

MOBILE DEVELOPMENT

NETWORKING: PART 1

William Martin
Head of Product, Floored

NETWORKING

LEARNING OBJECTIVES

- › Describe of REST, HTTP, JSON, and other networking-related terms
- › Deploy NSURLSession to make HTTP requests
- › Parse data returned in JSON format (SwiftyJSON) and/or XML (SWXMLHash)

NETWORKING

INTRO TO NETWORKING

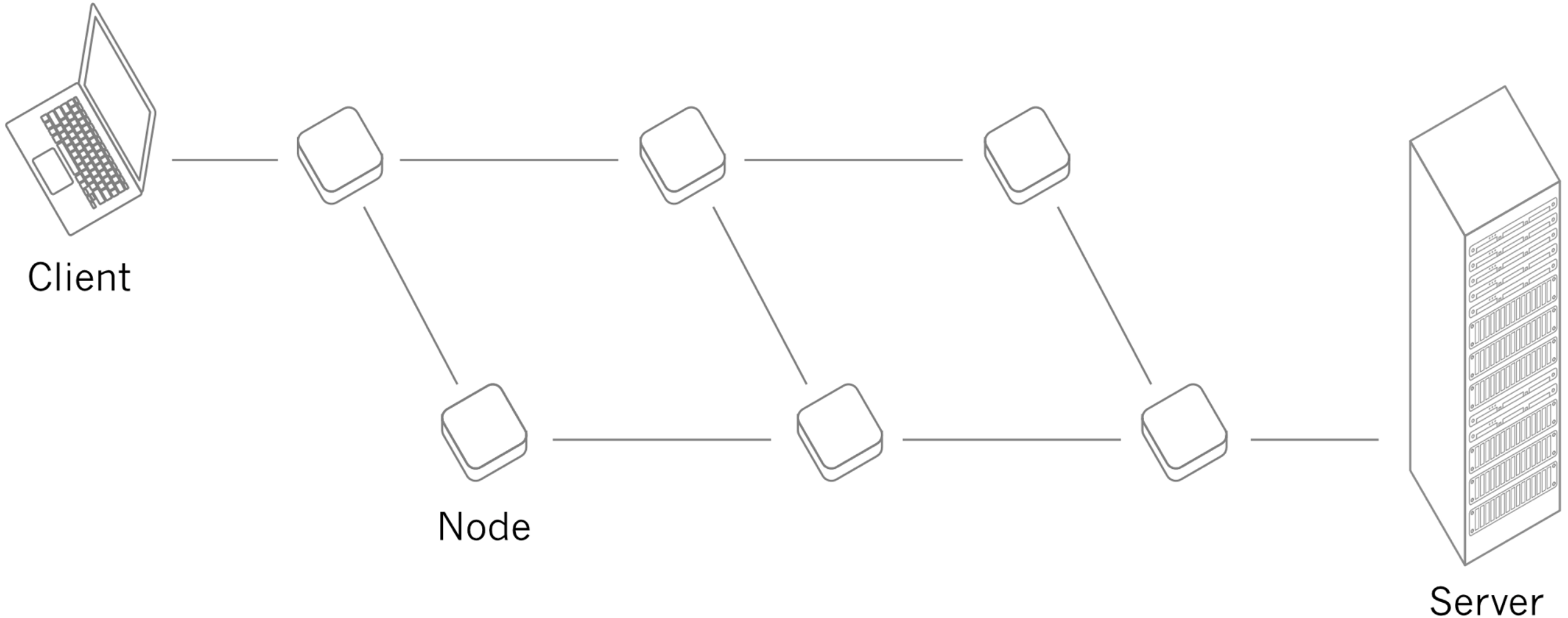
NETWORKING

WHAT IS A NETWORK?

- A network is a interconnected set of computers and devices.
- The Internet is essentially a network of networks.

NETWORKING

WHAT IS A NETWORK?



NETWORKING

EXAMPLES OF NETWORKED APPLICATIONS

- Apps that access online information (e.g. news reader)
- Mobile versions of productivity apps (e.g. Google docs)
- Social networking apps
- Multiple app endpoints with synced data (iPhone, Android, web, tablet, phone, API service, etc.)

NETWORKING

CONCEPTS

- Client-Server Architecture
 - Client – requester of a service
 - Server – provider of a service
- Peer-to-Peer
- Packets
- Request-Response

NETWORKING

INTERNET PROTOCOL SUITE

A set of communication protocols used over the Internet and similar networks.

- Application Layer
- Transport Layer
- Internet Layer
- Link Layer

NETWORKING

INTERNET PROTOCOL SUITE

Also known as TCP/IP, essentially the backbone of the Internet.

Determines how information is:

- Packetized
- Addressed
- Transmitted
- Routed
- Received

NETWORKING

WHAT IS A NETWORK PROTOCOL?

- › We've already talked about "Protocols" provided by Swift, which are a language feature that enables a class to act according to a required set of rules required by another class.
- › Network protocols are of the same ilk. They are essentially a set of rules of communication and data formatting for two entities to communicate over a network.
- › Communicating over a network and over the Internet (the largest network of networks in the world), requires a several protocols with different responsibilities.
- › Internet Protocol Suite - https://en.wikipedia.org/wiki/Internet_protocol_suite

NETWORKING

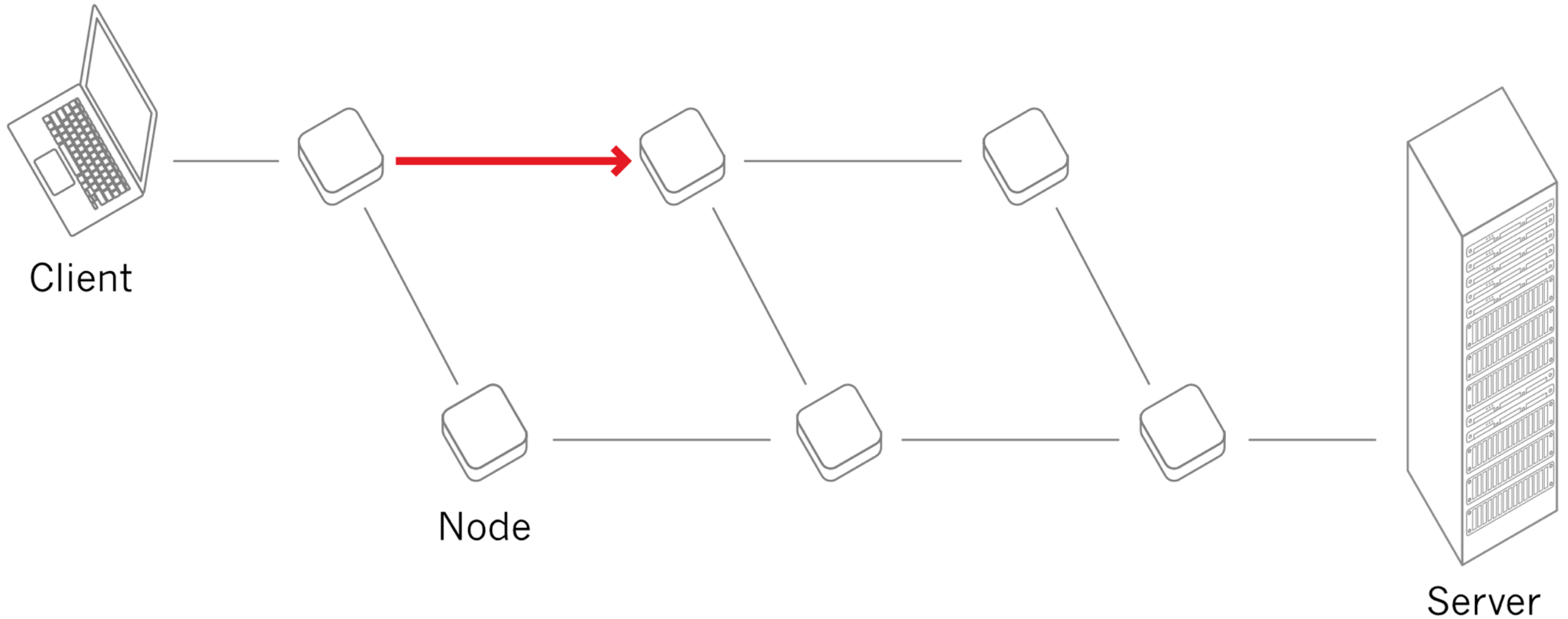
LINK LAYER

A communication protocol between adjacent nodes of a network.

- MAC – Media Access Control

NETWORKING

LINK LAYER



NETWORKING

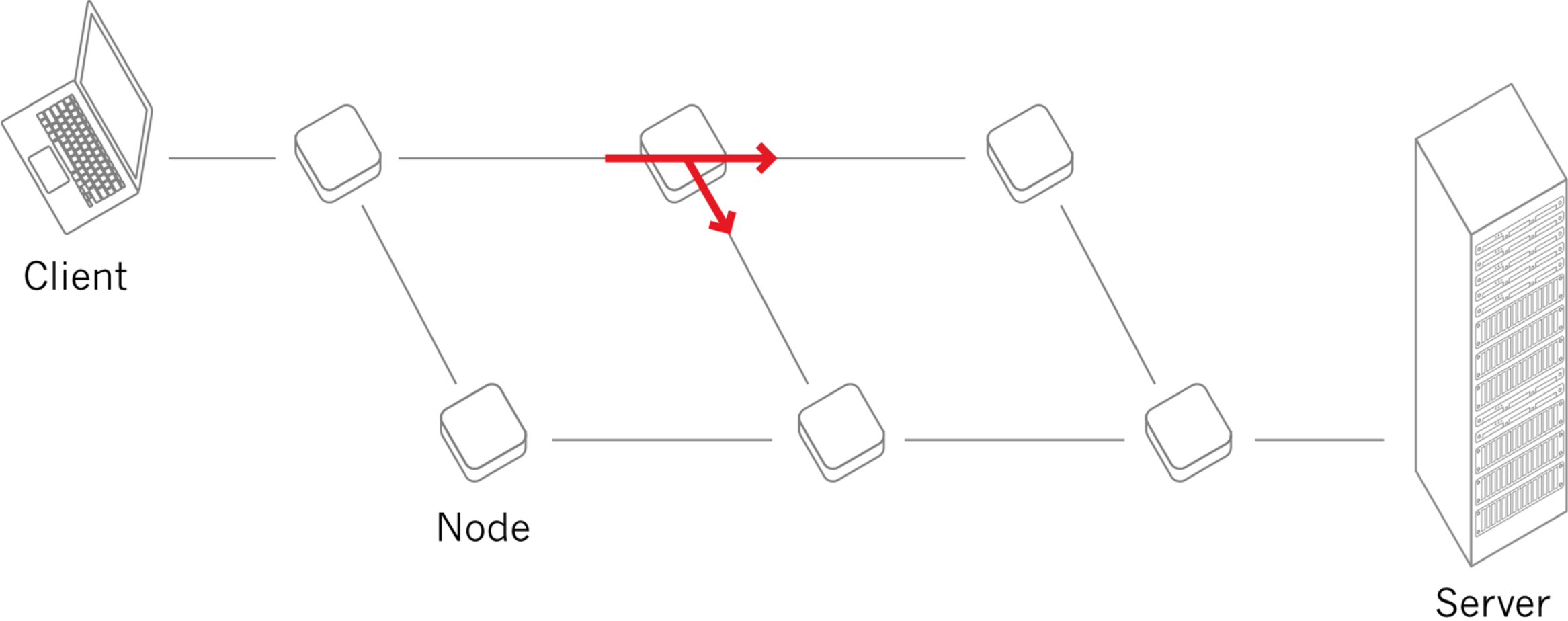
INTERNET LAYER

Provides addressing and routing services for hosts, especially across network boundaries.

- IP – Internet Protocol

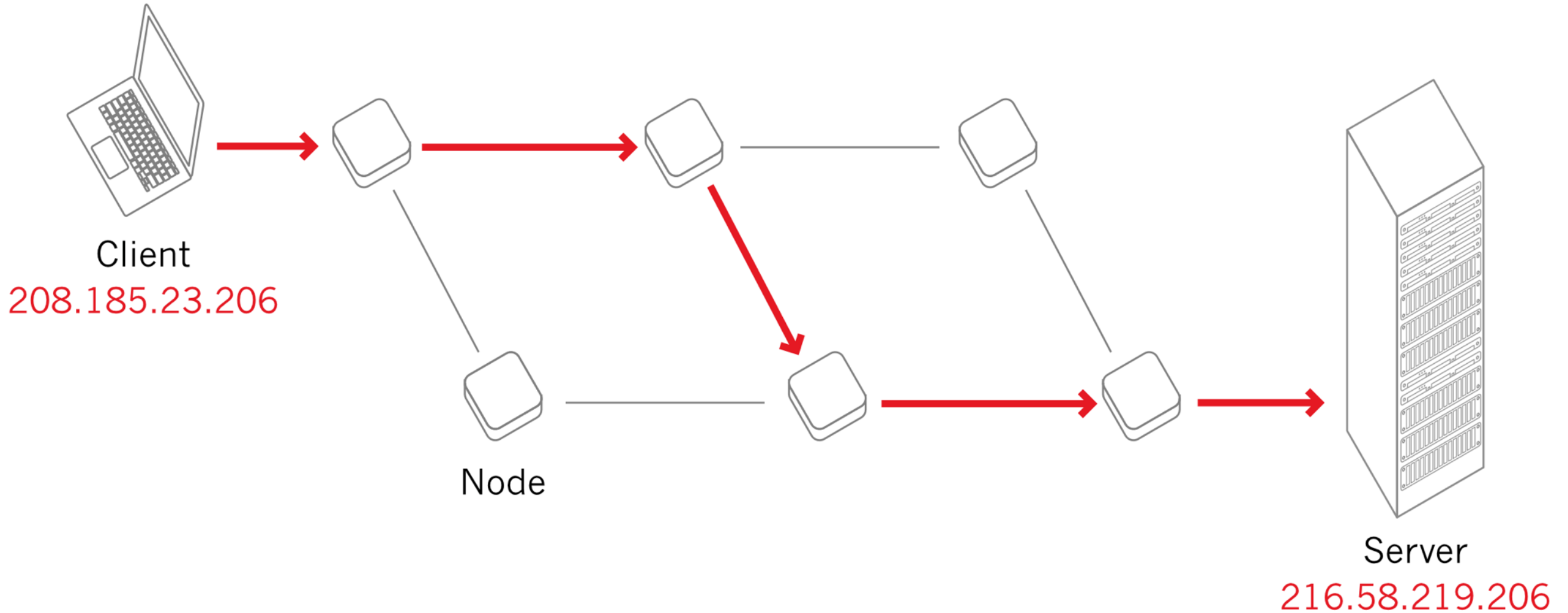
NETWORKING

INTERNET LAYER



NETWORKING

INTERNET LAYER



NETWORKING

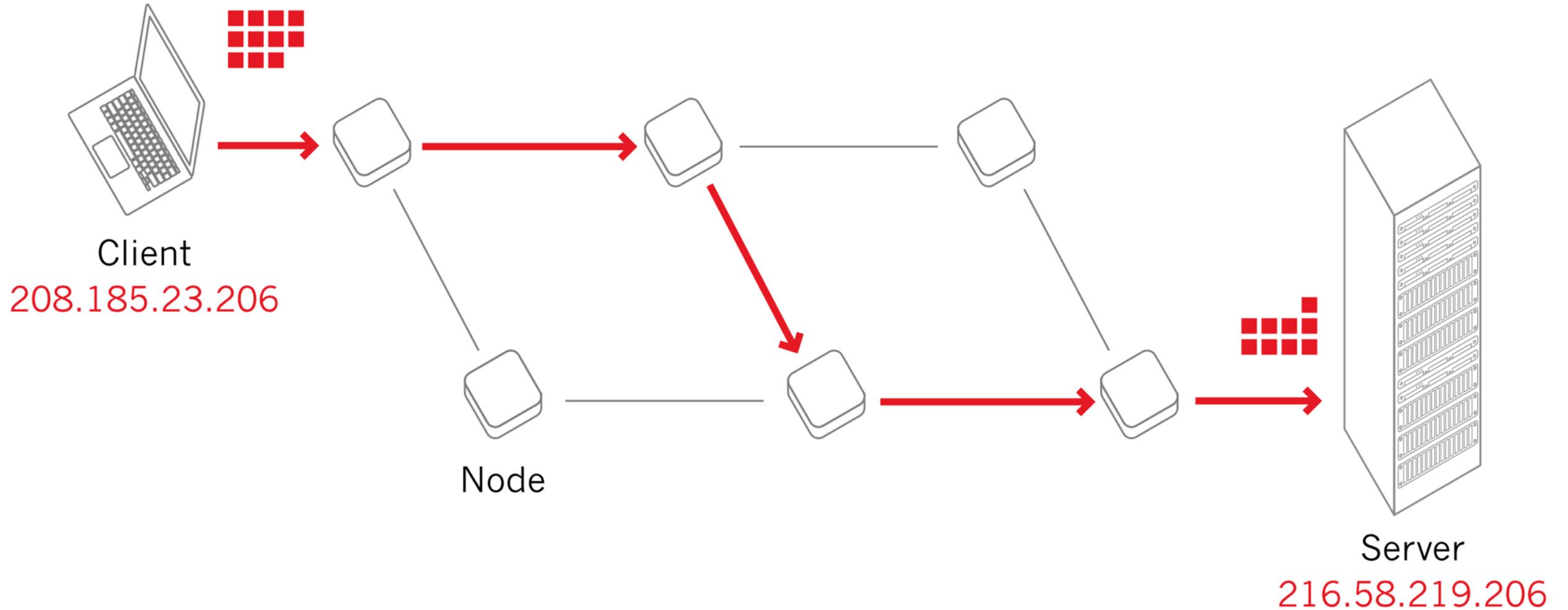
TRANSPORT LAYER

Provides host-to-host communication protocols.

- TCP – Transmission Control Protocol
- UDP — User Datagram Protocol

NETWORKING

TRANSPORT LAYER - PACKETS



NETWORKING

APPLICATION LAYER

An abstraction layer that defines a set of standard, high-level protocols used by applications.

- › HTTP – HyperText Transfer Protocol
- › FTP – File Transfer Protocol
- › SMTP – Simple Mail Transfer Protocol
- › IMAP – Internet Message Access Protocol
- › DNS – Domain Name System
- › DHCP – Dynamic Host Configuration Protocol

NETWORKING

COMMON PROTOCOLS

NETWORKING

IP PROTOCOL (INTERNET LAYER)

The Internet Protocol determines how data moves from computer to another over a network. It uses IP addresses to identify those computers, and how to location those computers given their IP addresses.

Uses IP addresses to identify computers on a network.

IPv4 – e.g. 192.168.0.10

IPv6 – e.g. FE80:0000:0000:0000:0202:B3FF:FE1E:8329

Responsible for sending datagrams across network boundaries.

The Internet Layer then is responsible for "routing" chunks of data through network and between networks. *Allows data to be lost.* Speed is key.

NETWORKING

IP PROTOCOL (INTERNET LAYER)

Find your IP address!

- › Use <http://whatsmyipaddress.com>.
- › Find the local IP address in the Network Tab in the System Preferences.

NETWORKING

DNS (APPLICATION LAYER)

DNS services are listing services that link IP addresses to domain names that we're accustomed to (e.g. google.com).

When you make a request, the domain name must be resolved to an IP address for the IP protocol to do its magic.

Use nslookup to determine the IP address of google.com or reddit.com.

NETWORKING

TCP PROTOCOL (TRANSPORT LAYER)

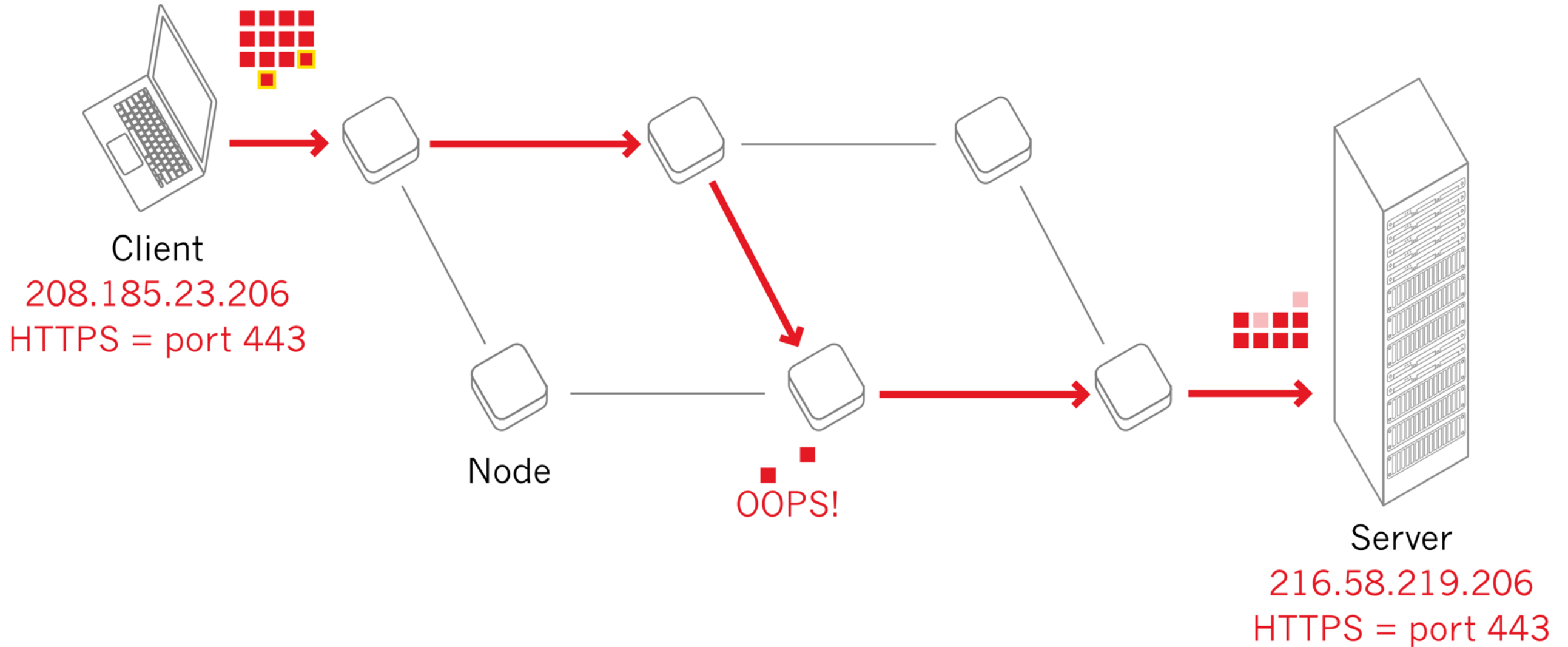
Transmission Control Protocol enables various distinct services to exist on a host via port numbers. Built on the IP protocol.

- › 65,536 ports are available on a typical system.
- › Some are standard ports reserved for various common services:
- › HTTP = port 80
- › HTTPS = port 443
- › Email = port 25
- › FTP = port 21 and 22

Ensures reliability (whereas IP doesn't) by receiving acknowledgements, resending packets, removing duplicates, and adding an order.

NETWORKING

TCP PROTOCOL (TRANSPORT LAYER) - LOST PACKETS!



NETWORKING

HTTP PROTOCOL (APPLICATION LAYER)

HyperText Transfer Protocol is the backbone of the World Wide Web.

- Typically utilizes the TCP transport to communicate packets.
- URIs used to identify resource names

NETWORKING

URI SYNTAX

From Wikipedia (https://en.wikipedia.org/wiki/Uniform_Resource_Identifier):

scheme : `[//[user:password@]host[:port]][/]path[?query][#fragment]`

- › `https://en.wikipedia.org/wiki/Uniform_Resource_Identifier`
- › `https://www.google.com/#q=ios`
- › `http://localhost:3000`
- › `mailto:person@example.com`

Common schemes include http, https, mailto, ftp, and file.

NETWORKING

HTTP REQUEST METHODS

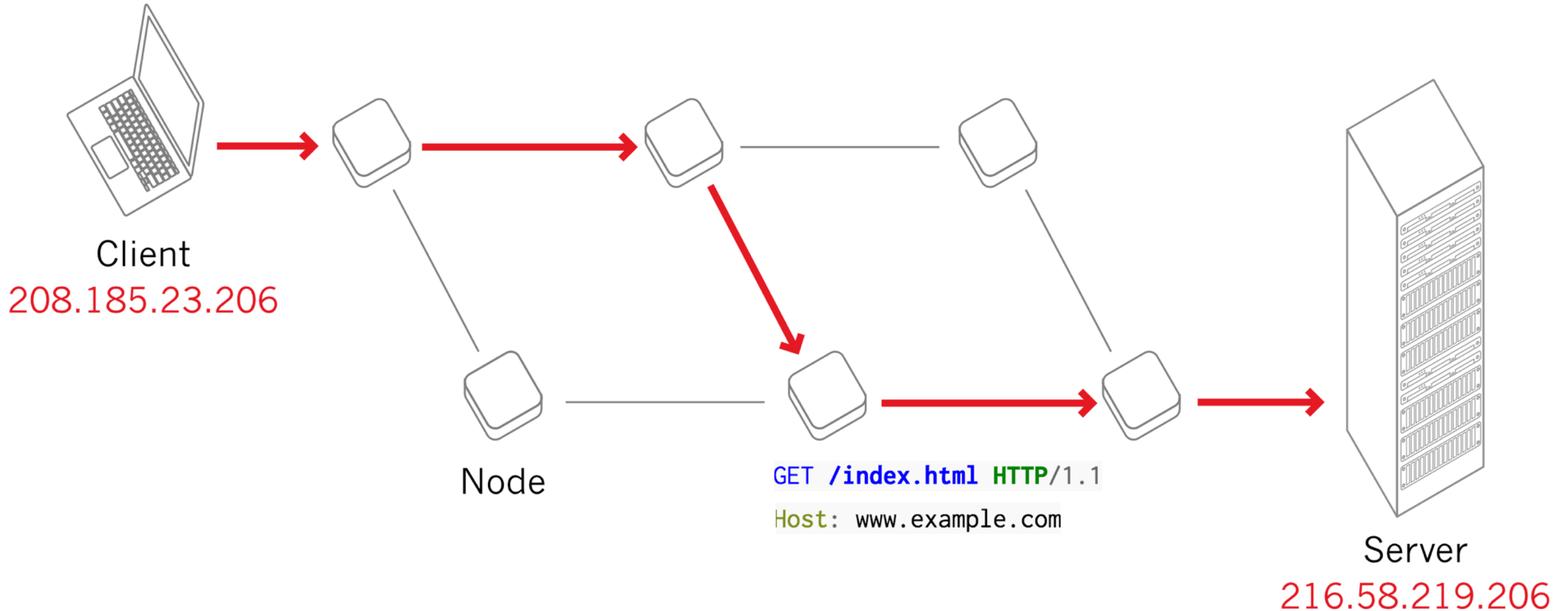
Represent an action to be performed on the server as part of the **request**.

- › GET – Requests to retrieve information from the server.
- › POST – Requests that new information be contained as a subordinate to the entity at the specified URI.
- › PUT – Requests that information be placed at the specified URI.
- › DELETE – Requests that the specified entity be removed at the given URI.
- › Others: HEAD, TRACE, OPTIONS, CONNECT, PATCH.

Afterwards, the server sends a "response" to the requester (the client), that includes either the information requested or an acknowledgment of some sort.

NETWORKING

HTTP REQUEST METHODS



NETWORKING

HTTP RESPONSES

The response typically includes a status code that communicates the nature of the response.

- 1XX - Informational
- 2XX - Success
- 3XX - Redirect
- 4XX - Client error
- 5XX - Server error

The response also comes with a messages, such as "Not Found" or "OK".

NETWORKING

HTTP RESPONSES

HTTP/1.1 200 OK

Date: Mon, 23 May 2005 22:38:34 GMT

Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)

Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT

ETag: "3f80f-1b6-3e1cb03b"

Content-Type: text/html; charset=UTF-8

Content-Length: 138

Accept-Ranges: bytes

Connection: close

<html>

<head>

<title>An Example Page</title>

</head>

<body>

Hello World, this is a very simple HTML document.

</body>

</html>

NETWORKING

HTTP REQUEST METHODS AND RESPONSES

Chrome developer tools!

Network tab!

NETWORKING

IOS NETWORKING

NETWORKING

WHAT COULD POSSIBLY GO WRONG?

NETWORKING

WHAT COULD POSSIBLY GO WRONG?

- › No network connection (client-side)
- › Destination offline (server-side)
- › The network can be slow
- › The request can not be authorized
- › Lost network connection mid-request/response
- › Lots more...

NETWORKING

HANDLE ERRORS

- If you make network connections in your app, it's a best practice to handle errors when possible, and communicate errors to the customer when not

NETWORKING

BEST PRACTICES

- › Users get charged for data: Use only what you must
 - › They do not share a filesystem
- › Allow users to cancel lengthy downloads
- › When something fails, either handle it or message the user
- › Cache when needed (this is a little more advanced)

NETWORKING

NSURLSESSION

- A shared context in which our requests are made:
- Configurable way to share state among connections.
 - e.g. “All my connections should use a cookie...”
- Can queue up several types of connections:
 - Download: For downloading files
 - Upload: For uploading files
 - Data: For small bits of data transfer, what we’ll use today.

NETWORKING

USING NSURLSESSION

```
func onCompletion(data:NSData?, response:NSURLResponse?, error:NSError?) {  
    let stringOfData = NSString(data: data!, encoding: NSUTF8StringEncoding)  
    print(stringOfData)  
}  
  
if let url = NSURL(string: "https://reddit.com/.json") {  
    let sessionMgr = NSURLSession.sharedSession()  
    sessionMgr.dataTaskWithURL(url, completionHandler: onCompletion)  
    task.resume()  
}
```

NETWORKING

NETWORKING CODE-ALONG