# Poster: State Tracking and Verification for Distributed Nodes through Traffic Inspection

Konrad Altenhofen
konrad.altenhofen@tu-darmstadt.de
Technical University of Darmstadt
Communication Networks Lab
Darmstadt, Germany

Julian Zobel
julian.zobel@tu-darmstadt.de
Technical University of Darmstadt
Communication Networks Lab
Darmstadt, Germany

Björn Scheuermann
scheuermann@kom.tu-darmstadt.de
Technical University of Darmstadt
Communication Networks Lab
Darmstadt, Germany

## ABSTRACT

Keeping track of protocol, node, or system states in a distributed network is non-trivial. However, state tracking combined with predefined allowed transitions for each state is essential to verify that nodes adhere to the specification and, finally, to detect faulty or malicious nodes. Typically, existing systems only consider context in network flows instead of the full states or use a simplified state model. Therefore, we examine challenges and factors influencing a state tracking and verification system.

## CCS CONCEPTS

• **Computing methodologies → Distributed algorithms**.

## KEYWORDS

traffic inspection, state tracking, transition validation

**ACM Reference Format:**
Konrad Altenhofen, Julian Zobel, and Björn Scheuermann. 2024. Poster: State Tracking and Verification for Distributed Nodes through Traffic Inspection. In *The 18th ACM International Conference on Distributed and Event-based Systems (DEBS '24), June 24–28, 2024, Villeurbanne, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3629104.3672431

## 1 INTRODUCTION

The observation of network traffic reveals many insights into the involved communication partners. The gathered observations can be used to detect faulty or malicious nodes. In some cases, individual messages viewed independently from each other provide enough information to detect undesired behavior. However, context or state information is required in many other cases to determine if a node adheres to specifications and its desired behavior. In this scope, the meaning of *state* can reach from protocol states of a system, states of a single node that are controlled through network messages, or a global system state, which consists of the state of multiple distributed nodes.

Intrusion Detection Systems (IDS) use network sensors to observe traffic and identify malicious traffic. For traffic classification, some IDS tools split the traffic into flows to consider simple states per flow, e.g., if a connection is already established [1], or context, e.g., the last 10 packets per flow. Firewalls also observe and analyze traffic to determine whether to forward or block it. Some firewalls also use connection state information to classify their traffic, e.g., NFTables [2]. However, states like *new*, *established*, *related*, *invalid*, and *untracked* only provide limited insights. Additionally, the mechanisms for tracking the connection state are kept very simple. While more complex state tracking mechanisms, combined with predefined transitions, enable more flexible and in-depth detection of nodes not adhering to specification, network influences such as delay and packet loss make this task non-trivial. Similar problems are considered in model checking using distributed transition systems [3]. However, they are used to verify a specification before deployment. In contrast, we aim at state tracking and verification for distributed, live systems through traffic inspection.

This work examines three important aspects of state tracking systems in distributed networks. Section 2 considers the disclosure of state and transition information through communicated messages, while Section 3 examines the influences within a network that impact state tracking. Both aspects are indispensable enablers for any state tracking system. Based on these, Section 4 discusses the issues of state tracking for distributed nodes through traffic inspection under varying network influences.

## 2 STATE INFORMATION DISCLOSURE

Naturally, enabling system state tracking based on network traffic observations requires the disclosure of state information within messages. When and how information is disclosed by network traffic influences the complexity of state tracking. Specifically, state information can be disclosed through (i) control messages triggering a transition, (ii) transition-specific messages, or the inclusion of explicit or implicit state or transition information in (iii) normal messages or (iv) acknowledgment messages.

## 3 NETWORK INFLUENCES

As a result of network traffic observation, multiple network factors influence the prediction of system states. Such factors include (i) communication delay, (ii) packet loss, (iii) packet reordering, and (iv) alternate message paths. Communication delay is prevalent in every network and temporarily induces inconsistencies between sender, sensor, and receiver. On the other hand, impact types (ii)-(iv) are strongly dependent on the network. While packet reordering and alternate message paths may be unlikely with careful placement and network design, packet loss remains likely in most networks. Network-dependent impact factors ultimately prevent a state tracking system from determining a single, definitive current state. Thus, we need to keep a set of potential current states.

## 4 WORST-CASE PERMITTED STATES AND POTENTIAL CURRENT STATES

To highlight the issues and complexities that arise with state tracking and verification for distributed nodes through traffic inspection, we examine the worst-case number of potential current states from the view of a state tracking system. For that, we assume a single sender and receiver, with a sensor between them that can observe all traffic. All state transitions at the receiver side are deterministic and triggered by sender messages. All trigger messages include disclosed, explicit, and absolute state information. In the following, we examine possible issues with state tracking and transitions under varying combinations of packet loss and delay.

In the case of optimal network conditions, i.e., **no delay and no packet loss**, at most one current state is possible. Thus, the worst-case number of permitted next states is defined by the state with the most transitions.

For a network model **with delay and no packet loss**, the sensor cannot determine a single, current state with absolute certainty. Instead, it can only determine a set of potential current states. The number of potential states depends on the maximum number of in-flight command messages: If only one in-flight message is allowed, the current state could either be the previous or the new state induced by that message. For two in-flight messages, the number of potential current states is increased to three: i.e., the old state or the state of the first or second message. With $n$ in-flight messages, the number of potential current states is $n + 1$ but is limited by the number of possible states. However, the allowed transitions only depend on the last state set by the control traffic, for one single sender and no message re-ordering. As a result, the worst-case number of permitted next states is, again, defined by the state with the most transitions.

State tracking gets more difficult when considering a network with **no delay but with packet loss**. In this case, each message is either (i) correctly received by both sensor and receiver, (ii) lost before the sensor (both sensor and receiver do not see the packet), or (iii) lost after the sensor (only the sensor receives the packet, but not the receiver). Case (ii) does not directly impact the tracking system, since the sensor does not capture the message. Nevertheless, this causes a desynchronization between the sender on one side and the sensor and the receiver on the other. As a result, the sender might send transition requests that are neither permitted by the sensor view nor the actual state of the receiver. In contrast, case (iii) causes a desynchronization between the receiver on one side and the sender and sensor on the other.

To provide reliable communication over networks with packet loss, typically, acknowledgment messages are used to indicate that packets have been received. If no acknowledgment is received after a certain time, the sender initiates a re-transmission of the unacknowledged message. Sequence numbers are used to order messages and to identify (lost) messages.

Nonetheless, acknowledgments can also be affected by packet loss. Assuming both receiver and sensor ignore messages with sequence numbers larger than previously received, retransmissions due to lost acknowledgments between sensor and sender do not interfere with the state tracking functionality — note that we retain the established naming here, i.e., the sender is still called *sender* despite receiving the acknowledgments. However, since acknowledgment messages might get lost between the receiver and the sensor, the sensor cannot determine the current system state for certain.

The maximum number of potential current states is equal to the maximum number of messages that can be sent without acknowledgment reception. Without sequence number mapping in the sensor, the maximum number of permitted next states is the sum of unique transitions for $n$ consecutive states, where $n$ is the maximum number of potential current states. With sequence number mapping, this number can be reduced. However, the validation that each consecutive transition is permitted requires two tracking mechanisms. One tracking mechanism must verify transitions in consecutive messages, and the other one for retransmissions, based on the last acknowledged state, the re-transmitted message, and the already sent next state.

In the worst case, no acknowledgments are used and the state tracking system can never determine a state with certainty. Thus, the worst-case number of potential current states equals the number of all possible states. Furthermore, the worst-case number of allowed transitions is the sum of all unique transitions.

The worst-case number of states in a network model **with delay and packet loss** does not change compared to the network model without delay. However, the delay of packets and acknowledgments causes the need to wait to identify if retransmissions are required. As a result, the number of unacknowledged in-flight messages is higher than in the scenario with packet loss but without delay. This also causes the number of to-be-tracked states to be higher in the non-worst case.

## 5 CONCLUSION AND OPEN ISSUES

Keeping track of states using network traffic allows the verification of transitions, and, finally, to detect faulty or malicious nodes in a network. However, existing systems only consider context in network flows or use a simplified state model.

This poster examines how and which state information can be disclosed through network traffic, which influences the feasibility and complexity of state tracking. Furthermore, we look at the network factors such as delay and packet loss, which further impact the complexity and prevent a state tracking system from determining a single, definitive state. Using different network models, we analyzed the worst-case number of potential current states and the maximum number of permitted transitions a state-keeping system has to track. In the future, we want to use our observation to implement network-based state tracking and verification. Further examination of potential optimizations and compacting is needed, too. While many considerations can be applied to systems consisting of multiple nodes, adaptions and further work are needed.

## REFERENCES

[1] [n. d.]. *flow - Snort 3 Rule Writing Guide.* https://docs.snort.org/rules/options/non_payload/flow
[2] 2021. *Matching connection tracking stateful metainformation - nftables wiki.* https://wiki.nftables.org/wiki-nftables/index.php/Connection_Tracking_System
[3] Clemens H. Cap. 2000. *Distributed Transition Systems.* Vieweg+Teubner Verlag, Wiesbaden, 59–116. https://doi.org/10.1007/978-3-322-86763-6_3