Detection of Network Flow Timestamp Reliability

Martin Žádnik, Erik Šabik, Václav Bartoš

Detection of network flow timestamp reliability

Martin Zadnik, Erik Sabik, and Vaclav Bartos

CESNET, a. l. e.,
Zikova 4, 160 00 Prague, Czech Republic
zadnik, sabik, bartos@cesnet.cz

Abstract. Network flow measurement and analysis are important parts of network management and security. Flow data analysis is a challenging task which is often rendered harder by pitfalls in a monitoring pipeline. In this paper we focus on timestamps since many analysis procedures utilize timestamps to reveal various characteristics of network traffic. Unfortunately, the timestamps are not always that reliable as it may seem. We propose an algorithm to estimate the percentage of correctly assigned timestamps to flow records with respect to the sequence of a request and a response flow. We simulate various timestamp failures and we evaluate the failures using the proposed algorithm. We demonstrate the usage of the algorithm in the use case of bidirectional flow orientation.

1 Introduction

Network administrators as well as service providers need to measure and analyse network traffic for maintenance, security and planning [1]. One popular approach is based on network flows. The flows are measured at observation points (e.g. routers, probes) and the flow records are exported to collectors for storage and analysis. Flow monitoring enables various security and management applications to reveal information about the state of network and its network traffic as well as the state of services and the behavior of connected machines (e.g. to detect infected machines).

Flow records contain timestamps besides other statistics. The timestamps usually express start and end of a flow. The flow start timestamp captures a significant event from the analysis perspective. For example, the start timestamps offer a possibility to identify initiators of communications (i.e. to establish bidirectional flow orientation [2]), distinguish and profile hosts [3], measure TCP response time [4] and order flows according to their arrival at the observation point.

The utilization of start timestamps may improve the data analysis but only if the start timestamps are reliable. This is often not the case based on our experience from practical deployments as well as based on the cases studied in the literature. For example, in [5] the authors state: "Furthermore, the flow timestamps have proved to be sometimes unreliable and more often, the request and reply flows have identical timestamps due to the granularity of the timestamps." On the other hand, if the monitoring pipeline provides reliable timestamps the

subsequent analysis methods may utilize this knowledge to improve on their results. To this end, we propose an algorithm to estimate reliability of start timestamps in the flow records belonging to a particular observation point. The algorithm assumes that a flow generated in reaction (response flow) cannot precede the initiating flow (request flow). The algorithm selects a certain subset of flow records and compares their start timestamps with another heuristic utilizing port numbers. The correlation provides an estimate on the reliability of the timestamps. The algorithm consumes only little amount of resources not to slow down any subsequent analysis.

We evaluate the algorithm on real packet and flow traces. To assess a baseline we utilize traces with reliable timestamps. Subsequently the algorithm is evaluated on traces containing originally as well as artificially generated timestamp failures.

In order to show the contribution of utilizing timestamps (if reliable) we propose a biflow orientation algorithm utilizing timestamp reliability estimate to modify its function when the timestamps are not reliable.

The rest of the paper is organized as follows. Related work on timestamp failures and timestamp utilization is discussed in Section 2. Section 3 and 4 describe the proposed timestamp reliability estimate algorithm and biflow orientation algorithm respectively. Section 5.2 analyzes several data sets and evaluates the algorithm on a problem of biflow orientation. The paper is concluded with summary and future work in Section 6.

2 Related work

2.1 Timestamp failures

Failures and errors in the monitoring pipeline are not an exception as documented in several works. In [6], [7] the authors investigate various flow measurement tools and look for artifacts caused by design decisions or implementation constraints. Besides other artifacts the authors found out that a certain flow exporter reports incorrect start times in a case a flow is exported due to an active timeout. Moreover, they analyze the effects of *flow learn failures* on flow data. The authors state in [7]: "Our experiments have shown that the first packets of flows are more likely to be subject to flow learn failures, because subsequent packets of accounted flows are matched until the records are expired". As a result some start timestamps need not necessarily correspond to the first packets in the flows. The timing artifacts may also be caused by the poor resolution of timestamps and this issue is analyzed closer in [8]. Also an underlying platform/system hosting the measurement process may introduce some timing issues, e.g. so called interrupt coalescence [9].

The timing error may appear also due to a design option. The work [10] characterizes timing errors in flow data caused by collection and export via NetFlow v9 [11]. The work identifies three sources of errors in the measurement methodology. These errors may lead to the timing bias in the order of a second.

Other causes of timing errors are not that well documented. The following list of potential timestamp issues is definitely not complete but summarizes our experience from real deployment.

Buffers. A timestamp must be assigned to each packet upon its arrival at the physical interface. Otherwise a packet may be subject to buffering resulting in an arbitrary delay in hardware (network card) and software (host memory) buffers. Since the buffers are usually dedicated per each interface some packets may or may not be buffered. As a result the order of packets at physical interfaces need not necessarily conform to the order of timestamps assigned by software. High-speed packet capture solutions usually utilize large buffers to increase transfer efficiency and to sustain short overloads. This results in an even larger unpredictability of the correct timestamp assignment in software.

Packet drop/sampling. A metering process may drop packets if overloaded or may sample packets deliberately. In both cases some first packets are lost resulting in a mismatch between the true start of a flow and the start timestamp stored in a flow record.

Timestamp representation. An inconsistent interpretation of timestamp format may result in a timestamp error as well. For example, a 64-bit timestamp may represent seconds (upper 32 bits) and a remainder (lower 32 bits). In the first case the remainder is represented as a number of 233 picosends intervals to fully utilize all 32 bits whereas in the second case the remainder is represented as a number of nanoseconds, i.e. utilizing only 30-bits. An error of up to 750 ms may appear if a timestamp stored in the first representation (e.g. by hardware) is interpreted as the second (e.g. by a flow measurement tool).

Deduplication. In certain metering infrastructures a flow may be measured multiple times since it traverses multiple observation points. Deduplication process merges corresponding flows into a single record. Based on the configured rules the order of flows (their start timestamps) may change during the deduplication process.

The low reliability of timestamps is also demonstrated on a popular collector NfDump [12] since the collector implements biflow orientation based on port numbers only.

2.2 Timestamp analysis

The time characteristics of flow exporters are analyzed in [8] from the perspective of one way delay (OWD) measurement. An offline algorithm is proposed to derive an exporter profile based on flow data. The profile captures clock skew, clock offset, resolution, bias and accuracy of timestamps. The profiles are inferred by correlation of records belonging to traversing flows (flows seen on at least two exporters). Therefore the algorithm needs at least two exporters (a reasonable assumption for OWD measurement) but the algorithm does not account for timestamp errors described in Section 2.1. The algorithm infers ordering of flows (request-response) based on timestamps and it is therefore susceptible to errors described in Section 2.1. The profiling algorithm would benefit from the

knowledge of timestamp reliability derived by our algorithm. Therefore we consider our algorithm to be orthogonal since their algorithm focuses on different error types, utilize different heuristics and report different results.

NfSight tool [5] utilizes Bayesian classifier to identify clients and servers. The classifier combines timestamp, port and IP address heuristics. The classifier is trained and evaluated using an annotated data set, i.e. server and clients are known a priori. The evaluation shows that the start timestamps in their data set are heavily biased. The ability of start timestamps to correctly identify biflow orientation is very low (25%) if the difference of the timestamp is lower than 1 second (see [5] page 4, Fig. 2). According to the evaluation 95% accuracy is reached only in cases where the difference is more than 5 seconds.

Clearly, each data set (exporter) may produce flow records of various reliability and therefore on a different data set the training phase would construct a different classifier. However, it is problematic in a real deployment to train the classifier due to various limitations, e.g. it is not always possible to capture reliable raw packet dump to apply annotation mechanisms. Therefore, we consider our work to be complementary to [5] since our algorithm does not require any training though it may reach lower precision when utilized in the use case of the biflow orientation. Also the output of our biflow orientation algorithm may serve as an input of the classifier if the training phase is feasible.

Minarik et al. [3] proposed an extension of host profiling with bidirectional flows. A request flow is identified by its start timestamp, i.e. the flow with an earlier start timestamp is a request. If the timestamps are not reliable the port number heuristic is utilized (a higher source port and a lower destination port is considered to appear in request). Our work extends the idea further by an algorithm to detect timestamp failure automatically. Moreover, we propose a decision tree to improve biflow orientation by combining timestamp and port heuristics.

3 Timestamp reliability algorithm

The goal of the algorithm is to estimate the percentage of flow records with a valid start timestamp. For the purpose of our work we consider start timestamps to be valid if the timestamps of a request flow precedes the timestamp of a flow generated in response. In our work, we utilize term flow to denote a set of packets with the same 5-tuple (IP addresses, port numbers and protocol number). The algorithm sets out its estimate according to the correlation of start timestamps with a request/response heuristic based on port numbers. Please note that our algorithm is able to estimate bad ordering of the timestamps but not their accuracy.

The algorithm starts by selecting flow records that are relevant from the perspective of start timestamps or port numbers. Let A denotes a set of all flow records, $T \subset A$ denotes a subset of flow records that is possible to evaluate based on start timestamps, $P \subset A$ denotes a subset of flow records that is possible to evaluate based on port number heuristic. $T_{req,resp} \subset T$ is a subset of flow records

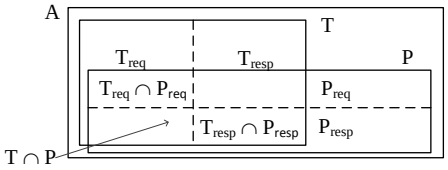

Fig. 1. Network flows split into sets according to timestamp and port number heuristic

denoted by timestamps as requests and responses respectively. $P_{req,resp} \subset P$ is a subset of flow records denoted by port numbers as requests and responses respectively. The sets are depicted in Fig. 1.

The subset T contains only flow records meeting following conditions.

1. The flow record corresponds to a flow in reverse direction, i.e. there are two flows forming a bidirectional flow in the set A. If a flow is only unidirectional it is not possible to distinguish request or response flow based on its start timestamp. A flow may be unidirectional due to various reasons such as its reverse flow is not observed (e.g. asymmetric routing), reverse flow is discarded (e.g. flow sampling), server is not responding and many others.
2. The bidirectional flow record accounts for a TCP connection.
3. The record contains TCP SYN flags for both directions. In that case the record is considered to account for the first packets of each flow since a flow may be reported as a sequence of flow records by the observation point due to, e.g., inactive and active timeouts, full flow cache or replacement policy.
4. The start timestamps in the record must differ otherwise it is not possible to decide the flow ordering.

The subset P contains flow records that is possible to validate by the port number heuristic. The utilization of port numbers rules out flow records with protocols other than TCP or UDP. The heuristic compares destination port number (destinationTransportPort) with source port number (sourceTransportPort). The heuristic classifies only flow records when one of the port numbers is lower than 1024 and the other is higher than or equal to 1024. If destinationTransportPort is lower than 1024 then the flow record is considered to be a request. The utilized classification condition is rather strict to achieve high confidence in the classified samples (a more vague rule would be, for example one of the port numbers is lower and the other is higher).

The algorithm selects an intersection $T \cap P$ and classifies these flow records by timestamps and ports. The estimate e on the timestamp reliability is expressed as the ratio of the number of flow records classified as requests/responses by the timestamps and the ports in mutual agreement and the number of flow records in the intersection plus $|T_equal|$:

$$e = \frac{|T_{req} \cap P_{req}| + |T_{resp} \cap P_{resp}|}{|T \cap P| + |T_{equal}|}, \tag{1}$$

where T_{equal} is a subset of TCP bidirectional records with TCP SYN flags but containing equal start timestamps thus filtered out of T by the fourth condition. If the start timestamp and port heuristic agree on flow ordering then e reaches 100% provided the start timestamps differ. If there are bidirectional flows with equal timestamps or the timestamps do not conform to port heuristic than e drops to 50% under normal circumstances. If e drops below 50% then it means that start timestamps and port numbers are in negative correlation. We also suggest to consider e value only if size of $|T|$ is at least five percent of all flows and the number of all flows is sufficiently large (e.g. 10 million flows).

4 Biflow orientation algorithm

We propose a flow orientation algorithm that decides the orientation based on timestamps and port number heuristic. The decision algorithm is driven by timestamp reliability estimate e. The algorithm orients the flows according to the decision tree depicted in Fig. 2.

Upon a flow or biflow arrival the algorithm follows the tree from root to leaves. If the flow is single then the port numbers must determine the flow orientation. If it is a bidirectional flow then the timestamps decide the orientation first but only if the timestamps can be trusted. The trust is expressed by the condition C_t which includes reliability estimation, flow start heuristic and condition on differing timestamps:

$$C_t : e > R \wedge (TCP \wedge SYN \vee (\neg TCP \wedge duration < D)) \wedge t_1 \neq t_2, \qquad (2)$$

where R is a reliability threshold. Values t_1, t_2 are start timestamps which must differ to determine the order of flows. The second expression identifies flows in which the start timestamps belong to the first packet of the flows. We consider these flows either to be TCP flows with SYN flags or non-TCP flows lasting less than D seconds. The flow fragmentation occurs when the flow duration reaches active timeout (e.g. 300 seconds) under normal circumstances. Therefore records describing long flows (duration close to active timeout) are more likely to be fragments of a long lasting flow and do not account for the first packet of a flow in most cases. The algorithm omits the flow reassembly procedure due to its memory cost and selects flows that are significantly shorter than active timeout. Still the algorithm may account for the last fragment of a flow accidentally. According to our traces, we suggest to setup the thresholds $R = 80\%$ and $D = 180s$. This balances the classification gain when utilizing timestamps and possible errors introduced by incorrect timestamps or fragmented flows.

If the condition C_t does not hold the algorithm proceeds with port number classification. This requires the ports to be available and to differ from each other. The classification results in request/response biflow orientation, single request/response flow and unknown orientation.

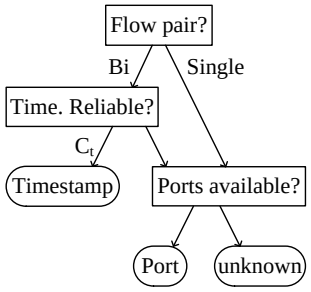

Fig. 2. Bidirectional flow orientation decision tree (C_t is condition described in expression (2)).

5 Evaluation

We analyze our data sets to determine the dependency between timestamps and port number heuristic from the perspective of timestamp reliability estimation algorithm. Subsequently we evaluate our biflow orientation algorithm under various timestamp failures.

5.1 Data sets

The experiments are carried out on three data sets. As a baseline we utilize a data set with verified timestamps. The timestamps are assigned by hardware prior to any buffering and offers nanosecond resolution. Manual analysis of the data set confirmed that the start timestamps are in line with the expected flow ordering. The baseline data set consists of raw packets observed during one day interval on a 10 Gbps backbone network link (link between CESNET and ACONET NRENs (National Research and Educational Network). The packets are aggregated into flow records offline thus no packet drop appears. The packet aggregation was performed by a program written specifically for this task (setup: inactive timeout 30 s, active timeout 300 s, do not interpret TCP flags, no memory limit). The second data set consists of flow records observed during one day interval on an Internet connection of a campus network. This data set consists of flow records which were aggregated online by a metering probe. These two data sets were thinned by accounting only first fifteen minutes of each hour (data set A - backbone (baseline), data set B - campus). In addition to our data sets, we also include data set that is publicly available. This data set was captured on WIDE backbone samplepoint-F on a 155 Mbps line [13]. Again the flow records were aggregated offline hence the only timestamp errors may come from the utilized capture solution (no details are available). Tab. 1 shows basic characteristics of the utilized data sets.

Fig. 3 displays cumulative distribution function of time differences (de facto a round trip time (RTT)) between start timestamps of the bidirectional flows. The

	Flows [mil.]	Packets [mil.]	Bytes [bil.]	Interval
data set A - Aconet	266	10379	8887	2014/01/10 0:00 - 23:15
data set B - VUT	190	7595	6668	2013/12/02 0:00 - 23:15
data set C - Mawi	8	58	27	2014/02/10 14:00 - 14:15

Table 1. Volumes of utilized data sets.

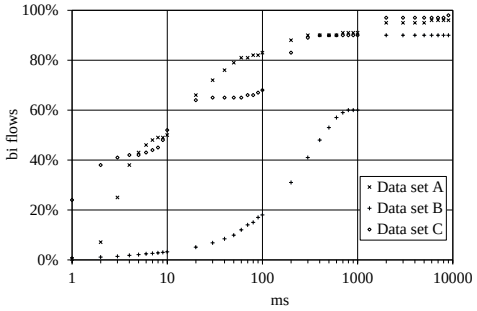

Fig. 3. Time differences between start timestamps of bidirectional flows (please note the logarithmic scale of the x-axis).

data set A and B exhibit similar properties – 90% of start timestamp differences fit into 1000 ms interval. Whereas data set C exhibit longer RTTs – 90% of start timestamp differences fit into 2000 ms interval.

We apply various modifications to data set A to observe dependency between certain timestamp failures and our estimation algorithm as well as our orientation algorithm. Three types of the artificial modifications are proposed:

1. To swap the start timestamps in bidirectional flows with probability $p_w = \{0.01, 0.02, 0.05, 0.1, 0.2, 0.5\}$. The modified data set is denoted as, for example, $A_{p_w=0.1}$. Such a modification may simulate buffering issues.
2. To set up start timestamps to the same value in case the difference between timestamps is less than d ms, $d = \{1, 2, 5, 10, 20, 50\}$. The modified data set is denoted as, for example, $A_{d=1}$. This modification may simulate poor timestamp resolution.
3. To apply random packet sampling prior to flow aggregation with probability $p_s = \{0.01, 0.02, 0.05, 0.1, 0.2, 0.5\}$ (the higher the sampling probability p_s the more packets is accounted for measurement). The modified data set is denoted as, for example, $A_{p_s=0.1}$. This modification may simulate packet drops and sampling.

Data set	$\|A\|$ [mil. flows]	$\|T\|$	$\|P\|$	$\|T \cap P\|$	$\|T_{req} \cap P_{req}\|+$ $\|T_{resp} \cap P_{resp}\|$	$\|T_{equal}\|$	e
A	266	19%	43%	18%	18%	0.02%	100%
B	190	27%	61%	22%	12%	0.04%	54%
C	8	6%	30%	5%	5%	0.7%	89%

Table 2. Breakdown of data sets into subsets.

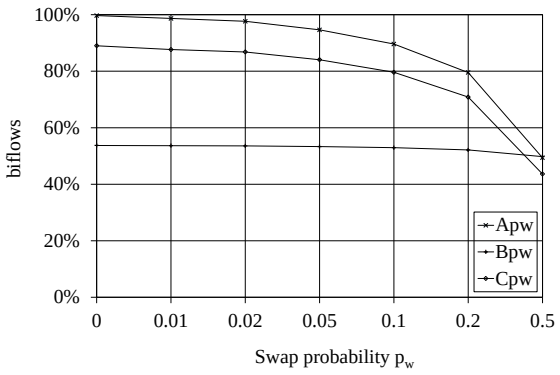

Fig. 4. Impact of the timestamp swap modification on the timestamp reliability estimate e.

5.2 Experimental results

First, we conduct experiments to determine the correlation between start timestamps and port heuristics. If the correlation between port and timestamps is high in case of the non-modified data set and if it differs in case of a modified data set then the reliability estimate algorithm can estimate cases of timestamp failures successfully. Tab. 2 breaks down the flow records in the data sets A, B and C into the subsets utilized in Eq. (1).

The baseline set A exhibit nearly perfect correlation between timestamps and port numbers as $|T_{req} \cap P_{req}| + |T_{resp} \cap P_{resp}|$ and $|T \cap P|$ are of equal size. Therefore the reliability may be worsened by $|T_{equal}|$ only. On the other hand the timestamps and ports correlation is low in the data set B. This conforms already suspicious results presented in Fig. 3 where the cumulative distribution of start timestamp differences is significantly lower to other data sets. Therefore we infer that these start timestamps already bear significant artifacts introduced by the monitoring pipeline. Tab. 2 also shows that in the case of the data set C the size of T is quite low but still above our threshold of 5% out of all flows. On the other hand the correlation of timestamps and ports is high (89%).

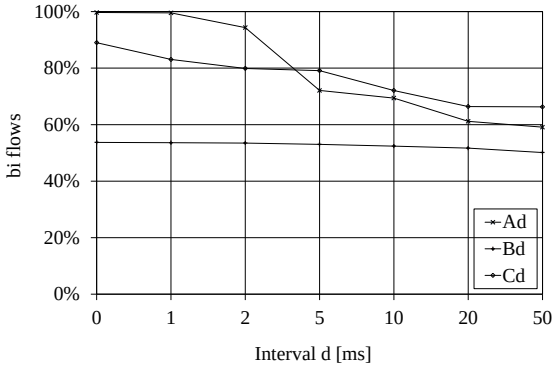

Fig. 5. Impact of the timestamp resolution modification on the timestamp reliability estimate e.

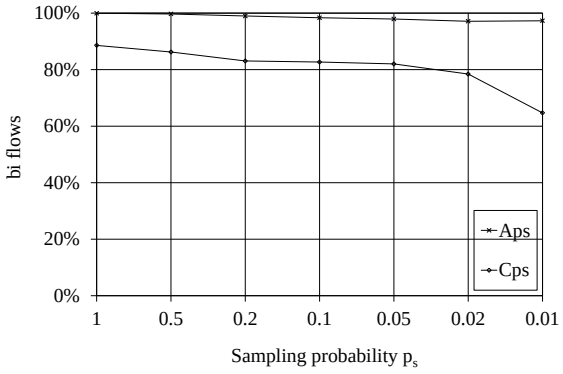

Fig. 6. Impact of the sampling modification on the timestamp reliability estimate e.

The impact of the first modification (start timestamp swap) on e is depicted in Fig. 4. The higher the number of swapped start timestamps the lower e. In case of the data set B the decrease of e is slow since the correlation between timestamps and port numbers is already bad in the first place. The second modification (equalizing timestamps) is applied on the data sets and the result is depicted in Fig. 5. Again Fig. 5 shows that with larger number of equal timestamps the estimation decreases.

	Data set	A			B			C		
Flow type	Classified by	Flows [mil]	PORT	TREE	Flows [mil]	PORT	TREE	Flows [mil]	PORT	TREE
Single flow	port	134	60%	60%	27	43%	43%	1.6	71%	71%
	unknown		40%	40%		57%	57%		29%	29%
Bi. flow	port	132	88%	39%	164	71%	20%	1	74%	7%
	timestamp		0%	57%		0%	70%		0%	75%
	unknown		12%	4%		29%	10%		26%	18%
Bi. flow	errors		8%	0%		nd	nd		8%	0%

Table 3. Comparison of TREE and PORT algorithms in bidirectional orientation problem (nd - not defined).

In case of data set modification by packet sampling we are not able to modify data set B since this data set consists of the flow records only and it is not considered for evaluation. Fig 6 depicts the behavior of e when the sampling rate decreases. Since the data set B consists of already aggregated flow records it is omitted from the evaluation. In case of data set C the estimate decreases with the decreasing sampling rate. In case of the data set A the e remains high despite higher sampling rate but the size of T quickly decreases below 5% for $A_{p_s \leq 0.2}$. Therefore the e should be considered low and timestamps should not be utilized. The experiments with the modified data sets show that the estimate e reacts on start timestamps errors as expected. We proceed with the experiments on the bidirectional flow orientation to demonstrate the ability of e to control the decision algorithm as well as the advantage of bidirectional flow orientation utilizing start timestamps in parallel to port numbers.

First we evaluate our orientation algorithm (TREE) against port number heuristic (PORT) on all non-modified data sets. The port heuristic classifies only such flow records in which one of the port numbers is lower than 1024 and the other is higher than or equal to 1024. The Tab. 3 depicts the capability of each algorithm to classify single flows and bidirectional flows. The single flows are always classified by port numbers since timestamps may only be utilized in case of bidirectional flows. Naturally, some single flows do not match the condition of PORT heuristic and are classified unknown. The bidirectional flows may be classified either by port or timestamps in case of TREE and only by ports in case of PORT. Naturally, some bidirectional flows cannot be oriented neither by timestamps nor by ports and these flows are marked unknown.

On the data set A the PORT heuristic is able to determine 88% of flows belonging to 66 mil. bidirectional flow (132 mil. flows). Since we know that the data set A contains verified timestamps than we can utilize classification results of TREE as a ground truth to estimate errors of PORT. As a result, if PORT is utilized 88% of bidirectional flows is classified but with 8% of errors. On the other hand TREE is able to classify 96% of bidirectional flows without an error. In case of data set B the timestamps cannot be trusted but if utilized the number of classified flows would reach 90% in comparison with only 71% in case

		A_{p_w}					
	A	0.5	0.2	0.1	0.05	0.02	0.01
port	39%[t]	88%[p]	88%[p]	39%[t]	39%[t]	39%[t]	39%[t]
timestamp	57%[t]	0%[p]	0%[p]	57%[t]	57%[t]	57%[t]	57%[t]
unknown	4%[t]	12%[p]	12%[p]	4%[t]	4%[t]	4%[t]	4%[t]
errors	0%[t]	8%[p] 29%[t]	8%[p] 12%[t]	6%[t]	3%[t]	1%[t]	1%[t]

Table 4. Modified data sets A_w and the impact on bidirectional flow orientation algorithm (bidirectional flows only, [t] and [p] mark values belonging to TREE and PORT respectively).

		A_d [ms]					
	A	1	2	5	10	20	50
port	39%[t]	39%[t]	41%[t]	51%[t]	53%[t]	88%[p]	88%[p]
timestamp	57%[t]	57%[t]	55%[t]	44%[t]	42%[t]	0%[p]	0%[p]
unknown	4%[t]	4%[t]	4%[t]	5%[t]	5%[t]	12%[p] 6%[t]	12%[p] 7%[t]
errors	0%[t]	0%[t]	0%[t]	1%[t]	1%[t]	8%[p] 2%[t]	8%[p] 3%[t]

Table 5. Modified data sets A_d and the impact on bidirectional flow orientation algorithm (bidirectional flows only).

of PORT. The classification results for data set C are similar to data set A since the timestamps are deemed correct.

Tab. 4 captures the behavior of TREE algorithm on modified data set A_{p_w}. As the number of swapped timestamps decreases the bidirectional flow algorithm starts to utilize timestamps for orientation. If the timestamps have been used when $p_w = 0.5, p_w = 0.2$ then the classification error would be quite large, 29% and 12% respectively. At the same time when TREE starts to utilize timestamps the error decreases further as well as the percentage of non-oriented biflows.

Tab. 5 captures the behavior of TREE algorithm on modified data set A_d. The algorithm stops utilizing timestamps upon $d = 20, d = 50$ although if the timestamps were used then the percentage of non-oriented flows and the number of errors would be lower. This is a prize given by the fact that we cannot anticipate which timestamp error appears on a given network.

6 Conclusion

The paper proposed two algorithms – first one to estimate the timestamp reliability and the second to orient bidirectional flows according to start timestamps and port numbers. Our future work will test the algorithms further to fully evaluate the performance of the classifiers on empirical data sets containing timestamp failures.

Acknowledgment. This research has been partially supported by the CESNET Large Infrastructure project no. LM2010005 funded by the Ministry of Education, Youth and Sports of the Czech Republic.

References

1. E. Hughes and A. Somayaji, "Towards network awareness." in *LISA*. USENIX, 2005, pp. 113–124. [Online]. Available: http://dblp.uni-trier.de/db/conf/lisa/lisa2005.html
2. B. Trammell and E. Boschi, "Bidirectional Flow Export Using IP Flow Information Export (IPFIX)," RFC 5103 (Proposed Standard), Internet Engineering Task Force, Jan. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5103.txt
3. P. Minarik, J. Vykopal, and V. Krmicek, "Improving host profiling with bidirectional flows," in *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 03*, ser. CSE '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 231–237. [Online]. Available: http://dx.doi.org/10.1109/CSE.2009.23
4. H. Jiang and C. Dovrolis, "Passive estimation of tcp round-trip times," *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 3, pp. 75–88, Jul. 2002. [Online]. Available: http://doi.acm.org/10.1145/571697.571725
5. R. Berthier, M. Cukier, M. Hiltunen, D. Kormann, G. Vesonder, and D. Sheleheda, "Nfsight: netflow-based network awareness tool," in *Proceedings of the 24th international conference on Large installation system administration*, ser. LISA'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 1–8. [Online]. Available: http://dl.acm.org/citation.cfm?id=1924976.1924988
6. I. Cunha, F. Silveira, R. Oliveira, R. Teixeira, and C. Diot, "Uncovering artifacts of flow measurement tools," in *Passive and Active Network Measurement*, ser. Lecture Notes in Computer Science, S. Moon, R. Teixeira, and S. Uhlig, Eds. Springer Berlin Heidelberg, 2009, vol. 5448, pp. 187–196.
7. R. Hofstede, I. Drago, A. Sperotto, R. Sadre, and A. Pras, "Measurement artifacts in netflow data," in *Passive and Active Measurement*, ser. Lecture Notes in Computer Science, M. Roughan and R. Chang, Eds. Springer Berlin Heidelberg, 2013, vol. 7799, pp. 1–10.
8. J. Kogel, "One-way delay measurement based on flow data: Quantification and compensation of errors by exporter profiling," in *Information Networking (ICOIN), 2011 International Conference on*, Jan 2011, pp. 25–30.
9. S. McPherson and A. Ortega, "Analysis of internet measurement systems for optimized anomaly detection system design," *CoRR*, vol. abs/0907.5233, 2009. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr0907.html
10. B. Trammell, B. Tellenbach, D. Schatzmann, and M. Burkhart, "Peeling away timing error in netflow data," in *Passive and Active Measurement*, ser. Lecture Notes in Computer Science, N. Spring and G. Riley, Eds. Springer Berlin Heidelberg, 2011, vol. 6579, pp. 194–203.
11. B. Claise, "Cisco Systems NetFlow Services Export Version 9," RFC 3954 (Informational), Internet Engineering Task Force, Oct. 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3954.txt
12. P. Haag, "Nfdump - netflow processing tools," 2013. [Online]. Available: http://sourceforge.net/projects/nfdump/
13. "The mawi archive, http://mawi.wide.ad.jp/mawi/," 2014.