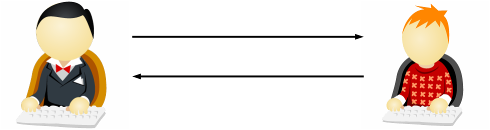
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Constructing Mid-points for Two party Asynchronous Protocols

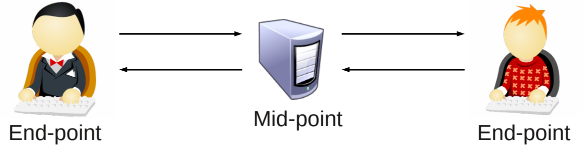
Petar Tsankov, Mohammad Torabi Dashti, David Basin
ETH Zürich

OPODIS’11
December 16, 2011

Protocols, end-points, mid-points



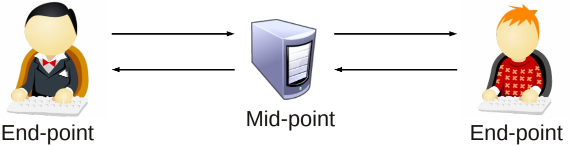
Protocols, end-points, mid-points



Mid-points:

- relay, redirect, filter communication

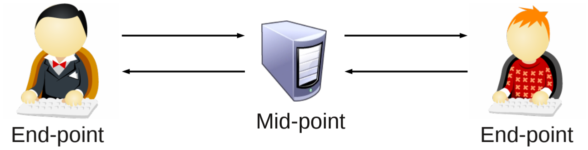
Protocols, end-points, mid-points



Mid-points:

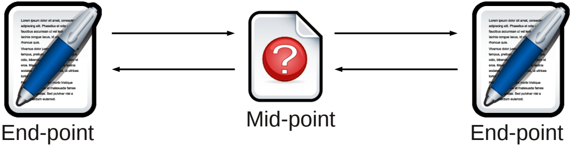
- relay, redirect, filter communication
- can ***enforce*** a protocol (e.g. stateful firewalls)

How to implement a mid-point?



We need a specification!

How to implement a mid-point?

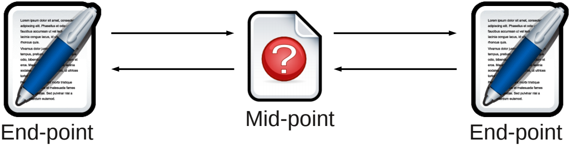


We need a specification!

Protocols specifications:

- specify the end-points' behavior
- ***do not*** specify the mid-point's behavior

How to implement a mid-point?



We need a specification!

Protocols specifications:

- specify the end-points' behavior
- ***do not*** specify the mid-point's behavior

The problem

How do we implement a system, when we don't know what it should do?

Why mid-point specifications?

Mid-points are often incorrectly implemented [1]:

- Checkpoint, netfilter/iptables, ISA Server

[1] Case study by D. Bidder-Senn, D. Basin, G. Caronni. *"Midpoints versus endpoints: From protocols to firewalls"*

Why mid-point specifications?

Mid-points are often incorrectly implemented [1]:

- Checkpoint, netfilter/iptables, ISA Server

Mid-point specifications are useful for:

- Model-driven development
- Code inspection
- Model-based testing

[1] Case study by D. Bidder-Senn, D. Basin, G. Caronni. *"Midpoints versus endpoints: From protocols to firewalls"*

Why mid-point specifications?

Mid-points are often incorrectly implemented [1]:

- Checkpoint, netfilter/iptables, ISA Server

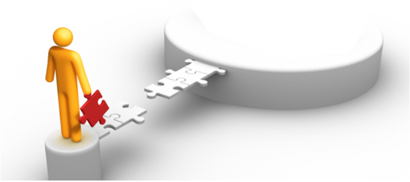
Mid-point specifications are useful for:

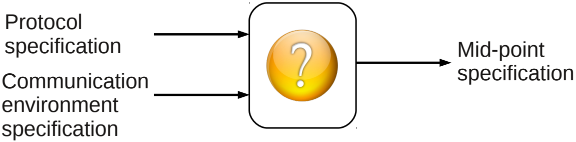
- Model-driven development
- Code inspection
- Model-based testing

. . . they are a good starting point to implement a mid-point

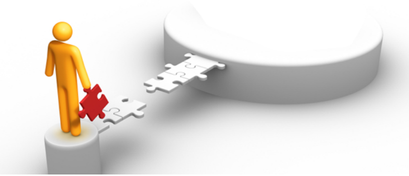
[1] Case study by D. Bidder-Senn, D. Basin, G. Caronni. *"Midpoints versus endpoints: From protocols to firewalls"*

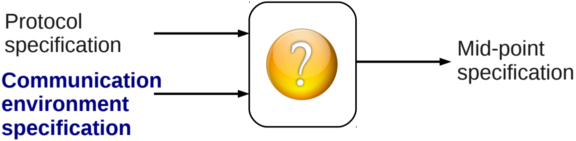
Goal




Protocol specification
Communication environment specification
Mid-point specification

Goal

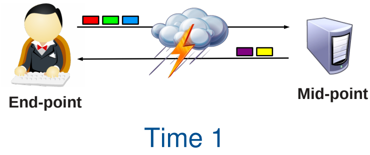


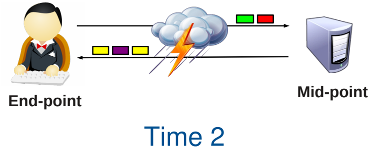

Protocol specification
Communication environment specification
Mid-point specification

Roadmap

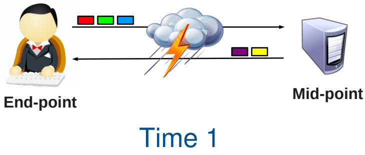
✓ Context, motivation, goals
- Challenges
- The model
- Framework
- TCP case study
- Future work

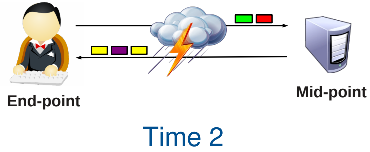
Challenge: Channels fidelity





Challenge: Channels fidelity

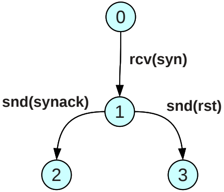




channel \ property	lose	duplicate	reorder
Reliable	no	no	no
Resilient	no	yes	yes
Lossy	yes	no	yes

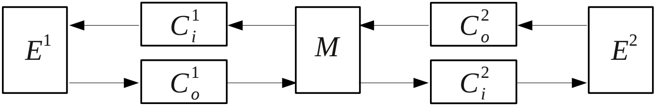
Challenge: Non-determinism

- Under-specification
  - allow alternative behaviors



- Abstraction
  - probabilistic choices

The setting



- E^1, E^2: the end-points
- $C_o^1, C_i^1, C_o^2, C_i^2$: channels

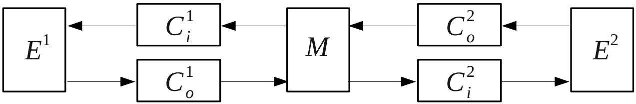
Assumption

The end-points and the channels are formally specified

We need to compute M

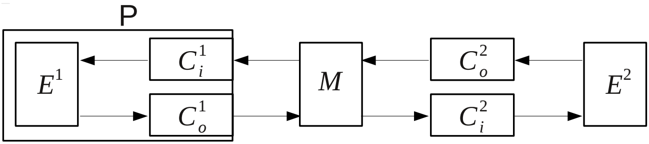
Process algebraic specifications

- End-points and channels are specified μCRL
  Benefits: General purpose process algebra with mature tool support



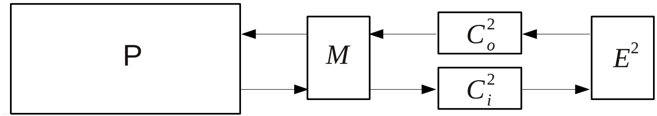
Process algebraic specifications

- End-points and channels are specified μCRL
  Benefits: General purpose process algebra with mature tool support
- We can compute the parallel composition of processes
  Example: $P = E^1 \| C_i^1 \| C_o^1$



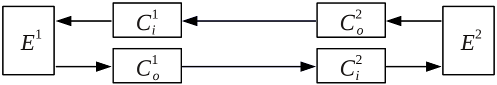
Process algebraic specifications

- End-points and channels are specified μCRL
  Benefits: General purpose process algebra with mature tool support
- We can compute the parallel composition of processes
  Example: $P = E^1 \| C_i^1 \| C_o^1$



Definition of enforcement

- Reference model

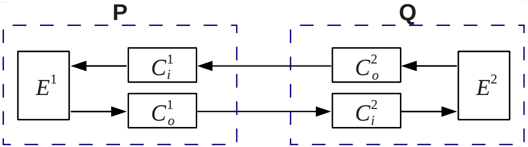


Definition of enforcement

- Reference model

$P = E^1 \| C_i^1 \| C_o^1$

$Q = E^2 \| C_i^2 \| C_o^2$



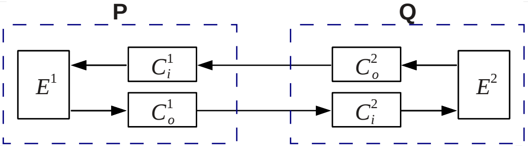
Definition of enforcement

- Reference model

$P = E^1 \| C_i^1 \| C_o^1$

$Q = E^2 \| C_i^2 \| C_o^2$

$R = P \| Q$

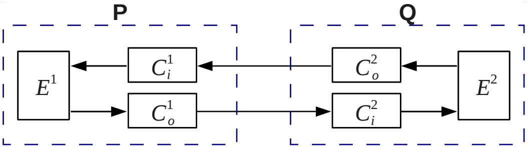


Definition of enforcement

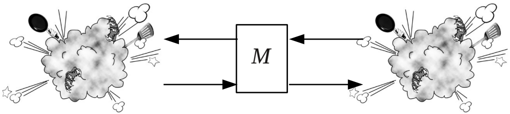
- Reference model

$$P = E^1 \| C_i^1 \| C_o^1$$
$$Q = E^2 \| C_i^2 \| C_o^2$$

$$R = P \| Q$$



- Implementation model

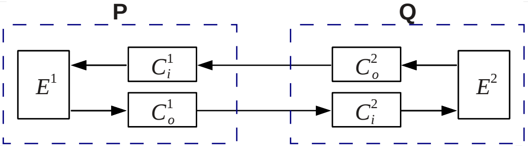


Definition of enforcement

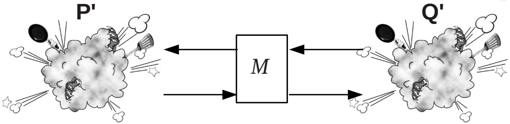
- Reference model

$$P = E^1 \| C_i^1 \| C_o^1$$
$$Q = E^2 \| C_i^2 \| C_o^2$$

$$R = P \| Q$$



- Implementation model

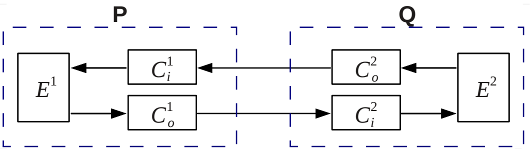


Definition of enforcement

- Reference model

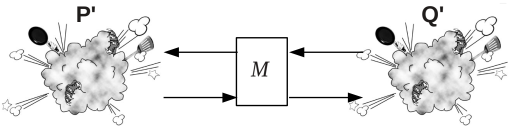
$$P = E^1 \| C_i^1 \| C_o^1$$
$$Q = E^2 \| C_i^2 \| C_o^2$$

$$R = P \| Q$$



- Implementation model

$$I = P' \| M \| Q'$$



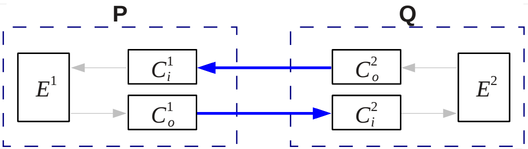
Definition of enforcement

- Reference model

$P = E^1 \| C_i^1 \| C_o^1$

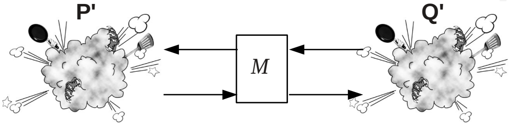
$Q = E^2 \| C_i^2 \| C_o^2$

$R = P \| Q$



- Implementation model

$I = P' \| M \| Q'$

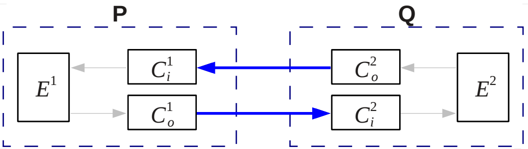


Definition of enforcement

- Reference model

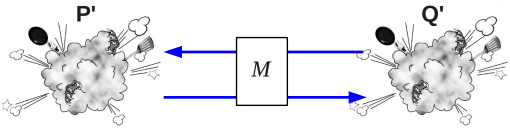
$$P = E^1 \| C_i^1 \| C_o^1$$
$$Q = E^2 \| C_i^2 \| C_o^2$$

$$R = P \| Q$$



- Implementation model

$$I = P' \| M \| Q'$$

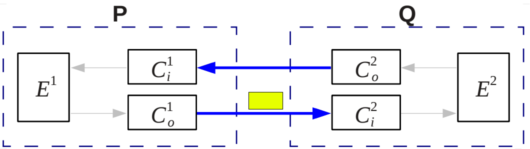


Definition of enforcement

- Reference model

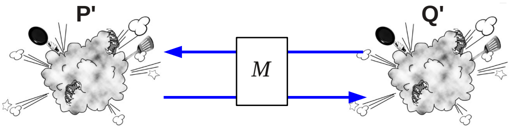
$$P = E^1 \| C_i^1 \| C_o^1$$
$$Q = E^2 \| C_i^2 \| C_o^2$$

$$R = P \| Q$$



- Implementation model

$$I = P' \| M \| Q'$$

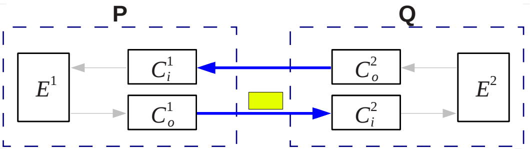


Definition of enforcement

- Reference model

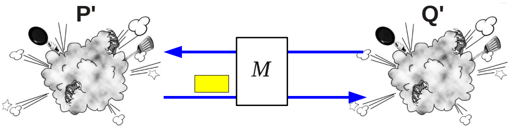
$$P = E^1 \| C_i^1 \| C_o^1$$
$$Q = E^2 \| C_i^2 \| C_o^2$$

$$R = P \| Q$$



- Implementation model

$$I = P' \| M \| Q'$$



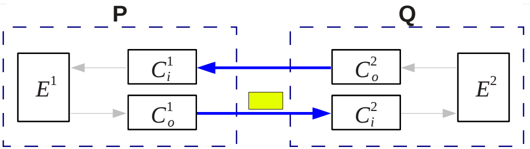
Definition of enforcement

- Reference model

$$P = E^1 \| C_i^1 \| C_o^1$$

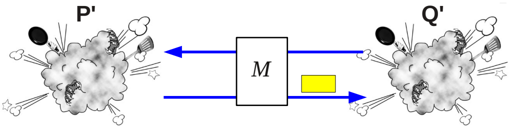
$$Q = E^2 \| C_i^2 \| C_o^2$$

$$R = P \| Q$$



- Implementation model

$$I = P' \| M \| Q'$$



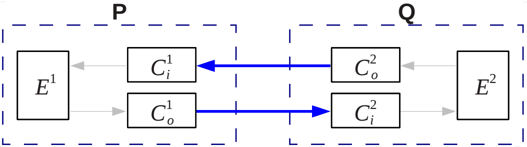
Definition of enforcement

- Reference model

$P = E^1 \| C_i^1 \| C_o^1$

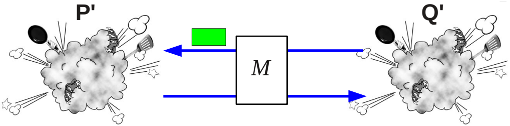
$Q = E^2 \| C_i^2 \| C_o^2$

$R = P \| Q$



- Implementation model

$I = P' \| M \| Q'$



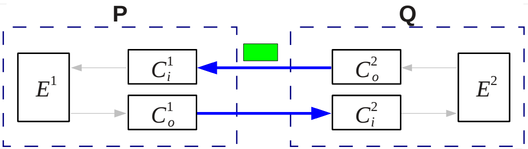
Definition of enforcement

- Reference model

$P = E^1 \| C_i^1 \| C_o^1$

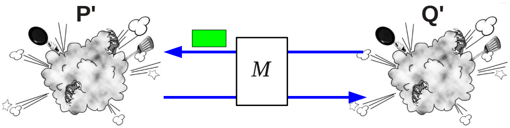
$Q = E^2 \| C_i^2 \| C_o^2$

$R = P \| Q$



- Implementation model

$I = P' \| M \| Q'$

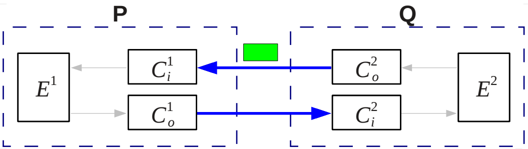


Definition of enforcement

- Reference model

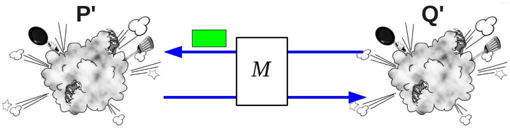
$$P = E^1 \| C_i^1 \| C_o^1$$
$$Q = E^2 \| C_i^2 \| C_o^2$$

$$R = P \| Q$$



- Implementation model

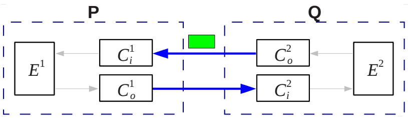
$$I = P' \| M \| Q'$$



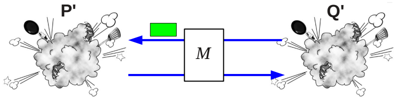
Definition: *Enforcement*

M enforces (E^1, E^2) iff $I \equiv_b R$

Computing the mid-point

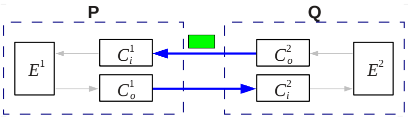


Reference model

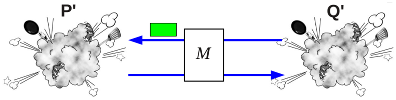


Implementation model

Computing the mid-point



Reference model



Implementation model

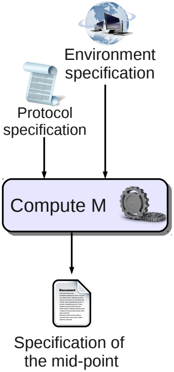
Observation: The mid-point is the reference model!

$$M := P \| Q$$

Theorem

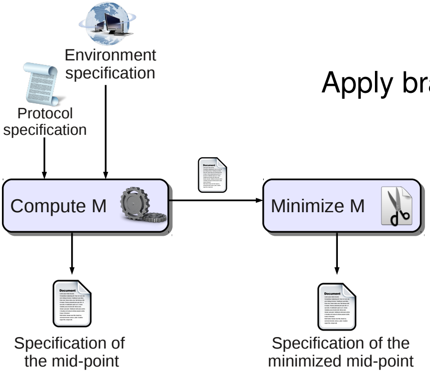
M enforces the protocol (E^1, E^2)

The framework



Compute $M = P \| Q$

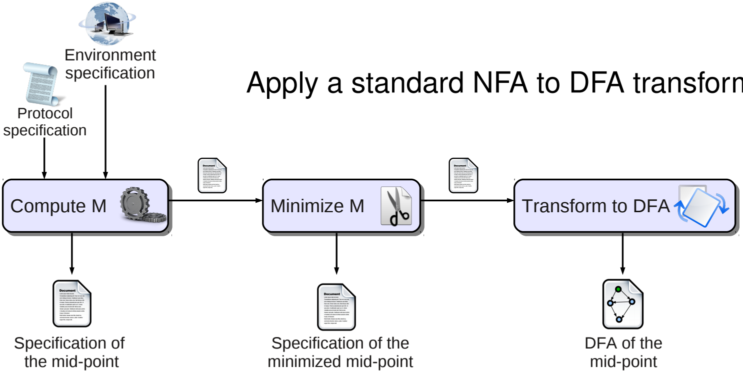
The framework



Apply branching bisimulation reduction

The framework

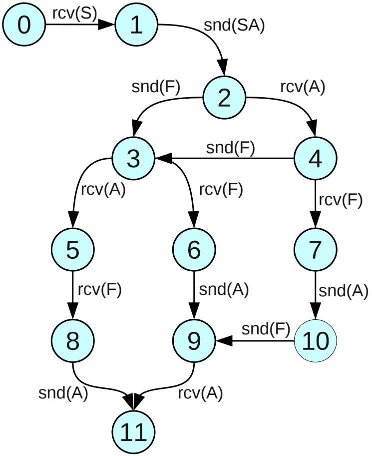
Apply a standard NFA to DFA transformation



Case study: TCP specification

We distinguish two TCP roles: initiator and responder

- Responder end-point



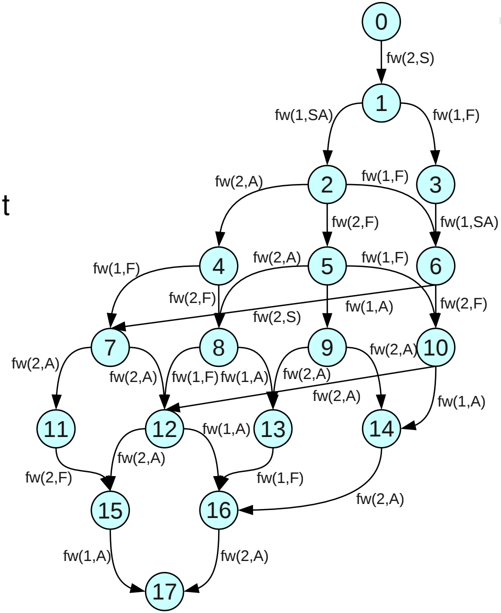
Input alphabet:

snd(msg), rcv(msg)

msg $\in \{S, SA, A, F\}$

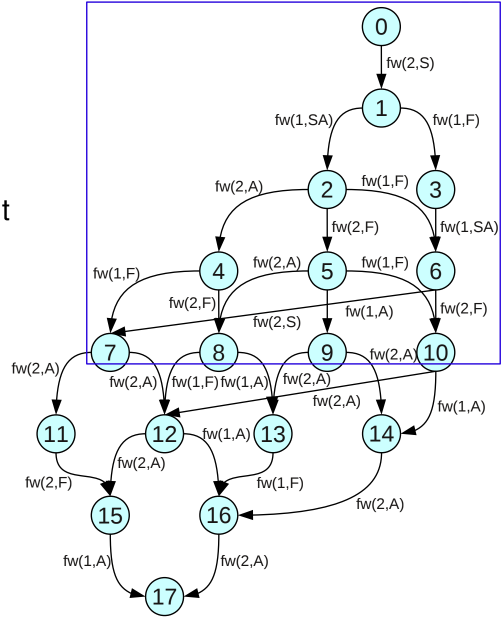
TCP mid-point

- E^1: initiator end-point
- E^2: responder end-point
- $C_o^1, C_i^1, C_o^2, C_i^2$: lossy channels
- Input alphabet: fw(id, msg)
  $msg \in \{S, SA, A, F\}$
  $id \in \{1, 2\}$



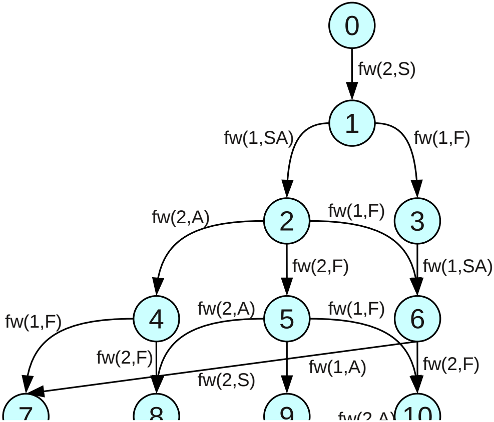
TCP mid-point

- E^1: initiator end-point
- E^2: responder end-point
- $C_o^1, C_i^1, C_o^2, C_i^2$: lossy channels
- Input alphabet:
  fw(id, msg)
  msg $\in \{S, SA, A, F\}$
  id $\in \{1, 2\}$



TCP mid-point

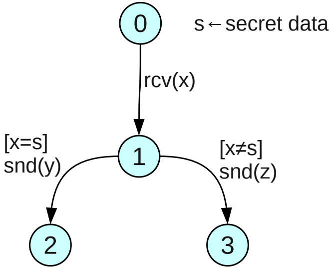
- E^1: initiator end-point
- E^2: responder end-point
- $C_o^1, C_i^1, C_o^2, C_i^2$: lossy channels
- Input alphabet: fw(id, msg)
  msg $\in \{S, SA, A, F\}$
  id $\in \{1, 2\}$



Future work

Secret data

- End-points (often) keep secret data (e.g. secret keys)
- Secret data is not exposed to the mid-point

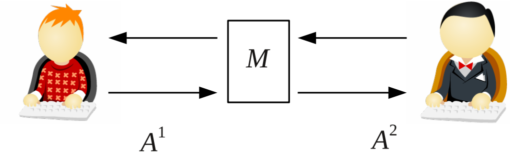


Branching bisimulation

A symmetric binary relation B over processes is a *branching bisimulation relation* iff $(P, P') \in B$ implies that for any action a, $P \xrightarrow{a} P_1$, then

- either $a = \tau$ and $(P_1, P') \in B$;
- or P' executes a sequence of (zero or more) silent actions $P' \xrightarrow{\tau} \cdots \xrightarrow{\tau} \hat{P}'$ such that $(P, \hat{P}') \in B$ and $\hat{P}' \xrightarrow{a} P_1'$ with $(P_1, P_1') \in B$.

Enforcing the protocol



$$R \sqsupseteq_r M \| A^1 \| A^2$$