity and therefore a greater portion of its unicast transmissions are at higher rates.

Impact of hello interval. One interesting trend that did not appear in the testbed experiments is that at 96 km/h, throughput starts to drop at high densities (> 5 vehicles per 100 m). A closer look showed that this is due to the impact of hello interval configuration. Recall that the unicast infostation broadcasts periodic hello messages to solicit signal strength reports from vehicles, which it uses for scheduling. Closer examination shows that the frequency of hello messages, if not chosen carefully, can impact unicast performance in two ways: *contention* and *suboptimal scheduling*.

While increasing the frequency of hello messages increases responsiveness to mobility, it can also cause *contention* at high vehicle densities. Many hello response messages can contend for time at the infostation, reducing time available for data transmissions, increasing delays for hello responses and causing them to be lost. Lost hello responses then lead to suboptimal scheduling of data transmissions. To quantify the impact of contention on throughput, we repeat our simulation and configure QualNet to eliminate hello response contention by having vehicles send hello responses via wired connections. Fig. 10 compares the throughput with and without contention using a 100 ms hello interval. Without contention, a unicast system converges to a maximum throughput (reached when nearly all transmissions from the infostation are at

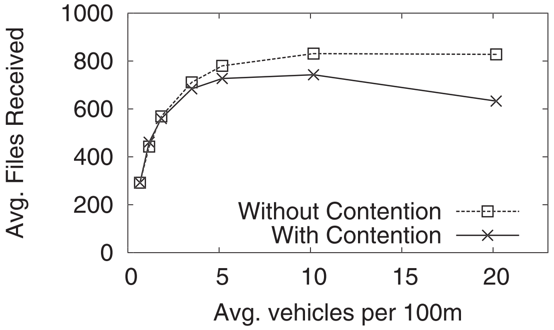


Fig. 10. Unicast after scavenging at 96 km/h with hello interval of 100 ms. Throughput drops at high densities due to hello response contention.

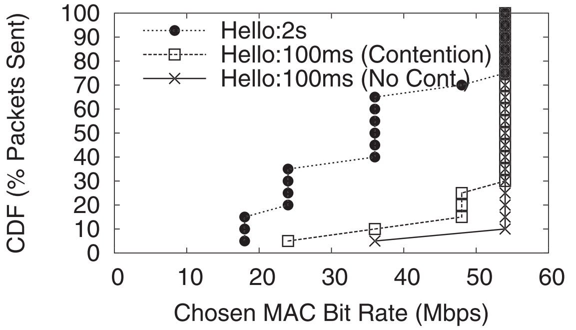


Fig. 11. As hello interval increases, infostation operates at lower unicast rates due to suboptimal scheduling (with 5 m vehicle spacing and 96 km/h vehicle speed).

the highest data rate). With contention, throughput drops up to 25 percent as vehicle density increases.

If we choose too low a hello frequency, we would expect lower performance due to poor scheduling of receivers by the infostation. We examine the impact of hello response frequency on rate adaptation in Fig. 11. We show the distribution of MAC bitrates chosen for each packet as a CDF, for three different scenarios: long intervals (2 s), short interval (100 ms), and short interval (100 ms) with contention disabled. All vehicles are traveling 5 m apart at 96 km/h. As expected, the larger interval (2 s) produces poor choices for receivers, thus reducing MAC bitrates for many packets. But at high densities (5 m separation), frequent hello messages (100 ms) causes contention, which delays hello responses and causes suboptimal receiver choices. Thus, the hello interval must be chosen carefully. Too low, it is unable to adapt to vehicle dynamics. Too high, hello response contention will delay responses and also reduce responsiveness. In this paper, we use experimental tests to locate the best values for a given combination of speed and fleet density. At all vehicle densities, we find that 200 ms works well for fast vehicles traveling at 96 km/h, and 2 s works well for vehicles at 8 km/h. In real world scenarios, we could imagine that the infostation gathers the speed and density information from the vehicles nearby (e.g., through beacons from vehicles) and sets the appropriate hello interval. For this to work, an infostation should have access to a model which maps the speed and density of vehicles to the appropriate hello interval. This model could be generated apriori based on a small set of measurements or could be dynamically learned and adapted over time by the infostation.

TABLE 3
Broadcast Infostation Performance Before Scavenging, for Vehicles Traveling at 96 km/h

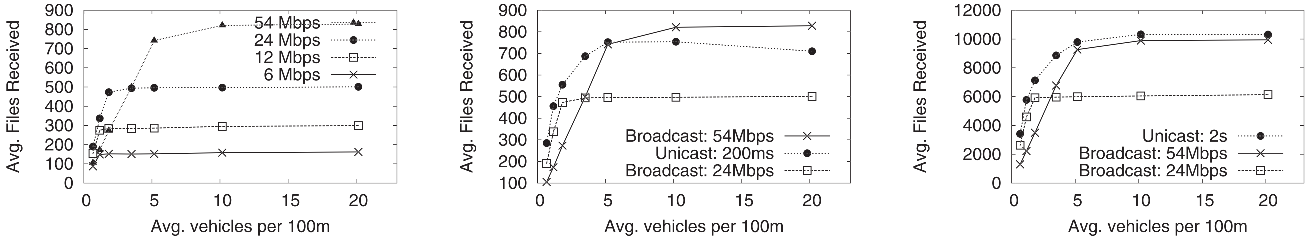
Broadcast Rate (Mbps)	Range (m)	LossRatio (@ 96 km/h)	Avg Per-Vehicle Files Received
6	226	32%	21.6
12	203	33%	38.6
24	120	29%	46.3
54	30	21%	24.5

6.2 Broadcast Infostations

Our evaluation focuses on the impact of vehicle density and data rates on overall throughput. We examined performance at different vehicle speeds, but limit our discussion to high (96 km/h) and low (8 km/h) speeds for brevity. To improve reliability, all files are encoded with 0.9 coding rate, as it performed the best in earlier tests.

Before scavenging. Table 3 shows the before scavenging performance in terms of infostation transmission range, packet loss ratio, and complete files received per vehicle. As expected, infostation range drops quickly from 226 m to 30 m as the broadcast rate increases. Packet loss rate, however, also drops with the rate, which differs from our experimental results. We believe this is due to the differences in wireless channel in the simulator and testbed. Overall, broadcasting at 24 Mbps provides the best per-vehicle results, and on average, a vehicle receives less than 50 files.

After scavenging. Fig. 12a shows that data scavenging significantly improves dissemination throughput. When broadcasting at 24 Mbps, each vehicle receives on average 200-500 files, a factor of 4-10 improvement over the < 50 files received without scavenging. As vehicle density increases, temporal diversity increases and vehicles in the



(a) Broadcast after scavenging (b) Unicast vs Broadcast @ 96 km/h (c) Unicast vs Broadcast @ 8 km/h

Fig. 12. (a) For each broadcast rate, scavenging dramatically improves throughput initially due to temporal diversity, then spatial diversity, and converges when spatial diversity is maximized. (b) and (c) Unicast outperforms broadcast at lower densities due to scheduling and rate adaptation, but drops at higher densities because of hello packet contention.

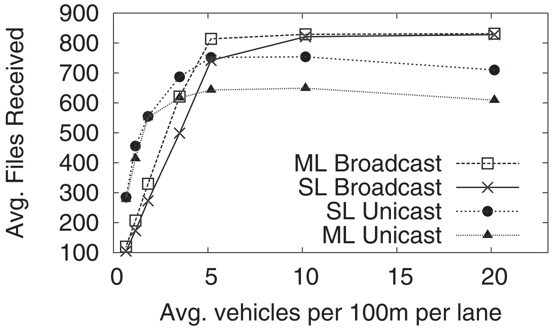


Fig. 13. Multilane experiments increase vehicles in range, improving broadcast performance and degrading unicast performance.

fleet recover many more files by scavenging from others. Broadcasting at 54 Mbps performs the best, raising the gain of scavenging up to a factor of 30.

One interesting observation is that, for each broadcast rate, there exists a "knee" density value, after which dissemination throughput flattens quickly. This point lies around a density of 2 vehicles/100 m for 24 Mbps and around 10 vehicles/100 m for 54 Mbps At this density, scavenging shifts from temporal diversity to spatial diversity (see Section 4). Before the knee, when temporal diversity dominates, scavenging improvement is proportional to vehicle density. After the knee, spatial diversity decreases the impact of packet loss, and throughput gradually flattens out.

6.3 Infostations: Broadcast versus Unicast

Finally, we compare in Figs. 12b and 12c the performance of unicast and broadcast infostations with scavenging. For broadcast, we use transmission rates of 24 and 54 Mbps because they performed the best in earlier tests. For unicast, we choose the hello interval that gives the best performance for each speed, e.g., 200 ms for 96 km/h and 2 s for 8 km/h. In practice, an infostation can adapt this parameter by having vehicles embed their speed in responses to hello messages. Our results show that, up to a density of 5 vehicles/100 m, unicast always outperforms broadcast. At higher densities, unicast performance falls below that of broadcast at 96 km/h. We note that this is an unlikely scenario, since this combination of high vehicle density and high speed is hard to achieve (and dangerous) in practice. At 8 km/h, unicast outperforms broadcast at all densities because hello intervals are larger and contention is not a factor.

Multilane results. To understand the impact of multiple lanes, we simulate a three-lane highway with 12-feet wide lanes and 96 km/h traffic. All the other parameters are kept the same as before. Note that for a given intervehicle spacing, there will be thrice as many vehicles in the multilane (ML) setup compared to single-lane (SL). We super-impose the curves from single lane for comparison and plot the results in Fig. 13. Since more vehicles are in infostation range in a multilane scenario than a single-lane scenario, the potential degree of spatial diversity is also higher. Thus, the performance of broadcast infostations improves (see Fig. 13). Unicast infostations, however, experience the opposite. At higher densities, having more vehicles means higher contention and consequently lower per-vehicle throughput.

6.4 Summary and Discussion

Both broadcast and unicast infostations can benefit from data scavenging. Broadcast infostations at lower data rates benefit from higher spatial diversity, and those at higher data rates benefit from temporal diversity. In comparison, unicast infostations obtain better performance because the infostation transmits longer at higher data rates. But at very high densities and vehicle speeds, hello protocol message contention reduces system throughput. At reasonable speeds and vehicle densities, infostations using well-designed unicast protocols will perform the best.

We can further optimize unicast infostations to limit the impact of hello response contention. For example, the infostation can embed a minimum RSSI threshold in hello messages so that vehicles with lower RSSI values suppress their responses. Or vehicles can keep brief RSSI history windows, and suppress their responses if their RSSI values are dropping consistently, i.e., they are moving away from the infostation. Recent work [30] considers using multihop beacon relays to reduce contention. We will explore the feasibility and effectiveness of these optimizations in future work.

7 RELATED WORK

Measurement studies [19], [20] have demonstrated the feasibility of connecting infostations with passing vehicles, and that considerable throughput can be achieved. Most of the subsequent studies of infostation-assisted data dissemination systems focus on single vehicle drive-through scenarios, while Starfish examines the dissemination performance when a fleet of vehicles drives by an infostation. Starfish is among the first to study this complex scenario via real measurements. In the following, we discuss related work organized by the data dissemination scenario considered.

Single infostation, single vehicle. For vehicles performing TCP downloads from open WiFi networks, existing work [17], [31] shows that considerable packet loss occurs even during the production phase. The authors propose a new transport protocol to improve reliability, focusing on distinguishing wired and wireless losses. Hadaller et al. [18] focused on understanding the protocol interactions during downlink TCP data transfers. They identify that the connection setup delay is due to timeouts at different layers, lengthy access point selection and unsuitable bit rate adaptation. In their recommended best practices, they suggest avoiding exit and entry phases altogether in order to compensate for TCP timeouts. The Cabernet study [31] found that 802.11b 11 Mbps performed better than all 802.11g rates in urban settings. Starfish differs from these works by considering common content dissemination to a fleet of vehicles.

Multiple infostations, single vehicle. Similar to our testbed results, other measurements [11] show that short bursts of non-connectivity periods (gray periods) can occur anywhere during a vehicle's contact with an infostation. To minimize the impact of such nonconnectivity, subsequent work ViFi [10] utilizes the fact that a vehicle's packet transmissions are independent across nearby infostations, and proposes a hand-off protocol to maintain continuous connectivity using multiple infostations. Starfish is complementary to [10], and differs in both the challenges addressed as well as the mechanisms used for reliability. While ViFi focuses on maintaining reliable connections to infostations for per-vehicle interactive applications, Starfish seeks to maximize infostation dissemination throughput by combining packets across nearby vehicles. And instead of assuming the presence of multiple infostations within a vehicle's contact range, Starfish utilizes the diversity of nearby vehicles to improve data dissemination.

When neighboring APs are interconnected, prior work has applied network coding to disseminating popular data from multiple APs to each vehicle [32], [33]. In contrast, Starfish assumes that infostations are not interconnected and may disseminate information independently, i.e., they can be deployed by different service providers or companies.

Single infostation, multiple vehicles. Hadaller et al. [21] identify a drastic reduction in unicast throughput when multiple vehicles access an infostation. They propose a new scheduling algorithm that grants infostation access to vehicles based on SNR. Starfish's scheduling algorithm for unicast is also based on SNR, but its goal is to disseminate more files into a fleet of receivers for better scavenging performance. Another work [34] proposes to relay packets among a cluster of vehicles to extend infostation unicast range, which is equivalent to exploiting the temporal diversity discussed earlier. Starfish, on the other hand, exploits both the temporal and spatial diversity to improve dissemination performance.

After vehicles download data from an infostation, prior works [35], [13], [15], [16] have proposed gossip, BitTorrent-like protocols, or network coding-based packet exchange among vehicles to further populate the data. However, all of these systems are based on network simulations. They are either designed on top of the wireless transport layer, or assume reliability through retransmissions at the wireless MAC layer. In contrast, Starfish examines the issue of disseminating and scavenging common data to a fleet of vehicles via real experiments, focusing on high mobility and high loss environments.

Vehicle-to-vehicle data dissemination. The majority of prior work in this area are driven by simulation and theoretical analysis [36], [37], [38], [39], [15], [16]. In contrast, we verify the feasibility and efficiency of V2V communication by implementing and experimenting with data scavenging on a real highway testbed.

A recent work proposed Broadside [24], a system that targets file exchange between two vehicles. Via real experiments, this work demonstrates the feasibility and efficiency of high-throughput file exchange among moving vehicles. Starfish is motivated by this finding, but differs in both the challenges addressed and the application scenario.

Finally, many studies [13], [14], [40], [16], [41] have shown the advantages of using network coding in disseminating popular data among multiple nodes. We believe that network coding is complementary to Starfish's scavenging and can be integrated into our design to further reduce the scavenging delay.

8 CONCLUSION

This paper presents the measurement-based design of an infostation-based content delivery system, where vehicles passing by a roadside infostation proactively fetch and cache content for offline browsing. We analyze extensive data gathered from nearly 200 h of on-road measurements, and find that both unicast- and broadcast-based systems fail to scale as vehicle density increases. In addition, we observe that data dissemination exhibits both temporal and spatial diversity across vehicles.

We propose to exploit this diversity via data scavenging, where vehicles passing by the infostation share data between them to both recover from wireless loss and extend the effective range of the data transmission. Through detailed experiments, we show that our scavenging techniques achieves close to the optimal predicted performance, and improves overall dissemination throughput by up to a factor of 6.

REFERENCES

[1] http://www.starbucks.com/coffeehouse/mobile-apps/mystarbucks, 2012.

[2] http://www.mcdonalds.com/us/en/services/free_wifi.html, 2012.

[3] http://www.starbucks.com/coffeehouse/wireless-internet, 2012.

[4] http://news.cnet.com/8301-30686_3-10451819-266.html, 2012.

[5] http://mobile.venturebeat.com/2010/06/22/, 2012.

[6] A.L. Iacono and C. Rose, "Infostations: New Perspectives on Wireless Data Networks," technical report, WINLAB, Rutgers Univ., 2000.

[7] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting Mobile 3G Using WiFi," *Proc. ACM MobiSys,* 2010.

[8] P. Deshpande, X. Hou, and S. Das, "Performance Comparison of 3G and Metro-Scale WiFi for Vehicular Network Access," *Proc. ACM SIGCOMM,* 2010.

[9] Tropos Networks, "Price-Performance Comparison: 3G and Tropos Metromesh Architecture," http://www.tropos.com/pdf/whitepapers, 2009.

[10] A. Balasubramanian, R. Mahajan, A. Venkataramani, B.N. Levine, and J. Zahorjan, "Interactive WiFi Connectivity for Moving Vehicles," *Proc. ACM SIGCOMM,* 2008.

[11] R. Mahajan, J. Zahorjan, and B. Zill, "Understanding WiFi-Based Connectivity from Moving Vehicle," *Proc. ACM SIGCOMM,* 2007.

[12] K. Ramachandran, M. Gruteser, R. Onichi, and T. Hikita, "Experimental Analysis of Broadcast Reliability in Dense Vehicular Networks," *Proc. Int'l Symp. Wireless Vehicular Comm. (WiVec),* 2007.

[13] S. Ahmed and S. Kanhere, "VANETCODE: Network Coding to Enhance Cooperative Downloading in Vehicular Ad-Hoc Networks," *Proc. Int'l Conf. Wireless Comm. and Mobile Computing (IWCMC),* 2006.

[14] S. Deb, M. Médard, and C. Choute, "Algebraic gossip: A Network Coding Approach to Optimal Multiple Rumor Mongering," *IEEE Trans. Information Theory,* vol. 52, no. 6, pp. 2486-2507, June 2006.

[15] K. Lee, S.-H. Lee, R. Cheung, U. Lee, and M. Gerla, "First Experience with CarTorrent in a Real Vehicular Ad Hoc Network Testbed," *Proc. Conf. Mobile Networking for Vehicular Environments (MOVE),* 2007.

[16] U. Lee, J. Park, J. Yeh, G. Pau, and M. Gerla, "Code Torrent: Content Distribution Using Network Coding in VANET," *Proc. First Int'l Workshop Decentralized Resource Sharing in Mobile Computing and Networking (MobiShare),* 2006.

[17] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden, "A Measurement Study of Vehicular Internet Access Using Unplanned 802.11 Networks," *Proc. ACM MobiCom,* 2006.

[18] D. Hadaller, S. Keshav, T. Brecht, and S. Agarwal, "Vehicular Opportunistic Communication Under the Microscope," *Proc. ACM MobiSys,* 2007.

[19] R. Gass, J. Scott, and C. Diot, "Measurements of In-Motion 802.11 Networking," *Proc. IEEE Seventh Workshop Mobile Computing Systems and Applications (HotMobile),* 2006.

[20] J. Ott and D. Kutscher, "Drive-THRU Internet: IEEE 802.11b for Automobile Users," *Proc. IEEE INFOCOM,* 2004.

[21] D. Hadaller, S. Keshav, and T. Brecht, "MV-MAX: Improving Wireless Infrastructure Access for Multi-Vehicular Communication," *Proc. ACM SIGCOMM,* 2006.

[22] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A Digital Fountain Approach to Reliable Distribution of Bulk Data," *Proc. ACM SIGCOMM,* 1998.

[23] M. Mitzenmacher, "Digital Fountains: A Survey and Look Forward," *Proc. IEEE Information Theory Workshop,* 2005.

[24] T. Zahn, G. O'Shea, and A. Rowstron, "Feasibility of Content Dissemination Between Devices in Moving Vehicles," *Proc. Fifth Int'l Conf. Emerging Networking Experiments and Technologies (CoNEXT),* 2009.

[25] Q. Xu, T. Mak, J. Ko, and R. Sengupta, "Vehicle-to-Vehicle Safety Messaging in DSRC," *Proc. First ACM Int'l Workshop Vehicular Ad Hoc Networks (VANET),* 2004.

[26] M. Raya and J. Hubaux, "Securing Vehicular Ad Hoc Networks," *J. Computer Security,* vol. 15, pp. 39-68, 2007.

[27] NGSIM HomePage. http://ngsim.fhwa.dot.gov, 2012.

[28] T. ElBatt, S.K. Goel, G. Holland, H. Krishnan, and J. Parikh, "Cooperative Collision Warning using Dedicated Short Range Wireless Communications," *Proc. Third ACM Int'l Workshop Vehicular Ad Hoc Networks (VANET),* 2006.

[29] A. Kamerman and L. Monteban, "WaveLan-II: A High-Performance Wireless LAN for the Unlicensed Band," *Bell Labs Technical J.,* vol. 2, pp. 118-133, 1997.

[30] J. Mittag, F. Thomas, J. Härri, and H. Hartenstein, "A Comparison of Single- and Multi-Hop Beaconing in VANETs," *Proc. Sixth ACM Int'l Workshop Vehicular Ad Hoc Networks (VANET),* 2009.

[31] J. Eriksson, H. Balakrishnan, and S. Madden, "Cabernet: Vehicular Content Delivery Using WiFi," *Proc. ACM MobiCom,* 2008.

[32] H. Lu, F. Wu, and C. Chen, "Stateful Scheduling with Network Coding for Roadside-to-Vehicle Communication," *Proc. IEEE Int'l Conf. Comm. (ICC),* 2009.

[33] Z.Y.M. Li and W. Lou, "CodeOn: Cooperative Popular Content Distribution for Vehicular Networks using Symbol Level Network Coding," *IEEE J. Selected Areas in Comm.,* vol. 29, no. 1, pp. 223-235, Jan. 2011.

[34] J. Zhao, T. Arnold, Y. Zhang, and G. Cao, "Extending Drive-Thru Data Access by Vehicle-to-Vehicle Relay," *Proc. Fifth ACM Int'l Workshop Vehicular Ad Hoc Networks (VANET),* 2008.

[35] A. Nandan, S. Das, G. Pau, M. Gerla, and M.Y. Sanadidi, "Co-Operative Downloading in Vehicular Ad-Hoc Wireless Networks," *Proc. Second Ann. Conf. Wireless On-Demand Network Systems and Services (WONS),* 2005.

[36] T. Nadeem, P. Shankar, and L. Iftode, "Comparative Study of Data Dissemination Models for VANETS," *Proc. Third Ann. Int'l Conf. Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous),* 2006.

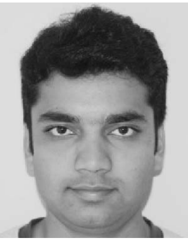
[37] M. Nekoui and H. Pishro-Nik, "Fundamental Tradeoffs in Vehicular Ad Hoc Networks," *Proc. Seventh ACM Int'l Workshop Vehicular Ad Hoc Networks (VANET),* 2010.

[38] H. Wu, R. Fujimoto, and G. Riley, "Analytical Models for Information Propagation in Vehicle-to-Vehicle Networks," *Proc. IEEE 60th Vehicular Technology Conf. (VTC Fall),* 2004.

[39] J. Zhao and G. Cao, "VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks," *Proc. IEEE INFOCOM,* 2006.

[40] C. Gkantsidis and P. Rodriguez, "Network Coding for Large Scale Content Distribution," *Proc. IEEE INFOCOM,* 2005.

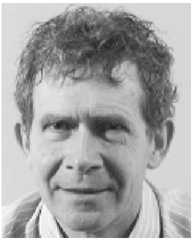
[41] J.Z. Yang Zhang and G. Cao, "Roadcast: A Popularity Aware Content Sharing Scheme in VANETs," *ACM SIGMOBILE Mobile Computing and Comm. Rev.,* vol. 13, pp. 1-14, 2009.



Vinod Kone received the BTech degree from the Indian Institute of Technology Guwahati, in 2006, and the MS and PhD degrees in computer science from the University of California, Santa Barbara, in 2010 and 2011, respectively. Currently, he is a software engineer in the Runtime Systems group at Twitter. His research interests include network measurements, wireless networks, and distributed systems.

Haitao Zheng received the BS degree (with highest honors) from Xian Jiaotong University in 1995 and the MSEE and PhD degrees in electrical and computer engineering from the University of Maryland, College Park, in 1998 and 1999, respectively. Currently, she is an associate professor of computer science at the University of California, Santa Barbara. Previously, she worked at Bell Labs and Microsoft Research Asia. Her research interests include wireless systems and networking and multimedia computing. She is a recipient of *MIT Tech Review*'s TR-35 Award, the Bell Labs President's Gold Award, and the George Harhalakis Outstanding Graduate Student Award.

Antony Rowstron received the MEng and DPhil degrees in computer science from the University of York, United Kingdom. He is a principal researcher and leads the Systems and Networking Group at Microsoft Research, Cambridge, United Kingdom. Prior to working for Microsoft Research, he worked at the Computer Laboratory and Engineering Department of Cambridge University, United Kingdom. His research interests include the spectrum of systems, distributed systems, and networking. He is a fellow of the British Computer Society.



Greg O'Shea is a research software development engineer in the Systems and Networking Group at Microsoft Research, Cambridge, working on the implementation and evaluation of experimental networked systems.

Ben Y. Zhao received the BS degree from Yale University, Branford, 1997, and the MS and PhD degrees in computer science at the University of California, Berkeley, in 2000 and 2004, respectively. He is an associate professor of computer science at the University of California, Santa Barbara. His research spans the areas of networking, security and privacy, distributed systems, simulation and modeling. His recent interests include online social networks, cognitive radios, and wireless data centers. He is a recipient of the US National Science Foundation's CAREER award, *MIT Tech Review*'s TR-35 Award, and is one of *ComputerWorld*'s Top 40 Technology Innovators.