Brief Announcement: Complexity and Solution of the Send-Receive Correlation Problem

Benito van der Zander Egon Wanke Wolfgang Kiess Björn Scheuermann
Heinrich Heine University, Düsseldorf, Germany
{zander, kiess, scheuermann}@cs.uni-duesseldorf.de, wanke@acs.uni-duesseldorf.de

ABSTRACT

During the analysis of packet log files from network experiments, the question arises which received packet belongs to which of the potentially many binary identical send events. We discuss this send-receive correlation problem for networks with local broadcast media. We can prove that assigning send and receive events is an NP-complete problem. However, there is a solution algorithm that is exponential only in the number of nodes; if the number of network nodes is fixed, its complexity is polynomial.

Categories and Subject Descriptors: C.2 [Computer-Communication Networks]: General; C.4 [Performance of Systems]: Measurement Techniques; F.2.2 [Algorithm Analysis, Complexity]: Nonnumerical Algorithms and Problems

General Terms: Algorithms, Experimentation, Theory

1. INTRODUCTION

During typical computer network experiments, each node records a packet log file. These logs are the basis for evaluating the experiment. Unfortunately, in many network protocols, packets are not always unique, in the sense that binary identical packet transmissions can occur. This may, for instance, be due to retransmissions of non-acknowledged data or repeated ARP requests, and frequently occurs for certain WiFi control frames. Given a reception event of a non-unique packet, it is not immediately clear which transmission of which other node is the corresponding one. We know what the received packet looked like—but we may be faced with many matching transmission events. This can be highly problematic for the calculation of performance metrics or if causal relationships are traced back. We call this the *send-receive correlation problem*.

This problem is not straightforward to overcome. Piggybacking additional information onto the packets to make them unique is clearly not a valid option, for at least two reasons: first, this would require modifications in the application, the operating system's network stack, or even hard- and firmware. This may be difficult to accomplish, can introduce errors, and may alter the system behavior. Second, more data in the packets increases their size and thus the transmission duration. The altered medium usage and interference pattern can significantly influence protocol behavior and performance, especially on lower layers. It is then not clear whether the system under test still behaves like an unmodified one; this is incompatible with a clean experiment design. This is also the reason why existing work on virtual clocks and timestamps—starting with Lamport's milestone work [2]—is not directly applicable.

PODC'10, July 25–28, 2010, Zurich, Switzerland.
ACM 978-1-60558-888-9/10/07.

Due to its fundamental nature, the send-receive correlation problem has kept appearing in the literature on network experiments—even though it was seldom recognized with all its consequences. For instance, it lies at the heart of the trace analysis methodologies discussed in [3, 6].Determining send-receive relationships is also a necessary prerequisite for offline clock synchronization techniques like [5]. Yet, so far, only heuristics based on node-recorded event timestamps have been discussed. But such timestamps are inherently unreliable, because they are affected by many unpredictable factors [5]. So, these heuristics always carry a risk of wrong assignments and may also fail completely [3].

We are the first to consider the send-receive correlation problem more fundamentally. We look at the case of local broadcast networks, and ask what we can tell *for sure* about corresponding send and receive events. To this end, we build upon the order of packet transmissions and receptions in the log files. Our previous extended abstract [4] introduced the problem, but left the questions about its complexity and a definite solution open. We are now able to prove that the problem is NP-complete, and have found a solution algorithm that takes exponential time only in the number of network nodes, but not in the length of the logs.

2. LOCAL BROADCAST NETWORKS

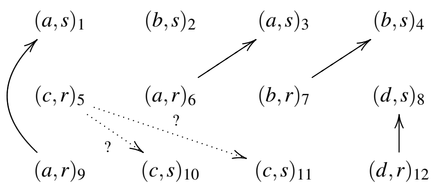
We consider networks with local broadcast media like WiFi networks, classical CSMA/CD Ethernet, or mobile and vehicular adhoc networks (MANETs, VANETs). We make the following assumptions that hold in many practically relevant local broadcast networks, including those mentioned above: 1) a transmission can be recorded by multiple receivers and 2) if multiple nodes generate log entries for the same pair of transmissions (as sender or as receiver), then these two transmissions are logged in the same order.

Essentially, property 2) means that transmissions cannot "overtake" each other: they may not appear in one order at one node and in a different order somewhere else. We can use this to "check" a hypothesis about which packet reception stems from which transmission: if we substitute all receptions with their assumed transmissions, we obtain sequences of transmissions, one per log file. If our hypothesis is compatible with what has been observed, these sequences must not be contradictory with respect to the event order.

Figure 1(a) illustrates the send-receive correlation problem with log files from an experiment with three nodes, one log file per line. Events have been recorded in the order from left to right. Each event is characterized by 1) a type (events are of the same type if they cannot be distinguished by looking at the transmitted data) and 2) whether a packet was sent (s) or received (r). The numerical indices in the figure serve the only purpose to allow us to refer to individual entries. In the example node 1 transmits four packets, of types a, b, a, and b (the first/third and second/fourth transmis-

$(a,s)_1$	$(b,s)_2$	$(a,s)_3$	$(b,s)_4$
$(c,r)_5$	$(a,r)_6$	$(b,r)_7$	$(d,s)_8$
$(a,r)_9$	$(c,s)_{10}$	$(c,s)_{11}$	$(d,r)_{12}$

(a) Log files.



(b) Intuitive solution.

Figure 1: An example set of local packet log files.

sions are two binary identical pairs). The second node records three packet reception events of types c, a, and b, before it transmits a packet of type d, and so on. Figure 1(b) shows what the event order tells us about send-receive pairs. For instance, $(d,r)_{12}$ must stem from the only transmission of type d, i. e., from $(d,s)_8$. $(c,r)_5$'s source can be either $(c,s)_{10}$ or $(c,s)_{11}$; it is not possible to decide between these two. However, the events of type c still provide helpful information: since $(a,r)_9$ was received *before* $(c,s)_{10}$ and $(c,s)_{11}$ were transmitted, and since $(a,r)_6$ took place *after* $(c,r)_5$, $(a,r)_9$ and $(a,r)_6$ cannot belong to the same send event. Since there are only two transmissions of a in total, we infer that $(a,r)_9$ belongs to the earlier transmission $(a,s)_1$, and $(a,r)_6$ was a reception of $(a,s)_3$. Similarly, all assignments in Figure 1(b) can be deduced.

In summary, such log files allow to derive information about possible and impossible send-receive pairs. But, as in the case of $(c,r)_5$, we will not always end up with definite assignments. Instead, for each receive event in all log files, there is a *set* of possible send events. For a reception event x, we call this set S_x the *send candidate set* of x.

3. RESULTS

To describe send candidate sets formally, we use *consistent global assignments*. Such an assignment maps each reception to a send event of matching type, such that no contradictions to any of the event orders in local log files occur. A send event y is in the send candidate set of receive event x iff a consistent global assignment exists that assigns x to y.

Given log files from an experiment, the most simple question is the decision problem: is there any consistent global assignment at all for a given set of logs? We call this the *send-receive correlation existence problem* (SRCEX). We can show that SRCEX is NP-complete; this implies that calculating send candidate sets is hard, too. The central proof idea is a polynomial-time transformation from the NP-complete problem EXACT COVER BY 3-SETS (X3C) [1]. Each set in an X3C instance is mapped to one sender node in SRCEX, one single receiver node "collects" the transmissions from all these sources. This is done in a way which ensures that a consistent global assignment exists iff X3C has a solution.

In a number of practical settings, binary identical packet transmissions are never initiated by distinct sender nodes. This *unique senders property* holds, for example, if transmissions include the sender's address. For a receive event, we may then still be in doubt which exact transmission was the source although we know the sender. One might expect that the respectively constrained problem, here termed USRCEX, is simpler. However, it is also NP-complete—even though the proof is significantly more tricky. We obtained it by first exchanging the roles of send and receive events in the proof of SRCEX, and then inserting a kind of "guard events" which force every send event to be received exactly once.

These results signify that we cannot expect to solve send-receive correlation problems in polynomial time, not even the unique sender variant. However, we found that SRCEX can be decided in polynomial time if the number of nodes is fixed. We devised an algorithm that verifies the existence of a consistent global assignment in time $O\left(|\mathcal{N}| \cdot \prod_{i\in\mathcal{N}} l_i\right)$, where $\mathcal{N}$ is the set of nodes and l_i is the number of entries in node i's log file. The algorithm is based on a breadth-first exploration of the space of consistent global assignments for increasingly larger fractions of the log files. It avoids to traverse large portions of this space by making greedy assignments where this will provably not miss a valid solution.

The algorithm can also be used to calculate send candidate sets. To this end, an assignment of a receive event x to a type-matching send event y can be "tested" by inserting an additional, artificial sequence of events into the log files around x and y. This can be done in a way which enforces a specific assignment without affecting any other assignment possibilities. If a consistent global assignment for the resulting, modified instance of SRCEX exists, then $y \in S_x$.

For instances where the unique senders property holds, a heuristic solution algorithm—based on ideas outlined in [4]—is able to significantly reduce the size of the solution space in polynomial time. With the unique senders property it is possible to identify an "earliest possible" and a "latest possible" send event for each reception. It is then possible to exclude send candidates based on partial order information. This narrows down the sets of send candidates for all receive events in parallel. The exclusions, in turn, provide refined information on the event order, so that the process can be iterated. It can be proven that this algorithm will never exclude too much, i. e., it calculates supersets of the send candidate sets. In order to determine exact send candidate sets for instances with the unique senders property, the exact solution algorithm can be combined with these ideas. By excluding many send candidates with the heuristic as a preprocessing step, the number of checks with the exponential-time decision algorithm can be vastly reduced.

4. REFERENCES

[1] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[2] L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Comm. ACM*, 21(7):558–565, July 1978.

[3] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Analyzing the MAC-level behavior of wireless networks in the wild. In *SIGCOMM '06*, Sept. 2006.

[4] B. Scheuermann and W. Kiess. Who said that? – the send-receive correlation problem in network log analysis. *ACM SIGMETRICS Performance Evaluation Review*, 37(2):3–5, Sept. 2009.

[5] B. Scheuermann, W. Kiess, M. Roos, F. Jarre, and M. Mauve. On the time synchronization of distributed log files in networks with local broadcast media. *IEEE/ACM Transactions on Networking*, 17(2):431–444, Apr. 2009.

[6] J. Yeo, M. Youssef, and A. Agrawala. A framework for wireless LAN monitoring and its applications. In *WiSe '04*, Oct. 2004.