CAPTURING PHY INFORMATION FROM CONTROL TRAFFIC FOR HE TRAFFIC SOURCES IN LEGACY SENSORS

Matt Silverman
Amir Kamalizad
Claudius Tewari
Ardalan Alizadeh

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation
Silverman, Matt; Kamalizad, Amir; Tewari, Claudius; and Alizadeh, Ardalan, 'CAPTURING PHY INFORMATION FROM CONTROL TRAFFIC FOR HE TRAFFIC SOURCES IN LEGACY SENSORS', Technical Disclosure Commons, (October 02, 2018)
https://www.tdcommons.org/dpubs_series/1545
CAPTURING PHY INFORMATION FROM CONTROL TRAFFIC FOR HE TRAFFIC SOURCES IN LEGACY SENSORS

AUTHORS:
Matt Silverman
Amir Kamalizad
Claudius Tewari
Ardalan Alizadeh

ABSTRACT

Presented herein are techniques that enable a PHY-level state machine to stitch together PPDUs, in real-time, in a network sensor that is resource constrained including limitation on the receiver decoding capacity. In a software-defined receiver for any incoming frame, a modified PHY state machine is employed to track MU-RTS/CTS, 802.11ax trigger frames, and other 802.11ax traffic to infer the source and destination of the undecodable packets. The modified PHY state machine further captures and ties PHY parameters into the receiver (Rx) path.

DETAILED DESCRIPTION

Sniffers generally allow packets that are not properly decoded to pass up to higher layer software. However, this could happen for packets where the hardware is actually capable of decoding, but fails (e.g., CRC errors). Techniques disclosed herein allow the passing up of any packet regardless of whether the hardware is capable of physically decoding.

Information about the wireless network is collected for the purposes of analytics with monitor radios that scan across channels. The techniques focus on network sensors with constrained resources that do not allow for decoding of high data rate and capture key information from PLCP protocol data units (PPDUs). The techniques also infer missing traffic from surrounding packets.

The existing technology in the field focuses on using sniffer traces and inferring missing data from protocol-based communications. For example, an association response implies that an association request occurred. RTS, CTS, and/or ACK messages may imply that a data frame occurred. These prior techniques only control traffic under pre-
802.11e/n/ac/ax, which covers inferring some information from surrounding traffic based on association requests/responses and certain specific messages.

For a legacy radio, many frames cannot be decoded and/or decrypted for the same reasons as explained above. However, the state machine can be written to include information from 802.11 frames that occur around the same time as a frame the radio is incapable of decoding. Techniques disclosed herein use some common types of frames, such as RTS/CTS, ACK, Block ACK, and Block ACK Request, to infer packet information therefrom.

An example RTS/CTS in IEEE 802.11ax is illustrated below. Figure 1 from the standard shows the messaging between an AP and different stations, where the red cells indicate the lack of certain contents/parameters in a message while the green cells indicate the existence of the particular contents/parameters.

![Figure 1](image.png)

An example of MU-RTS/CTS/DL MU PPDU Acknowledgement Response is shown in Figure 2 below.
An example of MU-RTS/CTS/Trigger/HE trigger-based PPDU/Multi-STA Block Ack is shown in Figure 3.

Also, the IEEE 802.11ax (from draft 3.0) specification indicates that an MU-RTS frame shall not be carried in an HE MU PPDU, and a CTS frame sent in response to an MU-RTS frame shall be carried in a non-HT or non-HT duplicate PPDU. Even though the
AP might not be able to decode HE trigger and trigger-based PPDU, it can infer the addresses of the AP and the clients communicating using 802.11ax HE mode.

According to the IEEE 802.11ax specification, multi-STA BlockAck frame transmissions are allowed in a non-HT Duplicate PPDU, HT PPDU, VHT PPDU, HE SU PPDU, HE-extended range SU PPDU, and OFDMA HE MU PPDU. If the frames are non-HT duplicate PPDU, they can be decoded. The Partial AID information can be tracked over time by monitoring management traffic. Actions can be taken to look for partial AID that appears in AP Multi-STA BlockAck to know all the clients that send uplinks.

Techniques disclosed herein provides MAC layer interference improvements over existing technology. For example, the techniques can handle multi-user (MU) traffic, implementing a version that could infer MU traffic, which is increasingly more common, not just legacy single-user (SU) traffic. The techniques also use IEEE 802.11e Block ACK, which is the majority of ACKs in 802.11 data traffic. Moreover, any 802.11ax/HE-related trigger or control frames, including MU-RTS, MU-CTS, and MultiSta BA, may be used in the techniques.

While existing technology focuses exclusively on operations happening at L2 (MAC) and above, any information from the PHY including undecoded packets is completely missing. As a result, the existing technology relies on retransmissions of the packets to help fill in the missing PHY information, which is not reliable since PHY information may change for those retries. In contrast, the techniques disclosed herein focus on the PHY layer. The In-phase/quadrature (IQ) samples of the PPDU are processed as they come in, but a sample is not decoded if it is undecodable. Certain PHY information including many PHY-level characteristics is captured and used for inference state machine. For example, general PPDU parameters may be captured. From various PHY SIG fields, the captured information may include a data rate, a number of spatial streams, encoding information, a bandwidth, a guard interval, a size of payload, a type of traffic (MU or SU), an HE-symbol length, a duration, etc. From IQ samples, the captured information may include a carrier frequency offset, a Doppler shift, and a duration. Also, location-related parameters may be captured. From raw IQ samples, the captured location-related information may include a received signal strength indication (RSSI), an angle of arrival, and high-precision timestamps.
The techniques enable a PHY-level state machine that stitches PPDUs together in real-time in a single resource-constrained network sensor. In a software-defined receiver for any incoming frame, a modified PHY state machine is employed to track MU-RTS/CTS, 802.11ax trigger frames, and other 802.11ax traffic to infer the source and destination of the undecodable packets. The modified PHY state machine further captures and ties PHY parameters into the receiver (Rx) path. The software-defined receiver for any incoming frame wraps in information from the surrounding control frames. For example, under a more generalized 802.11ht/vht concept, the techniques can substitute in 802.11ax version of frame where applicable.

To decode MU-RTS (Request-To-Send) packets, duration IDs and destination MAC addresses are saved in anticipation of a possible data frame that is not decodable. MAC addresses in the IP address table are looked up in advance to find the associated IP address available from any previous frame identifying the IP address for the source address and/or destination address. A timer is set to search for multi-device CTS (Clear-To-Send). Network Allocation Vector (NAV) information is obtained. A timeout for the timer can be:

\[
\text{TimestampCtsTimeout} = \text{Timestamp}_\text{RTS} + T_{\text{SIFS}} + T_{\text{RTS duration}} + \text{timeoutMargin}
\]

A timer can be set to search for data frame RTS. For example, a timer can be set as follows:

\[
\text{TimestampDataTimeout} = \text{Timestamp}_\text{RTS} + 2* T_{\text{SIFS}} + T_{\text{RTS duration}} + T_{\text{CTS duration}} + \text{timeoutMargin}
\]

Moreover, if CTS is fully decoded prior to the timeout in search for CTS, the duration ID and destination MAC address are saved in anticipation of a possible data frame that is not decodable. MAC addresses in the IP address table are looked up in advance to find the associated IP address available from any previous frame identifying the IP address, and the IP address is saved for DA. The DA MAC address is compared to that of SA of
RTS for sanity. The timer for data packet search is then updated, and NAV info is obtained from CTS. A timeout for this timer can be:

\[
\text{TimestampDataTimeout} = \max(\text{TimestampDataTimeout}, \text{Timestamp}_{\text{CTS}} + T_{\text{SIFS}} + T_{\text{CTS\_duration}} + \text{timeoutMargin})
\]

If the search for CTS is timed out, timeout for data is performed. In some cases, CTS is captured but the RTS that the CTS is responding to is not heard. The duration ID and destination MAC address are then saved in anticipation of a possible data frame that is not decodable. The MAC address in IP address table is looked up in advance to find associated IP address available from any previous frame identifying the IP address. And the IP address is saved for DA. The timer for data packet search is updated as:

\[
\text{TimestampDataTimeout} = \max(\text{TimestampDataTimeout}, \text{Timestamp}_{\text{CTS}} + T_{\text{SIFS}} + T_{\text{CTS\_duration}} + \text{timeoutMargin})
\]

To handle HT/VHT/HE-Data, if the data frame is fully decoded, the full contents of data frame can be used such that control frames are not needed. Information, such as duration, RSSI, format, bandwidth, and other PHY/MAC-level information, can be populated into RxInfo.

If a data frame cannot be decoded for any reason and the time for the data frame search from RTS or CTS has not run out, the following steps can be performed. First, the PHY format (HT, VHT, HE, etc.) and parameters that cause decoding failure are detected. PHY parameters including location-related parameters are estimated using the information from the L-SIG of the failed data frame, and the NAV information from the RTS, CTS, or both to determine the length of the data frame. An HT/VHT/SIG-A1/2/HE-SIG frame is decoded and relevant contents are saved into RxInfo/packet header for the frame. While attempting to decode the frame, IQ samples, RSSI and/or auto-correlation can be employed to determine symbols that can track if the PPDU is obtained. A timeout is set to look for the block ACK or ACK or Block ACK request. For example, a timeout can be:
ackTimeout = timestampData + T_SIFS + T_estimateData + timeoutMargin

The techniques disclosed herein also provides methods to handle ACK or block ACK (BAs) or Multi-sta BA messages. If a prior data packet is not decodable and the clock is less than the ackTimeout and the ACK/BA message is decodable, the ACK or BA message is decoded and the payload is copied into a buffer location of undecoded data frame. The MAC header addresses are indicated as being swapped by the host data path. The ownership of the buffer spot for data frame is relinquished so Rx data path consumes and releases the buffer spot for future packets. The full contents of Rx ACK or BA go through normal decoding and are sent to the Rx data path. Also, if clock $\geq$ ackTimeout or the previous data frame is decodable, the full contents of Rx ACK or BA go through normal decoding and are sent to the Rx data path.

In summary, the techniques described herein use control traffic to infer critical information about undecodable frames that the control frames surround, which allows cheaper IoT sensors to monitor network traffic that they are not physically capable of decoding while still capturing PHY-level information of the missed PPDUs.