Mini Project 2

thhk, emja, krbh, jglr

October 19th 2020

1 Instructions to run the program

To start a system of a service, sinks and sources, you start by running the service. Thereafter, you can freely connect and disconnect new sources or sinks as you please.

Service

Start the *Service* by running

```
java Service.java <sourcePort> <sinkPort>
```

where *sourcePort* and *sinkPort*, is the desired service port for incoming source and sink subscriptions, respectively.

Source

Start a *Source* by running

```
java Source.java <hostname> <sourcePort>
```

where *hostname* is the IP address[1] of the service and *sourcePort* is the service source subscription port.

Sink

Start a *Sink* by running

```
java Sink.java <hostname> <sinkPort>
```

where *hostname* is the IP address of the service and *sinkPort* is the service sink subscription port.

[1] If you are unsure about your local or public IP, you can run the convenience command: java IP.java, which uses an external web service for determing the public IP address.

Example of usage

The following example starts a server, then subscribes a new sink and a new source, and finally sends a message from the source, through the service and to the sink.

In terminal 1:

```
java Service.java 1000 2000
```

In terminal 2:

```
java Sink.java localhost 2000
```

In terminal 3:

```
java Source.java localhost 1000
Hello World!
```

Output of terminal 2 should be:

```
Hello World!
```

2 Choice of TCP

Every message is important for the system to live up to the specification, therefor it makes sense to use TCP because of its message delivery guarantee.

3 Are the processes a web service?

No XML is passed between the sinks and sources, so it is not a web service, according to the W3C definition of a web service. (Distributed Systems, Coulouris, 5th edition, p. 43)

4 Is the system time and/or space de-coupled?

The system is space de-coupled, as the sinks and sources don't know about each other. The service knows about their identities though.
The system is not time de-coupled, as the sources and sinks need to exist at the same time, for the sink to receive the message from the source.

5 Is the system a message queue?

No. Message queues allow one to one, but in the system when a source sends a message, it is received by all sinks.

6 Is the system a publish/subscribe system

Yes. A sink subscribes to the service, and a source publishes to the service.

7 Failure modes of the system

A source might send a message, and shortly after, a sink is closed. If the message is still in transit when the sink is closed, it will not be printed.

Byzantine failures may also occur, but these are difficult to document.