Poster: A Raspberry Pi Based Data-Centric MAC for Robust Multicast in Vehicular Networks

Mohammed Elbadry, Bing Zhou, Fan Ye, Peter Milder, YuanYuan Yang
Stony Brook University
Department of Electrical and Computer Engineering
{mohammed.salah,bing.zhou,fan.ye,peter.milder,yuanyuan.yang}@stonybrook.edu

ABSTRACT

Data-centric networks provide content instead of address (what vs. where) based communication primitives, and have been argued to be the proper candidate for data dissemination in high mobility vehicular networks (e.g., delivering road side accident video clips to affected drivers in both directions). However, current Medium Access Control (MAC) layers filter incoming frames based on destination addresses, not content. The data-centric network community has resorted to MAC broadcast, with high and greatly varying frame loss rates. We propose *V-MAC*, a data-centric MAC layer that filters frames by content. It supports one to many multicast at MAC level, and ensures a uniform and controllable small frame loss rate across all receivers, despite their varying reception qualities. We have created a *V-MAC* prototype using Raspberry Pis and WiFi dongles. Experiments under extremely noisy environment show that it reduces frame loss from 50% (broadcast) to less than 10%, and consistently among multiple receivers.

CCS CONCEPTS

• **Networks → Link-layer protocols**; **Mobile networks**; *Cross-layer protocols*; *Ad hoc networks*;

KEYWORDS

vehicular communication; mobile network; data-centric networks

ACM Reference Format:
Mohammed Elbadry, Bing Zhou, Fan Ye, Peter Milder, YuanYuan Yang. 2018. Poster: A Raspberry Pi Based Data-Centric MAC for Robust Multicast in Vehicular Networks. In *The 24th Annual International Conference on Mobile Computing and Networking (MobiCom '18), October 29-November 2, 2018, New Delhi, India.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3241539.3267759

1 MOTIVATION

More sensors in both quantity and variety (e.g., multiple cameras, radars, IMUs) are instrumented to connected/autonomous vehicles. Disseminating such sensor data is extremely valuable for the safety, efficiency and comfort of driving. E.g., video clips of a road side accident captured by vehicles passing by can be disseminated to drivers impacted in both directions, keeping them in cognitive awareness to reduce stress, and enabling them to make decisions of alternative routes. Such sharing needs are *data-centric*: drivers want data of certain content (e.g., accidents, speeds along a road segment) but care little about who produced the data.[1]Data-centric networks have been argued to meet such needs. Exemplified by Named Data Networking (NDN) [1], a consumer sends an Interest describing the attributes of desired content, and whoever have matching data, either produced locally or received then cached, can return the data.

Current wireless Medium Access Control layers remain address-centric. People have resorted to MAC broadcast [2], which passes all incoming frames to the network layer. This results in high and varying frame loss rates at different receivers, and greatly delayed actions (e.g., retransmission of missing frames), making efficient and consistent data dissemination difficult to achieve.

2 V-MAC DESIGN

We describe the design of *V-MAC*'s two major components: name encoding and DACK (Data-centric Acknowledgment).

2.1 Name Encoding

We assume a hierarchical naming convention combining different attributes to name data (e.g., */Accident/Nicolls-Road/34th-st/35th-st/20180710/11am-12pm/video-clip1/ min0/frame11* denotes the 11th video frame during the first minute of a video clip of an accident captured on Nicolls Road, between 34th

[1]We assume data fidelity/integrity are addressed by relevant mechanisms [1].

and 35th Street, on 7/10/18, between 11AM to 12PM). Following a common naming convention, consumers can send interests to describe desired data.

V-MAC receives Interests passed down from the local network layer. It filters incoming Data frames by name matching: those carrying names matching what are specified by Interests are forwarded up to the network layer; otherwise they are dropped. To avoid carrying long hierarchical names in frames, we propose an encoding that compresses possibly long, varying size names to a short, fixed size. First, the prefix part that denotes data granularity at a certain number of MAC frames is hashed. E.g., */Accident/Nicolls-Road/34th-st/35th-st/20180710/11am-12pm/video-clip1/min0/* is hashed to 16 bits. We denote each data unit at such granularity a *service stream*. The remainder of the name at granularity of MAC frames is assigned sequence numbers.

The network layer maintains an encoding table that facilitates the mapping and translation between hierarchical string names and encodings, when data are passed between network and V-MAC layers. Although the amount of possible hierarchical data names can be infinite, the number of service streams being actively requested by drivers along any given road segment at any time, is limited (e.g., a few hundred to thousand). Thus the table and encoding can be of much smaller sizes and sufficient to distinguish different active service streams.

2.2 Tunable Frame Reception Quality

V-MAC supports tunable robustness for frame reception by DACK, a data-centric acknowledgment. A typical MAC uses either per frame acknowledgment for unicast, or no acknowledgment for broadcast, and neither can achieve the desired low, tunable and consistent frame reception at multiple neighboring receivers.

DACK provides aggregated feedback from all neighboring receivers over multiple frames. To this end, we introduce the concept of a *"frame burst"*, which is a certain number (e.g., 5) of consecutive frames in a service stream. When a sender transmits a burst, neighbors do not send per-frame acknowledgment. This allows transmission of frames in short contact durations (~10s) to mobile nodes. Since multiple neighbors may each miss similar or different subsets of frames, DACK provides aggregated feedback through two mechanisms: *periodic* and *triggered*.

Periodic DACK. Periodic DACK addresses missing of similar frames in the most recent burst by multiple receivers. After a burst finishes transmission, each receiver prepares a DACK frame containing the sequence numbers of missing frames (for multiple consecutive missing frames, the first/last sequence numbers only, or no sequence number if all are received), and starts a timer, upon whose expiration the DACK will be sent. To avoid multiple DACKs being sent, the timer expiration is computed such that the receiver missing the most frames will send DACK first. Upon hearing another's DACK, a receiver cancels its timer and refrains from transmitting its DACK this round. This technique avoids DACKs carrying redundant frame sequence numbers, and is observed that one DACK addresses most of missing frames.

Triggered DACK. Periodic DACK provides feedback only for the latest burst. When reception quality is low, some receivers may still miss frames earlier than the latest burst. This can also be caused by loss of DACK. Each receiver keeps a sliding window the size of multiple bursts (e.g., 4 most recent bursts). If at any point the number of missing frames in this window exceeds a threshold, a receiver will create a DACK and send it. This threshold number is the tuning knob for frame loss rate: when a receiver misses more than this threshold fraction of frames in the window, a DACK is produced.

Retransmission Buffer. On the sender side, V-MAC keeps a sliding window per stream of recently transmitted frames in a retransmission buffer. Upon hearing DACK, it retransmits missing frames from the buffer, without a request from the network layer reducing latency. The buffer size is kept at the same size of the sliding window on the receiver side so that requested frames can be retransmitted.

3 IMPLEMENTATION

We implement *V-MAC* in two versions: Linux implementation using Raspberry Pis and WiFi dongles, and ns-3 simulator for controlled environments and verification of design.

Linux. We implement *V-MAC* on Raspberry Pi by modifying the existing Linux mac80211 module, used by WiFi drivers that have the MAC implemented in software (i.e. SoftMac). External USB WiFi dongles (Alfa AWUS036NHA due to its easiness of changing the SoftMac) are used with antennas removed to attenuate the signal (Figure 1). We leverage existing callbacks from user space, while retaining the beaconing for synchronization among nodes.

We describe what *V-MAC* does for each frame type for transmitting/receiving paths (Figure 2). In transmission, an Interest passed from upper the layer is transmitted to solicit matching data from other nodes; it also creates an entry in the *streams* table for matching incoming Data. Data frames are directed to *tx_controller* logic which builds the header first by examining the *streams* table. Then, the frame is copied into the retransmission buffer before being sent.

On the receiving path, *V-MAC* frames are first identified by a bit in the header through *rx_controller*. An Interest is forwarded to the network layer which will process and decide what actions to take. Data and DACK frames require further matching against names in the *streams* table. Matching

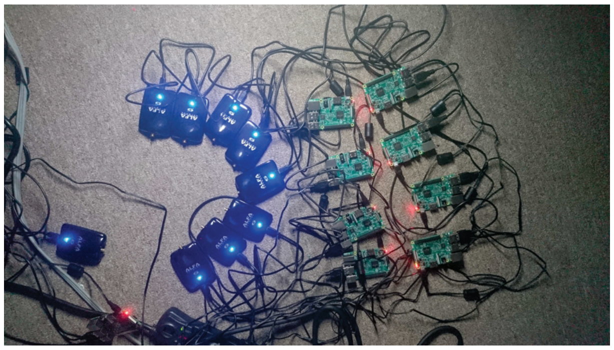

Figure 1: *V-MAC* testbed.

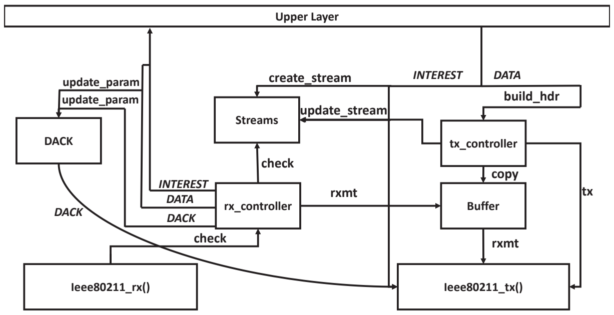

Figure 2: *V-MAC* Linux design.

Data frames are forwarded to the network layer, and trigger parameter updates for DACK (e.g., update latest frame sequence number.). For DACK, a sender will retrieve missing frames from the *Buffer* and retransmit; a receiver will update its own DACK operations.

ns-3. In addition to our Raspberry Pi-based evaluations, we have used ns-3 to perform simulations where we control noise, simulate larger networks, and nodes with high mobility. The simulation produced similar results as of Pi prototype. Due to space limitations, we show Linux results only.

4 EVALUATION

We compare *V-MAC* with ad-hoc broadcast in a 9-node testbed where one sender transmits 2000 frames to 8 neighboring receivers, using the base frame rate. Each receiver has different distance/orientation to the sender, thus different reception qualities. After one trial of 2000 frames in ad-hoc broadcast, we immediately switch to *V-MAC* for another trial, without moving anything. Thus the reception qualities and background traffic will likely remain similar in such short time.

The results for frames lost using adhoc mode broadcast are shown in Figure 3. We can observe that the frame losses are around 50% for trials 1-3, 7, while much lower (~10-20%) for trials 4-6. There are also a few very low or high (7%, 99%) cases. This shows that loss rates in broadcast are affected greatly by background traffic, and they vary greatly across different times or receivers. However, *V-MAC* is consistently low (~10%) at all times and for all receivers, regardless of the varying wireless reception quality (Figure 4). This shows the effectiveness of the DACK's tunable robustness mechanisms.

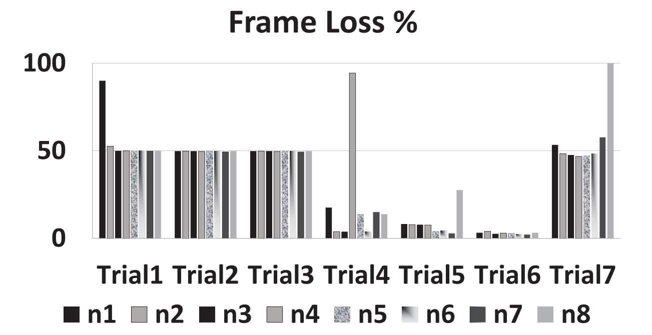

Figure 3: Cumulative frames lost for broadcast per receiver over time.

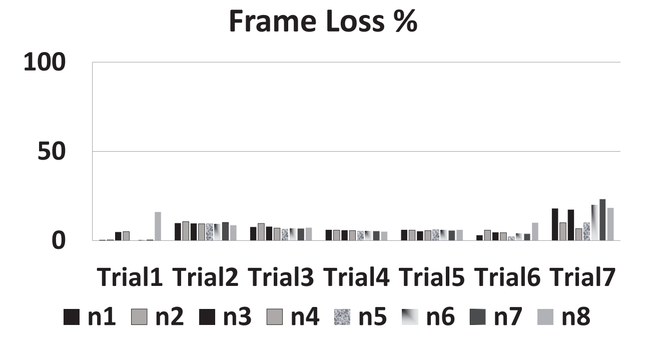

Figure 4: Cumulative frames lost for *V-MAC* per receiver over time.

5 CONCLUSION

The prototype demonstrates the effectiveness of V-MAC for content based filtering and tunable, consistent robustness. We plan to further study frame rate adaptation, larger scale of networks and high mobility using simulator and Pi/dongles on real vehicles.

6 ACKNOWLEDGEMENTS

This project is supported in part by NSF 1730291.

REFERENCES

[1] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang, et al. Named data networking.

[2] Uichin Lee, Joon-Sang Park, Joseph Yeh, Giovanni Pau, and Mario Gerla. Code torrent: content distribution using network coding in vanet. In *Proceedings of the 1st international workshop on Decentralized resource sharing in mobile computing and networking*, pages 1–5. ACM, 2006.