Accurate One-Way Delay Estimation with Reduced Client-Trustworthiness

AbdelRahman Abdou, Ashraf Matrawy, and Paul C. van Oorschot
Carleton University, Ottawa, Canada

***Abstract*—The requirement for accurate one-way delay (OWD) estimation lead to the recent introduction of an algorithm enabling a server to estimate OWDs between itself and a client by cooperating with two other servers, requiring neither client-clock synchronization nor client trustworthiness in reporting one-way delays. We evaluate the algorithm by deriving the probability distribution of its absolute error, and compare its accuracy with the well-known round-trip halving algorithm. While neither algorithm requires client-trustworthiness nor client clock synchronization, the analysis shows that the new algorithm is more accurate in many situations.**

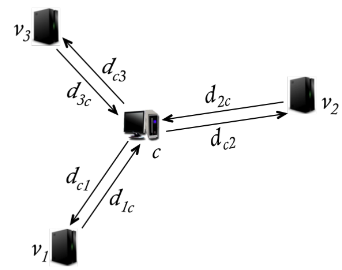

Fig. 1. OWDs between client c and verifiers v_1 (server), v_2 and v_3 [5].

I. Introduction

MANY Internet applications can benefit from accurate one-way delay (OWD) estimation mechanisms. A server and a client[1] can cooperate to estimate OWDs between themselves [1]. Cooperation typically involves clock synchronization and exchanging timestamps, e.g., One-way Active Measurement Protocol (OWAMP) [2]. Such approaches thus require considerable client trustworthiness.

Because round-trip times (RTTs) are easier to estimate than OWDs, they are often used instead [3]. Half the RTT is sometimes taken as a OWD estimate; we call this the *average* (av) algorithm. Nonetheless, the asymmetric nature of Internet routes [4] substantially affects the av algorithm's accuracy.

Client Presence Verification (CPV) [5] was proposed as a delay-based mechanism that verifies clients' geographic locations over the Internet. It introduced a new OWD-estimation algorithm, *minimum pairs* (mp), designed to be more accurate than av, yet requires less client cooperation (hence trustworthiness) than OWAMP-like tools. In mp, the server cooperates with two trusted verifiers (e.g., cloud-based servers) instead of the client. The accuracy of mp was informally discussed [5].

We formally analyze the accuracy of mp and av given the delay characteristics of the underlying network, enabling an informed choice between alternatives. Contributions:

- Deriving the probability mass function (PMF) of the absolute error for the mp (proposed in previous literature [5]) and the av algorithms as a function of the delay distribution between the client and the server/verifiers.
- Using the derived probability model to compare the accuracy of both algorithms assuming Poisson delay distribution with various representative means. This example comparison can now be drawn since the derived models allow general determination of the more accurate algorithm given the probability distribution of delays.

II. Review of the Minimum Pairs Algorithm

The mp algorithm estimates the smaller of the forward and reverse OWDs between the client and the server at current network conditions.[2] The server cooperates with two other trusted verifiers; for simplicity, we refer to the three parties as verifiers v_1, v_2 and v_3. Notation of OWDs between the three verifiers with the client is given in Fig. 1. The server is v_1, so the algorithm should estimate the smaller between d_{1c} and d_{c1}. Each verifier must possess a public-private key pair, and be aware of the public keys of the other two verifiers.

Using the established connection with the client,[3] v_1 notifies the client of the IP addresses of v_2 and v_3, the client connects to both verifiers and Algorithm 1 (below) starts. Notation:

- $S_a(m)$ denotes message m digitally signed by entity a.
- $A \xrightarrow{m} B$ means A sends message m to B.
- t_a is the most recent timestamp according to a's clock.
- d^+_{ij} corresponds to $d_{ic} + d_{cj}$ (see Fig. 1).
- β_i is an estimate to the smaller of d_{ic} and d_{ci}.

In lines 13 through 16, v_1 discards the larger sums between $d_{ic} + d_{cj}$ and $d_{jc} + d_{ci}$ for all $1 \le i, j \le 3$ and $i \ne j$, and uses the remaining sums to estimate the smaller OWD. This exclusion helps in reducing the effect of delay spikes happening in one direction but not the other. Note that, similar to av, the mp algorithm does not indicate the direction of the shorter delay.

[1] We use the server/client terminology to discriminate between the party measuring delays (server) and the one the delays are measured to/from (client).

[2] Although not noted by the authors of mp [5], the larger OWD can also be estimated based on the difference between the smaller and the RTT.

[3] In a delay-dependent application, the server and the client are typically assumed to have an established connection.

Algorithm 1: The mp algorithm [5]. See notation inline.

Input: The set of the three verifiers, V (see Fig. 1).
Output: An estimate to the smaller one-way delay between client c and verifier v_1.

1 **begin**
2   **foreach** v_i **in** V **do**
3   v_i retrieves its current system time $b := t_i$
4     $v_i \xrightarrow{b,S_i(b)} c$
5   **foreach** v_j **in** V **do**
6       $c \xrightarrow{b,S_i(b)} v_j$
7   v_j records the message-receiving time $r := t_j$
8   v_j validates $S_i(b)$
9   **if** *invalid signature* **then**
10   Abort "possible client cheating attempt"
11       $d^+_{ij} := r - b$
12   **if** $v_j \neq v_1$ **then** $v_j \xrightarrow{d^+_{ij}} v_1$
13   v_1 solves simultaneously for $\beta_1, \beta_2, \beta_3$:
14     $\beta_1 + \beta_2 = min(d^+_{12}\ ,\ d^+_{21})$
15     $\beta_2 + \beta_3 = min(d^+_{23}\ ,\ d^+_{32})$
16     $\beta_3 + \beta_1 = min(d^+_{31}\ ,\ d^+_{13})$
17   **return** β_1

III. Analyzing the average algorithm (av)

a) Notation: The *absolute error* is the absolute difference between the smaller of the forward and reverse OWDs and the OWD estimated by the algorithm. Let $f_x(d)$ be the PMF of the delay of edge d, for each of the six edges in Fig. 1.

b) Absolute error: The *av* algorithm estimates the smaller OWD between v_1 (the server) and c as:

$$t^{av} = \frac{\text{RTT}}{2} = \frac{d_{1c} + d_{c1}}{2} \tag{1}$$

The absolute error of the *av* algorithm is:

$$\epsilon^{av} = |t^{av} - min(d_{1c}, d_{c1})|$$

The magnitude of the error thus depends on the difference between d_{1c} and d_{c1}. Table I lists the three cases. Denoting by ϵ^{av}_i the error in Case i, then:

$$\epsilon^{av}_1 = \left| \frac{d_{1c} + d_{c1}}{2} - d_{1c} \right| = \frac{d_{c1} - d_{1c}}{2}$$

We can drop the *"absolute"* sign ($||$) because in Case 1, $d_{1c} < d_{c1}$. The error for the remaining two cases is given in Table I.

c) PMF of error: The PMF of ϵ^{av}_i depends on the probability of occurrence of Case i. Thus, for all $x \geq 0$:

$$\begin{aligned} P\{\epsilon^{av} = x\} &= \sum_{i=1}^{3} P\{\text{Case } i\} \cdot P\{\epsilon^{av}_i = x \ \ | \ \ \text{Case } i\} \\ &= \sum_{i=1}^{3} P\{\epsilon^{av}_i = x \ \ , \ \ \text{Case } i\} \end{aligned} \tag{2}$$

TABLE I. Cases relating d_{1c} with d_{c1}, the calculated delay (t^{av}) in each case, and the error (ϵ^{av}) of the *av* algorithm.

Case (i)	Condition d_{1c} [relation] d_{c1}	t^{av}_i	ϵ^{av}_i
1	$<$	$(d_{1c} + d_{c1})/2$	$(d_{c1} - d_{1c})/2$
2	$=$	$(d_{1c} + d_{c1})/2$	0
3	$>$	$(d_{1c} + d_{c1})/2$	$(d_{1c} - d_{c1})/2$

where the *"comma"* indicates the intersection of the two events. Expanding the term at $i = 1$ yields:

$$\begin{aligned} &P\{\epsilon^{av}_1 = x \ \ , \ \ \text{Case } 1\} \\ &\quad = P\{(d_{c1} - d_{1c})/2 = x \ \ , \ \ d_{1c} < d_{c1}\} \\ &\quad = P\{d_{c1} = 2x + d_{1c} \ \ , \ \ d_{1c} < 2x + d_{1c}\} \\ &\quad = P\{d_{c1} = 2x + d_{1c} \ \ , \ \ x > 0\} \\ &\quad = \left(\sum_{i=0}^{\infty} P\{d_{1c} = i\} \cdot P\{d_{c1} = 2x + i\} \right) \cdot P\{x > 0\} \\ &\quad = \begin{cases} \displaystyle\sum_{i=0}^{\infty} f_i(d_{1c}) \cdot f_{2x+i}(d_{c1}), & x > 0 \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{3}$$

Since $\epsilon^{av}_2 = 0$ (see Table I), therefore,

$$\begin{aligned} P\{\epsilon^{av}_2 = x \ \ , \ \ \text{Case } 2\} &= P\{x = 0 \ \ , \ \ d_{1c} = d_{c1}\} \\ &= \begin{cases} P\{d_{1c} = d_{c1}\}, & x = 0 \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

The term for $i = 3$ in (2), $P\{\epsilon^{av}_3 = x \ \ , \ \ \text{Case } 3\}$, can be expanded analogous to Case 1. We thus rewrite (2) as:

$$\begin{aligned} &P\{\epsilon^{av} = x\} \\ &= \begin{cases} P\{d_{1c} = d_{c1}\}, & x = 0 \\ P\{\epsilon^{av}_1 = x \text{ , Case } 1\} + P\{\epsilon^{av}_3 = x \text{ , Case } 3\}, & x > 0 \end{cases} \end{aligned}$$

where:

$$P\{d_{1c} = d_{c1}\} = \sum_{i=0}^{\infty} f_i(d_{1c}) \cdot f_i(d_{c1})$$

IV. Analyzing the minimum pairs algorithm (mp)

a) Absolute error: In Algorithm 1, lines 14 to 16 define three simultaneous equations that estimate the smaller OWD (t^{mp}). Although the *mp* algorithm does not enable the verifiers to calculate d^-_{ii} for all $i \in \{1, 2, 3\}$,[4] it enables them to sort these differences. For example, assume in line 14 that $d_{2c} + d_{c1} \leq d_{1c} + d_{c2}$. Rearranging yields $d^-_{22} \leq d^-_{11}$. Also assuming in line 15 that $d_{3c} + d_{c2} < d_{2c} + d_{c3}$ (equivalent to $d^-_{33} < d^-_{22}$), the verifiers can deduce that $d^-_{33} < d^-_{22} \leq d^-_{11}$.

The order of d^-_{11}, d^-_{22} and d^-_{33} identifies the cases in Table II; possible outcomes of the $min()$ function in lines 14 to 16 are indicated at the header of the *"Conditions"* column, with their

[4] d^+_{ij} denotes $d_{ic} + d_{cj}$; likewise, d^-_{ij} denotes $d_{ic} - d_{cj}$.

TABLE II. CASES RELATING d^+_{ij} WITH d^+_{ji}, THE CALCULATED DELAY IN EACH CASE (t^{mp}_i), AND THE ABSOLUTE ERROR (ϵ^{mp}) OF THE *mp* ALGORITHM. IN EACH CASE, A CIRCLED CONDITION IS IMPLIED BY THE OTHER TWO.

Case (i)	Conditions			Order	t^{mp}_i	$\epsilon^{mp}_{i,j}$	
	d^+_{31} [relation] d^+_{13}	d^+_{21} [relation] d^+_{12}	d^+_{32} [relation] d^+_{23}			$d_{1c} \leq d_{c1}$	$d_{1c} > d_{c1}$
1	$<$	$\leq$	$<$	$d^-_{33} < d^-_{22} \leq d^-_{11}$	$d_{c1} + d^-_{22}/2$	$\left\lvert d^-_{22}/2 - d^-_{11}\right\rvert$	$\left\lvert d^-_{22}/2\right\rvert$
2	$<$	$<$	$\geq$	$d^-_{22} \leq d^-_{33} < d^-_{11}$	$d_{c1} + d^-_{33}/2$	$\left\lvert d^-_{33}/2 - d^-_{11}\right\rvert$	$\left\lvert d^-_{33}/2\right\rvert$
3	$\leq$	$>$	$<$	$d^-_{33} \leq d^-_{11} < d^-_{22}$	$d^+_{11}/2$	$-d^-_{11}/2$	$d^-_{11}/2$
4	$=$	$=$	$=$	All three are equal	$d^+_{11}/2$	$-d^-_{11}/2$	$d^-_{11}/2$
5	$\geq$	$<$	$>$	$d^-_{22} < d^-_{11} \leq d^-_{33}$	$d^+_{11}/2$	$-d^-_{11}/2$	$d^-_{11}/2$
6	$>$	$>$	$\leq$	$d^-_{11} < d^-_{33} \leq d^-_{22}$	$d_{1c} - d^-_{33}/2$	$\left\lvert -d^-_{33}/2\right\rvert$	$\left\lvert d^-_{11} - d^-_{33}/2\right\rvert$
7	$>$	$\geq$	$>$	$d^-_{11} \leq d^-_{22} < d^-_{33}$	$d_{1c} - d^-_{22}/2$	$\left\lvert -d^-_{22}/2\right\rvert$	$\left\lvert d^-_{11} - d^-_{22}/2\right\rvert$
	d^-_{33} [relation] d^-_{11}	d^-_{22} [relation] d^-_{11}	d^-_{33} [relation] d^-_{22}				
	Rearranged Conditions						

rearrangements indicated at the bottom. Two conditions imply the third; the implied condition is circled in Table II.

The smaller between d_{1c} and d_{c1} is indicated by the t^{mp}_i column in Table II. In Case 1 for example, where $d^+_{31} < d^+_{13}$, $d^+_{21} \leq d^+_{12}$, and $d^+_{32} < d^+_{23}$, the simultaneous equations of lines 14 to 16 will be $\beta_1 + \beta_2 = d^+_{21}$, $\beta_2 + \beta_3 = d^+_{32}$, and $\beta_3 + \beta_1 = d^+_{31}$. In Algorithm 1, β_1 is returned as the estimate to the smaller between d_{1c} and d_{c1}, which evaluates to:

$$\begin{aligned} t^{mp}_1 = \beta_1 &= \frac{d^+_{21} + d^+_{31} - d^+_{32}}{2} \\ &= \frac{d_{2c} + d_{c1} + d_{3c} + d_{c1} - (d_{3c} + d_{c2})}{2} \\ &= \frac{d_{2c} - d_{c2} + 2d_{c1}}{2} = d_{c1} + \frac{d^-_{22}}{2} \end{aligned}$$

Similarly, t^{mp}_i can be calculated for the remaining cases.

The returned OWD estimate (t^{mp}) can indicate whether there were large delay asymmetries between each verifier and the client. For example, if $t^{mp} < 0$, then the difference between the forward and reverse delays of some links between the client and the verifiers is relatively large.

b) Comparison between t^{mp} and t^{av}: As is now shown, in none of the seven cases will the mp algorithm return a larger estimate to the smaller OWD than that of the av algorithm; that is, the inequality $t^{mp}_i \leq t^{av}$ holds for all $i \in \{1..7\}$. In Case 1, we have (Table II):

$$t^{mp}_1 = d_{c1} + \frac{d^-_{22}}{2} \tag{4}$$

Since $d^-_{22} \leq d^-_{11}$ in this case (second rearranged condition, bottom of the *"Conditions"* column in Table II), therefore:

$$t^{mp}_1 \leq d_{c1} + \frac{d^-_{11}}{2}$$

Simplifying yields

$$t^{mp}_1 \leq \frac{d_{1c} + d_{c1}}{2} = t^{av} \qquad \text{from (1)}$$

Analogous analysis applies to Cases 2, 6 and 7, which we omit for space reasons. The equation $t^{mp}_i = t^{av}$ already holds for $i \in \{3, 4, 5\}$ (see Table II). Thus, the mp algorithm never returns an estimate, to the smaller between the forward and reverse OWDs, that is larger than that of the av algorithm.

c) PMF of error: The PMF of error depends on the probability of occurrence of each case in Table II, and the probabilities of $d_{1c} \leq d_{c1}$ and $d_{1c} > d_{c1}$ in each case. We index those two additional conditions using the variable $j \in \{1, 2\}$ respectively. For example, to calculate the error in Case 1 given additional condition 2 (which is $d_{1c} > d_{c1}$):

$$\epsilon^{mp}_{1,2} = |t^{mp}_1 - min(d_{1c}, d_{c1})| = \left| d_{c1} + \frac{d^-_{22}}{2} - d_{c1} \right| = \left| \frac{d^-_{22}}{2} \right|$$

The probability that the error is equal to x is the probability that any of the expressions listed under the $\epsilon^{mp}_{i,j}$ column in Table II evaluates to x, for all $x \geq 0$. The PMF of the absolute error can, thus, be expressed as:

$$\begin{aligned} P\{\epsilon^{mp} = |x|\} &= \sum_{i=1}^{7} \sum_{j=1}^{2} P\{X_{i,j}\} \cdot P\{\epsilon^{mp}_{i,j} = |x| \quad | \quad X_{i,j}\} \\ &= \sum_{i=1}^{7} \sum_{j=1}^{2} P\{\epsilon^{mp}_{i,j} = |x| \quad , \quad X_{i,j}\} \end{aligned} \tag{5}$$

where $X_{i,j}$ is the intersection of all three conditions under the *"Conditions"* column of Case i with additional condition j. Because the error is the absolute difference, then:

TABLE III. MEANS OF THE POISSON DISTRIBUTIONS OF THE DELAYS FOR EACH EDGE IN FIG. 1, AND THEIR CORRESPONDING CHART IN FIG. 2.

Scenario		Mean λ (ms)					
		d_{1c}	d_{c1}	d_{2c}	d_{c2}	d_{3c}	d_{c3}
Fig. 2	(a)	30	30	30	30	30	30
	(b)	30	7	8	25	5	5
	(c)	2	20	5	50	7	80
	(d)	35	5	45	70	2	15
	(e)	10	10	30	12	30	60
	(f)	10	10	30	3	20	5

$$P\{\epsilon_{i,j}^{mp} = |x| \;\; , \;\; X_{i,j}\}$$
$$= \begin{cases} P\{\epsilon_{i,j}^{mp} = 0 \;, \; X_{i,j}\}, & x = 0 \\ P\{\epsilon_{i,j}^{mp} = x \;, \; X_{i,j}\} + P\{\epsilon_{i,j}^{mp} = -x \;, \; X_{i,j}\}, & \text{otherwise} \end{cases} \tag{6}$$

V. EXAMPLES OF ACCURACY COMPARISON

It has been established that Internet delays follow a Gamma distribution with varying parametrization [6], [7]. We model the OWDs of the six edges of Fig. 1 as independent and discrete random variables that follow Poisson distributions,[5] and take on integer values (e.g., delays in milliseconds). Poisson is used because it is a discrete distribution that is a special case of Gamma. Table III lists the distribution means in six example scenarios. The scenarios were chosen to analyze the effect of delay asymmetry between the client and the verifiers. Figure 2 plots the Cumulative Distribution Functions (CDFs) of the absolute errors for each scenario in Table III, using (2) and (5) for the av and the mp algorithms respectively.

Scenario (a) (Table III) addresses delay symmetry in all six edges. Figure 2(a) shows that mp is more accurate than av in this scenario, with a 54% chance of producing an absolute error <1.5 ms, versus 35% for av.

Scenario (b) addresses the effect of delay symmetry between the client and one verifier. In this scenario, we deduce that mp will operate in Case 2 most of the time (from the *"Order"* column in Table II), and thus $\epsilon^{mp} = \epsilon_{2,2}^{mp}$ as it is highly probable that $d_{1c} > d_{c1}$. Because d_{3c} and d_{c3} have equal means (5 ms), the error $\epsilon_{2,2}^{mp} = |d_{33}^{-}/2|$ becomes relatively small, as shown in Fig. 2(b). The mp algorithm has a 90% chance of resulting in <2.5 ms absolute error, versus 0.1% for the av, making it significantly more accurate in this scenario.

Scenarios (c) and (d) explore delay asymmetry in all six edges. Despite the huge asymmetries in (c), mp has a ∼25% chance to result in <2.5 ms absolute error, versus ∼0.2% for av. The smaller delay variations of scenario (d), compared to (c), caused mp to be substantially more accurate (Fig. 2(d)).

Scenarios (e) and (f) analyze the effect of delay symmetry between d_{1c} and d_{c1}, and asymmetry in the other two links. In Fig. 2(e), where the two graph lines coincide, the accuracy of mp is similar to that of av because, with higher probability, mp operates in Case 3 of Table II (the resulting OWD-estimates are similar to av). In (f), delay asymmetry between the client and $\{v_2, v_3\}$ mislead mp, but do not affect the average of d_{1c} and d_{c1}. Because d_{1c} and d_{c1} are highly symmetric (see Table III), av is more accurate.

[5]Note that this is not the packet arrival times.

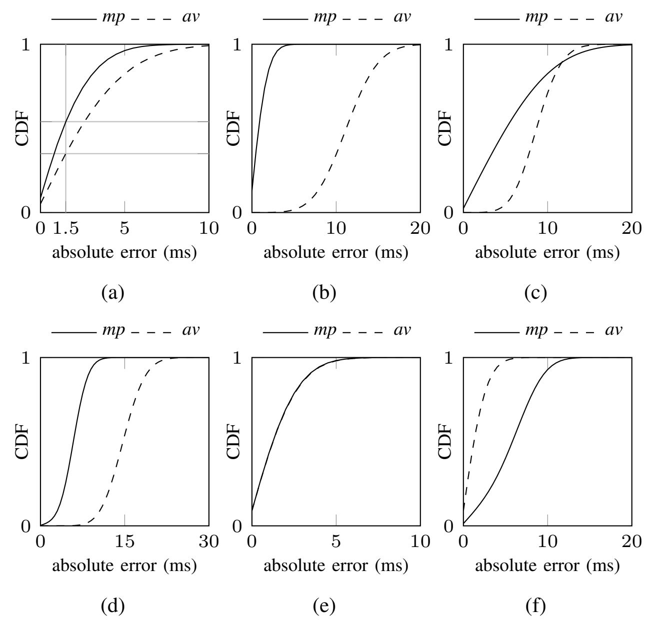

Fig. 2. Absolute errors between the estimated and the actual OWD, assuming Poisson delay distributions (see Table III for means) for the edges in Fig. 1.

VI. CONCLUSION

Errors due to imperfect clock synchronization among the verifiers can be mitigated as shown in the literature [3], and are thus not considered by the mp's PMF derived herein.

The analysis herein establishes that the mp algorithm [5] is in many cases more accurate in estimating OWDs than the commonly-used av algorithm. This is achieved with the added bonus of the mp's reduced client-cooperation requirements, making it suitable for adversarial environments, but comes at the cost of requiring extra infrastructure (the verifiers).

We highlight that the degree of delay asymmetry between the verifiers and the client is a key element affecting the accuracy of both algorithms. The PMFs derived herein are thus useful to an application deciding between both algorithms. This follows from the properties of the PMFs derived herein: (1) they allow determination of which algorithm is more accurate given the delay environment, and (2) they are generic—they evaluate the probability mass of error given any discrete delay distribution (Poisson was used herein).

Acknowledgments. This work was supported by Canada's NSERC through the Canada Research Chairs program (third author) and Discovery Grants (second and third authors).

REFERENCES

[1] Shin *et al.*, "Clock Synchronization for One-Way Delay Measurement: A Survey," in *Advanced Communication and Networking*. Springer, 2011, vol. 199, pp. 1–10.

[2] Shalunov *et al.*, "A One-way Active Measurement Protocol (OWAMP)," RFC 4656, 2006.

[3] A. Vakili and J. Gregoire, "Accurate One-Way Delay Estimation: Limitations and Improvements," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 9, pp. 2428–2435, 2012.

[4] Pathak *et al.*, “A Measurement Study of Internet Delay Asymmetry,” in *PAM*. Springer, 2008.

[5] 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*, Oct. 2014.

[6] A. Mukherjee, “On the Dynamics and Significance of Low Frequency Components of Internet Load,” *Internetworking: Research and Experience*, vol. 5, pp. 163–205, 1992.

[7] Bovy *et al.*, “Analysis of end-to-end delay measurements in Internet,” in *PAM*. Springer, 2002.