High precision time synchronization on Wi-Fi based multi-hop network

High Precision Time Synchronization on SDR based Multi-Hop Network

Muhammad Aslam, Wei Liu,
Xianjun Jiao, Jetmir Haxhibeqiri,
Jeroen Hoebeke, Ingrid Moerman
Ghent University - imec, IDLab
Email: {name.surname}@ugent.be

Esteban Municio, Pedro Isolani,
Gilson Miranda,
Johann Marquez-Barja
Antwerp University - imec, IDLab
Email: {name.surname}@uantwerpen.be

***Abstract*—Precise time synchronization amongst network devices is a basic requirement for time critical applications. Despite that time synchronization is a well-established functionality in the wired network domain, its wireless counterpart is very basic or even non-existing. This is because commercial off-the-shelf (COTS) wireless chipsets still focus on basic network connectivity for consumer applications, hence no dedicated software/hardware features are required for time-bounded services in professional wireless environments. This work leverages on openwifi — an opensource Wi-Fi chip design based on Software-Defined Radio (SDR), by adding the support of hardware timestamping in openwifi to support precise time protocol (PTP) application. In addition, openwifi Access Point (AP) is connected to nodes in wired network through a TSN capable switch, and synchroniztion offset between devices in wired and wireless network is measured. Next, the measurement is done with the same setup in wired network but replacing openwifi by COTS Wi-Fi devices. We observe the synchronization offset with COTS is 10^4 fold larger than the offset achieved with openwifi. The experiment setup is in w-iLab.t, an open testbed infrastructure freely accessible for the academic wireless research community.**

I. Introduction

Precise time synchronization amongst network devices is a basic requirement for applications to run in a time sensitive network (TSN). More specifically, Precise Time Protocol (PTP) or IEEE 1588 standard [1] is a widely adopted protocol for achieving sub micro-second time synchronization accuracy in a local network. PTP relies on the timestamps of event packets. Synchronization accuracy depends heavily on the timestamping mechanism. Timestamps can be captured at three levels, i.e. in the PTP application itself, in the driver or in the hardware level of the network interface. The location where the timestamp of a PTP message is captured is referred to as a timestamping point. The closer the timestamping point is to the physical layer (PHY), the less errors are introduced by the time variations when the packet is traveling through the network stack [1].

It is common to have dedicated hardware circuitry and software to assist the capture of the timestamps in Ethernet adapters. Many vendors clearly mention PTP support in their mainstream Ethernet product, such as the Intel I210 controller[1] and 700 series adapters[2], Marvell FastLinQ adapter[3] and Broadcom's Octal-port QSGMII transceiver[4]. However, such a feature barely exist on commercial Wi-Fi card. In fact, Wi-Fi alliance has introduced the "Wi-Fi Timesync" certificate in 2017. To obtain this certificate, a Wi-Fi card should use Fine Time Measurement (FTM) [2] to support PTP, and maintain the synchronization offset within 5.5 μs for 90% of the observation time [3]. To date, only few Intel chipsets (e.g. Intel AX201 card[5]) claimed this certificate. However, the Intel's Wi-Fi card driver and relevant cfg80211 interface in Linux[6] only exchanges round trip time as the result of FTM, without exposing timestamps. Hence the result of FTM can only be used for ranging application in a Linux environment. So, to the best of our knowledge, there is no COTS Wi-Fi chipset venders open the PTP support at hardware or driver level.

This work leverages on openwifi — an opensource Linux mac80211 compatible full-stack IEEE802.11/Wi-Fi design based on Software-Defined Radio (SDR) [4]. The physical layer (PHY) and low Media Access Control (MAC) is implemented in Field Programmable Gate Array (FPGA), whereas high MAC and the network stacks above are provided by Linux. We demonstrate that by adding necessary support in the driver and hardware of openwifi, the Wi-Fi interface can support PTP just like Ethernet interface. High precision time synchronization is achieved using existing PTP software, across multiple hops in wired and wireless network domain.

II. Demo Setup

There are two widely adopted PTP applications in Linux. One is *ptpd* [5] and the other is *Linux ptp* [6]. The former uses software timestamping in application, whereas the latter uses either software timestamping in driver, or hardware timestamping. Thanks to this difference, ptpd can run on any

[1]https://www.intel.com/content/www/us/en/embedded/products/networking/i210-ethernet-controller-family-brief.html

[2]https://www.intel.com/content/dam/www/public/us/en/documents/brief/ethernet-network-adapter-xxv710-product-brief.pdf

[3]https://www.marvell.com/products/ethernet-adapters-and-controllers/41000-ethernet-adapters.html

[4]https://www.broadcom.com/products/ethernet-connectivity/phy-and-poe/copper/gigabit/bcm54382

[5]https://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/wi-fi-6-ax201-module-brief.pdf

[6]Linux cfg80211 interface, retrieved from https://github.com/torvalds/linux/blob/v5.4/include/net/cfg80211.h

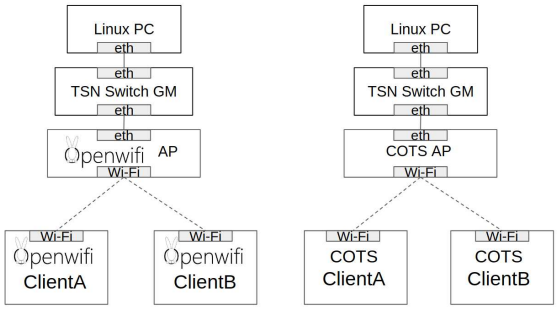

Fig. 1. Experiment setup in w-iLab.t testbed

network interface; whereas linuxptp cannot run on a COTS Wi-Fi card, as they do not provide any timestamping at driver or hardware level. As such, linuxptp offers better performance. Hence, we use *ptpd* on COTS Wi-Fi card, and *Linux ptp* on openwifi interfaces. Both types of PTP software report the synchronization offset from the immediate master, hence we can easily measure and compare the performance when openwifi or COTS devices are involved.

The high level view of the demo setup is shown in Figure 1, deployed in w-iLab.t testbed [7]. A TSN capable switch[7] is used to connect the Linux PC and openwifi Access Point (AP), and it behaves as a PTP Grand Master (GM). A Linux PC operates as a wired node, with its PTP Hardware Clock (PHC) as slave of the TSN switch. The openwifi AP is formed by Zynq UltraScale+ MPSoC ZCU102 Evaluation Kit[8] and AD-FMCOMMS2-EBZ[9] radio frontend. All the Ethernet ports involved are capable of hardware timestamping. The openwifi AP acts as a boundary clock, it has two PHCs, attached to Ethernet and Wi-Fi interfaces respectively. The PHC of Ethernet is slave of the TSN switch, whereas the PHC of Wi-Fi interface acts as the master in the wireless network domain. The two PHCs within the openwifi AP are synchronized by the $phc2sys$ software offered in the *Linux ptp* suite. The two openwifi Clients are associated to the openwifi AP. Their PHCs are the slave of openwifi AP. An openwifi Client is formed by Zynq-7000 SoC ZC706 Evaluation Kit[10] and AD-FMCOMMS2-EBZ board.

In the right side of Figure 1, the openwifi AP and Clients are replaced by Linux PCs with COTS Wi-Fi cards. All devices involved have a backbone port for configuration, managed by the w-iLab.t testbed. For simplicity, the control network is not shown in the figure. The details of openwifi usage in w-iLab.t testbed is further elaborated in [8].

III. Result Analysis

The time synchronization offset is measured in each network domain as described above. The measurement lasts for 300 seconds, the synchronization offset is measured once per second. The 90th percentile of the absolute synchronization offset on each of the network domains is shown in Table I. As seen the accuracy in wired domain is in the order of nanoseconds, thanks to the uni-cast nature of Ethernet and higher resolution of the attached PHC. On the wireless domain, this accuracy is worse, however, by using hardware timestamping in openwifi, we observe that the 90th percentile of the synchronization offset in COTS formed wireless network is 10^4 fold larger than the one formed by openwifi. This shows the addition of hardware assisted timestamping is crucial for the performance. It is also worth mentioning that iperf traffic is sent in parallel with PTP packets between Linux PC and openwifi Clients. We do observe an impact caused by the traffic and by whether the GM is the AP itself or in the wired network, which are potential factors for future improvement.

TABLE I
THE 90TH PERCENTILE OF THE ABSOLUTE SYNCHRONIZATION OFFSET IN NETWORK SEGMENTS FORMED BY COTS AND OPENWIFI.

	openwifi	COTS
ClientA to AP	3.05 μs	83.58 ms
ClientB to AP	3.50 μs	82.70 ms
AP to GM	1.87 μs	99.41 μs
Wired node to GM	582 ns	160 ns

IV. Conclusion

In this work, we demonstrate that by adding hardware timestamping support in a Software-Defined Radio based Wi-Fi interface, high precision time synchronization can be achieved, which reduces the time synchronization offset in an end-to-end connection by 10^4 times comparing to commercial Wi-Fi cards.

Acknowledgment

This work is partially funded by the Flemish FWO SBO S003921N VERI-END.com project. The authors would also like to thank the w-iLab.t admin Vincent Sercu and Pieter Becue for their assistance in the test setup.

References

[1] "Ieee standard for a precision clock synchronization protocol for networked measurement and control systems," IEEE, 2008.
[2] K. Stanton and C. Aldana, "Addition of p802. 11-mc fine timing measurement (ftm) to p802. 1as-rev: Tradeoffs and proposals," *Rev 0.10. IEEE Draft presented at IEEE*, vol. 802, 2015.
[3] "Wi-fi certified timesync technology overview," Wi-Fi Alliance, 2017.
[4] J. Xianjun, L. Wei, and M. Michael. (2019) open-source ieee802.11/wi-fi baseband chip/fpga design. [Online]. Available: https://github.com/open-sdr/openwifi
[5] K. Correll, N. Barendt, and M. Branicky, "Design considerations for software only implementations of the ieee 1588 precision time protocol," in *Conference on IEEE*, vol. 1588, 2005, pp. 11–15.
[6] R. Cochran *et al.* (2015) The linux ptp project. [Online]. Available: http://linuxptp.sourceforge.net
[7] S. Bouckaert, W. Vandenberghe, B. Jooris, I. Moerman, and P. Demeester, "The w-ilab. t testbed," in *International Conference on Testbeds and Research Infrastructures*. Springer, 2010, pp. 145–154.
[8] X. J. Wei Liu. (2019) Openwifi - how to use zynq sdr in linux mode. [Online]. Available: https://doc.ilabt.imec.be/ilabt/wilab/tutorials/openwifi.html

[7]https://www.nxp.com/design/designs/time-sensitive-networking-solution-for-industrial-iot:LS1021A-TSN-RD
[8]https://www.xilinx.com/products/boards-and-kits/ek-u1-zcu102-g.html
[9]https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/eval-ad-fmcomms2.html
[10]https://www.xilinx.com/products/boards-and-kits/ek-z7-zc706-g.html

Demo title:

High Precision Time Synchronization on SDR based Multi-Hop Network

Testbed and demo setup

This demonstration is conducted is IMEC's w-iLab.t testbed[1]. It is situated in a servicing area above a cleanroom, which is a semi-shielded environment without regular human activities and unkown wireless interference. Hence it has a rather controlled and stable environment. The testbed contains a large amount of heterogeneous wireless devices, including Linux PCs with regular Wi-Fi interfaces, Bluetooth dongles, IEEE 802.15.4 sensor nodes, and several Software-Defined Radio (SDR) devices. In this demo, we use 3 Zynq SDR devices and 4 embedded Linux PCs, and a TSN capable Ethernet switch. All involved devices are configured remotely via the testbed backbone network . The topology of the wired network segment is configured via jFED[2] tool. The configuration of SDR devices is introduced in our testbed tutorial [3].

The demo scenario

1/ Start PTP application on all involved devices:

- a/ PTP Grand Master on TSN switch
- b/ PTP Ordinary Clock on regular Linux PC (wired node)
- c/ PTP Boundary Clock (BC) on openwifi AP (zcu102 zynq SDR)
- d/ PTP Ordinary Clock (OC) on openwifi Clients (zc706 zynq SDR)

2/ Time synchronization offsets on different network segment (between AP and Client1, AP and Client 2, and Grand Master and AP) will be displayed via grafana dash board (shown in Figure below) in real time.

3/ Stop the PTP applications on openwifi AP and openwifi Clients (step c and d), replace it by running ptpd over commercial Wi-Fi interfaces

4/ Again display the time synchronization offset as in Step 2 and compare the performance.

Public Availability

This work leverages on openwifi --- an opensoruce Wi-Fi implementation on SDR, this is publicly available on github[4]. Though the feature of precise timestamping is not part of the opensource repository. We encourage the community to reach out to the authors and discuss the possibility of collaboration. All devices involved in the demo are part of the w-iLab.t testbed facility. Users can apply for a testbed account, make reservation for these devices and explore the feature.

---

[1] https://doc.ilabt.imec.be/ilabt/wilab/index.html

[2] Vermeulen, Brecht, Wim Van de Meerssche, and Thijs Walcarius. "jFed toolkit, Fed4FIRE, federation." *GENI engineering conference*. Vol. 19. 2014.

[3] https://doc.ilabt.imec.be/ilabt/wilab/tutorials/openwifi.html

[4] https://github.com/open-sdr/openwifi