Introduction to Operating Systems

Through tracing, analysis, and experimentation

George V. Neville-Neil

February 2017

Communication

Inter-Process Communication

Goals of IPC

- Data Sharing
- Signaling events
- Control of multiple processes

Mechanisms

- Shared files
- Semaphores and Mutexes
- Signals
- Sockets

Relationship to Networking

- Extension of mechanisms across machines
- Everything is a byte stream
- No record boundaries
- File like API

Signals

- Based on hardware interrupt model
- Not useful for data transfer
- Catch and process

Signal Handling

- Source raises a signal
- destination catches the signal
- Uncaught signals cause a program to exit

Available Signals

Name	Meaning	Name	Meaning
SIGHUP	line hangup	SIGURG	urgent condition present on socket
SIGINT	interrupt program	SIGSTOP	stop (cannot be caught or ignored)
SIGQUIT	quit program	SIGTSTP	stop signal generated from keyboard
SIGILL	illegal instruction	SIGCONT	continue after stop
SIGTRAP	trace trap	SIGCHLD	child status has changed
SIGABRT	abort program	SIGTTIN	background read attempted from control terminal
SIGEMT	emulate instruction executed	SIGTTOU	background write attempted to control terminal
SIGFPE	floating-point exception	SIGIO	I/O is possible on a descriptor
SIGKILL	kill program	SIGXCPU	cpu time limit exceeded
SIGBUS	bus error	SIGXFSZ	file size limit exceeded
SIGSEGV	segmentation violation	SIGVTALRM	virtual time alarm
SIGSYS	non-existent system call invoked	SIGPROF	profiling timer alarm
SIGPIPE	write on a pipe with no reader	SIGWINCH	Window size change
SIGALRM	eal-time timer expired	SIGINFO	status request from keyboard
SIGTERM	software termination signal	SIGUSR1	User defined signal 1
		SIGUSR2	User defined signal 2
		SIGTHR	thread interrupt
		SIGLIBRT	real-time library interrupt

Tracing Signals

Pipes

- Earliest bulk data IPC
- Key innovation of UNIX systems
- Depends on file descriptors
    - *STDIN*, *STDOUT*, *STDERR*

Pipe Demonstration

Internetworked Communication

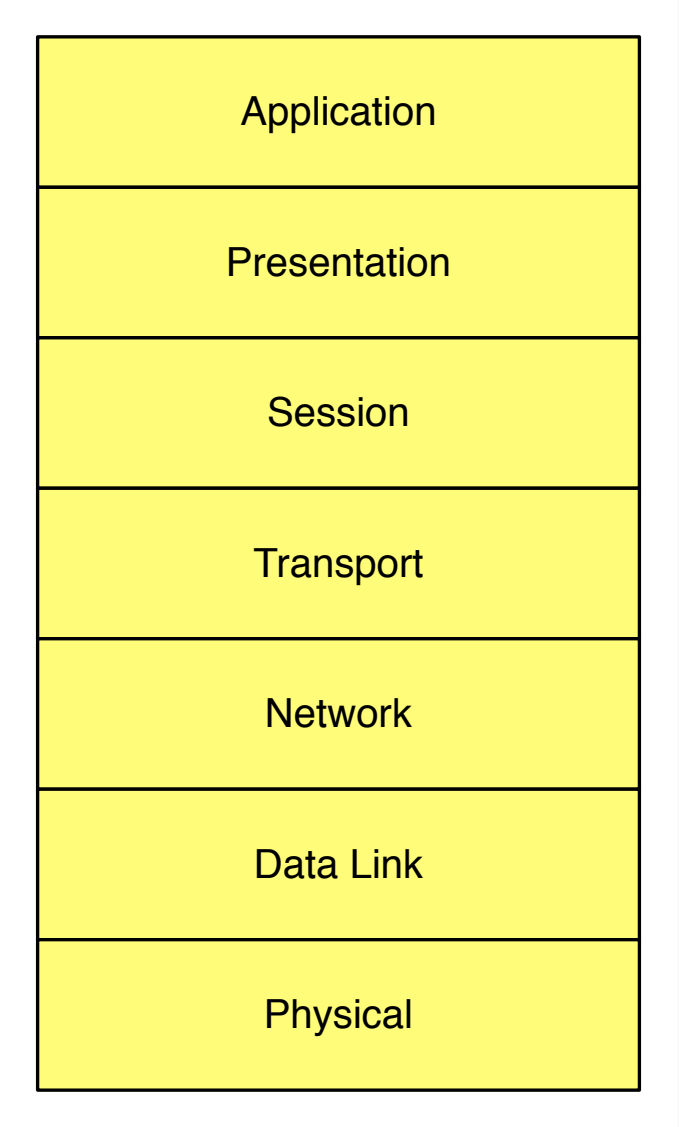
Networking and FreeBSD

- Everyone's TCP/IP Stack
- IPv4, IPv6, UDP, TCP, SCTP
- Various drivers
- Multiple firewalls

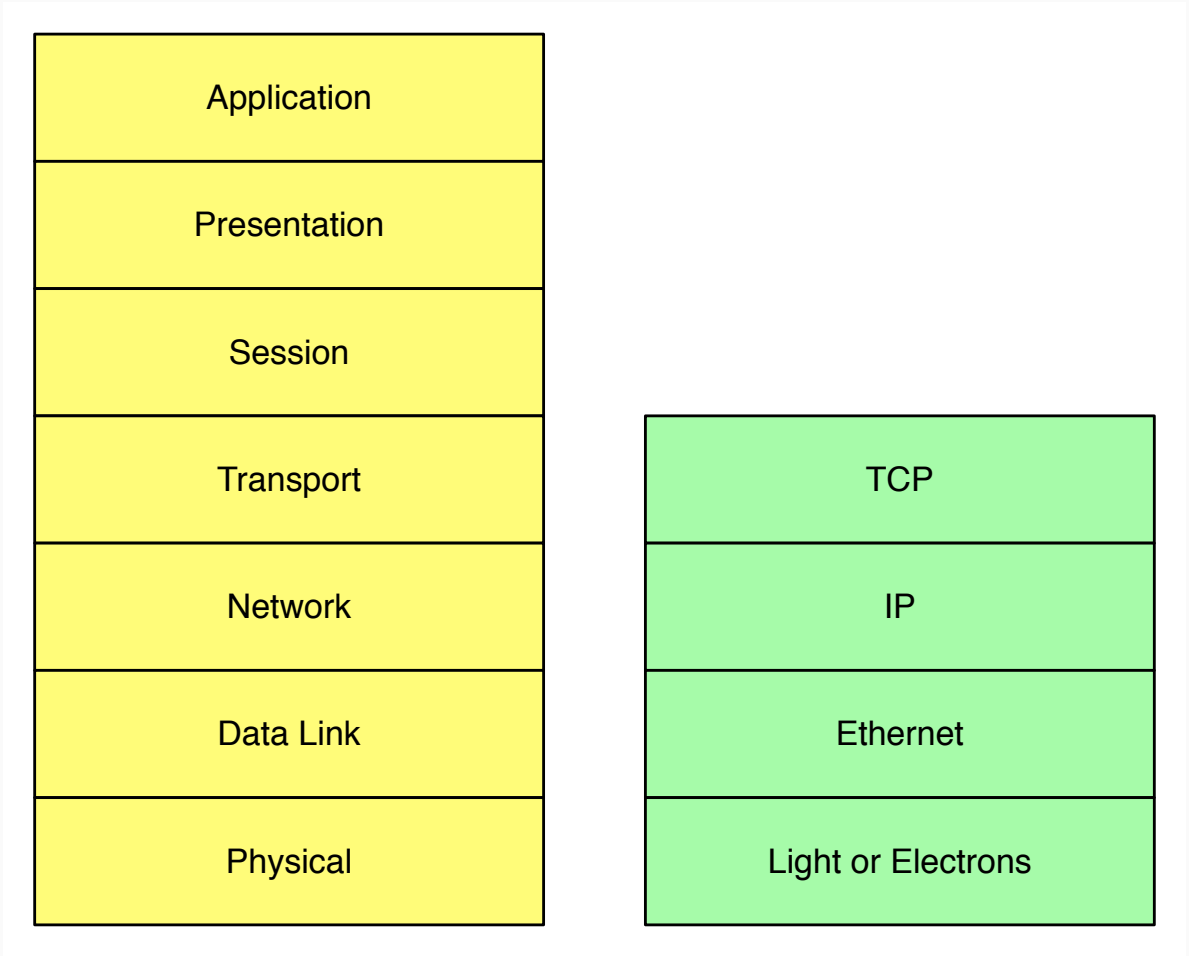
Networking: The ISO Model

- Canonical description of network protocols
- Each protocols are layered
- Seven layers in all
- Beware Van Jacobsen's warning!

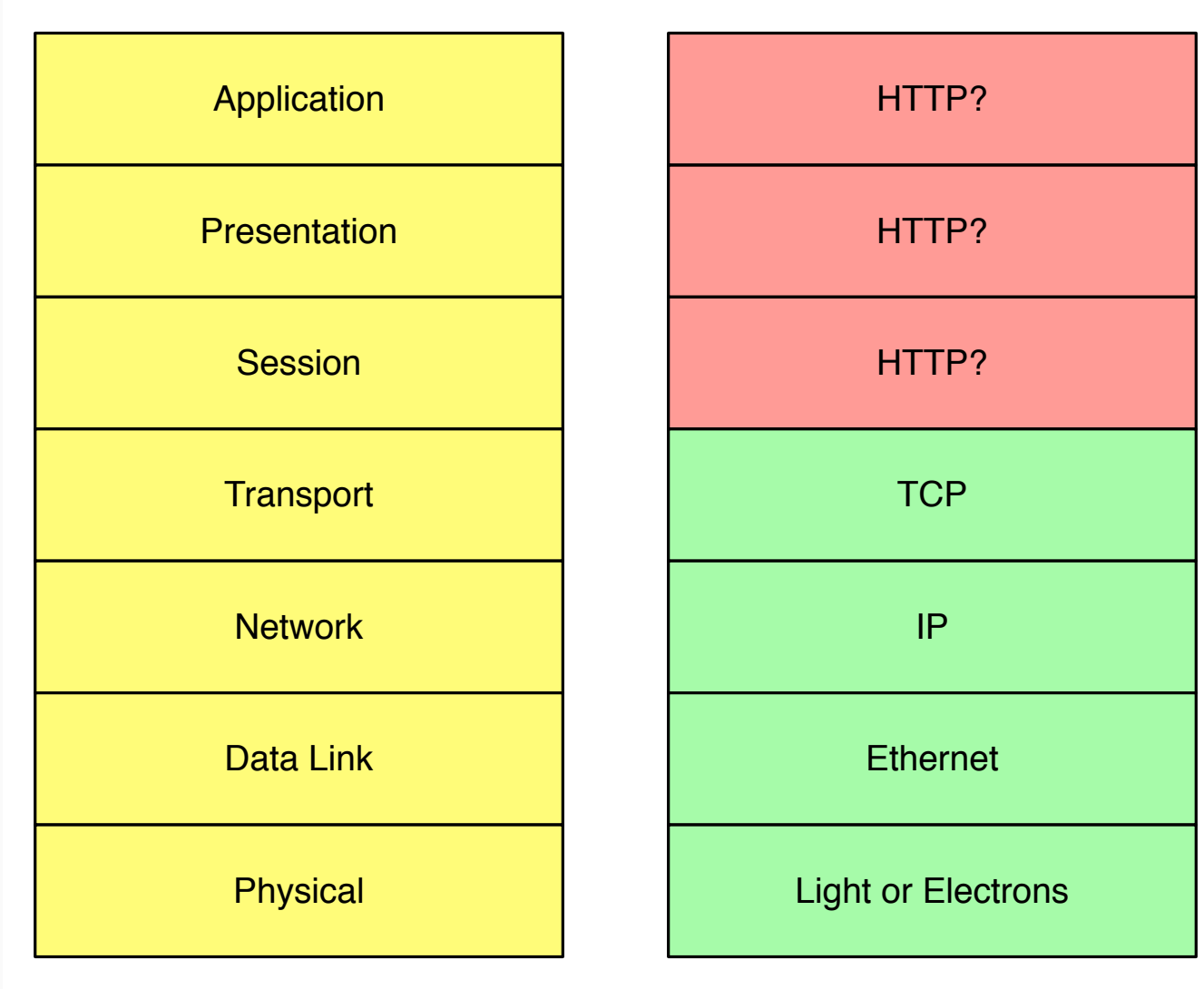
Networking and Layering



Networking and Layering



Networking and Layering



The User Program View

- User programs use sockets
- Network programs follow UNIX model
- Flexible interfaces for different protocols

Sockets

- Main programmer interface to networking
- Generic API
- Attempts to support read/write semantics

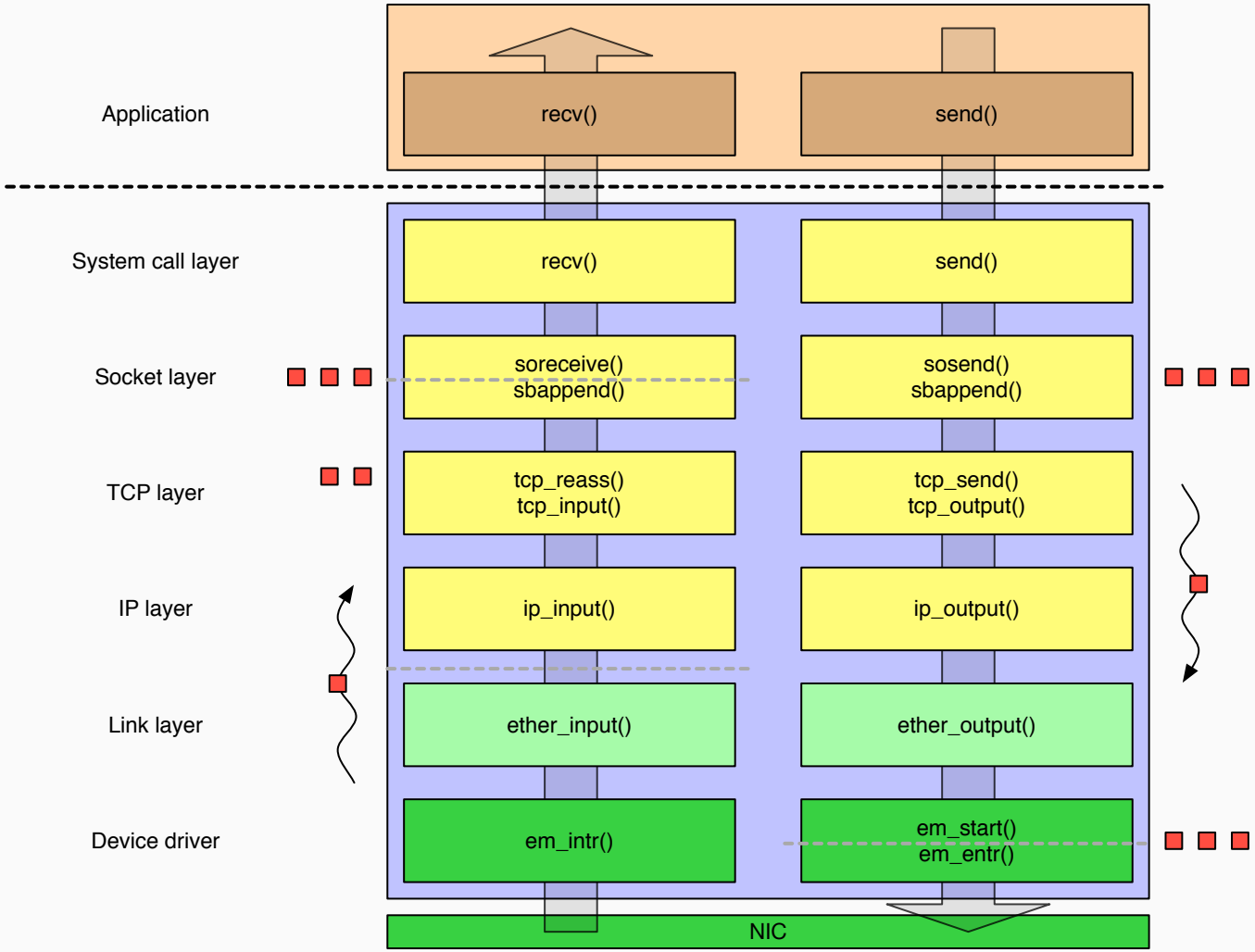
Socket System Calls

socket	Returns a file descriptor
connect	Connect to a remote program
bind	Bind a socket to a port
listen	Listen for connections
accept	Returns a new file descriptor

Transferring Data on Sockets

read	Just like a file
write	Just like a file
recv	Receive a single message
send	Send a single message
recvmsg	Receive a message with meta-data
sendmsg	Send a message with meta-data

Network Stack Overview



UDP

- Simplest transport protocol
- No states to maintain
- Data is sent immediately
- Supports multicast
- Only probes are `send` and `receive`

TCP

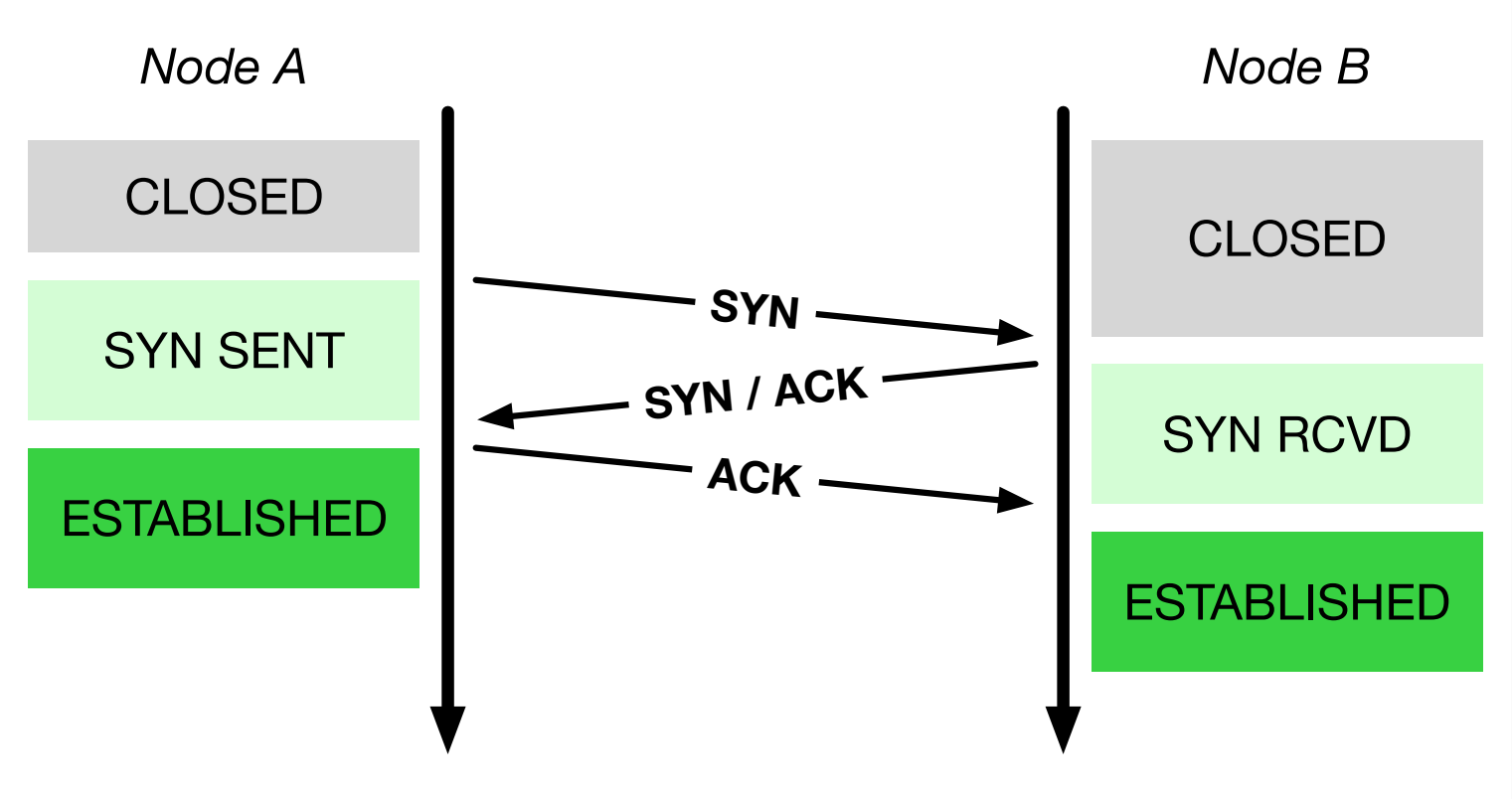
- Transmission Control Protocol
- Stream based
- In order delivery
- Maintains the illusion of a byte stream

Three Way Handshake

- Initiating a connection between two nodes

1. Start a connection with a Synchronize (SYN) packet.
2. Acknowledge the first SYN and initiate a full connection (SYN/ACK)
3. Acknowledge the second SYN.

Starting a Connection



TCP States

CLOSED	
SYN SENT	Client initiated a connection.
SYN RECEIVED	Server received initiation from client.
ESTABLISHED	Client and server can communicate.
FIN WAIT 1	
FIN WAIT 2	
TIME WAIT	
CLOSE WAIT	
LAST ACK	Awaiting client's final acknowledgment

TCP Data Flow

- Sequence Numbers
- Acknowledgements
- The sliding window
- Congestion Control

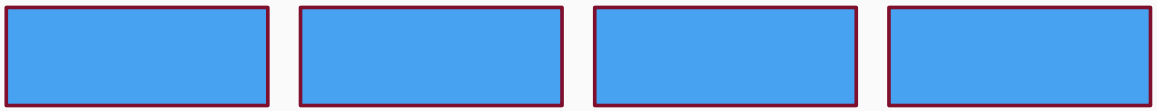
The Sliding Window

The Sliding Window

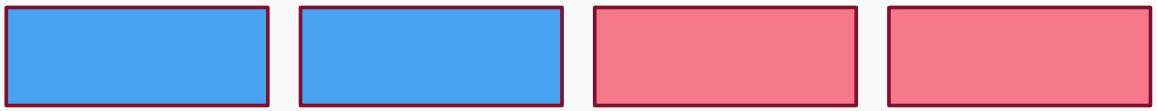
The Sliding Window

The Sliding Window

The Sliding Window

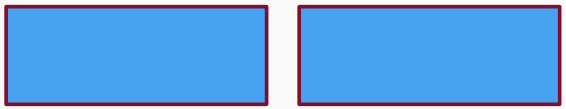


The Sliding Window



The Sliding Window

The Sliding Window

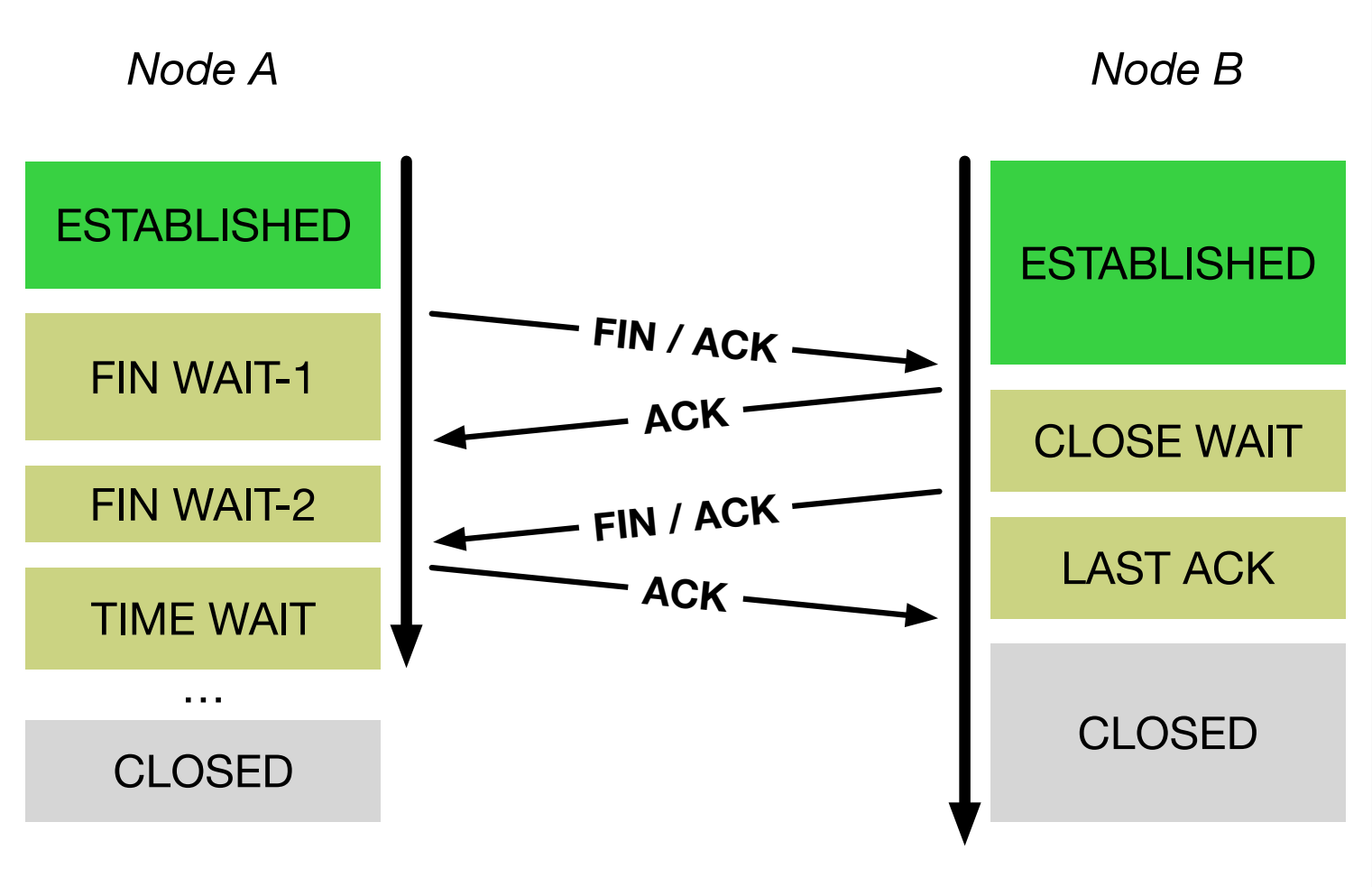


Four Way Close

- Closing a connection between two nodes
- Each node must close its side of the connection
- More complicated than opening a connection.

1. Node A sends a Finalize (FIN) packet
2. Node B acknowledges the FIN packet.
3. Node B sends a Finalize (FIN) packet
4. Node A acknowledges the FIN packet.

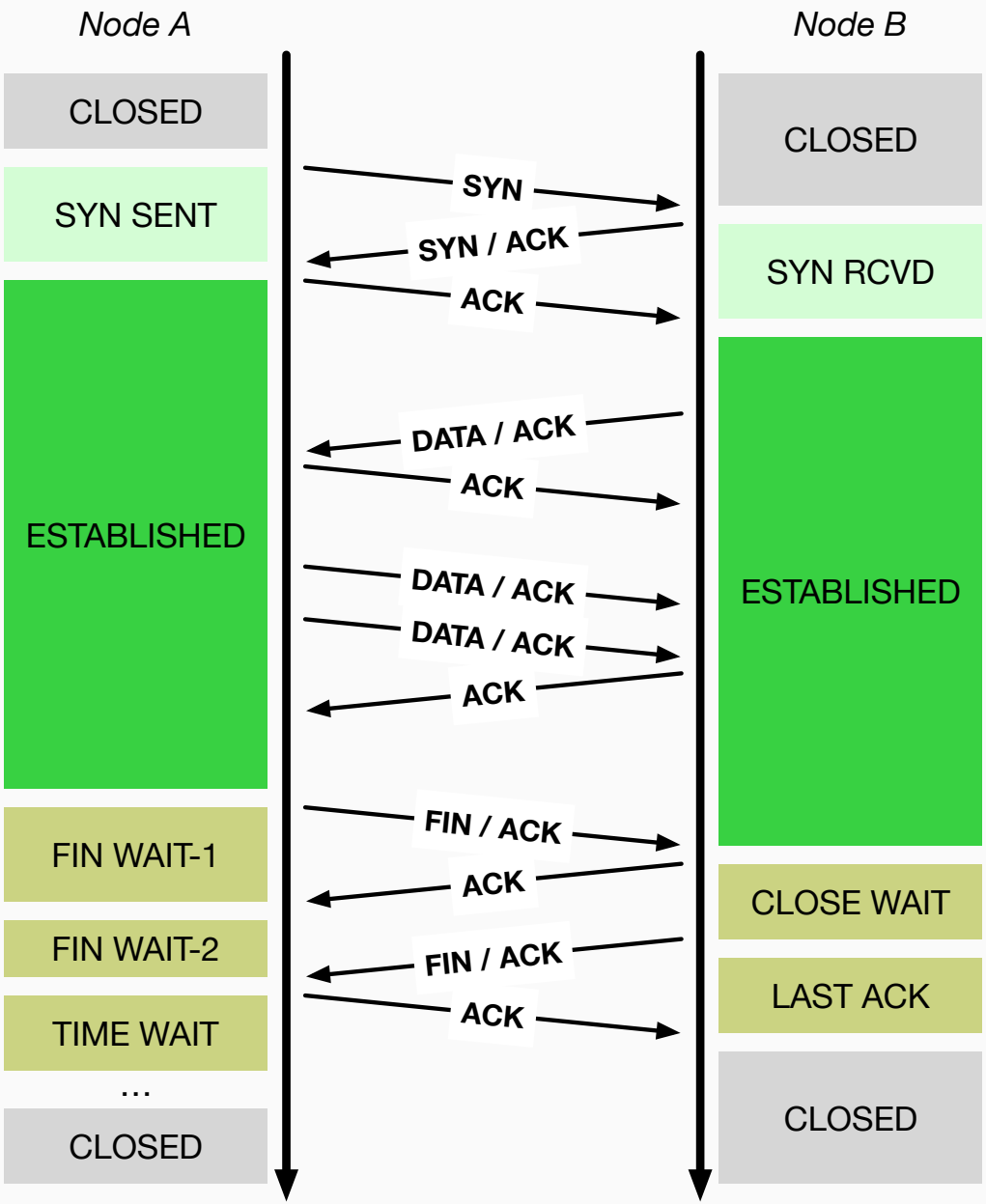
Closing a Connection



TCP States

FIN WAIT 1	
FIN WAIT 2	
TIME WAIT	
CLOSE WAIT	
LAST ACK	Awaiting client's final acknowledgment

TCP State Machine



DTrace and Networking Walkthrough

Communication Review

IPC Signals