

QMesh: a QoS Mesh Network with Mobility Support *

Edward Bortnikov Israel Cidon Idit Keidar Tal Kol Arkady Vaisman

`ebortnik@tx.technion.ac.il, {cidon, idish}@ee.technion.ac.il, {stalkol, sav}@t2.technion.ac.il`Department of Electrical Engineering,
The Technion, Haifa**Abstract**

We present QMesh, a software package that allows utilizing multiple geographically scattered Windows desktops as a wireless mesh network infrastructure with seamless user mobility support. QMesh supports its users through standard protocols, and does not require any client software installation. We optimize the solution's quality of service (QoS) by providing a centralized management infrastructure, which allows an assignment of users to Internet gateways that balances between distance and load considerations.

QMesh is implemented as a Windows XP kernel driver, on top of the Mesh Connectivity Layer (MCL) toolkit from Microsoft Research that provides basic routing within the mesh. To the best of our knowledge, this is the first mobile mesh solution implemented within the Win32 kernel space.

1 Introduction

Wireless mesh networks, or WMNs, is a rapidly maturing technology for providing inexpensive Internet access to residential areas with limited wired connectivity [6]. While initially designed for small-scale installations (e.g., isolated neighborhoods), WMNs are now envisioned to provide citywide access and beyond [4, 5]. Modern mesh networks are expected to handle mobile real-time applications with diverse QoS requirements like VoIP, VoD, and online gaming [11].

*This work was supported in part by Israel Science Foundation.

WMN users access the Internet through a multihop backbone of fixed wireless routers. Each external user associates at all times with a single router that provides it with access to the mesh, which is called the users access point, or AP. Some of the routers, called gateways, are connected to the wired infrastructure. A common practice in small-scale WMNs is always assigning each user to the nearest gateway (e.g., [7]). In this approach, gateway handoffs (macro-mobility) are tightly coupled with link-layer AP handoffs (micro-mobility). This solution cannot adapt to load peaks within the mesh, thus limiting the network's capacity.

This shortcoming can be resolved by assigning *some* users from congested areas to distant gateways, hence avoiding congested paths, providing an improved quality of service (QoS), and eventually increasing the WMN's capacity. Intelligent gateway assignment policies must balance between the impact of link loads and network distances – in other words, perform *load-distance balancing* [8]. Note that gateway selection is a traffic engineering policy, rather than a routing extension. It can work on top of any routing protocol within the WMN.

We designed and implemented QMesh (Section 2) – a prototype QoS mesh network that features seamless mobility support and load-distance balancing. QMesh's external users perform a minimum of standard configurations, without installing additional software at their side. The QMesh infrastructure is based on inexpensive Windows XP desktops equipped with wireless cards, which makes it an attractive choice for office environments. The routing software deployed on the infrastructure nodes is a small-footprint device driver (to the best of our knowledge, this is the first WMN solution implemented in the Win32 kernel space). QMesh is managed by a centralized controller, which intelligently associates wireless users with access points and gateways. The QMesh code (driver and management software) and documentation are available for download at [3].

QMesh was deployed on a testbed of 7 mesh nodes, including two gateways. It supports a variety of real-life applications, including VoIP and video streaming. Our performance measurements (Section 3) demonstrate the viability of QMesh's approach to mobility and user assignment.

2 QMesh Architecture

The QMesh routing software is implemented on top of the Mesh Connectivity Layer (MCL) – an ad-hoc routing and link quality measurement software package developed at Microsoft Research that features the LQSR routing protocol [1, 10]. Architecturally, the MCL code is a Win32 NDIS driver that elegantly plugs into the host networking stack between the network and link layers. It abstracts the WMN’s multihop nature from upper-layer software, which handles the entire mesh as a single L2 segment. MCL requires installing its code on all network nodes. QMesh extends it with an access infrastructure functionality, namely, with MAC address resolution and unicast/broadcast traffic forwarding for non-LQSR users.

The QMesh controller is a user-space software that runs on a selected mesh router, and communicates with the other routers through LQSR extensions. It collects the wireless user location information from the access points, and associates every WMN user with a single AP and a single gateway. The controller can be instantiated with multiple assignment policies, encompassing nearest-neighbor assignment, perfect load-balancing, and more sophisticated algorithms that consider distance and load together (e.g., [8]). Fig. 1 illustrates the QMesh architecture.

2.1 AP Association and Seamless Mobility

In QMesh, the mobile user’s current AP functions as its default IP router. The user is forced to route all its traffic via this AP (a sandbox subnet) by setting the subnet mask to 255.255.255.255. The two nodes communicate directly, through a 802.11 ad-hoc link. (The alternative of implementing APs as transparent bridges operating in the 802.11 infrastructure mode was infeasible, due to an inherent shortcoming of most Windows wireless card drivers that do not support the promiscuous mode – the same problem was reported in [1]).

The assignment mechanism works as follows. As a mobile user initially associates with the mesh or moves away from its original AP, it gets discovered by one or more APs that intercept the user’s broadcast control traffic - e.g., periodical DHCP requests. These APs enter the user’s MAC address into their *local user cache*, or LUC, which they periodically send to the controller. The latter computes the (possibly new) assignment, and disseminates it in the network. All WMN

nodes store the user-AP associations in a *global user cache*, or GUC, to maintain address resolution within the mesh infrastructure segment. We explored three methods of communicating the AP association back to the mobile node, seamlessly to the end user:

Gratuitous ARP: originally suggested in [7]. All mobile users perceive the WMN as an omnipresent virtual access point. Its IP address is pre-configured by the user. Upon the initial association or handoff, the prospective access point manipulates the mapping of this virtual IP address to a MAC address, through publishing its own link-layer address in an unsolicited address resolution (ARP) reply (Figure 2(a)). The downside of this approach is that ARP is a low-level protocol that cannot be properly secured (e.g., encrypted).

ICMP Router Discovery Protocol (IRDP): manipulating the default router’s IP address itself [9]. The mesh AP assigned to the user publishes its own network address as the user’s default gateway, using a specific ICMP packet. IRDP can be enabled at a Windows computer through a dedicated DHCP request.

DHCP Reconfigure: manipulation of the default gateway’s IP address through a dynamic update triggered by the DHCP server [12]. This option is not supported by the Windows XP host networking stack, and we chose not to implement it.

Unlike the previous implementations (e.g., [7]), QMesh does not employ any reliable messaging infrastructure for forwarding in-flight packets during the AP transition. Instead, we opt for a simple and lower-latency kernel-level implementation. Our performance measurements demonstrate the viability of this approach.

3 Performance Evaluation

We first study the performance impact of access point handoffs, as follows. We measure the fluctuations of jitter in a G.711 VoIP stream emerging from a mobile node upon two AP transitions. The jitter values stabilize in the acceptable range (below 20 ms) within 200-400 ms (Figure 2(b)), thus supporting the findings in previous WMN implementations [7, 11].

The next experiment demonstrates the importance of balancing loads and distances in user

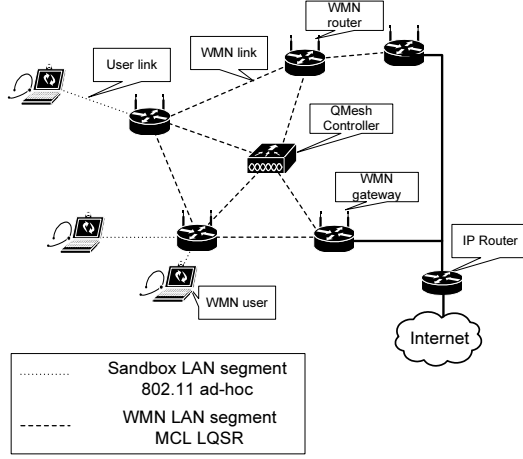


Figure 1: **The QMesh network architecture: users, mesh routers, and a centralized controller.**

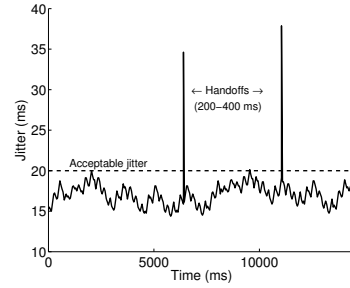
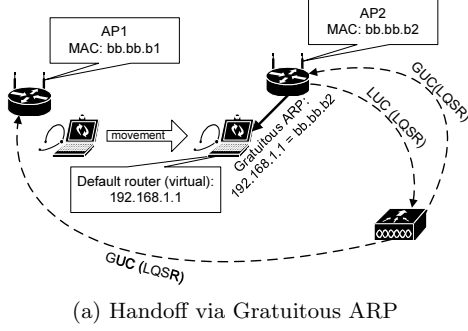


Figure 2: **AP handoff management in QMesh: (a) Gratuitous ARP-based handoff mechanism. (b) Fluctuations of VoIP jitter caused by AP handoffs.**

assignment. We measure the Mean Opinion Score (MOS) – the standard VoIP quality metric that combines the loss rate, jitter and delay experienced by the flow’s packets [2]. MOS values range from 0 to 5; values above 4.0 are assumed good. We consider a setting in which up to five wireless users are closer to one access point, which is also a gateway, than to any other mesh node. Therefore, assigning them to this nearest neighbor (the setup is depicted in Figure 3(a)) results in overloading the access link, and hence, in a degraded MOS. On the other hand, routing some flows through a more distant AP/gateway pair reduces the congestion, at the expense of an increased number of hops (Figure 3(b)). The measurements depicted in Figure 3(c) show that the second option can sustain all five flows within an acceptable quality, while the first one can handle only

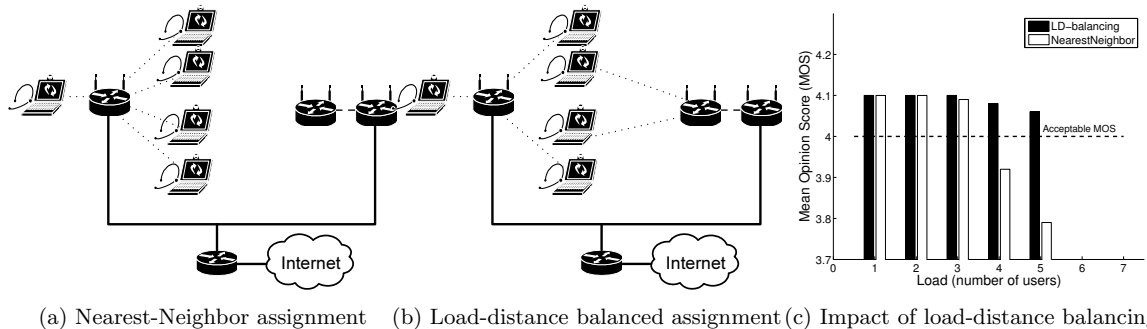


Figure 3: **Comparison of the (a) nearest-neighbor and (b) load-distance balancing assignment policies, for the VoIP application, in terms of the Mean Opinion Score (MOS) metric.**

three.

4 Conclusions

We presented QMesh – a novel WMN implementation within the Win32 kernel that features (1) native support of standard wireless clients, (2) transparent mobility, and (3) platform for intelligent user-to-gateway assignment. Performance evaluation conducted over a real testbed demonstrates the feasibility of QMesh’s approach to handoffs, as well as the importance of balancing distances and loads in assigning users to gateways.

Acknowledgements

We thank Microsoft Research for licensing the MCL software, Jitu Padhye for technical advise in MCL development, and Alex and Evgeny Hazanovich for their help with performance measurements.

References

- [1] Mesh Networking: Software Artifacts and Support. <http://research.microsoft.com/mesh>.
- [2] MOS Calculator. <http://davidwall.com/MOSCalc.htm>.

- [3] QMesh - a QoS Mesh Network with Mobility Support. <http://comnet.technion.ac.il/magma/qmesh/index.htm>.
- [4] Strix Systems. <http://www.strixsystems.com>.
- [5] Tropos Networks. <http://www.tropos.com>.
- [6] I.F. Akyildiz, X. Wang, and W. Wang. Wireless Mesh Networks: a Survey. *Computer Networks Journal (Elsevier)*, March 2005.
- [7] Y. Amir, C. Danilov, M. Hilsdale, R. Musaloiu-Eleftheri, and N. Rivera. Fast Handoff for Seamless Wireless Mesh Networks. *ACM MobiSys*, October 2006.
- [8] E. Bortnikov, I. Cidon, and I. Keidar. Scalable Load-Distance Balancing. *Pelc, A. (ed.) DISC 2007, LNCS 4731, Springer-Verlag*, 2007.
- [9] S. Deering. ICMP Router Discovery Messages. Internet-draft, IETF, June 1991. rfc1256.txt <http://www.ietf.org/rfc/rfc1256.txt>.
- [10] R. Draves, J. Padhye, and B. Zill. Routing in Multi-radio, Multi-hop Wireless Mesh Networks. *ACM MobiCom*, September 2004.
- [11] S. Ganguly, V. Navda, K. Kim, A. Kashyap, D. Niculescu, R. Izmailov, S. Hong, and S. Das. Performance Optimizations for VoIP Services in Mesh Networks. *JSAC*, 24:2147–2158, November 2006.
- [12] Y. T’Joens, C. Hublet, and P. De Shrijver. DHCP reconfigure extension. Internet-draft, IETF, December 2001. rfc3203.txt <http://www.ietf.org/rfc/rfc3203.txt>.