FTM-ns3: WiFi Fine Time Measurements for NS3

Anatolij Zubow, Christos Laskos, Falko Dressler
Technische Universität Berlin, Germany
{zubow, laskos, dressler}@tkn.tu-berlin.de

***Abstract*—Wireless networks having location-awareness will foster new classes of applications. Here, a precise range estimation is of critical importance. The Fine Time Measurement (FTM) protocol introduced in IEEE 802.11-2016 uses radio frequency based time-of-flight estimation, which enables precise indoor ranging and positioning. This paper presents `FTM-ns3`, an extension for the widely used ns3 network simulator to support the 802.11 FTM protocol. Using results from experiments with real FTM hardware, we developed error models to be used in simulations. These new error models allow to analyze the impact of channel bandwidth and Line of Sight (LOS) with multipath channel propagation on the performance of FTM-based localization schemes. First results from own simulations show that even a simple localization scheme is highly affected by ranging inaccuracy introduced due to multipath propagation in typical indoor environments even so a LOS component exists. Our extension is provided to the community as open source.**

***Index terms*—** WiFi Localization, IEEE 802.11, Fine Time Measurements, FTM, Network Simulation, NS3

I. Introduction

In recent years, we have seen a great interest from research and academia in making wireless networks to be location-aware. An indoor localization system (ILS) based on existing and already deployed IEEE 802.11 (WiFi) infrastructure would be very promising as indoor localization might become ubiquitous to any device equipped with a WiFi chip (e.g., smartphone, tablet) like the global positioning system (GPS), which is used outdoors. However, such an ILS needs to be accurate, deployable, and universal [1]; often also making use of machine learning to overcome technical weaknesses [2].

Recently, a Fine Time Measurement (FTM) protocol was standardized in IEEE 802.11-2016 [3]. FTM uses radio frequency based time-of-flight (ToF) estimation, which allows to develop precise indoor positioning schemes. Some major WiFi chipset vendors like Intel have already released devices that support the FTM protocol. However, first experimental results show disappointing performance especially in environments with strong multipath [4], [5], [6]. This was confirmed by our own experiments using the Intel 8260 (802.11ac) chips.

Studying the impact of such FTM-based protocols still depends on labor-intensive field experiments. To make the protocol more accessible and to foster further work on FTM improvements, it is necessary to not only implement FTM in a simulation toolkit but also particularly enable accurate error models to cope with the multipath-related inaccuracies.

In this paper, we present a standard compliant implementation of the WiFi-FTM protocol in the ns-3 network simulator [7]. The ns–3 is the *de facto* standard for academic and industry research into networking protocols and communications technology. Moreover, based on the results from extensive lab experiments with real FTM hardware, we developed empirical error models, which allow to analyze the impact of channel bandwidth and channel propagation characteristics on the performance of FTM-based ranging or localization schemes in a controlled simulation environment. This enables the quick development of novel FTM-based localization schemes in a well-controlled simulation environment. The `FTM-ns3` software package together with examples is provided to the community as open source.[1]

Our main contributions can be summarized as follows:

- We present `FTM-ns3`, a novel FTM extension for the ns3 network simulator;
- we study the most relevant factors influencing the precision of FTM-based ranging;
- we introduce an empirical error model based on extensive lab experiments; and
- we evaluate our `FTM-ns3` system in proof-of-concept study for ranging and localization.

II. 802.11 FTM in a Nutshell

This section gives a short overview of the Fine Time Measurement (FTM) protocol as defined in the 802.11 standard [3]. The FTM protocol is used for performing high accuracy ranging between two stations (cf. Figure 1). In order to mitigate the effect of clock synchronization error it uses a two-way time transfer (TWTT) protocol as two-way ranging (TWR) method. In an FTM exchange, one station is the initiator while the other is the responder. The FTM exchange starts with having the initiator sending an FTM request to the responder. In this request, the initiator transmits the parameters of the FTM session to be created. When the responder receives the request, it can either accept the request, change the parameters or deny the session. If the responder has accepted the initiator's request, an FTM session will be created between the two stations and ranging measurements can start. After the measurements have been performed, the session is closed. It is important to note that only the initiator of a session can determine the round-trip time (RTT).

The authors acknowledge the financial support by the Federal Ministry of Education and Research (BMBF, Germany) in the programme of "Souverän. Digital. Vernetzt." Joint project 6G-RIC, project identification number 16KISK030 as well as by the Federal Ministry for Economic Affairs and Climate Action (BMWK, Germany) within ZIM Programme, Grant 16KN073726.

[1] https://github.com/tkn-tub/wifi-ftm-ns3

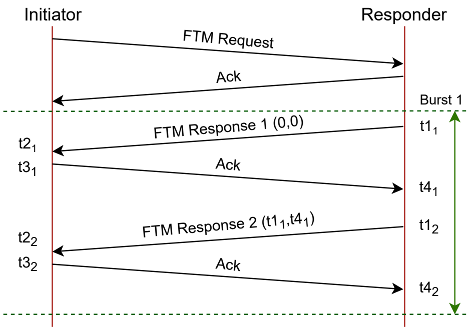


Fig. 1: An FTM exchange with one burst.

High accuracy ranging in FTM is achieved by taking precise timestamps at the physical layer in picoseconds (ps) resolution, which gives them an accuracy of 0.03 cm. According to the standard [3], the timestamps should be taken as soon as the start of the preamble has been detected to make these timestamps as accurate as possible.

As shown in Figure 1, the time measurement starts with the first frame the responder sends. As the session has just begun, the responder has no previous timestamps to transmit in the first FTM. Thus, both timestamps in the dialog number 1 are set to 0. A dialog consists of the FTM response sent by the responder and the ACK sent by the initiator. The timestamps taken by the responder are $t1$ and $t4$. The first is taken immediately before the responder starts transmitting its FTM response while the second represents the point in time when it receives the corresponding ACK frame. The initiator determines the timestamps $t2$ and $t3$, which represent the point in time the FTM response was received and the initiator started transmitting the corresponding ACK frame respectively. These four timestamps form an FTM dialog for which the RTT can be calculated. It is important to note that only $n-1$ RTT of n FTM exchanges can be calculated. The RTT is calculated as

$$\mathrm{RTT} = (t4 - t1) - (t3 - t2) \tag{1}$$

The distance in cm between two stations is obtained as (note, that the RTT is given in ps)

$$d = \frac{\mathrm{RTT}}{2} \times 0.03 \frac{\mathrm{cm}}{\mathrm{ps}} \tag{2}$$

There will always be some fluctuation in the estimated RTT due to the limited bandwidth and environmental influences (cf. Section III). Hence, in order to get higher accuracy, averaging is performed over the RTT values obtained within a session.

A WiFi station can have multiple active sessions with different stations. The stations do not need to be associated with each other in order to perform ranging. Instead, it is possible to perform an FTM exchange with a station in an unassociated state.

III. Factors influencing the precision of Ranging

There are many factors that have an impact on the accuracy of ToF-based ranging used in FTM, which are described in the following.

A. Channel Bandwidth

The detection of packet arrivals is a challenging task as a difference of only 1 ns could result in an error of 30 cm for the RF ranging system. Therefore, a fine resolution clock with 1 ns or higher is needed which is the case with FTM as it uses ps clock resolution. Another factor that limits the resolution of a ToF measurement is the sampling rate [8] and the channel bandwidth. This is known as range binning, which occurs when a matched filter is used to estimate the time of packet arrival with a sampling rate of up to $f_s = 2B$ where B is the channel bandwidth [9]. Sampling adds error to the estimate because the estimate space is divided up into range bins that are c/f_s wide where c is the speed of light. Sampling adds uniform range uncertainty in each bin of σ_s^2 [9]:

$$\sigma_s^2 = \frac{c^2}{12 f_s^2} \tag{3}$$

In the case of WiFi, with sampling at $1/B$, $B = 20$ MHz, the variance due to sampling can be calculated to be $(4.68\,m)^2$. Continuous tracking, filtering, or averaging can be used to improve the resolution, but this is not bandwidth or power efficient. To reduce this error, the signal can be oversampled. To further improve the raw resolution super-resolution spectral signal processing techniques are being used today, e.g., [10]. Finally, the channel bandwidth itself can be increased as newer WiFi standards like 802.11n/ac/ax support 40, 80, or even 160 MHz.

In order to understand the ToF range estimation accuracy of commodity 802.11 hardware we performed our own experiments in the lab. As WiFi hardware we used the Intel 8260 NIC, which is compliant to 802.11b/a/g/n/ac. The WiFi devices were configured to operate in 2.4 GHz ISM band and use the 802.11n configuration. The setup consisted of two WiFi nodes where we replaced the antennas with coax cable with 50 dB attenuator. Such a configuration represents the operation at high signal strength, i.e., RSSI at around -40 dBm, mimicking perfect channel conditions without any distortions like multipath propagation. For our FTM ranging experiment, we used the software provided by Gruteser.[2] We tested two different bandwidth configurations, i.e., 20 and 40 MHz. For each configuration 2560 FTM ranging tests were performed. The reported RTT values were post-processed.

The results are shown in Figure 2. Here the value of each RTT sample was corrected by the mean value over all measured RTT values. This is because we are not interested in the absolute values which dependent on the length of the coax cable. Instead we want to analyze the variations in RTT due to different channel bandwidth. From the results we see that the distribution of the RTT values follows a normal distribution. Its standard deviation is around two times larger for 20 MHz channel as compared to a 40 MHz channel, i.e., 2563 ps and 1075 ps, respectively. Using Equation (2), this translates into a distance of 38 cm and 16 cm respectively.

[2] http://www.winlab.rutgers.edu/~gruteser/projects/ftm/Setups.htm

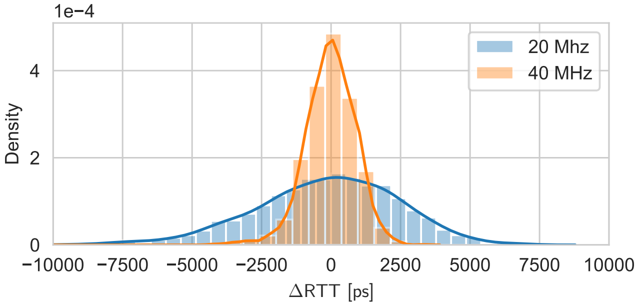


Fig. 2: Ranging error in over-the-cable experiment

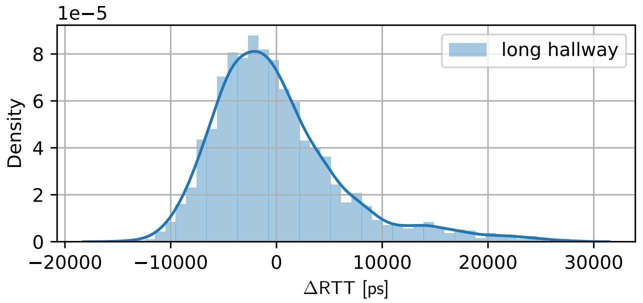


Fig. 3: Ranging error in long hallway (data from [4])

B. Multipath Propagation

In a typical scenario, indoor or outdoors, the wireless signal emitted by the transmitter is exposed to multipath propagation. Here the RF signals bounce off objects in the environment, causing the signal to arrive at the receiver through many paths. The consequence is that the received signal is the sum of the signals arriving along different paths. Except for the direct (Line of Sight, LOS) path all paths are the result of reflection and diffraction. Compared to LOS signal, the non-LOS signals are delayed and the phase and amplitude of the signal is different. The result of multipath is a randomly changing received signal power which is termed as fading.

Multipath propagation distorts the ToF-based ranging used by FTM. First, in absence of a LOS component, i.e., in a pure non-LOS environment, the measured distance typically overestimates the actual distance. Second, it is not uncommon that the indirect paths have higher power than the direct path [11]. Third, the multipath propagation complicates the proper detection of packet arrival time resulting in too late or too early detection.

Results from experimental studies analyzing the ToF range estimation accuracy using commodity 802.11 hardware (Intel 8260), e.g., [5], [6], [12], [13], [14], show that under real indoor conditions the FTM ranging error is around 1-2 m (RTT of 6.6-13.3 ns) which is 2.6-5.2× larger than the error we obtained with our wired setup (cf. Section III-A) and, thus, cannot be explained with the limitations due to the channel bandwidth.

It is very likely that this is because of the transmission over a real wireless channel having multipath propagation. Therefore, we took a closer look at the results obtained by Jathe et al. [4], who performed ranging experiments indoors in a long hallway. The propagation was characterized by having a clear LOS with additional multipath components. We used their provided dataset to compute the ranging error of each data point. As the reported distance values are mean values calculated over 20 FTM rangings, the impact of channel bandwidth was already averaged out. First, we converted both the ground truth distance and the measured distance into RTT (Equation (1)). Next, the RTT error is computed by subtracting the ground truth RTT from the measured RTT. With this subtraction an underestimation of the distance is visible as a negative value and an overestimation of the distance as a positive value. Afterwards, the mean RTT error is calculated and added to each RTT error value for compensation. This is required as the hardware was not calibrated [6].

The resulting dataset is shown in Figure 3. As can be seen, the distance error is no longer normally distributed. Instead, it has an exponentially modified normal distribution[3], i.e., sum of independent normal and exponential random variables, with parameters $\mu = -5478$, $\sigma = 2821$, and $\lambda = 0.000183$. Looking at the resulting error model, it is clear that its peak is negative, at about -2000 ps. Meaning that there is a tendency to underestimate the true distance by 30 cm. Also it is possible that sometimes the distance is overestimated significantly. This is likely caused by the multipath propagation in the indoor environment, resulting in detecting the signal too late or in the former case, too early.

IV. FTM Extension for NS3

In the following, we present our `FTM-ns3` system. We implemented it in form of an ns-3 extension to support WiFi-FTM. We discuss the design, the supported error models to account for channel bandwidth and multipath propagation, and the actual implementation.

A. Design

The goal of our work is to extend the WiFi module of the ns-3 network simulator to support the FTM protocol in a standard compliant way. Therefore, we added the support for FTM to the `RegularWifiMac` class. The actual logic for handling FTM requests and responses and the corresponding FTM sessions is provided by the `FtmManager` class. Each Wifi node has exactly one `FtmManager` instance if support for FTM has been enabled. Every `FtmSession` has its own parameters and can have an error model defined in `FtmErrorModel` if specified. Different FTM sessions can use different error models. The provided error models and their limitations will be discussed in next section.

Figure 4 shows the class diagram of the main FTM components. The extension of the `RegularWifiMac` class is kept as minimalistic as possible. The `RegularWifiMac` class receives all action frames and processes them accordingly, including

[3] The Kolmogorov-Smirnov test for goodness of fit gives a p-value of 0.86.

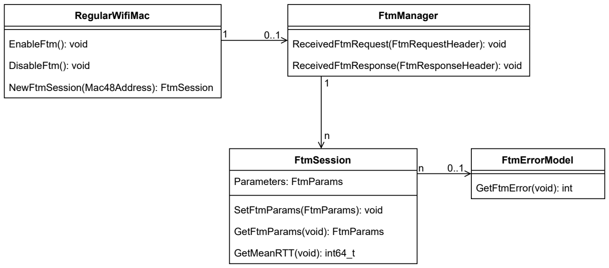

RegularWifiMac
EnableFtm(): void
DisableFtm(): void
NewFtmSession(Mac48Address): FtmSession
1
0..1
FtmManager
ReceivedFtmRequest(FtmRequestHeader): void
ReceivedFtmResponse(FtmResponseHeader): void
1
n
FtmSession
Parameters: FtmParams
SetFtmParams(FtmParams): void
GetFtmParams(void): FtmParams
GetMeanRTT(void): int64_t
n
0..1
FtmErrorModel
GetFtmError(void): int

Fig. 4: Class diagram showing the components of `FTM-ns3`.

FTM frames. If an action frame with category value public action is received, it is further processed to determine if it is an FTM request or response frame. In such a case it is handed over to the `FtmManager` to determine to which session it belongs. If the frame does not belong to an existing session and it is an FTM request, a new session is created. Finally, the received FTM frame is forwarded to the correct `FtmSession` by the manager.

B. Modelling Ranging Errors

`FTM-ns3` supports two error models which can be parameterized. The first one is the wired error model which models the impacts of channel bandwidth on ranging accuracy. Currently two bandwidths are available to select during simulations: 20 and 40 MHz. The model is based on our observation from experiments over coax cable (cf. Section III-A). The RTT error is drawn according to normal distribution with $\mu = 0$, i.e., zero mean, and standard deviation of $\sigma = 2563\,\mathrm{ps}$ and $\sigma = 1075\,\mathrm{ps}$ for 20 and 40 MHz, respectively.

The second one is a wireless error model which extends the wired error model to additionally model the impact of multipath propagation which is of great importance especially when performing simulations of indoor scenarios. This model is based on the observation made in Section III-B. First, the impact from multipath on the FTM accuracy introduces a bias (RTT offset) which depends on the locations of transmitter and receiver. That means that as long as the nodes' locations are fixed the introduced RTT offset stays the same and does not change over time. Therefore, the wireless error model needs to know the position of the WiFi nodes to which it is attached (cf. Figure 4). This position is given to the `GetBias` function of the `FtmMap`, which returns the bias for a given position. The `FtmMap` stores the pre-computed RTT bias for each possible node location.

The precomputed RTT bias is generated as follows. The RTT bias of closely positioned WiFi nodes is typically observed similar or correlated. Therefore, the bias can be obtained via interpolation in the following way. A uniformly spaced grid is generated using a pre-defined de-correlation distance d, with e.g. $d = 25\,cm$ when using 2.4 GHz carrier frequency. For each grid point a random value according to exponentially modified normal distribution with $\mu = -5478$, standard deviation of $\sigma = 2821$ and $\lambda = 0.000183$ is generated (cf. Section III-B). The values between the grid points are generated by interpolation using a cubic spline with a resolution of 1 cm which is sufficient from practical point of view. After adding to the RTT the bias obtained from the `FtmMap`, the error calculated by the wired error model is added as well.

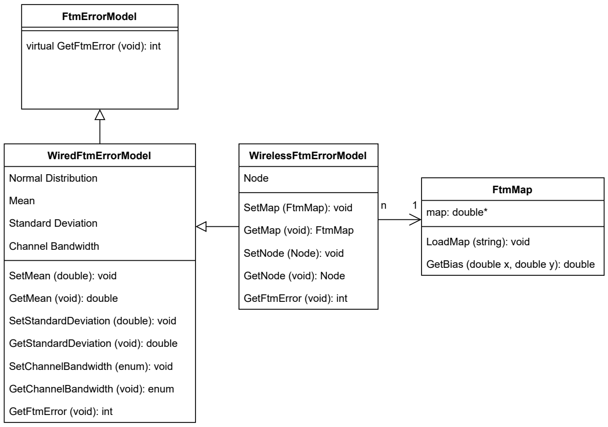


Fig. 5: Class diagram of the ranging error models.

In summary, when using the wireless error model the RTT in the simulation, $\widetilde{\mathrm{RTT}}$, is computed as follows:

$$\widetilde{\mathrm{RTT}} = \mathrm{RTT} + h + w, \tag{4}$$

where RTT is the ground truth RTT as computed by Equation (1), h and w are random variables representing the RTT errors due to multipath fading and channel bandwidth, respectively. Here w follows a normal distribution whereas h is exponentially modified normal distributed. Note, while w changes over time even in stationary setup it is not the case with h which changes over space. Both effects are additive.

Figure 5 shows the class diagram. While `WiredFtmErrorModel` class directly inherits from the `FtmErrorModel` base class, the `WirelessFtmErrorModel` is a subclass of `WiredFtmErrorModel`. All error models implement the `GetFtmError` method, which is used by the `FtmSession` to obtain the error for each RTT measurement. First, the RTT is calculated according to Equation (1). Thereafter the RTT error returned from the `GetFtmError` method is added to it to account for ranging inaccuracy. It is important to note, that the `FtmSession` is unaware of the error model currently being used. As shown in Figure 4, the `FtmSession` holds only a reference to the base class `FtmErrorModel`. By default the base class is used by every session, which always returns zero RTT error.

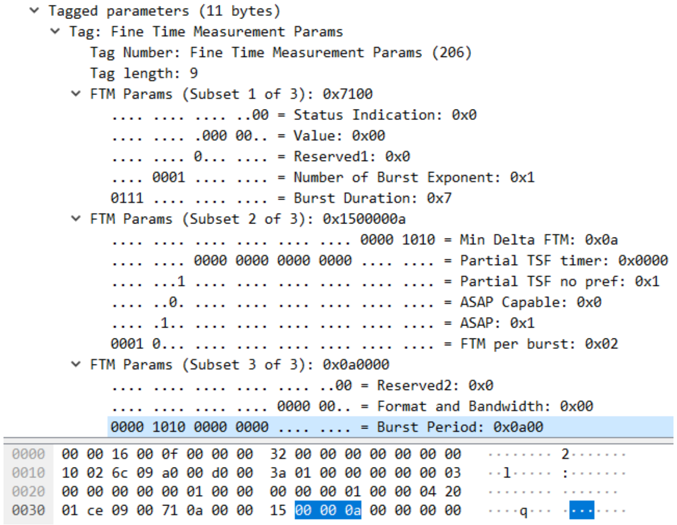
C. Implementation

In this part, we will give some implementation details of `FTM-ns3` like FTM framing, usage of time stamps, integration with 802.11 PHY/MAC layers. For the FTM implementation, the support of action frames in the 802.11 module of ns3 is required. The module supports action frames but is limited to block ack, mesh, multihop, self protected and vendor specific action. Hence we extended it to also support public action frames, in which only the FTM request and FTM response are supported. For this support, the `WifiActionHeader` class has been extended. All of the required FTM specific frames, like request, response, parameters and TSF sync, have been implemented and can be found in the ’ftm-header’ files. The FTM request and response headers are implemented as defined by the 802.11 standard. The TSF sync info header also complies with the standard and is being transmitted in the first frame of every burst instance. It is not used actively and always has a value of 0, because time is always synchronized in the simulator. It is added to comply to standard and to have an accurate overhead representation of the FTM protocol. The FTM parameters header is also implemented as defined in the standard but two of its fields are used differently. The partial TSF timer is a time value specified in ms and the initiator indicates to the responder when the FTM measurements should begin. For example when the TSF timer value is set to 10, it means that the measurements should begin in 10 ms. This was done out of simplicity and because initiator and responder always have synchronized time. The other is the format and bandwidth field, which is not used and is always set to 0. All of the other fields are used as they are defined in the standard.

Next, we discuss the integration of FTM to the 802.11 PHY layer. It is needed in order to retrieve the time stamps of incoming and outgoing packets related to an FTM measurement. The way this is done, is by connecting the `FtmManager` of each WiFi node to the `PhyTxBegin` and `PhyRxBegin` callbacks of the `WifiPhy`. These callbacks are fired when the preamble of a packet has either been successfully transmitted or received. The main difference to the definition in the standard is, that the time stamps are set after the preamble has been received. This leads to some inaccuracies by having the preamble detection period in the calculated RTT. To remove this delay, the preamble detection duration is subtracted twice during RTT calculation. It is removed twice, because two frames are transmitted for each measurement, the FTM response and its Ack. The FTM retransmissions have been handled in a way that an FTM frame can be transmitted as many times as needed. This is possible because the time stamps are renewed every time the frame is transmitted or received, even if the dialog token already exists and has time stamps. In this case the time stamps for that dialog will be overridden with the newest ones, but the ones transmitted in the frame using the followup dialog token are not set again if they have already been set at the initiator. This makes handling retransmissions very simple.

In order to analyze the correctness of our FTM protocol implementation, we replicated the simple scenario shown in Figure 1 with two WiFi nodes where one node was triggering FTM ranging. We enabled the tracing functionality of ns3 in order to capture all transmitted frames during the simulation. Those traces we later analyzed using the Wireshark tool. As can be seen from Figure 6 Wireshark was able to correctly display the FTM request frame.

Our software implementation of FTM extension for ns3 together with examples is provided to the community as open source under GPL license: https://github.com/tkn-tub/wifi-ftm-ns3.



```
⌄ Tagged parameters (11 bytes)
  ⌄ Tag: Fine Time Measurement Params
      Tag Number: Fine Time Measurement Params (206)
      Tag length: 9
    ⌄ FTM Params (Subset 1 of 3): 0x7100
        .... .... .... ..00 = Status Indication: 0x0
        .... .... .000 00.. = Value: 0x00
        .... .... 0... .... = Reserved1: 0x0
        .... 0001 .... .... = Number of Burst Exponent: 0x1
        0111 .... .... .... = Burst Duration: 0x7
    ⌄ FTM Params (Subset 2 of 3): 0x1500000a
        .... .... .... .... .... .... 0000 1010 = Min Delta FTM: 0x0a
        .... .... 0000 0000 0000 0000 .... .... = Partial TSF timer: 0x0000
        .... ...1 .... .... .... .... .... .... = Partial TSF no pref: 0x1
        .... ..0. .... .... .... .... .... .... = ASAP Capable: 0x0
        .... .1.. .... .... .... .... .... .... = ASAP: 0x1
        0001 0... .... .... .... .... .... .... = FTM per burst: 0x02
    ⌄ FTM Params (Subset 3 of 3): 0x0a0000
        .... .... .... .... .... ..00 = Reserved2: 0x0
        .... .... .... .... 0000 00.. = Format and Bandwidth: 0x00
        0000 1010 0000 0000 .... .... = Burst Period: 0x0a00

0000  00 00 16 00 0f 00 00 00  32 00 00 00 00 00 00 00   ········ 2·······
0010  10 02 6c 09 a0 00 d0 00  3a 01 00 00 00 00 00 03   ··l····· :·······
0020  00 00 00 00 00 01 00 00  00 00 00 01 00 00 04 20   ········ ·······
0030  01 ce 09 00 71 0a 00 00  15 00 00 0a 00 00 00 00   ····q··· ········
```

Fig. 6: FTM request packet displayed in Wireshark.

TABLE I: Mean ranging error

	20 MHz		**40 MHz**	
	number of FTM		number of FTM	
Error model	2	40	2	40
wired	0.267 m	0.047 m	0.131 m	0.020 m
wireless	0.679 m	0.572 m	0.596 m	0.577 m

V. Performance Evaluation

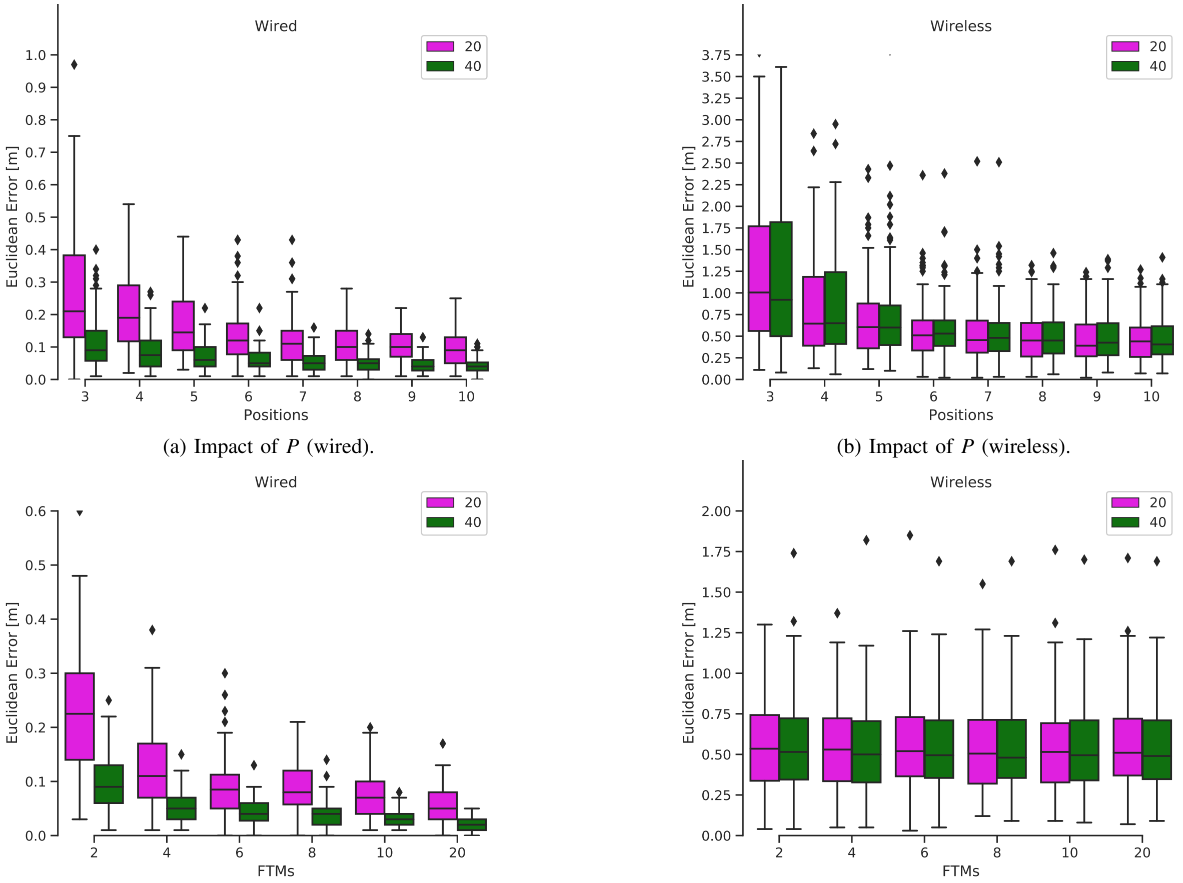
In this section, we present simulation results. We begin with performing some ranging experiment illustrating the impact of the used error models on the ranging accuracy. Afterwards, we analyze the performance of a simple localization scheme based on multi-lateration which uses the distances obtained from ranging to estimate the 2D position of a WiFi node.

A. FTM Ranging

First, we analyze the FTM ranging accuracy for different configurations like channel bandwidth and used error model. In this experiment, the FTM responder was fixed while the initiator was moving around on a circle with a 5 m radius. The initiator performed ranging operation at 180 different positions. The mean error was analyzed for different bandwidth configuration, error models and number of FTM measurements F. In case of $F = 2$ measurements there is only a single burst with 2 FTMs. With $F = 40$ there are 4 bursts with 10 FTMs per burst. This would result in 1 and 39 RTT calculations for these cases.[4]

The results are shown in Table I. We see that while the ranging error could be significantly reduced by sending 40 instead of 2 FTM requests/responses in the wired model this is not the case for the wireless model. This is because in the bias introduced in the wireless model cannot be averaged out by sending multiple FTM requests. We also see the improvement due to larger channel bandwidth. Note, that without any error model the mean error is just 3 mm.

[4] Note, averaging is carried out in case of $F > 2$.



(a) Impact of P (wired).

(b) Impact of P (wireless).

(c) Impact of F (wired).

(d) Impact of F (wireless).

Fig. 7: Localization accuracy for different number of locations P, number of rangings F, and error models.

B. Localization

In a second experiment, we study the impact of the FTM ranging error on a more complex localization problem. We implemented a localization scheme based on multi-lateration where a mobile WiFi station knowing its 2D location performs FTM ranging while moving to a fixed anchor node with unknown position, e.g., some WiFi AP. The goal is to determine the 2D location of that fixed node. Therefore, the mobile station performs FTM ranging at randomly selected locations P (positions). Both the wired and wireless error models as well as different channel bandwidth configurations are used to test the accuracy of the localization estimation. First, we analyze the impact of the number of locations P at which the mobile station performs ranging. Second, we want to see whether the number of FTM requests/responses, F, sent at each location helps improving the localization accuracy. Each of the experiments was repeated 100 times. The multilateration algorithm we used is based on minimizing the error function.[5]

1) Number of Positions: We start with analyzing the impact of the number of locations P at which the mobile station performs ranging. The FTM parameters for all the measurements are the same: $F = 6$, i.e., single burst with 6 FTMs.

The results under the wired error model are shown in Figure 7a. With increasing P the accuracy of the localization can be increased significantly. It is also visible that using 40 MHz bandwidth yields more accurate results than using the narrower 20 MHz channel. Hence, the usage of wider channel directly translates in less required positions to perform ranging to achieve the same accuracy. In our experiment using 40 MHz and $P = 3$ positions, achieves the same accuracy as 20 MHz and $P = 7$ positions.

Next, the same experiment is repeated but using the wireless error model (Figure 7b). From the results we see that the used channel bandwidth has a minor impact, i.e., the performance at 40 MHz is only slightly better. For small P the variance is even smaller with 20 MHz channel. This is due to the large bias introduced by the wireless error model.

2) Number of Measurements: In this experiment, we fix the number of randomly selected locations to $P = 10$ and vary the number of FTM rangings, F, performed at each location. Hence, the FTM session parameters are 1 burst and an increasing number of FTMs per burst F, chosen at 2, 4, 6, 8, 10, and 20. Again both error models are tested.

Figure 7c shows the results for the wired error model. It is evident that an increasing amount of FTM measurements F at each location leads to a higher accuracy. The biggest improvements happen when increasing F to 8. Afterwards

[5] https://github.com/glucee/Multilateration

an improvement is still visible but it is smaller. Comparing 20 and 40 MHz, shows that the higher bandwidth improves the accuracy significantly. This means that there is a trade-off between the amount of measurements F needed and channel bandwidth used. For example, using 40 MHz channel with $F = 2$ achieves the same accuracy as using 20 MHz with $F = 8$. This trade-off should always be considered, because it can reduce the load on the wireless channel to free up more resources for actual data transmissions while being able to still achieve the same accuracy. Thus the bandwidth should be chosen as high as possible, so that the amount of measurements can be reduced.

Finally, Figure 7d shows the performance under the wireless error model. Wee see that neither bandwidth nor amount of measurements F have a significant impact on the location accuracy. This is because the RTT bias due to multipath is location bound, so it is always the same for a given location and cannot be tackled by higher F. Hence, we can conclude that when using such a simple localization scheme in scenarios with strong multipath, e.g., indoors, there is no gain from using larger channel bandwidth and higher number of measurements at each location.

VI. Related Work

Our work is inspired by the experimental studies showing the performance of WiFi FTM in real-world environments. Bullmann et al. [5] evaluated FTM in realistic indoor scenarios using Intel 802.11ac WiFi hardware as well as Android smartphones in the 2.4 GHz ISM band together with 20 MHz channels. They discovered poor FTM performance in NLOS scenarios where they claim that environmental factors of the building like Shadowing from heavy metal fire doors affect the distance estimation process. At some measurement points they observed that the distance obtained from FTM ranging is bimodally distributed. With similar hardware Guo et al. [12] performed FTM ranging measurements indoors and in an outdoor open area with LOS. The authors conclude that the distribution of the RTT ranging error can be modeled as a Gaussian random process with zero mean and some variance. Hashem et al. [13] performed measurements in two typical indoor environments: a college campus building floor and a work office floor. As hardware they used Google WiFi APs together with Google Pixel XL Android smartphones. Finally, Jathe et al. [4] performed FTM ranging experiments indoors in a long hallway with clear LOS but strong multipath. Table II summarizes the key results on the reported FTM ranging accuracy from different experimental studies.

TABLE II: Reported FTM ranging accuracy.

Study	Mean error
Guo et al. [12, Figure 5]	1.27 m
Hashem et al. [13, Figure 5]	1.15 m
Bullmann et al. [5, Table 1]	1.76 m
Retscher [14, Figure 18]	1.41 m

VII. Conclusion

In this paper, we presented FTM-ns3, an extension to the ns3 network simulator to support the IEEE 802.11 FTM protocol. The system supports the use of empirical error models. As a proof of concept, we provide two error models which allow to analyze the impact of channel bandwidth and LOS with multipath propagation on the performance of FTM-based localization schemes. Our simulation results confirm the impact on ranging solutions. We also show that even simple localization schemes are highly affected by ranging inaccuracy introduced due to multipath. Our extension can also be used for benchmarking different localization schemes. As future work we will focus on updating the wireless error model to support environments with either non-LOS or obstructed-LOS, e.g., shopping center or outdoor. Moreover, FTM-ns3 can be extended to support wider channel bandwidth, e.g., 80 or 160 MHz, as defined in IEEE 802.11ac/ax.

References

[1] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "SpotFi: Decimeter Level Localization Using WiFi," *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 45, no. 4, pp. 269–282, Sep. 2015.

[2] A. Zubow, S. Bayhan, P. Gawłowicz, and F. Dressler, "DeepTxFinder: Multiple Transmitter Localization by Deep Learning in Crowdsourced Spectrum Sensing," in *IEEE International Conference on Computer Communication and Networks (ICCCN 2020)*. Virtual Conference: IEEE, Aug. 2020.

[3] IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE, Std 802.11-2016, Dec. 2016.

[4] N. Jathe, M. LÃ¼tjen, and M. Freitag, "Indoor Positioning in Car Parks by using Wi-Fi Round-Trip-Time to support Finished Vehicle Logistics on Port Terminals," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 857–862, 2019.

[5] M. Bullmann, T. Fetzer, F. Ebner, M. Ebner, F. Deinzer, and M. Grzegorzek, "Comparison of 2.4 GHz WiFi FTM- and RSSI-Based Indoor Positioning Methods in Realistic Scenarios," *Sensors*, vol. 20, no. 16, p. 4515, Aug. 2020.

[6] M. Ibrahim, H. Liu, M. Jawahar, V. Nguyen, M. Gruteser, R. Howard, B. Yu, and F. Bai, "Verification: Accuracy Evaluation of WiFi Fine Time Measurements on an Open Platform," in *24th ACM International Conference on Mobile Computing and Networking (MobiCom 2018)*. New Delhi, India: ACM, Oct. 2018.

[7] NS-3 Consortium, "ns-3 documentation," https://www.nsnam.org/, accessed: 2021-05-23.

[8] K. I. Ahmed and G. Heidari-Bateni, "WSN06-3: Improving Two-Way Ranging Precision with Phase-offset Measurements," in *IEEE Global Telecommunications Conference (GLOBECOM 2006)*. San Francisco, CA: IEEE, Nov. 2006.

[9] S. Lanzisera, D. Zats, and K. S. J. Pister, "Radio Frequency Time-of-Flight Distance Measurement for Low-Cost Wireless Sensor Localization," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 837–845, Mar. 2011.

[10] X. Li and K. Pahlavan, "Super-Resolution TOA Estimation With Diversity for Indoor Geolocation," *IEEE Transactions on Wireless Communications (TWC)*, vol. 3, no. 1, pp. 224–234, Jan. 2004.

[11] Q. H. Spencer, B. D. Jeffs, M. A. Jensen, and A. L. Swindlehurst, "Modeling the statistical time and angle of arrival characteristics of an indoor multipath channel," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 18, no. 3, pp. 347–360, Mar. 2000.

[12] G. Guo, R. Chen, F. Ye, X. Peng, Z. Liu, and Y. Pan, "Indoor Smartphone Localization: A Hybrid WiFi RTT-RSS Ranging Approach," *IEEE Access*, vol. 7, pp. 176 767–176 781, Jan. 2019.

[13] O. Hashem, M. Youssef, and K. A. Harras, "WiNar: RTT-based Sub-meter Indoor Localization using Commercial Devices," in *18th IEEE International Conference on Pervasive Computing and Communications (PerCom 2020)*. Austin, TX: IEEE, Mar. 2020.

[14] G. Retscher, "Fundamental Concepts and Evolution of Wi-Fi User Localization: An Overview Based on Different Case Studies," *Sensors*, vol. 20, no. 18, p. 5121, Sep. 2020.