Lab Introduction 2

Continued

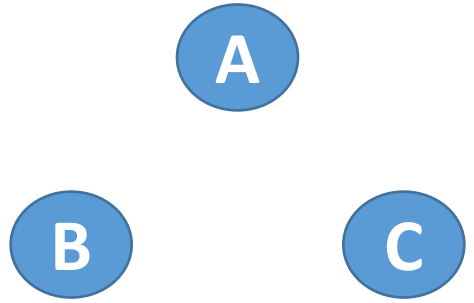
Agenda

- Lab 2 Introduction
- Solution cost for Lab 1
- Mininet Python API (optional)
- A small note on threads

Solution cost

- We care about the solution cost in terms of **communication**
- We can measure the cost of a solution (e.g. in Lab 1) in terms of
  - **number of nodes** to which a new post is propagated
  - **payload**: number of blackboard entries per message

- For example, consider the case of three nodes A, B and C, and the following events:
  - Event 1: User posts “msg1” to node A.
  - Event 2: User posts “msg2” to node A.

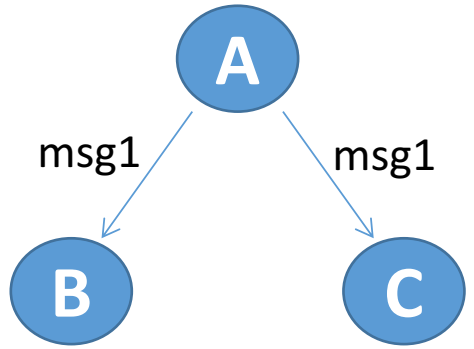


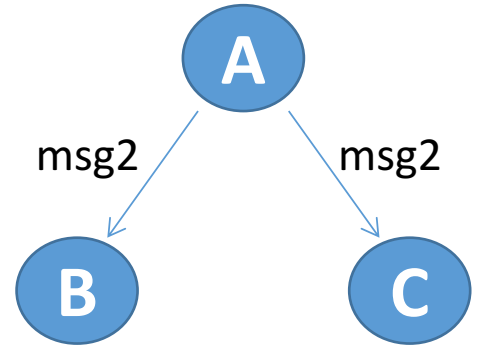
Example: a good scenario

- Propagate only the new post.
- Upon Event 1,
  Vessel A sends “msg1” to vessels B and C
  - Payload for each message = 1
  - Overall cost = 2

- Upon Event 2,
  Vessel A sends “msg2” to vessels B and C
  - Payload for each message = 1
  - Overall cost = 2

- Overall cost per post: *#nodes-1*



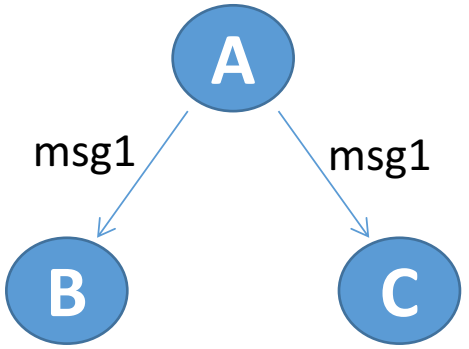


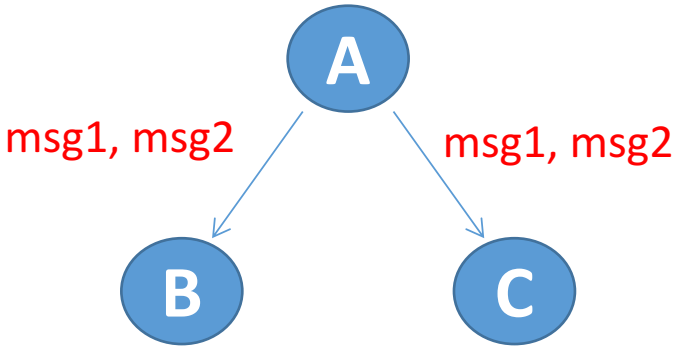
Example: a costly scenario

- Propagate the whole blackboard
- Upon Event 1, Vessel A sends “msg1” to vessels B and C
  - Payload for each message = 1
  - Overall cost = 2

- Upon Event 2, Vessel A sends “msg1, msg2” to vessels B and C
  - Payload for each message = 2
  - Overall cost = 4

- Overall cost per post: *b(n-1)*
  - board size *(#nodes-1)





Cost of Lab2

- We want you do the same simple communication cost analysis for Lab2:
  - First for the leader election part.
    - What is the cost of the whole leader election?
  - Then for the centralized blackboard.
    - What is the cost of a new post?

Agenda

- Lab 2 Introduction
- Solution cost for Lab 1
- Mininet Python API (optional)
- A small note on threads

Mininet Python API

- Staff that are not really required for the labs, but it is good to know ☺
- You can use a Python API to:
  - Set up your own topologies.
  - Configure link properties e.g. bandwidth, delay etc.
  - Run arbitrary commands on the hosts.
  - much more…
- …all through python scripts.
- For example, *lab1.py* uses that API.

Example 1: Building a custom topology (from the Mininet tutorial)

Defines a topology with n nodes connected to a single switch

```
class SingleSwitchTopo(Topo):

    def build(self, n=2):
        switch = self.addSwitch('s1')
        # Python's range(N) generates 0..N-1
        for h in range(n):
            host = self.addHost('h%s' % (h + 1))
            self.addLink(host, switch)
```

Example 1: Building a custom topology (from the Mininet tutorial)

Defines a topology with n nodes connected to a single switch

```
class SingleSwitchTopo(Topo):

    def build(self, n=2):
        switch = self.addSwitch('s1')
        # Python's range(N) generates 0..N-1
        for h in range(n):
            host = self.addHost('h%s' % (h + 1))
            self.addLink(host, switch)
```

Inherit class *Topo* and override method *build*

add a switch

create a host

link it to the switch

Example 1: Building a custom topology
(from the Mininet tutorial)

Defines a topology with n nodes connected to a single switch

```
class SingleSwitchTopo(Topo):

    def build(self, n=2):
        switch = self.addSwitch('s1')
        # Python's range(N) generates 0..N-1
        for h in range(n):
            host = self.addHost('h%s' % (h + 1))
            self.addLink(host, switch)
```

Starts Mininet with the specified topology

```
def simpleTest():

    topo = SingleSwitchTopo(n=4)
    net = Mininet(topo)
    net.start()
    print "Dumping host connections”
    dumpNodeConnections(net.hosts)
    print "Testing network connectivity"
    net.pingAll()
    net.stop()
```

Example 2: Configuring link parameters (from the Mininet tutorial)

```
class SingleSwitchTopo(Topo):

    def build(self, n=2):
        switch = self.addSwitch('s1')
        # Python's range(N) generates 0..N-1
        for h in range(n):
            host = self.addHost('h%s' % (h + 1))
            self.addLink( host, switch, bw=10, delay='5ms', loss=2 )
```

configure link properties

bandwidth in Mbs

delay in ms

loss rate %

Example 2: Configuring link parameters (from the Mininet tutorial)

Test the bandwidth between two nodes

```
h1, h4 =  net.get(  'h1',  'h4'  )
net.iperf( (h1, h4) )
```

or run any command on a node

```
result = h1.cmd('ifconfig')
print result
```

```
class SingleSwitchTopo(Topo):

    def build(self, n=2):
        switch = self.addSwitch('s1')
        # Python's range(N) generates 0..N-1
        for h in range(n):
            host = self.addHost('h%s' % (h + 1))
            self.addLink( host, switch, bw=10, delay='5ms', loss=2 )
```

Mininet Python API

- If you want to know more:
  1. here is walkthrough of the API https://github.com/mininet/mininet/wiki/Introduction-to-Mininet
  2. repo with many example scripts https://github.com/mininet/mininet/tree/master/examples
  3. Lots of videos about Mininet, making custom topologies etc. https://github.com/mininet/mininet/wiki/Videos
  4. or look at the script that we gave you for lab 1 (lab1.py).

Agenda

- Lab 2 Introduction
- Solution cost for Lab 1
- Mininet Python API (optional)
- A small note on threads

A small note about threads(1)

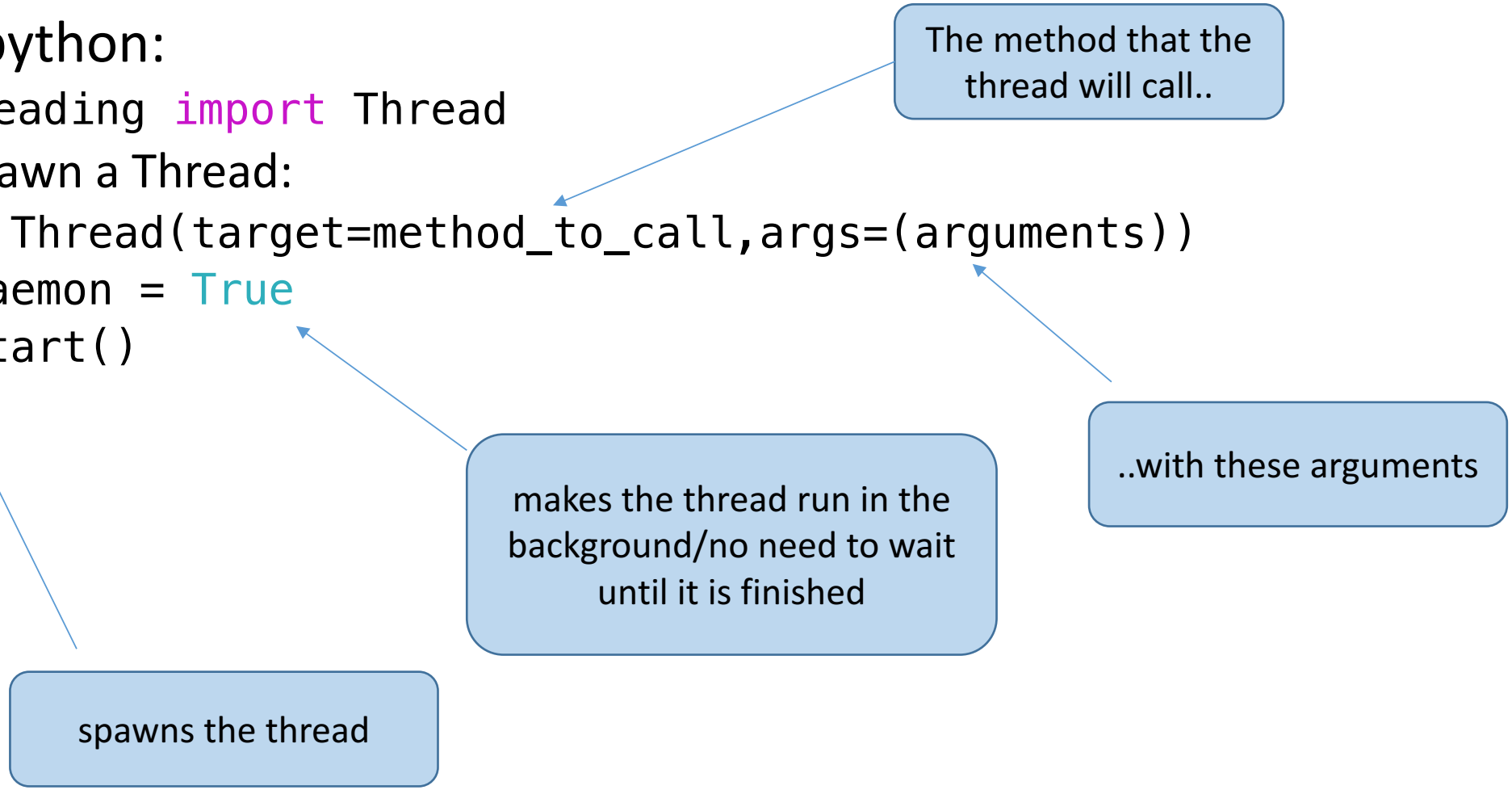
- At the *server.py* that we give you
  - Inside method *client_add_received*

```
# you should propagate something
# Please use threads to avoid blocking
#thread = Thread(target=???,args=???)
# you should create the thread as a deamon
```

- The idea is to offload the propagation of messages (*propagete_to_vessels*) to a different thread.

A small note about threads(2)

- Threads in python:
  - `from threading import Thread`
  - How to spawn a Thread:
    - 1) `t = Thread(target=method_to_call,args=(arguments))`
    - 2) `t.daemon = True`
    - 3) `t.start()`



A small note about threads(3)

- No need to worry much about it in the labs.
- Make sure you run *propagete_to_vesssels* on a different thread.

… but if you want to know more…

- the threading library:
  - https://docs.python.org/3/library/threading.html
- An interesting presentation (advanced):
  - https://www.slideshare.net/dabeaz/an-introduction-to-python-concurrency