cloudkick

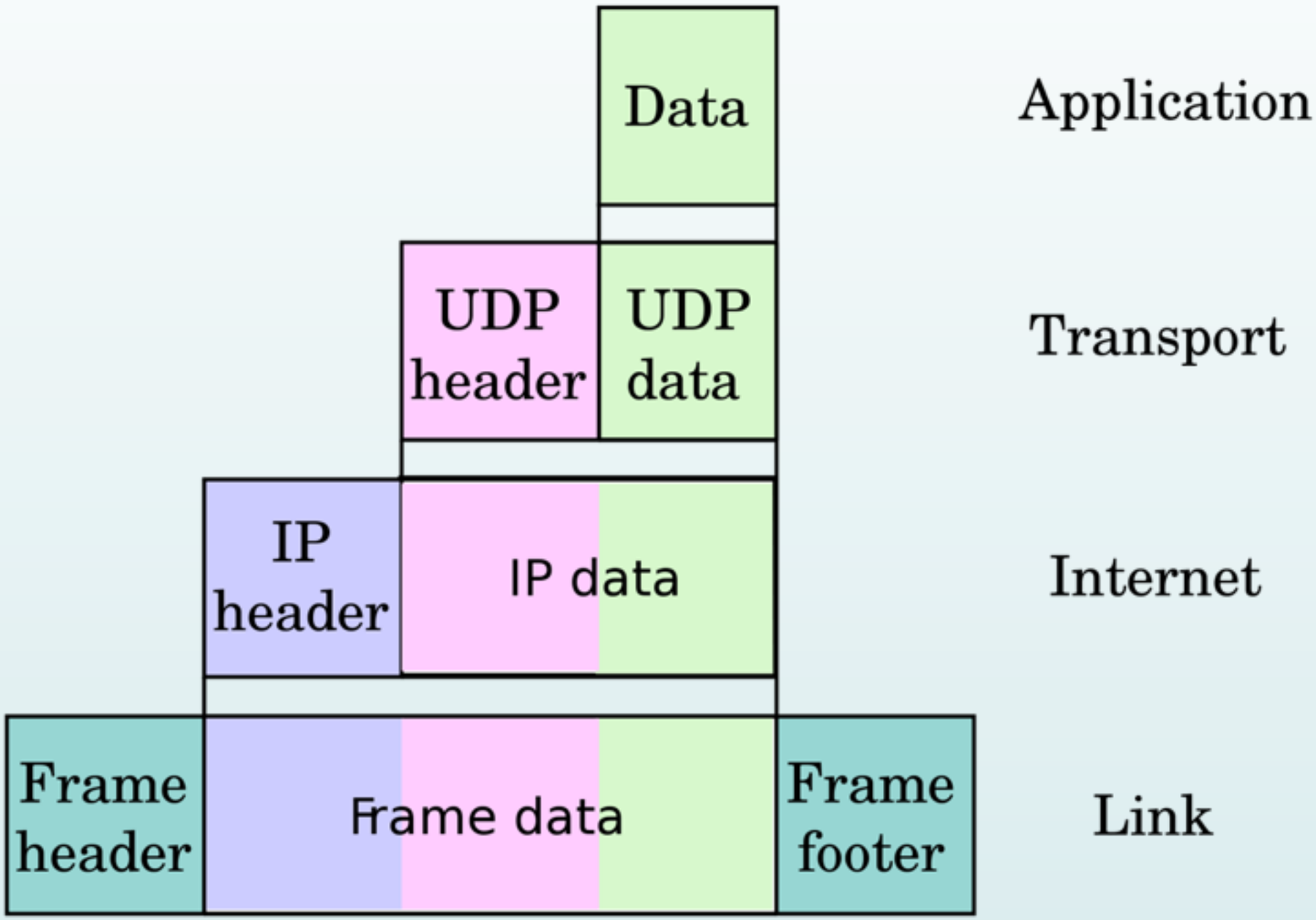
UDP in Node.js

Paul Querna
December 14, 2010

Code

- https://github.com/pquerna/node-examples
- s.apache.org/node-udp
- (same url)

Layers. Like Onions.



http://en.wikipedia.org/wiki/TCP/IP_model

TCP vs UDP

TCP

- Streams
- Ordered
- Reliable
- Unknown Latency

UDP

- Datagrams
- Unordered
- Unreliable
- Low Latency

TCP vs UDP

TCP Protocols

- HTTP
- SMTP
- BitTorrent
- SSH

UDP Protocols

- DNS
- DHCP
- UPnP / NAT-PMP
- Games / VoIP / Skype

Sending Hello

```
var Buffer = require('buffer').Buffer;
var dgram = require('dgram');

var sock = dgram.createSocket("udp4");
```

```
var buf = new Buffer("hello world");

sock.sendto(buf, 0,
            buf.length,
            8000, "127.0.0.1");
sock.close();
```

Receiving Hello

```
sock = dgram.createSocket("udp4", function (msg, r)
{
  log('got message from '+ r.address +':'+ r.port);
  log('data len: '+ r.size + " data: "+
msg.toString('ascii', 0, r.size));
  sock.close();
});

sock.bind(8000, '0.0.0.0');
```

```
$ node hello-server.js  &
$ node hello-client.js
13 Dec 14:58:41 - got message from 127.0.0.1
port: 52581
13 Dec 14:58:41 - data len: 11 data: hello world
```

Source of Packets

- Each Message can come from a different Peer.
- recvfrom() system call returns the source peer information, node exposes:
  - info.size // bytes read
  - info.port // from port
  - info.address // from address

UDP Event Emitters

- Does emit:
  - listening
  - message
  - error
- Does not emit:
  - connect, close, drain, end, secure, timeout, etc

Binary

- Node.js has a Buffer Type.
- With UDP, binary formats are more common
- TFTP is a simple example.

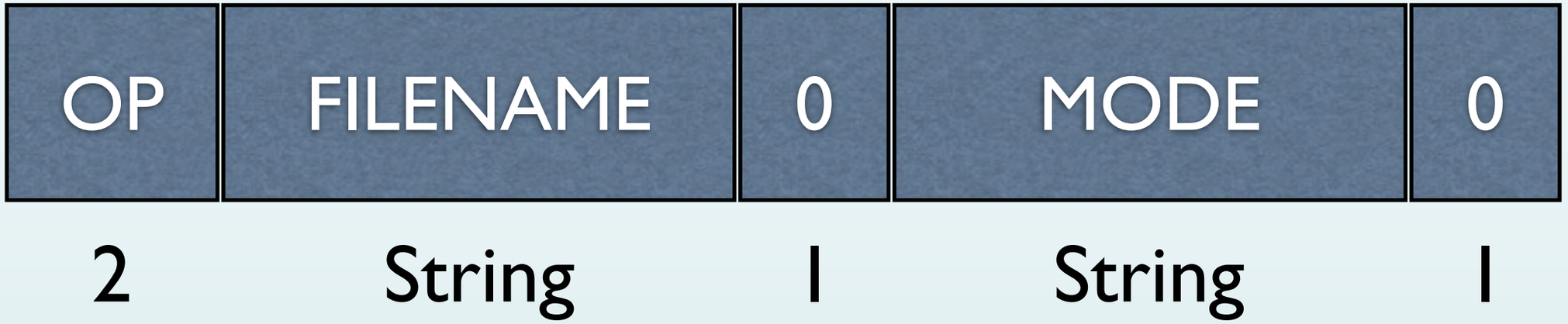
TFTP

- RFC 1350
- 5 operation codes, simple format.
- You probably last used it to brick (or unbrick) your router.

TFTP Opcodes

- 1: Read Request
- 2: Write Request
- 3: Data Chunk
- 4: Acknowledgment
- 5: Error

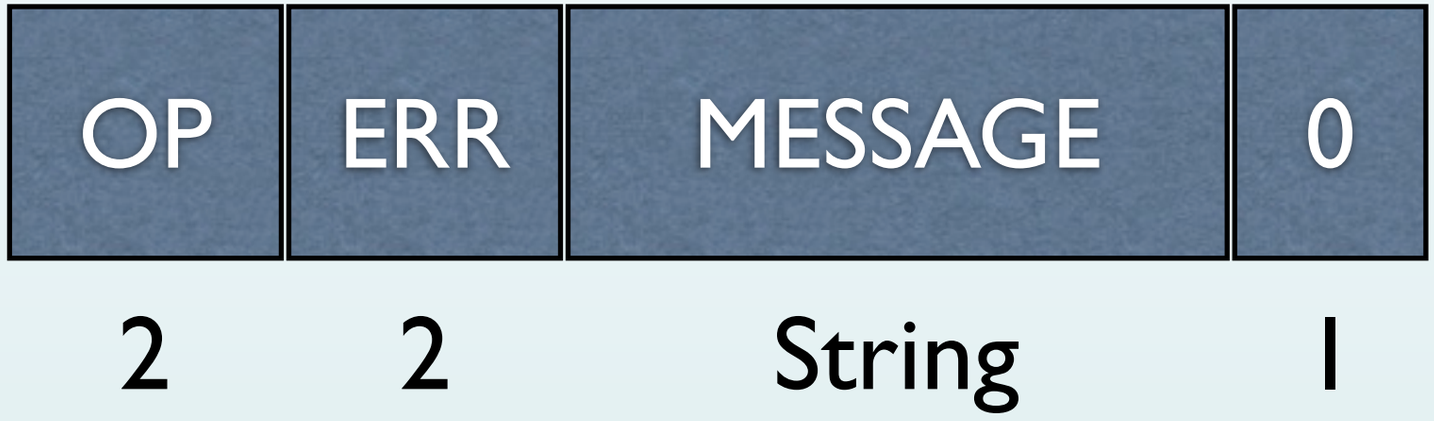
Requesting a File



Parsing a Request

```
// is a read request?
if (msg[0] == 0 && msg[1] == 1) {
  for (slen = 0; slen <msg.length; slen++) {
    if (msg[slen] === 0) break;
  }
  filename = msg.toString('ascii', 0, slen);
  // go send the file to them
}
```

Errors



Sending Errors

```
var buf = new Buffer(6 + msg.length);
buf[0] = 0;
buf[1] = 5;
buf[2] = 0;
buf[3] = errorcode;
buf.write(msg, 4);
buf[4 + msg.length] = 0;
sock.send(buf, 0, buf.length,
    peer.port, peer.address);
```

Sending Data

OP	BLOCK	DATA
2	2	512

Sending Data

```
fs.open(file, 'r', function(fp) {
var buf = new Buffer(4 + 512);
fs.read(fp, buf, 4, 512, (blcok - 1) * 512, function()
{
    buf[0] = 0;
    buf[1] = 3;
    buf[2] = (block >> 8) & 0xFF;
    buf[3] = block & 0xFF;
    sock.send(buf, 0, buf.length,
              peer.port, peer.address);
    fs.close(fp);
});
```

- tftp demo engage.

Libraries for Binary Data (use them!)

- node-jspack:
  - https://github.com/pgriess/node-jspack
- node-strtok:
  - https://github.com/pgriess/node-strtok
- node-bufferlist:
  - https://github.com/substack/node-bufferlist

UDP Chat Client

- Ping server they are alive
- Receive messages and print them
- Send messages from stdin

```
var sock = dgram.createSocket("udp4");

stdin.on('data', function (input) {
  var buf = new Buffer(input);
  sock.send(buf, 0, buf.length, SERVER_PORT,
SERVER_HOST);
});

sock.on('message', function (buf) {
  process.stdout.write(buf.toString());
});
```

UDP Chat Server

- Keep track of active peers
- Push messages to clients
- Log messages
- Provide bridge to HTTP

- activate chat demo go go go

Questions?

- Slides:
  - http://paul.querna.org/slides/
- Code:
  - https://github.com/pquerna/node-examples
- Cloudkick is hiring Node.js people!
  - https://www.cloudkick.com/careers