Low Overhead, Fine-grained End-to-end Monitoring of Wireless Networks using In-band Telemetry

Jetmir Haxhibeqiri*, Ingrid Moerman* and Jeroen Hoebeke*
* IDLab, Ghent University – imec, Ghent, Belgium
Email: [firstname.lastname]@ugent.be

***Abstract*—Wireless networks are becoming more complex while applications on top are becoming more demanding. To maintain network performance in terms of latency, throughput and reliability, continuous verification of the performance, possibly followed by on-the-fly network (re)configuration is needed. To achieve this, the way wireless network monitoring is being done needs to be reconsidered and should evolve towards more timely, low overhead and fine-grained monitoring. This paper shows how in-band network telemetry (INT) monitoring can achieve these objectives. An INT-enabled node architecture is designed as well as novel INT options. By means of an implementation on WiFi Linux devices, the concept is validated by tracking the behavior of a real network.**

***Index Terms*—INT, WiFi, SDN, Industry 4.0.**

I. Introduction

Wireless networks are becoming increasingly complex and thus harder to be managed by human administrators. This is also witnessed by several trends in wireless networking, like Software Defined Networking (SDN), 5G, IEEE 802.11ax, etc., that all introduce more advanced (re)configuration capabilities to meet application demands. Considering this growing complexity, more automated network management should become a reality. According to a market study on network automation [1], network monitoring tools are considered as one of the drivers to achieve such network automation.

Networking monitoring is not a new topic. Recently it has evolved from monitoring for network troubleshooting purposes to monitoring for verifying network configuration. This requires flexible, fine-grained, reliable and timely monitoring information from the network. On top, the network monitoring should only introduce limited overhead in the (wireless) network. Moreover, it should offer a full view on the network considering both wireless and wired parts.

Today, different types of network monitoring exist, including traffic probing and network equipment polling. The first technique determines the network performance by sending special packets between two entities, while the later polls the network entities for traffic statistics. Both of these approaches yield additional traffic that needs to travel over the network for monitoring purposes. Consequently, this might affect the data traffic, creating performance issues, or may lead to misdetection of the actual performance experienced by the data traffic itself. A final drawback of such techniques includes the absence of flow specific monitoring information.

In-band Network Telemetry (INT) is a new approach that offers low overhead in-band network monitoring possibilities. Monitoring is done on a per-packet basis by letting each node on the path adding telemetry data regarding the data packet itself. This new approach overcomes the drawbacks of the previous approaches: there is no need for probing packets, the telemetry information is collected on a per-packet basis showing the network performance of the actual data traffic, and there is no need for polling network equipment.

Recently a draft proposal for INT standardization was issued by the Internet Engineering Task Force (IETF). The draft includes the telemetry data format [2] and INT encapsulation for different protocols [3]. However, currently all the standardization focuses on supporting INT for wired networks.

In this paper, we study the feasibility and benefits of INT for wireless networks. We present an INT enabled node architecture for performing INT on both end devices and wireless equipment. We specify the INT hop-by-hop option for collecting wireless telemetry information. We implement the proposed INT enabled node architecture in Linux based WiFi devices and validate the proposed design on real hardware.

The remainder of this paper is structured as follows. In section II we motivate the use of INT for wireless communication and list a number of use cases where INT is beneficial. Section III gives a short overview of how INT for wired networks is currently being standardized by IETF. In section IV the design of our INT enabled node architecture is discussed along with its implementation. Section V validates the implementation. It focuses on validating the monitoring of different wireless parameters in a real test-bed network. Finally, section VI concludes the paper and gives some possible future works.

II. Towards INT for Wireless

In traditional networks, network monitoring was triggered by the need for network problem troubleshooting. Different tools (*ping* or *zing* [4]) were used to determine traffic latency and end-to-end reliability. Other approaches consisted of polling the network devices to retrieve various statistics. These approaches introduce additional traffic that might affect the data traffic or they require separate control links.

In wireless networking an in-band network monitoring approach will be more beneficial. First of all, it decreases the traffic overhead, removes the need for control links and introduces the possibility for higher monitoring granularity. Such an approach is not limited to network troubleshooting only, but also to more frequent network monitoring for verification of network configurations.

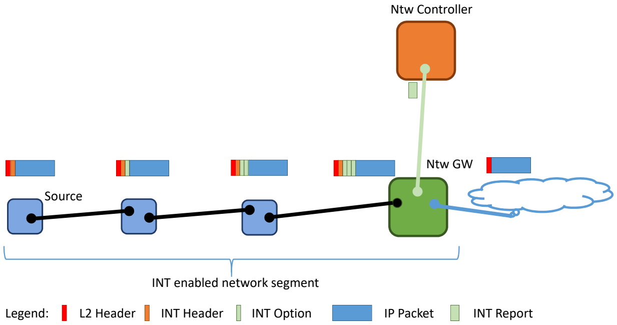
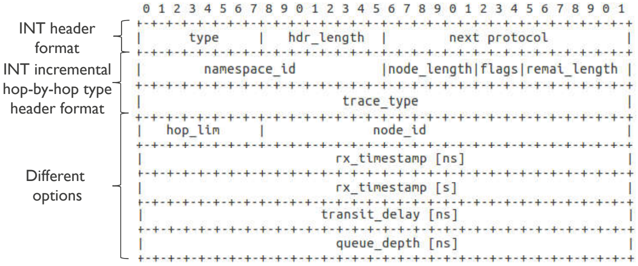


Fig. 1. Simple INT enable network example.



```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      type     |   hdr_length  |         next protocol         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        namespace_id           |node_length|flags|remai_length |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           trace_type                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   hop_lim     |                    node_id                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       rx_timestamp [ns]                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       rx_timestamp [s]                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      transit_delay [ns]                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       queue_depth [ns]                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 2. INT information packet.

In Industry 4.0, network monitoring is used to verify whether performance metrics are met, in order to avoid any down times. In such use cases, the in-band collection of measurements is required. In different process control loop or emergency applications, strict end-to-end latency requirements should be met at any time. Using INT the latency information can be monitored on a per-hop basis. This will enable to precisely determine the latency bottleneck in the network.

Also robotic applications can benefit from INT. Because of the network dynamics, an active monitoring approach will not give full insights in the network performance. The end-to-end path will change over time and active measurements will not give information per-packet and per-hop. Contrary, INT measurements are able to determine the parts of the network and the timing when performance issues occurred.

As a last example, also home networks can take advantage of INT. Network providers usually guarantee the network performance up to the access point, leaving the the performance of connections within your wireless home network out of scope. Different wireless based audio streaming services, video streaming, wireless range extenders, meshed WiFi devices and other WiFi enabled devices can perform differently depending on the specific environment. Using INT, performance monitoring within the home network and terminated at the border of that network, can make it easier for end users to troubleshoot and check the performance of their home network.

III. In-band Network Telemetry Background

INT is a new way of network monitoring where monitoring information is appended directly to the data packets. A simple INT enabled network example is shown in Figure 1. The source node that generates the data packet attaches the initial INT header that includes a certain bit-map vector. The bit-map vector specifies all types of information that should be collected on each hop. Each node along the path will add the requested information without affecting the application payload. The destination node or the INT termination node (network gateway) extracts the monitored information collected at each hop. This information can be used by other network entities, such as a network controller, network optimization entity or traffic visualizer entity.

The INT specification was initiated by the P4 language consortium in 2016 [5]. Now, INT standardization is continuing under the name of In-situ Operations, Administration, and Maintenance (IOAM) by the network working group of the IETF. It resulted in a number of drafts regarding different topics: the data fields for IOAM [2], Ethernet encapsulation of IOAM [3] and proof of transit [6]. Throughout this paper INT and IOAM will be used interchangeably.

The INT header format [3] is composed of the INT header type, INT header length and the protocol that follows the INT data option fields as shown in Figure 2. The INT header type, based on its value, specifies different INT types (hop-by-hop type, proof of transit type or end-to-end type). The header length contains the length of the INT header. The last field contains the protocol that follows the INT data fields.

The INT header is followed by the INT type header. In Figure 2, the INT header is followed by the incremental hop-by-hop INT type header. The namespace ID identifies the INT namespace that is known to each node in the network. In absence of a certain namespace ID, a node is not allowed to add or process the telemetry data to/of the packet. The node length field specifies the data length that each node will add as a multiple of 4-octets. The flag bits are used to specify if the packet has overflown (there is no space left to further add telemetry data), if the packet needs to be forwarded back to the source (loop-back bit), or if the telemetry data needs to be processed immediately at every INT enabled node in the network. The remaining length shows the remaining length of data that can be added by the intermediate nodes before the INT data options are considered to be overflown. The trace type field specifies the data types to be collected for the packet by each INT enabled node along the path. This paper will focus on hop-by-hop telemetry data.

IV. INT-enabled Node Architecture

Our design and implementation, as outlined in this section, targets INT in wireless networks, focusing on WiFi.

A. Telemetry Options

The design enables the collection of telemetry data related to wireless links. To achieve this, a new hop-by-hop option is defined, the presence of which is indicated by bit 12 in the hop-by-hop trace type bit vector.

Depending on the openness of the wireless card's driver, various information about a wireless link can be collected, including: RSSI, SNR, MCS, contention window size, channel

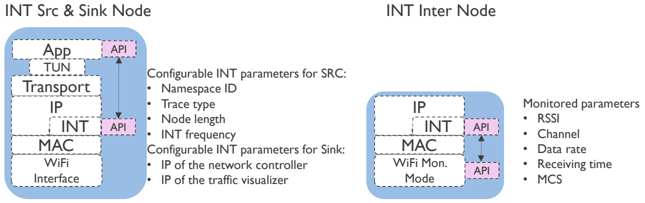


Fig. 3. INT enabled node architecture.

info, data rate and reception time. For now the wireless option is composed of three fields: RSSI, data rate and channel info. The reception time is not included, as it is already part of an existing option. The *channel* field specifies the wireless channel at which the link is operating, the *rssi* field specifies the received signal strength of the packet, while the *dr* field specifies the data rate used.

Currently, we use layer 2 encapsulation of INT information, with the INT header and option data field residing between the layer 2 and layer 3 header. This way, also switches can process the INT information, while the sink nodes should process the monitoring data right after L2 header processing. As a downside, packet processing problems will occur in case the INT enabled packet traverses a router that is not INT enabled.

B. Node Architecture and Proof of Concept (PoC) Implementation

The INT enabled node architecture is shown in Figure 3 for different node roles. The INT *layer* resides between layer 2 and layer 3 and is responsible to add/extract/process the telemetry info based on the role of the INT enabled node.

As shown in Figure 3, different INT parameters (namespace ID, trace type, node length and INT frequency) can be controlled using an API socket at the source node. The trace type field can be adjusted based on the application requirements, while the namespace ID is used to specify the flow ID. In addition to this, the granularity of the INT data can be changed on the fly by decreasing/increasing the INT period.

An intermediate node will access the wireless link information from the wireless card itself and create the wireless option header. In case the telemetry information is not accessible or not available, the node will add 0xFFFFFFFF as option value.

Finally, the INT sink node can extract the INT data and send the data to different entities in the network (network controller or visualizer). Various APIs can be used to communicate between the INT sink and controller(s), e.g. push/pull, broker-based communication and request/reply. The INT sink node can concatenate a number of telemetry measurements in order to decrease the traffic towards the network controller(s).

The Click Router framework [7] is used to implement a proof of concept (PoC) of our proposed INT enabled node architecture. Click Router is a modular software router toolkit that enables packet processing in user-level or kernel-level space [7]. In addition to our custom Click router extensions, we implemented three new Click elements representing different node roles: INT source, INT intermediate and INT sink.

Click router exposes a TUN interface towards the application layer. The application layer sends all traffic through this TUN interface. Once the packets have been processed by the Click framework and their next hop has been determined, they are processed by the INT layer. The INT layer will add the INT header between layer 2 and layer 3.

At each hop the packet length is increased by appending new INT data with the PHY parameters taken from the monitor mode. The INT data collection can be terminated either at the destination node or at a gateway node at the border of an INT enabled network segment. The INT sink node will process all the INT information and create a JSON data structure that includes the information in a structured way.

V. Validation

To validate the proposed design and implementation, we first evaluate the INT overhead to data packets. Secondly, we conduct a number of measurements in a real office environment. The measurements encompass the detection of the wireless network behaviour based on the collected INT data.

A. INT overhead

The usage of INT does not come without packet overhead. The INT overhead increases with the number of hops the packet traverses in the network. Nevertheless, it is minimal compared to active probing as no additional packets have to be generated. The increase in packet size due to INT as a function of the number of communication hops can be calculated according to the following formula:

$$\begin{aligned} INT_{overhead} &= 4 + 8 * INT_{HBH} \\ &+ INT_{HBH} * (h * 4 * TRACE_{LGTH}) \\ &+ INT_{ETE} * (4 + 4 * E2E_{OPTIONS}); \end{aligned} \tag{1}$$

where INT_{HBH} is 1 when the hop-by-hop header is present, h is the number of hops the packet has passed, $TRACE_{LGTH}$ is the number of set bits in the trace type byte vector, INT_{ETE} is 1 when the INT end-to-end header is present and $E2E_{OPTIONS}$ is the number of options in the end-to-end INT header. The unit of the formula is in bytes.

To monitor the performance of wireless links, we have to enable at least three bits in the trace type: the node ID option, the wireless telemetry option and the hop-by-hop reliability option. Next to this, the end-to-end header should be present with the counter option, that is used to detect end-to-end reliability. Thus equation (1), will become: $INT_{wireless} = 20 + 12 * h$.

In case when probes are used, there will be an additional probe request/reply packet at each wireless link. For WiFi, this is further increased by two layer 2 ACKs, 2 RTS and 2 CTS packets (assuming RTS/CTS is used). Only the layer 2 overhead to transmit the probes over a single link will be: 2*20 bytes L2 ACKs, 2*20 bytes for RTS and 2*20 bytes for CTS, 60 bytes in total. For a ping request/reply packet, the ICMP payload can be as low as 20 bytes, resulting in a total of 60 bytes, counting for IPv4 and layer 2 headers

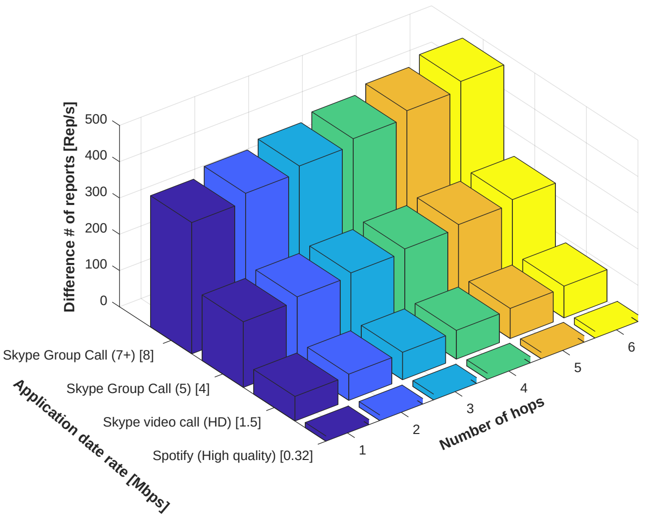


Fig. 4. Difference between number of INT packets and probing packets that can be generated for different application throughput requirements.

too. So, over a single link, an additional 180 bytes is sent if active probing is used, compared to only 32 bytes for INT. This is around 6 times less overhead. The main benefit of using INT in terms of overhead is that it does not introduce additional wireless channel accesses, avoiding any additional L2 packets. In order to see the benefit of using INT, we check which telemetry granularity, i.e. number of reports per second, INT can offer compared to traditional probing techniques for a given application data rate. For this, we perform the following calculation. First, we determine the additional network capacity that is being consumed when every data packet is augmented with INT data, assuming the maximum IEEE 802.11 MTU of 2304 bytes and a given application data rate. For the second case, in the absence of performing INT, we assume this capacity is used to perform active probing, but at the expense of generating more packets. This way, we end up with two monitoring approaches that result in the same additional network capacity consumed, but that differ in the number of telemetry reports. This is shown in Figure 4. It can be seen that for highly intensive application (like Skype group call) we can send between 300 to 450 more INT reports per second compared to probe reports. When the application data rate requirements are lower, the difference decreases too.

B. Real time network monitoring

To validate our INT layer implementation, we monitored our office WiFi network for 24 hours. Our office environment has a number of dual band APs. We used two different nodes to send traffic to each other every second. INT telemetry data was added to the data packets every 10 seconds. As the same network was used by other users during the day, we wanted to see which insights the INT monitoring data would reveal.

Figure 5 shows the monitored RSSI, used channel and data rate over the measurement period of 24 hours for the downlink traffic of the first node. It can be seen that in the late afternoon and during the night, when the data traffic in general is lower, the RSSI value was ∼ -65 dBm and channels in the 5 GHz band were used. On the other hand, during the day between 1 pm and 4 pm, the RSSI value was ∼ -83 dBm. Mostly, a data rate of 18 Mbps was used, expect when the end node switched to the 2.4 MHz band. The switch from 2.4 GHz to 5 GHz band happened due to unlicensed LTE traffic experiments for LTE/WiFI coexistence that were going on. Even though the RSSI value at 2.4 GHz was higher, due to another interfering technology (LTE) the end node decided to switch to the 5 GHz band, which is a normal behaviour of the WiFi client.

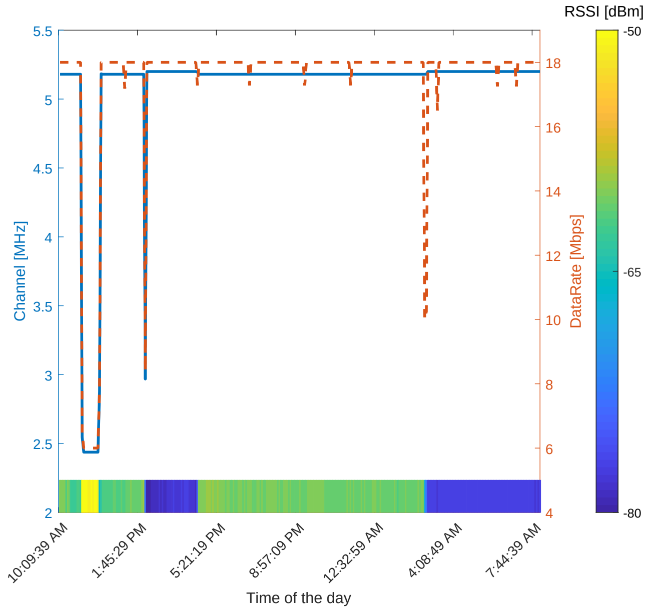


Fig. 5. Real time monitoring of a wireless network. Monitoring channel, data rate and RSSI value.

VI. Conclusions

An INT enabled node architecture was presented, that is able to collect wireless information on a per-hop basis. The implementation uses the Linux Click router framework. We showed that various information related to the wireless interface can be collected, including RSSI, MCS, used channel as well as packet reception time. Regarding the INT overhead, INT has around 6 time less data overhead compared to active probing over a single hop. This benefit increases with the number of hops. This study can be extended by designing a full INT enabled network architecture. To this end, APIs between the network nodes and network controller should be defined as well as data structures. In terms of INT implementation improvement, a loop-back option should be implemented to support appending the INT report to data packets.

Acknowledgment

Part of this research was funded by the FWO-Flanders under grant agreement #G055619N.

REFERENCES

[1] V. Bhalla and S. Ganguli, "Market guide for network automation," *Gartner*, 26.03.2018.

[2] F. Brockners, S. Bhandari, C. Pignataro, H. Gredler, J. Leddy, S. Youell, D. Mozes, T. Mizrahi, P. Lapukhov, R. Chang, D. Bernier, and J. Lemon, "Data fields for in-situ oam, draft-ietf-ippm-iom-data-05," 2019.

[3] B. Weis, F. Brockners, C. Hill, S. Bhandari, V. Govindan, C. Pignataro, H. Gredler, J. Leddy, S. Youell, T. Mizrahi, A. Kfir, B. Gafni, P. Lapukhov, and M. Spiegel, "Ethertype protocol identification of in-situ oam data, draft-weis-ippm-ioam-eth-01," 2019.

[4] A. Adams, J. Mahdavi, M. Mathis, and V. Paxson, "Creating a scalable architecture for internet measurement," *IEEE Network*, 1998.

[5] C. Kim, P. Bhide, E. Doe, H. Holbrook, A. Ghanwani, D. Daly, M. Hira, and B. Davie, "Inband network telemetry (int)," *Accessed 15-04-2019.*, 2016.

[6] F. Brockners, S. Bhandari, S. Dara, C. Pignataro, J. Leddy, S. Youell, D. Mozes, T. Mizrahi, A. Augado, and D. Lopez, "Proof of transit, draft-ietf-sfc-proof-of-transit-02," 2019.

[7] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM Transactions on Computer Systems (TOCS)*, vol. 18, no. 3, pp. 263–297, 2000.