Real and Simulator Testbeds for Content Dissemination in High-density Large-scale WANET

Yu-Jen Lai*, Youyang Ng*, Takeshi Sakoda*, Yosuke Bando*†, Arata Miyamoto*,
Masahiro Ishiyama*, Ken-ichi Maeda*, Yusuke Doi*

*Toshiba Corporation, Tokyo, Japan †MIT Media Lab, Cambridge, MA, USA
{yujen.lai, youyang.ng, takeshi1.sakoda, yosuke1.bando, arata.miyamoto, masahiro.ishiyama, ken.maeda, yusuke.doi}@toshiba.co.jp

***Abstract*—A wireless ad hoc network (WANET) is an effective approach to disseminating digital content without the need for infrastructure networks. It is not only useful in the absence of commodity connections such as cellular networks, but also useful for avoiding overloading those connections in crowded scenarios, an example of which is delivering movie clips throughout spectators in a stadium. However, experiments of WANET have been oftentimes conducted using only up to 1,000 nodes in a simulator or dozens of real devices, both typically scattered across a wide area at a density of around 10^{-4} node/m^2.**

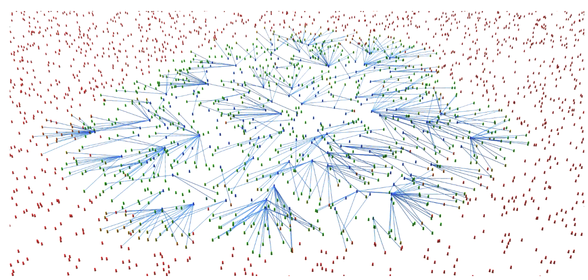
This paper focuses on the problem of disseminating common content to all the devices in the same area via 802.11-based WANET, and presents a pair of testbeds, one based on real devices and the other a simulator, both capable of high-density and large-scale experiments with roughly matched performance. We have found naive implementation of content dissemination protocols alone not to work in high density, and we identify necessary adjustment of wireless parameters to allow for experiments even with an extreme density of 20 nodes/m^2. We show that 2.5 MB of data can be delivered to 100 real devices and to 10,000 simulated devices in a few minutes on our testbeds, demonstrating the feasibility of orders of magnitude larger-scale experiments.

I. Introduction

Dissemination of digital content using wireless ad hoc networks (WANET) offers promising applications without the aid of network infrastructures. It is not only useful in extending the coverage of commodity connections such as cellular networks and Wi-Fi hotspots, but also effective in offloading traffic from those connections in cases where a crowd of people gathered in the same place have location-specific common interests. For example, daily news and traffic reports could be pushed to commuters, video clips of fine game play to spectators in a stadium, course materials to students in a class, or a map and waiting time list of attractions to theme park visitors.

While many content dissemination protocols in WANET have been proposed, the scale (i.e., the number and density of nodes) of evaluation has often been relatively small compared with the above scenarios, either due to the target use cases of those protocols or simply because of computational or installation cost. Typically, up to 1,000 simulated nodes [1]–[4] or dozens of real devices [5]–[7] are scattered across a wide area at a density of around 10^{-4} nodes/m^2. This is in contrast with trains (up to 5 nodes/m^2 when highly crowded), stadiums (2 nodes/m^2), classrooms (0.5 node/m^2), or theme parks (0.01 node/m^2), where 100-10,000 people can gather.

(a) 100-node real-device testbed.



(b) 10,000-node simulator testbed.

Fig. 1. Real and simulator testbeds capable of disseminating common content from a single node to all the other nodes in high-density large-scale scenarios. With identical experimental conditions and parameters, the performance of the two testbeds roughly matches within a factor of two. The dots in (b) show nodes (red waiting for content and green finished receiving it), while the blue lines show wireless links.

In this paper, we build a pair of testbeds, one based on real devices (Fig. 1a) and the other a simulator (Fig. 1b), both geared to content dissemination in large-scale (i.e., high-density and many-node, not necessarily wide-area) WANET. As a first step, we focus on the problem of delivering common content to all the devices in the same area via 802.11-based WANET as shown in Fig. 2, in which a source node disseminates content data to all the nodes nearby in a multi-hop fashion. We have found that, in high density scenarios, content dissemination can be prohibitively slow because of tremendous number of control packets preventing data transmission, indicating that naive implementation of content dissemination protocols alone is not sufficient. We identify necessary adjustment of wireless settings that allows us to conduct experiments even with an extreme density of 20 nodes/m^2. While there are a few reports of experiments involving a large number of

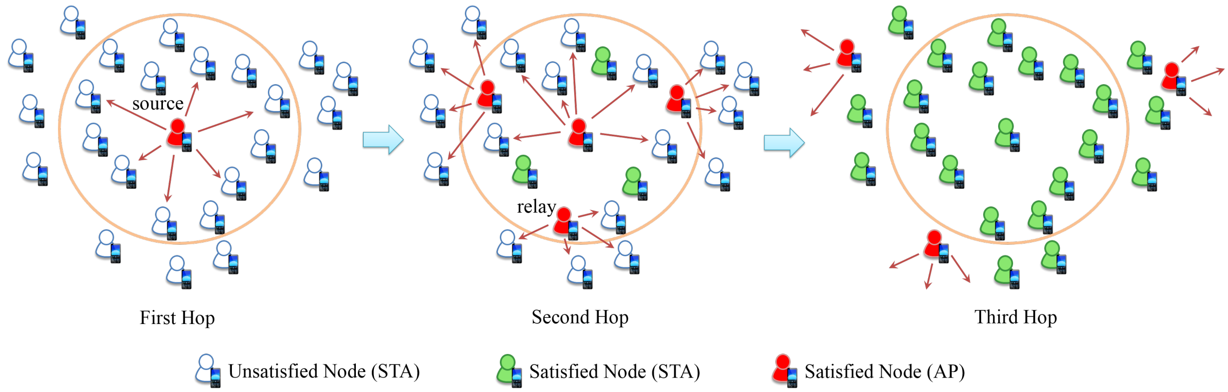


Fig. 2. The target scenario of content dissemination. A source access point (AP, red) with content to share starts content dissemination to the nearby stations (STA) in a single hop, and then a few satisfied STAs (green) become relay APs and forward the content to other unsatisfied STAs in a multi-hop fashion.

either real nodes [8] or simulated nodes [9], to the best of our knowledge, testbeds that can handle up to 100 real devices and 10,000 simulated devices with roughly matched performance capable of delivering content with such high density have not been reported in the literature.

To demonstrate the capability of our testbeds, we conduct performance evaluations at densities ranging from 0.01 to 20 nodes/m^2 using reference content dissemination protocols. With the best combination of these protocols and parameters, 2.5 MB of data can be delivered to 100 real devices in 115 seconds and to 10,000 simulated devices in 190 seconds.

The rest of the paper is organized as follows. Section II presents related work. Section III explains our target scenario of content dissemination. Section IV explains the details of the two testbeds and the high-density solution implemented inside. Section V explains the reference content dissemination protocols we implement on both testbeds. Section VI presents example performance evaluations. Section VII presents discussions. Finally, Section VIII concludes the paper.

II. Related Work

Content dissemination via WANET is a hot research topic because of its promising applications in many fields [10]–[15]. Many research works choose simulator as an evaluation tool because of the cost and difficulties associated with real-world wireless environments. Existing works on large-scale WANET simulation typically use up to 1,000 nodes. For example, Goyal presented performance comparison of routing protocols using a 60-node simulator [4], Zhou et al. studied an IP address allocation problem using 250 nodes [1], Gottron et al. investigated voice communication in 500-node WANET [3], and Ahvar and Fathy conducted 1,000-node experiments for evaluating energy consumption [2]. While high numbers of nodes such as 100,000 can be found in the literature [9], all of these experiments assume 1-250 km^2 simulation areas and the densities are in the order of 10^{-4} nodes/m^2.

On the other hand, running experiments on real devices is also necessary to understand real-world wireless behaviors, as noted by Kotz et al., who collected data from 40 laptops and fed it into their simulator [16]. Other works also typically use several dozens of real devices. Subramanian et al. tested their UFlood protocol on 25 nodes placed in three floors of a building [7]. Chambers created an experimental mesh network testbed called Roofnet consisting of 40 nodes installed on the roofs of buildings in an urban area [5]. Owada et al. built a 50-node testbed on a university campus [6]. All of them are sparsely distributed at densities of around 10^{-4} nodes/m^2 or less. A notable exception is ORBIT, a well-known radio grid testbed consisting of 400 nodes with 1 m spacing (i.e., 1 node/m^2) that can be dynamically interconnected into specified topologies [8]. See Appendix for a summary of node densities found in the previous work.

The difference of our real and simulated testbeds from these previous works is that we aim for high-density environments where the order of density ranges from 0.01 to 10 nodes/m^2, so as to be able to evaluate content dissemination through a crowd of user devices. To reliably achieve this density level, we use 100 and 10,000 nodes for real-world and simulated experiments, respectively, which are larger than most of the existing testbeds. Following Kotz et al. [16], our simulator testbed uses parameters taken from the real-device testbed, and the two testbeds perform consistently up to 100-node experiments. We note that density is a relative metric especially on simulators, and existing testbeds can likely emulate high-density scenarios by increasing transmission power, which has not been attempted because their respective research objectives are different from ours.

III. Target Scenario

This section explains our target scenario for content dissemination in our testbeds as shown in Fig. 2. To establish a WANET without network infrastructures, 802.11 ad hoc mode is a natural choice. However, due to its vulnerable security, lack of high bit-rate support, and less availability [17], the use of 802.11 infrastructure mode, which is supported by most consumer devices, is suggested and demonstrated in [18], [19]. We follow this approach here.

More specifically, in our scenario, an access point (AP) acts as a sender, whereas a station (STA) acts as a receiver. At the beginning, one node owning content data becomes a *source* AP while other nodes are running as STAs. The source AP then broadcasts MAC-layer beacons with an SSID filled with an identifier of content dissemination process. If an STA has not yet received the content (i.e., *unsatisfied*), it captures the beacon from the source AP and joins the local network created by the AP. Then, the AP starts to deliver the content to the STAs. STAs can upgrade to *relay* APs to propagate the content further after they finished receiving the content (i.e., *satisfied*), so that the content can be disseminated in a multi-hop fashion. In this study, we deal with presumably the simplest scenario where all the nodes are stationary and there is one source AP in the beginning while all the other nodes start as STAs wanting to receive the same content.

IV. Testbed Design and High-Density Solution

This section illustrates the design details and the high-density solution of the proposed real-device testbed and simulator testbed. Our main focus in building these testbeds are threefold: (1) both work in high-density large-scale environments, (2) their performance is consistent so that evaluation is realistic thanks to the real-device testbed while embracing the flexibility of the simulator testbed, and (3) for ease of use and porting, off-the-shelf devices are used for the real-device testbed, and most implementation is done in the application layer.

A. Real-Device Testbed

1) Environment: Fig. 3 shows the architecture of our real-device testbed. Each node is a BeagleBone Black single-board computer with a wireless network interface controller (NIC) connected by a USB port. Each node has an AM335x 1GHz ARM Cortex-A8 CPU, and the NIC we use is Onkyo UWF-1. We build a testbed with 100 nodes, where each 50 nodes form a group and are connected to one central PC via mini USB ports as a control interface. We input commands, update content dissemination protocols, monitor network statuses, and collect experiment logs through the control interface.

The real-device testbed is implemented in the application layer with Ruby socket programming on Linux. For controlling low-level wireless behaviors, we use *iw* utility for scanning, *hostapd* for launching AP (master mode), and *wpa_supplicant* for starting STA (managed mode). The NICs in the nodes are set to 802.11n mode with transmission power at 20 dBm, and the connection capacity of the NIC is 7, which means an AP can host up to 7 STAs. We control the multicast bit-rate with modification of *open_ath9k_firmware* and set the application-layer transmission rate with Ruby.

2) Monitoring and Analysis: For controlling 100 nodes at the same time, we exploit *parallel ssh* over the control interface to issue commands and update source code to the nodes. For real-time monitoring of content dissemination progress, the nodes send events back through the control interface. Therefore, we can monitor events through the central PCs in real time, such as which node is connected to which node, how much data each node has received, what status each node is in

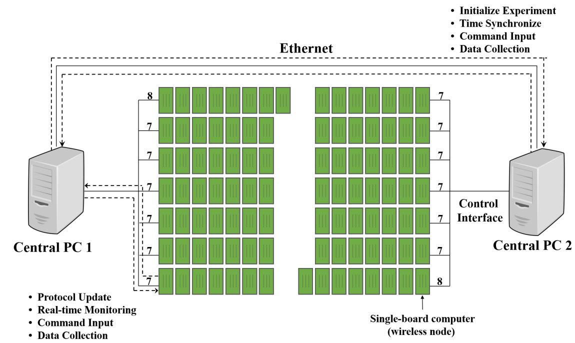


Fig. 3. The architecture of the real-device testbed.

(i.e., satisfied sender AP or unsatisfied receiver STA), etc. We also modify the blinking pattern of the indicator LEDs of the devices to display their statuses. This helps when conducting experiments in a wide area, in which it can be troublesome for a human operator to return to the central PCs to check the experiment progress. Collection of experiment logs and received content for checking the files' completeness can also be done over the control interface.

3) High-Density Solution: With current 802.11 WANET in a large-scale and high-density scenario, we found that wireless parameter tuning is needed to avoid unnecessary control traffic coming from connection establishment trials, which can be severe as wireless NICs for consumer devices usually have a limited connection capacity (7 STAs in our case). In a high-density environment, the number of STAs can easily exceed APs' capacity, causing APs to be confronted by severe contention. We use the following three countermeasures to alleviate this issue. Without these tunings, we observed that content dissemination stopped making progress in the middle of experiments as STAs could no longer connect to any AP.

- *AP scan: We change the scan method to passive scan.* Active scan is enabled by default because it normally has better performance. However, in the scenario of content dissemination among dense nodes, active scan causes STAs to send a large amount of probe requests, which consume the wireless bandwidth inefficiently as well as force APs to deal with probe requests from many STAs nearby.
- *Beacon interval: We raise the AP's beacon interval.* Since we shift the active role of connection establishment from STAs to APs by using passive scan, we further reduce the overhead management frames from APs.
- *Beacon bit-rate: We raise the bit-rate of beacon.* We modify AP beacon's transmission bit-rate. By default, it is set at the lowest bit-rate, i.e. conventionally 1 Mbps at 802.11. However, in the dense scenario we target, the low bit-rate beacons not only occupy extra bandwidth but also cause more STAs to send association requests to APs because of the broader coverage. These drawbacks directly result in a longer connection time between an AP and STAs and can even disable the whole network.

B. Simulator Testbed

1) Environment: The simulator testbed is implemented with NS-3 [20] running on Linux machine with CPU Xeon E5-2680 with eight 2.7 GHz cores and 32 GB RAM. We have modified the primitive modules controlling AP's behavior (ap.c) and STA's behavior (sta.c) in MAC layer because nodes can switch between STA and AP. The physical model is set to 802.11n with MCS index set to 2. The propagation delay we adopt is *constant speed propagation delay model*. For further consistency with the real-device testbed, we measure received signal strength indicator (RSSI) at different distances of nodes in the real-device testbed. We calculate the path loss exponent by an asymptotic equation and set it to 1.8 in the path loss model.

2) Monitoring and Analysis: Our simulator testbed records events during a simulation similar to the real-device testbed. After finishing a simulation, a simulation log can be analyzed and the progress of content dissemination can be visualized. The visualization tool implemented with OpenGL displays statuses of content delivery to help understand the dissemination behavior. A snapshot of the simulation is shown in Fig. 1b.

3) High-Density Solution: We observed connection establishment failures similar to the real-device testbed where probe requests and responses consume bandwidth as the result of many STAs trying to connect an AP, causing slowing down of content dissemination. We alleviated this issue by making STAs to hibernate for a predetermined time without issuing further requests once all the connection trials have failed during a certain time interval. To deal with high computation load of simulating 10,000 nodes, we prune computations performed in default NS-3 simulator that is unnecessary for our specific scenarios, such as computations of satisfied nodes that are not promoting to APs and of links between far away nodes. This halves computation time: 800-node simulation takes 9 hours and 10,000-node simulation takes 7 days to finish.

V. Reference Protocols for Evaluation

In order to show performance evaluation of content dissemination on our testbeds, we implement representative protocols based on the target scenario explained in Section III. We divide the protocols into two parts: one is single-hop transmission from a sender AP to receiver STAs, and the other is multi-hop relay selection used to determine which of satisfied STAs upgrade to relay APs. The following subsections introduce two reference single-hop transmission schemes: *unicast* and *reliable multicast*, and two reference multi-hop relay selection schemes: *flooding* and *RSSI-based*.

A. Single-hop Transmission

1) Unicast: We use unicast as a baseline single-hop transmission scheme, where an AP uses a TCP socket to transmit the content to each of connected STAs. In comparison with multicast, the advantage is that we do not need to worry about its reliability as TCP handles the retransmission mechanism.

2) Reliable Multicast: To achieve better content dissemination performance, multicast is a natural choice because it reduces the need of sending the same data several times. However, multicast in 802.11 does not provide a TCP-like reliable link. Hence, we implement a NACK-based retransmission mechanism similar to [21] over a UDP socket. To alleviate NACK collision problems, we add the following three features. (1) STA checks lost packets and aggregates retransmission requests in one packet to reduce the amount of requests. (2) STA randomly delays the time to send a retransmission request, where the expected delay is inversely proportional to the number of lost packets. That is, the more packet losses, the more likely the request will be sent earlier. (3) STA cancels sending retransmission requests if it overheard other STA's requests including the same packets it needs.

B. Multi-hop Relay Selection

1) Flooding: Here we again use the most naive approach as a baseline scheme. In this flooding scheme, we allow all the satisfied STAs to turn into relay APs and start content dissemination to their respective surrounding nodes.

2) RSSI-Based: Flooding will create large number of APs, which can cause congestions. Hence, we implement a smarter relay selection mechanism based on received signal strength indicator (RSSI). The basic principle is to let the farthest (from the source AP) satisfied STAs become relay APs, as shown in Fig. 2. This strategy makes multi-hop propagates faster as well as lightens signal interference between relay APs and the source AP.

To be specific, once an STA is satisfied, it scans and checks the number of surrounding APs. If the number exceeds a certain threshold (3 in our implementation), the satisfied STA will not become a new AP and go idle. If the number is zero, the STA turns into a relay AP immediately. If the number is below the threshold but not zero, the STA generates a random waiting time based on the beacons' RSSIs from the nearby APs, where the expected waiting time is proportional to the maximum of the RSSIs. That is, the closer the STA is to other APs, the longer the waiting time will be. The STA scans the channel again after the waiting time, and if the number of APs is still below the threshold, the STA becomes a new AP and starts to serve the surrounding unsatisfied nodes. With this mechanism, satisfied STAs farther away from existing APs have higher opportunity to become new APs.

VI. Experiments

We conducted a series of experiments to demonstrate the capability of our two testbeds. First, (A) we evaluate performance of content dissemination protocols at an extremely high density on the real-device testbed. Second, (B) we study the relationship between performance and density including more realistic (but still high) density values. Next, (C) we show the consistency of the real-device testbed and simulator testbed. Finally, (D) we conduct experiments on the simulator testbed beyond the limits of the real-device testbed.

TABLE I
EXPERIMENT PARAMETERS

Parameter	Value
Content data size	2.5 MB
Transmission power	20 dBm
Physical layer bit-rate	19.5 Mbps
Application layer bit-rate	5.2 Mbps
Connection capacity	7 STAs per AP
*AP scan	Passive scan
*Beacon interval	250 ms
*Beacon bit-rate	19.5 Mbps

TABLE II
PROTOCOLS TO BE EVALUATED

Protocol	Single-hop Transmission	Multi-hop Relay Selection
1	Unicast	RSSI-based
2	Reliable Multicast	Flooding
3	Reliable Multicast	RSSI-based

Table I lists the parameters used in all the experiments unless otherwise noted. The items with asterisks are for handling high-density environments as illustrated in Section IV-A.

Each experiment starts with one source node while all the others are waiting for the content. An experiment ends when all the nodes are satisfied. As a performance metric, we define the time-dependent system throughput $T(t)$ as:

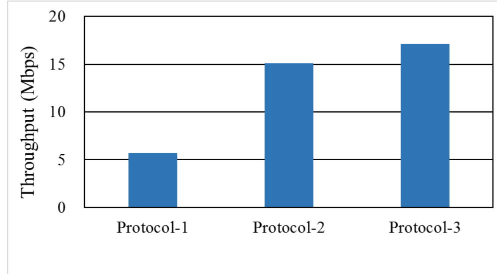
$$T(t) = \frac{1}{t} \sum_{n \in S} d_n(t), \tag{1}$$

where $d_n(t)$ is the data size the node n has received at time t since the source starts to disseminate the content, and S is the set of all the nodes excluding the source. The system throughput $T(t)$ has a unit of Mbps and indicates how fast the content has been disseminated to all the nodes in a system until time t. Most of the performance statistics shown below will be expressed as the terminal system throughput $T_1 = T(t_1) = D|S|/t_1$, where t_1 denotes the time at which all the nodes are satisfied, and D is the content size.

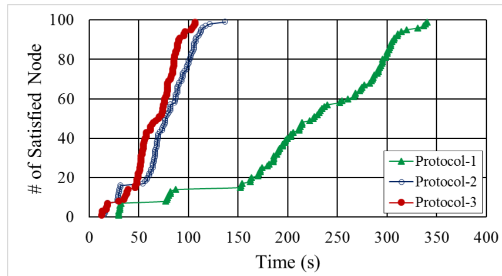
A. Experiment 1: Protocol Evaluation on Real-Device Testbed

This experiment compares the three reference protocols summarized in Table II, which are combinations of the protocol components illustrated in Section V. The experiment was conducted at an extreme density of 22.2 nodes/m^2 on the real-device testbed, showing the feasibility of protocol evaluation for high-density scenarios in a real-world setting.

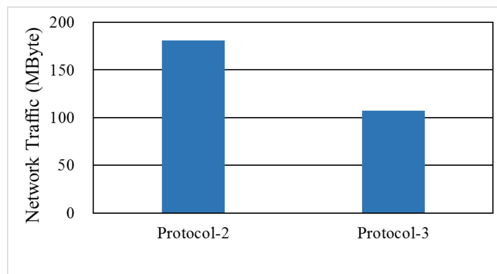
As shown in Fig. 4a, the terminal system throughput T_1 of these protocols were 5.7 Mbps, 15.1 Mbps, and 17.2 Mbps, respectively. With Reliable Multicast, both Protocol-2 and Protocol-3 outperformed Protocol-1. We can further observe the process of the content dissemination in Fig. 4b. Protocol-1 took 340 seconds to finish because it does not exploit the multicast gain. Protocol-2 has a fair performance, finishing the dissemination at 137 seconds, which is not far from 115 seconds achieved with Protocol-3. However, we found that Protocol-3 only created 60% of the network traffic of Protocol-2, as shown in Fig. 4c. This shows the RSSI-based relay selection in Protocol-3 contributes to higher throughput as well as to lower traffic than Flooding-based relay.



(a) Terminal system throughput of the three protocols.



(b) Time development of the number of satisfied nodes.

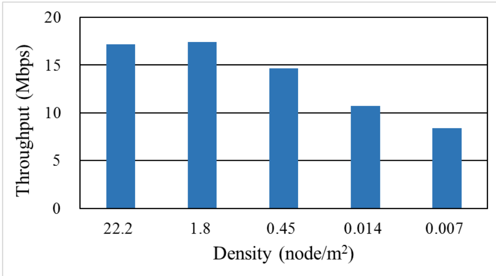


(c) Network traffic of Protocol-2 and Protocol-3.

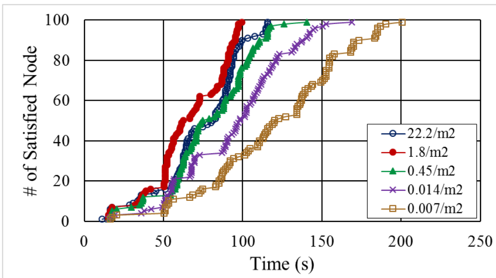
Fig. 4. Performance comparison on the real-device testbed between Protocol-1 (Unicast + RSSI-based Relay), Protocol-2 (Reliable Multicast + Flooding), and Protocol-3 (Reliable Multicast + RSSI-based Relay). Protocol-3 has the highest system throughput. The throughput of Protocol-2 is only slightly lower than Protocol-3, but it causes 1.7 times more traffic due to larger number of APs eventually generated. [Exp. 1]

B. Experiment 2: Study on Density on Real-Device Testbed

In this experiment, we take the best-performing Protocol-3 (Reliable Multicast + RSSI-based Relay), and investigate its performance again on the real-device testbed at more realistic densities of 1.8, 0.45, 0.014, and 0.007 node/m^2, corresponding to stadiums, classrooms, and (more and less crowded) theme parks, respectively. We realized densities of 1.8 and 0.45 node/m^2 by physically spreading the nodes in wider areas, and emulated 0.014 and 0.007 node/m^2 by decreasing the transmission power from 20 dBm to 5 dBm and 2 dBm, respectively, because of the lab space limitation.



(a) Terminal system throughput for different densities.



(b) Time development of the number of satisfied nodes.

Fig. 5. Performance of Protocol-3 (Reliable Multicast + RSSI-based Relay) at varying densities on the real-device testbed. The system throughput drops as the density decreases due to higher packet error rate, but the degradation is moderate despite the density differing by orders of magnitude. [Exp. 2]

As shown in Fig. 5, we observe that the terminal system throughput remained mostly unchanged at a decreased density of 1.8 node/m^2, achieving 17.5 Mbps. This is because all the nodes can hear each other well above around this density. The throughput drops as the density decreases further due to higher packet error rate as the result of longer distances between nodes, but we see that the performance degradation is moderate, and we still obtain reasonable throughput at 0.007 node/m^2, finishing the content dissemination in 200 seconds.

C. Experiment 3: Study on Consistency of the Two Testbeds

Here, we investigate the consistency of our real-device and simulator testbeds. We conduct experiments on both testbeds with varying numbers of nodes up to 100, with Protocol-3 (Reliable Multicast + RSSI-based Relay) at a density of 22.2 node/m^2. Fig. 6 shows two plots indicating the terminal system throughput of the two testbeds. We observe that the throughput for both testbeds increases in a non-linear manner as the number of nodes increases. This comes from stepwise increase in the content dissemination finish time t_1 when the limited APs' connection capacity requires additional hops for the content to reach a larger number of nodes. The two plots both exhibit this non-linear behavior with the same increasing trend. While the simulator always obtains better throughput due to the absence of real-world complexity, this tendency is also consistent, and the performance of the two testbeds matches roughly within a factor of two. This also demonstrates that our simulator can likewise handle high-density scenarios.

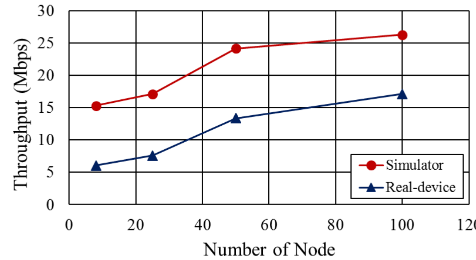


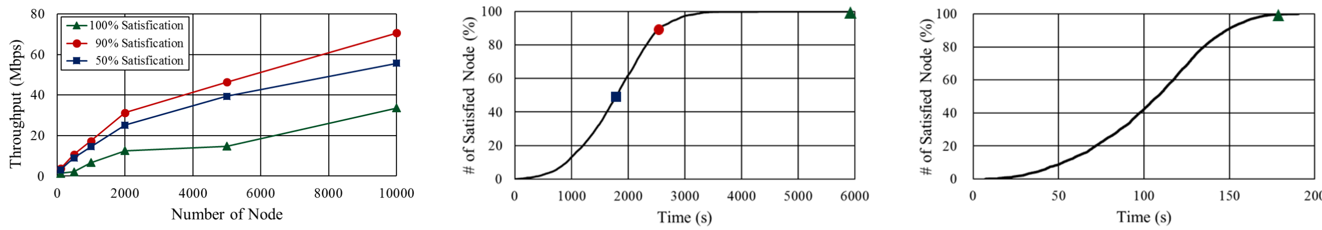
Fig. 6. Terminal system throughput of the real-device and simulator testbeds for varying numbers of nodes up to 100, which shows that the performance of the two testbeds roughly matches within a factor of two. [Exp. 3]

D. Experiment 4: Simulation beyond Real-Device Testbed

One of the merits of having a simulator testbed is that we can conduct experiments beyond the limits of the real-device testbed. Here we increase the number of nodes and the connection capacity in the following experiments.

1) Number of Nodes > 100: Fig. 7a shows the results of experiments with up to 10,000 nodes with Protocol-3. We set the density to 0.01 node/m^2, which represents a theme park scenario where 10,000 nodes are distributed in a 1 km^2 area. We observe that the system throughput grows as the number of node increases similarly to Experiment 3 shown in Fig. 6. In addition to terminal system throughput T_1 for which 100% of the nodes are satisfied, we also show system throughputs $T_{0.9}$ and $T_{0.5}$ corresponding to the times when 90% and 50% of the nodes are satisfied, respectively, in Fig. 7a. We see that the throughput $T_{0.9}$ is the highest, followed by $T_{0.5}$, and T_1 comes last for all the experiment variations. This is because, as can be seen in Fig. 7b, the number of satisfied nodes begins to saturate as the content dissemination approaches the border of the experimental area, where there are less unsatisfied STAs.

2) Number of Nodes > 100 & Connection Capacity > 7 : Furthermore, we consider increasing the connection capacity as this is one of the major limitations of currently available consumer devices. We increased the value from 7 to 50, and optimized the other parameters as well to fit with this setting. We increased the transmission power by 16 dBm, the physical layer bit-rate to 65 Mbps, and the application layer bit-rate to 16 Mbps. In Fig. 7c, we observe that the finish time drastically shortened from 5,958 seconds to 190 seconds, suggesting the potential of achieving this level of performance with real devices if such parameters were available. While there is no guarantee that the performance of the two testbed would match beyond the current limitations of the real-device testbed, we believe that our simulator can provide more grounded estimates for real-world performance. The result obtained here also illustrates that simulation evaluation alone may be unreliable because only moderate change in parameters can lead to a significant difference in system-wide results, especially in large-scale experiments. This is also indicative of the importance of having a real-device counterpart.



(a) System throughput for varying numbers of nodes with the same parameters as the real-device testbed.

(b) Time development of the number of satisfied nodes for a 10,000-node experiment with the same parameters as the real-device testbed.

(c) Time development of the number of satisfied nodes for a 10,000-node experiment with optimized parameters.

Fig. 7. Experiments on the simulator testbed beyond the limits of the real-device testbed. (a) shows results for larger numbers of nodes beyond 100 with the same parameters as the real-device testbed. (b) shows the time development corresponding to the 10,000-node experiment in (a). (c) shows the result with an increased connection capacity and other optimized parameters, resulting in 30x performance increase. The colored lines and dots indicate the system throughput and times at different node satisfaction percentages. [Exp. 4]

VII. Discussion

Our goal in this paper is to study the feasibility of large-scale (high-density and many-node) WANETs, and our primary concern here is whether wireless congestion caused by densely packed nodes renders the ad hoc network they form impractical. Therefore, as a first step, we have considered a simple yet challenging wireless environment where devices at an extreme density of up to 20 nodes/m^2 all try to share the same content. The reason why we built our own testbeds is simply because we could not find readily available testbeds suitable for such experiments, and outperforming existing testbeds is not the focus of this paper.

Nonetheless, we would like to highlight some of the key differences from ORBIT [8], which is arguably the closest existing testbed to ours in terms of scale. Firstly, ORBIT is a general-purpose programmable testbed, whereas ours are geared to heavy-traffic environments targeted at an order of magnitude higher-density scenarios. Since ORBIT is served as a fixed infrastructure, repositioning the devices to change their density is outside the scope of the testbed's features. Another difference is that our real-device testbed has a simulator counterpart for reproducibility and scalability with which we can estimate the performance of content dissemination protocols with up to 10,000 nodes.

For other existing approaches, we would like to note that their differences from ours not only come from assumed node density but also from research objectives. For example, many of the previous work are targeted at end-to-end communication rather than flooding-like content dissemination. Although UFlood [7] is an exception, it is a protocol rather than a testbed. Implementing more sophisticated protocols than the basic ones described in Section V will be a promising extension to the current testbeds.

VIII. Conclusion

This paper has presented a pair of testbeds based on real devices and a simulator that are specifically designed for content dissemination in high-density large-scale WANET. Without our high-density solution implemented on both testbeds, we have found that nodes cannot even establish connections. Through experiments, we have shown that both testbeds are capable of dealing with high density of up to 20 nodes/m^2 with roughly matched performance, delivering 2.5 MB of data to 100 real devices and to 10,000 simulated devices within several minutes. This is in contrast to commonly seen testbeds where fewer nodes are scattered at 10^{-4} nodes/m^2.

This paper has focused on increasing the scale (i.e., density and numbers of nodes) of experiments, and there are many other aspects of WANET testbeds that we have not looked at here. These include dissemination of multiple contents, routing from specified sources to destinations, use of multiple wireless channels, node mobility, security, and energy consumption, to name but a few. Addressing them to create more versatile testbeds will be an important future research area.

Acknowledgments

We thank Hiroki Kudo and Ren Sakata for their insightful opinion and the contribution to the simulation testbed. We thank Arturo Parrales and Rafael Rodriguez for their participation of building the real-device testbed. We thank Kohsuke Harada, Daisuke Iwai, Juan Shi, Yu-Chieh Lin for helpful comments about this paper's structure and writing. We thank our project's shepherd, Hiroto Nakai, for supporting this research topic all along, and also the anonymous reviewers for their valuable feedback for improving the quality of this paper.

References

[1] H. Zhou, L. M. Ni, and M. W. Mutka, "Prophet address allocation for large scale MANETs," in *IEEE Int. Conf. on Computer Communications (INFOCOM)*, vol. 2, 2003, pp. 1304–1311.

[2] E. Ahvar and M. Fathy, "Performance evaluation of routing protocols for high density ad hoc networks based on energy consumption by GloMoSim simulator," *World Academy of Science, Engineering and Technology*, vol. 29, 2007.

[3] C. Gottron, A. König, M. Hollick, S. Bergstrasser, T. Hildebrandt, and R. Steinmetz, "Quality of experience of voice communication in large-scale mobile ad hoc networks," in *The 2nd IFIP Wireless Days (WD)*, 2009, pp. 1–6.

[4] P. Goyal, "Simulation study of comparative performance of AODV, OLSR, FSR & LAR, routing protocols in MANET in large scale scenarios," in *World Congress on Information and Communication Technologies (WICT)*, 2012, pp. 283–286.

[5] B. A. Chambers, "The grid roofnet: a rooftop ad hoc wireless network," Ph.D. dissertation, Massachusetts Institute of Technology, 2002.

[6] Y. Owada, Y. Takahashi, T. Suda, H. Terui, F. Taki, T. Yagi, and K. Mase, "A large scale wireless mobile ad hoc network testbed," in *IEEE 62nd Vehicular Technology Conference (VTC)*, vol. 1, 2005, pp. 324–328.
[7] J. Subramanian, R. Morris, and H. Balakrishnan, "UFlood: High-throughput flooding over wireless mesh networks," in *IEEE Int. Conf. on Computer Communications (INFOCOM)*, 2012, pp. 82–90.
[8] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh, "Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols," in *IEEE Wireless Communications and Networking Conference*, vol. 3, 2005, pp. 1664–1669.
[9] D. Ramasamy and U. Madhow, "Geographic routing in large-scale MANETs," University of California, Santa Barbara, Tech. Rep., Jan. 2012.
[10] J. W. Hui and D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale," in *The 2nd International Conference on Embedded Networked Sensor Systems*, 2004, pp. 81–94.
[11] T. Stathopoulos, J. Heidemann, and D. Estrin, "A remote code update mechanism for wireless sensor networks," University of California, Los Angeles, Tech. Rep. CENS-TR-30, Nov. 2003.
[12] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks," in *The 1st USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI)*, 2004, pp. 15–28.
[13] J. Su, J. Scott, P. Hui, J. Crowcroft, E. De Lara, C. Diot, A. Goel, M. H. Lim, and E. Upton, "Haggle: Seamless networking for mobile applications," in *9th International Conference on Ubiquitous Computing (UbiComp)*, 2007, pp. 391–408.
[14] D. J. Dubois, Y. Bando, K. Watanabe, A. Miyamoto, M. Sato, W. Papper, and V. M. Bove, "Supporting heterogeneous networks and pervasive storage in mobile content-sharing middleware," in *The 12th IEEE Consumer Communications and Networking Conference (CCNC)*, 2015, pp. 841–847.
[15] Ó. R. Helgason, E. A. Yavuz, S. T. Kouyoumdjieva, L. Pajevic, and G. Karlsson, "A mobile peer-to-peer system for opportunistic content-centric networking," in *The 2nd ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds*, 2010, pp. 21–26.
[16] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental evaluation of wireless simulation assumptions," Dartmouth College, Dept. of Computer Science, Tech. Rep. TR2004-507, Jun. 2004.
[17] P. Chaki, M. Yasuda, and N. Fujita, "Seamless group reformation in WiFi peer to peer network using dormant backend links," in *The 12th IEEE Consumer Communications and Networking Conference (CCNC)*, 2015, pp. 773–778.
[18] S. Trifunovic, B. Distl, D. Schatzmann, and F. Legendre, "WiFi-Opp: ad-hoc-less opportunistic networking," in *The 6th ACM Workshop on Challenged Networks (CHANTS)*, 2011, pp. 37–42.
[19] H. Wirtz, T. Heer, R. Backhaus, and K. Wehrle, "Establishing mobile ad-hoc networks in 802.11 infrastructure mode," in *The 6th ACM Workshop on Challenged Networks (CHANTS)*, 2011, pp. 49–52.
[20] NS3 Network Simulator, https://www.nsnam.org/.
[21] A. Sobeih, H. Baraka, and A. Fahmy, "ReMHoc: a reliable multicast protocol for wireless mobile multihop ad hoc networks," in *The 1st IEEE Consumer Communications and Networking Conference (CCNC)*, 2004, pp. 146–151.
[22] R. Mahindra, G. D. Bhanage, G. Hadjichristofi, I. Seskar, D. Raychaudhuri, and Y. Zhang, "Space versus time separation for wireless virtualization on an indoor grid," in *Next Generation Internet Networks (NGI)*, 2008, pp. 215–222.

APPENDIX

In Section II, we state that densities of nodes in the previous work are mostly in the order of 10^{-4} node/m^2 or less. We explain this in more detail here.

Table III summarizes the node densites found for both simulated and real-device testbeds in the previous work. In simulation testbeds, for each piece of work, if it conducts multiple experiments with varying parameters, we pick the experiment conducted in the smallest simulated area among the ones involving the largest number of nodes. For the node densities of the real-device testbeds of the previous work, the area values are taken as follows. Kotz et al. state that they used an athletic field measuring 225 by 365 meters [16]. Subramanian et al. used three floors of a building and the size of each floor is 150 by 250 meters [7]. Chambers states that the nodes spread over an eight square kilometer urban area [5]. Owada et al. built a testbed on the Niigata University campus [6], and we estimated the deployment area as 1,000 by 400 meters by comparing the figure in [6] with Google Maps. Raychaudhuri et al. state that, in ORBIT, the nodes are arranged in a 8 by 8 matrix with 1 m spacing, but after the publication the testbed has been extended to 20 by 20 nodes with 1 m spacing [22].

As can be seen in Table III, the densities in the previous work are in the order of 10^{-4} node/m^2 or less, except for ORBIT (1 node/m^2). In contrast, experiments at a density over 20 nodes/m^2 have been conducted on both of our real and simulator testbeds.

TABLE III
NODE DENSITIES IN PREVIOUS WORK

Simulation Testbeds	**# nodes**	**Area [km^2]**	**Density [m^{-2}]**
Goyal [4]	60	1.5×1.5	2.7×10^{-5}
Zhou et al. [1]	250	1.0×1.0	2.5×10^{-4}
Gottron et al. [3]	500	3.3×3.3	4.6×10^{-5}
Ahvar et al. [2]	1,000	2.0×2.0	2.5×10^{-4}
Ramasamy et al. [9]	100,000	16.5×16.5	3.7×10^{-4}
Real-Device Testbeds	**# nodes**	**Area [m^2]**	**Density [m^{-2}]**
Kotz et al. [16]	40	225×365	4.9×10^{-4}
Subramanian et al. [7]	25	$150 \times 250 \times 3$	2.2×10^{-4}
Chambers [5]	40	8×10^6	5×10^{-6}
Owada et al. [6]	50	$1,000 \times 400$	1.3×10^{-4}
Raychaudhuri et al. [8]	400	20×20	1.0