Seamless Multimedia Streaming in Controller-Less Wireless Mesh Networks With Mobile Stations

Markus Theil and Martin Backhaus and Michael Rossberg and Guenter Schaefer
Technische Universität Ilmenau, Germany
[markus.theil, martin.backhaus, michael.rossberg, guenter.schaefer][at]tu-ilmenau.de

***Abstract*—Seamless roaming in distributed Wireless Mesh Networks (WMNs) is still a complex problem in practice. We employ a 20-node testbed that enables mobile clients to stream live video with real-time properties despite frequent roaming events. A Linux-based prototype implements enhancements to the Babel distance vector routing and novel network-assisted roaming improvements.**

***Index Terms*—Wireless Mesh Networks, Client Mobility**

I. Introduction

Wi-Fi is a part of everyday life for many years already. In large infrastructures, clients usually attach to an Access Point (AP) that relays all traffic to a centralized (mobility) controller using wired connections. Capabilities of these wireless systems enable even delay-sensitive applications, e.g., interactive video streaming, site surveillance or real-time control data exchange.

Omitting controllers and forming a decentralized Wireless Mesh Network (WMN) will lead other to unique advantages: 1) High robustness (no single point of failure). 2) Rapid deployment, as communication between APs in a WMN is wireless as well, rendering wires superfluous. A controller-based WMN with high performance would be impossible in the first place, because all traffic traverses the controller – leading to additional strain of wireless resources, especially if clients are topologically close, but the controller is many wireless hops away. Meanwhile, the absence of a centralized controller depicts a major challenge, as decentralized algorithms to provide basic connectivity and mobility support will be much more complex. This challenge and wireless backbone communication render a WMN's applicability for delay-sensitive use cases questionable.

To enable distributed WMN to keep up with real-time requirements, significant enhancements to routing protocols and client roaming are required to keep additional delay low – especially when dealing with client mobility and frequent roaming events. In particular, client handover efficiency and consequently, the required time for routing convergence are essential for a WMN's applicability for real-time use cases.

We employ a 20-node wireless testbed to demonstrate a distributed WMN that enables attached clients to stream live video with real-time properties despite frequent roaming events due to mobility. A smartphone moving through the building will stream real-time video and audio to a stationary notebook. Our general architecture of the prototype as well as required enhancements to the routing protocol and mobility support, i.e., client handovers, are explained. Custom-made tools to setup the video stream are presented and the overall performance will be displayed during demonstration.

This work was supported by RWTÜV-Stiftung (Essen, Germany) under grant number S0189/10033/2018.

Annex to ISBN 978-3-903176-28-7© 2020 IFIP

II. System Design and Tools

The prototype consists of a WMN component combined with a custom Babel-MC routing protocol [1], a low-latency live-streaming Android app and its counter-part on a Linux PC.

A. Prototype Architecture

Our Linux-based prototype (see Fig. 1) written in C++ uses a component-based architecture and is structured into four custom daemons, which directly use the Linux kernel and communicate with each other via D-Bus interfaces if necessary. Additionally, hostapd is leveraged to manage Stations (STAs). *Meshd* configures cryptographically secured mesh peerings with a custom authentication protocol [2]. *APInterface* manages local STAs through hostapd and reports low signal events via D-Bus. *Routed* implements our enhanced Babel-MC [1], manages remote STAs and gathers neighbor information through the routing which are then signaled to APInterface via D-Bus. The last daemon, *stationDump* (not depicted in Fig. 1, as for demo purposes only), sends STA and mesh peer statistics to a monitoring component using UDP datagrams. Statistics are JSON-formatted and zstd-compressed. For enhanced scalability,

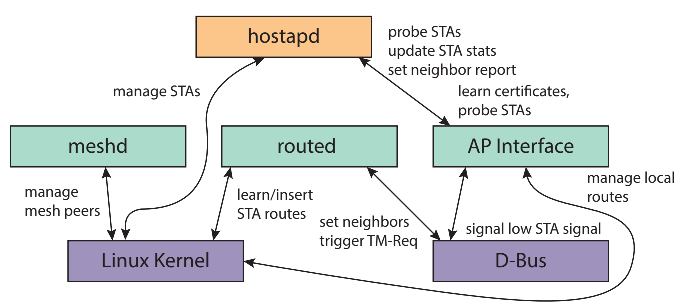

Fig. 1. Control flow of prototypical implementation on a Mesh AP.

our mesh system operates solely on layer 3 to diminish mesh-wide broadcasting or flooding, which would be necessary when using ARP. Furthermore, X.509 certificates with signed IP addresses and EAP-TLS as authentication mechanism are employed for each STA. With this combination, every AP can extract the STA's IP address from the certificate and adapt the routing, which is faster and more reliable than relying on DHCP snooping or an additional query mechanism. Furthermore, attested addresses can be used for further security measures, see [2]. A custom web-based monitoring component (explained in more detail in Sec. III) is used to gain system visibility and collect syslog messages, as well as custom events. It displays information ranging from network to single link and component level effectively, allowing for in-depth functional insights. JSON-formatted network events can be subscribed to in order to automate system monitoring further. The very same functionality can also be used to get performance insights into the system when devising and implementing new functionalities. For example, APInterface has a built-in thread, which can be activated to start continuous latency measurements of connected STAs. Events reporting high latency are then delivered to the monitoring components and can be used to monitor excessive scanning on STAs in order to optimize Wi-Fi signal coverage.

B. Enhanced Routing and Network Assistance

We use Babel-MC [1] with custom extensions (see Fig. 1), which improve (among others) integration of mobile clients. Dedicated client route updates are used to handle roaming explicitly, enabling specific optimizations. As IP addresses of clients are statically assigned, Babel-MC integrates clients as if they were announcing themselves. In particular, relevant Babel mechanisms (router IDs, sequence numbers) are modified to achieve this. Details can be obtained from [1]. Another issue when dealing with clients is the attachment itself, i.e., associations to APs, becoming even more important for mobility. We developed a distributed roaming architecture, which noticeably improves roaming times based on network-assisted recommendations of suitable APs. Recent roaming events are the basis for refined neighbor reports and transition management requests aid to initiate roaming for clients that may otherwise decide to roam too late or initiate superfluous scans. For details, see [3].

C. Live-Streaming Application

Low-latency streaming has high Quality of Service (QoS) requirements and is therefore used as showcase to illustrate our systems performance. Common live-streaming applications usable with smart phones or notebooks mostly employ HTTP Live Streaming (HLS) or Real-time Transport Protocol (RTP), using rather large de-jitter buffers to smooth out experienced network delay, making it unsuitable for real-time purposes. The gstreamer framework enables us to build custom low-latency streaming applications, is usable on Android and Linux and therefore a perfect fit for our purpose. The Opus codec is used as audio codec and H.264 (x264) as video codec – both tuned for low-latency performance. Fig. 2 shows screenshots of "Stream tool" for Linux and "Stream app" for Android.

III. Demonstration

The demonstration involves one moving client (an Android-based smartphone, walking pedestrian), which transmits live audio and video using our stream app to the stationary notebook

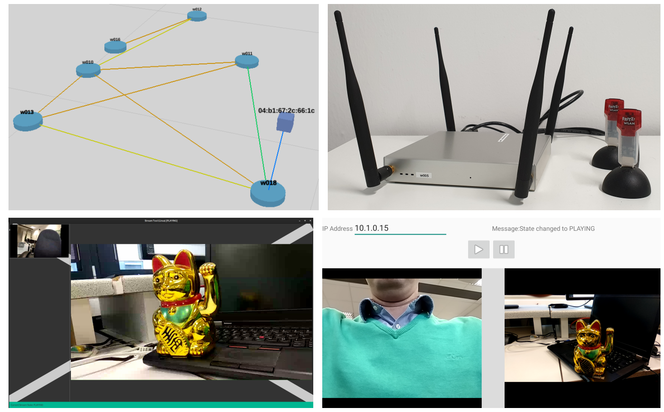

Fig. 2. Top left: Monitoring. Top right: Device hardware. Bottom left: Stream tool on Linux. Bottom right: Stream app on Android.

(running stream tool on Linux) over the WMN. Furthermore, throttling transmission power of all AP wireless interfaces leads to even more roaming events and compensates for the rather limited space our testbed needs to fit in, which is an area of roughly 40 m by 60 m in an office building. The live stream and important events regarding roaming and mobility are shown to the viewer. You will see extremely short video disruptions (mean roaming time 118 ms, see [1]) with our mechanisms in place, despite frequent roaming events due to client movement. Furthermore, a web interface also usable for monitoring our testbed is presented. It shows mesh nodes, client stations and statistics – useful for network/metric visualization in a single-page application. A force-based 3D layout based on mesh connectivity, with additional modifications to enable iterative layouting, is used to display the network topology. Currently, all Wi-Fi boxes reside on one floor, but with a 3D layout, even mesh networks spanning multi-floor-buildings or large outdoor areas can be efficiently visualized. A mesh-wide health indicator shows a pie chart for every AP which includes multiple segments for different signal levels. Furthermore, we show some debugging techniques for large mesh networks, which we used in order to get a working prototype with low delays.

IV. Conclusion

This demo showcases our Linux-based prototype of a distributed WMN architecture efficiently handling client mobility and allowing for fast and nearly seamless roaming between mesh points. We therefore used an audio and video streaming setup, which is very latency sensitive, allowing visitors to visually experience already mild impacts of frequent roaming events to the real-time behavior of the video stream.

References

[1] M. Backhaus, M. Theil, M. Rossberg, and G. Schaefer, "Poster: Seamless Client Integration for Fast Roaming in Wireless Mesh Networks," in *IFIP Networking Conference*, 2020, accepted and in press.

[2] M. Theil, M. Backhaus, M. Rossberg, and G. Schaefer, "Towards a Security Architecture for Hybrid WMNs," in *Intl. Conference on Availability, Reliability and Security*, 2019.

[3] M. Backhaus, M. Theil, M. Rossberg, and G. Schaefer, "Improving Network-Assisted Roaming for Controller-Less Wi-Fi," in *Intl. Symp. on Personal, Indoor and Mob. Radio Comm.*, 2020, currently under review.