

# **Laboratorio: Implementación del diagrama de transición de estados de TCP usando Walnut**

---

Laboratorio de Redes y Sistemas Distribuidos

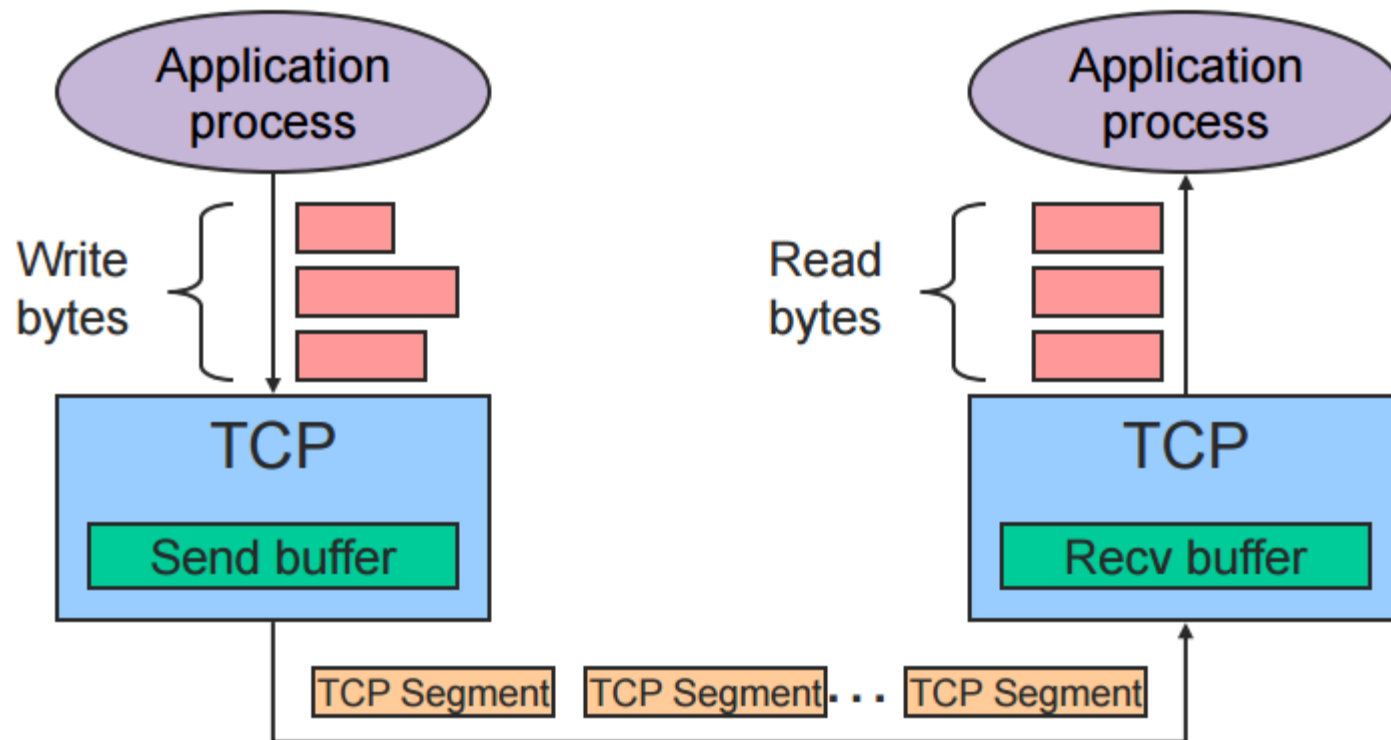
Sergio Canchi (2015)

FaMAF, UNC

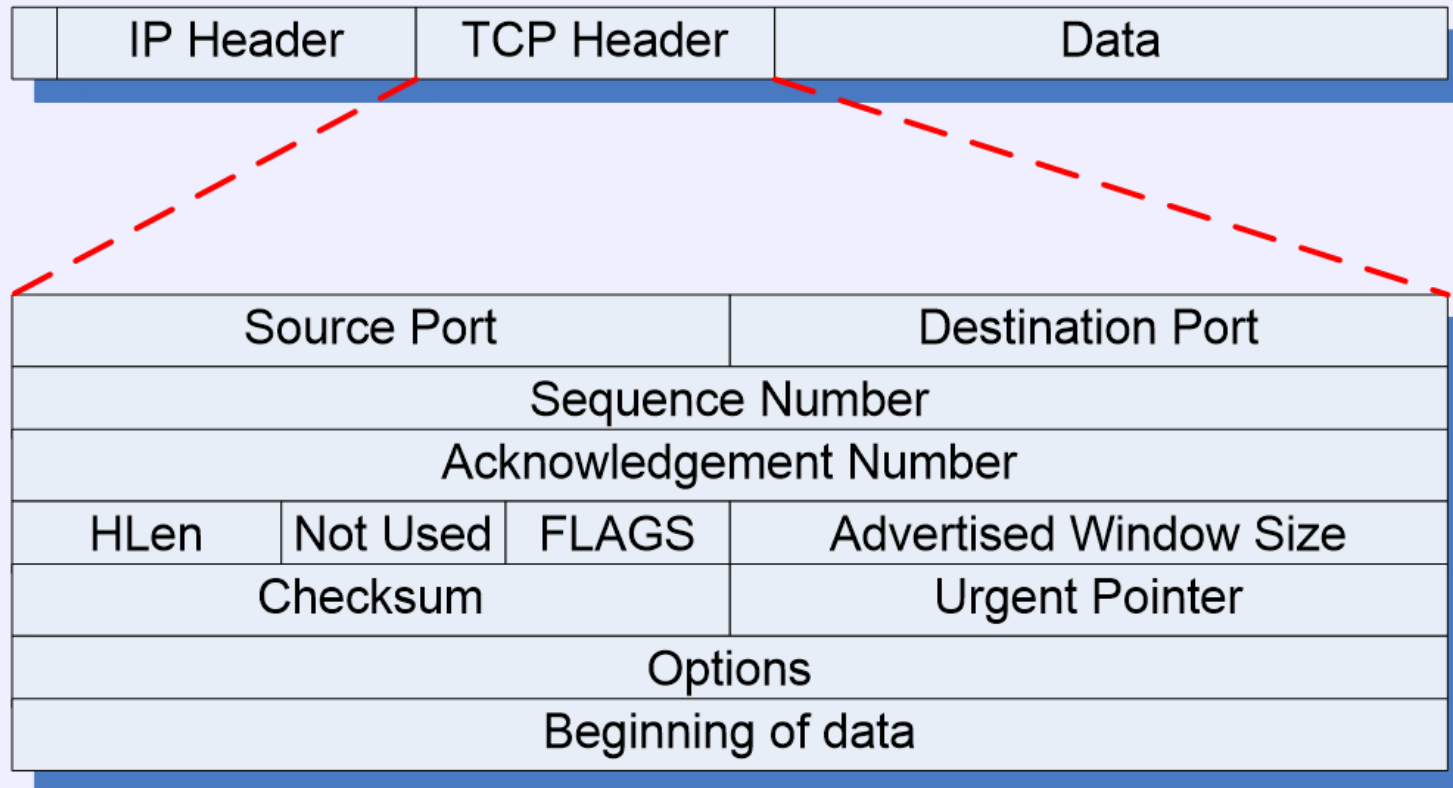
# Introducción

## TCP = Transmission Control Protocol

- Protocolo fundamental en internet
- Orientado a la conexión y provee entrega de datos confiable



# Encabezado TCP

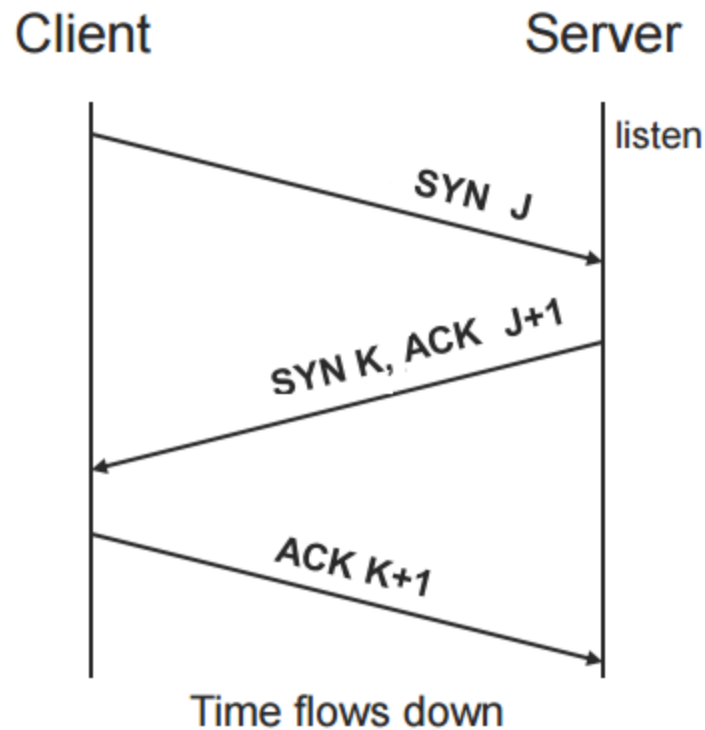


- Flags

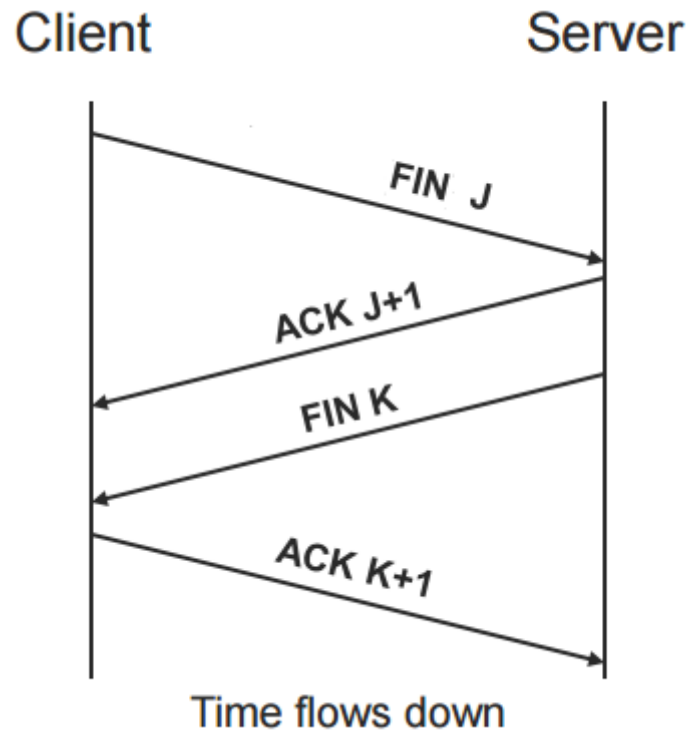
URG:	Contains urgent data	RST:	Reset connection
ACK:	Valid ACK seq. number	SYN:	Synchronize for setup
PSH:	Do not delay data delivery	FIN:	Final segment for teardown

# Establecimiento de una conexión TCP

- TCP usa una **negociación de tres pasos** para abrir una conexión:



# Terminación de una conexión TCP



# Transiciones de estados en TCP

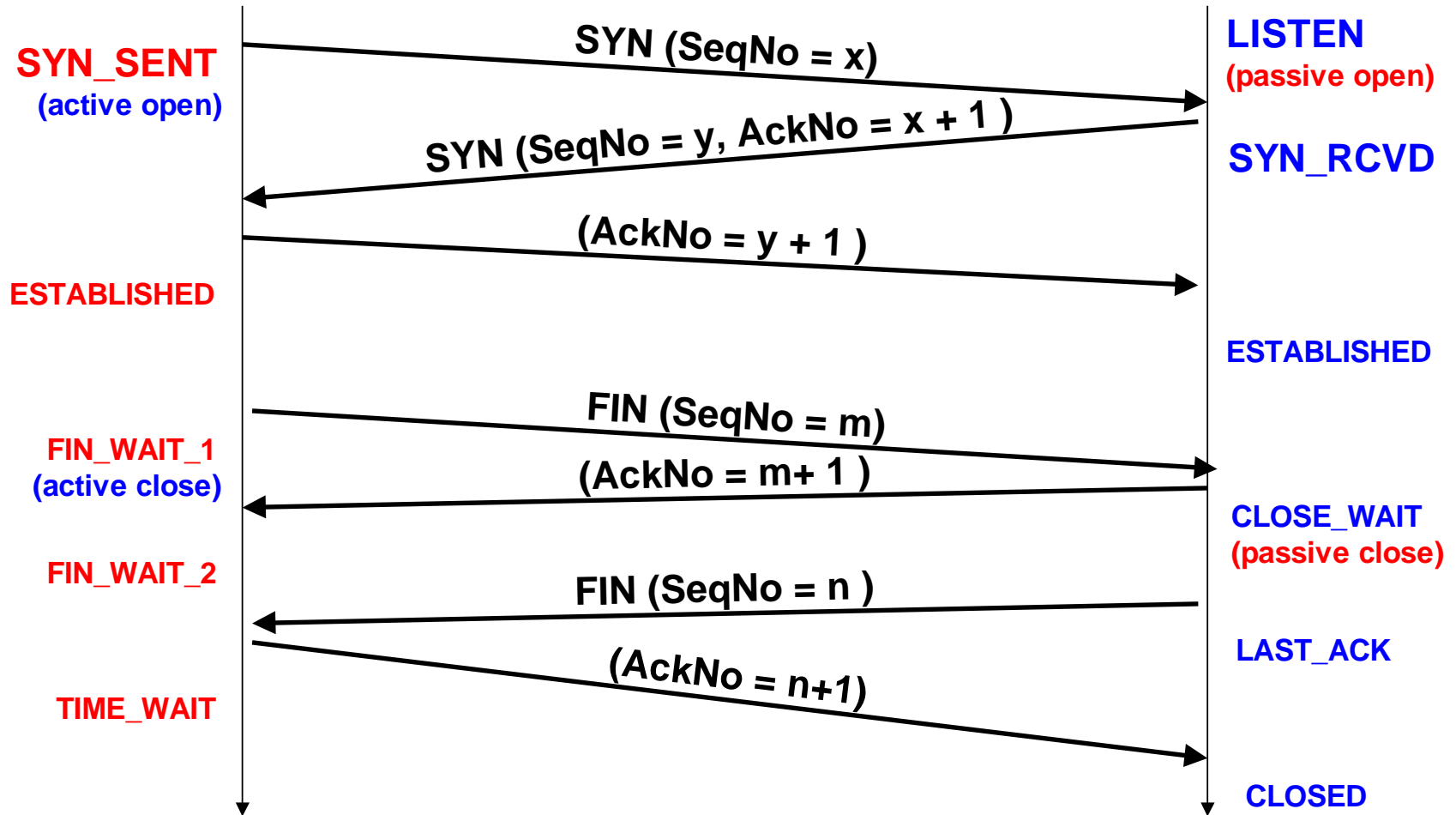
---

- **Cada conexión TCP mantiene un registro de su estado, lo que permite a la conexión progresar a través de varias transiciones hasta el cierre de la misma.**
- **Las transiciones de estado facilitan el establecimiento y terminación**
- **Toda la transferencia de datos se lleva a cabo dentro del estado Established**

# Estados TCP

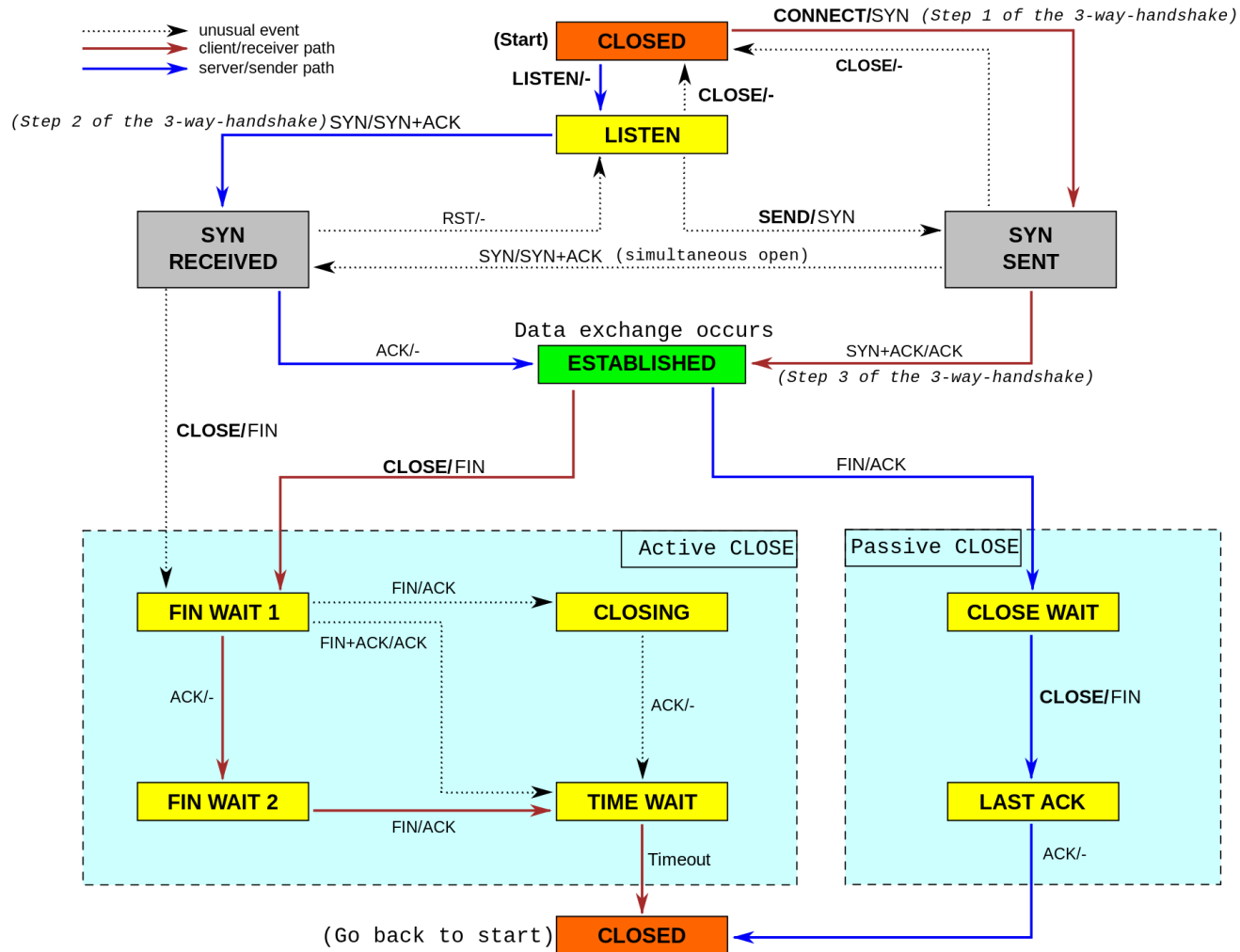
State	Description
CLOSED	No connection is active or pending
LISTEN	The server is waiting for an incoming call
SYN RCVD	A connection request has arrived; wait for Ack
SYN SENT	The client has started to open a connection
ESTABLISHED	Normal data transfer state
FIN WAIT 1	Client has said it is finished
FIN WAIT 2	Server has agreed to release
TIMED WAIT	Wait for pending packets (“2MSL wait state”)
CLOSING	Both Sides have tried to close simultanesously
CLOSE WAIT	Server has initiated a release
LAST ACK	Wait for pending packets

# Estados TCP durante el tiempo de vida de una conexión “típica”





# Diagrama de transición de estados de TCP



# Implementación de TCP STD usando Walnut

The screenshot shows the Walnut World editor interface. At the top, there's a browser window with the URL `aimara.machinalis.com/simulations/edit_world/96/`. The main area is divided into several sections:

- Name:** A text input field containing "TCP state transition diagram".
- Sharing:** Radio buttons for "Private" and "Public" (selected).
- World definition:** A code editor showing the following code:

```
1 type Address = String # A host+port
2
3 # A TCP connection progresses through a series of states during its lifetime. The states are
4 type S = Closed | Listen | SynReceived | SynSent | Established | FinWait1 | FinWait2 |
5         Closing | TimeWait | CloseWait | LastAck | Stop
6
7 # The TCP segment
8 type Flags(syn: Boolean, ack: Boolean, fin: Boolean, rst: Boolean)
9 type Message = Nothing | Segment(flags: Flags)
10
11 # Input for the state transition function
12 type P(my_address: Address, state: S)
13
14 state: Null
15
16 role Host(address: String, state: S)
17   sensor = P(agent.address, agent.state)
18   action: Action(sent_message: Message, new_state: S)
19   actuator_for Action(m, s)
20   do {
21     agent.state = s
22   }
23
```
- Problems:** A section with a "Run" button and a list of problems. One problem is visible: "Two hosts".
- Visualizations:** A section with a "Create new visualization" link and a list of visualizations. Three visualizations are visible: "State transition diagram for two hosts (Client + Server)", "State transition diagram for client", and "State transition diagram for server".
- Agents:** A section with a "Create new agent" link and a list of agents. One agent is visible: "State transition".

- Secciones
  - World definition
  - Problems
  - Visualizations
  - Agents

# Completar el agente “State Transition”

- Luego de “forkear” el proyecto (Ver los pasos en moodle)
- Ir a la sección Agents e ingresar a “Transition state”

The screenshot shows a web browser window with two tabs: "Walnut Home" and "Walnut Agent State transition". The address bar shows the URL `aimara.machinalis.com/simulations/edit_agent/144/`. The page title is "Agent: State transition". Below the title bar, there are navigation links: "Go to World: TCP state transition diagram", "Delete", "Docs Home", and a user profile "Sergio".

The main content area has three sections:

- Name:** A text input field containing "State transition".
- Language:** Two radio buttons: "Python" (selected) and "Javascript".
- Sharing:** Two radio buttons: "Private" and "Public" (selected).

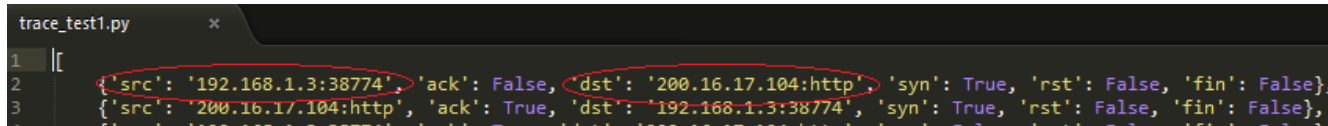
Below these is a **Code:** section with a text area containing Python code. The code is as follows:

```
39 flags += "A"
40 if m["fin"]:
41     flags += "F"
42 if m["rst"]:
43     flags += "R"
44 if m["syn"]:
45     flags += "S"
46
47 if m["dst"] == my_address:
48     # Caso mensaje recibido
49     # COMPLETAR: Actualizar la maquina de estado:
50     # A partir del estado actual y los flags TCP
51     # se debe deducir el nuevo estado del diagrama de estado
52     # - state.label obtiene un string con el nombre del estado
53     # por ej: state.label == "SynSent" determina si el estado es SynSent
54     # - Se debe usar el modulo agent.types para asignar el nuevo estado
55     # por ejemplo: state = agent.types.Established()
56     pass
57 elif m["src"] == my_address:
58     # Caso mensaje enviado
59     # COMPLETAR: Actualizar la maquina de estado:
60     # A partir del estado actual y los flags TCP se debe deducir el
61     # nuevo estado a transitar en el diagrama de transicion de estado
```

A green "Save" button is located at the bottom right of the code editor area.

# Configurar el problema “two hosts”

- Ir a la sección Problems e ingresar a “Two hosts”
- Modificar los campos “address” usando la información de la traza (valores de 'src' y 'dst' respectivamente). Por ejemplo



```
1 [
2   {'src': '192.168.1.3:38774', 'ack': False, 'dst': '200.16.17.104:http', 'syn': True, 'rst': False, 'fin': False},
3   {'src': '200.16.17.104:http', 'ack': True, 'dst': '192.168.1.3:38774', 'syn': True, 'rst': False, 'fin': False},
4   {'src': '192.168.1.3:38774', 'ack': True, 'dst': '200.16.17.104:http', 'syn': False, 'rst': False, 'fin': True},
5   {'src': '200.16.17.104:http', 'ack': True, 'dst': '192.168.1.3:38774', 'syn': False, 'rst': False, 'fin': True}
6 ]
```

## Agents

Agent ID client	Role Host
Program State transition	Initial state ▼ Role Host(...) address : 192.168.1.3:38774 state : Closed

Agent ID server	Role Host
Program State transition	Initial state ▼ Role Host(...) address : 200.16.17.104:http state : Listen

# Ejecutar la simulación de TCP STD

- Luego de completar el agente y configurar el problema, ir a la sección Problems y presionar “run”

Walnut Home x Walnut World TCP state transiti... x Walnut Trace visualization 1890 x +

aimara.machinalis.com/visualizer/visualize/1890/416/#time=2

Walnut Two hosts (June 11, 2015, 4:12 p.m.) — finished Sharing: private Go to ... Delete ... Docs Home Sergio

State transition diagram for server Save As Save Edit

Time steps

t	Agent	Perception	Action
1	"client"	P(my_address="192.168.1.3:38774", state=Closed)	Action(sent_message=N, new_state=Stop)
2	"server"	P(my_address="200.16.17.104:http", state=Listen)	Action(sent_message=N, new_state=Stop)

end:  
• server: 0  
• client: 0

Console  
Sprites

On state: 2 of 3