Demo: Video over Synchronous Flooding with OSFv6

Michael Baddeley*, Markus Schuß§, Yevgen Gyl†,
Monika Prakash*, Xiaoyuan Ma‡, and Carlo Alberto Boano§
*Technology Innovation Institute, AE; †Unikie, FI; ‡SKF Group, CN; §Graz University of Technology, AT
{michael.baddeley; monika.prakash}@tii.ae, yevgen.gyl@unikie.com,
ma.xiaoyuan.mail@gmail.com, {markus.schuss; cboano}@tugraz.at

Abstract

In recent years, a considerable body of literature has shown how Synchronous Flooding (SF)-based communication can provide reliability and latency guarantees in multi-hop networks. However, to-date, the technique has failed to gain traction outside of the research community. While disappointing, this is perhaps not surprising. Many of the key works in SF have focused on improving performance gains or proposing tailored protocols, with little thought to the end-user or wider networking ecosystem. This demo addresses this gap and presents `OSFv6`, an IPv6-compatible extension to the Open Synchronous Flooding (OSF) framework, that connects SF networks to the wider IP ecosystem and allows users to seamlessly run socket applications between host machines. We benchmark `OSFv6` across the four BLE physical layers and show that, using the 2 Mbps physical layer (PHY), `OSFv6` can achieve up to *250 kbps at 3 hops*. Importantly, we use `OSFv6` to transmit a live video stream between two hosts – demonstrating the application for SF networks outside of use-cases envisioned in current literature.

Keywords

Video, BLE, Low-Power Wireless, Embedded, Concurrent Transmissions, Synchronous Flooding, Mesh Networks.

1 Introduction

Over the previous decade there has been notable interest in the application of concurrent transmissions (also referred to as synchronous transmissions) to provide highly reliable, time-synchronized flooding protocols in low-power mesh networks [3]. Many of the works on Synchronous Flooding (SF) – as this technique is known – have focused on employing the high reliability and low latency characteristics of SF to address real-time challenges for industrial sensor/actuator networks [8]. Baumann et. al. [4], for example, have demonstrated that SF protocols can be used to provide wireless control for time-critical applications whilst leveraging the inherent mobility advantages provided by a flooding-based (rather than routing-based) approach, while works such as [6, 1] have shown how SF can be used to reliably disseminate control messaging alongside traditional Internet of Things (IoT) protocols. However, despite such successes, the technique is relatively unknown outside of a core community, and has yet to be championed as part of a standard or considered within industry. Indeed, while long-standing standardization efforts in the field of sensor networks have made it easy for applications to be run on top of an IP-compliant networking stack (notably, through 6LoWPAN and 6TiSCH standards), to-date there has been no similar effort within the SF community. This lack of interoperability with the IoT ecosystem presents, among other challenges, a considerable barrier-to-entry for the wider adoption of SF-based protocols.

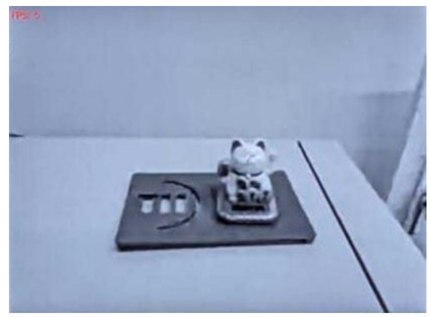

Figure 1: `OSFv6` can provide throughputs of up to 250 kbps at 3 hops – sufficient for low resolution, low frame rate video.

IPv6 over Synchronous Flooding. We address this gap with `OSFv6`, an open-source[1] IPv6 extension to the Open Synchronous Flooding (OSF) [2] framework for multi-PHY SF protocols. Specifically, `OSFv6` considers the specific traffic requirements of UDP and TCP traffic and makes changes to the OSF MAC driver to allow integration with the Contiki-NG IPv6 networking stack. We deploy `OSFv6` on `nRF52840` [7] devices and provide a SLIP bridge to a Linux-based host, allowing IPv6-based applications to run over the SF-based network. We show that, depending on the underlying PHY, `OSFv6` can achieve throughput of up to *a few hundred kbps* between two host machines. Moreover, we demonstrate that this throughput is sufficient to support the transmission of low-resolution video over a multi-hop SF-based network.

[1] `https://github.com/open-sf/osf`

PHY	Mean UDP Throughput
BLE 2M	253 kbps
BLE 1M	162 kbps
BLE 500K	84 kbps
BLE 125K	26 kbps

Table 1: `iPerf3` between 2 nodes (3 logical hops).

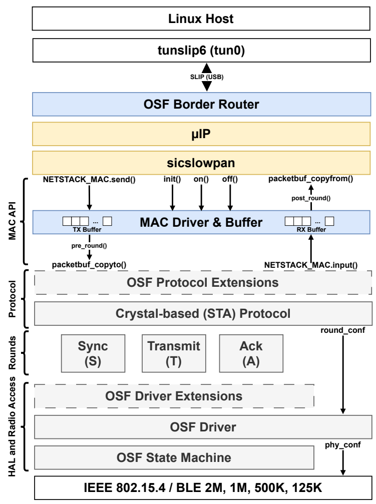

Figure 2: `OSFv6` stack. Modified OSF components are highlighted in blue, while Contiki-NG components are in yellow.

2 Supporting IPv6 in OSF and Contiki-NG

Fig. 2 provides an overview of `OSFv6`. Original components are shown in gray, while modified/additional components are highlighted in blue, and Contiki-NG components are highlighted in yellow. Importantly, `OSFv6` fully integrates the OSF MAC Driver & Buffer into the Contiki-NG `NETSTACK` and makes changes to the `osf_send()` and `osf_receive()` functions so they are compliant with the `NETSTACK_MAC` interface. At the SF protocol layer, `OSFv6` employs an STA (Sync., TX, ACK) protocol based on Crystal [5], allowing many-to-many communication between any network device. As with Contiki-NG's IEEE 802.15.4 TSCH implementation, `OSFv6` must wait for a TX timeslot rather than immediately sending egress packet from higher layers (as is the case with other MAC approaches). Similarly, there is insufficient time between the first RX slot and subsequent TX slots within an OSF round to immediately process ingress packets. `OSFv6` therefore utilizes buffers to store TX/RX packets from/to higher layers. This allows the `sicslowpan` layer to push to and fetch from those buffers in processes before/after an OSF round where, to support the larger BLE MTU, Contiki-NG `packetbuf` and `sicslowpan` fragment sizes are increased to 255 bytes rather than IEEE 802.15.4's 127 bytes. Finally, an OSF border router establishes SLIP over USB to Contiki-NG's `tunslip6`, which creates a layer 3 tunnel interface on the host. It should be noted that rather than receiving an IP from the host, node IPs (as well as the routes to each host) are generated through a statically declared deployment table – in essence, every node on the `OSFv6` network acts as if it is one hop away at layer 3. Table 1 shows the mean UDP throughput across each of the four OSF BLE PHYs in a 2-node setup where OSF was configured for 3 transmission slots in each round ($NTX = 3$) – effectively a 3-hop logical network.

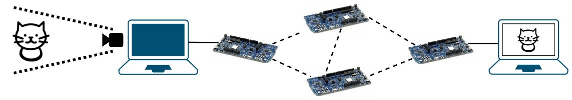

Figure 3: `OSFv6` video streaming demo setup.

3 Video over Synchronous Flooding

`OSFv6`'s ability to support TCP/UDP socket-based protocol options opens SF networks to the possibility of supporting new and exciting user applications. Indeed, given that `OSFv6` can achieve up to a few hundred kbps on the uncoded BLE physical layers, this throughput is more-than sufficient for low resolution, low frame rate video. To this end, we demonstrate that `OSFv6` is capable of streaming video across a logical 3-hop SF network (i.e., a network with an OSF configuration of $NTX = 3$). Fig. 3 provides an overview of the demo setup. On the left, a camera-capable laptop records a (constantly moving) figurine. Using a UDP client/server application, this video recording is streamed over UDP as raw JPG frames across the SF network to the laptop on the right where, as shown in Fig. 1, the video is displayed in monochrome at a resolution of 320x240 at 5 frames per second. While transmitting video over low-rate BLE physical layers is perhaps extreme, it successfully shows IPv6-capable SF networks could potentially be used in other applications such as the transmission of audio, as well as supporting protocols such as HTTP, MQTT, and Fast-DDS.

4 Conclusions

While standardization efforts such as 6LoWPAN have extended the 'IoT stack' into the smallest edge devices for over a decade, there has been a notable lack of support for this capability in SF-based protocols and solutions. `OSFv6` addresses this gap by extending the original OSF framework to support IPv6 applications from connected host devices. While there exists a number of open challenges – namely IP assignment for SF networks with multiple border routers to host devices – we have shown it is possible to achieve end-to-end throughput of up to *a few hundred kbps* between hosts over an SF network. Furthermore, we have practically demonstrated that these rates are sufficient for streaming video, opening up multi-hop SF-based networks to IPv6-capable applications running on host devices or external networks.

5 References

[1] M. Baddeley et al. 6TiSCH++ with Bluetooth 5 and Concurrent Transmissions. In *Proc. of the* 18^{th} *EWSN Conf.*, 2021.
[2] M. Baddeley et al. OSF: An Open-Source Framework for Synchronous Flooding over Multiple PHYs. In *Proc. of the 18th EWSN Conf.*, 2022.
[3] M. Baddeley et al. Understanding Concurrent Transmissions: The Impact of Carrier Frequency Offset and RF Interference on Physical Layer Performance. *ACM TOSN*, 2023.
[4] D. Baumann et al. Fast Feedback Control over Multi-Hop Wireless Networks with Mode Changes and Stability Guarantees. *ACM TCPS*, 2019.
[5] T. Istomin et al. Data Prediction + Synchronous Transmissions = Ultra-Low Power Wireless Sensor Networks. In *ACM SenSys*, 2016.
[6] T. Istomin et al. Route or Flood? Reliable and Efficient Support for Downward Traffic in RPL. *ACM TOSN*, 2019.
[7] Nordic Semiconductors. nRF52840 Product Specification., v1.7, 2022.
[8] M. Zimmerling et al. Synchronous Transmissions in Low-Power Wireless: A Survey of Communication Protocols and Network Services. *ACM CSUR*, 2020.