Networking and I/O

HW4 Code Review due Wednesday!

Outline

- Review: Internet Communication
- Network-Interface Card (NICs)
- Case study: remote procedure calls (RPCs)

Download Example Code!

Log into the course VM!

```
$ wget https://cs.pomona.edu/classes/cs134/demos/lec20.tar
$ tar xvf lec20.tar
$ cd lec20
```

Review: Communicating Over the Network

What does the message look like?

How do we know the destination IP address?

How is the message sent?

www.website.com



134.173.84.13

104.20.26.208

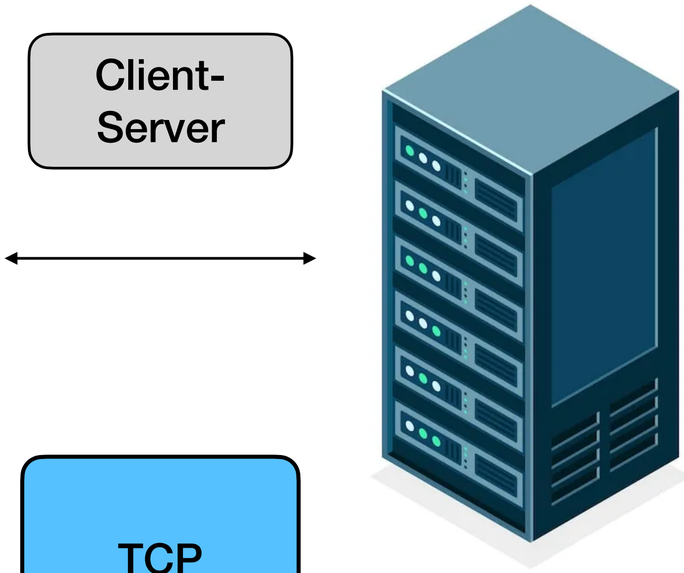


Types of Networked Communication

Unreliable network, primary goal is to guarantee consistency of connection between devices

Client-Server

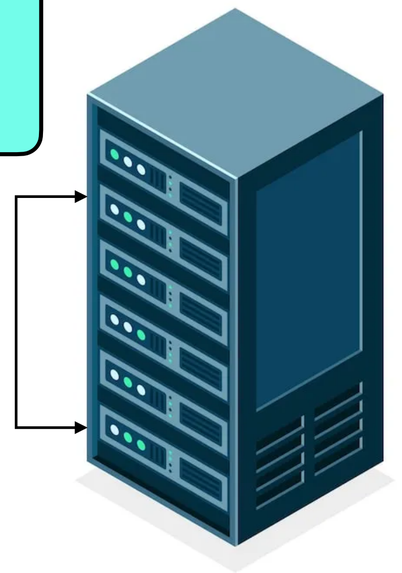




TCP

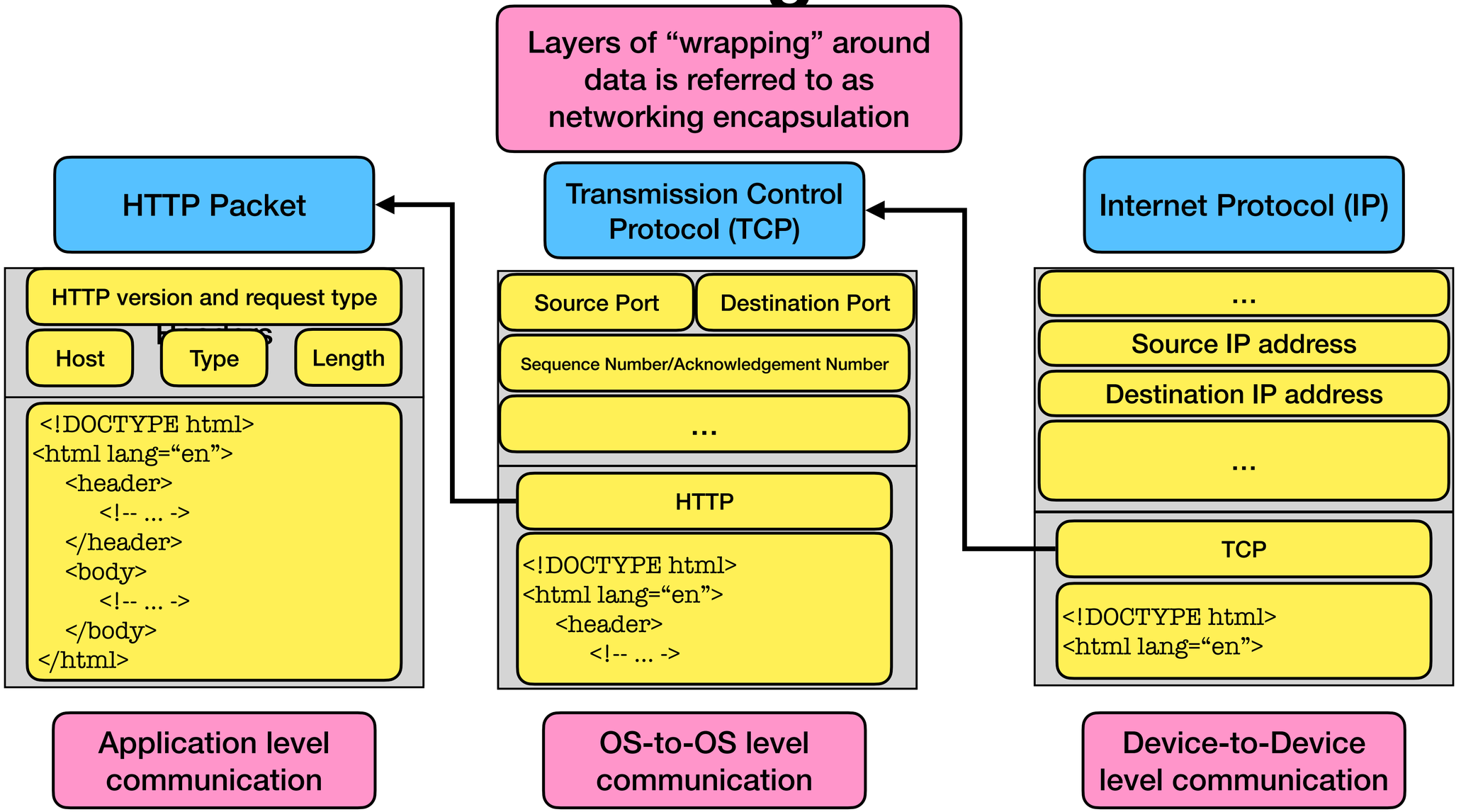
Very reliable network (physical wires), primary goal is speed of communication

Intra-Server

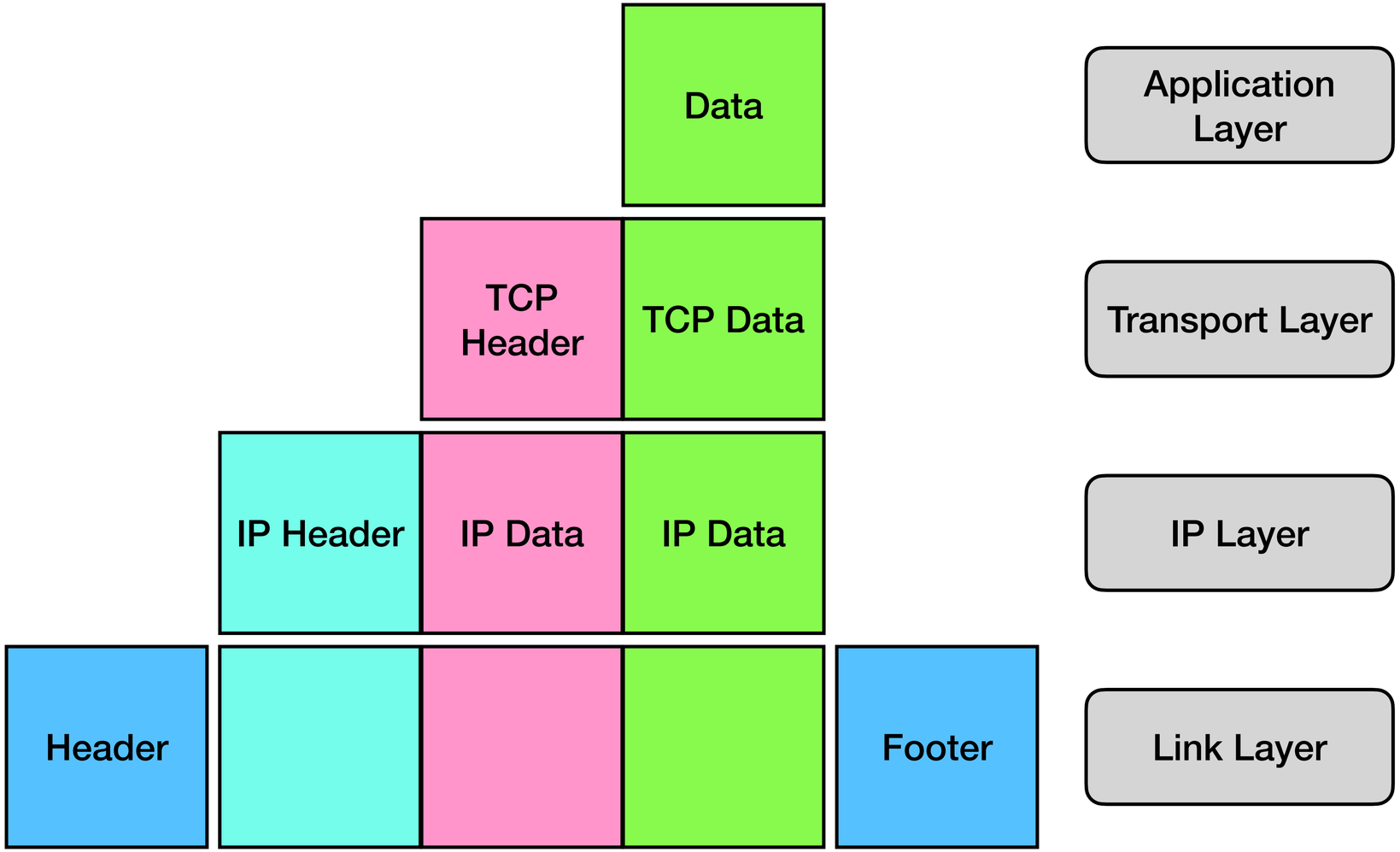


UDP

Review: Communicating Over the Network



Review: Communication Over the Network

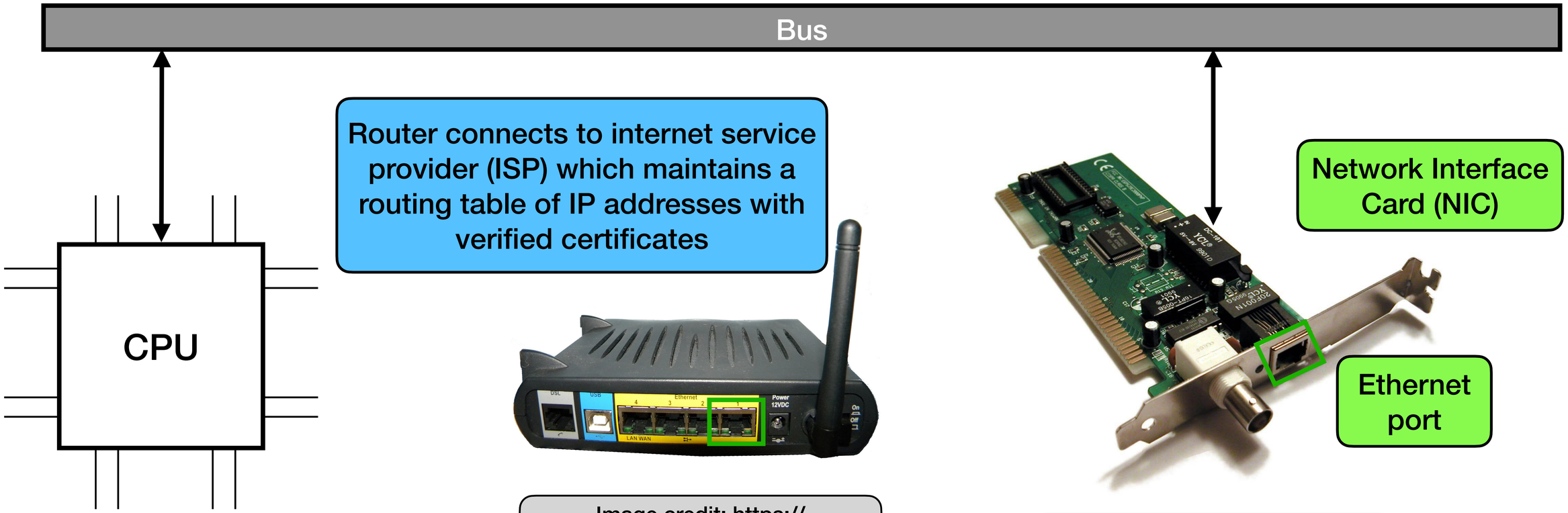


Review: Communication Over the Network

- For devices to be able to communicate with one another, they need to send traffic in such a way that other devices can help them perform the point-to-point communication that the application layer expects (e.g., client-server applications)
- To allow applications to be implemented at the level of abstraction that allows them to reason about the state, the operating system needs to reason about the reliability of the packets being sent (e.g., partial messages, lost messages, etc…) and managing connections while the hardware is tasked with finding the appropriate end-points
- The minimal communication units are generally described as “packets” which are fragments of the initial request from the application to be sent to the connected end point ➡️ the operating system and the hardware need to process the application request and transform it into a network appropriate request

How is the message sent?

We need a driver to communicate with the NIC!



Chat with your neighbor(s)!

Thinking about the communication between OS and devices via drivers, think about the implementation of a NIC driver. What kinds of communication will the OS need to send to the NIC and what kinds of events will trigger responses?

The OS will send HTTP requests to the NIC via the shared driver state

The NIC will raise an interrupt for the OS whenever there network traffic in response

Case Study: RPCs as Client-Server Optimization

Thread on Client

Function executed by the server

Value returned to client



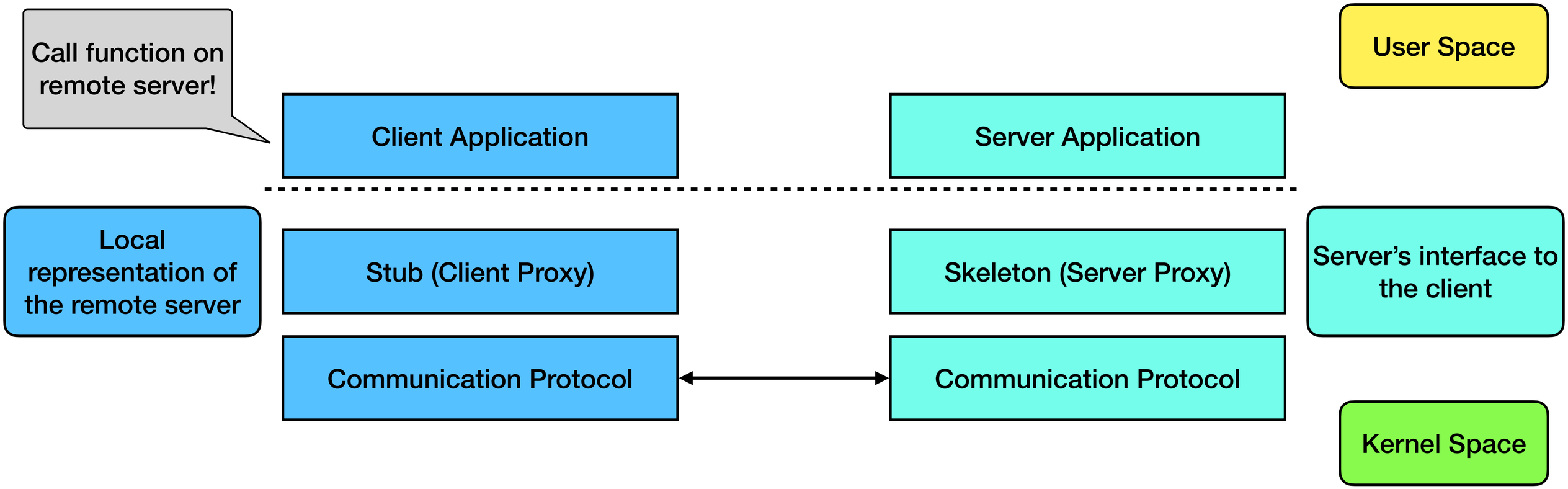


Applications: distributed systems, microservices, cloud computing, fault tolerant databases

Case Study: RPCs as Client-Server Optimization

- Instead of implementing client-server applications where the client knows nothing about the server, the connection can be setup to allow clients to have knowledge of types of functions that can be run on servers ➡️ this allows for more expressive communication
- When the initial connection is established, the server communicates the function signatures to the client of the types of functions that can be executed
- The client then sends a specific type of communication to the server to execute that function on the remote server ➡️ this should interrupt whatever the server is doing to perform the execution and the response should be sent back

Case Study: RPCs as Client-Server Optimization



Chat with your neighbor(s)!

How can we use the NIC driver to implement the RPC properties in the OS code? Think about what we would want to send on the initialization of the connection and what would need to be sent while the connection is alive!

Summary

- The operating system uses network protocols to implement cross-device systems (client and server applications)
- NICs and NIC drivers are implemented using the same PIO/DMA interfaces as other drivers in the system ➡️ the OS writes to the registers, and the device interrupts the execution of the operating system
- We can use RPCs as a mechanism to optimize the execution of client-server interactions… we will extend this with the NFS on Wednesday!