WiFi Multicast to Very Large Groups - Experimentation on the ORBIT Testbed

Varun Gupta[†], Raphael Norwitz[†], Savvas Petridis[†], Craig Gutterman[†], Gil Zussman[†], Yigal Bejerano[∗]
[†] Electrical Engineering, Columbia University, New York, NY, USA.
[∗] Bell Labs, Alcatel-Lucent, Murray Hill, NJ, USA.

***Abstract*—While WiFi has been proposed for multimedia content distribution, its lack of adequate support for multicast services hinders its ability to provide multimedia content distribution to a large number of devices. In our recent papers we proposed AMuSe, a scalable and adaptive system for WiFi multicast which is based on accurate receiver feedback and that incurs a small control overhead. Specifically, the system includes a scheme for dynamic selection of a subset of the multicast receivers as feedback nodes, which periodically send information, such as channel quality or received packet statistics, to the multicast sender. We implemented the AMuSe system in the ORBIT testbed and evaluated its performance in large groups with 150-200 receivers. We present a dynamic web-based application that demonstrates the operation of the system based on actual traces collected on the testbed in several experiments. It demonstrates the operation of AMuSe in various setting and environments.**

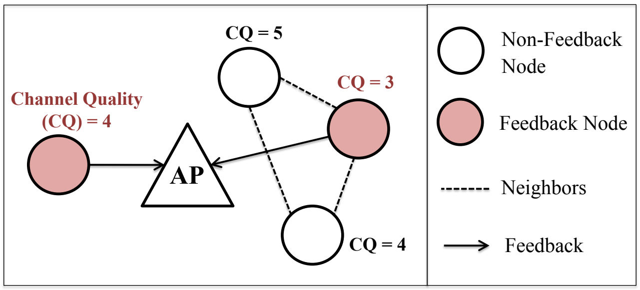


Fig. 1. Feedback node selection by *AMuSe*. A node with the poorest channel quality in every neighborhood is selected as a Feedback node. Each feedback node periodically sends updates about the service quality to the Access Point.

I. INTRODUCTION

Multimedia content delivery is an essential service for wireless networks. However, current techniques may not satisfy the user demand in crowded areas due to lack of enough spectrum. Several solutions [1], [2] have been proposed to address the problem of content delivery in crowded environments. Most of these are based on dense deployment of WiFi Access Points (APs) which require considerable capital and operational expenditure and may suffer from extensive interference between adjacent APs. Wireless multicast offers another approach for delivering multimedia content to large groups, where users share common interests (e.g. sports arenas, entertainment centers, lecture halls, or transportation hubs). Standard WiFi multicast frames are transmitted without any feedback. In such a situation, high packet losses due to interference and hidden node problem can significantly degrade service quality, while transmission at low bit-rates leads to low network utilization.

The current schemes to address the problem of feedback for WiFi multicast can be broadly divided into two categories: (i) Individual acknowledgement of packets from each node (e.g. [3], [4]), (ii) Leader-based protocols where a node, generally with the weakest channel quality, provides feedback (e.g. [4]–[6]). The individual feedback schemes are not scalable to environments of hundreds or thousands of nodes. In [7], [8], we showed that in large scale multicast environments, a few nodes (termed *abnormal nodes*) may always experience very poor channel quality due to multipath effects. Further, it has been shown in several studies that interference has a largely localized effect [7]. This implies that a multicast service with feedback from only a few poorly performing nodes may result in wasted network capacity.

In [7], we proposed a scalable feedback scheme for WiFi multicast referred to as *Adaptive multicast Services (AMuSe)*. *AMuSe* provides accurate receiver feedback and incurs a small control overhead. *AMuSe* enables the APs to efficiently utilize the capacity, while simultaneously ensuring high *Packet Delivery Ratio* (PDR) at a large fraction of the nodes.

In this demonstration, we present an interactive web-based application that demonstrates the performance of a large multicast system based on experimental traces collected on the ORBIT testbed. We collected the traces over several days in different experimental settings with 150-200 nodes. Each experimental trace consisted of channel measurements at each node using several metrics such Link Quality, RSSI, and Packet Delivery Ratio (PDR). The application allows users to replicate different scenarios such as different AP bitrates, channel conditions etc. For each scenario, the user will be shown the dynamic conditions over a certain period of time on the testbed from the appropriate experimental traces. Users can evaluate the performance of several feedback algorithms under different scenarios on the testbed using the application.

Our demonstration shows the presence of abnormal nodes on the system. We compare the performance of *AMuSe* system with other multicast feedback node selection schemes. We also demonstrate the performance of these schemes in different scenarios that have been measured on the testbed (e.g. no noise, AWGN noise, simultaneous WiFi flows, etc.) as well as syntactic scenarios based on manipulating the measured data. Finally, we demonstrate how effective feedback from

Fig. 2. The ORBIT testbed with 400 WiFi enabled nodes arranged in a grid topology.

AMuSe can utilized at the AP for diverse actions such as tuning multicast rate, Forward Error Correction (FEC) etc. We note that the application is extremely flexible and can be used for testing even more scenarios and algorithms in the future.

II. AMUSE OVERVIEW

The design of *AMuSe* is based on the observation that adjacent nodes experience similar channel quality and interference patterns [9]. *AMuSe* dynamically divides the nodes in a network into a few clusters based on adjacency of nodes and maximum cluster size (D m). In each cluster, the node with the weakest channel quality is selected as the *Feedback (FB) node*. In addition, abnormal nodes (i.e. node with PDR below a certain threshold) always become a FB node. An example is presented in Fig. 1. The FB nodes periodically update the AP about their service quality, e.g., channel quality.

AMuSe can be implemented in a quasi-distributed manner at the application layer and does not require any changes to the IEEE 802.11 protocol. The *AMuSe* server can be the WiFi Access Point (AP). When a node joins the network, it can volunteer to serve as a FB node while joining a multicast group by transmitting its location and channel quality. The AP in turn may accept or reject the volunteer request to ensure that each neighborhood has a FB node. The AP also periodically broadcasts a list of the current FB nodes so that in an event of changes, non-FB nodes may volunteer to serve as FB nodes as well. More details about *AMuSe* are in [7].

III. TESTBED ENVIRONMENT

The ORBIT testbed [10] is a dynamically configurable grid of 20 × 20 (400 overall) nodes equipped 802.11 Network Interface Cards (NIC) (Fig. III). The separation between adjacent nodes is 1 meter. We label each node in the grid according to its location (x, y) with x and y indicating its column and row location respectively.

In our experiments, the node at the corner $(1, 1)$ serves as a single multicast AP, configured in master mode. All the 802.11 radios are configured in channel 40 of 802.11a. We observed that channel 40 at the 5Ghz band suffers from less external interferences on the ORBIT grid than the channel at 2.4Ghz band regardless of the time frame of the experiments. The AP sends multicast UDP flows with each UDP packet of payload size 1400 bytes. The farthest node from the AP in the testbed is roughly 28 meters away which is significantly less than the typical transmission range of an AP. Thus, transmission power is set to $1mW = 0dBm$ to compensate for the relatively small size of the testbed. The other nodes are configured in managed mode and act as receivers. In order to avoid performance artifacts stemming from a mismatch of WiFi hardware and software, we only choose nodes equipped with *Atheros 5212/5213* wireless cards with *ath5k* wireless driver for our experiments. The experimental parameters are summarized in Table I.

TABLE I
EXPERIMENTAL PARAMETERS

Parameter	**Setting**
Mode	802.11a
Channel	40
Transmit Power	0 dBm
Wireless Driver	ath5k
Wireless Cards	Atheros 5212/5213
Transport Protocol	UDP
UDP payload size	1400
AP Location	(1,1)

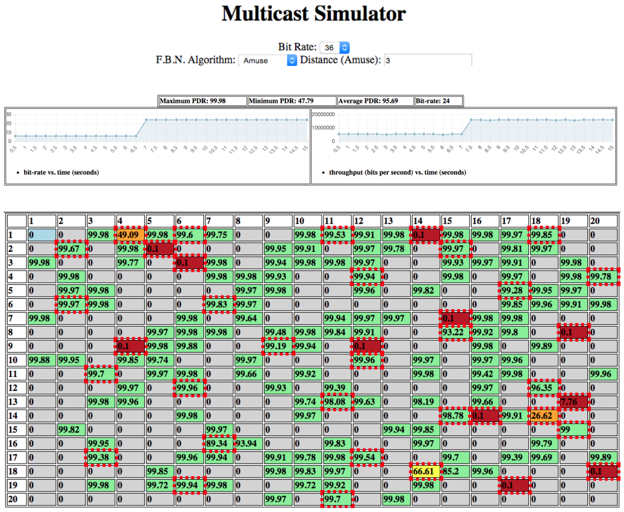
TABLE II
EVALUATION PARAMETERS

Parameter	Definition
LQ_i	Link Quality at node i
$RSSI_i$	RSSI at node i
P_i^{vec}	Vector of the packets received by node i
x_i, y_i	Location of node i
TX_{AP}	Broadcast/Multicast transmission bit-rate at the AP

Every node i keeps track of the parameters listed in Table II. The AP records the broadcast and multicast transmission bit-rate TX_{AP}. The nodes keep Link Quality (LQ_i), Received Signal Strength Indication ($RSSI_i$), a vector of packet sequence numbers for packets received by the node P_i^{vec}, node location in its column position in the grid x_i, and row position y_i. These parameters are transferred to a local machine for offline processing after each experiment. To enable the measurement of packet statistics, we add packet sequence numbers in the UDP payload. The *Packet Delivery Ratio* (PDR) value of each node i is calculated from its P_i^{vec} vector as a percentage of correctly received packets. The LQ and RSSI are read directly from the card. These three parameters are the only practically measurable *channel quality* metrics on a typical wireless card.

IV. DEMONSTRATION APPLICATION

We developed an interactive web-based application to demonstrate the performance of the multicast system. The application has been created using Django [11] which is a Python-based web application framework. Our application has two components: (i) the back-end where the experimental data is stored and managed, and (ii) the front-end which provides the user interface. Both the front-end and the back-end are light weight applications. The front-end is web-based and can operate on any standard browser while the back-end requires installation of easily available open source libraries. The front-



Multicast Simulator

Bit Rate: 36

F.B.N. Algorithm: Amuse Distance (Amuse): 3

Maximum PDR: 99.98 | Minimum PDR: 47.79 | Average PDR: 95.69 | Bit-rate: 24

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	99.98	49.09	99.98	99.6	99.75	0	0	99.98	99.53	99.91	99.98	0.1	99.98	99.98	99.97	99.85	0	0
2	0	99.67	0	99.98	0.1	0	0	0	99.95	99.91	0	99.97	99.78	0	99.97	0	99.81	99.97	0	0
3	99.98	0	0	99.77	0	0.1	99.98	0	99.94	99.98	99.98	99.97	0	0	99.93	99.97	99.91	0	99.98	0
4	0	99.98	0	0	0	0	99.98	99.98	99.93	0	0	99.94	0	0	99.98	0	99.97	0	99.98	99.78
5	0	99.97	99.98	0	0	0	0	99.97	99.98	0	0	99.96	0	99.82	0	0	99.28	99.95	99.97	0
6	0	99.97	99.98	0	0	0	99.83	99.97	0	0	0	0	0	0	0	0	99.85	99.96	99.91	99.98
7	99.98	0	0	0	0	99.98	0	99.64	0	0	99.94	99.97	99.97	0	0.1	99.98	99.98	0	0	0
8	0	0	0	0	99.97	99.98	99.98	0	99.48	99.98	99.84	99.91	0	0	93.22	99.92	99.8	0	0.1	0
9	0	0	0	0.1	99.98	99.88	0	0	99.19	99.94	0	0.1	0	0	0	99.98	0	99.89	0	0
10	99.88	99.95	0	99.85	99.74	0	0	99.97	0	0	0	99.96	0	99.97	0	99.97	99.96	0	0	0
11	0	0	99.7	0	99.97	99.98	0	99.66	0	99.92	0	0	0	99.98	0	99.42	99.98	0	0	99.96
12	0	0	0	99.97	0	99.96	0	0	99.93	0	99.39	0	0	0	0	99.97	0	96.35	0	0
13	0	0	99.98	99.96	0	0	0	0	0	99.74	98.08	99.63	0	98.19	0	99.66	0	0	7.76	0
14	0	0	0	0	0	99.98	0	0	0	99.97	0	0	0	0	98.78	0.1	99.91	26.62	0	0
15	0	99.82	0	0	0	0	99.97	0	0	0	0	0	99.94	99.85	0	0	0	0	99	0
16	0	0	99.95	0	0	0	89.34	93.94	0	0	99.83	0	0	99.97	0	0	0	99.79	0	0
17	0	0	99.38	0	0	99.96	99.94	0	99.91	99.78	99.98	99.54	0	0	99.7	0	99.39	99.69	0	99.89
18	0	0	0	0	99.85	0	0	0	99.98	99.83	99.97	0	0	66.61	85.2	99.96	0	0	0	0.1
19	0	0	99.98	0	99.72	99.94	99.98	0	0	99.72	99.92	0	0	99.98	0	0	0.1	0	0	0
20	0	0	0	0	0	0	0	0	99.97	0	99.7	0	99.98	0	0	0	0	0	0	0

Fig. 3. A screenshot of the demo application. The control panel for selecting the AP bitrate, feedback algorithm etc. is on the top. Several system statistics appear below. The performance of the client nodes is shown on the grid where numbers in each box indicate the PDR and the color of the box indicates the range of PDR. The nodes highlighted with a red border are FB nodes and nodes in grey are non-functional due to hardware issues.

end and the back-end can either reside on the same machine or on two different machines if the user machines cannot be configured with the back-end libraries.

The back-end utilizes a Postgres [12] database and interfaces with Django. The database is populated using the data derived from the experimental traces. The database consists of parameters in Table II at each node at different times for each experimental scenario. This allows us to show the performance of the testbed with evolving channel conditions. The statistics about performance at each node are derived from the packet vectors P^i_{vec}. The FB algorithms from [7] are built in the Django framework. The application is very flexible and allows other FB algorithms to be incorporated as well. The users can change the FB algorithms and tune the algorithm specific parameters at any given time on the front-end.

The front end is built using Angular [13] which is a JavaScript framework designed for rendering dynamic features on web applications. The front-end periodically relays the user defined parameters to Django which runs the corresponding FB algorithm and responds with information (including the state of the nodes and the system) to Angular. Angular in turn renders the information on user's screen. The period of rendering at the front-end as well as calculation of system performance parameters can be changed by the user. Typically, we will use a period of 500ms.

Fig. 3 shows a screenshot of the application. Demo participants can select different experiment settings such as AP bit rate, feedback algorithm, number of feedback nodes etc. on the web interface. This information is used along with data collected from the experiments to show how the performance at all the nodes on the grid. The feedback nodes are highlighted with a red border. The application also shows some system statistics such as the multicast throughput, the average PDR at the nodes etc. The information on the front-end is updated periodically.

V. DEMONSTRATION REQUIREMENTS

Our web application only requires a standard PC or a laptop for displaying the web-based front-end of the application. The back-end of the application is light-weight and can be placed on the machine used for the front-end. Both the front-end and the back-end will be incorporated in our laptop. Additionally, we will have a wide screen monitor which will greatly enhance the visual component of our application.

Another possibility is running a more powerful server in our lab that supports the back-end and providing a link to the server to the demo participants to enable more interaction with the application. Even while interacting with the server in the lab, we expect the bandwidth overhead to be very low and a standard wired or wireless broadband internet link is sufficient. The setup time for the demo is only a few minutes.

VI. ACKNOWLEDGEMENTS

This work was supported in part by NSF grant CNS-10-54856, CIAN NSF ERC under grant EEC-0812072, and the People Programme (Marie Curie Actions) of the European Unions Seventh Framework Programme (FP7/20072013) under REA grant agreement no. [PIIF-GA-2013-629740].11.

REFERENCES

[1] "Cisco, white-paper, cisco connected stadium Wi-Fi solution," 2011. [Online]. Available: http://www.cisco.com/web/strategy/docs/sports/c78-675064_svcs.pdf

[2] "Yinzcam," http://www.yinzcam.com/.

[3] "IEEE draft standard for information technology telecommunications and information exchange between systems local and metropolitan area networks - specific requirements, part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications - amendment: MAC enhancements for robust audio video streaming," IEEE, July 2011.

[4] Z. Feng, G. Wen, C. Yin, and H. Liu, "Video stream groupcast optimization in WLAN," in *Proc. IEEE ITA'10*, 2010.

[5] O. Alay, T. Korakis, Y. Wang, and S. Panwar, "Dynamic rate and FEC adaptation for video multicast in multi-rate wireless networks," *ACM/Springer Mobile Netw. and Appl.*, vol. 15, no. 3, pp. 425–434, 2010.

[6] J. K. Kuri and S. Kumar, "Reliable multicast in multi-access wireless LANs," *ACM/Kluwer Wirel. Netw.*, vol. 7, pp. 359–369, 2001.

[7] Y. Bejerano, J. Ferragut, K. Guo, V. Gupta, C. Gutterman, T. Nandagopal, and G. Zussman, "Scalable WiFi multicast services for very large groups," in *IEEE ICNP'13*, 2013.

[8] ——, "Experimental evaluation of a scalable WiFi multicast scheme in the ORBIT testbed," in *3rd GENI Research and Educational Experiment Workshop (GREE)*, 2014.

[9] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," *Proc. ACM SIGCOMM'04*, 2004.

[10] "ORBIT testbed," http://orbit-lab.org/.

[11] "Django project." [Online]. Available: https://www.djangoproject.com/

[12] "PostgreSQL." [Online]. Available: http://www.postgresql.org/

[13] "AngularJS." [Online]. Available: https://angularjs.org/