

Lab Assignments

Getting Started

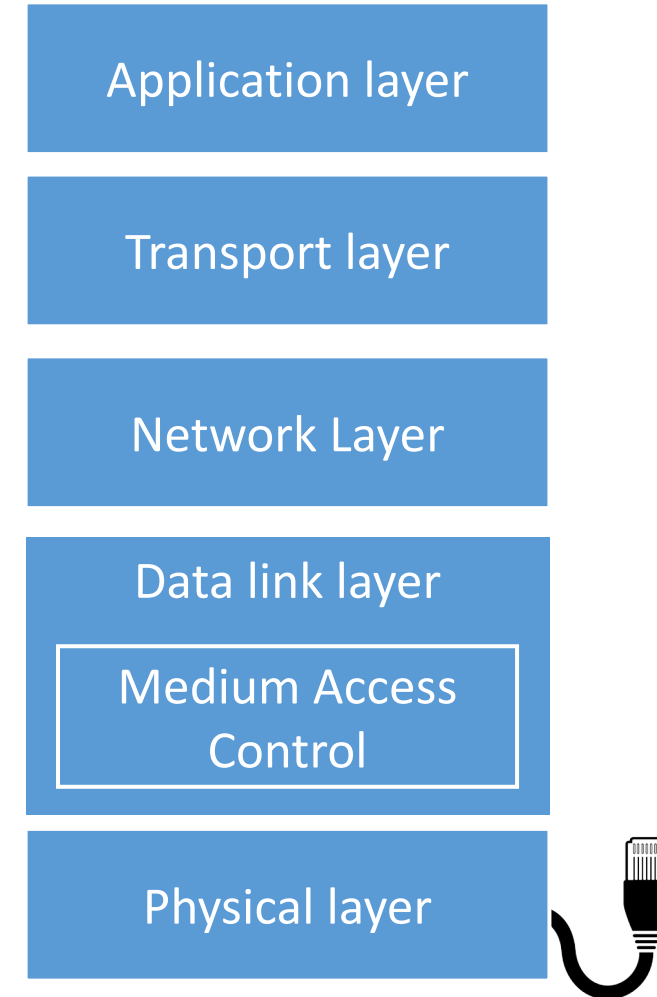
Network layer services

Sends segments from one **process** to another (over a network)

Sends packets from one **machine** to another over a network

Sends frames from one machine to another over a single link

Sends bits over a physical medium

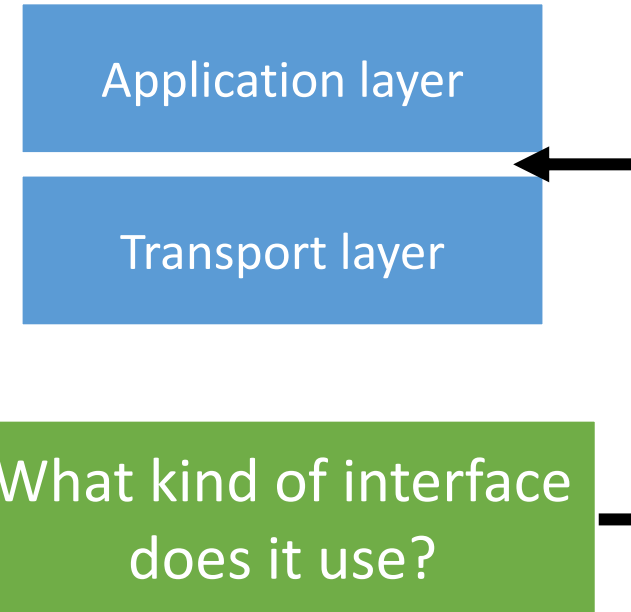


Network layer services

Sends segments from one **process** to another (over a network)

This is the service used by
your application

What kind of interface
does it use?



Socket Primitives in TCP

Socket – create a new communication *endpoint*.

Connect – connect to a remote *listening* socket.

Q: Are we missing something?

Send – send data to the other application.

Receive – receive data from the other application.

Close – close the connection.

Used to allow incoming connections

Bind – assign a *local address* to the socket.

Listen – wait for a connection.

Accept – passively accept an incoming *connection request*.

TCP Sockets in Python

```
# Import the socket library.
```

```
import socket
```

```
# Create a new socket.
```

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```
# Connect to another application.
```

```
s.connect(("hostname", port_number))
```

```
# Send bytes.
```

```
num_bytes_sent = s.send(buffer)
```

```
s.sendall(buffer)
```

```
# Receive bytes.
```

```
buffer = s.recv(2048)
```

```
# Close connection.
```

```
s.close()
```

Network layer protocol

Transport layer protocol

Max number of bytes to receive

Application layer

Transport layer

Network Layer

Data link layer

Medium
Access Control

Physical layer



TCP provides a reliable byte-stream

Q: What does this mean for your application?

1. The program waits until data is available.
2. It may return an arbitrary number of bytes.

s.recv(2048)

H	E	L	L	O	-	F	R	O	M		J	E	S
---	---	---	---	---	---	---	---	---	---	--	---	---	---

s.recv(2048)

S	E	\n	W	H	O	\n
---	---	----	---	---	---	----

s.recv(2048)

S	E	N	D		E	C	H	O	B	O	T		H
---	---	---	---	--	---	---	---	---	---	---	---	--	---

s.recv(2048)

E	L	L	O		W	O	R	L	D	\n
---	---	---	---	--	---	---	---	---	---	----

Threading Python

```
# Import threading library.  
import threading  
# A regular call to print.  
print("Hello", "World")  
# A threaded call to print.  
t = threading.Thread(target=print, args=("Hello", "World"))  
# Run target in new thread.  
t.start()  
# Wait 100ms for thread to finish.  
t.join(0.1)
```