Project thesis: Compositional Testing of Communication Systems - Tools and Case Studies

Tom Ansay

December 15, 2008

This thesis has been supervised by
Prof. Reinhard Gotzhein, Fachbereich Informatik, Technische Universität Kaiserslautern, Germany and
Prof. Ferhat Khendek, Faculty of Engineering and Computer Science, Concordia University, Canada

Contents

1 Introduction

Like in most domains of computer science, the field of protocol engineering investigates the use of component based software engineering and its reusability.

On the other hand there are well founded test methods that guarantee the correctness of a protocol. These test methods are applied to a whole system. Thus, some components are tested multiple times when existing code is reused.

The C-Method [8] was developed to overcome the superfluous testing of a reused component. To achieve this goal, the C-Method first tests the component by a known test method, like the UIOv-method [10]. If the component is tested successfully, the composed system is tested by constructing test suites that are optimized to only test the inter-component behaviour.

The C-Method should save significant testing effort because the effort to verify the overall protocol increases exponentially with the number of possible states. With the approach of the C-Method, only single components and the glue code are tested. To be efficient, the component tests and the construction of the test suites and verification of the glue code must require less effort than to verify the composite system.

If a new protocol is constructed by reusing components that were already tested in another protocol, we simply integrate the test along with the component in the code base and thus reuse not only the component but also the test suite itself.

This project thesis is primarily concerned with the C-Method and looks further into the process that constructs the test suites. In particular, a tool is implemented to construct the test suites, and several case studies are performed.

2 C-Method

We assume that protocols are specified using finite state machines (FSM). Furthermore, an individual component consisting of an FSM and its implementation is correct if the code exposes the same behaviour as the FSM. Since the implementation is a black box, the most important criterion is the input- and output-alphabet of the FSM.

In the following, we consider only testing methods relying on protocols specified as FSMs.

2.1 Fundamentals and Graph Algorithms

We use the standard definition of the FSM, but a different notation:

Definition 1 *A* finite state machine (FSM) M *is a tuple* $(S, I, O, s_0, \lambda_e)$ *with:*

- S *is a finite set of states.*
- I *is a finite input alphabet.*
- O *is a finite output alphabet.*
- $s_0 \in S$ *is the initial state.*
- $\lambda_e \subseteq S \times I \times O \times S$ *defines the* transitions *of* M.

An FSM is *completely specified*, if for each state and each input, a transition is defined. This is not necessary for the C-Method but for most of the testing methods which are used in combination with the C-Method. One possibility is to amend the FSM by introducing a transition, in each state, that leads to an error state with an error output, for every possible input.

Definition 2 *A* completely specified finite state machine ($csFSM$) $N = (S, I, O_e, s_0, \lambda)$ *is derived from a* FSM $M = (S, I, O, s_0, \lambda_e)$ *as follows:*

- S, I, s_0 *as in* M.
- $O_e = O \cup \{e\}$, *where* $e \notin O$ *is called* error output.
- $\lambda = \lambda_e \cup \lambda_i$ *is the transition relation of* N. *Tuples of* λ *are called* transitions *of* N.
- λ_e *defines the* explicit transitions *of* N.
- $\lambda_i = \{(s, i, e, s) \in S \times I \times O_e \times S | \neg\exists o \in O, s' \in S : (s, i, o, s') \in \lambda_e\}$ *defines the* implicit transitions *of* N.

Step 1: Start the construction of $H(s)$ with its root node n_r, labeled with s.

Step 2: Assume $H(s)$ has been constructed up to level k, $k \geq 1$. Then level $k+1$ is built by examining the nodes of level k:

- Step 2.1: A node n of level k is terminated, if its label is identical to the label of a node on level j, where $1 \leq j < k$, or if it is identical to s_0.
- Step 2.2: Otherwise, let s denote the label of node n. Then, for all transitions (s, x, y, s'), attach a branch and successor node to the current node labeled x/y and s', respectively.

Step 3: Prune the resulting tree by successively removing all leaf nodes that have a label $s \neq s_0$, and the corresponding edges.

Table 1: Construction of a homing tree $H(s)$

In the following, the transition λ_i with error output e is omitted for readability reasons.

For the communication of the *csFSM*s we assume an asynchronous reliable medium. The FSMs communicate via exchange of messages, which are collected in an ordered input queue, respectively, for each FSM. Often the messages are also referred to as signals and are modeled as input and output of the finite state machine transitions.

The FSM is represented as a labeled directed graph, where states correspond to nodes, and transitions correspond to edges labeled with input and output.

Definition 3 *A* labeled directed graph *G is a tuple (V, L, E), consisting of a set of nodes V, a set of labels L, and a relation $E \subset V \times V \times L$, defining the directed edges of the graph. A* path *is a non-empty sequence of consecutive edges. A* tour *is a path that starts and ends at the same node. It is called* minimal *if no edge is contained more than once in the tour. An* initial tour *is a tour that starts and ends at the initial node. A directed graph G is* strongly connected *if for each pair of nodes (v, v'), there is a path from v to v', with $v \neq v'$.*

Definition 4 *Given a csFSM $N = (S, I, O_e, s_0, \lambda)$ and a state $s \in S$, where the underlying graph is strongly connected, a* homing tree *$H(s)$ is a minimal tree that covers all cycle-free paths of N leading from s to the initial state s_0.*

The underlying graph of a directed graph is the graph where all directed edges have been replaced with undirected edges. The method to construct homing trees is described in Table 1 and an example is shown in Figure 1.

Definition 5 *Let $N = (S, I, O_e, s_0, \lambda)$ be a csFSM with the underlying graph G, where G is strongly connected. An* initial tour coverage tree *T is a tree containing all minimal initial tours such that every edge is covered at least once and no tour is contained as a prefix or a suffix of another tour in the set.*

The method to construct an initial tour coverage tree is described in Table 2 and an example is shown in Figure 2.

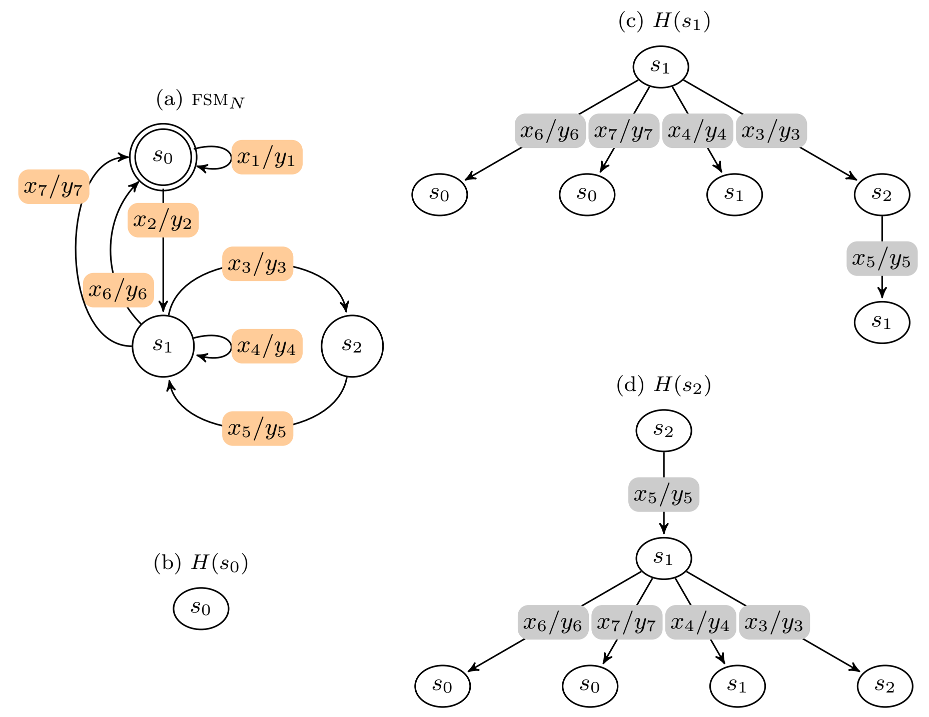


Figure 1: Homing trees example

In order to use the C-Method we assume that the FSMs are at least able to synchronize in their initial states. Two FSMs are synchronized in their initial state if one FSM reaches its initial state and the second FSM reaches its initial state by processing the messages in its input queue without any further interaction. The original C-Method refers it suffices to consider the initial tours of both FSMs and to check if there is a matching tour of the other FSM. We will see later that it is necessary to consider matches of concatenations of initial tours, but the argument is the same.

Definition 6 *A* test case tc *is a non-empty sequence of inputs* $i_1 \cdot i_2 \cdot \ldots \cdot i_n$*. A* test suite ts *is a non-empty set of test cases* $tc_1, tc_2, \ldots, tc_m$*. An* augmented test case atc *is defined as a non-empty sequence of tranisitions (also called test elements)* $i_1/o_1 \cdot i_2/o_2 \cdot \ldots \cdot i_n/o_n$*. An* augmented test suite ats *is a non-empty set of augmented test cases* $atc_1, atc_2, \ldots, atc_m$*.*

2.2 Original C-Method

The original C-Method is the method in Table 3 which is taken from [8].

There are several foci on how to test composed systems. Most assume different preconditions and composition scenarios. The idea of the C-Method is to verify a composed system assuming that the components have already been verified and are correct. Furthermore, the communication medium is assumed to be lossless and to preserve the signal order. Thus, we generate test suites from

Step 1: For each state s of N, construct a homing tree $H(s)$.

Step 2: Start the construction of T with the root node n_r, labeled with the initial state s_0 of N. This is level 1 of T.

Step 3: Assume that T has been constructed up to level k, $k \geq 1$. Then level $k+1$ is built by examining the nodes of level k:

- Step 3.1: A node n of level k is terminated, if its label is identical to the label of a node on level j, where $1 \leq j < k$.
- Step 3.2: Otherwise, let s denote the label of node n. Then, for all transitions $(s, x, y, s^{'})$, attach a branch and successor node to the current node labeled x/y and $s^{'}$, respectively.

Step 4: To each leaf node n, attach the homing tree $H(s)$ by merging the root node of $H(s)$ with n, where s denotes the label of n.

Table 2: Construction of the initial tour coverage tree T

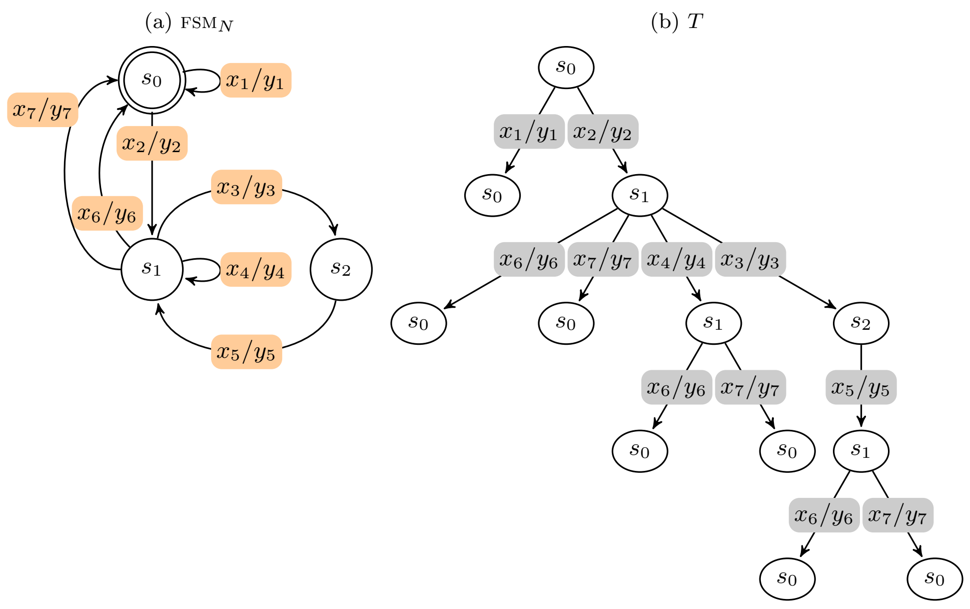


Figure 2: Initial tour coverage tree example

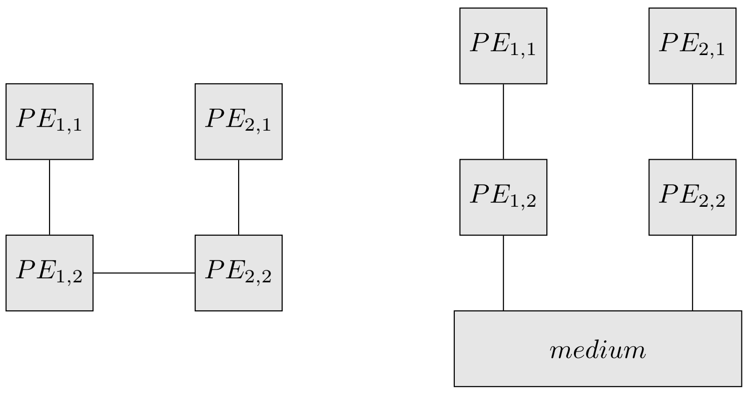


Figure 3: Concurrent composition of protocol entities

the components' interactions and reduce those to the input- and output-signals from and to the environment, which we can observe and control. If those test cases are executed on the composed system and if another output-sequence is detected than the one stated by the given test case we can assume that the glue code is defective, because all the components have already been verified.

In order for the FSMs to communicate, we need composition rules. In the C-Method, we use *concurrent composition* to compose two FSMs. Another type of composition would be sequential composition which is often used in general component-based software systems. For the specification, concurrent composition is expressed by the composition operator $||$, which is replaced by glue code in the implementation.

Put differently: Let N_1 and N_2 be the specifications of two components, and I_1 and I_2 be their implementations, where I_1 and I_2 satisfy their specifications N_1 and N_2, respectively. Then, derive a minimal test suite that is sufficient to check whether the system I consisting of I_1, I_2, and glue code satisfies the specification $N_1||N_2$.

Protocols are often structured as layers of different protocol entity blocks. The concurrent composition can be applied to any two neighboring blocks. The blocks can be local or remote. For example, we may compose protocol entities $PE_{1,1}$ and $PE_{1,2}$ as well as $PE_{1,2}$ and $PE_{2,2}$ concurrently (Fig. 3). The glue code may consist of internal data structure and methods or an entire logical communication medium. The C-Method can also be applied recursively. An example would be to compose the protocol entities given by the composition of $PE_{1,1}$ and $PE_{1,2}$ as well as $PE_{2,1}$ and $PE_{2,2}$.

Definition 7 *Let $N_1 = (S_1, I_1, O_1, s_{0,1}, \lambda_1)$ and $N_2 = (S_2, I_2, O_2, s_{0,2}, \lambda_2)$ be a csFSMs. Let $OI_{1,2} = O_1 \cap I_2 (OI_{2,1} = O_2 \cap I_1)$ be the set of signals exchanged between N_1 and N_2 (N_2 and N_1).Signals in $OI_{1,2}$ and $OI_{2,1}$ are called internal signals. The concurrent composition of N_1 and N_2, denoted $N_1||N_2$, is defined by the derived state machine $Q = (S, I, O, s_0, \lambda)$ with:*

- *$S = S_1 \times {I_1}^* \times S_2 \times {I_2}^*$ is the set of states.*

- *$I = (I_1 \setminus OI_{2,1}) \cup (I_2 \setminus OI_{1,2})$ is the (finite) input alphabet.*
- *$O = (O_1 \setminus OI_{1,2}) \cup (O_2 \setminus OI_{2,1})$ is the (finite) output alphabet.*
- *$s = (s_{0,1}, <>, s_{0,2}, <>)$ is the initial state, consisting of the initial states of N_1 and N_2 and the initial states of the input queues associated with N_1 and N_2, respectively.*
- *$\lambda \subseteq S \times I \times O \times S$ is the transition relation of Q. Tuples of λ are called transitions of Q. λ is derived from λ_1 and λ_2 as follows:*
  $(s, i, o, s') \in \lambda$ with $s = (s_1, q_1, s_2, q_2)$ and $s' = (s'_1, q'_1, s'_2, q'_2)$ iff

  $(\exists (s_1, i, o, s'_1) \in \lambda_1 : (q_1 =< i >^\frown q'_1 \wedge (q'_2 = \textit{if } o \in OI_{1,2} \textit{ then } q_2{}^\frown < o > \textit{ else } q_2) \wedge s_2 = s'_2)) \vee$
  $(\exists (s_2, i, o, s'_2) \in \lambda_2 : (q_2 =< i >^\frown q'_2 \wedge (q'_1 = \textit{if } o \in OI_{2,1} \textit{ then } q_1{}^\frown < o > \textit{ else } q_1) \wedge s_1 = s'_1))$

Where $< a >^\frown < b >=< a, b >$ is the concatenation of two sets.

It is possible to compose all kinds of *$csFSM$*s. Depending on the intended global behaviour, this is not always meaningful. However, some general *composition criteria* can be stated:

CC_1: Internal signals of either machine are eventually consumed by the other machine in an explicit transition, i.e. the composed system is free of internal unspecified receptions. This excludes transitions that have been added to obtain a completely specified state machine, i.e., implicit transitions yielding an error output (see Def. 2).

CC_2: The composed system is free of internal deadlocks. Since it is assumed that external signals can be produced in any order, this again restricts the internal interaction only.

Definition 8 *Let q be a* queue *with $q =< e, q' > \vee <>$, where e is the head element of the queue. An* ordered input queue *is a queue associated with a* FSM *$N_1 = (S, I, O, s_0, \lambda_e)$(Def. 1). The order of the elements in the queue $q_{N_1} =< i_1, \ldots, i_n >$, $i_1, \ldots, i_n \in I$ cannot be changed. At runtime, the head of the queue (i_1) is the next element that is processed by the* FSM.

One of the preconditions of the C-Method is a perfect medium, so no exchanged messages are lost or reordered. Furthermore the transmission needs no time. A system P of interacting FSM $N_1, \ldots, N_m$ is defined at runtime through the definition of a FSM(Def. 1) and an ordered input queue associated with each FSM. The current state s of a running FSM $N_j \in N_1, \ldots, N_m$ is described as $N_i(s)$.

We have undefined reception, $\exists s \in S$, if $N(s) \wedge q_N =< i_1, \ldots, i_n >, i_1, \ldots, i_n \in I \Rightarrow \nexists (s, i_1, o_1, s') \in \lambda_e$.

We have a deadlock, A FSM N_1 is in a deadlock if $\exists s \in S$ with $s \neq s_0 \wedge q_{N_1} =<>$. A system P of interacting FSM $N_1, \ldots, N_m$ is in a deadlock, if $N_1, \ldots, N_m$ is in a deadlock.

Definition 9 *Unspecified reception for two* FSM*:*

$\exists(s_1, < i_1^1, \ldots, i_{n_1}^1 >, s_2, < i_1^2, \ldots, i_{n_2}^2 >) \in S_E :$

$\neg\Big(\Big(\exists s_1' \in S_1, \exists o_1 \in OI_{1,2} : \big((s_1, < i_1^1, \ldots, i_{n_1}^1 >, s_2, < i_1^2, \ldots, i_{n_2}^2 >), i_1, o_1, (s_1', < i_2^1, \ldots, i_{n_1}^1 >, s_2, < i_1^2, \ldots, i_{n_2}^2 >)\big) \in \lambda_E\Big)$

$\vee\Big(\exists s_2' \in S_2, \exists o_2 \in OI_{2,1} : \big((s_1, < i_1^1, \ldots, i_{n_1}^1 >, s_2, < i_1^2, \ldots, i_{n_2}^2 >), i_1, o_1, (s_1, < i_1^1, \ldots, i_{n_1}^1 >, s_2, < i_2^2, \ldots, i_{n_2}^2 >)\big) \in \lambda_E\Big)\Big)$

with:

S_E *set of possible states, i.e. there is a path from* s_0 *to every* $s \in S_E$

$S_E := \{s \in S | \exists(e_1, \ldots, e_n) \in P_N : \exists s' \in S, i \in I, o \in O : e_n = (s', i, o, s)\}$

and P_N *is the set of all possible path in* N*:*

$P_N = \{(e_1, \ldots, e_n) \in \lambda^* | (\exists i \in I, \exists o \in O, \exists s' \in S : e_1 = (s_0, i, o, s')) \wedge (\forall k \in 1, \ldots, k-1 : e_i[k] = e_i + 1[1])\}$

Definition 10 *Deadlock for two* FSM*:*

$\exists(s_1, <>, s_2, <>) \in S_E :$ *(*$s_1 \neq s_0 \in S_1 \wedge s_2 \neq s_0 \in S_2$*)*

$\nexists\big((s_1, <>, s_2, <>), i_1, o_1, (s_1', <>, s_2, < i_1^2 >)\big) \vee \big((s_1, <>, s_2, <>), i_2, o_2, (s_1, < i_1^1 >, s_2, <>)\big)$

The tools check for unspecified reception and deadlock using the reachability analysis.

Fault model

In protocol engineering, the usual fault model assumes that the implementation I is a mutant of its specification S. The mutant may have a different output on a transition, which is called an output fault. It can also have transitions that end in different states than anticipated, which is called a transfer fault.

Most of the methods to test the components derive an input sequence from its specification and apply it to the implementation. If the given output does not match the output of the implementation, a fault is detected. Depending on the method, one can get more or less detailed information about the nature of the fault and where it occurred. In the following, we assume that a test method detects all output- and transfer-faults of a given component specified as FSM. Those test methods can also be applied to the composed system as long as the composition itself can be represented as an FSM. However the state set of the global state machine tends to become very large, and as such it can consume a huge amount of computational resources. It also leads to lengthy test cases.

One of the goals of the C-Method is to avoid these drawbacks. We want to consider the structural aspect of the composition and thus introduce two new fault categories:

The *component fault*, where the implementation of a component does not satisfy its specification.

The *composition fault*, where the glue code does not satisfy its specification in the given context.

The C-Method is a classical test method. We derive test cases from the interacting components and apply those sequences to the implementation. Beforehand all the components have been tested successfully. So if an error is detected it can be categorized as a composition fault.

In order to produce a minimal test suite that can be executed realistically we make the following assumptions:

- Whenever I_1 and I_2 are both in their initial states, the glue code is in a determined state w.r.t. I_1 and I_2.
- The behaviour of the glue code is deterministic w.r.t. I_1 and I_2.
- If the glue code interacts with other components, this has no effect on its behaviour towards I_1 and I_2.
- The glue code is not creating messages for I_1 or I_2.

The first assumption limits the maximum length of test suites to the set of all initial tours, i.e., paths that start and end in the initial state. All assumptions together ensure that a finite number of test cases is sufficient.

Notations

Definition 11 *Let atc_1 and atc_2 be augmented test cases (sequence of transitions) of deterministic $csFSMs$ N_1 and N_2 that communicate via a common channel ch with sets $OI_{1,2}$ and $OI_{2,1}$ of internal signals. The concurrent composition of atc_1 and atc_2, denoted $atc_1||atc_2$, is one path $atc_{1,2}$ of the tree obtained by sequencing the test elements in atc_1 and atc_2 according to the following ordering constraints:*

- *The order of the test elements of atc_1 and atc_2 is preserved.*
- *A test element of atc_1 (atc_2) triggered by an internal signal is constrained by the corresponding test element in atc_2 (atc_1) that produces this internal signal.*
- *The order of the output is preserved.*

Definition 12 *The* concurrent composition *of two augmented test cases is called* complete*, iff all their test elements are included, and the input queues of the corresponding $csFSM$ will be empty after their execution. Otherwise, it is called* incomplete.

Example

A step by step analysis of the original C-Method from Table 3.

1. First of all, the single components need to be tested. (Step 1)
2. Remove unnecessary transitions, i.e. the error transitions. (Step 2.1)
3. The augmented test cases derived from the components form the basis to construct a test suite that verifies the correctness of the composition. We construct an *initial tour coverage tree* (table 2) of each component. From the initial tour coverage tree we take the maximal paths (each path from the root to a leaf node) which are the augmented test suites (Step 2.2)
4. Then the augmented test suites are reduced to the minimal necessary elements. (Step 2.3, 2.4, 2.5)
5. Now we construct the concurrent augmented test suite following Def. 11. (Step 2.6)
6. Check the composition criteria 9, i.e. and whether if there is a matching test case for every test case in the augmented test suite. (Step 2.7)
7. Reduce the augmented test cases. (Step 2.8)
8. Finally apply the resulted test cases to the implementation and check the resulting output. (Step 2.9)

Step 1: Test the implementations I_1 and I_2 of components N_1 and N_2.

Step 1.1: Select a test method (e.g., DS [], UIOv [10], WP []).

Step 1.2: Derive the test suites for N_1 and N_2.

Step 1.3: Execute the tests. If all tests are successful, continue with Step 2. If not, correct the faults and repeat Step 1.

Step 2: Test the implementation of the concurrent composition of N_1 and N_2.

Step 2.1: Remove all transitions of N_1 and N_2 that yield an error output. These transitions have already been tested during component testing, and need not be tested again.

Step 2.2: Build the initial tour coverage trees for N_1 and N_2, and determine all maximal path, i.e., all paths that start at the root node and end at a leaf node, constituting augmented test suites ATS_1 and ATS_2.

Step 2.3: From the augmented test suites ATS_1 (ATS_2), remove all internally triggered test cases, i.e., those test cases that are triggered by N_2 (N_1).

Step 2.4: From the augmented test cases, remove all local tours, i.e., (sub)sequences of test case elements that (1) start and end in the same state, and (2) contain only external inputs and outputs. They have already been checked during component testing, and need not be tested again.

Step 2.5: Remove the maximum suffix that does not contain an interaction with the other component. These test elements have been checked already.

Step 2.6: For each test case $\text{ATC}_{1,j}$ of the augmented test suite ATS_1 after Step 2.5, find an augmented test case $\text{ATC}_{2,j}$ of N_2 from Step 2.2 such that $\text{ATC}_{1,j} \parallel \text{ATC}_{2,j}$ is complete, and determine $\text{ATC}_{1,2,j} = \text{ATC}_{1,j} \parallel \text{ATC}_{2,j}$, yielding the concurrent augmented test suite $\text{ATS}_{1,2}$. Analogously for each test case $\text{ATC}_{2,j}$ of ATS_2.

Step 2.7: Based on ATS_1, ATS_2, and $\text{ATS}_{1,2}$, check whether N_1 and N_2 meet the composition criteria CC_1 and CC_2, i.e., whether for each test case ATS_1 (ATS_2), there is a matching test case of N_2 (N_1). Yes: Continue with Step 2.8; No: Stop.

Step 2.8: For each test case in $\text{ATS}_{1,2}$: Merge adjacent test case elements in case where (1) the internal output of the first matches the internal input of the second, and (2) the output is the only signal in the queue after being sent. Replace internal inputs and outputs by "-", and remove test case elements "-/-".

Step 2.9: Execute the test.

Table 3: The original C-Method

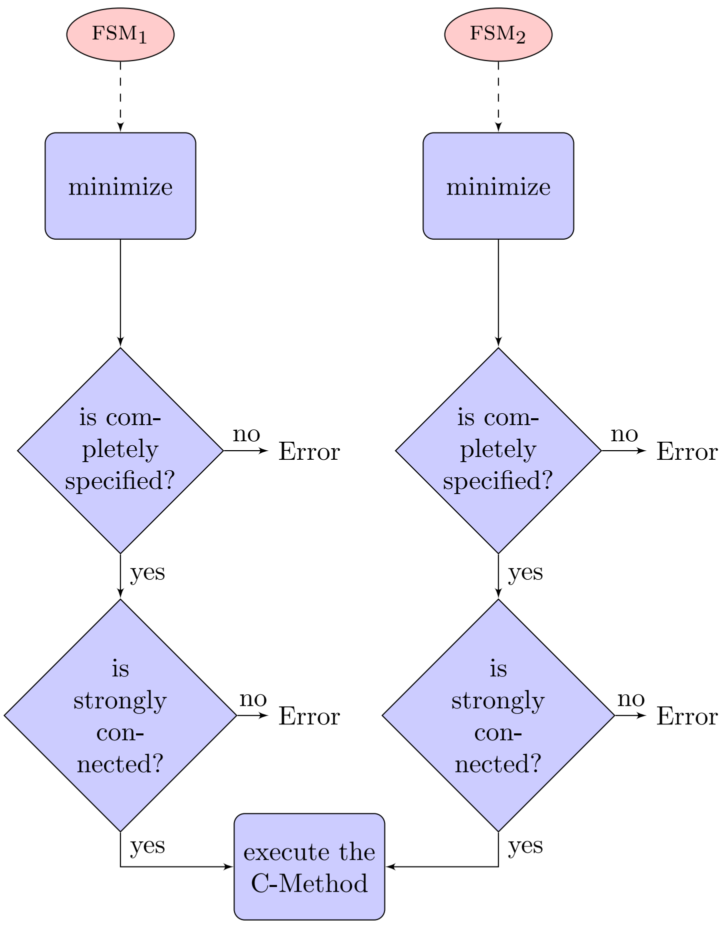


Figure 4: The control flow for two FSMs

3 Tool Support

We implemented a tool-set that generates the output of the modified C-Method. The tool is implemented in Python. The different tools take *graph*s and/or FSMs as arguments. The *graph*s and FSMs are stored as XML in files. The output is either on the console, as graph-pictures or as a LaTeX-file, depending on the given task.

The tool-set can be found on the CD in the appendix.

Structure

In order to execute the C-Method we must check the pre-conditions and then execute the C-Method as described in Table 4. This is illustrated in Figure 4. All data (*graph*s and FSMs) is passed as xml-files and parsed into objects. A simplified UML-representation is shown in Figure 5.

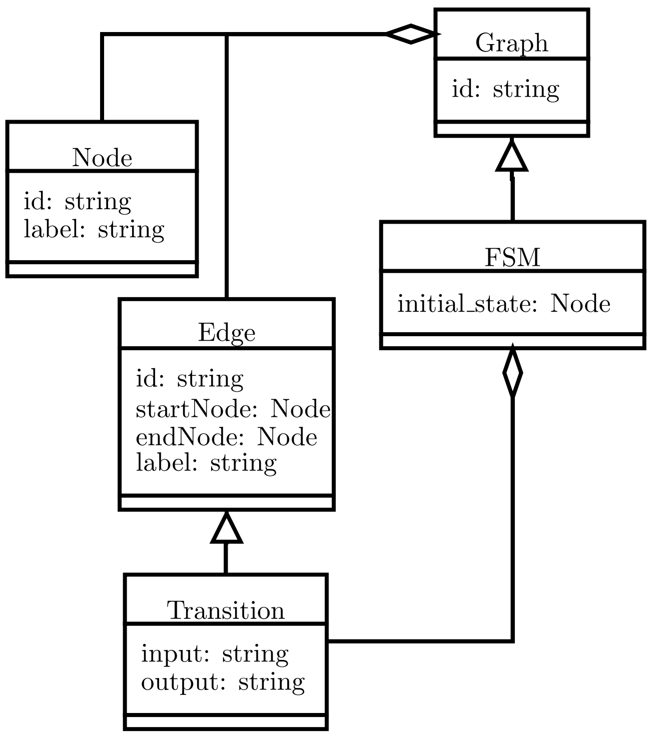


Figure 5: Simplified UML-representation of the FSM-Class.

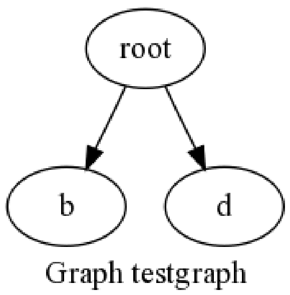
File Format

Both *graph*s and FSMs can be read from xml-files.

A *graph* consists of *node*s and *edge*s. It has a unique *id* and an optional *root_node*. A *node* has a unique *id* and an optional *label* which is used to label the *node* in the output-methods. If no *label*-field is present the node is labeled with its *id*. An *edge* has an optional unique *id* field, if no id is given, a unique id is automatically generated. The *edge* has a *start*- and an *end*-node, where the value is the unique *id* of the respective *node*. In order to add a *node* to the *graph*, the *start*- and *end*-node must exist.

Here is an example of a graph:

```
<?xml version="1.0"?>
<graph id="graph" root_node="a">
    <node id="a" label="root"/>
    <node id="b"/>
    <node id="d" label="d"/>
    <edge id="0" start="a" end="d"/>
    <edge start="a" end="b"/>
</graph>
```



The xml-data is saved in a file called *testgraph.xml* and was converted to a graphical representation with the *xml2graphviz.py* script, which is part of the tool-set.

A FSM consists of *state*s and *transition*s. It has a unique *id* and an *initial_state*. The *initial_state* is the same as the *root_node* in of the *graph*. The *state* is equal to the *node* in the *graph*. A *transition* extends an *edge* by an *input* and an *output* which contain the input/output-signals. If - is used as an *input* it marks a spontaneous transition, if it is used as *output* it is a transition with no outgoing signal. Error-transitions are characterised by e as outgoing signal.

The xml-representation for the FSM$_{Res}$ (Fig. 6b) used in the Case Studies is given below:

```
<?xml version="1.0"?>
<fsm id="res" initial_state="s1">
    <state id="s1" label="idle"/>
    <state id="s2" label="wait4ICONrsp"/>
    <state id="s3" label="connected"/>
    <state id="s4" label="AK2send"/>
    <transition start="s1" end="s2" input="CR" output="ICONind" id="1"/>
    <transition start="s2" end="s3" input="ICONrsp" output="CC" id="2"/>
    <transition start="s3" end="s4" input="DT" output="IDATind" id="3"/>
    <transition start="s4" end="s3" input="-" output="AK" id="4"/>
    <transition start="s3" end="s1" input="IDISreq" output="DR" id="5"/>
    <transition start="s2" end="s1" input="IDISreq" output="DR" id="6"/>
    <transition start="s1" end="s1" input="DT" output="-" id="7"/>
</fsm>
```

Algorithms

Output

The generated LaTeX-output can be seen in the Case Studies section 4.

4 Case studies

4.1 InRes

4.1.1 Applying the Original C-Method

The C-Method has been applied to the initiator responder protocol in [8]. It is defined by the communicating finite state machines FSM$_{In}$ (Fig. 6a) and FSM$_{Res}$ (Fig. 6b). During the development of the tool set we found that the InRes example given in [8] (Section 4) is incomplete.

(i) The initial tour coverage tree of the initiator (FSM$_I$) is missing the path from the second *connected*-node, via *wait4AK* to *idle*.

(ii) In Step 2.6 a transition ($DT/-$) appears in ATS$_{R,c}$ that is not part of it in Step 2.2.

(iii) Furthermore there are some possible matches missing in Step 2.6, which results in an incomplete ATS and minor changes for the concurrent ATS.

Those are inconsistencies in the paper [8] itself, but under further investigation are partially meaningful.

4.1.2 Corrections and Modifications

In order to overcome the differences we expand the C-Method. The modifications are as following:

In order to be able to find all matches for the ATS$_{1,2}$ it is necessary to consider not only every combination of test cases from ATS$_1$ and ATS$_2$. There is the possibility that the second FSM is already back in its initial state but the first has not yet finished. To consume the possible output and to match the test cases we must consider all possible concatenations of test cases from ATS$_2$ with itself. The concatenation represents also an initial tour and is henceforth a valid composition. So we obtain in some special cases, like in (ii) a match consisting of two test cases.

The modified C-Method can be found in Table 4.

4.1.3 Applying the Modified C-Method

Step 2.1: The transitions containing an error-output have been omitted in this example to improve readability. Therefore we begin in Step 2.2.

Step 2.2: Read the augmented test suites from the initial tour coverage trees (Fig. 7), where ATS$_I$ is the augmented test suite from FSM$_{In}$ and ATS$_R$ is the augmented test suite from FSM$_{Res}$.

- ATS$_I$ $= \{$ATS$_{I,1}$, ATS$_{I,2}$, ATS$_{I,3}$, ATS$_{I,4}$, ATS$_{I,5}\}$

  ATS$_{I,1}$ $= \{$ICONreq/CR, DR/IDISind$\}$

Step 1: Test the implementations I_1 and I_2 of components N_1 and N_2.

Step 1.1: Select a test method (e.g., DS [], UIOv [10], WP []).

Step 1.2: Derive the test suites for N_1 and N_2.

Step 1.3: Execute the tests. If all tests are successful, continue with Step 2. If not, correct the faults and repeat Step 1.

Step 2: Test the implementation of the concurrent composition of N_1 and N_2.

Step 2.1: Remove all transitions of N_1 and N_2 that yield an error output. These transitions have already been tested during component testing, and need not be tested again.

Step 2.2: Build the initial tour coverage trees for N_1 and N_2, and determine all maximal path, i.e., all paths that start at the root node and end at a leaf node, constituting augmented test suites $\textsc{ats}_1$ and $\textsc{ats}_2$.

Step *2.3'*: From the augmented test cases, remove all local tours, i.e., (sub)sequences of test case elements that (1) start and end in the same state, and (2) contain only external inputs and outputs. They have already been checked during component testing, and need not be tested again.

Step *2.4'*: From the augmented test suites $\textsc{ats}_1$ ($\textsc{ats}_2$), remove all internally triggered test cases, i.e., those test cases that are triggered by N_2 (N_1).

Step 2.5: Remove the maximum suffix that does not contain an interaction with the other component. These test elements have been checked already.

Step 2.6 *For each test case* $\textsc{atc}_{1,i}$ *of the augmented test suite* $\textsc{ats}_1$ *after Step 2.5, find a concatenation of augmented test cases* $\textsc{atc}_{2,j_1,\ldots,j_n}$ *of* N_2 *from Step 2.3' such that* $\textsc{atc}_{1,i} \parallel \textsc{atc}_{2,j_1,\ldots,j_n}$ *is complete, and determine* $\textsc{atc}_{1,2,i,j_1,\ldots,j_n} = \textsc{atc}_{1,i} \parallel \textsc{atc}_{2,j_1,\ldots,j_n}$*, yielding the concurrent augmented test suite* $\textsc{ats}_{1,2}$*. Analogously for each test case* $\textsc{atc}_{2,i}$ *of* $\textsc{ats}_2$*.*

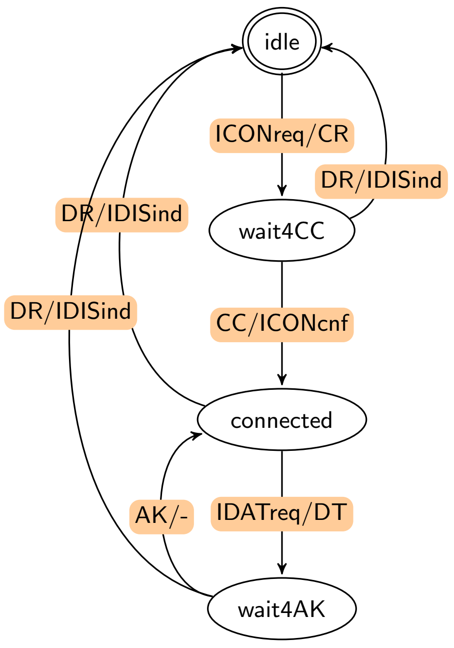
Step 2.7: Based on $\textsc{ats}_1$, $\textsc{ats}_2$, and $\textsc{ats}_{1,2}$, check whether N_1 and N_2 meet the composition criteria CC_1 and CC_2, i.e., whether for each test case $\textsc{ats}_1$ ($\textsc{ats}_2$), there is a matching test case of N_2 (N_1). Yes: Continue with Step 2.8; No: Stop.

Step 2.8: For each test case in $\textsc{ats}_{1,2}$: Merge adjacent test case elements in case where (1) the internal output of the first matches the internal input of the second, and (2) the output is the only signal in the queue after being sent. Replace internal inputs and outputs by "-", and remove test case elements "-/-".

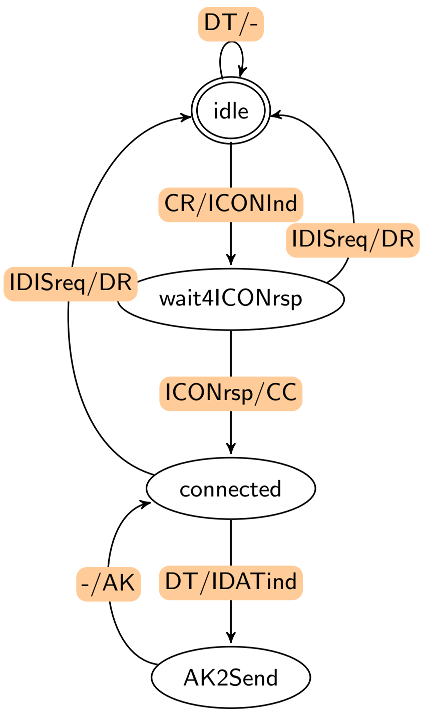
Step 2.9: Execute the test.

Table 4: The modified C-Method

(a) FSM_{In}



(b) FSM_{Res}



(c) Communication sets

$$X_{In,env} = \{\text{ICONcnf, IDISind}\}$$

$$X_{env,In} = \{\text{ICONreq, IDATreq}\}$$

$$X_{In,Res} = \{\text{CR, DT}\}$$

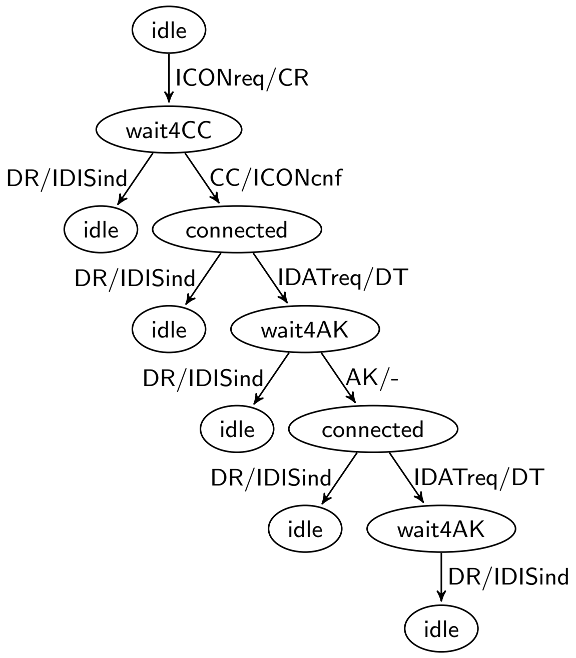
$$X_{Res,In} = \{\text{CC, DR, AK}\}$$

$$X_{Res,env} = \{\text{ICONind, IDATind}\}$$

$$X_{env,Res} = \{\text{ICONrsp, IDISreq}\}$$

Figure 6: FSM_{In} and corresponding automaton FSM_{Res}

(a) Initial tour coverage tree for FSM_{In}



(b) Initial tour coverage tree for FSM_{Res}

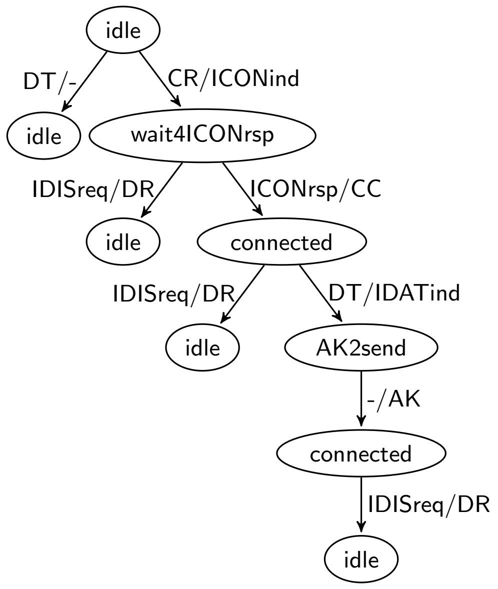


Figure 7: Initial tour coverage trees for FSM_{In} and FSM_{Res}

$\text{ATS}_{I,2} = \{$ICONreq/CR, CC/ICONcnf, DR/IDISind$\}$

$\text{ATS}_{I,3} = \{$ICONreq/CR, CC/ICONcnf, IDATreq/DT, DR/IDISind$\}$

$\text{ATS}_{I,4} = \{$ICONreq/CR, CC/ICONcnf, IDATreq/DT, AK/-, DR/IDISind$\}$

$\text{ATS}_{I,5} = \{$ICONreq/CR, CC/ICONcnf, IDATreq/DT, AK/-, IDATreq/DT, DR/IDISind$\}$

- $\text{ATS}_R = \text{ATS}_{R,a}, \text{ATS}_{R,b}, \text{ATS}_{R,c}, \text{ATS}_{R,d}$

  $\text{ATS}_{R,a} = \{$DT/-$\}$

  $\text{ATS}_{R,b} = \{$CR/ICONind, IDISreq/DR$\}$

  $\text{ATS}_{R,c} = \{$CR/ICONind, ICONrsp/CC, IDISreq/DR$\}$

  $\text{ATS}_{R,d} = \{$CR/ICONind, ICONrsp/CC, DT/IDATind, -/AK, IDISreq/DR$\}$

Step 2.3: After removing all internally triggered test cases ATS_I and ATS_R are as following:

- $\text{ATS}'_I = \{\text{ATS}_{I,1}, \text{ATS}_{I,2}, \text{ATS}_{I,3}, \text{ATS}_{I,4}, \text{ATS}_{I,5}\}$
- $\text{ATS}'_R = \{\}$

Step 2.4: There are no local tours in the InRes example. ATS'_I and ATS'_R remain unchanged.

Step 2.5: There are no suffixes with local interaction only in the InRes example. ATS'_I and ATS'_R remain unchanged.

Step 2.6: Construct the concurrent augmented test suite.

- $\text{ATS}_{I,R} = \{\text{ATS}'_{I,1} \parallel \text{ATS}_{R,b}, \text{ATS}'_{I,2} \parallel \text{ATS}_{R,c}, \text{ATS}'_{I,3} \parallel \text{ATS}_{R,c \cap a}, \text{ATS}'_{I,4} \parallel \text{ATS}_{R,d}, \text{ATS}'_{I,5} \parallel \text{ATS}_{R,d \cap a}\}$
- $\text{ATS}_{R,I} = \{\}$

with:

$\text{ATS}'_{I,1} \parallel \text{ATS}_{R,b} = \{$ICONreq/CR $\bullet$ CR/ICONind $\bullet$ IDISreq/DR $\bullet$ DR/IDISind$\}$

$\text{ATS}'_{I,2} \parallel \text{ATS}_{R,c} = \{$

ICONreq/CR $\bullet$ CR/ICONind $\bullet$ ICONrsp/CC $\bullet$ CC/ICONcnf $\bullet$ IDISreq/DR $\bullet$ DR/IDISind,

ICONreq/CR $\bullet$ CR/ICONind $\bullet$ ICONrsp/CC $\bullet$ IDISreq/DR $\bullet$ CC/ICONcnf $\bullet$ DR/IDISind

$\}$

$\text{ATS}'_{I,3} \parallel \text{ATS}_{R,c \cap a} = \{$

ICONreq/CR $\bullet$ CR/ICONind $\bullet$ ICONrsp/CC $\bullet$ CC/ICONcnf $\bullet$ IDATreq/DT $\bullet$ DT/- $\bullet$ IDISreq/DR $\bullet$ DR/IDISind, ICONreq/CR • CR/ICONind • ICONrsp/CC • CC/ICONcnf • IDATreq/DT • IDISreq/DR • DT/- • DR/IDISind,

ICONreq/CR • CR/ICONind • ICONrsp/CC • CC/ICONcnf • IDISreq/DR • IDATreq/DT • DT/- • DR/IDISind,

ICONreq/CR • CR/ICONind • ICONrsp/CC • CC/ICONcnf • IDISreq/DR • IDATreq/DT • DR/IDISind • DT/-,

ICONreq/CR • CR/ICONind • ICONrsp/CC • IDISreq/DR • CC/ICONcnf • IDATreq/DT • DT/- • DR/IDISind,

ICONreq/CR • CR/ICONind • ICONrsp/CC • IDISreq/DR • CC/ICONcnf • IDATreq/DT • DR/IDISind • DT/-

}

$\text{ATS}'_{I,4} \parallel \text{ATS}_{R,d} = \{$

ICONreq/CR • CR/ICONind • ICONrsp/CC • CC/ICONcnf • IDATreq/DT • DT/IDATind • -/AK • AK/- • IDISreq/DR • DR/IDISind,

ICONreq/CR • CR/ICONind • ICONrsp/CC • CC/ICONcnf • IDATreq/DT • DT/IDATind • -/AK • IDISreq/DR • AK/- • DR/IDISind

}

$\text{ATS}'_{I,5} \parallel \text{ATS}_{R,d\cap a} = \{$

ICONreq/CR • CR/ICONind • ICONrsp/CC • CC/ICONcnf • IDATreq/DT • DT/IDATind • -/AK • AK/- • IDATreq/DT • IDISreq/DR • DR/IDISind • DT/-,

ICONreq/CR • CR/ICONind • ICONrsp/CC • CC/ICONcnf • IDATreq/DT • DT/IDATind • -/AK • AK/- • IDATreq/DT • IDISreq/DR • DT/- • DR/IDISind,

ICONreq/CR • CR/ICONind • ICONrsp/CC • CC/ICONcnf • IDATreq/DT • DT/IDATind • -/AK • AK/- • IDISreq/DR • IDATreq/DT • DT/- • DR/IDISind,

ICONreq/CR • CR/ICONind • ICONrsp/CC • CC/ICONcnf • IDATreq/DT • DT/IDATind • -/AK • AK/- • IDISreq/DR • IDATreq/DT • DR/IDISind • DT/-,

ICONreq/CR • CR/ICONind • ICONrsp/CC • CC/ICONcnf • IDATreq/DT • DT/IDATind • -/AK • IDISreq/DR • IDATreq/DT • DT/- • DR/IDISind,

ICONreq/CR • CR/ICONind • ICONrsp/CC • CC/ICONcnf • IDATreq/DT • DT/IDATind • -/AK • IDISreq/DR • IDATreq/DT • DR/IDISind • DT/-

}

Step 2.7: For each test case of ATS_I, there is a test case of ATS'_R such that their concurrent composition is complete. This trivially holds for ATS_R which is empty.

Step 2.8: Merged test case elements and replace inputs/outputs:

- $\text{ATS}_{I,R} = \{\text{ATS}'_{I,1} \parallel \text{ATS}_{R,b}, \text{ATS}'_{I,2} \parallel \text{ATS}_{R,c}, \text{ATS}'_{I,3} \parallel \text{ATS}_{R,c\cap a}, \text{ATS}'_{I,4} \parallel \text{ATS}_{R,d}, \text{ATS}'_{I,5} \parallel \text{ATS}_{R,d\cap a}\}$

- $\text{ATS}_{R,I} = \{\}$

where:

$\text{ATS}'_{I,1} \parallel \text{ATS}_{R,b} = \{$ICONreq/ICONind • IDISreq/IDISind$\}$

$\text{ATS}'_{I,2} \parallel \text{ATS}_{R,c} = \{$

ICONreq/ICONind • ICONrsp/ICONcnf • IDISreq/IDISind,

ICONreq/ICONind • ICONrsp/- • IDISreq/- • -/ICONcnf • -/IDISind

$\}$

$\text{ATS}'_{I,3} \parallel \text{ATS}_{R,c\cap a} = \{$

ICONreq/ICONind • ICONrsp/ICONcnf • IDATreq/- • IDISreq/IDISind,

ICONreq/ICONind • ICONrsp/ICONcnf • IDATreq/- • IDISreq/- • -/IDISind,

ICONreq/ICONind • ICONrsp/ICONcnf • IDISreq/- • IDATreq/- • -/IDISind,

ICONreq/ICONind • ICONrsp/ICONcnf • IDISreq/- • IDATreq/- • -/IDISind,

ICONreq/ICONind • ICONrsp/- • IDISreq/- • -/ICONcnf • IDATreq/- • -/IDISind,

ICONreq/ICONind • ICONrsp/- • IDISreq/- • -/ICONcnf • IDATreq/- • -/IDISind

$\}$

$\text{ATS}'_{I,4} \parallel \text{ATS}_{R,d} = \{$

ICONreq/ICONind • ICONrsp/ICONcnf • IDATreq/IDATind • IDISreq/IDISind,

ICONreq/ICONind • ICONrsp/ICONcnf • IDATreq/IDATind • IDISreq/- • -/IDISind

$\}$

$\text{ATS}'_{I,5} \parallel \text{ATS}_{R,d\cap a} = \{$

ICONreq/ICONind • ICONrsp/ICONcnf • IDATreq/IDATind • IDATreq/- • IDISreq/IDISind,

ICONreq/ICONind • ICONrsp/ICONcnf • IDATreq/IDATind • IDATreq/- • IDISreq/- • -/IDISind,

ICONreq/ICONind • ICONrsp/ICONcnf • IDATreq/IDATind • IDISreq/- • IDATreq/- • -/IDISind,

ICONreq/ICONind • ICONrsp/ICONcnf • IDATreq/IDATind • IDISreq/- • IDATreq/- • -/IDISind,

ICONreq/ICONind • ICONrsp/ICONcnf • IDATreq/IDATind • IDISreq/- • IDATreq/- • -/IDISind,

ICONreq/ICONind • ICONrsp/ICONcnf • IDATreq/IDATind • IDISreq/- • IDATreq/- • -/IDISind

}

Step 2.9: Execute the test with:

- $\text{ATS}_{I,R} = \big\{\text{ATS}'_{I,1} \parallel \text{ATS}_{R,b}, \text{ATS}'_{I,2} \parallel \text{ATS}_{R,c}, \text{ATS}'_{I,3} \parallel \text{ATS}_{R,c\cap a}, \text{ATS}'_{I,4} \parallel \text{ATS}_{R,d}, \text{ATS}'_{I,5} \parallel \text{ATS}_{R,d\cap a}\big\}$
- $\text{ATS}_{R,I} = \{\}$

where:

$\text{ATS}'_{I,1} \parallel \text{ATS}_{R,b} = \big\{$ICONreq/ICONind • IDISreq/IDISind$\big\}$

$\text{ATS}'_{I,2} \parallel \text{ATS}_{R,c} = \{$

ICONreq/ICONind • ICONrsp/ICONcnf • IDISreq/IDISind,

ICONreq/ICONind • ICONrsp/- • IDISreq/- • -/ICONcnf • -/IDISind

}

$\text{ATS}'_{I,3} \parallel \text{ATS}_{R,c\cap a} = \{$

ICONreq/ICONind • ICONrsp/ICONcnf • IDATreq/- • IDISreq/IDISind,

ICONreq/ICONind • ICONrsp/ICONcnf • IDATreq/- • IDISreq/- • -/IDISind,

ICONreq/ICONind • ICONrsp/ICONcnf • IDISreq/- • IDATreq/- • -/IDISind,

ICONreq/ICONind • ICONrsp/- • IDISreq/- • -/ICONcnf • IDATreq/- • -/IDISind

}

$\text{ATS}'_{I,4} \parallel \text{ATS}_{R,d} = \{$

ICONreq/ICONind • ICONrsp/ICONcnf • IDATreq/IDATind • IDISreq/IDISind,

ICONreq/ICONind • ICONrsp/ICONcnf • IDATreq/IDATind • IDISreq/- • -/IDISind

}

$\text{ATS}'_{I,5} \parallel \text{ATS}_{R,d\cap a} = \{$

ICONreq/ICONind • ICONrsp/ICONcnf • IDATreq/IDATind • IDATreq/- • IDISreq/IDISind,

ICONreq/ICONind • ICONrsp/ICONcnf • IDATreq/IDATind • IDATreq/- • IDISreq/- • -/IDISind,

ICONreq/ICONind • ICONrsp/ICONcnf • IDATreq/IDATind • IDISreq/- • IDATreq/- • -/IDISind,

}

4.2 Simple TCP

4.2.1 Applying the Modified C-Method

We apply the C-Method to a simplified version of the 'transmission control protocol' (TCP), which only consists of the connect and disconnect procedure. All data transfer is done in the established ($ESTAB$) state and can be represented as a separate FSM.

The TCP specification [6] contains an automaton which we transfer to the FSM representation of the protocol. With some minor modifications it is represented in Fig. 8. The names of some states and signals were changed to better fit the terminology so far used. Furthermore the different ACKs were separated into categories. This was necessary, in order to match the ACK to be corresponding either to a previously sent SYN or FIN and are now called $ACKS$ and $ACKF$. The same is done in the TCP itself, by using different bits to indicate the proper ACK, but is unfortunately not represented in their automaton representation.

The connect procedure is the part from the initial state ($CLOSED$) to the $ESTAB$ state where a connection has been established. The disconnect procedure is the part from the $ESTAB$ state back to the initial state. Apart from this regular behaviour the connection can be denied, aborted or terminated. This is done with the *Close* transitions, if the FSM is not in the $ESTAB$ state, and is henceforth called irregular behaviour.

In order to consume some leftover signals we introduce the term of a *consumer loop*, which is a transition from a state to the same state. The input of the transition is the signal to be consumed and it has no output.

4.2.2 Problems and Solutions

In order to execute the C-Method successfully there must not be any unspecified reception or deadlocks. In the FSM however there is a deadlock which occurs when one FSM decides to *Close* the connection in its initial phase. The partner FSM can process the signals until it gets stuck with an empty queue in the SYN_RCVD state. In order to avoid the deadlock, the closing FSM must notify it's partner. We add a RST signal to the *Close* transition and a corresponding receiving $RST/-$ transition from SYN_RCVD to the initial state like shown in [7] on page 293. In the automaton given in [7] are other modification, like timeouts which we don't need since our medium is perfect an we do not have a time notion in the C-Method and its FSMs. Nevertheless a timeout can be modeled as a signal coming from the environment. This model is useful if it is necessary to apply the timeout as a part of an ATC. As a consequence we need no further changes concerning the timeout already present in the FSM (Fig. 9).

Apart from the deadlock there exist several scenarios in which some signals are left in the queue after the connection has been closed. This is the case if the connection has been closed irregularly, with all the *Close* events before the FSM was in the $ESTAB$ state. In some cases this would not be a problem, since most implementations or modeling languages just delete the first signal in the

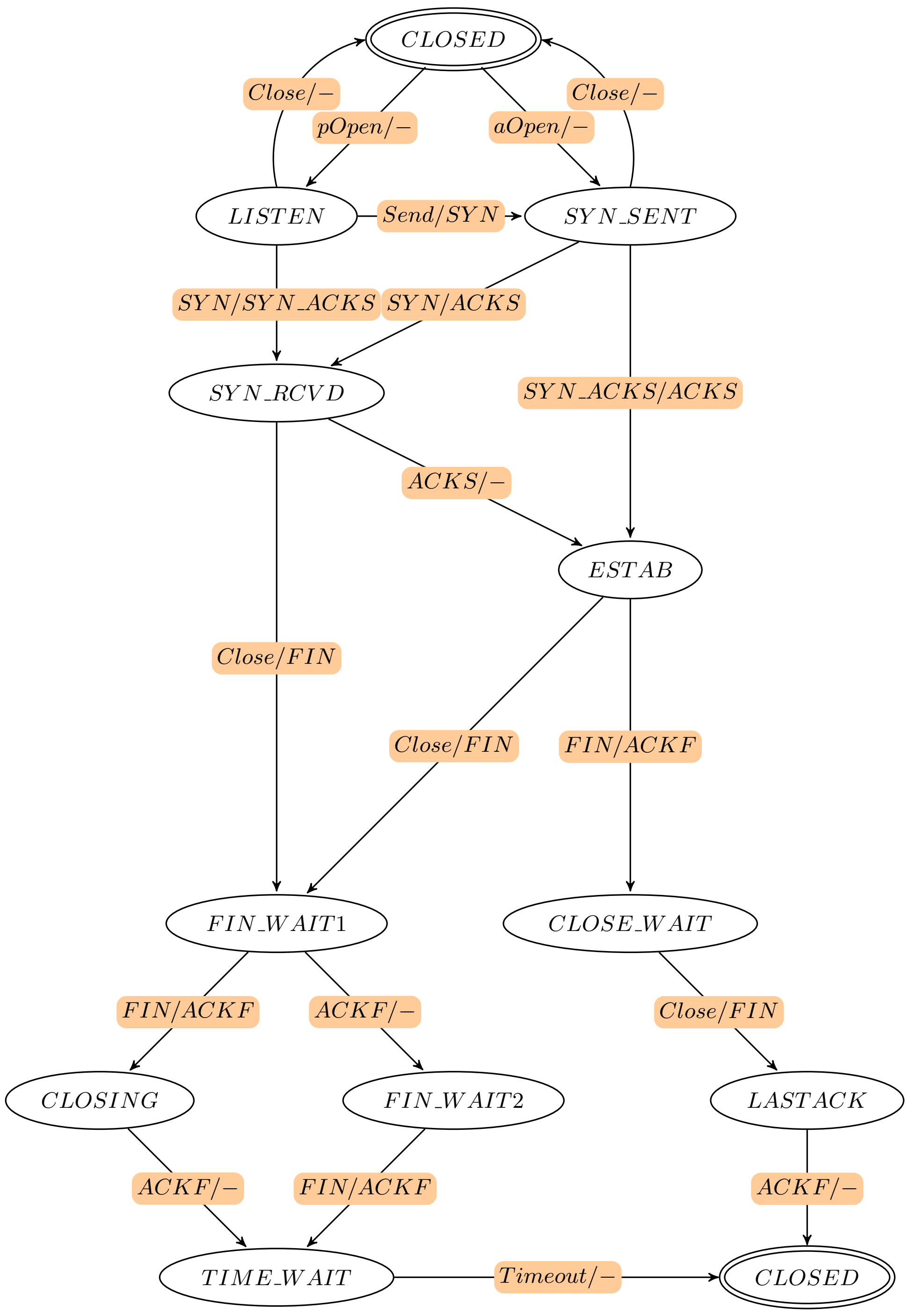


Figure 8: $\text{FSM}_{SimpleTCP}$ from the specification.

queue if it cannot be consumed in this state. It is an undefined reception, and in the C-Method every reception must be defined. Furthermore the FSM should be able to synchronize at least in the initial state, so we add some consumer loops to the initial states which get rid of possible signals after the connection has been closed. The consumer loop in the initial state consumes SYN, $ACKS$, SYN_ACKS and RST signals. The consumer loop shows the same behaviour as many implementations or modeling languages do natively. They just delete the first signal in the queue if it cannot be consumed in this state.

Another unspecified reception occurs when one FSM closes the connection while being in the SYN_RCVD state. There is still the $ACKS$-signal left in the queue which has to be consumed. Normally this would be indicated by a consumer loop in the FIN_WAIT1 state. It is the same as deleting the signal from the queue in the implementation. This is legitimate and is done every time when coming directly from SYN_RCVD.

However, introducing a consumer loop would give an extra subtree when constructing the initial tour coverage tree. So we have generated the same ATC, once where an $ACKS$ is consumed and once without its consumption. This is due, because formally the $ACKS$ consumer loop is only a possibility. So even if an $ACKS$ is consumed in FIN_WAIT1 every time we close the connection in the SYN_RCVD state and the implementation with the consumer loop is correct, the C-Method generates additional ATCs without the consumption for which it is impossible to find a matching test case since none can exist. To avoid this we must introduce an additional state to consume the $ACKS$ from the queue. This does not modify the protocol behaviour. We introduce the $CONSUME\_ACKS$ state with the $ACKS/-$ transition leading to FIN_WAIT1 instead of a $ACKS/-$ consumer loop on FIN_WAIT1. All the modifications can bee seen in Fig. 9.

Since it is necessary to introduce new states in order for the C-Method to work correctly, whereas the simple modification (consumer loop) just work fine for the FSM and the implementation, can be considered a drawback of the C-Method. It is the cost which comes with the initial tour coverage tree, which relies on stricter preconditions. It might be another source of errors by introducing a special FSM to generate test cases or by rendering the FSM more complicated.

Using this approach a valid test caes from the itct would be $\text{ATS}_1 = \big\{\text{Connect/SYN}, \text{Close/RST}\big\}$. But it is impossible to find a matching test case even tough the FSM fullfills all the requirements. It is impossible to find a matching test case because the partner FSM consumes the SYN signal and responds to it by sending another signal. This signal can be consumed by the running FSM and both FSM would end in a synchronized state whereas for the test case I still have this one signal in the queue and hence no matching test caes can be found.

A solution would be to ignore messages left in the queue of the first FSM, assuming it is in its initial state and that the messages cannot change the state. It is the same problem we encountered in the InRes example for which we introduced the concatenation of test cases. This time it is not possible since we would have to find matching test case concatenations for all possible combinations of test cases. This is a problemand should be investiganted further.

In order to be able to execute the C-Method we modifiy the FSM by removing the

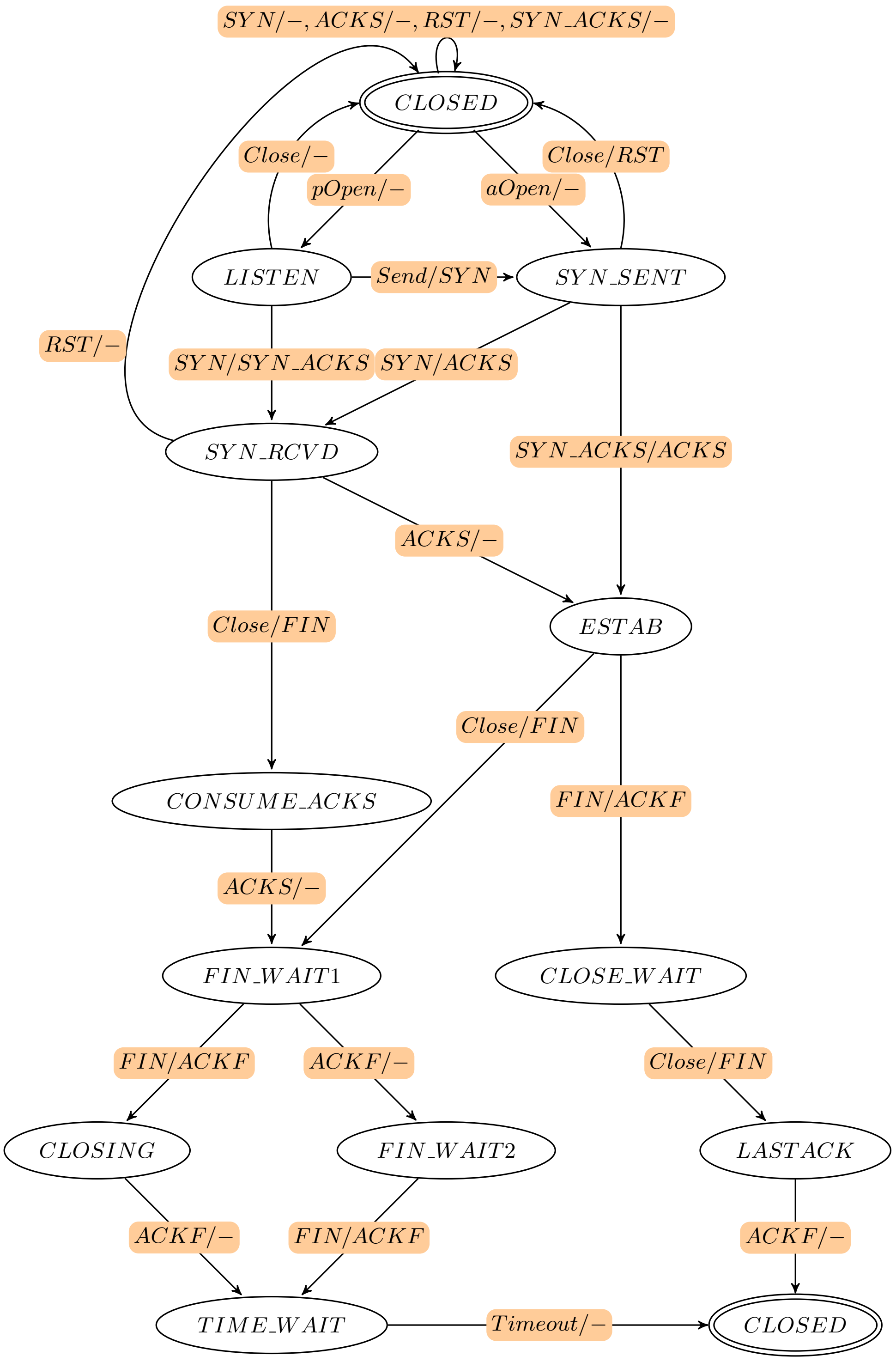


Figure 9: Modified $\text{FSM}_{SimpleTCP}$.

$Close/RST$ transition. As a sideeffect the $RST/-$ transition can be removed as well as all the consumer loops in the state $CLOSED$. The modified simple TCP example can then be executed by the C-Method.

4.2.3 Applying the Modified C-Method

We apply the C-Method to the modified simple TCP example.

Step 2.1: Remove all transitions of N_1 and N_2 that yield an error output. These transitions have already been tested during component testing, and need not to be tested again.

Step 2.2: Build the initial tour coverage trees for N_1 ($\text{FSM}_{RedSimpleTCP}$) and N_2 ($\text{FSM}_{RedSimpleTCP}$), and determine all maximal paths, i.e., all path that start at the root node and end at a leaf node, constituting augmented test suites $\text{ATS}_{RedSimpleTCP}$ and $\text{ATS}_{RedSimpleTCP}$.

- $\text{ATS}_{RedSimpleTCP} = \{\text{ATS}_{RedSimpleTCP,1}, \text{ATS}_{RedSimpleTCP,2}, \text{ATS}_{RedSimpleTCP,3}, \text{ATS}_{RedSimpleTCP,4}, \text{ATS}_{RedSimpleTCP,5}, \text{ATS}_{RedSimpleTCP,6}, \text{ATS}_{RedSimpleTCP,7}, \text{ATS}_{RedSimpleTCP,8}, \text{ATS}_{RedSimpleTCP,9}, \text{ATS}_{RedSimpleTCP,10}, \text{ATS}_{RedSimpleTCP,11}, \text{ATS}_{RedSimpleTCP,12}, \text{ATS}_{RedSimpleTCP,13}\}$

  $\text{ATS}_{RedSimpleTCP,1} = \{$SYN/SYN_ACKS, Close/FIN, ACKS/-, FIN/ACKF, ACKF/- $\}$

  $\text{ATS}_{RedSimpleTCP,2} = \{$SYN/SYN_ACKS, Close/FIN, ACKS/-, ACKF/-, FIN/ACKF$\}$

  $\text{ATS}_{RedSimpleTCP,3} = \{$SYN/SYN_ACKS, ACKS/-, Close/FIN, FIN/ACKF, ACKF/- $\}$

  $\text{ATS}_{RedSimpleTCP,4} = \{$SYN/SYN_ACKS, ACKS/-, Close/FIN, ACKF/-, FIN/ACKF$\}$

  $\text{ATS}_{RedSimpleTCP,5} = \{$SYN/SYN_ACKS, ACKS/-, FIN/ACKF, Close/FIN, ACKF/- $\}$

  $\text{ATS}_{RedSimpleTCP,6} = \{$aOpen/SYN, SYN/ACKS, ACKS/-, Close/FIN, FIN/ACKF, ACKF/-$\}$

  $\text{ATS}_{RedSimpleTCP,7} = \{$aOpen/SYN, SYN/ACKS, ACKS/-, Close/FIN, ACKF/- , FIN/ACKF$\}$

  $\text{ATS}_{RedSimpleTCP,8} = \{$aOpen/SYN, SYN/ACKS, ACKS/-, FIN/ACKF, Close/FIN, ACKF/-$\}$

  $\text{ATS}_{RedSimpleTCP,9} = \{$aOpen/SYN, SYN/ACKS, Close/FIN, ACKS/-, FIN/ACKF, ACKF/-$\}$

  $\text{ATS}_{RedSimpleTCP,10} = \{$aOpen/SYN, SYN/ACKS, Close/FIN, ACKS/-, ACKF/- , FIN/ACKF$\}$

  $\text{ATS}_{RedSimpleTCP,11} = \{$aOpen/SYN, SYN_ACKS/ACKS, Close/FIN, FIN/ACKF, ACKF/-$\}$

  $\text{ATS}_{RedSimpleTCP,12} = \{$aOpen/SYN, SYN_ACKS/ACKS, Close/FIN, ACKF/-, FIN/ACKF$\}$

  $\text{ATS}_{RedSimpleTCP,13} = \{$aOpen/SYN, SYN_ACKS/ACKS, FIN/ACKF, Close/FIN, ACKF/-$\}$

Step 2.3': From the augmented test cases, remove all local tours, i.e., (sub)sequences of test case elements that (1) start and end in the same state, and (2) contain only external inputs and outputs. They have already been checked during component testing, and need not be tested again.

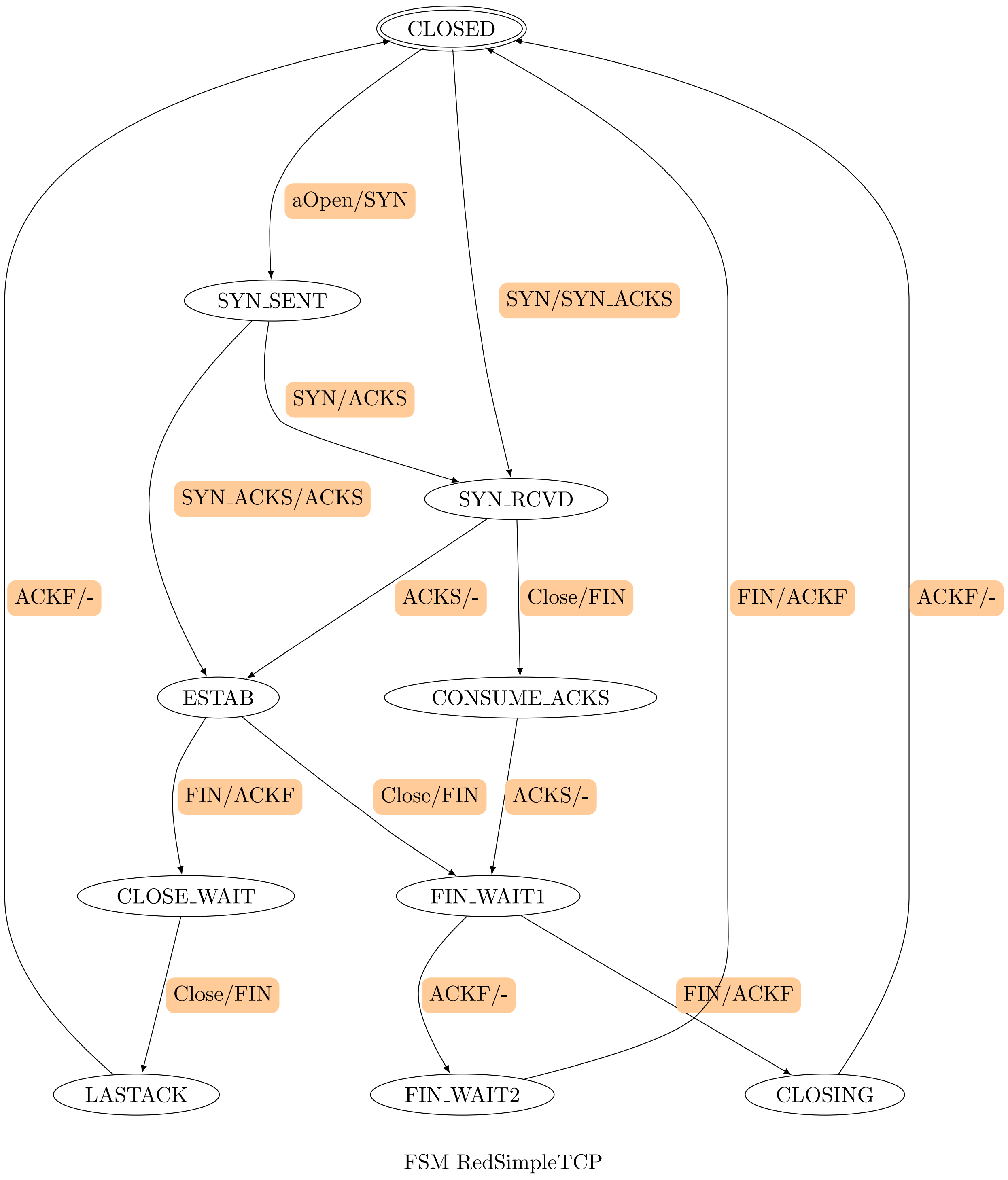


Figure 10: $\text{FSM}_{RedSimpleTCP}$

- $\text{ATS}_{RedSimpleTCP} = \big\{\text{ATS}_{RedSimpleTCP,1},\ \text{ATS}_{RedSimpleTCP,2},\ \text{ATS}_{RedSimpleTCP,3},\ \text{ATS}_{RedSimpleTCP,4},\ \text{ATS}_{RedSimpleTCP,5},\ \text{ATS}_{RedSimpleTCP,6},\ \text{ATS}_{RedSimpleTCP,7},\ \text{ATS}_{RedSimpleTCP,8},\ \text{ATS}_{RedSimpleTCP,9},\ \text{ATS}_{RedSimpleTCP,10},\ \text{ATS}_{RedSimpleTCP,11},\ \text{ATS}_{RedSimpleTCP,12},\ \text{ATS}_{RedSimpleTCP,13}\big\}$

  $\text{ATS}_{RedSimpleTCP,1} = \big\{$SYN/SYN_ACKS, Close/FIN, ACKS/-, FIN/ACKF, ACKF/- $\big\}$

  $\text{ATS}_{RedSimpleTCP,2} = \big\{$SYN/SYN_ACKS, Close/FIN, ACKS/-, ACKF/-, FIN/ACKF$\big\}$

  $\text{ATS}_{RedSimpleTCP,3} = \big\{$SYN/SYN_ACKS, ACKS/-, Close/FIN, FIN/ACKF, ACKF/- $\big\}$

  $\text{ATS}_{RedSimpleTCP,4} = \big\{$SYN/SYN_ACKS, ACKS/-, Close/FIN, ACKF/-, FIN/ACKF$\big\}$

  $\text{ATS}_{RedSimpleTCP,5} = \big\{$SYN/SYN_ACKS, ACKS/-, FIN/ACKF, Close/FIN, ACKF/- $\big\}$

  $\text{ATS}_{RedSimpleTCP,6} = \big\{$aOpen/SYN, SYN/ACKS, ACKS/-, Close/FIN, FIN/ACKF, ACKF/-$\big\}$

  $\text{ATS}_{RedSimpleTCP,7} = \big\{$aOpen/SYN, SYN/ACKS, ACKS/-, Close/FIN, ACKF/- , FIN/ACKF$\big\}$

  $\text{ATS}_{RedSimpleTCP,8} = \big\{$aOpen/SYN, SYN/ACKS, ACKS/-, FIN/ACKF, Close/FIN, ACKF/-$\big\}$

  $\text{ATS}_{RedSimpleTCP,9} = \big\{$aOpen/SYN, SYN/ACKS, Close/FIN, ACKS/-, FIN/ACKF, ACKF/-$\big\}$

  $\text{ATS}_{RedSimpleTCP,10} = \big\{$aOpen/SYN, SYN/ACKS, Close/FIN, ACKS/-, ACKF/- , FIN/ACKF$\big\}$

  $\text{ATS}_{RedSimpleTCP,11} = \big\{$aOpen/SYN, SYN_ACKS/ACKS, Close/FIN, FIN/ACKF, ACKF/-$\big\}$

  $\text{ATS}_{RedSimpleTCP,12} = \big\{$aOpen/SYN, SYN_ACKS/ACKS, Close/FIN, ACKF/-, FIN/ACKF$\big\}$

  $\text{ATS}_{RedSimpleTCP,13} = \big\{$aOpen/SYN, SYN_ACKS/ACKS, FIN/ACKF, Close/FIN, ACKF/-$\big\}$

Step 2.4': From the augmented test suites $\text{ATS}_{RedSimpleTCP}$ ($\text{ATS}_{RedSimpleTCP}$), remove all internally triggered test cases, i.e., those test cases that are triggered by N_1 and N_2.

- $\text{ATS}'_{RedSimpleTCP} = \big\{\text{ATS}'_{RedSimpleTCP,1},\ \text{ATS}'_{RedSimpleTCP,2},\ \text{ATS}'_{RedSimpleTCP,3},\ \text{ATS}'_{RedSimpleTCP,4},\ \text{ATS}'_{RedSimpleTCP,5},\ \text{ATS}'_{RedSimpleTCP,6},\ \text{ATS}'_{RedSimpleTCP,7},\ \text{ATS}'_{RedSimpleTCP,8}\big\}$

  $\text{ATS}'_{RedSimpleTCP,1} = \big\{$aOpen/SYN, SYN/ACKS, ACKS/-, Close/FIN, FIN/ACKF, ACKF/-$\big\}$

  $\text{ATS}'_{RedSimpleTCP,2} = \big\{$aOpen/SYN, SYN/ACKS, ACKS/-, Close/FIN, ACKF/- , FIN/ACKF$\big\}$

  $\text{ATS}'_{RedSimpleTCP,3} = \big\{$aOpen/SYN, SYN/ACKS, ACKS/-, FIN/ACKF, Close/FIN, ACKF/-$\big\}$

  $\text{ATS}'_{RedSimpleTCP,4} = \big\{$aOpen/SYN, SYN/ACKS, Close/FIN, ACKS/-, FIN/ACKF, ACKF/-$\big\}$

  $\text{ATS}'_{RedSimpleTCP,5} = \big\{$aOpen/SYN, SYN/ACKS, Close/FIN, ACKS/-, ACKF/- , FIN/ACKF$\big\}$

$\text{ATS}'_{RedSimpleTCP,6} = \{$aOpen/SYN, SYN_ACKS/ACKS, Close/FIN, FIN/ACKF, ACKF/-$\}$

$\text{ATS}'_{RedSimpleTCP,7} = \{$aOpen/SYN, SYN_ACKS/ACKS, Close/FIN, ACKF/-, FIN/ACKF$\}$

$\text{ATS}'_{RedSimpleTCP,8} = \{$aOpen/SYN, SYN_ACKS/ACKS, FIN/ACKF, Close/FIN, ACKF/-$\}$

Step 2.5: Remove the maximum suffix that does not contain an interaction with the other component. These test elements have been checked already.

- $\text{ATS}'_{RedSimpleTCP} = \{\text{ATS}'_{RedSimpleTCP,1}, \text{ATS}'_{RedSimpleTCP,2}, \text{ATS}'_{RedSimpleTCP,3}, \text{ATS}'_{RedSimpleTCP,4}, \text{ATS}'_{RedSimpleTCP,5}, \text{ATS}'_{RedSimpleTCP,6}, \text{ATS}'_{RedSimpleTCP,7}, \text{ATS}'_{RedSimpleTCP,8}\}$

  $\text{ATS}'_{RedSimpleTCP,1} = \{$aOpen/SYN, SYN/ACKS, ACKS/-, Close/FIN, FIN/ACKF, ACKF/-$\}$

  $\text{ATS}'_{RedSimpleTCP,2} = \{$aOpen/SYN, SYN/ACKS, ACKS/-, Close/FIN, ACKF/- , FIN/ACKF$\}$

  $\text{ATS}'_{RedSimpleTCP,3} = \{$aOpen/SYN, SYN/ACKS, ACKS/-, FIN/ACKF, Close/FIN, ACKF/-$\}$

  $\text{ATS}'_{RedSimpleTCP,4} = \{$aOpen/SYN, SYN/ACKS, Close/FIN, ACKS/-, FIN/ACKF, ACKF/-$\}$

  $\text{ATS}'_{RedSimpleTCP,5} = \{$aOpen/SYN, SYN/ACKS, Close/FIN, ACKS/-, ACKF/- , FIN/ACKF$\}$

  $\text{ATS}'_{RedSimpleTCP,6} = \{$aOpen/SYN, SYN_ACKS/ACKS, Close/FIN, FIN/ACKF, ACKF/-$\}$

  $\text{ATS}'_{RedSimpleTCP,7} = \{$aOpen/SYN, SYN_ACKS/ACKS, Close/FIN, ACKF/-, FIN/ACKF$\}$

  $\text{ATS}'_{RedSimpleTCP,8} = \{$aOpen/SYN, SYN_ACKS/ACKS, FIN/ACKF, Close/FIN, ACKF/-$\}$

Step 2.6: For each test case $\text{ATC}_{1,i}$ of the augmented test suite ATS_1 after Step 2.5, find a concatenation of augmented test cases $\text{ATC}_{2,j_1,\ldots,j_n}$ of N_2 from Step 2.3' such that $\text{ATC}_{1,i} \parallel \text{ATC}_{2,j_1,\ldots,j_n}$ is complete, and determine $\text{ATC}_{1,2,i,j_1,\ldots,j_n} = \text{ATC}_{1,i} \parallel \text{ATC}_{2,j_1,\ldots,j_n}$, yielding the concurrent augmented test suite $\text{ATS}_{1,2}$. Analogously for each test case $\text{ATC}_{2,i}$ of ATS_2. The signal-index marks the affiliation to the corresponding FSM.

- $\text{ATS}_{RedSimpleTCP,RedSimpleTCP} = \{\text{ATS}'_{RedSimpleTCP,1} \parallel \text{ATS}_{RedSimpleTCP,6}, \text{ATS}'_{RedSimpleTCP,2} \parallel \text{ATS}_{RedSimpleTCP,8}, \text{ATS}'_{RedSimpleTCP,3} \parallel \text{ATS}_{RedSimpleTCP,7}, \text{ATS}'_{RedSimpleTCP,4} \parallel \text{ATS}_{RedSimpleTCP,6}, \text{ATS}'_{RedSimpleTCP,5} \parallel \text{ATS}_{RedSimpleTCP,8}, \text{ATS}'_{RedSimpleTCP,6} \parallel \text{ATS}_{RedSimpleTCP,1}, \text{ATS}'_{RedSimpleTCP,7} \parallel \text{ATS}_{RedSimpleTCP,5}, \text{ATS}'_{RedSimpleTCP,8} \parallel \text{ATS}_{RedSimpleTCP,2}\}$

with:

Since some concatenated test cases are very long, we choose to omit them in this place in order to enhance readability. The complete RedSimpleTCP example can be found in the appendix.

$\mathrm{ATS}'_{RedSimpleTCP,1} \parallel \mathrm{ATS}_{RedSimpleTCP,6} = \{\ldots\}$

$\mathrm{ATS}'_{RedSimpleTCP,2} \parallel \mathrm{ATS}_{RedSimpleTCP,8} = \{\ldots\}$

$\mathrm{ATS}'_{RedSimpleTCP,3} \parallel \mathrm{ATS}_{RedSimpleTCP,7} = \{\ldots\}$

$\mathrm{ATS}'_{RedSimpleTCP,4} \parallel \mathrm{ATS}_{RedSimpleTCP,6} = \{\ldots\}$

$\mathrm{ATS}'_{RedSimpleTCP,5} \parallel \mathrm{ATS}_{RedSimpleTCP,8} = \{\ldots\}$

$\mathrm{ATS}'_{RedSimpleTCP,6} \parallel \mathrm{ATS}_{RedSimpleTCP,1} = \{\ldots\}$

$\mathrm{ATS}'_{RedSimpleTCP,7} \parallel \mathrm{ATS}_{RedSimpleTCP,5} = \{$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$

$\}$

$\mathrm{ATS}'_{RedSimpleTCP,8} \parallel \mathrm{ATS}_{RedSimpleTCP,2} = \{$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKS_2/-_2 \bullet ACKF_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_2/FIN_2 \bullet ACKS_2/-_2 \bullet FIN_1/ACKF_1 \bullet Close_1/FIN_1 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_2/FIN_2 \bullet ACKS_2/-_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet Close_2/FIN_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet FIN_1/ACKF_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet Close_2/FIN_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet FIN_1/ACKF_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet Close_2/FIN_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet FIN_1/ACKF_1 \bullet ACKS_2/-_2 \bullet ACKF_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet Close_2/FIN_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet ACKS_2/-_2 \bullet FIN_1/ACKF_1 \bullet Close_1/FIN_1 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet Close_2/FIN_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet ACKS_2/-_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$

}

Step 2.7: Based on ATS$_1$, ATS$_2$, and ATS$_{1,2}$, check whether N_1 and N_2 meet the composition criteria CC_1 and CC_2, i.e., whether for each test case ATS$_1$ (ATS$_2$), there is a matching test case of N_2 (N_1). Yes: Continue with Step 2.8; No: Stop.

Unspecified reception: False

Deadlocks: False

Step 2.8: For each test case in ATS$_{1,2}$: Merge adjacent test case elements in case where (1) the internal output of the first matches the internal input of the second, and (2) the output is the only signal in the queue after being sent. Replace internal inputs and outputs by "-", and remove test case elements "-/-". (Remaining - are made anonymous (common index 0) in order to remove duplicates resulting from the merge.)

- $\text{ATS}_{RedSimpleTCP,RedSimpleTCP} = \{\text{ATS}'_{RedSimpleTCP,1} \parallel \text{ATS}_{RedSimpleTCP,6}, \text{ATS}'_{RedSimpleTCP,2} \parallel \text{ATS}_{RedSimpleTCP,8}, \text{ATS}'_{RedSimpleTCP,3} \parallel \text{ATS}_{RedSimpleTCP,7}, \text{ATS}'_{RedSimpleTCP,4} \parallel \text{ATS}_{RedSimpleTCP,6}, \text{ATS}'_{RedSimpleTCP,5} \parallel \text{ATS}_{RedSimpleTCP,8}, \text{ATS}'_{RedSimpleTCP,6} \parallel \text{ATS}_{RedSimpleTCP,1}, \text{ATS}'_{RedSimpleTCP,7} \parallel \text{ATS}_{RedSimpleTCP,5}, \text{ATS}'_{RedSimpleTCP,8} \parallel \text{ATS}_{RedSimpleTCP,2}\}$

with:

$\text{ATS}'_{RedSimpleTCP,1} \parallel \text{ATS}_{RedSimpleTCP,6} = \{$

$aOpen_1/-_0 \bullet aOpen_2/-_0 \bullet Close_1/-_0 \bullet Close_2/-_0$,

$aOpen_1/-_0 \bullet aOpen_2/-_0 \bullet Close_2/-_0 \bullet Close_1/-_0$,

$aOpen_2/-_0 \bullet aOpen_1/-_0 \bullet Close_1/-_0 \bullet Close_2/-_0$,

$aOpen_2/-_0 \bullet aOpen_1/-_0 \bullet Close_2/-_0 \bullet Close_1/-_0$

}

$\text{ATS}'_{RedSimpleTCP,2} \parallel \text{ATS}_{RedSimpleTCP,8} = \{$

$aOpen_1/-_0 \bullet aOpen_2/-_0 \bullet Close_1/-_0 \bullet Close_2/-_0$,

$aOpen_2/-_0 \bullet aOpen_1/-_0 \bullet Close_1/-_0 \bullet Close_2/-_0$

}

$\text{ATS}'_{RedSimpleTCP,3} \parallel \text{ATS}_{RedSimpleTCP,7} = \{$

$aOpen_1/-_0 \bullet aOpen_2/-_0 \bullet Close_2/-_0 \bullet Close_1/-_0$,

$aOpen_2/-_0 \bullet aOpen_1/-_0 \bullet Close_2/-_0 \bullet Close_1/-_0$

}

$\text{ATS}'_{RedSimpleTCP,4} \parallel \text{ATS}_{RedSimpleTCP,6} = \{$

$aOpen_1/-_0 \bullet aOpen_2/-_0 \bullet Close_1/-_0 \bullet Close_2/-_0$,

$aOpen_1/-_0 \bullet aOpen_2/-_0 \bullet Close_2/-_0 \bullet Close_1/-_0,$

$aOpen_2/-_0 \bullet aOpen_1/-_0 \bullet Close_1/-_0 \bullet Close_2/-_0,$

$aOpen_2/-_0 \bullet aOpen_1/-_0 \bullet Close_2/-_0 \bullet Close_1/-_0$

$\}$

$\text{ATS}'_{RedSimpleTCP,5} \parallel \text{ATS}_{RedSimpleTCP,8} = \{$

$aOpen_1/-_0 \bullet aOpen_2/-_0 \bullet Close_1/-_0 \bullet Close_2/-_0,$

$aOpen_2/-_0 \bullet aOpen_1/-_0 \bullet Close_1/-_0 \bullet Close_2/-_0$

$\}$

$\text{ATS}'_{RedSimpleTCP,6} \parallel \text{ATS}_{RedSimpleTCP,1} = \{$

$aOpen_1/-_0 \bullet Close_1/-_0 \bullet Close_2/-_0,$

$aOpen_1/-_0 \bullet Close_2/-_0 \bullet Close_1/-_0$

$\}$

$\text{ATS}'_{RedSimpleTCP,7} \parallel \text{ATS}_{RedSimpleTCP,5} = \{$

$aOpen_1/-_0 \bullet Close_1/-_0 \bullet Close_2/-_0$

$\}$

$\text{ATS}'_{RedSimpleTCP,8} \parallel \text{ATS}_{RedSimpleTCP,2} = \{$

$aOpen_1/-_0 \bullet Close_2/-_0 \bullet Close_1/-_0$

$\}$

Step 2.9: Execute the test.

5 Related Work

This chapter gives an overview on different testing methods related to compositional testing. Most of the methods use another approach as the C-Method. Therefore this chapter is not intended so evaluate other methods in respect to the C-Method, but rather give an overview and situate the C-Method in the field of compostional testing.

5.1 Interoperability Testing

Interoperability Testing, as described in [[3], [2]] ,checks if the interaction of several implementations is conform to their specification. Especially considering more passive components which until now are often reported as failures due to the lack of interaction. If the component is an observer only, this behaviour has been modeled in the specification and should not lead to false test verdicts.

In order to execute the test each component is described as a black box with an upper and lower interface. The test system monitors both interfaces of every participating component. In the C-Method we consider the components as correct and defined a fault model for the glue code. The fault model in interoperability testing assumes the components to be faulty and the medium to be correct.

Interoperability testing as well as the C-Method check if a certain service is given. The difference is that in the C-Method it is done in the specification versus monitoring the implementation in interoperability testing.

5.2 Partial Specification

The approach in [5] assumes that a system specification is not necessary in order to formulate test cases. It combines a series of elementary test cases with a set of requirements. The elementary test cases are extracted from the specification or the source code. The set of requirements is used to formulate a general desirable behaviour. The requirements can be some expected behaviour at runtime or a restriction on input possibilities by the user. They can also state some security rules in order for the system to be conform to security outlines. The requirements are expressed by logical formulas build on a set of predicates.

The tool set described in [5] uses the basic test cases and combines their results to check the validity of the overall logical formula. This step is automated and the authors believe that some logical formalisms can be used to express the requirements. It can be seen as a monitor additionally to the extracted test cases.

5.3 Research related to the C-Method

In [1] the author considers component based real-time systems (CBRTS). Timed Input/Output Automata (TIOA) are introduced in order to describe the CBRTS. The developed testing framework uses an adaptation of the *initial tour coverage tree* to derive compositional test cases using a symbolic approach.

In another paper [4] it is stated that future research will be conducted with the fault model from the C-Method. It is also intended to research testing methodologies for component based systems taking into consideration that the components have already been tested; Which is the same context the C-Method is placed in.

6 Conclusion

By implementing the tool set for the C-Method we encountered some problems in the C-method [8]. Their solution let us to the modified C-Method.

By conducting the case study we can also say that it is very time consuming to modify or extend existing protocols to be deadlock and unspecified reception free. Most protocols specified as FSM do often neglect unspecified reception by assuming an implicit signal consumption in some states.

The InRes protocol communicates very intensively with its environment and therefore we get a lot of incoming and outgoing signals to consider and elaborate test cases are formed by the C-Method. On the other side we saw the reduced simple TCP example, which has a rich inter-component communication and a reduced interaction with the environment. As we can see in the resulting test cases, this leads to very short and unsatisfying results. Those test cases are not enough to ensure a proper verification of the components. Especially considering the numerous and lengthy concurrent augmented test cases which result from symmetrical protocols.

The completeness of the resulting test cases from the C-Method rely on the degree of communication between the participating FSM and the environment. This is a general problem which results from the design of the C-Method itself.

References

[1] Rachid Bouaziz and Ismail Berrada. Testing component-based real time systems. *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008. SNPD '08. Ninth ACIS International Conference on*, pages 888–894, Aug. 2008.

[2] Alexandra Desmoulin and César Viho. Formalizing interoperability testing: Quiescence management and test generation. In Farn Wang, editor, *FORTE*, volume 3731 of *Lecture Notes in Computer Science*, pages 533–537. Springer, 2005.

[3] Alexandra Desmoulin and César Viho. Quiescence management improves interoperability testing. In Ferhat Khendek and Rachida Dssouli, editors, *TestCom*, volume 3502 of *Lecture Notes in Computer Science*, pages 365–379. Springer, 2005.

[4] Alain Faivre, Christophe Gaston, and Pascale Le Gall. Symbolic model based testing for component oriented systems. In Petrenko et al. [9], pages 90–106.

[5] Yliès Falcone, Jean-Claude Fernandez, Laurent Mounier, and Jean-Luc Richier. A compositional testing framework driven by partial specifications. In Petrenko et al. [9], pages 107–122.

[6] Prepared for. Transmission Control Protocol: DARPA Internet Program Protocol Specification, Sep. 1981. See also RFC0793.

[7] Behrouz A. Forouzan. *TCP/IP Protocol Suite, Third Edition.* McGraw-Hill, 1221 Avenue of the Americas, New York, NY 10020, USA, 2006.

[8] Reinhard Gotzhein and Ferhat Khendek. Compositional Testing of Communication Systems. In M.Ü. Uyar, A.Y. Duale, and M.A. Fecko, editors, *LNCS: Testing of Communicating Systems*, volume 3964, pages 227 – 244. Springer, 2006.

[9] Alexandre Petrenko, Margus Veanes, Jan Tretmans, and Wolfgang Grieskamp, editors. *Testing of Software and Communicating Systems, 19th IFIP TC6/WG6.1 International Conference, TestCom 2007, 7th International Workshop, FATES 2007, Tallinn, Estonia, June 26-29, 2007, Proceedings*, volume 4581 of *Lecture Notes in Computer Science*. Springer, 2007.

[10] S. T. Vuong, W. W. L. Chan, and M. R. Ito. The UIOv-Method for Protocol Test Sequence Generation. In *2nd International Workshop on Protocol Test Systems*, pages 203 – 225, 1989.

APPENDIX A

Tool-set on CD

CD

APPENDIX B

Tool output for the InRes example

C-Method applied to FSM_{in} and FSM_{res}

Automatically generated by c_method.py

December 5, 2008

Step 2.1: Remove all transitions of N_1 and N_2 that yield an error output. These transitions have already been tested during component testing, and need not to be tested again.

Step 2.2: Build the initial tour coverage trees for N_1 (FSM_{in}) and N_2 (FSM_{res}), and determine all maximal paths, i.e., all path that start at the root node and end at a leaf node, constituting augmented test suites ATS_{in} and ATS_{res}.

- $\text{ATS}_{in} = \{\text{ATS}_{in,1}, \text{ATS}_{in,2}, \text{ATS}_{in,3}, \text{ATS}_{in,4}, \text{ATS}_{in,5}\}$

  $\text{ATS}_{in,1} = \{$ICONreq/CR, CC/ICONcnf, DR/IDISind$\}$

  $\text{ATS}_{in,2} = \{$ICONreq/CR, CC/ICONcnf, IDATreq/DT, DR/IDISind$\}$

  $\text{ATS}_{in,3} = \{$ICONreq/CR, CC/ICONcnf, IDATreq/DT, AK/-, IDATreq/DT, DR/IDISind$\}$

  $\text{ATS}_{in,4} = \{$ICONreq/CR, CC/ICONcnf, IDATreq/DT, AK/-, DR/IDISind$\}$

  $\text{ATS}_{in,5} = \{$ICONreq/CR, DR/IDISind$\}$

- $\text{ATS}_{res} = \{\text{ATS}_{res,1}, \text{ATS}_{res,2}, \text{ATS}_{res,3}, \text{ATS}_{res,4}\}$

  $\text{ATS}_{res,1} = \{$CR/ICONind, ICONrsp/CC, DT/IDATind, -/AK, IDISreq/DR$\}$

  $\text{ATS}_{res,2} = \{$CR/ICONind, ICONrsp/CC, IDISreq/DR$\}$

  $\text{ATS}_{res,3} = \{$CR/ICONind, IDISreq/DR$\}$

  $\text{ATS}_{res,4} = \{$DT/-$\}$

Step 2.3': From the augmented test cases, remove all local tours, i.e., (sub)sequences of test case elements that (1) start and end in the same state, and (2) contain only external inputs and outputs. They have already been checked during component testing, and need not be tested again.

- $\text{ATS}_{in} = \{\text{ATS}_{in,1}, \text{ATS}_{in,2}, \text{ATS}_{in,3}, \text{ATS}_{in,4}, \text{ATS}_{in,5}\}$

  $\text{ATS}_{in,1} = \{$ICONreq/CR, CC/ICONcnf, DR/IDISind$\}$

  $\text{ATS}_{in,2} = \{$ICONreq/CR, CC/ICONcnf, IDATreq/DT, DR/IDISind$\}$

  $\text{ATS}_{in,3} = \{$ICONreq/CR, CC/ICONcnf, IDATreq/DT, AK/-, IDATreq/DT, DR/IDISind$\}$

  $\text{ATS}_{in,4} = \{$ICONreq/CR, CC/ICONcnf, IDATreq/DT, AK/-, DR/IDISind$\}$

  $\text{ATS}_{in,5} = \{$ICONreq/CR, DR/IDISind$\}$

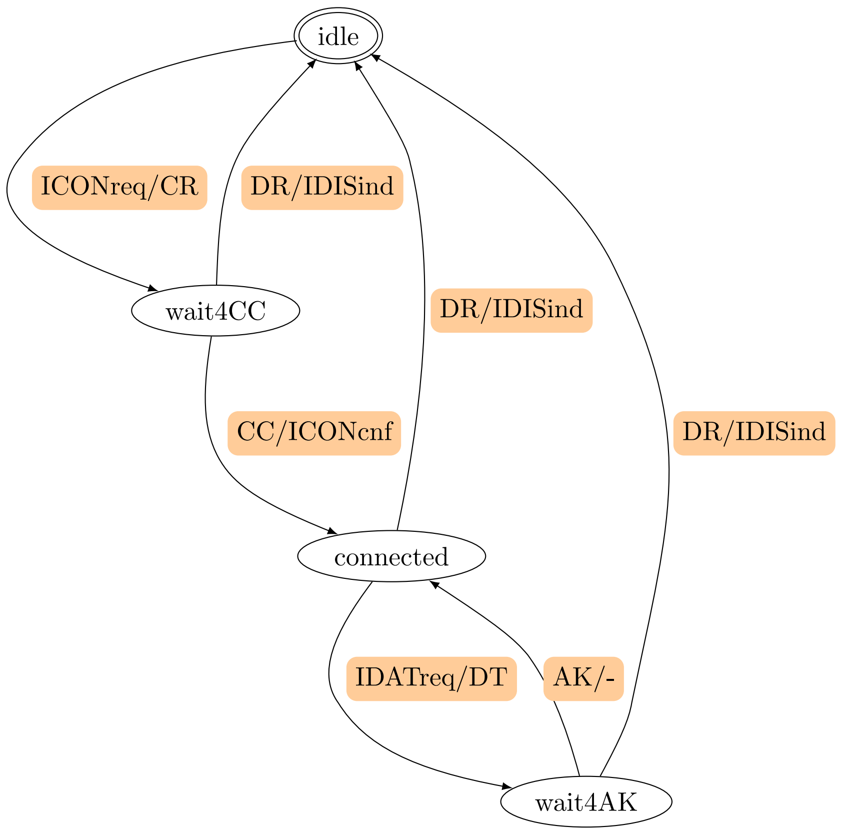
- $\text{ATS}_{res} = \{\text{ATS}_{res,1}, \text{ATS}_{res,2}, \text{ATS}_{res,3}, \text{ATS}_{res,4}\}$

  $\text{ATS}_{res,1} = \{$CR/ICONind, ICONrsp/CC, DT/IDATind, -/AK, IDISreq/DR$\}$

  $\text{ATS}_{res,2} = \{$CR/ICONind, ICONrsp/CC, IDISreq/DR$\}$

  $\text{ATS}_{res,3} = \{$CR/ICONind, IDISreq/DR$\}$

  $\text{ATS}_{res,4} = \{$DT/-$\}$



FSM in

Figure 1: FSM_{in}

Step 2.4': From the augmented test suites ATS_{in} (ATS_{res}), remove all internally triggered test cases, i.e., those test cases that are triggered by N_1 and N_2.

- $\text{ATS}'_{in} = \{\text{ATS}'_{in,1}, \text{ATS}'_{in,2}, \text{ATS}'_{in,3}, \text{ATS}'_{in,4}, \text{ATS}'_{in,5}\}$

  $\text{ATS}'_{in,1} = \{$ICONreq/CR, CC/ICONcnf, DR/IDISind$\}$

  $\text{ATS}'_{in,2} = \{$ICONreq/CR, CC/ICONcnf, IDATreq/DT, DR/IDISind$\}$

  $\text{ATS}'_{in,3} = \{$ICONreq/CR, CC/ICONcnf, IDATreq/DT, AK/-, IDATreq/DT, DR/IDISind$\}$

  $\text{ATS}'_{in,4} = \{$ICONreq/CR, CC/ICONcnf, IDATreq/DT, AK/-, DR/IDISind$\}$

  $\text{ATS}'_{in,5} = \{$ICONreq/CR, DR/IDISind$\}$

- $\text{ATS}'_{res} = \{\}$

Step 2.5: Remove the maximum suffix that does not contain an interaction with the other component. These test elements have been checked already.

- $\text{ATS}'_{in} = \{\text{ATS}'_{in,1}, \text{ATS}'_{in,2}, \text{ATS}'_{in,3}, \text{ATS}'_{in,4}, \text{ATS}'_{in,5}\}$

  $\text{ATS}'_{in,1} = \{$ICONreq/CR, CC/ICONcnf, DR/IDISind$\}$

  $\text{ATS}'_{in,2} = \{$ICONreq/CR, CC/ICONcnf, IDATreq/DT, DR/IDISind$\}$

  $\text{ATS}'_{in,3} = \{$ICONreq/CR, CC/ICONcnf, IDATreq/DT, AK/-, IDATreq/DT, DR/IDISind$\}$

  $\text{ATS}'_{in,4} = \{$ICONreq/CR, CC/ICONcnf, IDATreq/DT, AK/-, DR/IDISind$\}$

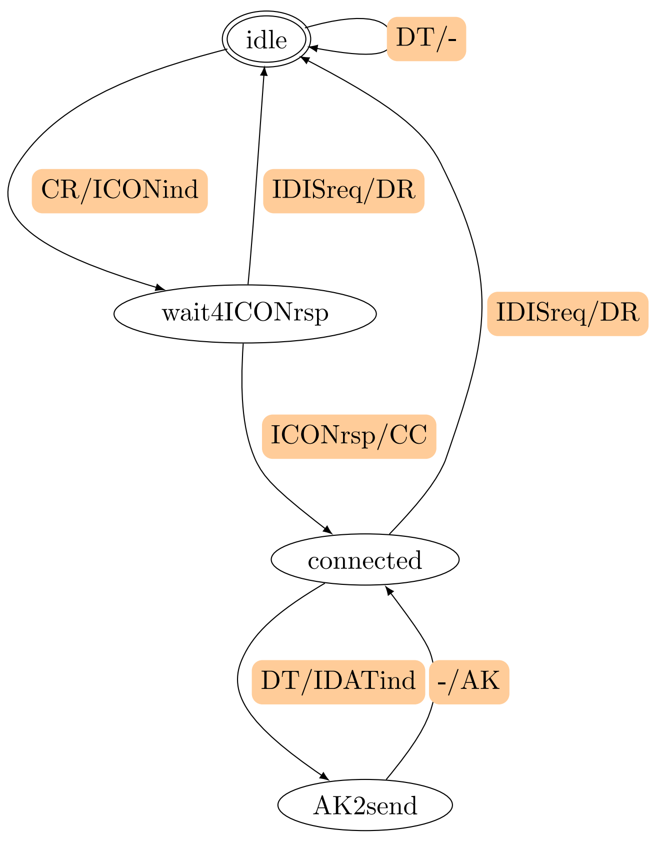


Figure 2: FSM_{res}

$$\text{ATS}'_{in,5} = \{\text{ICONreq/CR, DR/IDISind}\}$$

- $\text{ATS}'_{res} = \{\}$

Step 2.6: For each test case $\text{ATC}_{1,i}$ of the augmented test suite ATS_1 after Step 2.5, find a concatenation of augmented test cases $\text{ATC}_{2,j_1,\ldots,j_n}$ of N_2 from Step 2.3' such that $\text{ATC}_{1,i} \parallel \text{ATC}_{2,j_1,\ldots,j_n}$ is complete, and determine $\text{ATC}_{1,2,i,j_1,\ldots,j_n} = \text{ATC}_{1,i} \parallel \text{ATC}_{2,j_1,\ldots,j_n}$, yielding the concurrent augmented test suite $\text{ATS}_{1,2}$. Analogously for each test case $\text{ATC}_{2,i}$ of ATS_2. The signal-index marks the affiliation to the corresponding FSM.

- $\text{ATS}_{in,res} = \{\text{ATS}'_{in,1} \parallel \text{ATS}_{res,2},\ \text{ATS}'_{in,2} \parallel \text{ATS}_{res,2\cap 4},\ \text{ATS}'_{in,3} \parallel \text{ATS}_{res,1\cap 4},\ \text{ATS}'_{in,4} \parallel \text{ATS}_{res,1},\ \text{ATS}'_{in,5} \parallel \text{ATS}_{res,3}\}$
- $\text{ATS}_{res,in} = \{\}$

with:

$\text{ATS}'_{in,1} \parallel \text{ATS}_{res,2} = \{$

$ICONreq_1/CR_1 \bullet CR_2/ICONind_2 \bullet ICONrsp_2/CC_2 \bullet CC_1/ICONcnf_1 \bullet IDISreq_2/DR_2 \bullet DR_1/IDISind_1,$

$ICONreq_1/CR_1 \bullet CR_2/ICONind_2 \bullet ICONrsp_2/CC_2 \bullet IDISreq_2/DR_2 \bullet CC_1/ICONcnf_1 \bullet DR_1/IDISind_1$

$\}$

$\text{ATS}'_{in,2} \parallel \text{ATS}_{res,2\cap 4} = \{$

$ICONreq_1/CR_1 \bullet CR_2/ICONind_2 \bullet ICONrsp_2/CC_2 \bullet CC_1/ICONcnf_1 \bullet IDATreq_1/DT_1 \bullet IDISreq_2/DR_2 \bullet DR_1/IDISind_1 \bullet DT_2/-_2,$

$ICONreq_1/CR_1 \bullet CR_2/ICONind_2 \bullet ICONrsp_2/CC_2 \bullet CC_1/ICONcnf_1 \bullet IDATreq_1/DT_1 \bullet IDISreq_2/DR_2 \bullet DT_2/-_2 \bullet DR_1/IDISind_1,$

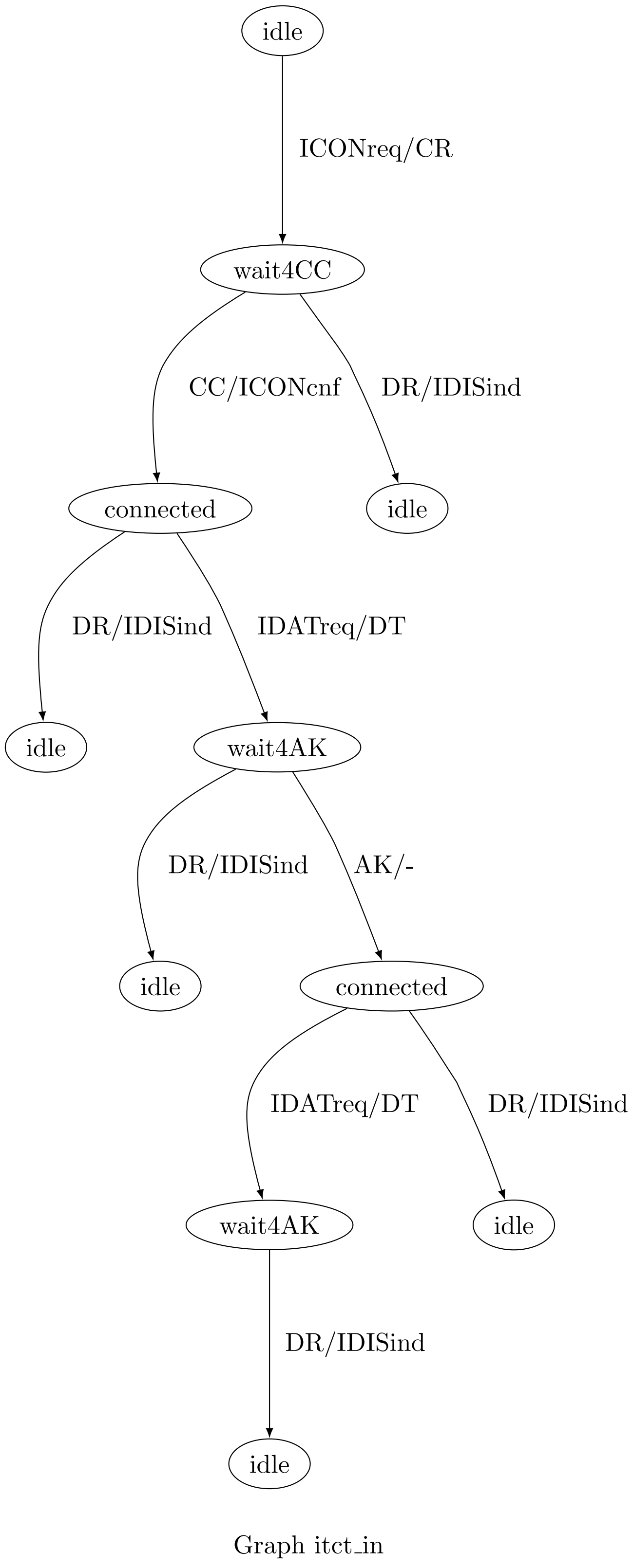
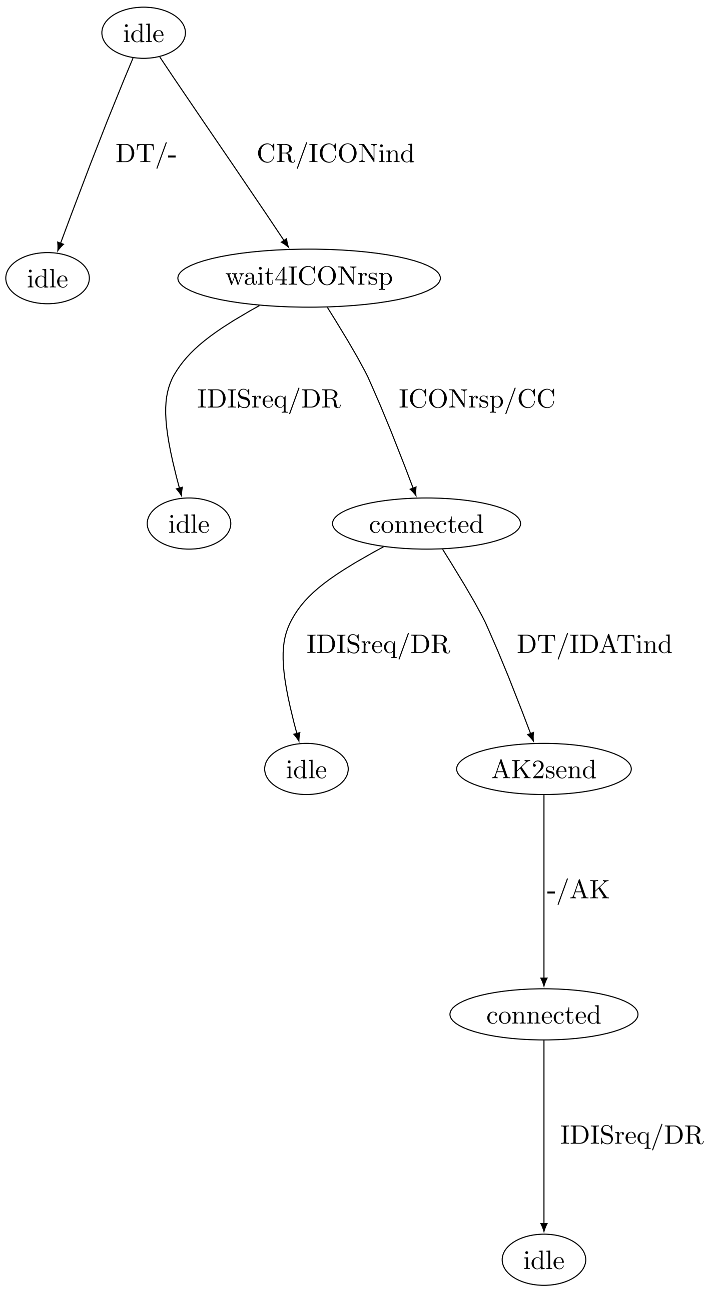


Figure 3: Initial tour coverage tree for $\textsc{fsm}_{in}$



Graph itct_res

Figure 4: Initial tour coverage tree for FSM$_{res}$

$ICONreq_1/CR_1 \bullet CR_2/ICONind_2 \bullet ICONrsp_2/CC_2 \bullet CC_1/ICONcnf_1 \bullet IDISreq_2/DR_2 \bullet IDATreq_1/DT_1 \bullet DR_1/IDISind_1 \bullet DT_2/-_2$,

$ICONreq_1/CR_1 \bullet CR_2/ICONind_2 \bullet ICONrsp_2/CC_2 \bullet CC_1/ICONcnf_1 \bullet IDISreq_2/DR_2 \bullet IDATreq_1/DT_1 \bullet DT_2/-_2 \bullet DR_1/IDISind_1$,

$ICONreq_1/CR_1 \bullet CR_2/ICONind_2 \bullet ICONrsp_2/CC_2 \bullet IDISreq_2/DR_2 \bullet CC_1/ICONcnf_1 \bullet IDATreq_1/DT_1 \bullet DR_1/IDISind_1 \bullet DT_2/-_2$,

$ICONreq_1/CR_1 \bullet CR_2/ICONind_2 \bullet ICONrsp_2/CC_2 \bullet IDISreq_2/DR_2 \bullet CC_1/ICONcnf_1 \bullet IDATreq_1/DT_1 \bullet DT_2/-_2 \bullet DR_1/IDISind_1$

}

$\text{ATS}^{'}_{in,3} \parallel \text{ATS}_{res,1\cap 4} = \{$

$ICONreq_1/CR_1 \bullet CR_2/ICONind_2 \bullet ICONrsp_2/CC_2 \bullet CC_1/ICONcnf_1 \bullet IDATreq_1/DT_1 \bullet DT_2/IDATind_2 \bullet -_2/AK_2 \bullet AK_1/-_1 \bullet IDATreq_1/DT_1 \bullet IDISreq_2/DR_2 \bullet DR_1/IDISind_1 \bullet DT_2/-_2$,

$ICONreq_1/CR_1 \bullet CR_2/ICONind_2 \bullet ICONrsp_2/CC_2 \bullet CC_1/ICONcnf_1 \bullet IDATreq_1/DT_1 \bullet DT_2/IDATind_2 \bullet -_2/AK_2 \bullet AK_1/-_1 \bullet IDATreq_1/DT_1 \bullet IDISreq_2/DR_2 \bullet DT_2/-_2 \bullet DR_1/IDISind_1$,

$ICONreq_1/CR_1 \bullet CR_2/ICONind_2 \bullet ICONrsp_2/CC_2 \bullet CC_1/ICONcnf_1 \bullet IDATreq_1/DT_1 \bullet DT_2/IDATind_2 \bullet -_2/AK_2 \bullet AK_1/-_1 \bullet IDISreq_2/DR_2 \bullet IDATreq_1/DT_1 \bullet DR_1/IDISind_1 \bullet DT_2/-_2$,

$ICONreq_1/CR_1 \bullet CR_2/ICONind_2 \bullet ICONrsp_2/CC_2 \bullet CC_1/ICONcnf_1 \bullet IDATreq_1/DT_1 \bullet DT_2/IDATind_2 \bullet -_2/AK_2 \bullet AK_1/-_1 \bullet IDISreq_2/DR_2 \bullet IDATreq_1/DT_1 \bullet DT_2/-_2 \bullet DR_1/IDISind_1$,

$ICONreq_1/CR_1 \bullet CR_2/ICONind_2 \bullet ICONrsp_2/CC_2 \bullet CC_1/ICONcnf_1 \bullet IDATreq_1/DT_1 \bullet DT_2/IDATind_2 \bullet -_2/AK_2 \bullet IDISreq_2/DR_2 \bullet AK_1/-_1 \bullet IDATreq_1/DT_1 \bullet DR_1/IDISind_1 \bullet DT_2/-_2$,

$ICONreq_1/CR_1 \bullet CR_2/ICONind_2 \bullet ICONrsp_2/CC_2 \bullet CC_1/ICONcnf_1 \bullet IDATreq_1/DT_1 \bullet DT_2/IDATind_2 \bullet -_2/AK_2 \bullet IDISreq_2/DR_2 \bullet AK_1/-_1 \bullet IDATreq_1/DT_1 \bullet DT_2/-_2 \bullet DR_1/IDISind_1$

}

$\text{ATS}^{'}_{in,4} \parallel \text{ATS}_{res,1} = \{$

$ICONreq_1/CR_1 \bullet CR_2/ICONind_2 \bullet ICONrsp_2/CC_2 \bullet CC_1/ICONcnf_1 \bullet IDATreq_1/DT_1 \bullet DT_2/IDATind_2 \bullet -_2/AK_2 \bullet AK_1/-_1 \bullet IDISreq_2/DR_2 \bullet DR_1/IDISind_1$,

$ICONreq_1/CR_1 \bullet CR_2/ICONind_2 \bullet ICONrsp_2/CC_2 \bullet CC_1/ICONcnf_1 \bullet IDATreq_1/DT_1 \bullet DT_2/IDATind_2 \bullet -_2/AK_2 \bullet IDISreq_2/DR_2 \bullet AK_1/-_1 \bullet DR_1/IDISind_1$

}

$\text{ATS}^{'}_{in,5} \parallel \text{ATS}_{res,3} = \{$

$ICONreq_1/CR_1 \bullet CR_2/ICONind_2 \bullet IDISreq_2/DR_2 \bullet DR_1/IDISind_1$

}

Step 2.7: Based on ATS_1, ATS_2, and $\text{ATS}_{1,2}$, check whether N_1 and N_2 meet the composition criteria CC_1 and CC_2, i.e., whether for each test case ATS_1 (ATS_2), there is a matching test case of N_2 (N_1). Yes: Continue with Step 2.8; No: Stop.

Unspecified reception: False

Deadlocks: False

Step 2.8: For each test case in $\text{ATS}_{1,2}$: Merge adjacent test case elements in case where (1) the internal output of the first matches the internal input of the second, and (2) the output is the only signal in the queue after being sent. Replace internal inputs and outputs by "-", and remove test case elements "-/-". (Remaining - are made anonymous (common index 0) in order to remove duplicates resulting from the merge.)

- $\text{ATS}_{in,res} = \{\text{ATS}^{'}_{in,1} \parallel \text{ATS}_{res,2}, \text{ATS}^{'}_{in,2} \parallel \text{ATS}_{res,2\cap 4}, \text{ATS}^{'}_{in,3} \parallel \text{ATS}_{res,1\cap 4}, \text{ATS}^{'}_{in,4} \parallel \text{ATS}_{res,1}, \text{ATS}^{'}_{in,5} \parallel \text{ATS}_{res,3}\}$
- $\text{ATS}_{res,in} = \{\}$

with:

$\text{ATS}^{'}_{in,1} \parallel \text{ATS}_{res,2} = \{$

$ICONreq_1/ICONind_2 \bullet ICONrsp_2/ICONcnf_1 \bullet IDISreq_2/IDISind_1,$

$ICONreq_1/ICONind_2 \bullet ICONrsp_2/-_0 \bullet IDISreq_2/-_0 \bullet -_0/ICONcnf_1 \bullet -_0/IDISind_1$

$\}$

$\text{ATS}^{'}_{in,2} \parallel \text{ATS}_{res,2\cap 4} = \{$

$ICONreq_1/ICONind_2 \bullet ICONrsp_2/ICONcnf_1 \bullet IDATreq_1/-_0 \bullet IDISreq_2/IDISind_1,$

$ICONreq_1/ICONind_2 \bullet ICONrsp_2/ICONcnf_1 \bullet IDATreq_1/-_0 \bullet IDISreq_2/-_0 \bullet -_0/IDISind_1,$

$ICONreq_1/ICONind_2 \bullet ICONrsp_2/ICONcnf_1 \bullet IDISreq_2/-_0 \bullet IDATreq_1/-_0 \bullet -_0/IDISind_1,$

$ICONreq_1/ICONind_2 \bullet ICONrsp_2/-_0 \bullet IDISreq_2/-_0 \bullet -_0/ICONcnf_1 \bullet IDATreq_1/-_0 \bullet -_0/IDISind_1$

$\}$

$\text{ATS}^{'}_{in,3} \parallel \text{ATS}_{res,1\cap 4} = \{$

$ICONreq_1/ICONind_2 \bullet ICONrsp_2/ICONcnf_1 \bullet IDATreq_1/IDATind_2 \bullet IDATreq_1/-_0 \bullet IDISreq_2/IDISind_1,$

$ICONreq_1/ICONind_2 \bullet ICONrsp_2/ICONcnf_1 \bullet IDATreq_1/IDATind_2 \bullet IDATreq_1/-_0 \bullet IDISreq_2/-_0$
$\bullet -_0/IDISind_1,$

$ICONreq_1/ICONind_2 \bullet ICONrsp_2/ICONcnf_1 \bullet IDATreq_1/IDATind_2 \bullet IDISreq_2/-_0 \bullet IDATreq_1/-_0$
$\bullet -_0/IDISind_1$

$\}$

$\text{ATS}^{'}_{in,4} \parallel \text{ATS}_{res,1} = \{$

$ICONreq_1/ICONind_2 \bullet ICONrsp_2/ICONcnf_1 \bullet IDATreq_1/IDATind_2 \bullet IDISreq_2/IDISind_1,$

$ICONreq_1/ICONind_2 \bullet ICONrsp_2/ICONcnf_1 \bullet IDATreq_1/IDATind_2 \bullet IDISreq_2/-_0 \bullet -_0/IDISind_1$

$\}$

$\text{ATS}^{'}_{in,5} \parallel \text{ATS}_{res,3} = \{$

$ICONreq_1/ICONind_2 \bullet IDISreq_2/IDISind_1$

$\}$

Step 2.9: Execute the test.

APPENDIX C

Tool output for the RedSimpleTCP example

C-Method applied to $\text{FSM}_{RedSimpleTCP}$ and $\text{FSM}_{RedSimpleTCP}$

Automatically generated by c_method.py

December 5, 2008

Step 2.1: Remove all transitions of N_1 and N_2 that yield an error output. These transitions have already been tested during component testing, and need not to be tested again.

Step 2.2: Build the initial tour coverage trees for N_1 ($\text{FSM}_{RedSimpleTCP}$) and N_2 ($\text{FSM}_{RedSimpleTCP}$), and determine all maximal paths, i.e., all path that start at the root node and end at a leaf node, constituting augmented test suites $\text{ATS}_{RedSimpleTCP}$ and $\text{ATS}_{RedSimpleTCP}$.

- $\text{ATS}_{RedSimpleTCP} = \{\text{ATS}_{RedSimpleTCP,1}, \text{ATS}_{RedSimpleTCP,2}, \text{ATS}_{RedSimpleTCP,3}, \text{ATS}_{RedSimpleTCP,4}, \text{ATS}_{RedSimpleTCP,5}, \text{ATS}_{RedSimpleTCP,6}, \text{ATS}_{RedSimpleTCP,7}, \text{ATS}_{RedSimpleTCP,8}, \text{ATS}_{RedSimpleTCP,9}, \text{ATS}_{RedSimpleTCP,10}, \text{ATS}_{RedSimpleTCP,11}, \text{ATS}_{RedSimpleTCP,12}, \text{ATS}_{RedSimpleTCP,13}\}$

  $\text{ATS}_{RedSimpleTCP,1} = \{$SYN/SYN_ACKS, Close/FIN, ACKS/-, FIN/ACKF, ACKF/-$\}$

  $\text{ATS}_{RedSimpleTCP,2} = \{$SYN/SYN_ACKS, Close/FIN, ACKS/-, ACKF/-, FIN/ACKF$\}$

  $\text{ATS}_{RedSimpleTCP,3} = \{$SYN/SYN_ACKS, ACKS/-, Close/FIN, FIN/ACKF, ACKF/-$\}$

  $\text{ATS}_{RedSimpleTCP,4} = \{$SYN/SYN_ACKS, ACKS/-, Close/FIN, ACKF/-, FIN/ACKF$\}$

  $\text{ATS}_{RedSimpleTCP,5} = \{$SYN/SYN_ACKS, ACKS/-, FIN/ACKF, Close/FIN, ACKF/-$\}$

  $\text{ATS}_{RedSimpleTCP,6} = \{$aOpen/SYN, SYN/ACKS, ACKS/-, Close/FIN, FIN/ACKF, ACKF/-$\}$

  $\text{ATS}_{RedSimpleTCP,7} = \{$aOpen/SYN, SYN/ACKS, ACKS/-, Close/FIN, ACKF/-, FIN/ACKF$\}$

  $\text{ATS}_{RedSimpleTCP,8} = \{$aOpen/SYN, SYN/ACKS, ACKS/-, FIN/ACKF, Close/FIN, ACKF/-$\}$

  $\text{ATS}_{RedSimpleTCP,9} = \{$aOpen/SYN, SYN/ACKS, Close/FIN, ACKS/-, FIN/ACKF, ACKF/-$\}$

  $\text{ATS}_{RedSimpleTCP,10} = \{$aOpen/SYN, SYN/ACKS, Close/FIN, ACKS/-, ACKF/-, FIN/ACKF$\}$

  $\text{ATS}_{RedSimpleTCP,11} = \{$aOpen/SYN, SYN_ACKS/ACKS, Close/FIN, FIN/ACKF, ACKF/-$\}$

  $\text{ATS}_{RedSimpleTCP,12} = \{$aOpen/SYN, SYN_ACKS/ACKS, Close/FIN, ACKF/-, FIN/ACKF$\}$

  $\text{ATS}_{RedSimpleTCP,13} = \{$aOpen/SYN, SYN_ACKS/ACKS, FIN/ACKF, Close/FIN, ACKF/-$\}$

Step 2.3': From the augmented test cases, remove all local tours, i.e., (sub)sequences of test case elements that (1) start and end in the same state, and (2) contain only external inputs and outputs. They have already been checked during component testing, and need not be tested again.

- $\text{ATS}_{RedSimpleTCP} = \{\text{ATS}_{RedSimpleTCP,1}, \text{ATS}_{RedSimpleTCP,2}, \text{ATS}_{RedSimpleTCP,3}, \text{ATS}_{RedSimpleTCP,4}, \text{ATS}_{RedSimpleTCP,5}, \text{ATS}_{RedSimpleTCP,6}, \text{ATS}_{RedSimpleTCP,7}, \text{ATS}_{RedSimpleTCP,8}, \text{ATS}_{RedSimpleTCP,9}, \text{ATS}_{RedSimpleTCP,10}, \text{ATS}_{RedSimpleTCP,11}, \text{ATS}_{RedSimpleTCP,12}, \text{ATS}_{RedSimpleTCP,13}\}$

  $\text{ATS}_{RedSimpleTCP,1} = \{$SYN/SYN_ACKS, Close/FIN, ACKS/-, FIN/ACKF, ACKF/-$\}$

  $\text{ATS}_{RedSimpleTCP,2} = \{$SYN/SYN_ACKS, Close/FIN, ACKS/-, ACKF/-, FIN/ACKF$\}$

  $\text{ATS}_{RedSimpleTCP,3} = \{$SYN/SYN_ACKS, ACKS/-, Close/FIN, FIN/ACKF, ACKF/-$\}$

  $\text{ATS}_{RedSimpleTCP,4} = \{$SYN/SYN_ACKS, ACKS/-, Close/FIN, ACKF/-, FIN/ACKF$\}$

  $\text{ATS}_{RedSimpleTCP,5} = \{$SYN/SYN_ACKS, ACKS/-, FIN/ACKF, Close/FIN, ACKF/-$\}$

  $\text{ATS}_{RedSimpleTCP,6} = \{$aOpen/SYN, SYN/ACKS, ACKS/-, Close/FIN, FIN/ACKF, ACKF/-$\}$

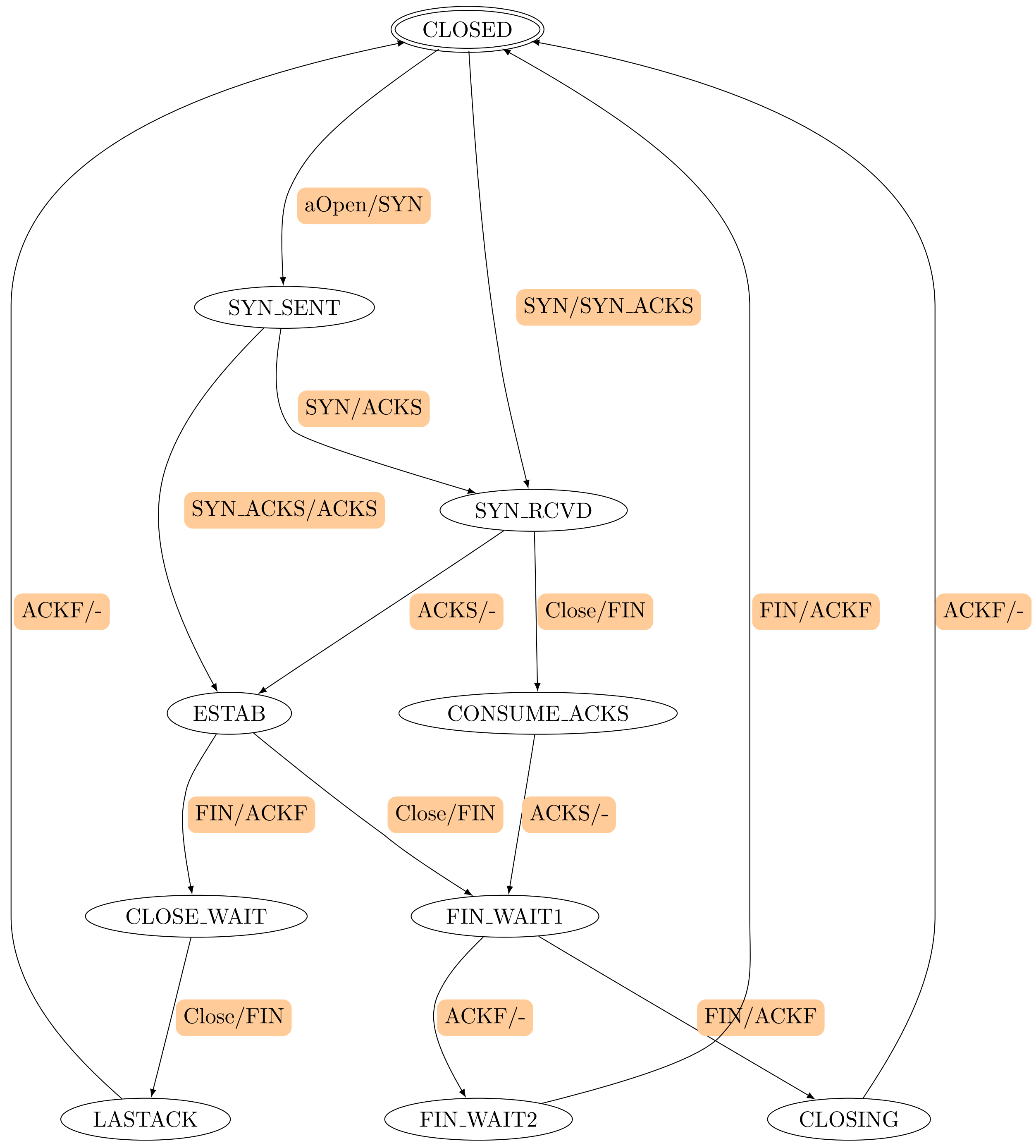


Figure 1: FSM$_{RedSimpleTCP}$

$\text{ATS}_{RedSimpleTCP,7} = \{$aOpen/SYN, SYN/ACKS, ACKS/-, Close/FIN, ACKF/-, FIN/ACKF$\}$

$\text{ATS}_{RedSimpleTCP,8} = \{$aOpen/SYN, SYN/ACKS, ACKS/-, FIN/ACKF, Close/FIN, ACKF/-$\}$

$\text{ATS}_{RedSimpleTCP,9} = \{$aOpen/SYN, SYN/ACKS, Close/FIN, ACKS/-, FIN/ACKF, ACKF/-$\}$

$\text{ATS}_{RedSimpleTCP,10} = \{$aOpen/SYN, SYN/ACKS, Close/FIN, ACKS/-, ACKF/-, FIN/ACKF$\}$

$\text{ATS}_{RedSimpleTCP,11} = \{$aOpen/SYN, SYN_ACKS/ACKS, Close/FIN, FIN/ACKF, ACKF/-$\}$

$\text{ATS}_{RedSimpleTCP,12} = \{$aOpen/SYN, SYN_ACKS/ACKS, Close/FIN, ACKF/-, FIN/ACKF$\}$

$\text{ATS}_{RedSimpleTCP,13} = \{$aOpen/SYN, SYN_ACKS/ACKS, FIN/ACKF, Close/FIN, ACKF/-$\}$

Step 2.4': From the augmented test suites $\text{ATS}_{RedSimpleTCP}$ ($\text{ATS}_{RedSimpleTCP}$), remove all internally triggered test cases, i.e., those test cases that are triggered by N_1 and N_2.

- $\text{ATS}'_{RedSimpleTCP} = \{\text{ATS}'_{RedSimpleTCP,1}, \text{ATS}'_{RedSimpleTCP,2}, \text{ATS}'_{RedSimpleTCP,3}, \text{ATS}'_{RedSimpleTCP,4}, \text{ATS}'_{RedSimpleTCP,5}, \text{ATS}'_{RedSimpleTCP,6}, \text{ATS}'_{RedSimpleTCP,7}, \text{ATS}'_{RedSimpleTCP,8}\}$

  $\text{ATS}'_{RedSimpleTCP,1} = \{$aOpen/SYN, SYN/ACKS, ACKS/-, Close/FIN, FIN/ACKF, ACKF/-$\}$

  $\text{ATS}'_{RedSimpleTCP,2} = \{$aOpen/SYN, SYN/ACKS, ACKS/-, Close/FIN, ACKF/-, FIN/ACKF$\}$

  $\text{ATS}'_{RedSimpleTCP,3} = \{$aOpen/SYN, SYN/ACKS, ACKS/-, FIN/ACKF, Close/FIN, ACKF/-$\}$

  $\text{ATS}'_{RedSimpleTCP,4} = \{$aOpen/SYN, SYN/ACKS, Close/FIN, ACKS/-, FIN/ACKF, ACKF/-$\}$

  $\text{ATS}'_{RedSimpleTCP,5} = \{$aOpen/SYN, SYN/ACKS, Close/FIN, ACKS/-, ACKF/-, FIN/ACKF$\}$

  $\text{ATS}'_{RedSimpleTCP,6} = \{$aOpen/SYN, SYN_ACKS/ACKS, Close/FIN, FIN/ACKF, ACKF/-$\}$

  $\text{ATS}'_{RedSimpleTCP,7} = \{$aOpen/SYN, SYN_ACKS/ACKS, Close/FIN, ACKF/-, FIN/ACKF$\}$

  $\text{ATS}'_{RedSimpleTCP,8} = \{$aOpen/SYN, SYN_ACKS/ACKS, FIN/ACKF, Close/FIN, ACKF/-$\}$

Step 2.5: Remove the maximum suffix that does not contain an interaction with the other component. These test elements have been checked already.

- $\text{ATS}'_{RedSimpleTCP} = \{\text{ATS}'_{RedSimpleTCP,1}, \text{ATS}'_{RedSimpleTCP,2}, \text{ATS}'_{RedSimpleTCP,3}, \text{ATS}'_{RedSimpleTCP,4}, \text{ATS}'_{RedSimpleTCP,5}, \text{ATS}'_{RedSimpleTCP,6}, \text{ATS}'_{RedSimpleTCP,7}, \text{ATS}'_{RedSimpleTCP,8}\}$

  $\text{ATS}'_{RedSimpleTCP,1} = \{$aOpen/SYN, SYN/ACKS, ACKS/-, Close/FIN, FIN/ACKF, ACKF/-$\}$

  $\text{ATS}'_{RedSimpleTCP,2} = \{$aOpen/SYN, SYN/ACKS, ACKS/-, Close/FIN, ACKF/-, FIN/ACKF$\}$

  $\text{ATS}'_{RedSimpleTCP,3} = \{$aOpen/SYN, SYN/ACKS, ACKS/-, FIN/ACKF, Close/FIN, ACKF/-$\}$

  $\text{ATS}'_{RedSimpleTCP,4} = \{$aOpen/SYN, SYN/ACKS, Close/FIN, ACKS/-, FIN/ACKF, ACKF/-$\}$

  $\text{ATS}'_{RedSimpleTCP,5} = \{$aOpen/SYN, SYN/ACKS, Close/FIN, ACKS/-, ACKF/-, FIN/ACKF$\}$

  $\text{ATS}'_{RedSimpleTCP,6} = \{$aOpen/SYN, SYN_ACKS/ACKS, Close/FIN, FIN/ACKF, ACKF/-$\}$

  $\text{ATS}'_{RedSimpleTCP,7} = \{$aOpen/SYN, SYN_ACKS/ACKS, Close/FIN, ACKF/-, FIN/ACKF$\}$

  $\text{ATS}'_{RedSimpleTCP,8} = \{$aOpen/SYN, SYN_ACKS/ACKS, FIN/ACKF, Close/FIN, ACKF/-$\}$

Step 2.6: For each test case $\text{ATC}_{1,i}$ of the augmented test suite ATS_1 after Step 2.5, find a concatenation of augmented test cases $\text{ATC}_{2,j_1,\dots,j_n}$ of N_2 from Step 2.3' such that $\text{ATC}_{1,i} \parallel \text{ATC}_{2,j_1,\dots,j_n}$ is complete, and determine $\text{ATC}_{1,2,i,j_1,\dots,j_n} = \text{ATC}_{1,i} \parallel \text{ATC}_{2,j_1,\dots,j_n}$, yielding the concurrent augmented test suite $\text{ATS}_{1,2}$. Analogously for each test case $\text{ATC}_{2,i}$ of ATS_2. The signal-index marks the affiliation to the corresponding FSM.

- $\text{ATS}_{RedSimpleTCP,RedSimpleTCP} = \{\text{ATS}'_{RedSimpleTCP,1} \parallel \text{ATS}_{RedSimpleTCP,6}, \text{ATS}'_{RedSimpleTCP,2} \parallel \text{ATS}_{RedSimpleTCP,8}, \text{ATS}'_{RedSimpleTCP,3} \parallel \text{ATS}_{RedSimpleTCP,7}, \text{ATS}'_{RedSimpleTCP,4} \parallel \text{ATS}_{RedSimpleTCP,6}, \text{ATS}'_{RedSimpleTCP,5} \parallel \text{ATS}_{RedSimpleTCP,8}, \text{ATS}'_{RedSimpleTCP,6} \parallel \text{ATS}_{RedSimpleTCP,1}, \text{ATS}'_{RedSimpleTCP,7} \parallel \text{ATS}_{RedSimpleTCP,5}, \text{ATS}'_{RedSimpleTCP,8} \parallel \text{ATS}_{RedSimpleTCP,2}\}$

with:

$\mathrm{ATS}'_{RedSimpleTCP,1} \parallel \mathrm{ATS}_{RedSimpleTCP,6} = \{$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1$
$\bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1$
$\bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1$
$\bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1$
$\bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2$
$\bullet Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2$
$\bullet Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2$
$\bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2$
$\bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1$
$\bullet Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1$
$\bullet Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1$
$\bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1$
$\bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2$
$\bullet\ Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2$
$\bullet\ Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2$
$\bullet\ Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2$
$\bullet\ Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1$
$\bullet\ Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1$
$\bullet\ Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1$
$\bullet\ Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1$
$\bullet\ Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2$
$\bullet\ Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2$
$\bullet\ Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2$
$\bullet\ Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2$
$\bullet\ Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1$
$\bullet\ Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1$
$\bullet\ Close_1/FIN_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1$
$\bullet\ Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1$
$\bullet\ Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$

$\}$

$\mathrm{ATS}'_{RedSimpleTCP,2} \parallel \mathrm{ATS}_{RedSimpleTCP,8} = \{$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet\ FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet\ FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet\ FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet\ FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1$
$\bullet\ FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1$
$\bullet\ FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet\ FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet\ FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet\ FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet\ FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1$
$\bullet\ FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1$
$\bullet\ FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet\ FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet\ FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$

}

$\text{ATS}^{'}_{RedSimpleTCP,3} \parallel \text{ATS}_{RedSimpleTCP,7} = \{$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet Close_1/FIN_1 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet Close_1/FIN_1 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet Close_1/FIN_1 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet Close_1/FIN_1 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet Close_1/FIN_1 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet Close_1/FIN_1 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet Close_1/FIN_1 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet Close_1/FIN_1 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet Close_1/FIN_1 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet Close_1/FIN_1 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet Close_1/FIN_1 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet Close_1/FIN_1 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$

}

$\mathrm{ATS}'_{RedSimpleTCP,4} \parallel \mathrm{ATS}_{RedSimpleTCP,6} = \{$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2$
$\bullet\ FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2$
$\bullet\ FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1$
$\bullet\ ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1$
$\bullet\ ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1$
$\bullet\ ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1$
$\bullet\ ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1$
$\bullet\ FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1$
$\bullet\ FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2$
$\bullet\ FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2$
$\bullet\ FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1$
$\bullet\ ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1$
$\bullet\ ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1$
$\bullet\ ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1$
$\bullet\ ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1$
$\bullet\ FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1$
$\bullet\ FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2$
$\bullet\ FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1$
$\bullet\ Close_2/FIN_2 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2$
$\bullet\ ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2$
$\bullet\ FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2$
$\bullet\ FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1$
$\bullet\ ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1$
$\bullet\ ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1$
$\bullet\ ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1$
$\bullet\ ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1$
$\bullet\ FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1$
$\bullet\ FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$

$\}$

$\text{ATS}'_{RedSimpleTCP,5} \parallel \text{ATS}_{RedSimpleTCP,8} = \{$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2$
$\bullet\ FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2$
$\bullet\ FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2$
$\bullet\ ACKS_1/-_1 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2$
$\bullet\ ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2$
$\bullet\ Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2$
$\bullet\ FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2$
$\bullet\ FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_1/SYN_1 \bullet aOpen_2/SYN_2 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1 \bullet FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$,

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2$
$\bullet\ FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2$
$\bullet\ ACKS_1/-_1 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2$
$\bullet\ ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2$
$\bullet\ Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1$
$\bullet\ FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1$
$\bullet\ FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2$
$\bullet\ ACKS_1/-_1 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2$
$\bullet\ ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_1/ACKS_1 \bullet SYN_2/ACKS_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2$
$\bullet\ Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2$
$\bullet\ FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1 \bullet ACKS_2/-_2$
$\bullet\ FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKS_1/-_1$
$\bullet\ FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2$
$\bullet\ ACKS_1/-_1 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2$
$\bullet\ ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2$
$\bullet\ Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1$
$\bullet\ FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKS_1/-_1$
$\bullet\ FIN_2/ACKF_2 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2$
$\bullet\ ACKS_1/-_1 \bullet ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2$
$\bullet\ ACKS_1/-_1 \bullet Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_2/SYN_2 \bullet aOpen_1/SYN_1 \bullet SYN_2/ACKS_2 \bullet SYN_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2$
$\bullet\ Close_2/FIN_2 \bullet ACKS_1/-_1 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$

$\}$

$\mathrm{ATS}'_{RedSimpleTCP,6} \parallel \mathrm{ATS}_{RedSimpleTCP,1} = \{$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1$
$\bullet\ ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1$
$ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2 \bullet ACKS_2/-_2$
$\bullet\ FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2 \bullet ACKS_2/-_2$
$\bullet\ FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2 \bullet ACKS_2/-_2$
$\bullet\ FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_1/FIN_1 \bullet Close_2/FIN_2 \bullet ACKS_2/-_2$
$\bullet\ FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet FIN_1/ACKF_1$
$\bullet\ ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet FIN_1/ACKF_1$
$\bullet\ ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet\ FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet\ FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet\ FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_2/FIN_2 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet\ FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_2/FIN_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet\ FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_2/FIN_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet\ FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_2/FIN_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet\ FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_2/FIN_2 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet\ FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet Close_2/FIN_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_1/FIN_1 \bullet FIN_1/ACKF_1$
$\bullet\ ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet Close_2/FIN_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_1/FIN_1 \bullet FIN_1/ACKF_1$
$\bullet\ ACKS_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet Close_2/FIN_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet\ FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet Close_2/FIN_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet\ FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet Close_2/FIN_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet\ FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet Close_2/FIN_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2$
$\bullet\ FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet Close_2/FIN_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet\ FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet Close_2/FIN_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet\ FIN_1/ACKF_1 \bullet FIN_2/ACKF_2 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet Close_2/FIN_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet\ FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_1/-_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet Close_2/FIN_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1$
$\bullet\ FIN_2/ACKF_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet ACKF_1/-_1$

$\}$

$\text{ATS}'_{RedSimpleTCP,7} \parallel \text{ATS}_{RedSimpleTCP,5} = \{$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2$
$\bullet\ ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet FIN_2/ACKF_2$
$\bullet\ Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2$
$\bullet\ ACKF_1/-_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2$
$\bullet\ Close_2/FIN_2 \bullet ACKF_1/-_1 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2$

$\}$

$\text{ATS}'_{RedSimpleTCP,8} \parallel \text{ATS}_{RedSimpleTCP,2} = \{$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_2/FIN_2 \bullet FIN_1/ACKF_1 \bullet ACKS_2/-_2 \bullet ACKF_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_2/FIN_2 \bullet ACKS_2/-_2 \bullet FIN_1/ACKF_1 \bullet Close_1/FIN_1 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet Close_2/FIN_2 \bullet ACKS_2/-_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet Close_2/FIN_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet FIN_1/ACKF_1 \bullet Close_1/FIN_1 \bullet ACKS_2/-_2 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet Close_2/FIN_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet FIN_1/ACKF_1 \bullet ACKS_2/-_2 \bullet Close_1/FIN_1 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet Close_2/FIN_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet FIN_1/ACKF_1 \bullet ACKS_2/-_2 \bullet ACKF_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet Close_2/FIN_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet ACKS_2/-_2 \bullet FIN_1/ACKF_1 \bullet Close_1/FIN_1 \bullet ACKF_2/-_2 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1,$

$aOpen_1/SYN_1 \bullet SYN_2/SYN\_ACKS_2 \bullet Close_2/FIN_2 \bullet SYN\_ACKS_1/ACKS_1 \bullet ACKS_2/-_2 \bullet FIN_1/ACKF_1 \bullet ACKF_2/-_2 \bullet Close_1/FIN_1 \bullet FIN_2/ACKF_2 \bullet ACKF_1/-_1$

$\}$

Step 2.7: Based on ATS_1, ATS_2, and $\text{ATS}_{1,2}$, check whether N_1 and N_2 meet the composition criteria CC_1 and CC_2, i.e., whether for each test case ATS_1 (ATS_2), there is a matching test case of N_2 (N_1). Yes: Continue with Step 2.8; No: Stop.

Unspecified reception: False

Deadlocks: False

Step 2.8: For each test case in $\text{ATS}_{1,2}$: Merge adjacent test case elements in case where (1) the internal output of the first matches the internal input of the second, and (2) the output is the only signal in the queue after being sent. Replace internal inputs and outputs by "-", and remove test case elements "-/-". (Remaining - are made anonymous (common index 0) in order to remove duplicates resulting from the merge.)

- $\text{ATS}_{RedSimpleTCP,RedSimpleTCP} = \{\text{ATS}'_{RedSimpleTCP,1} \parallel \text{ATS}_{RedSimpleTCP,6}, \text{ATS}'_{RedSimpleTCP,2} \parallel \text{ATS}_{RedSimpleTCP,8}, \text{ATS}'_{RedSimpleTCP,3} \parallel \text{ATS}_{RedSimpleTCP,7}, \text{ATS}'_{RedSimpleTCP,4} \parallel \text{ATS}_{RedSimpleTCP,6}, \text{ATS}'_{RedSimpleTCP,5} \parallel \text{ATS}_{RedSimpleTCP,8}, \text{ATS}'_{RedSimpleTCP,6} \parallel \text{ATS}_{RedSimpleTCP,1}, \text{ATS}'_{RedSimpleTCP,7} \parallel \text{ATS}_{RedSimpleTCP,5}, \text{ATS}'_{RedSimpleTCP,8} \parallel \text{ATS}_{RedSimpleTCP,2}\}$

with:

$\text{ATS}'_{RedSimpleTCP,1} \parallel \text{ATS}_{RedSimpleTCP,6} = \{$

$aOpen_1/-_0 \bullet aOpen_2/-_0 \bullet Close_1/-_0 \bullet Close_2/-_0,$

$aOpen_1/-_0 \bullet aOpen_2/-_0 \bullet Close_2/-_0 \bullet Close_1/-_0,$

$aOpen_2/-_0 \bullet aOpen_1/-_0 \bullet Close_1/-_0 \bullet Close_2/-_0,$

$aOpen_2/-_0 \bullet aOpen_1/-_0 \bullet Close_2/-_0 \bullet Close_1/-_0$

$\}$

$\text{ATS}'_{RedSimpleTCP,2} \parallel \text{ATS}_{RedSimpleTCP,8} = \{$

$aOpen_1/-_0 \bullet aOpen_2/-_0 \bullet Close_1/-_0 \bullet Close_2/-_0,$

$aOpen_2/-_0 \bullet aOpen_1/-_0 \bullet Close_1/-_0 \bullet Close_2/-_0$

$\}$

$\text{ATS}'_{RedSimpleTCP,3} \parallel \text{ATS}_{RedSimpleTCP,7} = \{$

$$aOpen_1/-_0 \bullet aOpen_2/-_0 \bullet Close_2/-_0 \bullet Close_1/-_0,$$
$$aOpen_2/-_0 \bullet aOpen_1/-_0 \bullet Close_2/-_0 \bullet Close_1/-_0$$

$\}$

$\text{ATS}^{'}_{RedSimpleTCP,4} \parallel \text{ATS}_{RedSimpleTCP,6} = \{$

$$aOpen_1/-_0 \bullet aOpen_2/-_0 \bullet Close_1/-_0 \bullet Close_2/-_0,$$
$$aOpen_1/-_0 \bullet aOpen_2/-_0 \bullet Close_2/-_0 \bullet Close_1/-_0,$$
$$aOpen_2/-_0 \bullet aOpen_1/-_0 \bullet Close_1/-_0 \bullet Close_2/-_0,$$
$$aOpen_2/-_0 \bullet aOpen_1/-_0 \bullet Close_2/-_0 \bullet Close_1/-_0$$

$\}$

$\text{ATS}^{'}_{RedSimpleTCP,5} \parallel \text{ATS}_{RedSimpleTCP,8} = \{$

$$aOpen_1/-_0 \bullet aOpen_2/-_0 \bullet Close_1/-_0 \bullet Close_2/-_0,$$
$$aOpen_2/-_0 \bullet aOpen_1/-_0 \bullet Close_1/-_0 \bullet Close_2/-_0$$

$\}$

$\text{ATS}^{'}_{RedSimpleTCP,6} \parallel \text{ATS}_{RedSimpleTCP,1} = \{$

$$aOpen_1/-_0 \bullet Close_1/-_0 \bullet Close_2/-_0,$$
$$aOpen_1/-_0 \bullet Close_2/-_0 \bullet Close_1/-_0$$

$\}$

$\text{ATS}^{'}_{RedSimpleTCP,7} \parallel \text{ATS}_{RedSimpleTCP,5} = \{$

$$aOpen_1/-_0 \bullet Close_1/-_0 \bullet Close_2/-_0$$

$\}$

$\text{ATS}^{'}_{RedSimpleTCP,8} \parallel \text{ATS}_{RedSimpleTCP,2} = \{$

$$aOpen_1/-_0 \bullet Close_2/-_0 \bullet Close_1/-_0$$

$\}$

Step 2.9: Execute the test.