Networking with Node-RED

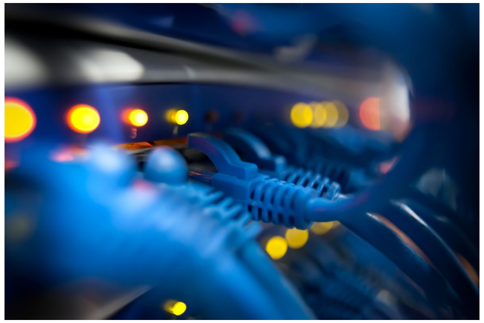
Luca Mottola
`luca.mottola@polimi.it`

(version 0.1)

Outline

- About networking
- UDP sockets
- MQTT

About Networking



- Node-RED offers **regular** networking abstractions
- Allowing Node-RED applications to interact with other applications
  - Either running on other Node-RED installations
  - Or using different platforms
- Many networking functionality are available, examples are
  - **UDP/TPC** sockets, mainly meant for pair-wise interactions
  - **HTTP/WebSockets**, for application-level interactions
  - **MQTT** mainly about many-to-many interactions

Data Serialization

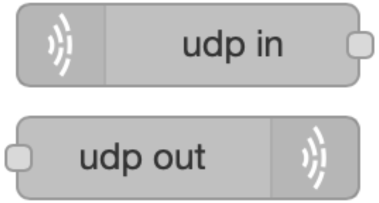
- Data structures are arbitrarily complex
  - They may include JavaScript built-in types, objects, arrays, …
  - The network, however, is akin to a pipe that can only transfer **serial data!**
- As data traverses a network, complex data structures are eventually **serialized**
- Often, we need to gain control on how this happens

UDP Sockets

UDP

- UDP is the User Datagram Protocol
  - A **lightweight Internet protocol** useful for sending small messages
  - It requires **no connection setup**: pack a message and send it off!
  - It also provides **no guarantees**: messages may get lost, arrive out of order…
- A UDP port is a number identifying different applications on a machine using UDP
  - When receiving UDP messages, we specify what **port we listen on**
  - When sending a UDP message, we **specify the destination IP address and port**

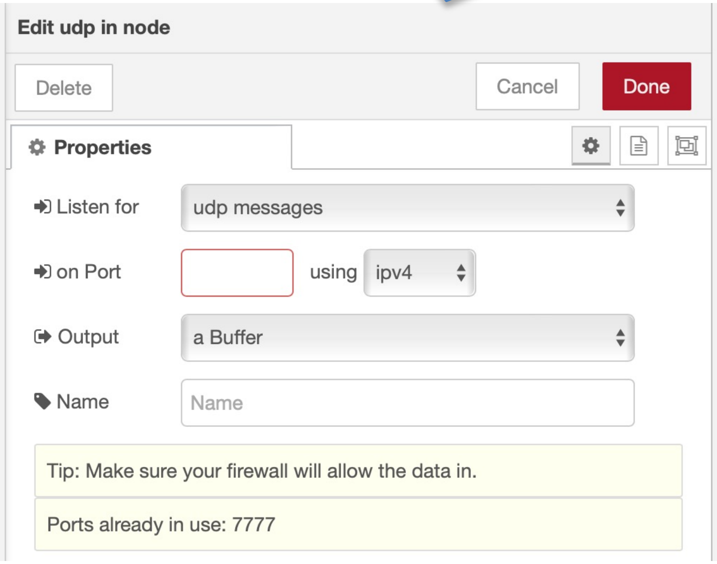
UDP Nodes



- Node-RED provides nodes for **receiving** and **sending** UDP messages
  - The content is taken from **`msg.payload`**!

When receiving UDP messages, we specify what port to listen on and how we want the output to be formatted

When sending UDP messages, we specify the destination IP address and port



Edit udp in node

Listen for: udp messages

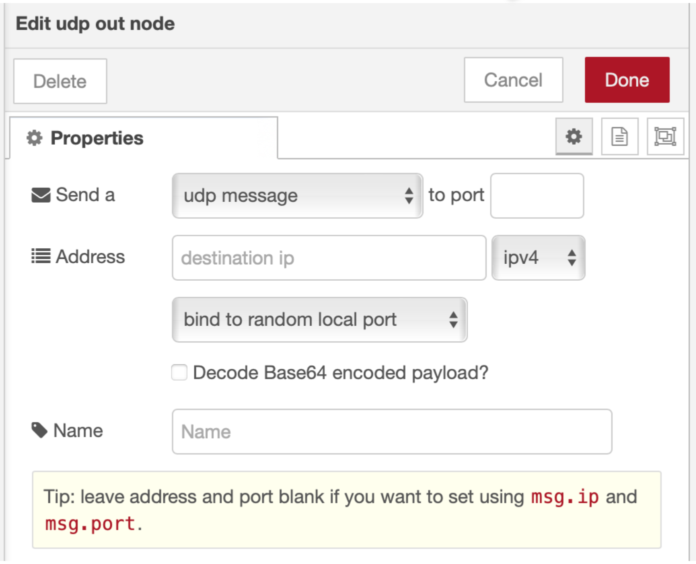
on Port: [ ] using ipv4

Output: a Buffer

Name: Name

Tip: Make sure your firewall will allow the data in.

Ports already in use: 7777



Edit udp out node

Send a: udp message to port [ ]

Address: destination ip ipv4

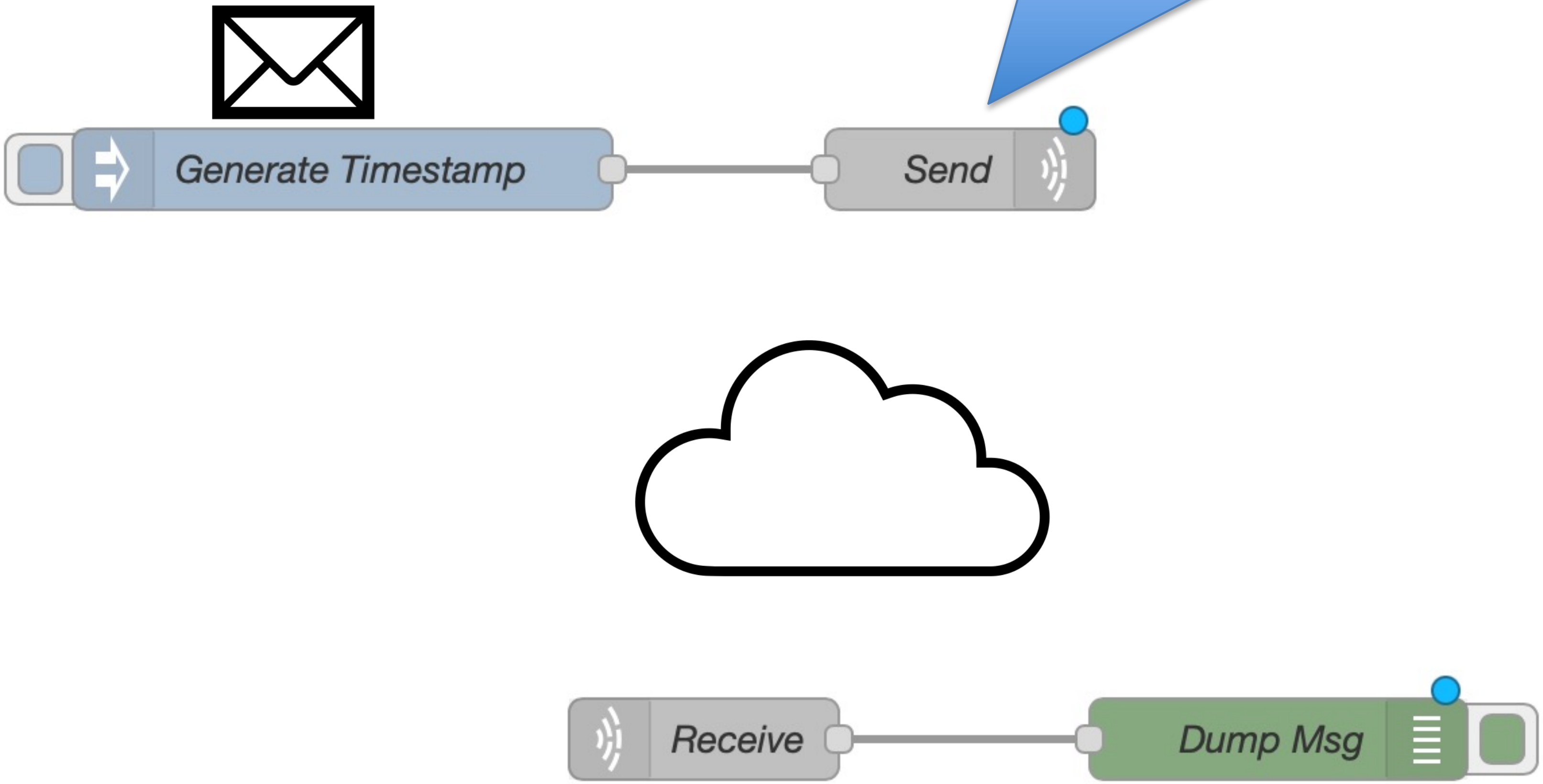
bind to random local port

□ Decode Base64 encoded payload?

Name: Name

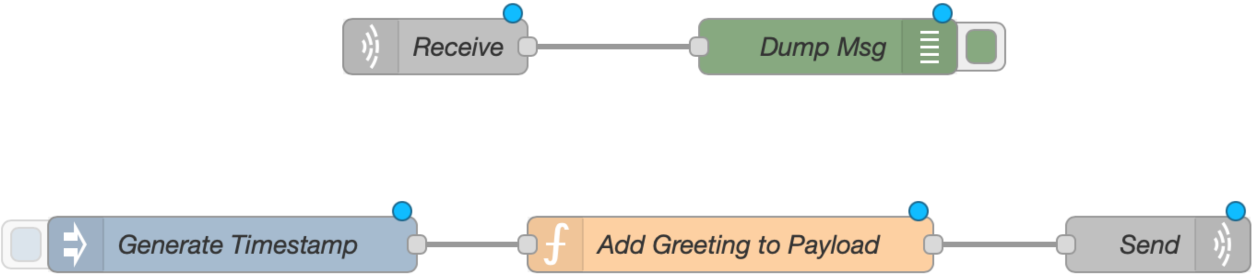
Tip: leave address and port blank if you want to set using `msg.ip` and `msg.port`.

UDP Example



Sending UDP messages to IP address 127.0.0.1, the flow sends the UDP messages to the local machine

Let’s Try Sending More



From a simple Numeric data type, the payload now contains an object with two fields… what happens?

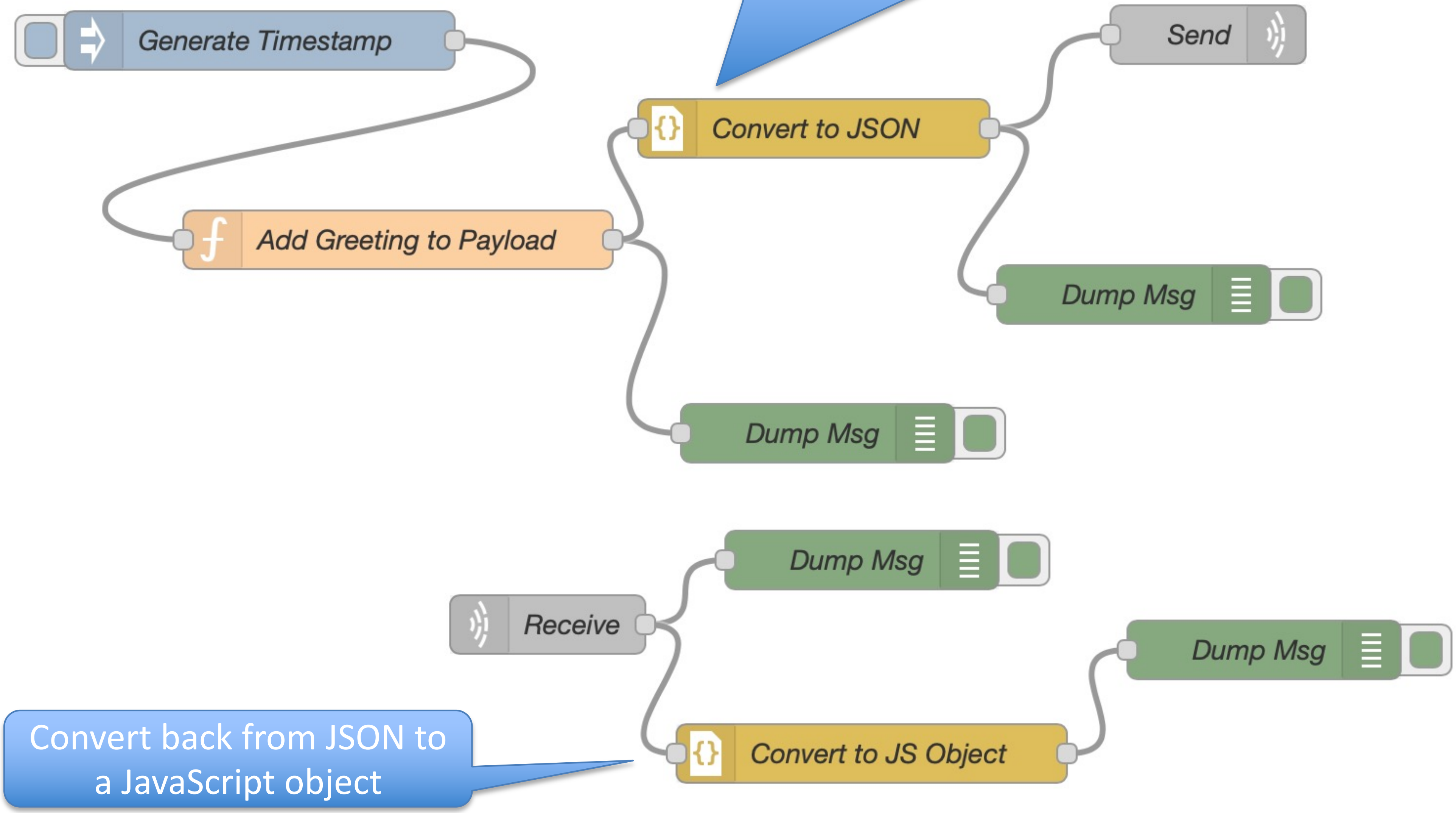
```
let newPayload = {
    greeting: "Hello!",
    timestamp: msg.payload,
};
let newMsg = {
    topic: msg.topic,
    payload: newPayload,
    _msgid: msg._msgid,

};
return newMsg;
```

JSON for Serialization

- JSON is an **open data interchange format** that uses human-readable text to store and transmit objects
  - Used a lot with JavaScript, but not only that!
  - JSON String
- Node-RED has a built-in node to convert to/from JavaScript objects and JSON
  - Can auto-detect what to convert from/to
  - Often gets it wrong though…
  - You can configure it manually

Using JSON

Convert from a JavaScript object, the payload in this case, to a JSON representation



Convert back from JSON to a JavaScript object

Find the code at: bit.ly/3KJID0O