Location Verification of Wireless Internet Clients: Evaluation and Improvements

AbdelRahman Abdou, Ashraf Matrawy, and Paul C. van Oorschot,

Abstract—Client Presence Verification (CPV) was proposed in previous literature as a delay-based location verification algorithm that iteratively estimates Internet delays to corroborate assertions about a client's geographic presence in a prescribed region, e.g., before granting access to a location-based service. We evaluate CPV's performance in the presence of clients that use 802.11 networks by analyzing how the following factors affect CPV: the number of wireless clients, how far adversaries' are from their true locations, and the required number of CPV iterations to neutralize the effect of wireless networks. We use a mix of real-world traffic measurements from PlanetLab and existing wireless-delay probability models to create the evaluation datasets. The results indicate that, while wireless delays affect CPV's performance (e.g., from 3% to $\sim$4.7% FR plus FA rates), CPV can mitigate the impact of such delays by performing more delay measurements prior to location verification. This work highlights the importance of including mitigation capabilities while designing security-sensitive applications and protocols to deal with the effect of wireless delays. This will become increasingly important with the ubiquitous use of mobile devices that is expected to increase with the introduction of new computing and communication paradigms such as the Internet of Things.

Index Terms—Location-aware Authentication; Location verification; Wireless testing.

1 Introduction

The number of location-sensitive services is increasing over the Internet, e.g., location-based authentication [1], [2], location awareness in cognitive networks [3], geo-restricted media streaming [4]. Location information can also supplement contextual information about objects in Internet of Things (IoT) [5]. Previous literature [6], [7] shows that common location determination techniques, such as tabulation-based IP geolocation (e.g., MaxMind[1]), GPS [8], [9], and measurement-based geolocation [10], [11], are not reliable in the face of adversarial clients wishing to forge their own locations. A location verification protocol was ergo required to satisfy the needs of security-sensitive location-based services.

CPV [12] (see the appendix for a brief summary) was proposed earlier as a realtime delay-based Internet location verification technique that mitigates common geolocation-evasion tactics. In CPV, three verifiers iteratively estimate One-Way Delays (OWDs) between themselves and a client, and use these delays to corroborate the clients' geographic presence inside the triangle determined by their geographic positions. CPV was evaluated [12] with clients connected through wired access networks, using delay measurements from PlanetLab [13]. The evaluation was performed by having sets of three verifiers (running on PlanetLab nodes) measure OWDs to/from the clients (other PlanetLab nodes) using different OWD-estimation protocols [14]. The measured OWDs were logged, along with the ground-truth about *legitimate clients* (PlanetLab nodes inside the verification triangle) and *adversaries* (nodes outside the triangle). CPV was then run locally on the collected delay logs, and the rates of false reject/accept were quantified.

The nature of delays in wireless and wireline networks is different due to multiple factors such as characteristics of the shared medium, effect of signal strengths, difficulty of collision detection, and the possibility of hidden terminals [15]. This paper evaluates CPV when legitimate clients are connected through wireless access networks. We refer to those clients as *wireless clients*. A wireless client is assumed to be one-hop away from its access point, which serves as the client's gateway to the Internet. Beyond the gateway, all hops until the verifiers are assumed to be wired. We examine various factors that affect CPV's efficacy, including the number of devices actively competing for the wireless media in the vicinity of a wireless legitimate CPV client.

To evaluate CPV with wireless clients, we leverage Internet delay information collected for wired clients from PlanetLab, and model additional delays representing the last-mile wireless link. The additional delays were generated following wireless delay models studied in the literature [16], [17], [18]. This evaluation methodology addresses the effect of delays in wireless networks, while retaining the advantages of PlanetLab, e.g., real-world network delays, logical and geographical network topology, exterior gateway routing policies, and congestion behaviour. In addition, by using the data logs collected from the wired evaluation of Abdou *et al.* [12], we unify all experimental parameters across wireless and wireline testing. Root causes of improvement/retrogression can then be more reliably identified.

We evaluate the case with wireless legitimate clients

- *A. Abdou is with the School of Computer Science, Carleton University. Email: abdou@scs.carleton.ca*
- *A. Matrawy is with the School of Information Technology, Carleton University. Email: ashraf.matrawy@carleton.ca*
- *P.C. van Oorschot is with the School of Computer Science, Carleton University. Email: paulv@scs.carleton.ca*

Version created on: August 29, 2016. The final version of this paper will appear in the IEEE Transactions on Emerging Topics in Computing (TETC).

1. https://www.maxmind.com

and wireline adversaries.[2] Wireless adversaries are not considered because wireless networks tend to, among other effects, increase delays and the delay variance, which in CPV increase the likelihood of rejecting assertions. Therefore, by modelling wireless legitimate clients and wireline-connected adversaries, we test CPV in the most demanding (to the defender) situation among the four possible combinations of (two) access networks and (two) types of clients, i.e, legitimate and adversary.

Our evaluation shows that wireless access networks do affect CPV's correctness. However, increasing the number of CPV iterations can address that, as shown herein, and mitigates such an effect on CPV. We thus propose a formal means (Section 4) by which CPV can compute the appropriate number of iterations to be performed for the wireless effect to be almost neutralized.

Contributions. This paper aims to study the impact of the varying wireless delays on CPV by exploring the following three factors: (1) the number of devices connected to the wireless access network used by a legitimate CPV client; (2) the minimum distance the adversary should be away from the triangle's nearest side so that CPV correctly rejects it;[3] and (3) the number of CPV iterations the verifiers should perform in order to essentially eliminate the effect of the additional wireless delays.

Outline. Section 2 reviews recent literature that models delays of single-hop wireless networks. The reviewed models are then used to evaluate CPV in Section 3. Section 4 analyzes the effect of the number of iterations on the efficacy of CPV when legitimate clients are using wireless access networks. Related work is discussed in Section 5, and a conclusion is provided in Section 6.

2 Background: Wireless Delay Models in the Literature

This section reviews three wireless delay models in the literature, both assume a single-hop wireless network with one access point and k wireless devices. The k devices are *saturated*, i.e., always have frames to send. In the first two models, the channel is assumed ideal, meaning that the only source of frame corruption is collision. The third (Section 2.4) considers the effect of non-ideal channel conditions by incorporating the Signal to Noise Ratio (SNR) as a factor while deriving the probability distribution delays.

Note that the focus of this section is not to compare the three wireless delay models, nor not to evaluate their accuracies. We rather review these models to use them in evaluating CPV later in Sections 3 and 4 below.

2.1 Average back-off time at a stage

Carvalho and Garcia-Luna-Aceves [16] derived the average time a device spends backing off. In Distributed Coordination Function (DCF) [19], a device backs-off for $X = \mathcal{U}\{0, 2^m \cdot W_{\min}\}$ time slots. Thus, the expected backing-off time, α, is the time spent while counting down X time slots plus the time where the countdown is paused during a transmission [16]:

$$\alpha = \sigma p_i + t_c p_c + t_s p_s \tag{1}$$

The constant σ is the length of the time slot (in μsec); p_i is the probability the channel is idle (i.e., the subscript is not an index, it denotes "idle") during a time slot; and p_c and p_s are the probabilities of collision and successful transmission respectively during a time slot. t_s and t_c are the number of time units a device spends while pausing the countdown during a successful transmission and during a transmission with collision respectively. Bianchi *et al.* [20] expressed these durations as follows:

$$t_s = \frac{l(\text{RTS}) + l(\text{CTS}) + l(\text{MAC}) + l(\text{DTA}) + l(\text{ACK})}{\text{rate}} + (3 \cdot \text{SIFS} + \text{DIFS}) + 4\delta \tag{2}$$

$$t_c = \frac{l(\text{RTS})}{\text{rate}} + \text{DIFS} + \delta \tag{3}$$

where the function $l(.)$ indicates the frame (or packet) length in bits; RTS/CTS are the Ready/Clear To Send frames [15]; δ is the propagation delay (in μsec); SIFS is a technology-specific amount of time (in μsec); MAC, DTA and ACK are the header, data packet, and ACK packets respectively; and *rate* is the media's transmission rate in Mbps.

Using a 2-dimensional discrete-time Markov process, Bianchi *et al.* derived the probability, ψ, that a transmission occurs (successful or with collision) at a time slot as:

$$\psi = \frac{2(1-2p)}{(1-2p)(W_{\min}+1) + pW_{\min}(1-(2p)^m)} \tag{4}$$

where p is the probability of collision occurring at a time slot. Note that p is different from p_i, p_c and p_s in (1). Bianchi *et al.* [20] then assumed that a packet collides with a constant and independent probability regardless of the number of retransmissions it suffers. Assuming k devices in the network, if one device transmits, the only case that results in no collision is when none of the $k-1$ other devices transmit, i.e., the probability of no collision is $(1-\psi)^{k-1}$. Therefore, p can be expressed in terms of ψ as [20]:

$$p = 1 - (1-\psi)^{k-1} \tag{5}$$

Thus, the relationship between p and ψ is non-linear. Carvalho and Garcia-Luna-Aceves [16] linearized this model in order to use ψ to derive the expected total back-off time (see Section 2.2.1 below).

Using ψ and assuming k devices, the probability (P_{tr}) that at least one of the k devices is transmitting, and the probability (P_{suc}) that a transmission for any of the k devices is successful are calculated as follows [20], [21]:

$$P_{\text{tr}} = 1 - (1-\psi)^k \tag{6}$$

$$P_{\text{suc}} = \frac{k\psi(1-\psi)^{k-1}}{P_{\text{tr}}}$$

The probabilities p_i, p_c and p_s in (1) are calculated as $p_i = 1 - P_{\text{tr}}$, $p_c = P_{\text{tr}}(1 - P_{\text{suc}})$, and $p_s = P_{\text{tr}}P_{\text{suc}}$ [20].

2. The case when both, legitimate clients and adversaries, were using a wired access network was evaluated in previous literature [12].

3. As noted by the authors of CPV [12], the chosen parameterization affects the rates of false reject/accept. Thus, the presence of wireless legitimate clients is expected to affect, not only the rate of false rejects, but also false accepts.

2.2 Using the Model of Carvalho and Garcia-Luna-Aceves

2.2.1 Expected total back-off time

Carvalho and Garcia-Luna-Aceves [16] give an approximate solution to the nonlinear relation between ψ in (4) and p in (5), and reduce ψ to:

$$\psi_{\mathrm{A}} = \frac{2W_{\min}}{(W_{\min}+1)^2}(1-p) \tag{7}$$

Using (7), the authors derived p independent of ψ_{A} as [16]:

$$p = \frac{2W_{\min}(k-1)}{(W_{\min}+1)^2 + 2W_{\min}(k-1)}$$

Carvalho and Garcia-Luna-Aceves [16] then used this approximation to obtain α in terms of σ, k, $W_{\min}$, t_s and t_c, as explained above. Finally, they derived the expected time a device backs off $\overline{T}_B$ as [16]:

$$\overline{T}_B = \frac{\alpha(W_{\min}F - 1)}{2q} + \left(\frac{1-q}{q}\right)t_c \tag{8}$$

where

$$F = \frac{q - 2^m(1-q)^{m+1}}{1-2(1-q)}$$

and $q = 1-p$ represents the probability of no collision.

2.2.2 Mean delay and jitter

Carvalho and Garcia-Luna-Aceves [16] expressed the expected delay $E_{\mathrm{A}}[T]$ of a frame from (8) and (2) as follows:

$$E_{\mathrm{A}}[T] = \overline{T}_B + t_s \tag{9}$$

The variance of T was derived as:

$$\mathrm{Var}[T] = \left[\frac{\alpha(W_{\min}\gamma - 1)}{2} + t_c\right]^2 \frac{1-q}{q^2}$$

where

$$\gamma = \frac{(2q^2 - 4q + 1 - mq(2q-1))(2-2q)^m + 2q^2}{(2q-1)^2}$$

Thus the jitter (or the standard deviation) is:

$$\mathrm{Std}_{\mathrm{A}}[T] = \sqrt{\mathrm{Var}(T)} \tag{10}$$

2.2.3 Assuming a Truncated Gaussian Distribution

Carvalho and Garcia-Luna-Aceves [16] only provide information about the mean and jitter of the delays given some number of wireless devices, k. We assume that delays will follow a Gaussian distribution with mean and variance derived as in (9) and (10) respectively. However, since the x-axis of the Gaussian distribution (which would be the delays in that case) goes from $-\infty$ to ∞, the model can result in negative delay values. Thus, we assume a truncated Gaussian [22] in the range $[0, \infty]$.

The mean of the Gaussian distribution truncated from a to b is given by [22]:

$$\mathrm{GausMean}_{\mu,\sigma}(a,b) = \mu - \sigma \cdot Z(\alpha,\beta)$$

where μ and σ are respectively the mean and standard deviation of the parent (non-truncated) Gaussian distribution; $\alpha = (a-\mu)/\sigma$ and $\beta = (b-\mu)/\sigma$; and the function $Z(\alpha,\beta)$ is defined as:

$$Z(\alpha,\beta) = \frac{\phi(\beta) - \phi(\alpha)}{\Phi(\beta) - \Phi(\alpha)}$$

The functions $\phi(.)$ and $\Phi(.)$ are respectively the PDF and the CDF of the standard Gaussian distribution (i.e., with $\mu = 0$ ms and $\sigma = 1$ ms).

The standard deviation of the Gaussian distribution truncated from a to b is [22]:

$$\mathrm{GausStd}_{\mu,\sigma}(a,b) = \sqrt{\sigma^2 \cdot \left(1 - \frac{\beta\cdot\phi(\beta) - \alpha\cdot\phi(\alpha)}{\Phi(\beta)-\Phi(\alpha)} - Z^2(\alpha,\beta)\right)}$$

To obtain a CDF of the wireless delays that has a mean and standard deviation as in (9) and (10), we need to solve simultaneously for μ and σ:

$$\mathrm{GausMean}_{\mu,\sigma}(0,\infty) = E_{\mathrm{A}}[T] \tag{11}$$

and

$$\mathrm{GausStd}_{\mu,\sigma}(0,\infty) = \mathrm{Std}_{\mathrm{A}}[T] \tag{12}$$

Those are two equations in two unknowns, which can be solved using numerical methods [23]. Table 1 shows the mean and standard deviations calculated using (9) and (10) for various values of k, and the corresponding μ and σ of the parent (non-truncated) Gaussian distribution calculated by solving (11) and (12) simultaneously.

TABLE 1
Mean μ, and standard deviation σ, of the single-hop wireless delays when k devices are simultaneously competing with the media.

Parameters (ms)	Eqn.	k				
		2	5	10	20	30
$E_{\mathrm{A}}[T]$	(9)	2	5	12	40	87
$\mathrm{Std}_{\mathrm{A}}[T]$	(10)	0.6	4.7	21	89	186
μ	–	-110	-246	-691	-2419	-5156
σ	–	15	36	95	328	700

Using μ and σ, the CDF of the Gaussian distribution truncated from a to b is [22]:

$$\mathrm{GausCDF}_{\mu,\sigma}(x;a,b) = \frac{\Phi(\zeta) - \Phi(\alpha)}{\Phi(\beta) - \Phi(\alpha)} \tag{13}$$

where $\zeta = (x-\mu)/\sigma$. Figure 1a plots the delay distribution, $\mathrm{GausCDF}_{\mu,\sigma}(x;0,\infty)$, using (13) for various values of k. Unsurprisingly, the chart shows that the wireless delays generally increase with k. These delay distributions are used in Sections 3 and 4 to evaluate CPV in wireless networks.

The model of Carvalho and Garcia-Luna-Aceves provides an upper bound on the average delay a frame is expected to suffer [16]; when they compared their model to simulations, delays from the simulations were always smaller, which the authors [16] expect could be due to that there is a non-zero probability that a frame backs off indefinitely. However, the DCF standard [19] specifies that the MAC layer must discard the frame if transmission failed after R back off trials, for some predefined value of R.

2.3 Using the model of Raptis *et al.*

Similar to Carvalho and Garcia-Luna-Aceves [16], Raptis *et al.* [17] used the basis of Binachi [20] to derive a CDF (and jitter) for the single-hop 802.11 access delays. However, Raptis *et al.* [17] took into consideration the reality that the frame being transmitted will be discarded after failing transmission in R back-off stages. Thus, they started by deriving the expected delay that a frame suffers after a failed transmission at stage j ($0 \leq j \leq R$) as [17]:

$$U_j = (j+1) \cdot t_c + \alpha \cdot \sum_{i=0}^{j} \frac{W_i - 1}{2} \tag{14}$$

where t_c and α are analogous to those in (3) and (1) respectively, and

$$W_i = \begin{cases} 2^i \cdot W_{\min}, & \text{if } 0 \leq i \leq m \\ 2^m \cdot W_{\min}, & m < i \leq R \end{cases} \tag{15}$$

To derive the CDF of delays, Raptis *et al.* [17] first calculated the probability that a frame is successfully transmitted at stage j:

$$Q_j = \frac{p^j(1-p)}{1-p^{R+1}} \tag{16}$$

Since at any stage j, selecting any back-off value in the range $0 \leq i < W_j$ is equiprobable, then the probability of transmitting a frame at stage j after backing off for i stages is (independent of i) [17]:

$$P_j = Q_j \cdot \frac{1}{W_j} \tag{17}$$

Using (1), (2) and (14), Raptis *et al.* [17] derived the CDF of delays as follows. Let Ω be a finite set of delays, such that $\Omega_{j,i}$ is the delay a frame suffers before it gets successfully transmitted at stage j, given that i back-off slots were selected at stage j. The average of $\{\Omega_{0,0}, .. \ \Omega_{j,i}\}$, is calculated as [17]:

$$E[\Omega_{j,i}] = t_s + i \cdot \alpha + U_{j-1} \tag{18}$$

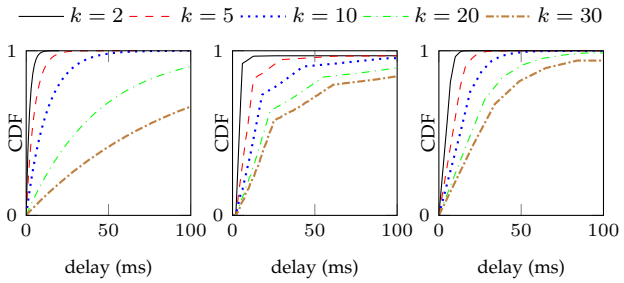
For any randomly-chosen delay value D, the probability that $D \leq d$ for all $0 \leq d \leq \infty$ is given by [17]:

$$P\{D \leq d\} = \sum_{j=0}^{R} \sum_{i=0}^{W_j - 1} P_{j,i}(d) \tag{19}$$

where

$$P_{j,i}(d) = \begin{cases} P_j, & \text{if } E[\Omega_{j,i}] \leq d \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

Using (19), Fig. 1b plots the wireless delay CDFs of Raptis *et al.* [17] at various values of k. Similar to the previous model, the model shows that delays generally increase with k, which is unsurprising. However the distributions derived by Raptis *et al.* [17] (Fig. 1b) are not exactly similar to those derived by Carvalho and Garcia-Luna-Aceves [16] (Fig. 1a). Differences between both models are discussed in Section 2.5 below.



(a) Carvalho [16] (b) Raptis [17] (c) Maadani [18]

Fig. 1. CDF of single-hop wireless delays that a frame endures when there are k saturated wireless devices in the network. A truncated Gaussian is used in (a) with means and standard deviations derived by Carvalho *et al.* [16].

2.3.1 Jitter

Similar to Carvalho and Garcia-Luna-Aceves [16], Raptis *et al.* [17] also derived an expression for the delay jitter in a single-hop wireless network with k devices. The authors [17] first derived the expected total delay that a frame suffers before being successfully transmitted at stage j:

$$\omega_j = U_j - t_c + t_s \tag{21}$$

Then, using (21) and (16), the expected delay, $E[T]$, a frame suffers before being successfully transmitted is [17]:

$$E_{\text{B}}[T] = \sum_{j=0}^{R} (\omega_j \cdot Q_j) \tag{22}$$

And the expected value for the square of a delay, T^2, is [17]:

$$E_{\text{B}}[T^2] = \sum_{j=0}^{R} \left(P_j \cdot \sum_{i=0}^{W_j - 1} (E[\Omega_{j,i}])^2 \right) \tag{23}$$

Finally, in contrast to the delay jitter of Carvalho and Garcia-Luna-Aceves [16] in (10), the jitter of Raptis *et al.* [17] is calculated using (23) and (22) as:

$$\text{Std}_{\text{B}}[T] = \sqrt{E_{\text{B}}[T^2] - (E_{\text{B}}[T])^2} \tag{24}$$

2.4 Using the model of Maadani and Motamedi

Also basing their derivations on Binachi's model [20], Maadani and Motamedi [18] started by deriving the probability that a transmission occurs at a time slot from (4):

$$\psi_{\text{C}} = \psi(1 - P_{\text{tr}}) \tag{25}$$

where P_{tr} is calculated as in (6). The probability, p, that a collision occurs at a time slot is [18]:

$$p = p_e + P_{\text{tr}} - p_e P_{\text{tr}} \tag{26}$$

where

$$p_e = p_{\text{Data}} + p_{\text{ACK}} - p_{\text{Data}} p_{\text{ACK}} \tag{27}$$

The probabilities p_{Data} and p_{ACK} are such that:

$$p_{\text{Data}} = 1 - (1 - p_b)^{l(\text{MAC}) + l(\text{DTA})} \tag{28}$$

and

$$p_{\text{ACK}} = 1 - (1 - p_b)^{l(\text{ACK})} \tag{29}$$

where $p_b = 0.5e^{-\text{SNR}}$.

The non-linear equations (25) and (6) are to be solved simultaneously for ψ_C and P_{tr} to obtain their values [18]. The expected back-off time α is then calculated as in (1).

Finally, for any delay value D, the probability that $D \leq d$ for all $0 \leq d \leq \infty$ is given by [18]:

$$P\{D \leq d\} = \sum_{L=0}^{\infty} P_L(d) \tag{30}$$

where

$$P_L(d) = \begin{cases} (1-p)p^L, & \text{if } D(\alpha, L+1) \leq d \\ 0, & \text{otherwise} \end{cases}$$

Maadani and Motamedi define the function $D(.)$ as [18]:

$$D(\alpha, i) = (i-m)(\alpha \cdot W_m \cdot p^m) + \sum_{j=0}^{m} \alpha \cdot W_j \cdot p^j$$

Using (30), Fig. 1c plots the wireless delay CDFs of Maadani and Motamedi *et al.* [18] at various values of k. In Sections 3 and 4, we use the CDFs in (13), (19) and (30) to evaluate CPV.

2.4.1 Mean delay and jitter

Maadani and Motamedi first derive the mean delay as [18]:

$$E_C[T] = \left(\frac{\alpha W}{2}\right)\left(\frac{1-(2p)^m}{1-2p} + \frac{(2p)^m}{1-p}\right)$$

The authors then derive the jitter as a function of the mean delay [18]:

$$\text{Std}_C[T] = \sqrt{E_C[T^2] - (E_C[T])^2} \tag{31}$$

where

$$E_C[T^2] = \left(\frac{\alpha W}{2}\right)^2\left(\frac{1-(4p)^m}{1-4p} + \frac{(4p)^m}{1-p}\right)$$

2.5 Differences between the models

Figure 2a plots the truncated Gaussian distribution with the parameters obtained from the model of Carvalho and Garcia-Luna-Aceves [16] modeling single-hop wireless delays, the distribution derived by Raptis *et al.* [17], and that derived by Maadani and Motamedi [18] at $k = 2$ and $k = 10$. The distributions are not drastically different. Their dissimilarities might however stem from differences in their assumptions, e.g., Raptis *et al.* assumes the frame is discarded after failing transmissions in R stages, while Carvalho *et al.* does not make this assumption. Maadani *et al.* assume the probability of packet loss due to imperfect channel conditions, whereas the other two models assume transmission failures occur only due to collision.

Figure 2b shows the difference in the jitter between the models, obtained using (10), (24) and (31) respectively. At first glance, the individual values of the Raptis and Carvalho *et al.* curves over the region up to $k = 20$ are reasonably similar, but the model of Raptis *et al.* appears to grow almost linear, while that of Carvalho *et al.* gives values lower in the region up to $k = 20$, but rising much faster starting for values shortly beyond $k = 20$. On the other hand, the jitter of Maadani *et al.* grows significantly slower compared to the other two.

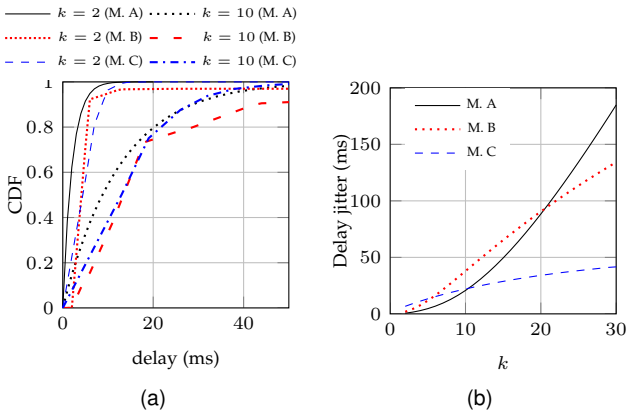


Fig. 2. Comparison of the reviewed models. M. A means using the model of Carvalho *et al.* [16]; M. B means that of Raptis *et al.* [17]; and M. C means that of Maadani *et al.* [18]. (a) Truncated Gaussian delay distribution with parameters derived from the model of Carvalho *et al.* [16], and the distributions derived by Raptis *et al.* [17] and Maadani *et al.* [18] at $k = 2$ and $k = 10$. (b) The jitter as derived by the respective authors [16], [17], [18].

In the rest of this paper, all three models are used to analyze CPV in wireless networks, with a truncated Gaussian distribution assumed for the parameters of Carvalho *et al.*

2.6 Summary of reviewed literature on wireless models

All the models reviewed herein, in Section 2, consider a wireless network with a single access point and no hidden terminals, typically addressing a small (e.g., home) network. In public places (e.g., coffee shops or hotel rooms), this may not be the case. However, the models already incorporate the additional delays due to the RTS/CTS mechanism of the 802.11 DCF and thus, we believe the existence of hidden terminals is unlikely to result in significant differences.

Finally, the reviewed literature assumes all k devices are saturated (i.e., always have packets to send). However, k devices are typically expected to alternate between phases of transmission, reception and idle activity. We suspect that this assumption will tend to cause the delays resulting from the derived models to be larger than those in practice.

3 Evaluating CPV in 802.11 Networks

We now evaluate CPV with wireless clients using the delay models discussed in Section 2. Our objective is to study the impact of the varying wireless delays on CPV by exploring the following two questions:

1. Assuming k wireless devices actively competing for the wireless media with the legitimate client, how does k affect CPV? Here, the number of wireless legitimate clients is varied, and CPV's efficacy is analyzed. We test by modeling clients using IEEE 802.11b as a representative access technology. This analysis is presented in Section 3.4.

2. For a given triangle verifying assertions of wireless legitimates and a wired adversary, what is the minimum distance the adversary should be away from the triangle's nearest side so that CPV correctly rejects it? To answer this question, we test CPV when varying the width of the adversary-free region

outside the triangle. We do this by progressively excluding nearby adversaries from the experiments and reevaluating CPV. This analysis is presented in Section 3.5.

3.1 A review of CPV's parameterization

We review CPV's parameterization in that section. A brief description of the algorithm is provided in the appendix.

In CPV [12], when a client asserts to be at a geographic location, three verifiers encompassing this assertion within the triangle determined by their geographic locations are selected to verify this assertion. The verifiers estimate OWDs between themselves and the client iteratively $n_{\triangle}$ times, where $\triangle$ is the triangle determined by their locations. CPV then maps these delays to distances; the client's presence inside $\triangle$ is then verified by comparing the sum of the areas of the three triangles determined by each pair of verifiers and the client with the area of $\triangle$.

CPV counts the event that the absolute difference between those two values is $\leq \epsilon_{\triangle}$ km^2 as an evidence supporting the client's presence inside $\triangle$. Otherwise, the event is counted as an evidence denying the presence thereof. The location assertion is finally accepted as true if the number of events with supporting evidence exceeds $n_{\triangle} \cdot \tau_{\triangle}$, where $0 < \tau_{\triangle} \leq 1$. All three parameters ($\epsilon_{\triangle}$, $\tau_{\triangle}$, and $n_{\triangle}$) are periodically calibrated for each $\triangle$ (i.e., calibration occurs independently between sets of three verifiers) [12].

Abdou *et al.* [12] define the function $awy(\triangle, g)$ for any point $g = \{\text{latitude}, \text{longitude}\}$ inside $\triangle$ as "the ratio of the distance between g and the side $z^g_{\triangle}$ to the length of $z^g_{\triangle}$, where $z^g_{\triangle}$ is the closest side to g". All results reported in this paper follow CPV's recommendation [12]: no triangle is used to verify an assertion at a location g such that $awy(\triangle, g) \geq \lambda$, where $\lambda = 0.1$.

3.2 Evaluation Setup

We evaluate CPV by quantifying the False Reject (FR) and False Accept (FA) rates at some values for the input parameters (see Section 3.1) that allow CPV to adequately distinguish legitimates from adversaries.

We use the same PlanetLab setup of Abdou *et al.* [12]; there were 49 legitimate clients available in the authors' PlanetLab experiments at $\lambda = 0.1$. Thus, we can model a maximum of 49 distinct wireless access networks, each having one wireless CPV client. This client is assumed to compete for the wireless media with $k - 1$ other wireless devices, where $k \geq 2$. In such case, k affects the last-mile wireless delays of that CPV client. Because the delay models reviewed in Section 2 are functions of k, we can model the number of devices in the vicinity of the CPV client.

Of those 49 legitimate wireless client, we assume that only a proportion of them is using a wireless access network. Note that the proportion of wireless clients modeled in the experiments is different from the number of wireless devices, k. For example, if a proportion of 0.2 of all 49 legitimate clients was using a wireless access network with $k = 4$, this means there are 10 wireless clients modeled at distinct wireless networks, and each network has 4 wireless devices (constant across all 10 networks) including the wireless CPV client itself. Figure 3 shows an example of eight legitimate CPV clients (*legitimate*, as they are truly present inside the triangle), and a proportion equal to 0.5 of them is using a wireless access network that has $k = 2$ devices.

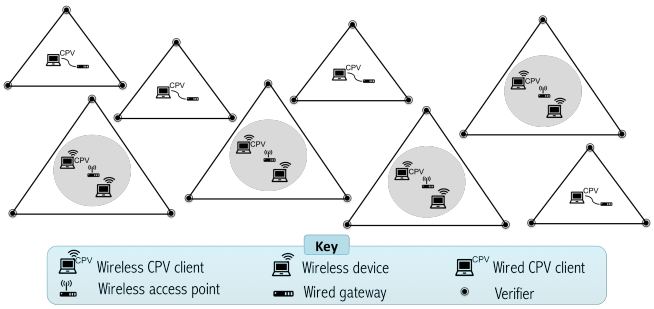


Fig. 3. An example of eight CPV clients, half of which are using a wireless access network that has $k = 2$ devices.

TABLE 2
DSSS characteristics

Item	Value
W_{min}	32 time slots
W_{max}	1024 time slots
Retransmission limit (R)	6 stages
Physical header (PHY)	192bits at 1 Mbit/s
MAC header	224 bits at 11 Mbit/s
ACK length	112 bits at 11 Mbit/s + PHY
RTS length	160 bits at 1 Mbit/s + PHY
CTS length	112 bits at 1 Mbit/s + PHY
Propagation delay (δ)	1 μsec
Slot time (σ)	20 μsec
SIFS	10 μsec
DIFS	50 μsec

All k devices are using an 802.11b access network over Direct-Sequence Spread Spectrum (DSSS) on the physical layer with a 11Mbps data rate. Characteristics of DSSS are shown in Table 2. While using the model of Maadani and Motamedi, we assume the SNR is 9.0 [18].

We assume that all k devices are saturated (the packet queues of all k device are never empty), and are transmitting at the same time according to a Constant Bit Rate (CBR) with a packet size equal to 8148 bits. We assume no hidden terminals [15]—the transmission of any device is sensed by all others. Note that those two assumption allows us to use the models in Section 2.

3.3 Statistical Confidence of the Results

Figure 4 shows the mean FRs and FAs of 100 runs resulting from using three models reviewed in the previous section. All 49 legitimate clients were using a wireless access network, and there was a total of $k = 5$ devices in the network of each wireless CPV client. The number of CPV iterations (see Section 3.1) was fixed at $n_{\triangle} = 600$ for all $\triangle$. FRs and FAs using all three models lied between $\sim$1.8% and $\sim$4.5%.

Because FRs and FAs are estimated empirically from 100 runs, we calculate their confidence interval (CI) [24]. At 90% confidence level, the critical probability is:

$$p* = 1 - \frac{\alpha}{2} = 1 - \frac{0.1}{2} = 0.95$$

At a degree of freedom equal to 99 and $p* = 0.95$, the critical value (from the statistics tables [24]) is 1.66.

TABLE 3
SE and Margin of Error (ME) at 90% confidence level for the rest of the results

Model	Parameter	Std	SE	ME at 90% CI
Carvalho *et al.* [16]	FRs	0.97	0.097	±0.16
	FAs	0.74	0.074	±0.12
Raptis *et al.* [17]	FRs	0.92	0.092	±0.15
	FAs	0.14	0.014	±0.02
Maadani *et al.* [18]	FRs	0.79	0.079	±0.13
	FAs	0.66	0.066	±0.11

Std = Standard deviation; SE = Standard error; ME = Margin of Error.

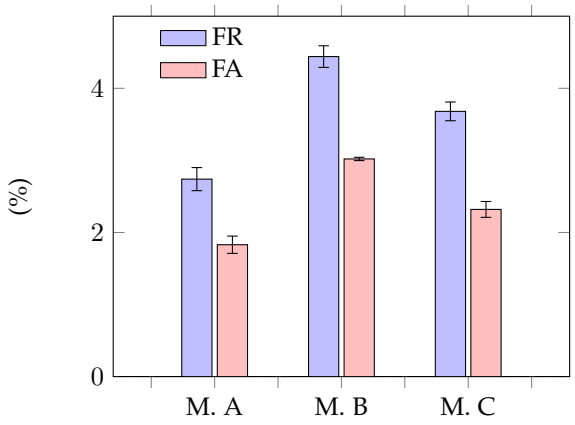


Fig. 4. Statistical confidence of CPV results in wireless networks. Models A, B and C are those of Carvalho *et al.* [16], Raptis *et al.* [17], and Maadani *et al.* [18] respectively.

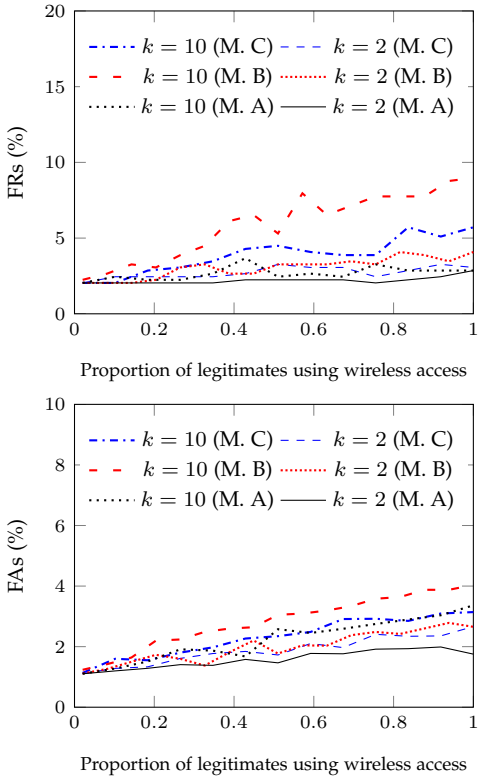


Fig. 5. Results when a proportion of the 49 legitimate clients (i.e., PlanetLab nodes inside triangles) use a wireless access network that has k wireless devices. $n_{\triangle} = 600$ CPV iterations for all $\triangle$. M. A means using the model of Carvalho *et al.* [16]; M. B means using the model of Raptis *et al.* [17].

For the FRs obtained using the model of Carvalho *et al.* [16], the standard error (SE) is:

$$\text{SE (FRs)} = \frac{\text{Std}}{\sqrt{n}} = \frac{0.97}{\sqrt{100}} = 0.097 \tag{32}$$

giving a CI of 1.66 × 0.097 = 0.16 at 90% confidence level. Table 3 shows the confidence intervals for the remaining five results.

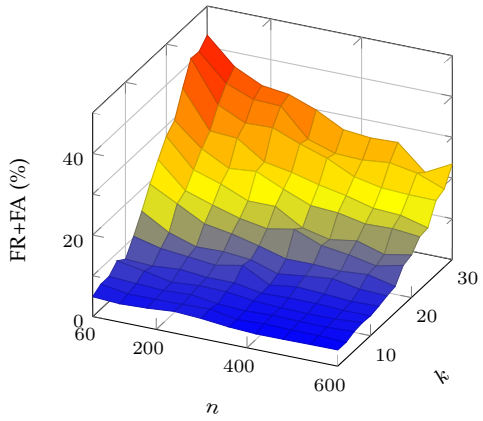
The CIs at 90% confidence level are depicted using vertical lines atop the bars in Fig. 4 for the mean FRs and FAs. None of the MEs exceeds ±0.16%, highlighting that the means estimated from the sample runs are relatively precise. For the rest of the results in this section, the average of 10 runs is reported for each experimental scenario.

3.4 Effect of the number of wireless devices (k) on CPV

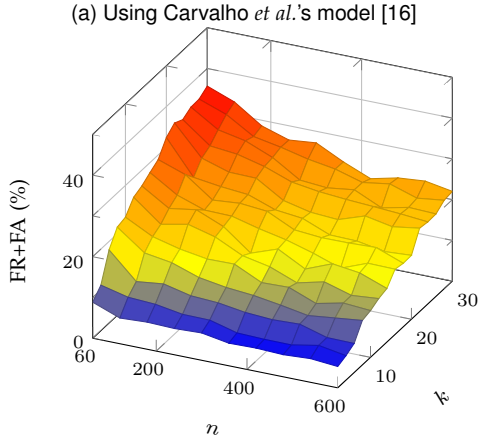
Figure 5 shows the FRs and FAs when the proportion of wireless clients is varied from 0 to 1; the number of CPV iterations was fixed at $n_{\triangle} = 600$ for all $\triangle$. Using the model of Carvalho and Garcia-Luna-Aceves [16], there was a non-sever degradation in CPV's efficacy with an increased k. For example, when all 49 legitimate clients were using a wireless access network (i.e., at $x = 1$ in Fig. 5), the sum FR+FA went from ∼4.61% at $k = 2$ to ∼6.22% at $k = 10$. We believe these results stem from the probability that the endured wireless delay is very small (or relatively negligible), e.g., 3 ms. From the truncated Gaussian distribution in Fig. 1a, at $k = 10$, there is a ∼20% chance the transmitted frame suffers <3 ms delay, i.e., if one iteration was performed. As more iterations are performed, the chances that one or more iterations result in such negligible delay increase. Because CPV requires only a proportion τ of the performed iterations to result in *supporting evidence* (which is more likely to happen with smaller delays between the verifiers and the client [12]) in order to accept a client, it still accepts a client when a proportion of $1 - \tau$ of all iterations result in large delays and *denying evidence* (see Section 3.1). The required number of iterations is derived in terms of k and the acceptance threshold τ in Section 4 below.

Using the model of Raptis *et al.* [17], and assuming that half the legitimate clients are wireless, the sum FR+FA went from 5.1% at $k = 2$ to 8.3% at $k = 10$. Those results are to be compared to 3.1% (2.0% + 1.1%) when none of the legitimate clients are using a wireless access network (i.e., at $x = 0$ in Fig. 5). In conclusion, under this model, when a wireless CPV client competes for the media with another device (i.e., $k = 2$), it has double the chances of being falsely rejected compared to a wired legitimate client. Finally, results obtained using the model of Maadani *et al.* [18] show that CPV's efficacy lies somewhere in between both models for the selected values of k.

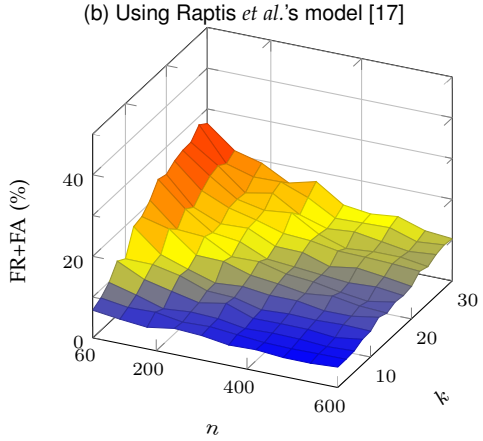
Figure 6 shows the summation of FRs and FAs with respect to the number of iterations n (i.e., $n_{\triangle}$ for all $\triangle$), and the number of wireless devices, k, in each wireless network when 25 of the 49 legitimate CPV clients are using a wireless



(a) Using Carvalho *et al.*'s model [16]



(b) Using Raptis *et al.*'s model [17]



(c) Using Maadani *et al.*'s model [18]

Fig. 6. FR+FA when half of the evaluated legitimate clients were using a wireless access network with k devices.

access network.[4] Using the model of Carvalho and Garcia-Luna-Aceves [16], the effect of k on the results begins to manifest starting around $k = 1$. For example, at $k = 2$ the sum FR+FA is almost constant regardless of the performed number of CPV iterations, n. In contrast, at $k = 30$, the impact of n on the sum FR+FA is large. In conclusion, increasing the number of CPV iterations has large impact only when more than $k = 15$ devices are present in each wireless network.

4. Recall that the number of wireless legitimate clients being verified by triangle $\triangle$ affects the calibration of $\epsilon_{\triangle}$ and $\tau_{\triangle}$, which is how those 25 wireless clients are expected to influence CPV's decisions on others.

The case is different using the wireless models of Raptis *et al.* [17], where k has a significant impact on the results, for all values of k. For example, at $k = 6$, the sum FR+FA decreases from ∼18% at $n = 60$ to ∼7% at $n = 600$; and at $k = 30$, FR+FA decreases from ∼36% at $n = 60$ to ∼22% at $n = 600$. These results highlight the potential for a larger number of iterations to mitigate the effect of the wireless delays on CPV.

Both models agree that CPV's efficacy decreases as k increases, suggesting that CPV may perform poorly in public places where numerous devices are actively competing for the media. However, the model of Maadani and Motamedi [18] suggests a better CPV efficacy with an increasing k, yet agrees with both previous models about the effect of n on CPV. We suspect such a difference is due to the large jitter disagreements between the models at large values of k, as we show in Fig. 2b (Section 2.5).

3.5 Minimum adversarial distance from the triangle

Figure 7 shows the minimum distance, between an (outside-triangle) adversary and the triangle encapsulating the adversary's asserted location, that enables CPV to maintain similar efficacy compared to when all clients are using a wired access network. Results are obtained when 25 of all 49 legitimate clients are using a wireless access network, and when $n_{\triangle} = 600$ iterations for all $\triangle$.

Using the model of Carvalho and Garcia-Luna-Aceves [16], and at $k = 5$, the sum FR+FA≈3% when (outside-triangle) adversaries were at least ∼250 km away from the triangles' sides. At $k = 15$, the minimum adversary-free distance outside the triangle that maintains FR+FA≈3% becomes $1,250$ km.

With the model of Raptis *et al.* [17], the minimum adversarial distance is 700 km at $k = 5$ (see Fig. 7) and ∼1,600 km at $k = 10$. In contrast, the model of Maadani and Motamedi resulted in a distance of 300 km and 700 km at those values of k respectively.

In conclusion, the minimum adversary-free distance outside the triangle for CPV to not be affected by wireless legitimate users clearly increases with k in all models. Therefore, as more saturated devices exist in the network of CPV's legitimate wireless clients, the likelihood of accepting (outside) adversaries close the triangles' sides increases.

4 Addressing Wireless Delays

It is noticeable from the previous section that increasing the number of CPV iterations, n, enhances the results even in the presence of wireless clients. We thus ask the question: *Assume that the number of wireless devices in the client's access network, k, is known to the verifiers; how many CPV iterations, whereby OWDs are measured between the client and the verifiers, should be performed such that with a very high probability the wireless client gets correctly accepted?*

To answer this question, let t be a small delay value (i.e., due to the wireless access network) that when added to the (Internet) end-to-end delays of a legitimate wired client that CPV would typically accept, will not cause CPV to falsely reject that client (i.e., due to the increased delay). Using the wireless delay models in Section 2, we can obtain the

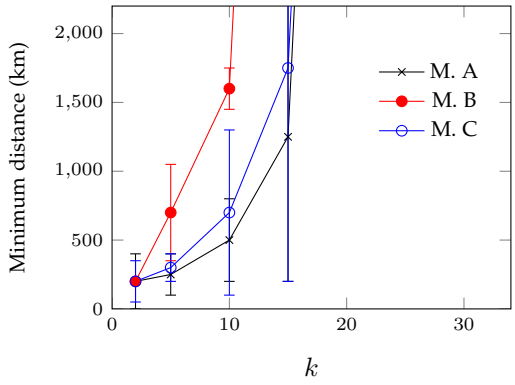


Fig. 7. The minimum distance, between the (outside) adversary and the triangle, that enables CPV to maintain similar efficacy compared to when all clients are using a wired access network. Results are obtained when 25 of all 49 legitimate clients are using a wireless access network, and when $n_{\triangle} = 600$ CPV iterations, for all $\triangle$. The error bars indicate the smallest and largest y (minimum distance) obtained from 10 runs, and the marker is their average. M. A means using the model of Carvalho *et al.* [16]; M. B means using the model of Raptis *et al.* [17]; and M. C means using the model of Maadani *et al.* [18].

probability $p_k(t) = P_k\{D < t\}$ that a transmitted frame experiences less than t ms additional delay while sharing the wireless media with $k-1$ other actively participating devices.

If two CPV iterations are performed, the probability that the frames experience $< t$ ms delay in one of them (either the first or the second) is:

$$\begin{aligned}\varrho_1(t,k,2) &= p_k(t)\cdot(1-p_k(t)) + (1-p_k(t))\cdot p_k(t) \\ &= 2\cdot p_k(t)\cdot(1-p_k(t))\end{aligned} \tag{33}$$

This equation is similar to the probability of getting a number x once from a dice rolled twice, such that $x < 3$ (i.e., the probability of getting either 1 or 2). This probability would be: either getting x from the first roll but not the second, or from the second roll but not the first; the number of dice rolls is analogous to the number of CPV iterations.

For three iterations:

$$\varrho_1(t,k,3) = 3\cdot p_k(t)\cdot(1-p_k(t))^2 \tag{34}$$

In general, the probability that a transmitted frame experiences $< t$ ms in exactly one of n iterations is given by:

$$\varrho_1(t,k,n) = n\cdot p_k(t)\cdot(1-p_k(t))^{n-1} \tag{35}$$

Considering more than one iteration, the probability ϱ_2 that the transmitted frames experience $< t$ ms in exactly two of n iterations is given by:

$$\varrho_2(t,k,n) = \left(\frac{n(n-1)}{2}\right)\cdot p_k(t)^2\cdot(1-p_k(t))^{n-2} \tag{36}$$

That is because there are $n(n-1)/2$ ways of choosing two of n iterations. In general, there are ${}^nC_r$ ways of choosing r of n iterations, where:

$${}^nC_r = \frac{n!}{r!(n-r)!} \tag{37}$$

Accordingly, the probability that the transmitted frames experience $< t$ ms in exactly r of n iterations is given by:

$$\varrho_r(t,k,n) = {}^nC_r\cdot p_k(t)^r\cdot(1-p_k(t))^{n-r} \tag{38}$$

TABLE 4
The probability $p_k(3)$ that an additional delay of < 3 ms is incurred by the wireless network at different values of k.

	Model	k					
		2	5	10	20	25	30
$p_k(3)$	[16]	0.77	0.45	0.21	0.07	0.04	0.03
	[17]	0.24	0.08	0.04	0.02	0.02	0.02
	[18]	0.34	0.18	0.16	0.08	0.07	0.06

And thus, the probability that the wireless delay is $< t$ ms in at least r of n iterations is given by:

$$\rho_r(t,k,n) = \sum_{i=r}^{n} \varrho_i(t,k,n) \tag{39}$$

Calculating this probability is fundamental to the operation of CPV. For example, let the number of iterations that CPV performs be $n = 600$, and let CPV be calibrated such that it requires at least 30 of those 600 iterations to pass the triangular area check [12]. Assuming that $t = 3$, then using (39) we can calculate the probability, $\rho_{30}(3,k,600)$, that the timestamps exchanged between the verifiers and the client are delayed (additionally by the wireless access network) < 3 ms in at least 30 of the 600 iterations. This probability will thus serve as an upper bound probability of that client being correctly accepted. It is "upper bound" because if $\rho_{30}(3,k,600) = 1$, the client may still get falsely rejected due to other non-wireless factors [12]. Equation (39) is used below to derive a function calculating the number of CPV iterations required to mitigate the negative effect of wireless delays.

Note that $p_k(t)$ is calculated using (13), (19) and (30). For example, for the model of Carvalho *et al.*, we have:

$$p_k(t) = \text{GausCDF}_{\mu,\sigma}(t;0,\infty) \tag{40}$$

where μ and σ are functions of k as discussed in Section 2. Example values for $p_k(3)$ are listed in Table 4.

Figure 8 shows a plot of $\rho_5(3,k,n)$ and $\rho_{20}(3,k,n)$ against n at $k = 2$ and $k = 10$. The charts show that at $k = 2$, the verifiers need to perform 11, 45, or 30 iterations using the three models respectively to be almost certain (i.e., with probability $\rho_5(3,2,n) \geq 0.99$) that the transmitted frames will endure < 3 ms delay in at least 5 iterations. To achieve < 3 ms wireless delay in 20 or more iterations, and at $k = 10$, the verifiers will need to perform ∼150, ∼700, or ∼270 iterations respectively using respectively the three models to satisfy $\rho_{20}(3,10,n) \geq 0.99$.

CPV requires a proportion $0 < \tau_{\triangle} \leq 1$, for each $\triangle$, to result in evidence supporting the client's presence inside the triangle (see Section 3.1) in order to accept a client. By policy, if $n\cdot\tau_{\triangle}$ of the n iterations pass the area checks, the client gets accepted. To mitigate the effect (on CPV's decisions) of wireless delays with probability ≥ 0.99, the verifiers need to perform n iterations that satisfy:

$$\rho_{n\tau_{\triangle}}(t,k,n) \geq 0.99 \tag{41}$$

Using linear iterative root finding [23], we solved (41) for n at various values of k. A plot of both variables is shown in Fig. 9 for different values of τ. Once again, the differences between the wireless delay models in the reviewed literature manifest in our analysis. For example, assuming $k = 5$,

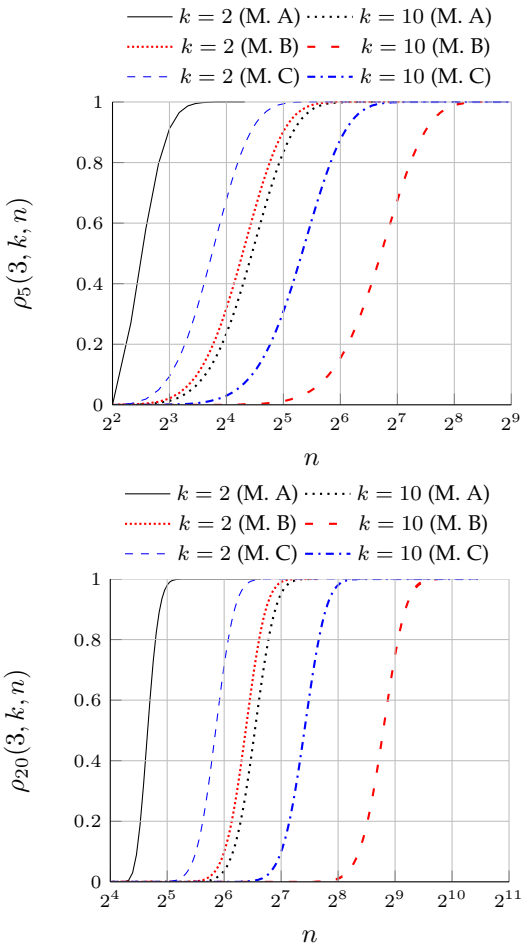
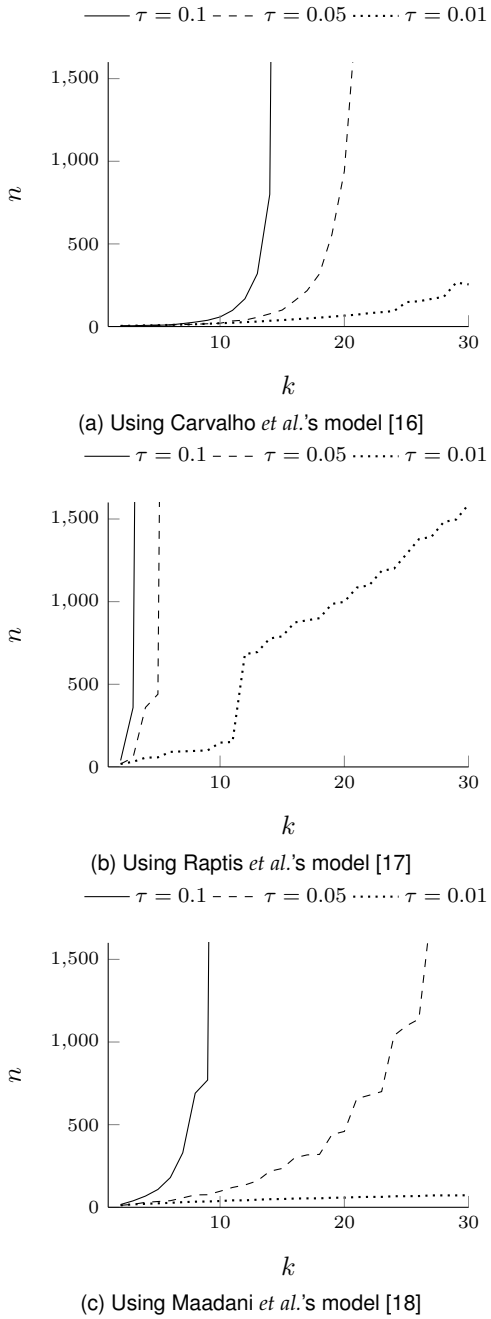


Fig. 8. The probability that a transmitted frame experiences $< t = 3$ ms of wireless delay in at least 5 and 20 of n iterations, when k wireless devices are sharing the access network. See Table 4 (or similarly Figures 1a and 1b at $x = 3$ ms) for the values of $p_k(t)$. M. A means using the model of Carvalho *et al.* [16]; M. B means using the model of Raptis *et al.* [17].



(a) Using Carvalho *et al.*'s model [16]

(b) Using Raptis *et al.*'s model [17]

(c) Using Maadani *et al.*'s model [18]

Fig. 9. Required number of iterations to essentially eliminate the effect of wireless network delays at different values of τ.

using the model of Carvalho and Garcia-Luna-Aceves [16], if $\tau = 0.05$, then only 8 iterations are required to mitigate the effect of the wireless delays on CPV, versus 440 iterations using the model of Raptis *et al.* [17], and 35 using that of Maadani *et al.* [18]. At $k = 30$ wireless devices, and $\tau = 0.01$, the required number of iterations is ∼250, ∼1590, and ∼75 respectively, asserting the optimistic nature of Maadani *et al.*'s model at large values of k.

5 Related Work

Considerable literature (e.g., [25], [26]) exists evaluating the efficacy of network measurements in determining the locations of (or geolocating) Internet hosts. The evaluations commonly find correlation between network delays and geographic distances, reporting varying degrees of geolocation accuracy. However, to the best of our knowledge, no literature examines the effect of wireless Internet clients on such techniques. Our work herein sheds light on the potential effect of wireless access on the accuracy of measurement-based protocols (albiet location *verification* rather than *determination*), highlighting the importance of evaluating such geolocation techniques in wireless settings.

The literature is rich with proximity verification and distance bounding protocols in wireless network environments, such as wireless sensor networks [27], [28] and Radio-Frequency Identifiers (RFIDs). Brands and Chaum [29] devised distance bounding protocols to prove an upper bound to the distance between a prover and a verifier using Radio Frequency (RF). Wagner *et al.* [30] proposed to use ultrasound instead to address shortcomings in RF-based protocols, such as processing delay sensitivity and accurate clock synchronization requirements. Capkun *et al.* [31] emphasized the importance of having at least three verifiers surrounding a prover to account for delay-adding attacks introduced by the prover or a third party attacker.

However, contrary to proximity verification and distance

bounding protocols in wireless networks, CPV [12] was designed to verify location assertions over the Internet, at the scale of tens or hundreds of miles. The former protocols focus on location verification within the wireless network itself (typically within a few inches [32]), while the latter's main focus is Internet clients [33] which could be using wired or wireless access networks. As such, CPV addresses different challenges, such as higher delay uncertainty due to the stochastic nature of Internet delays [34]. Therefore, assessing CPV's Internet scale verification abilities and significantly coarser granularity against extremely local protocols with substantially higher accuracy is not applicable.

Proximity verification and distance bounding protocols cannot easily scale to the Internet level because a considerable trustworthy infrastructure would be required. For example, if a Hulu[5] client was connected to the Internet through a wireless access network, then Hulu must trust a device in the client's wireless network to verify the client's geographic presence in the device's vicinity. Accordingly, a sufficient number of trusted wireless devices must be present to cover, at a high granularity, all the geographic regions of interest (e.g., the US in the case of Hulu). It is thus imperative to evaluate the behavior of CPV when Internet clients are connected through wireless access networks.

6 Concluding Remarks

We show that the impact of wireless networks on CPV depends fundamentally on the number of wireless devices in the vicinity of a CPV client. CPV is more likely to falsely accept adversaries close to the triangles' sides (e.g., <1,000 km) when there are wireless (legitimate) clients. Increasing the number of CPV iterations can be an effective way to deal with the negative effect of wireless delays on CPV. However, an excessively large number of iterations may reduce the practicality of CPV due to longer convergence times. Nonetheless, we found that when CPV is calibrated to be more tolerant to high delays between the client and the verifiers (i.e., at smaller values of τ), the rate for which the required number of iterations increases with k slows down. This highlights the importance of conducting the appropriate number of iterations, especially when CPV is verifying locations of wireless clients.

The results herein suggest that the impact of wireless networks on delay-based applications should be given more attention, e.g., most delay-based geolocation techniques in the literature are not evaluated with wireless networks [10], [34]. We hope this work encourages further evaluation to these applications considering wireless access networks.

Appendix A
Background on CPV

CPV [12] is a delay-based Internet location verification technique that accounts for classical geolocation evasion tactics [7], including IP-address hiding through the use of middleboxes (e.g., proxy servers, Virtual Private Networks, anonymizers) and delay manipulation [6].

5. Hulu (http://www.hulu.com/) is a video-on-demand website that provides geographically-restricted video streaming services.

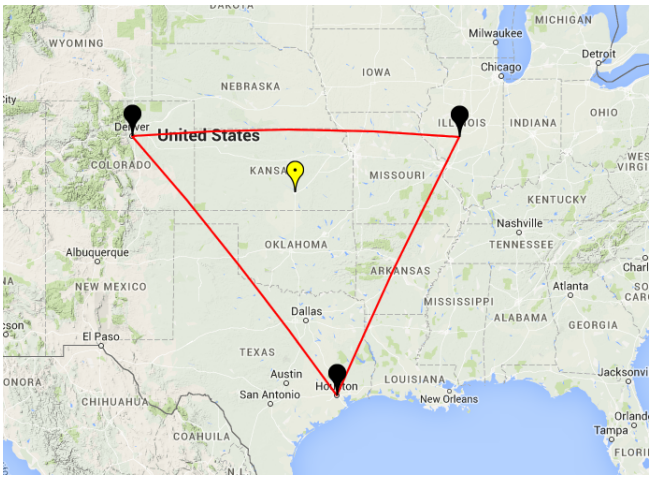


Fig. 10. An example triangle determined by three verifiers, and a legitimate client within. Map data: Google, INEGI, ORION-ME.

When an Internet client asserts its presence in a location, CPV selects three verifiers (e.g., cloud-based servers) geographically encapsulating the asserted location to corroborate that assertion (see Fig. 10). The verifiers exchange timestamps among themselves and the client to estimate one-way Internet delays on the application layer, leveraging cryptographic measures to protect delay estimates from being maliciously tampered, e.g., by the client. Delay estiamtion on the application layer (instead of the network-layer) allows CPV to detect the use of middleboxes and generic IP-hiding tactics [12], especially when CPV is combined with a Proof-of-Work mechanism [35]. The measured delays are then mapped into distances, which are then used to verify the client's true presence inside the triangle determined by the verifiers' locations. To mitigate factors that negatively affect the accuracy of delay-to-distance mapping, CPV uses heuristics such as employing multiple one-way delay estimation protocols and a per-region (rather than universal) mapping function. The authors of CPV [12] showed how the verifiers can obtain one-way delay estimates with high accuracy without requiring client clock synchronization [14].

Various factors affecting the correctness of CPV were tested using real-world extensive evaluations on PlanetLab, where nodes are connected to the Internet using wired access networks. It was found that the closer a legitimate client is from the sides of a triangle, the more likely it is for the client to be falsely rejected. The authors of CPV [12] have demonstrated the importance of appropriate verifier selection to avoid false rejections. Another crucial factor was the rate of Triangular Inequality Violations (TIVs) [36], where the authors of CPV have shown the ability of iterative delay measurements in dealing with TIVs. In summary, under classical conditions, when used against clients with wireline access networks, CPV can achieve false reject and false accept rates of 2% and 1% respectively [12].

Acknowledgments

The second author acknowledges support from the Natural Sciences and Engineering Research Council of Canada (NSERC) through a Discovery Grant. The third author

acknowledges funding from NSERC for both his Canada Research Chair in Authentication and Computer Security, and a Discovery Grant.

REFERENCES

[1] D. E. Denning and P. F. MacDoran, "Location-based authentication: Grounding cyberspace for better security," *Computer Fraud & Security*, vol. 1996, pp. 12–16, 1996.

[2] D. Berbecaru, "LRAP: A Location-Based Remote Client Authentication Protocol for Mobile Environments," in *Euromicro PDP*, 2011.

[3] H. Celebi and H. Arslan, "Utilization of location information in cognitive wireless networks," *IEEE Wireless Communications*, vol. 14, no. 4, pp. 6–13, 2007.

[4] J. Burnett, "Geographically Restricted Streaming Content and Evasion of Geolocation: the Applicability of the Copyright Anticircumvention Rules," *HeinOnline MTTLR*, vol. 19, p. 461, 2012.

[5] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.

[6] P. Gill, Y. Ganjali, B. Wong, and D. Lie, "Dude, where's that IP? Circumventing measurement-based IP geolocation," in *USENIX Security*, 2010.

[7] J. A. Muir and P. C. van Oorschot, "Internet geolocation: Evasion and counterevasion," *ACM Comput. Surv.*, vol. 42, no. 1, pp. 4:1–4:23, 2009.

[8] D. Hu and C.-L. Wang, "GPS-Based Location Extraction and Presence Management for Mobile Instant Messenger," *LNCS Embedded and Ubiquitous Computing*, vol. 4808, pp. 309–320, 2007.

[9] I. Polakis, S. Volanis, E. Athanasopoulos, and E. P. Markatos, "The Man Who Was There: Validating Check-ins in Location-based Services," in *ACM ACSAC*, 2013.

[10] S. Laki, P. Mátray, P. Hága, T. Sebók, I. Csabai, and G. Vattay, "Spotter: A Model Based Active Geolocation Service," in *IEEE INFOCOM*, 2011.

[11] Li *et al.*, "IP-Geolocation Mapping for Moderately Connected Internet Regions," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 2, pp. 381–391, 2013.

[12] A. M. Abdou, A. Matrawy, and P. C. van Oorschot, "CPV: Delay-based Location Verification for the Internet," *IEEE Trans. Dependable and Secure Computing, TDSC (to appear; accepted June 14, 2015)*.

[13] Chun *et al.*, "PlanetLab: An Overlay Testbed for Broad-coverage Services," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 3, pp. 3–12, 2003.

[14] A. M. Abdou, A. Matrawy, and P. C. van Oorschot, "Accurate One-Way Delay Estimation with Reduced Client-Trustworthiness," *IEEE Commun. Lett.*, vol. 19, no. 5, pp. 735–738, 2015.

[15] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 6th ed. Pearson Higher Ed, 2013, vol. 1.

[16] M. Carvalho and J. Garcia-Luna-Aceves, "Delay analysis of IEEE 802.11 in single-hop networks," in *IEEE Network Protocols*, 2003.

[17] P. Raptis, V. Vitsas, and K. Paparrizos, "Packet Delay Metrics for IEEE 802.11 Distributed Coordination Function," *Mobile Networks and Applications*, vol. 14, pp. 772–781, 2008.

[18] M. Maadani and S. A. Motamedi, "A simple and comprehensive saturation packet delay model for wireless industrial networks," *Wireless personal communications*, vol. 77, no. 1, pp. 365–381, 2014.

[19] IEEE, "EEE 802.11, The Working Group Setting the Standards for Wireless LANs," http://www.ieee802.org/11/.

[20] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *Selected Areas in Communications, IEEE Journal on*, vol. 18, pp. 535–547, 2000.

[21] G. Bianchi, L. Fratta, and M. Oliveri, "Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs," in *IEEE PIMRC*, 1996.

[22] N. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions*. John Wiley, 1994, vol. 1.

[23] J. F. Traub, *Iterative Methods for the Solution of Equations*. AMS Bookstore, 1982.

[24] I. Miller, J. E. Freund, and R. A. Johnson, *Probability and statistics for Engineers*. Prentice-Hall Englewood Cliffs, NJ, 1965, vol. 1110.

[25] M. Arif, S. Karunasekera, and S. Kulkarni, "GeoWeight: Internet Host Geolocation Based on a Probability Model for Latency Measurements," in *Australian Computer Society, Inc. ACSC*, 2010.

[26] Y. Wang, D. Burgener, M. Flores, A. Kuzmanovic, and C. Huang, "Towards street-level client-independent IP geolocation," in *USENIX NSDI*, 2011.

[27] J. Garcia-Alfaro, M. Barbeau, and E. Kranakis, "Secure geolocalization of wireless sensor nodes in the presence of misbehaving anchor nodes," *Annals of Telecommunications*, vol. 66, no. 9-10, pp. 535–552, 2011.

[28] C.-Y. Chow, M. F. Mokbel, and T. He, "A privacy-preserving location monitoring system for wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 94–107, 2011.

[29] S. Brands and D. Chaum, "Distance-Bounding Protocols," in *Advances in Cryptology–EUROCRYPT'93*. Springer, 1994, pp. 344–359.

[30] N. Sastry, U. Shankar, and D. Wagner, "Secure Verification of Location Claims," in *ACM WiSec*, 2003.

[31] S. Capkun and J.-P. Hubaux, "Secure positioning of wireless devices with application to sensor networks," in *IEEE INFOCOM*, 2005.

[32] K. Finkenzeller and D. Müller, *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. Wiley-Blackwell, 2010.

[33] A. M. Abdou, A. Matrawy, and P. C. van Oorschot, "Location Verification on the Internet: Towards Enforcing Location-aware Access Policies Over Internet Clients," in *IEEE CNS*, 2014.

[34] Z. Dong, R. D. Perera, R. Chandramouli, and K. Subbalakshmi, "Network measurement based modeling and optimization for IP geolocation," *Elsevier Computer Networks*, vol. 56, no. 1, pp. 85–98, 2012.

[35] A. M. Abdou, A. Matrawy, and P. C. van Oorschot, "Taxing the Queue: Hindering Middleboxes from Unauthorized Large-Scale Traffic Relaying," *IEEE Commun. Lett.*, vol. 19, 2015.

[36] C. Lumezanu, R. Baden, N. Spring, and B. Bhattacharjee, "Triangle inequality and routing policy violations in the Internet," in *Springer PAM*, 2009.

AbdelRahman Abdou is a Post-Doctoral Fellow in the School of Computer Science at Carleton University. He received his PhD in 2015 from Carleton University. His research interests include location-aware security, Future Internet Architectures (FIAs), and using Internet measurements to solve problems related to Internet security.

Ashraf Matrawy is an Associate Professor of the School of Information Technology at Carleton University. He is a senior member of the IEEE and serves on the editorial board of the IEEE Communications Surveys and Tutorials journal. He has served as a technical program committee member of IEEE CNS, IEEE ICC, IEEE Globecom, IEEE LCN, and IEEE/ACM CCGRID. He is also a Network co-Investigator of Smart Cybersecurity Network (SERENE-RISC). His research interests include reliable and secure computer networking, software defined networking and cloud computing.

Paul C. van Oorschot is a Professor of Computer Science at Carleton University, and the Canada Research Chair in Authentication and Computer Security. He was the program chair of USENIX Security 2008, NDSS 2001-2002, NSPW 2014-2015, a co-author of the Handbook of Applied Cryptography and a past associate editor of IEEE TDSC, IEEE TIFS, and ACM TISSEC. His research interests include authentication and Internet security.