QMesh: a QoS Mesh Network with Mobility Support

Edward Bortnikov — ebortnik@techunix.technion.ac.il — Ph.D. student

Tal Kol, Arkady Vaisman — {stalkol,sav}@techst02.technion.ac.il — Undergraduate students

Department of Electrical Engineering,
The Technion – Israel Institute of Technology

***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 intelligent 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) academic resource 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 completely within the Windows kernel space.

I. Introduction

Wireless mesh networks, or WMNs, is a rapidly maturing technology for providing inexpensive Internet access to residential areas with limited wired connectivity [5]. While initially designed for small-scale installations (e.g., isolated neighborhoods), WMNs are now envisioned to provide citywide access and beyond [3], [4]. Mesh networks are expected to handle real-time applications with diverse QoS requirements like VoIP, VoD, and online gaming [8].

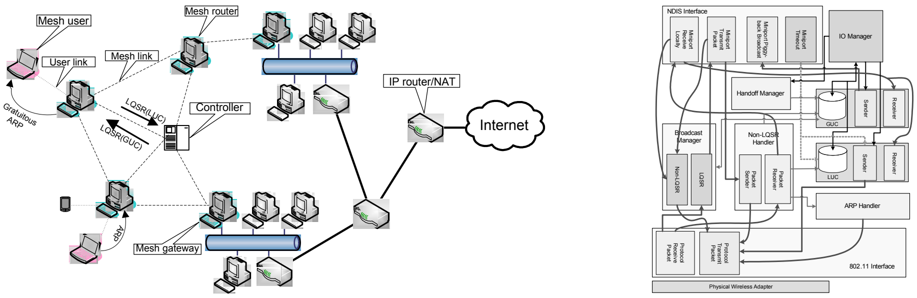
WMN users access the Internet through a multihop backbone of fixed wireless routers. Each external user associates at all times with a single router that provides it with access to the mesh, which is called the users access point, or AP. Some of the routers, called gateways, are connected to the wired infrastructure. A common practice in small-scale WMNs is always assigning each user to the nearest gateway (e.g., [6]). 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. Note that gateway selection is a traffic engineering policy, rather than a routing extension. It can work on top of any routing protocol. In particular, it does not induce self-interfering link metrics, which could affect network stability, or host-specific paths, which do not scale with network size. All that is required is the gateway id information at the user's current AP.

We designed and implemented `QMesh` – a prototype QoS mesh network with seamless mobility support. The `QMesh` implementation supports external users without installing additional software at their side. `QMesh` is managed by a centralized controller, which collects the wireless user location information from the access points, and intelligently associates wireless users with access points and gateways to optimize the QoS. The `QMesh` router platform is a standard Windows XP desktop with one or more wireless cards. The routing software is a small-footprint device driver (to the best of our knowledge, this is the first WMN solution implemented in the Windows kernel space). The `QMesh` code (driver and management software) and documentation are available for download at [2].

II. QMesh Architecture

`QMesh` is implemented on top of the Mesh Connectivity Layer (MCL) – an ad-hoc routing and link quality measurement software package developed at Microsoft Research [1]. MCL implements the LQSR routing protocol – a modified version of DSR [9] that implements sophisticated link-layer metrics for routing in multihop wireless networks. We extend MCL with a functionality of serving as a wireless access infrastructure, which provides seamless micro-mobility between the mesh access points. Architecturally, the MCL code is an NDIS driver that implements LQSR by elegantly plugging into the host networking stack between the network and link layers. `QMesh` extends this driver with MAC address resolution and unicast/broadcast traffic forwarding for non-LQSR users. The controller manages the user-AP and user-gateway assignments through LQSR extensions. Fig. 1 illustrates the `QMesh` architecture.



(a) Network perspective

(b) Software perspective

Fig. 1. **The** QMesh **architecture.**

QMesh associates each user with a single AP at all times. Unfortunately, we could not implement APs as transparent bridges like in the 802.11 LAN infrastructure mode, 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]). Mobile users therefore need to communicate with their APs directly, through the 802.11 ad-hoc mode. The user's current AP functions as its default IP router. The user is forced to route all its traffic via this AP by setting the subnet mask to 255.255.255.255. We explored three methods of user-AP association:

Gratuitous ARP: originally suggested in [6]. All mobile users perceive the WMN as an omnipresent virtual access point. Its IP address is pre-configured by the user. As the user initially associates with the mesh or moves away from its original AP, 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 resolution reply. 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 [7]. 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 [10]. This option is not supported by the Windows XP host networking stack, and we chose not to implement it.

QMesh maintains two main data structures in each mesh node: a *local user cache*, or LUC, which holds the MAC addresses of the user nodes whose 802.11 beacons are received by the node, and a *global user cache*, or GUC, which holds a mapping of the MAC addresses of the network's users that this node is aware of to the AP's that they can be reached at. The LUC is periodically sent to the controller, which uses it for assignment decisions. The GUC is maintained through standard ARP requests, which are handled by AP's in the name of their users.

Unlike the previous implementations (e.g., [6]), 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.

The QMesh infrastructure was deployed on a small testbed of 5 mesh routers. We demonstrated a smooth operation of a video streaming application under end user mobility. Our initial performance measurements [2] show that the impact of handoffs is small – the application's performance stabilizes in about 100 milliseconds, thus supporting the previous findings [6], [8].

Acknowledgements

We thank Microsoft for licensing the MCL software.

References

[1] Mesh Networking: Software Artifacts and Support. http://research.microsoft.com/mesh.
[2] QMesh - a QoS Mesh Network with Mobility Support. http://comnet.technion.ac.il/magma/qmesh.
[3] Strix Systems. http://www.strixsystems.com.
[4] Tropos Networks. http://www.tropos.com.
[5] I.F. Akylidiz, X. Wang, and W. Wang. Wireless Mesh Networks: a Survey. *Computer Networks Journal (Elsevier)*, March 2005.
[6] Y. Amir, C. Danilov, M. Hilsdale, R. Musaloiu-Elefteri, and N. Rivera. Fast Handoff for Seamless Wireless Mesh Networks. *ACM MobiSys*, October 2006.
[7] S. Deering. ICMP Router Discovery Messages. Internet-draft, IETF, June 1991. rfc1256.txt http://www.ietf.org/rfc/rfc1256.txt.
[8] 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.
[9] D. Johnson, Y. Hu, and D. Maltz. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. Internet-draft, IETF, February 2007. rfc4728.txt http://www.ietf.org/rfc-editor/rfc3261.txt.
[10] 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.