On the Fidelity of 802.11 Packet Traces*

Aaron Schulman, Dave Levin, and Neil Spring

Department of Computer Science
University of Maryland, College Park
{schulman,dml,nspring}@cs.umd.edu

Abstract. Packet traces from 802.11 wireless networks are incomplete both fundamentally, because antennas do not pick up every transmission, and practically, because the hardware and software of collection may be under provisioned. One strategy toward improving the completeness of a trace of wireless network traffic is to deploy several monitors; these are likely to capture (and miss) different packets. Merging these traces into a single, coherent view requires inferring access point (AP) and client behavior; these inferences introduce errors.

In this paper, we present methods to evaluate the fidelity of merged and independent wireless network traces. We show that wireless traces contain sufficient information to measure their *completeness* and *clock accuracy*. Specifically, packet sequence numbers indicate when packets have been dropped, and AP beacon intervals help determine the accuracy of packet timestamps. We also show that trace completeness and clock accuracy can vary based on load. We apply these metrics to evaluate fidelity in two ways: (1) to visualize the completeness of different 802.11 traces, which we show with several traces available on CRAWDAD and (2) to estimate the uncertainty in the time measurements made by the individual monitors.

1 Introduction

Studying wireless networks "in the wild" gives researchers a more accurate view of 802.11 behavior than simulations alone. Researchers deploy monitors at hotspots such as cafes or conferences [10], or measure other deployed networks [1], to obtain traces of MAC and user behaviors. These traces provide realistic models of mobility [11, 18] and interference [1, 3] and many traces are readily available through sites such as CRAWDAD [7].

However, traces of real wireless networks have their own errors or assumptions. Indeed, capturing a high-quality wireless trace requires great care. Using too few monitors, placing them poorly, or using inadequate hardware can introduce missed or reordered packets and incorrect timestamps [10, 16, 17]. If multiple monitors are used, a *merging* algorithm combines the independent traces into a single view of the wireless network [10], but this process may order

* This work was supported by NSF-0643443 (CAREER). Dave Levin was supported in part by NSF Award CNS-0626964 and NSF ITR Award CNS-0426683.

packets incorrectly. These potential errors mean that publicly available wireless traces vary greatly in quality (§5). Researchers must decide for themselves which wireless trace will provide them the most accurate, reproducible results.

We consider the problem of measuring the *fidelity* of wireless traces, which we decompose to their *completeness*—what fraction of the packets that could have been captured in fact were—and the *accuracy of their timestamps*. Our work is motivated by others' observations on how to use and improve the data that drives the networking community. As Paxson [12] notes, it is beneficial to identify how closely a measurement compares to reality before using it as experiment data. Haeberlen et al. also observe that researchers may fall into the trap of inappropriately generalizing their results if based on very specific or perhaps error-ridden data [8]. The difficult nature of capturing wireless traces further motivates a set of metrics and systematic means of measuring their quality.

We discuss how wireless trace fidelity can be measured by exploiting information in the trace (§3); external validation data is rarely available. We analyze a scoring method for wireless traces (§4). The percent of packets captured has been thought to be sufficient for quantifying a trace's fidelity, but we show that a richer description of fidelity is important and propose a way to visualize trace completeness that incorporates load (§5). We present several case studies from the CRAWDAD repository. We then study the accuracy of monitor and beacon timestamps, showing that clock accuracy is largely inversely proportionate to load and that clocks may need to be synchronized more frequently than at beacon intervals (§6). We conclude with lessons learned and directions for future work (§7). http://www.cs.umd.edu/projects/wifidelity holds our code and results.

2 Related Work

Because wireless traces are imperfect, many researchers have sought to improve trace fidelity. Yeo et al. [16, 17] and Rodrig et al. [14] discuss the steps they took to obtain high-fidelity traces, and use missing packets (§4) as a measure of fidelity. We focus on the relationship between trace quality and load on the monitor, and compare existing traces using our metrics.

Wit [10] attempts to refine existing traces by inferring and inserting missing packets. We believe traces that are as complete as possible at the time of capture are preferable, but that more complete traces will help the missing packet inference. Our tools are intended to help guide researchers toward capturing better traces and choosing the trace that best suits their needs.

Wireless traces are used for many reasons: to validate models of wireless behavior, study usage characteristics, and so on. Jigsaw [4, 5] uses wireless traces to measure and troubleshoot wireless networks. We emphasize that these pieces of work evaluate the *network*, and not the trace. We expect our work to complement these and other similar projects as pathologies in the input trace data could easily lead to false diagnosis by troubleshooting tools.

3 Self-Evident Truths of Wireless Traces

Ideally, one could determine a trace's fidelity by comparing it to "truth": a perfect, complete trace of what was sent and when. In practice, only the trace itself is available. We show how the information in a wireless trace itself can be used to measure the trace's fidelity by detecting missed packets and measuring clock skew, and discuss the limitations of our methods.

3.1 Core data in wireless traces

Traces vary in the information they include. Some traces have timestamps precise to nanoseconds, others only to milliseconds; not all traces record 802.11 acknowledgments; to maintain users' anonymity, few researchers release full payloads, and so on [13, 15]. The following data are available in all 802.11 CRAWDAD traces; we assume them as the *core* data that are likely to be available in future wireless traces:

1. All types of data packets.
2. All types of management packets including beacons, probe requests, and probe responses.
3. Full 802.11 header in all captured packets, including source and destination addresses (possibly anonymized), sequence number, retransmission bit, type, and subtype. Beacon packets also have timestamps applied by the AP.
4. Monitor's timestamp (set by the kernel or possibly the device).

3.2 Detecting missed packets

Monitors can fail to capture a packet because the monitor is overloaded, because there is interference and perhaps no stations receive the packet, because the signal is too weak at the monitor, and so on (Fig. 1). A common practice to reduce the number of missed packets is to place each monitor near an AP.

Most packet loss at the monitor can be inferred from 802.11 sequence numbers and the retransmission bit. When initially transmitted, each host (AP and client) assigns a packet a monotonically increasing sequence number from 0 to 4095 (or 2047 in some Cisco APs), and sets the retransmission bit to zero. One sign of missed packets is a gap in captured sequence numbers from a given host. Another sign of missed packets is a retransmitted packet without the corresponding first (non-re)transmission.

Missed retransmissions are more difficult to infer. Upon retransmission, the packet's sequence number remains unchanged, but the retransmission bit is set to one; future retransmissions of this packet are identical, which means that not all retransmissions can be inferred. If 802.11 acks and accurate timestamps are available, some of these retransmissions could be inferred. For instance, if a monitor captures an ack that is too late to correspond to any captured retransmission, we could infer that there must have been another retransmission. We do not consider this approach further, since not all traces contain acknowledgments.

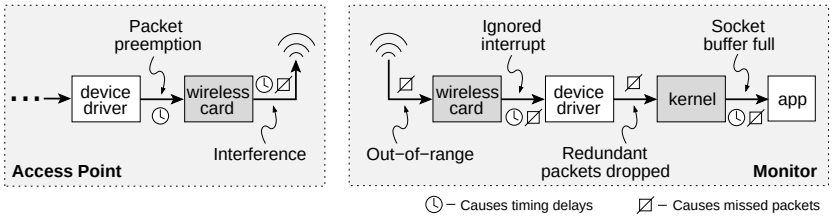

Fig. 1. Example sources of packet loss or timing errors in capturing wireless traces.

3.3 Detecting incorrect timestamps

Monitors apply a timestamp to every packet in the kernel or possibly in the wireless device itself. The accuracy of these timestamps is vulnerable to delay at the AP and clock skew or clock drift at the monitor. Delay at monitors can come for many reasons, some of which we show in Fig. 1.

Beacon packets serve as a source of "truth" in that they allow us to synchronize the monitor's clock [5, 10]. However, this introduces its own sources of inaccuracy; timestamps in the beacon packets are subject to delay errors at the AP. Delay at the AP comes predominately in times of high load. When it is time to send a beacon packet, the AP creates the payload (including the timestamp), and attempts to send it. The timestamp in the beacon packets denotes when the packet was *created*, not necessarily when it was *sent*. Under high load, the packet may be stalled until the medium becomes free [2], increasing the difference between the packet's timestamp and when it was actually sent.

4 Scoring a Wireless Trace's Completeness

We propose a method to *score* wireless trace completeness. We value *completeness*—the fraction of packets captured—with the expectation that the more complete a trace is, the more useful it is. In the following section, we use our score along with traffic load to visualize completeness.

4.1 Estimating the number of missed packets

Our scoring method is based on the number of missing packets from the wireless trace. This is an extension of what was introduced by Yeo et al. [16]. We define $\mathcal{P}_t$ to be the number of packets that should have appeared over time t.

$$\mathcal{P}_t \stackrel{\text{def}}{=} \sum_{nodes} \text{SeqNumChange}_t + \sum_{nodes} \text{Retransmissions}_t$$

The number of missing packets during time t, $\mathcal{M}_t$, is the number of packets that should have been captured minus the number of packets that were captured:

$$\mathcal{M}_t \stackrel{\text{def}}{=} \mathcal{P}_t - \sum_{nodes} \text{NumPacketsCaptured}_t$$

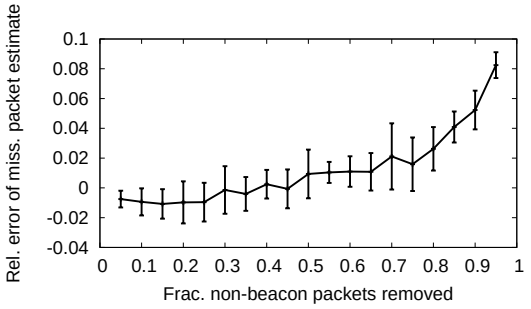

Fig. 2. Validation of our missing packets estimation. Starting with a high-quality trace (the Portland State University ug trace [13]), we remove non-beacon packets uniformly at random. Error bars represent 95% confidence intervals.

To evaluate the accuracy of this expression, we apply it to traces that we intentionally degrade. Starting with a high-quality trace (the Portland ug trace), we created progressively lower-quality traces by removing non-beacon packets uniformly at random and computed our score on these degraded traces (we expect monitors to capture most beacon packets: §5). We present the error of our missing packets estimation in Figure 2. Ideally, our method would detect all of these removed packets, but it is impossible to detect missing retransmission packets without 802.11 acknowledgments (§3). Even with a drastically degraded trace missing 95% of non-beacon packets, our score underestimates actual packet loss by only 10%. For more reasonable packet loss, our score has less than 5% error. These results indicate that *this method of detecting missing packets is accurate for both high- and low-quality traces.*

4.2 Score definition

We define the *score* of a wireless trace's completeness during time t, $\mathcal{S}_t$, as the fraction of packets captured during time t: $\mathcal{S}_t \stackrel{\text{def}}{=} 1 - \frac{\mathcal{M}_t}{\mathcal{P}_t}$. Both APs and clients increment an independent sequence number for each unique packet transmitted. The technique used to reveal missing packets sent by an AP can do the same for clients. Unlike APs, clients do not transmit beacon packets at a regular interval. We must therefore be careful to keep track of how long it has been since the monitor last received a packet from a given client, so as to distinguish loss from, say, mobility. Our scoring method is subject to the same limitations as the missing packet estimation; the score cannot identify missing retransmissions.

5 Visualizing Wireless Trace Completeness

Trace completeness is an important component of fidelity. Rodrig et al. [14], for example, have used the percent of packets captured, similar to the score from

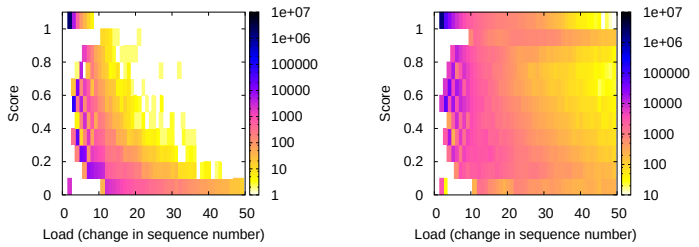

Fig. 3. Example T-Fi plots from the Sigcomm 2004 "chi" dataset, with scoring for only the AP (left), and scoring for APs and clients in a BSS (right).

§4, but we find a single number to be insufficient. This is in part because trace quality can depend on load. A monitor may appear to capture a high percentage of packets, and one may be inclined to use that percentage to quantify the quality of a trace, but this number is misleading. For example, the Sigcomm 2004 trace "chi" contains 81% of AP data and management transmissions on channel 11. This percentage does not reveal that 37% of the packets collected were beacon packets sent when the AP was idle; not sending any other data or management packets. Excluding beacon packets sent during otherwise idle times, the monitor only saw 70% of the AP's transmissions.

5.1 T-Fi plots

To overcome this problem, we visualize the score with a colormap, as shown in Figure 3. We refer to the colormaps as *T-Fi* or *Trace Fidelity* plots. The x-axis denotes the load from an epoch (beacon interval) in terms of the sequence number change during that epoch, and the y-axis denotes the score for that load. Color intensity denotes how often that (x, y)-pair occurred throughout the trace. The T-Fi plot displays these trace features:

1. The location on the y-axis shows completeness.
2. The width of the shaded region on the x-axis shows the range of load.
3. The intensity of the shaded region shows the frequency of load.

An ideal trace would have no missing packets and therefore a score of 1; in our visualization, this corresponds to a dark bar only at the top of the graph (the closest example of this is the Portland UG trace in Fig. 4).

Fig. 3 (left) shows how the single number problem can be overcome with a T-Fi plot. The darkest point on the plot is in the upper left hand corner. The upper left hand corner (sequence number change 1 and score 1) represents idle time beacon packets sent from an AP. The number of beacon intervals in this trace that fell in this region is 100 times larger than any other region in the plot. This would dominate a simple percentage, but is relegated to a small, clear region of the T-Fi plot. For load between 30 and 50, the trace scores no greater than 0.1, indicating low fidelity under high load. Indeed, Fig. 3 (left) shows a negative correlation of fidelity to load.

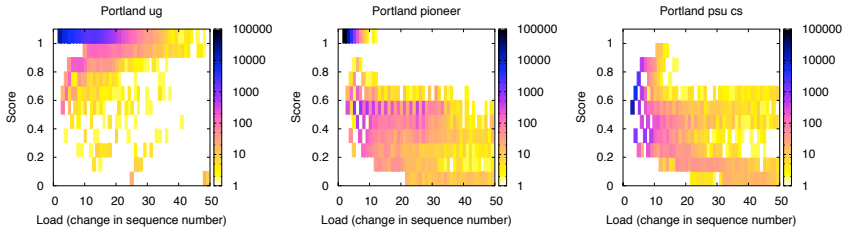

Fig. 4. Trace completeness visualization for Portland PDX traces [13].

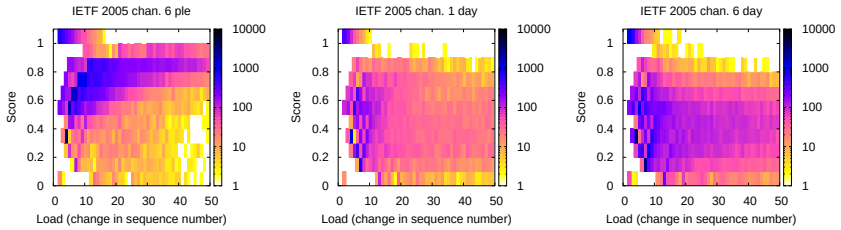

Fig. 5. Trace completeness visualization for IETF 2005 conference traces [9].

5.2 Case studies

We analyzed the completeness of several traces obtained from CRAWDAD using the T-Fi visualization. We show two sets of traces: the Portland PDX VWave dataset and traces collected during the 2005 IETF meeting. Monitors from these traces may have captured unintended traffic from outside sources. The T-Fi plots shown in Figs. 4 and 5 are filtered to show only the BSS with the highest traffic.

Portland PDX traces show how specialized 802.11 monitor equipment can improve trace quality. Phillips et al. [13] used a VeriWave WT20 commercial wireless monitor to capture their traces. VeriWave has a hardware radio interconnect to provide real time merging with 1 microsecond synchronization accuracy. UG has the best combination of high score and load. UG's T-Fi plot has a wide shaded region scoring 1 covering load values 1 to 40. This trace is close to complete and contains both high and low load epochs; Fig. 3 (left) represents a comparatively incomplete trace.

The pioneer trace (Fig. 4 center) was captured from an outdoor courtyard. Even with powerful monitor hardware, the monitor missed many packets in the pioneer trace. The trace contains a wide range of load values (1 to 50) but rarely scored above 0.5 in higher load epochs. Evidently, the pioneer trace is missing packets independently of the load. We believe the clients and AP captured by the trace were out of range or the monitor was receiving interfered signals. The psu-cs T-Fi plot (Fig. 4 right) has few dark-colored regions, indicating that there was low load on the network.

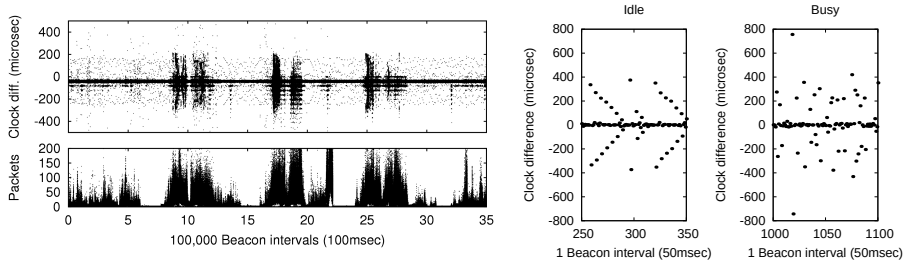

Fig. 6. Difference in monitor timestamps and beacon timestamps for the Sigcomm'04 "chi" trace (top left), with the load shown (bottom left). A controlled experiment with 50msec beacon intervals without load (middle) and with (right).

IETF 2005 traces exhibit high score variability under any given load. A load that scores consistently is represented in a T-Fi plot by a column that has only a few dark bars close together. This can be seen at sequence number change 40 on the T-Fi plot of "chan 6 ple" in Fig. 5. If the score varies greatly for a sequence number change the column will consist of similar colored bars; "chan 1 day" shows this behavior between sequence number changes 10 and 40.

The traces captured during the plenary sessions are of higher quality than the day sessions, showing the apparent effects of mobility on trace completeness. T-Fi plots of the day traces in Figure 5 do not score as highly as the plenary trace. For example, the plenary session traces score higher in high bandwidth epochs. We posit that the day traces scored lower in high bandwidth epochs because clients are mobile during the day. During the plenary sessions, the meeting participants were likely to be stationary more often than in the day traces.

6 Timestamp Accuracy

The accuracy of a trace's timestamps is important for many applications; merging algorithms [5, 10], for instance, use monitor and beacon timestamps to form a single, coherent view of the wireless network as viewed from potentially many monitors. A common assumption in these algorithms is that the difference between a monitor's timestamp—stamped in the kernel or the device itself—and the AP's timestamp—included in the beacon packet—is predictable and consistent on at least the order of beacon intervals (100msec).

We test this hypothesis by observing the difference between monitor timestamp and beacon timestamp over time throughout a trace. For the Sigcomm'04 trace (Fig. 6 left), we plot the clock difference (top) and the load in number of packets captured (bottom). The clock difference is not consistent from one beacon interval to the next, indicating that there is clock skew at the monitor and/or the AP. To see whether the clock difference was at least consistent *within* a given beacon interval, we collected our own trace using the MeshTest testbed [6] with a beacon interval of 50msec. When no clients are sending data (Fig. 6 middle),

the clock difference does change *between* normal (100msec) beacon intervals, but in what appears, in this case at least, to be a predictable manner. However, when a client is sending (Fig. 6 right), the clock changes are not predictable, again indicating a correlation of clock difference with load.

These results show that *the common assumption underlying known merging algorithms is false*. The question remains whether this is sufficient to cause a mis-ordering of packets. Though we have observed mis-orderings from Wit [10], it is unclear whether this is due to an algorithmic error or simply a bug in Wit. Nonetheless, we propose as a sanity check that merging algorithms ensure proper sequence number order (not necessarily strictly increasing: §7).

7 Discussion

We considered the problem of quantifying wireless trace fidelity and evaluated a scoring method, proposed the T-Fi visualization, and presented an analysis of clock accuracy in wireless traces. Wireless trace fidelity applies when choosing, improving, or inferring gaps in wireless traces.

Choosing a trace. Researchers will choose traces from a repository like CRAWDAD based primarily on the type of data in the trace, for example mobility or traffic type. However, we expect fidelity to decide which trace—or subset of the trace—to use.

Improving traces. Measuring trace fidelity need not be strictly a post-mortem analysis; rather, researchers ought to measure the fidelity of their measurements *during* their measurement, so that they may, for example, move their monitors. An interesting and important area of future work is to develop tools to aid in the active capture of wireless traces, so that researchers can ensure high-fidelity traces in unique hotspots such as a conference.

We conclude with lessons we learned about merging and processing wireless traces in the process of working with as many traces as we could collect.

Update tools in accordance with new specs. Tools to measure the fidelity of wireless traces must be updated frequently, as new 802.11 specs are deployed. The 802.11e QoS amendment introduced a new sequence number space for QoS in mid-2006. This did not turn up in our initial testing on the Sigcomm'04 trace, but did in the Portland traces (late 2006), and we had to adjust our tool accordingly.

Account for vendor-specific behavior. Some vendors introduce behavior not specified in 802.11, and this may make the trace appear to be of lower fidelity. We observed that the Cisco access point in the Sigcomm'04 trace assigned sequence numbers to broadcast and multicast packets, then transmitted the packets after others were sent, causing some sequence numbers to appear out of order. To account for this, we allowed these packets to appear out of order in sequence number.

Acknowledgements. We thank Justin McCann and the anonymous reviewers for their helpful comments, Brenton Walker and Charles Clancy for allowing us to use the MeshTest testbed, and Ratul Mahajan for supporting Wit.

References

1. D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurements from an 802.11b mesh network. In *SIGCOMM*, 2004.
2. ANSI/IEEE. Std 802.11, 1999.
3. S. Biswas and R. Morris. Opportunistic routing in multi-hop wireless networks. In *SIGCOMM*, 2005.
4. Y.-C. Cheng, M. Afanasyev, P. Verkaik, P. Benkö, J. Chiang, A. C. Snoeren, S. Savage, and G. M. Voelker. Automating cross-layer diagnosis of enterprise wireless networks. In *SIGCOMM*, 2007.
5. Y.-C. Cheng, J. Bellardo, P. Benkö, J. Chiang, A. C. Snoeren, G. M. Voelker, and S. Savage. Jigsaw: Solving the puzzle of enterprise 802.11 analysis. In *SIGCOMM*, 2006.
6. T. Clancy and B. Walker. MeshTest: Laboratory-based wireless testbed for large topologies. In *TridentCom*, 2007.
7. CRAWDAD Website. http://crawdad.cs.dartmouth.edu/.
8. A. Haeberlen, A. Mislove, A. Post, and P. Druschel. Fallacies in evaluating decentralized systems. In *IPTPS*, 2006.
9. A. Jardosh, K. N. Ramachandran, K. C. Almeroth, and E. Belding. CRAWDAD data set ucsb/ietf2005 (v. 2005-10-19). Downloaded from http://crawdad.cs.dartmouth.edu/ucsb/ietf2005, Oct. 2005.
10. R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Analyzing the MAC-level behavior of wireless networks in the wild. In *SIGCOMM*, 2006.
11. W. Navidi and T. Camp. Stationary distributions for random waypoint models. *IEEE Transactions on Mobile Computing*, 3(1), 2004.
12. V. Paxson. Strategies for sound Internet measurement. In *IMC*, 2004.
13. C. Phillips and S. Singh. CRAWDAD data set pdx/vwave (v. 2007-08-13). Downloaded from http://crawdad.cs.dartmouth.edu/pdx/vwave, Aug. 2007.
14. M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan. Measurement-based characterization of 802.11 in a hotspot setting. In *E-WIND*, 2005.
15. M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, J. Zahorjan, and E. Lazowska. CRAWDAD data set uw/sigcomm2004 (v. 2006-10-17). Downloaded from http://crawdad.cs.dartmouth.edu/uw/sigcomm2004, Oct. 2006.
16. J. Yeo, S. Banerjee, and A. Agrawala. Measuring traffic on the wireless medium: Experience and pitfalls. Technical report, CS-TR 4421, University of Maryland, College Park. Available at http://hdl.handle.net/1903/124, Dec. 2002.
17. J. Yeo, M. Youssef, and A. Agrawala. A framework for wireless LAN monitoring and its applications. In *WiSE*, 2004.
18. J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *INFOCOM*, 2003.