

Four-way TOA and Software-Based Trilateration of IEEE 802.11 Devices

Christian Hoene and Jörg Willmann

Wilhelm-Schickard-Institute

University of Tübingen

72076 Tübingen, Germany

Email:hoene|joerg.willmann@uni-tuebingen.de

Abstract—In this paper we present an algorithm that measures the round trip times of IEEE 802.11 MAC packets in order to trilaterate WLAN nodes. We show that if we use packet sequences (e.g., RTS, CTS, DATA and ACK) instead of packet pairs (e.g., DATA and ACK), the TOA measurements can be implemented more easily and a higher observation rate can be achieved. In addition, the receiver and any listening third node can measure the round trip times. We present the open source software “Goodtry” that implements this algorithm and overcomes the low clock resolution and other implementation constraints of off-the-shelf IEEE 802.11 cards. Experimental results show that our software-based trilateration has an accuracy of about four meters.

I. INTRODUCTION

State of the art location algorithms for determining the position of WLAN nodes use Received Signal Strength Indications (RSSI). In 1999, Bahl and Padmanabhan presented one of the first scientific location tracking algorithms based on WLAN RSSI measurements [1]. Many other researchers have extended those results. In addition, numerous commercial solutions have been developed, making WLAN RSSI location tracking a mature technology.

Location tracking based on time of arrival (TOA) algorithms measures the duration of the propagation of the physical transmission signal, which travels at the speed of light through vacuum. As compared to RSSI measurements, TOA has the benefit that its measurement results scale linearly with the open-air propagation distances. In addition, it has the potential to determine the geographic position more precisely and also without setting up a map of RSSI values that most RSSI-based algorithms require [1]. As such, TOA algorithms are a technology that can replace or enhance RSSI-based solutions.

TOA-based algorithms require special hardware support to measure times of flight precisely. Technically systems are GPS, ultrawideband or WLAN chip sets extended by a TOA unit. In previous work [2], we proved that even off-the-shelf IEEE 802.11 network interface cards can conduct time of flight measurements at accuracy of few meters. We utilized the immediate acknowledgements of the IEEE 802.11 protocol to measure the round trip time of packets. In this paper, we extend our early work. We introduce a four-way TOA algorithm that enhances TOA measurements. Four-way TOA

has the advantage that both sender and receiver can measure the round trip times. Even cooperative distance measurements via third nodes become possible. Our four-way TOA algorithm is implemented in the open source software “Goodtry”.

In this paper, we use existing WLAN hardware enhanced by our software. The motivation behind our pure software-based solution is the following: First, a software-based algorithm can be easily implemented as an update to existing WLAN device drivers. As such, it is a cost saving alternative to more expensive redesign and enhancement of WLAN chip sets. Secondly, as we will see in the following sections, software-based solutions are generic enough to work regardless of the precision of the underlying hardware. As soon as the hardware increases its TOA support, the algorithms can be applied in same way as with existing hardware in order to enhance the accuracy.

The remainder of this paper includes in the second section references to related work. In the third section, we describe the four-way TOA algorithms that increase the effectiveness of round trip time measurements. Then, we describe the implementation of Goodtry. Section 5 contains first experimental results. Finally, we conclude and give an outlook to the next steps in software-based locating research.

II. RELATED WORK

Time of arrival measurements in indoor environment have been conducted in [3], [4]. An overview on wireless indoor geolocation techniques is given in [5], [6]. The Cramér-Rao bound can be applied for synchronization of modulated signals to determine the lower bound of accuracy of TOA measurements [7]. More precisely, the variance of a radio frequency time of flight (TOF) measurement is bounded by:

$$\sigma_{TOF}^2 = \frac{1}{8\pi^2 \cdot SNR \cdot \sqrt{\alpha} \cdot BW^2 \cdot N}$$

where SNR is the average signal to noise ratio, α is the number of code copies averaged, BW is the spectral bandwidth, and N is the number of chips in the pseudo noise code [8]. If we apply this equation to IEEE 802.11 modulation, the lower limit of TOF measurements is below one meter.

McCrary et al. [9] were the first to propose two-way TOA measurements using DSSS communication systems like IEEE

This work has been financed by the Landesstiftung Baden Württemberg GmbH, Germany, in the BW-FIT project AmbiSense.

802.11. The authors used a dedicated hardware with low-accuracy clocks (+1 ppm). The hardware supported an internal calibration via a feedback loop. A dedicated DSP was used for estimating the leading edge of the signal in cases of multipath propagation. In addition, using different transmission frequencies helps to orthogonalize the multi and direct path. Based on this work, McCrady et al. developed a prototype achieving 0.5m horizontal Circular Error Probable (CEP) in open space and 3-5m CEP in harsh environments [10]. CEP refers to “hitting” the position within the given radius at a probability of 50% [11]. Günther and Hoene [2] used two-way TOA measurements to estimate the distance between IEEE 802.11 nodes. The round trip time of the data and acknowledgement packets were measured using two commercial WLAN chipsets. To cope with a rounding error of $1\mu\text{s}$ caused by hardware limitations, the authors calculated the mean over multiple measurements. Because of the clock drift between both nodes, the rounding error is alternating over time. This effect helps to overcome the quantization error of the hardware clock if enough measurement values are averaged (about 500). Instead of using a pure software approach, Ciurana et al. [12], [13] built a hardware extension, which measures the transmission times being clocked by the 44 MHz built-in quartz of the WLAN card. The author extended their work with a Kalman filter in [14]. Ahmed and Heidari-Batani [15] mathematically analyzed the positive effect of phase-offsets in cases of finite clock resolution for two-way TOA measurements. Golden and Bateman [16] also applied two-way TOA rangings for locating with good results. Their testbed is based on an Intel WLAN Card 2915AGB extended by debug facilities and special firmware. To combat the multipath effect, they used antenna and frequency diversity. They also compared TOA measurements with signal strength measurements pointing out that TOA is superior as it has a lower root mean square error (RMSE). Further work has been done by Winkler et al. [17], adding a correlate module into a IEEE 802.11a receiver. Last but not the least, the IEEE P802.11 TGv objectives include the requirement that “TGv will provide a mechanism to coordinate the gathering and possibly generation of data to support various location methods such as time of arrival, time difference of arrival, and signal strength” [18]. One can assume that the upcoming 802.11v standard will support location determination algorithms based on both RSSI and TOA.

III. ALGORITHM

As stated above, related work has proposed two-way TOA measurements. For example, if considering IEEE 802.11 standards, the packet pairs DATA and ACK can be applied for ranging purposes. Then, the time of flight is calculated as

$$t_{tof} = \frac{1}{2}(t_1 - t_0 - t_{DATA} - t_{SIFS})$$

if the transmission duration of the data packet is known and the time stamps at the start of transmission of the data packet and at the start of receiving the acknowledgement packet (Fig 1).

Let us extend the measurement procedure by using multiple packets sent in a so-called *packet sequence* instead of just

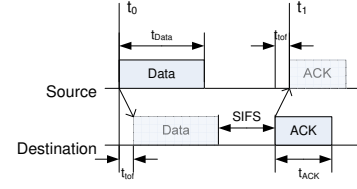


Fig. 1. Two-way TOA round trip time measurement

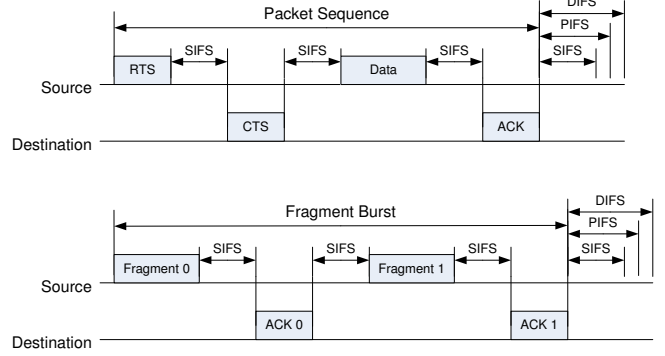


Fig. 2. Examples of IEEE 802.11 packet sequences

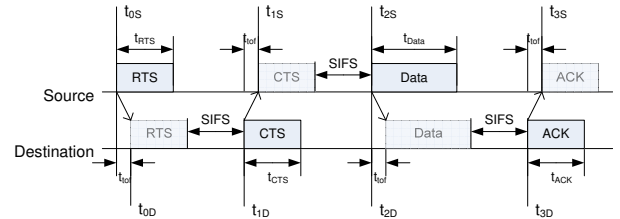


Fig. 3. Four-way packet sequences

a packet pair. The IEEE 802.11 standards support the transmission of packet sequences if—for example—the RTS/CTS mode is switched on, if fragmented packets are transmitted, or if Link Adaption Control (LAC) packets (in the upcoming IEEE 802.11n) are used (Fig. 2). In the following, we select the RTS, CTS, Data, and ACK packet sequence (Fig. 3) but the equations can easily be modified to other packet sequences.

With four-way TOA, more measurements are possible during the transmission of one network-layer data packet. At the source, (1), (2) and (3) calculate the time of flight; at the destination (4) can also be used for the same purpose.

$$t_{tof} = \frac{1}{2}(t_{1S} - t_{0S} - t_{RTS} - t_{SIFS}) \quad (1)$$

$$t_{tof} = \frac{1}{2}(t_{3S} - t_{2S} - t_{DATA} - t_{SIFS}) \quad (2)$$

$$t_{tof} = \frac{1}{4}(t_{3S} - t_{0S} - t_{DATA} - t_{CTS} - t_{RTS} - 3t_{SIFS}) \quad (3)$$

$$t_{tof} = \frac{1}{4}(t_{2D} - t_{1D} - t_{CTS} - t_{SIFS}) \quad (4)$$

Not all WLAN cards provide the time of transmission of packets (refer to Table I). Thus, some implementations cannot use the time stamps pointing to the transmission of a packet but only those of received packets. Still, the four-way packet sequence is useful if (5) or (6) are applied at the sender or

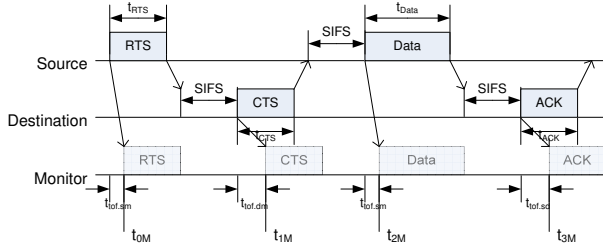


Fig. 4. Indirect four-way TOA

respectively at the destination.

$$t_{tof} = \frac{1}{4}(t_{3S} - t_{1S} - t_{DATA} - t_{CTS} - 2t_{SIFS}) \quad (5)$$

$$t_{tof} = \frac{1}{4}(t_{2D} - t_{0D} - t_{CTS} - t_{RTS} - 2t_{SIFS}) \quad (6)$$

Let us assume a third node that is just monitoring the transmission between source and destination as displayed in Fig.4. It can estimate the distance between source and destination using (7) or (8).

$$\begin{aligned} t_{0M} - t_{tof} &= t_{2M} - t_{tof} - 2t_{SIFS} - 2t_{tof} - t_{CTS} - t_{RTS} \\ \Leftrightarrow t_{tof} &= \frac{1}{2}(t_{2M} - t_{0M} - 2t_{SIFS} - t_{CTS} - t_{RTS}) \end{aligned} \quad (7)$$

$$\begin{aligned} t_{1M} - t_{tof} &= t_{3M} - t_{tof} - 2t_{SIFS} - 2t_{tof} - t_{Data} - t_{CTS} \\ \Leftrightarrow t_{tof} &= \frac{1}{2}(t_{3M} - t_{1M} - 2t_{SIFS} - t_{Data} - t_{CTS}) \end{aligned} \quad (8)$$

Knowing t_{tof} , the monitoring node can estimate its relative position to source and destination with—for example—the following equation:

$$\begin{aligned} t_{0M} - t_{tof} &= t_{1M} - t_{tof} - t_{SIFS} - t_{tof} - t_{RTS} \\ \Leftrightarrow t_{tof} - t_{tof} &= t_{0M} - t_{1M} + t_{SIFS} + t_{tof} + t_{RTS} \\ \stackrel{(7,8)}{\Leftrightarrow} t_{tof} - t_{tof} &= \frac{1}{2}(t_{2M} + t_{0M} - t_{CTS} + t_{RTS}) - t_{1M} \end{aligned}$$

The distance between two nodes cannot be calculated with one TOF measurement because it is not precise enough. Instead, it is required to average the measurement results of multiple observations. In addition, to cope with an absolute measurement error, a predefined $tara_i$ value must be subtracted from the calculated mean. The tara values must be determined prior to operation in a calibration process. We conducted the calibration by placing the two nodes side-by-side, conducting a distance measurement, and selecting the tara value so that the distance estimate becomes zero. If WLAN card supports an internal calibration, such as described in [9], then the above described calibration process can be omitted and the tara values could be determined and distributed automatically.

After calculating the mean transmission time $\overline{t_{tof_i}}$ with equation i , we can calculate the distance d_i assuming that the packets have travelled in open air at the speed of light c . The following equation calculates the distance:

$$d_i = (\overline{t_{tof_i}} - tara_i) \frac{c}{w_i} \quad (9)$$

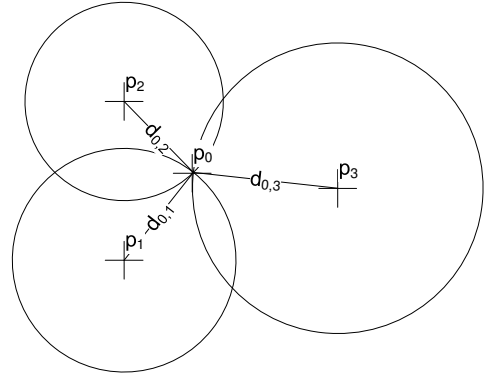


Fig. 5. A trilateration of p_0 in 2D requires three anchors at p_1, p_2, p_3 and three distances between the anchors and p_0 .

The constant w_i refers to the number of transmissions (two or four). If the multiple distances are calculated with (1)-(8), then these results can be combined. For example, the results can be averaged using the following equation:

$$d_{all} = \frac{\sum_i d_i n_i}{n_{all}}$$

with $n_{all} = \sum_i n_i$, d_i referring to the distance using equation i , and n_i to the number of observations that have been averaged to calculate d_i with(9).

A. Trilateration

Trilateration is a method of determining the relative positions of objects. Unlike triangulation, which uses angle measurements (together with at least one known distance) to calculate the subject's location, trilateration uses the known locations of two or more reference points (anchors), and the measured distance between the subject and each reference point [19]. To locate a node in two dimensions, three anchors are required (Fig.5); three dimensions require four anchors. With more than four anchors, the location cannot be determined consistently. Instead, we determine the position by using the method of the lowest weighted squared errors:

$$error^2 = \frac{\sum_{i=1}^n \left(\|p_0 - p_i\| - d_{all}^{0,i} \right)^2 n_{all}^{0,i}}{\sum_{i=1}^n n_{all}^{0,i}} \quad (10)$$

with p_0 the (unknown) position of the node; p_i are the known positions of n anchors; $d_{all}^{0,i}$ is the measured distance between the node and the anchor i ; and $n_{all}^{0,i}$ is the number of observations on which $d_{all}^{0,i}$ is based. To estimate p_0 , we conduct an iterative, multi-dimensional minimization of $error$ using the Nelder-Mead simplex method [20].

IV. IMPLEMENTATION

In the following, we describe the implementation of the software Goodtry that includes the above-mentioned TOA and trilateration algorithms. Goodtry is available under BSD license at [21]. Fig.6 gives an overview of Goodtry TOA

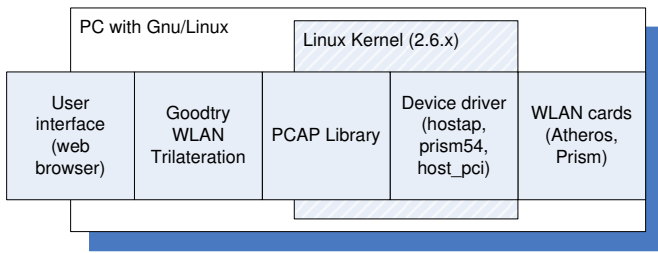


Fig. 6. Components of the Goodtry TOA system

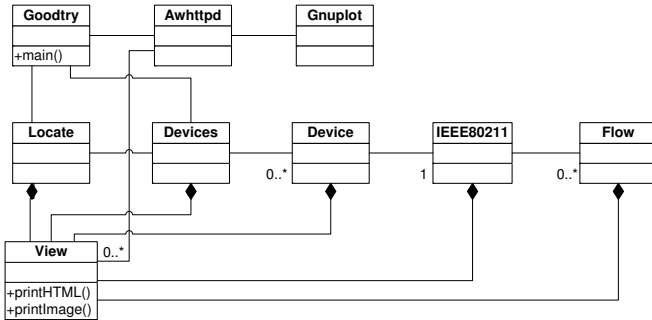


Fig. 7. UML class diagram

system and Fig. 7 shows the C++ classes of the Goodtry source code. To foster a better understanding of our open source software, we will refer to its classes in this publication (e.g., \Rightarrow Class).

Goodtry requires a PC with a Linux operating system. Windows cannot be supported because it lacks the support for low-level access to the WLAN device drivers. FreeBSD is not yet supported due to time constraints but porting Goodtry to FreeBSD should be possible. The PC must be equipped with IEEE 802.11 compliant WLAN networking cards. Not all cards are supported and not all supported cards are working equally well. Table I gives an overview of WLAN chip sets listing the most common products.

The most important requirement is that the WLAN chip set must provide time stamps taken at either the time of sending or receiving packets. This time stamp, frequently called MAC time or TSF, has to be set by the chip set hardware. It is not sufficient if software sets the time stamp, for example, in the device driver [22]. As a second requirement, the chip set must provide access IEEE 802.11 control packets such as RTS, CTS, and ACK. If these requirements are not met, a software-based TOA measurement is not possible. Currently, chip sets by Conexant and Atheros provide these features. Others can only be supported if their device drivers will be extended.

The PCAP library [23] provides low-level access to packets that are to be transmitted or have been received. These packets include network headers, transport headers and user data. However, for TOA measurements, we need access to the IEEE 802.11 MAC header and— even more importantly— to the chip set header fields, which include modulation type, length, and MAC time stamps. Three different chip set header types are implemented by different device drivers. These are

PrismHeader, AVS, and the RadioTap headers. Usually, the device drivers have to be configured properly to provide these headers. Goodtry supports all three header types and tries to configure the WLAN cards accordingly (\Rightarrow Device).

If packets arrive via PCAP, Goodtry identifies and stores physical parameters such as MAC time, modulation type, packet length, and physical preamble type. However, PCAP does not deliver packets always in order. Thus, Goodtry stores all received PCAP packets for one second (\Rightarrow IEEE80211) until they are processed, with an increasing MAC time stamp. Next, valid packet sequences are identified (\Rightarrow IEEE80211), which consists of RTS, CTS, Data, ACK packets.

Packet sequences must be differentiated according to source and destination address. Each source/destination address pair is considered a unidirectional flow, for which Goodtry has to calculate the time of flight separately.

To calculate the time of flight with (1)-(8), we need to know the transmission duration of the packets. Having the frequency, modulation rates and packet lengths, Goodtry calculates the packet transmission duration using the equation given in the IEEE 802.11 standards. However, these calculations are not trivial, because the chip sets do not provide information on the physical preamble used. For example, the IEEE 802.11g standard supports three different preambles: long (1 Mbps), short (2 Mbps) and g-only (OFDM) preamble, which all have different transmission durations. Thus, Goodtry generates a list of all possible preamble types (if one considers the modulation rate and frequency, the amount of choices is reduced). Then, after one entire packet sequence has been received, it calculates all possible transmission durations and compares these with the actual, measured values. The closest guess is considered valid. Finally, each valid TOA measurement result is stored for future analysis (\Rightarrow Flow).

Once the distance between two nodes has been calculated (e.g., every three seconds), the TOA measurements are analyzed. Goodtry calculates average, variance, and confidence interval of the measurement results. It limits itself either to the results of 5 minutes or at most some user selected maximum number of observations (\Rightarrow Flow).

Goodtry conducts a 3D trilateration using a node and multiple anchors in (\Rightarrow Location). The iterative, multi-dimensional minimization to estimate p_0 uses the GNU Scientific Library [20]. Besides the position, 10 gives an estimate on the error of the determined location.

Goodtry implements a Web based user interface (Fig. 8). It includes the open source Anti-Web HTTPD server [24], which has been extended to display the status of most of the objects (\Rightarrow View). To display graphical plots and diagrams, we use Gnuplot, which calculates the images after each HTTP request.

V. EXPERIMENTAL RESULTS

Using the Goodtry software, we have conducted first experiments at static positions. For example, Fig. 9 displays the distribution of TOA measurements results showing just the time of flight (after subtracting the tara value). All values are

TABLE I
OVERVIEW ON WLAN CHIPSETS AND THEIR TOA SUPPORT

Manufacturer	Chipset	Device number	Driver	Monitoring	Header type	Time stamp (TSF)	field	Simultaneous TX and Monitoring
Atheros	AR5006X	168c:001b	madwifi 0.9.4	yes	Radiotap, Prism	yes, but very imprecise		yes, but no TSF monitoring of tx packets
Atheros	AR5212	168c:0013	madwifi 0.9.4	yes	Radiotap, Prism	yes		dito.
Atheros	AR5416	na	madwifi 0.9.4	yes	Radiotap, Prism	yes, but imprecise		dito.
Atheros	ZD1211	0ace:1215	2.6.24+	yes	no	no		no
Broadcom	BCM4318	14e4:4318	2.6.24+(bcm43xx)	yes	IEEE80211	no		no
Broadcom	BCM4318	14e4:4318	2.6.24+(b43)	yes	Radiotap	no		not implemented
Conexant	PRISM2	000b:7300	2.6.24+	yes	IEEE80211, AVS, Prism	yes		not implemented
Intel	2915ABG	8086:4224	2.6.24+	yes	Radiotap	no (just beacons)		yes, but RTS/CTS mode does not work in IBSS mode.
Intel	3945ABG	na	2.6.24+	yes	Radiotap	no		no
Intel	LAN2100	8086:1043	2.6.24+	yes	Radiotap	no		no
Intersil	ISL3890	1260:3890	2.6.24+	yes	AVS	yes		no
Marvell	88w8335	1260:3890	Windows only	no	no	no		no

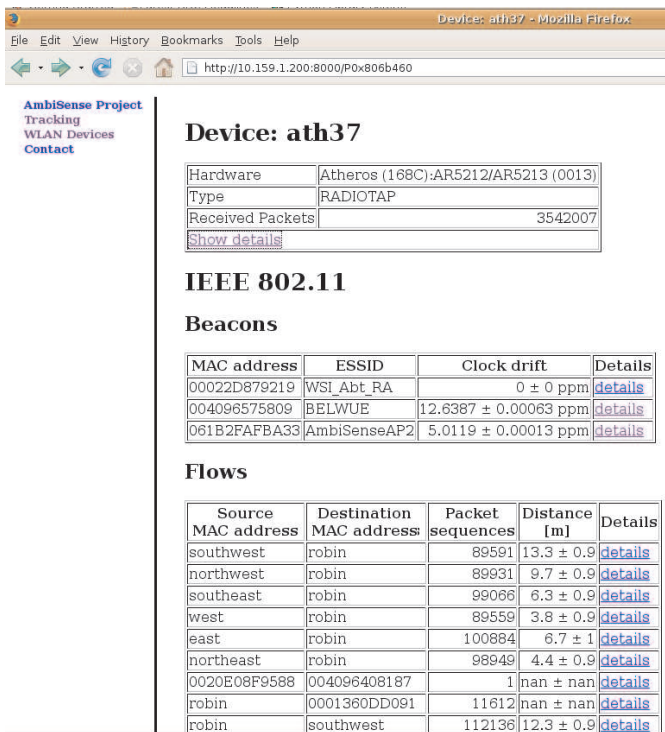


Fig. 8. Goodtry user interface

either -1 , 0 , or $1\mu s$. These results are typical for precise WLAN cards such as Prism WLAN chip sets or old Atheros cards. More recent Atheros chip-sets are relatively imprecise. We assume that Atheros has decided to switch from pure hardware solution to an embedded-CPU based MAC processing, which introduces variable latency (e.g., interrupt latency).

In Fig. 10, we display the results of different packet sequences if one node was moved away from the AP 15 meters and shortly later placed back to its original position. Goodtry can measure distance fast enough to support the dynamic location tracking of slow moving humans.

Fig. 11 shows a 2D trilateration of a moving robot in our laboratory having a size of $15.4 * 5.1 * 3.345_m^3$ [25]. We

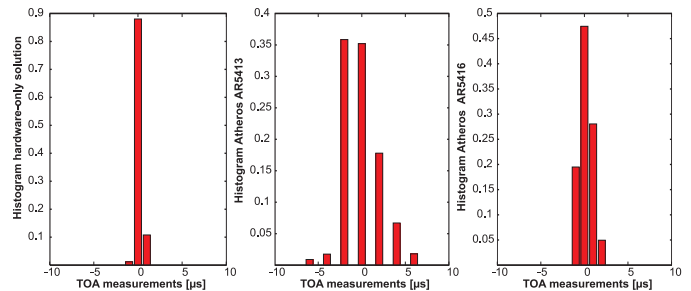


Fig. 9. Histogram of TOA measurements (left: hardware solution, middle: AR5413, right: AR5416)

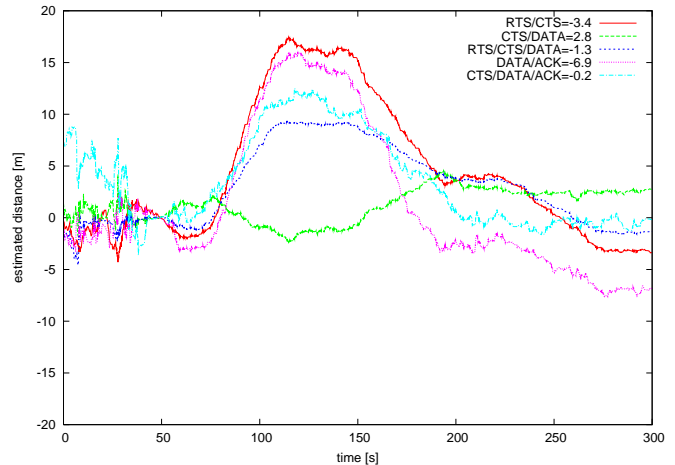


Fig. 10. Walking 15 meters and coming back.

placed five anchors (LinkSys WRT54GL operated in client mode) in the room. Each measured the distance to the moving object (operated in AP mode). Pings were sent from the moving objects to the anchors. The black cross shows the actual position estimate of the moving object. The black ellipse displays the error of the approximation. The grey line displays the history of the position estimates (the object has indeed moved).

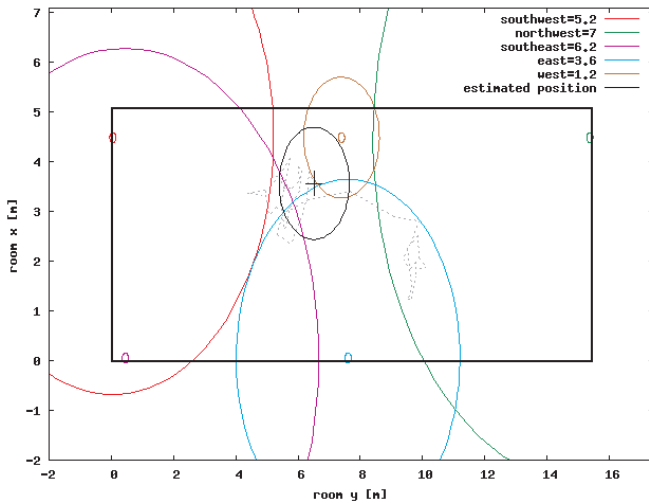


Fig. 11. Example of dynamic trilateration with five anchors.

VI. SUMMARY AND OUTLOOK

This publication shows that a software-based, real-time trilateration even with many off-the-shelf WLAN cards is possible. In [25] we have shown that a precision of about 4 m can be achieved.

Instead of using two-way TOA measurements, we propose to use packet sequences to support four-way TOA, which has many benefits but comes at the costs of lower transmission efficiency. However, the upcoming IEEE 802.11n standard requires four packets in sequence to support link adaptation control (LAC) and thus four-way TOA will not decrease throughput in IEEE 802.11n.

We made our WLAN trilateration software “Goodtry” open source to foster the development of TOA algorithms. For example, our future work will concentrate on increasing and assessing the accuracy of software-based TOA. Currently, we use artificially generated ping packets, all of them having the same size and modulation. If using different packet sizes and modulation rates, as it is typically for WLAN and IP traffic, the measurement errors increases significantly because then the clock drifts falsify the results. Thus, we will have to determine the clock drift and to compensate its effect to support TOA measurements during ongoing data transmissions.

ACKNOWLEDGMENT

The authors would like to thank Georgios Tokas and Dr. Glikeria Gestwa for filing patent, Prof. Georg Carle for his support and for Nikhil Komawar and Dr. E.-L. Hoene for proofreading this publication.

REFERENCES

[1] P. Bahl and V. N. Padmanabhan, “RADAR: an in-building RF-based user location and tracking system,” in *Infocom 2000*, vol. 2, Tel-Aviv, Israel, 2000, pp. 775–784.
 [2] A. Günther and C. Hoene, “Measuring round trip times to determine the distance between WLAN nodes,” in *Networking 2005*, Waterloo, Canada, 2005.

[3] A. Saleh and R. Valenzuela, “A statistical model for indoor multipath propagation,” *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 2, p. 128, 1987, 0733-8716.
 [4] K. Pahlavan, P. Krishnamurthy, and A. Beneat, “Wideband radio propagation modeling for indoor geolocation applications,” *IEEE Communications Magazine*, vol. 36, no. 4, p. 60, 1998, 0163-6804.
 [5] P. Kaveh, L. Xinrong, Y. Mika, C. Ranvir, and L. Matti, “An overview of wireless indoor geolocation techniques and systems,” in *Proceedings of the IFIP-TC6/European Commission International Workshop on Mobile and Wireless Communication Networks*. Springer-Verlag, 2000, 759983 1-13.
 [6] A. Hatami, B. Alavi, K. Pahlavan, and M. Kanaan, “A comparative performance evaluation of indoor geolocation technologies,” *Interdisciplinary Information Sciences*, vol. 12, no. 2, pp. 133–146, 2006.
 [7] T. Albery, “Frequency domain interpretation of the Cramer-Rao bound for carrier and clock synchronization,” *IEEE Transactions on Communications*, vol. 43, no. 234, p. 1185, 1995, 0090-6778.
 [8] S. Lanzisera, D. T. Lin, and K. S. J. Pister, “RF time of flight ranging for wireless sensor network localization,” in *International Workshop on Intelligent Solutions in Embedded Systems*, 2006, p. 1.
 [9] D. McCrady, L. Doyle, H. Forstrom, T. Dempsey, and M. Martorana, “Mobile ranging using low-accuracy clocks,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 48, no. 6, p. 951, 2000, 0018-9480.
 [10] D. McCrady, R. Goldberg, and T. Pfister, “RF ranging for indoor tracking and positioning,” in *WPI Technology Workshop: Precision Indoor Personnel Location and Tracking for Emergency Responders*, Aug. 2006.
 [11] Wikipedia. (2007) Circular error probable — wikipedia, the free encyclopedia. Accessed 12-December-2007. [Online]. Available: <http://www.wikipedia.org>
 [12] F. Izquierdo, M. Ciurana, F. Barcelo, J. Paradells, and E. Zola, “Performance evaluation of a TOA-based trilateration method to locate terminals in wlan,” in *1st International Symposium on Wireless Pervasive Computing*, 2006, p. 1.
 [13] M. Ciurana, F. Barcelo-Arroyo, and F. Izquierdo, “A ranging system with IEEE 802.11 data frames,” in *IEEE Radio and Wireless Symposium*, 2007, p. 133.
 [14] M. Ciurana, F. Barceló, and S. Cugno, “Indoor tracking in WLAN location with TOA measurements,” New York, NY, USA, pp. 121–125, 2006.
 [15] K. I. Ahmed and G. Heidari-Bateni, “Improving two-way ranging precision with phase-offset measurements,” in *IEEE Global Telecommunications Conference (GLOBECOM '06)*, 2006, p. 1.
 [16] S. A. Golden and S. S. Bateman, “Sensor measurements for Wi-Fi location with emphasis on time-of-arrival ranging,” *Transactions on Mobile Computing*, vol. 6, no. 10, p. 1185, 2007, 1536-1233.
 [17] F. Winkler, E. Fischer, E. Grass, and P. Langendörfer, “An indoor localization system based on DTDOA for different wireless lan systems,” in *3rd Workshop on Positioning, Navigation and Communication*, Hannover, Germany, Mar. 2006.
 [18] S. J. Kerry and D. Stanley, “Status of project IEEE 802.11v, wireless network management,” Dec. 2007.
 [19] Wikipedia. (2007) Trilateration — wikipedia, the free encyclopedia. Accessed 20-December-2007. [Online]. Available: <http://www.wikipedia.org>
 [20] (2003) GNU Scientific Library Reference Manual - second edition. [Online]. Available: <http://www.gnu.org/software/gsl/>
 [21] C. Hoene. (2008, Jun.) Goodtry - software based wlan trilateration. [Online]. Available: <http://sourceforge.net/projects/goodtry/>
 [22] —, “Internet telephony over wireless links,” PhD, Technical University of Berlin, TKN, 2005, phdthesis.
 [23] (2008) Tcpdump/libpcap public repository. [Online]. Available: <http://www.tcpdump.org/>
 [24] D. Hoyte. (2007) Anti-web HTTPD homepage. [Online]. Available: <http://www.hcsw.org/awhttpd/>
 [25] P. Vorst, J. Sommer, C. Hoene, P. Schneider, C. Weiss, T. Schairer, W. Rosenstiel, A. Zell, and G. Carle, “Indoor positioning via three different RF technologies,” in *RFID Systech*, Freiburg, Germany, Jun. 2008.