2024 IEEE Symposium on Computers and Communications (ISCC) | 979-8-3503-5423-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISCC61673.2024.10733566

Automatic Bridging for Mobile Wireless Backhaul System

Tan Tai Phan, Shang-Chun Tai, Yu-Shuo Chang, Wei-Xun Chen, Chi-Yu Li
Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
Email: {taiphan.ee10, tsc39}@nycu.edu.tw, {ljk25679, chrissy81527}@gmail.com, chiyuli@nycu.edu.tw

***Abstract*—Backhaul networks, serving as the intermediary link from the front-end radio network to the central core network, play a crucial role in end-to-end communication by transporting significant traffic volumes. Apparently, packet transmission efficiency depends on their deployment scenarios; indeed, the more flexibility often translates to the better user experience for the end-user in critical situations. In this work, we propose an automatic bridging system for mobile wireless backhaul designed for high flexibility and compatibility. It features a fully automated establishment of wireless backhaul links, supported by a VXLAN-based packet transmission pipeline and an auto-binding mechanism. We prototype the system on Linux-based operating system with standard packages to ensure the compatibility operation and WiGig modules; the evaluation demonstrates that the VXLAN-based system can achieve lower latency, with a reduction ranging from 15% to 38% compared to the static forwarding-based system.**

***Index Terms*—Wireless backhaul, VXLAN tunnel, Auto-binding, WiGig.**

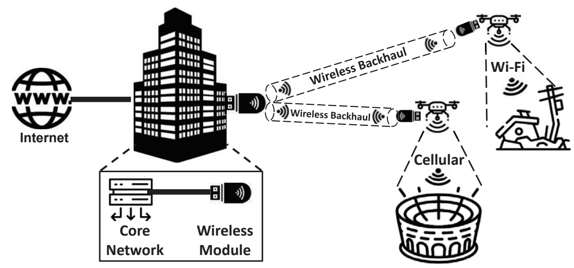


Fig. 1: Wireless backhaul deployment scenarios.

I. Introduction

Wireless network systems have become a critical extension part of the Internet, supporting a variety of applications, such as the Internet of Things, unmanned vehicles, and virtual reality. The market size of wireless network infrastructure is projected to reach approximately 375.1 billion USD by 2032 from 156.4 billion USD in 2022, with 2.4 times growth [1]. These systems typically comprise three main components: the Radio Access Network (RAN), the transport network, and the core network. The RAN is commonly deployed by two technologies: cellular and Wi-Fi networks. Mature technologies in the core network include the 4G Evolved Packet Core (EPC), 5G Core Network (5GC) for cellular and the Wireless LAN Controller (WLC) for Wi-Fi. In some cases, the 5GC can serve as the core network for Wi-Fi via a Non-trusted 3GPP network function [2].

Moreover, the transport network, also known as the backhaul network, is the intermediary link that transports the uplink/downlink traffic between the RAN to its core network. They are primarily relied on the fixed network infrastructure, such as fiber-optic cable technologies [3]–[6] that once the cable is laid down, any modifications require significant civil works and are costly. To deal with the fixed infrastructure issue, a recent mature technology, Wireless Distribution System (WDS) [7], [8] offers a bridge mode for extending wireless signal, potentially serving as a solution for the backhaul network. However, it lacks compatibility and flexibility across different products. The former issue is that the IEEE 802.11 does not define the implementation details; instead, the recent commodity products are defined by manufacturers themselves, while the latter requires manual configuration of a complex set of parameters.

In this work, we address the deployment challenges of backhaul networks in scenarios where there is a need for rapid deployment of wireless connectivity at remote places, as demonstrated in Fig. 1. The drones are equipped with RAN modules, such as cellular small cell, and Wi-Fi Access Points (APs) to provide wireless service for the clients under a live events at a stadium and the disaster areas. The wireless backhaul then transmits the client traffic to the central core network for the packet treatment before routing it to the Internet. These connections require a highly flexible and compatible infrastructure to ensure seamless connectivity. The reason is that the more flexible and compatible a system is, the faster it can respond to rapidly changing network demands.

To this end, we have implemented a plug-and-play wireless module for an automatic mobile wireless backhaul bridge system, leveraging the Virtual Extensible Local Area Network (VXLAN) approach. First, we designed a packet transmission pipeline for an overlay wireless network that ensures seamless transmission of the packets by VXLAN tunneling technology. Second, we developed an auto-binding mechanism, enabling the dynamic establishment of the VXLAN tunnels without any manual configuration in advance, making it highly flexible for wireless backhaul in Wi-Fi and 5G cellular networks. The evaluation results reveal that the VXLAN-based system can achieve lower latency, ranging from 15% to 38%, compared to the static forwarding-based system.

II. Related Work

The mobile backhaul network can be categorized into wireline, and wireless under various deployment scenarios.

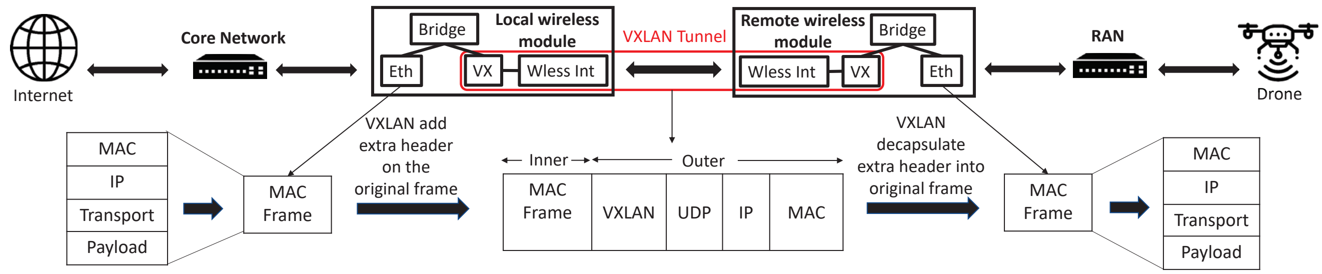


Fig. 2: Architecture of wireless module with VXLAN-based design.

Wireline backhaul. Traditional Fiber-to-the-Node (FTTN) backhaul network offers advantages such as stability and high throughput with two common deployment scenarios, including Point-to-Point (PTP) and Passive Optical Network (PON) [9]–[11]. Akhtar et al. [9] proposed and simulated the Recursive Clustering Algorithm (RAC) in a Time and Wavelength Division Multiplexed-Passive Optical Network (TWDM-PON) within a Distributed-RAN scenario to minimize the deployment length of fiber cable. Ranaweera et al. [10] can reduce 60% deployment costs in 5G mobile backhaul by utilizing PON technology for small cell backhaul networks. Mowla at al. [11] incorporated fiber with the PON method and mmWave to minimize the power consumption of the backhaul network in a flexible way by setting up various traffic conditions at different hours of the day.

Wireless backhaul. Various wireless backhaul solutions utilize major technologies such as WDS, Millimeter Wave (MmWave), and Free-Space-Optics (FSO) [7], [12], [13]. WDS is a recent mature technology that emerged as a prevalent solution for backhaul networks within Wi-Fi mesh topology aiming to relay layer-two IEEE 802.11 [12] frames between Access Points (APs) for wireless network extension. Muhammad et al. [7] verified the WDS feature, achieving a 39-meter wireless backhaul extension using DD-WRT firmware on two APs. However, this setup required manual configuration of parameters such as the wireless channel, SSID, IP, and MAC addresses. Kafafy at al. [13] employed both MmWave and FSO in the wireless backhaul to bridge transmitter and receiver nodes, aiming to maximize the transmission rate.

However, these aforementioned backhaul solutions either focus on various deployment scenarios to minimize the costs or entail a complex setup process with numerous static configuration parameters to establish a backhaul network. There is a gap for flexible deployment of mobile backhaul that has yet to be fulfilled.

III. Plug-and-Play Wireless Module Design

We aim to address the inflexibility of network structure due to fixed network infrastructure in the wireline backhaul network by developing a highly flexible and compatible deployment of the wireless backhaul network, realized through the integration of WiGig and VXLAN tunneling technologies.

A. *VXLAN-based Wireless Module*

The architecture of our wireless module and protocol stack based on VXLAN design is shown in Fig. 2. Each wireless module contains a wireless interface (Wless Int), a VXLAN interface (VX), a Linux bridge, and an Ethernet interface (Eth). The wireless bridge backhaul is established from one wireless module to another by the wireless interfaces. By binding the IP address of VXLAN interfaces with the wireless interfaces (e.g., WiGig NICs), the layer two MAC frames as the Inner part will be encapsulated inside the Outer part as the layer four UDP packets for transmission in the VXLAN tunnel. With this approach, we only need to set up the VXLAN interfaces, which can be dynamically configured in the runtime with support from an advanced mechanism. A Linux bridge is utilized to bridge the VXLAN and the Ethernet interfaces so that it does not need an extra IP address to deliver packets.

Furthermore, the Ethernet interfaces are connected to the RAN and core network of various wireless systems, such as 5G and Wi-Fi systems. Specifically, at the remote site, the Ethernet interface establishes connectivity with the RAN, including 5G base station and Wi-Fi Access Points (APs). Conversely, at the local site, the Ethernet interface connects to the core network infrastructure, including the 5G Core Network (5GC) and wireless LAN controllers, as the connection between the access network and core network is IP-based [7], [14]. For example, in the 5G deployment scenario, the two IP addresses assigned to the VXLAN interfaces could be considered as the endpoints for N2/N3 interfaces between the 5G base station and the Access and Mobility Function (AMF)/User Plane Function (UPF), as defined in 3GPP TS 23.501 specification [14].

B. *Transmission Packet Pipeline*

The packet transmission from one to another Ethernet interface in the wireless backhaul is carried out by the pipeline in Fig 3. The pipeline is executed concurrently at both the local and remote site wireless modules. Initially, the data packet arrives at the Ethernet interface of the local wireless module. It is then directed to the VXLAN interface through the bridge. Upon reaching the VXLAN interface, the packet is encapsulated with a VXLAN header, prepared for transport within a UDP packet via the VXLAN tunnel. Notably, the

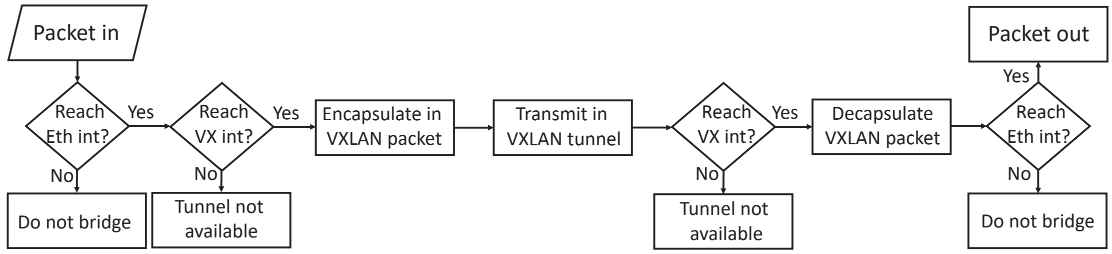


Fig. 3: VXLAN-based packet transmission pipeline.

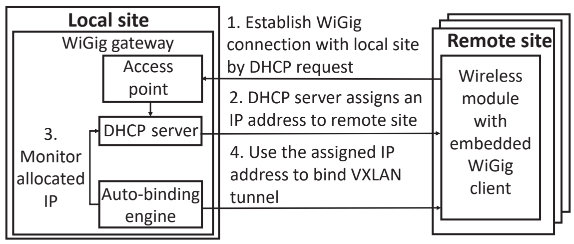


Fig. 4: Auto-binding mechanism workflow.

establishment of the VXLAN tunnel is facilitated by the auto-binding mechanism described in Subsection III-C.

Subsequently, when the packet arrives at the remote site's VXLAN interface, it is decapsulated to the original MAC frame. Importantly, the original MAC frame remains unchanged throughout the transmission process. The final phase of transmission involves the data packet being relayed to its intended destination through the Ethernet interface.

C. Auto-binding Mechanism

Fig 4 illustrates the workflow of the auto-binding mechanism involving the local and remote wireless modules. The goal is to enable dynamic support for remote sites to automatically associate with a local site using VXLAN tunnels. Given an example of wireless backhaul deployment in Fig. 1, when the drone moves into the coverage of the wireless backhaul at the stadium and the disaster areas, the remote wireless module will initiate a request to establish the wireless backhaul bridge with the local site by using DHCP procedures (1). Upon receiving the request, the DHCP server at the local site assigns an IP address to the wireless interface of the remote site (2). Concurrently, the auto-binding engine keeps monitoring the logs of the DHCP server to detect the newly assigned IP address (3). Once the IP address is detected, it then initiates the automatic binding feature to establish the VXLAN tunnel (4); thus, the VXLAN tunnel requires only the IP addresses of the two endpoints without a prior need for their MAC addresses.

IV. Implementation and Evaluation

In this section, we first describe the implementation of the VXLAN-based system. We then evaluate its throughput and latency under various applications compared to the forwarding-based system. The forwarding-based system is set up similarly to the VXLAN-based system but lacks the auto-binding feature. It means that all the routing rules have to be manually configured using the Linux command-line interface.

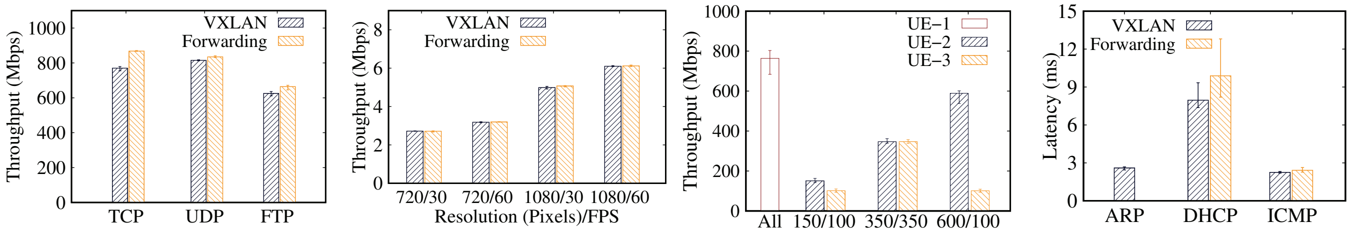
A. Implementation of VXLAN-based system

We developed the prototype of the wireless backhaul system, including the WiGig gateway and the WiGig client, as depicted in Fig. 4. On the former, we use `hostapd` incorporating with WiGig interface for access point and authentication server in master role. The DHCP server `isc-dhcp-server` is configured with the subnet of WiGig interface for dynamically assigning the IP addresses to the clients. Finally, we utilized `python3` to develop the auto-binding engine, which follows the workflow in Section III-C. Furthermore, it maintains an IP pool record to promptly delete VXLAN tunnels when clients disconnect from the server. On the latter, the WiGig client module assumes the slave role. It autonomously scans for the master board access point and establishes a WiGig connection. Another `python3` program monitors the status of the WiGig interface. As soon as the interface is up and assigned an IP address, it immediately triggers the establishment of a VXLAN tunnel to the server.

B. Evaluation comparison

Fig 5 shows the evaluation of the system under different traffic patterns. Fig. 5a illustrates the throughput of TCP, UDP, and FTP traffic. It is observed that the average throughput values for VXLAN-based systems are 770 Mbps, 815 Mbps, and 625 Mbps for TCP, UDP, and FTP traffic, respectively, whereas the forwarding-based's one are 868 Mbps, 841 Mbps, and 672 Mbps, respectively. The reason the proposed system underperformed compared to the forwarding-based is that of the tunneling overhead introduced by the VXLAN header. The VXLAN-based necessitates performing encapsulation and decapsulation for sending and receiving packets, respectively. However, the forwarding-based solution lacks practical flexibility despite its higher throughput, especially for users unfamiliar with Linux command lines.

Fig. 5b illustrates the throughput achieved by the streaming application under varying resolutions and FPS scenarios. It is revealed that the throughput for both VXLAN-based and forwarding-based systems is similar under conditions of lower transmission pressure. For instance, both systems have a throughput of 2.71 Mbps at video stream 720/30. Similarly,



(a) Different applications (b) Different streams (c) Bandwidth control (d) Latency comparison

Fig. 5: Performance evaluation results in comparing VXLAN-based and Forwarding-based systems.

the throughput for 720/60 is 3.18 Mbps for the VXLAN-based and 3.21 Mbps for the forwarding-based; for 1080/30, they are 4.99 Mbps and 5.01 Mbps, respectively; and for 1080/60, they are recorded at 6.10 Mbps and 6.16 Mbps, respectively.

Fig. 5c illustrates the throughput under bandwidth control for the slicing scheme. In this evaluation, we validate the functionality of slicing through two scenarios: One-size-fits-all (All) and slicing. The former represents the throughput measured under maximum allocation in one slice, while the latter represents the bandwidth allocated for two slices to two UEs with a specific ratio. For instance, UE-1 can achieve an average throughput of 764 Mbps, while the results of slicing confirm that it accurately restricts the bandwidth of UE-2 and UE-3 to the assigned throughput ratios: 150/100 Mbps, 350/350 Mbps, and 600/100 Mbps.

Fig. 5d shows the latency between the two systems. The ARP latency observed in the VXLAN-based system is 2.60 ms, whereas ARP requests cannot be responded to in the forwarding method. DHCP incurs the highest latency due to the DHCP server to process the discovery request and respond with the IP allocation to the UE. The DHCP latency for the forwarding-based system is higher than that of the VXLAN-based, at 12.80 ms and 7.95 ms, respectively. The reason is that we additionally use the DHCP relay at the WiGig gateway. Finally, the transmission latency of the VXLAN-based and forwarding-based system are 2.25 ms and 2.63 ms.

V. Conclusion

In this work, we have proposed the VXLAN-based automatic bridging for wireless backhaul systems, designed to deliver a backhaul system that is flexible and compatible with any Linux-based operating system. The proposed system leverages the concept of a VXLAN-based packet transmission pipeline to establish the wireless data tunnels. We realized this concept by implementing the auto-biding mechanism along with WiGig technologies to transmit the FTP and video streaming application. Even though the evaluation results confirm that it is lower in terms of throughput because of the overhead of adding a VXLAN tunnel, the system achieves greater latency with full automation in establishing the wireless backhaul.

Acknowledgment

This work is supported in part by the National Science and Technology Council (NSTC) under Grants No. 110-2221-E-A49-031-MY3, 112-2628-E-A49-016-MY3, 112-2218-E-A49-023, and 112-2634-F-A49-001-MBK.

References

[1] G. M. Insights, “Wireless infrastructure market,” 2023. [Online]. Available: https://www.gminsights.com/industry-analysis/wireless-infrastructure-market

[2] M. T. Lemes, A. M. Alberti, C. B. Both, A. C. D. O. Júnior, and K. V. Cardoso, “A tutorial on trusted and untrusted non-3gpp accesses in 5g systems-first steps toward a unified communications infrastructure,” *IEEE Access*, vol. 10, pp. 116 662–116 685, 2022.

[3] R. Bonk, “The future of passive optical networks,” in *IEEE International Conference on Optical Network Design and Modeling (ONDM)*, 2021.

[4] P. Monti, S. Tombaz, L. Wosinska, and J. Zander, “Mobile backhaul in heterogeneous network deployments: Technology options and power consumption,” in *IEEE International Conference on Transparent Optical Networks (ICTON)*, 2012.

[5] C. Ranaweera, M. G. Resende, K. Reichmann, P. Iannone, P. Henry, B.-J. Kim, P. Magill, K. N. Oikonomou, R. K. Sinha, and S. Woodward, “Design and optimization of fiber optic small-cell backhaul based on an existing fiber-to-the-node residential access network,” *IEEE Communications Magazine*, vol. 51, no. 9, pp. 62–69, 2013.

[6] A. Bekkali, T. Kobayashi, K. Nishimura, N. Shibagaki, K. Kashima, and Y. Sato, “Millimeter-wave-based fiber-wireless bridge system for 8k uhd video streaming and gigabit ethernet data connectivity,” *IEEE Journal of Lightwave Technology*, vol. 36, no. 18, pp. 3988–3998, 2018.

[7] D. Muhammad, G. I. Hapsari, and G. A. Mutiara, “An experimental connectivity performance of simple wireless mesh implementation using wireless distribution system (wds),” *International Journal of Applied Information Technology (IJAIT)*, vol. 01, no. 12, pp. 18–27, 2017.

[8] R. C. Carrano, L. C. Magalhães, D. C. M. Saade, and C. V. Albuquerque, “Ieee 802.11 s multihop mac: A tutorial,” *IEEE Communications Surveys and Tutorials*, vol. 13, no. 1, pp. 52–67, 2010.

[9] M. S. Akhtar, P. Biswas, A. Adhya, and S. Majhi, “Cost-efficient mobile backhaul network design over twdm-pon,” in *IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2020.

[10] C. Ranaweera, C. Lim, A. Nirmalathas, C. Jayasundara, and E. Wong, “Cost-optimal placement and backhauling of small-cell networks,” *IEEE Journal of Lightwave Technology*, vol. 33, no. 18, pp. 3850–3857, 2015.

[11] M. M. Mowla, I. Ahmad, D. Habibi, and Q. V. Phung, “Energy efficient backhauling for 5g small cell networks,” *IEEE Transactions on Sustainable Computing*, vol. 4, no. 3, pp. 279–292, 2018.

[12] IEEE, *IEEE 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*, IEEE Std., 2021.

[13] M. Kafafy, Y. Fahmy, M. Khairy, and M. Abdallah, “Secure backhauling over adaptive parallel mmwave/fso link,” in *IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020.

[14] 3GPP.TS.23.501, *System architecture for the 5G System (5GS)*, 3GPP Std. version 16.6.0 Release 16, 2020.