



HAL
open science

FTM-Broadcast : Détermination efficace de coordonnées sur un réseau à grande échelle

Yann Busnel, Hervé Rivano

► **To cite this version:**

Yann Busnel, Hervé Rivano. FTM-Broadcast : Détermination efficace de coordonnées sur un réseau à grande échelle. CoRes 2024: 9èmes Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication, May 2024, Saint-Briac-sur-Mer, France. hal-04569538

HAL Id: hal-04569538

<https://hal.science/hal-04569538>

Submitted on 6 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

FTM-Broadcast : Détermination efficace de coordonnées sur un réseau à grande échelle

Yann Busnel¹ et Hervé Rivano²

¹*IMT Nord Europe, IRISA, 59500 Douai, France*

²*INSA Lyon, Inria, CITI, UR3720, 69621 Villeurbanne, France*

La géolocalisation intérieure a connu une avancée significative grâce au perfectionnement du protocole 802.11 FTM (Fine Timing Measurement). Une géolocalisation intérieure précise a de nombreuses applications dans des domaines tels que le suivi des équipements, la navigation intérieure et les services fondés sur la localisation. Le protocole standard 802.11 FTM permet un positionnement intérieur précis en mesurant le temps de propagation entre un appareil mobile et plusieurs points d'accès (AP). Il peut être généralisé à la localisation entre appareils. Toutefois, la mise en œuvre conventionnelle du protocole FTM souffre d'une complexité accrue lorsque le nombre d'appareils augmente, ce qui limite son extensibilité. FTM implique en effet un échange de messages point à point entre chaque paire d'appareils, ce qui entraîne une augmentation quadratique du nombre de messages lorsque le nombre d'appareils dans le voisinage augmente. Dans cet article, une méthode novatrice est proposée pour améliorer le protocole FTM en tirant parti de la communication par diffusion, ce qui se traduit par une réduction substantielle de la complexité en nombre de messages, qui se réduit de quadratique à linéaire. Notre approche élimine en effet la nécessité d'échanges ponctuels multiples et propose un mécanisme où un seul message provenant de l'appareil mobile est diffusé simultanément à tous les voisins. Chaque message échangé sera alors utilisable pour calculer chaque temps de propagation par paire, en incluant tous les horodatages dans les messages, rendant ainsi le protocole plus efficace et plus évolutif. Nous avons mené des simulations détaillées pour évaluer les performances du protocole FTM amélioré. Les résultats attestent de l'efficacité de la méthode proposée et mettent en évidence une réduction significative de la charge de calcul en comparaison avec l'implémentation classique du protocole FTM.

Mots-clefs : Géolocalisation, FTM, Wi-Fi, Broadcast, Piggybacking

1 Introduction

Obtaining accurate indoor location information presents a significant challenge, as GPS signals are often inaccessible indoors. For many years, alternative techniques have been actively pursued to address the challenge of indoor location tracking [ZGL19], in particular using wireless communication for ranging [CP20]. One such technique is Fine Timing Measurement (FTM) specified in the 802.11-2016 standard [IEE16], which has been further enhanced through the 802.11az amendment. FTM leverages Wi-Fi ranging to enable precise indoor location tracking. FTM relies on a Time of Flight (ToF)-based procedure. FTM enables ranging exchanges between an initiating station (ISTA) and a responding station (RSTA), facilitating indoor localization.

The current version of FTM assumes that ISTAs perform an unicast exchange with each RSTA (between 3 and 6) to estimate its position. If we consider use-cases with numerous mobile ISTAs, such as smartphones in large shopping malls or autonomous robots in mega warehouses, it seems unattractive to multiply unicast communications. In this paper, we propose to take advantage of the use of broadcast communication in wireless networks such as Wi-Fi, in order to save a significant amount of message exchanges, by reducing the complexity of the self-location protocol while guaranteeing a reduction in latency in most cases.

The remainder of this paper is organized as follows : Section 2 exposes the system model and the FTM background. Section 3 introduces then our network-scale 2-way ranging protocol. Section 4 provides theoretical and experimental validation of the proposed method in various scenarios.

2 FTM Principle and Related Works

2.1 System Model

We consider a spatial area containing a set of n devices (called nodes in the following) that can communicate through a wireless communication channel, using the IEEE 802.11-2016 [IEE16]. Each node is identified by their id d_i , $i \in \llbracket n \rrbracket$, and located in the three-dimensional space. Let $p_i = (x_i, y_i, z_i)$ be respectively the position (x_i, y_i) of node d_i in the 2D plane, and z_i its altitude. Each node d_i is equipped with an omnidirectional antenna. It is able to communicate with other nodes located within a ball of radius ρ_{p_i} . Let \mathcal{N} denotes the set of all reachable nodes in the ball centered on d_i .

To take a step forward in the direction of a realistic network model, a MAC layer is modeled, namely a classical CSMA/CA with RTS/CTS-like mechanism [BDSZ94] and clear channel assessment (CCA). Before any transmission, a node senses the channel. If the channel is occupied by the transmission to or from a neighboring node, its transmission is delayed by a backoff, which is randomly picked in range $[0; \beta]$, with β as a user parameter. The MAC layer is ideal, meaning it is lossless, and no simultaneous transmission happens in the transmission range of a node.

2.2 Fine Timing Measurement

FTM introduces a novel exchange of unassociated user-targeted content between an initiating station (ISTA) and a responding station (RSTA). Unlike previous provisions in the 802.11 standard, which mainly served general information or radio parameter exchanges, FTM leverages unassociated exchanges for user-centric purposes, such as indoor localization.

ISTA negotiates ranging session parameters with RSTA, followed by a series of bursts. In each burst, RSTA sends a frame at time t^1 , received by ISTA at time t^2 . ISTA responds with an acknowledgement frame at time t^3 , received by RSTA at time t^4 . RSTA then communicates the values (t^1, t^4) to ISTA, enabling ISTA to compute the Time of Flight (ToF) : $(t^4 - t^1) - (t^3 - t^2)$ and determine its distance d from RSTA using :

$$d = \frac{(t^4 - t^1) - (t^3 - t^2)}{2} \times c \quad (1)$$

where c refers to the speed of light. In the following, the original version of FTM is denoted FTM-UC.

3 A network-scale 2-way ranging protocol

The main idea of our proposal is to use broadcasting to reduce the number of messages exchanged during pairwise exchanges. Thus, instead of initiating a 2-way ranging with all its neighbors within communication range, each node of the network will broadcast its message without targeting any destination a priori. Thus, any node in the network that receives a previously broadcast message can consider it as the initial message of its own FTM with the concerned emitter.

In addition, each message sent can be used to carry all the information known locally up to that time. Any message will be piggybacked with the set of timestamps stored within the current round.

Figure 1 depicts the timeline of our protocol for a clique of 5 nodes. In the first phase of the round, each node responds to the initial broadcast of Node 0 by transmitting the timestamps of all broadcast messages received so far (t_*^2). Each timestamp t_*^3 will be considered as the timestamp t^1 for those pairs that had not yet communicated during phase 1. Thus, at the end of phase 1, any pair of nodes will have made a round trip communication. As with the classic FTM protocol, phase 2 of the algorithm is used to transmit to all the other nodes in range the send and receive timestamps not yet broadcast in phase 1.

At the end of phase 2, all the nodes have the 4 timestamps, enabling them to calculate their distance to any other node in range using Equation 1.

It is important to note that the order in which messages are sent between phases 1 and 2 does not matter. The transmitted data can be done in any order, with no consistency between the two phases. The only constraint is that all messages in phase 1 must be completed before phase 2 is initialized.

The algorithm of the protocol FTM-BC consists of two parts : the packet processing, where received and emitted packet information is recorded and analyzed, and the sending of packets based on specific conditions. Let us describe each part in detail.

3.1 Packet Processing

The algorithm takes as input three data structures : *reception*, *emission*, and *received_at*. These dictionaries store information about received packets, emitted packets, and reception times, respectively. The algorithm receives a packet p from source s at time t and extracts relevant information such as the packet ID (pid), current phase, and lists of sent and received packets timestamps ($sent_s$ and $received_s$).

First, the algorithm updates all the dictionaries :

- The reception of the packet p is recorded in the *reception* dictionary, with the sending time set to an empty value (\emptyset).
- The emission times of previous packets from s are retrieved in the $sent_s$ list and the *reception* dictionary for source s is updated.
- The reception times by source s are updated similarly.

When all needed timestamps from s have been received, the Time of Flight (ToF) can be computed.

The last part of the algorithm checks whether the node has already managed the necessary reaction to the current phase. On the one hand, if we are on the *root* node initiating the FTM-BC (here, set to 0 without losing generality), it prepares the next packet to be sent at the start of the next phase. On the other hand, if we are on another node, it checks whether a response message has already been sent. If not, it prepares the answer packet for the current phase and marks the phase as answered.

Finally, the algorithm delays the sending of the next packet by the current time plus the *phase_delay*.

3.2 Sending Packet

The algorithm takes as input a buffer, which is a list of packets to send. It is executed only when the underlying MAC layer signals that an emission is possible (*e.g.* when the backoff counter of the packet reaches 0 in a CSMA/CA mechanism). For each packet p in the buffer, the algorithm checks if it is time to send the packet ($p.time_to_send = now$). If so, the algorithm loads the unsent data from the *emission* and *received_at* dictionaries into the packet p and sends it.

The latter sending mechanism yields an issue for an implementation in a real networking stack. It is usually not possible to modify the payload of a packet once it is in the lower layers buffers. It may need that the protocol is implemented in the driver of the network interface, similarly to actual FTM implementations. Discussion is provided in the companion paper [BR23].

4 Validation and evaluation

In companion paper [BR23], we analyze the asymptotic behavior of our protocol. In brief, let \bar{N} be the average degree of the nodes in the underlying communication graph. Each round of the FTM-UC protocol requires $2n\bar{N}$ messages, 2 per pair of nodes, to estimate the complete set of pairwise distances of the participating nodes. Each round of the FTM-BC protocol only requires $2n$ messages to complete the same task. In the case of very high density graphs, the message complexity thus goes from quadratic to linear. The gain in transmitted information is however lower, only a factor 2 from $2n\bar{N}$ timestamps per round for FTM-UC to $n(\bar{N} + 1)$ for FTM-BC.

Both FTM-BC and FTM-UC have been implemented in a dedicated simulator available on gitlab[†]. All nodes are autonomous agents that can communicate only through an ideal communication layer with a

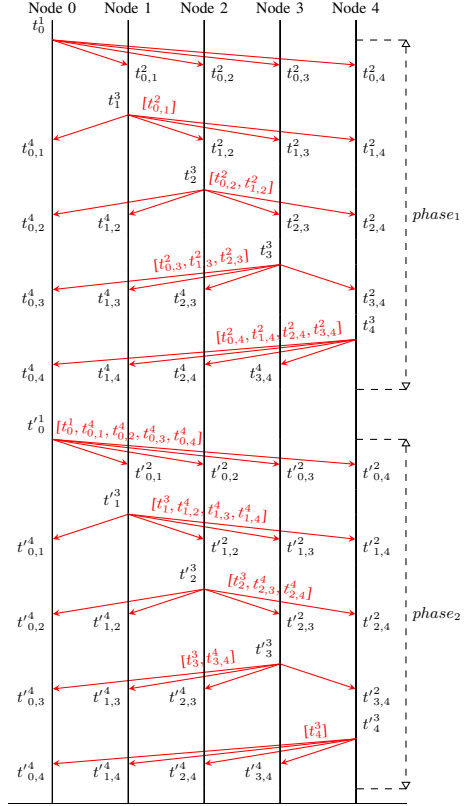
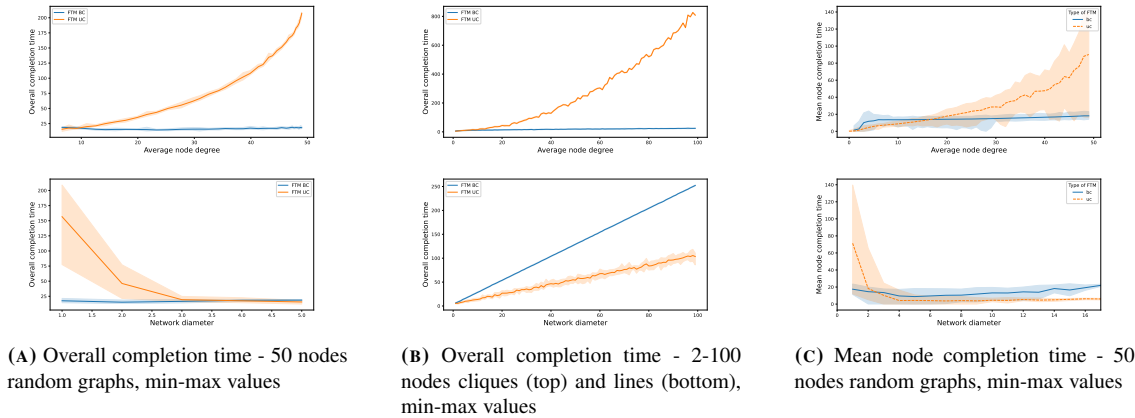


FIGURE 1 : FTM-BC protocol

[†]. Gitlab of our code : https://gitlab.inria.fr/hrivano/ftm_broadcast



(A) Overall completion time - 50 nodes random graphs, min-max values

(B) Overall completion time - 2-100 nodes cliques (top) and lines (bottom), min-max values

(C) Mean node completion time - 50 nodes random graphs, min-max values

FIGURE 2 : Performance evaluation

CSMA/CA MAC layer, also implemented as an autonomous agent. In the following, we report simulation results on 2 to 100 nodes lines, 2 to 200 cliques and 50 nodes random graphs for which the communication range vary from 1 (sparse graph) to high values (when the graph becomes a clique).

Figure 2a and Figure 2b reports the overall completion time. For both protocols, a high degree implies more contention for accessing the medium, hence larger delays due to the random backoff in the CSMA/CA layer. This slows down both FTM-BC and FTM-UC. For FTM-UC however, the degree has an additional and more significant impact since one transaction has to be performed with each neighbor. Sparse networks with large diameters and low degrees are the most efficient configurations for FTM-UC. It indeed is able to perform several ranging in parallel. Oppositely, the two waves produced by FTM-BC induces a delay that is proportional to the diameter time the *phase_delay* parameter. Overall, for very sparse networks, FTM-UC can complete the ranging faster than FTM-BC by taking advantage of spatial reuse in radio networks. Note that FTM-UC needs that a BFS is done beforehand so that each node knows its neighborhood, in particular their ID. But, since there are other mechanisms, potentially more effective, that can provide this information (or that this information can be needed by other protocols), the time taken by BFS is not counted here for sake of fairness.

Figure 2c reports the average node completion time, only on the 50 nodes random graphs, since the results on lines and cliques are directly related to the overall completion time. More than 120 topologies have been simulated because the randomness of CSMA/CA have a significant impact on the result, in particular for FTM-UC, as one can see in the min-max discrepancy. FTM-UC is faster in most of the sparse network cases, which is expected since each FTM-UC transaction is run in a row and each node loops over its neighbors (with higher IDs) while each node waits that the first FTM-BC wave is completed in its neighborhood before answering to the second one. Besides, the waiting time implemented might be over dimensioned and need to be carefully optimized. However, the difference between the mean node completion time with FTM-BC and FTM-UC is much lower in sparse networks than in dense ones.

Références

- [BDSZ94] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. Macaw : A media access protocol for wireless lan's. In *ACM SIGCOMM*, 1994.
- [BR23] Yann Busnel and Hervé Rivano. FTM-Broadcast : Efficient Network-wide Ranging. In *IPIN 2023 (Indoor Positioning and Indoor Navigation)*, Nuremberg, Germany, 2023.
- [CP20] Pablo Corbalán and Gian Pietro Picco. Ultra-wideband concurrent ranging. *ACM Transactions on Sensor Networks (TOSN)*, 2020.
- [IEE16] Wireless LAN medium access control (MAC) and physical layer (PHY) specification, 2016.
- [ZGL19] Faheem Zafari, Athanasios Gkelias, and Kin K. Leung. A survey of indoor localization systems and technologies. *IEEE Comm. Surveys and Tutorials*, 2019.