



CSE 461 Section Week 9

Mark Guan & Bradford Chen



Administrivia

Final: Scheduled for ***June 11th, 8:30-10:20AM***

- Start thinking about review
- Next week's section will be Final review

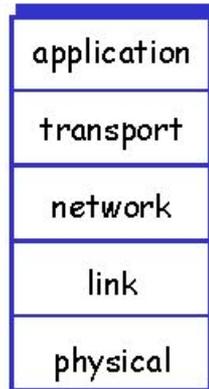
Project 3 Proxy: Due ***June 3rd***

Homework 5: Due ***June 1st***

Fair Game for the Final

Internet protocol stack

- ❑ **application:** supporting network applications
 - ftp, smtp, http
- ❑ **transport:** host-host data transfer
 - tcp, udp
- ❑ **network:** routing of datagrams from source to destination
 - ip, routing protocols
- ❑ **link:** data transfer between neighboring network elements
 - ppp, ethernet
- ❑ **physical:** bits "on the wire"



- 5 layers we've covered, but a heavy focus on Application, Link and Physical Layers
- Security
- Cloud computing networks (tentative)

MACA

- Protocol rules:
 1. A sender node transmits a RTS (Request-To-Send, with frame length)
 2. The receiver replies with a CTS (Clear-To-Send, with frame length)
 3. Sender transmits the frame while nodes hearing the CTS stay silent
 - Collisions on the RTS/CTS are still possible, but less likely





Security - Symmetric vs. Asymmetric Encryption

Symmetric (Shared Key)

Asymmetric (Public Key)

Key
Distribution

Hard - need to share secret key to both parties securely

Easier - each user publishes their own public key

Runtime
Performance

Fast - only need to reverse the encryption with the same key

Slow - need to run RSA decryption, very expensive



Quiz Review



Project 3 - HTTP Proxy

Tips and Common Errors

Forwarding Data with Byte Buffers (Java)

```
byte[] buf = new byte[1024];  
while (true) {  
    input.read(buf);  
    output.write(buf);  
}
```

Where is the issue?

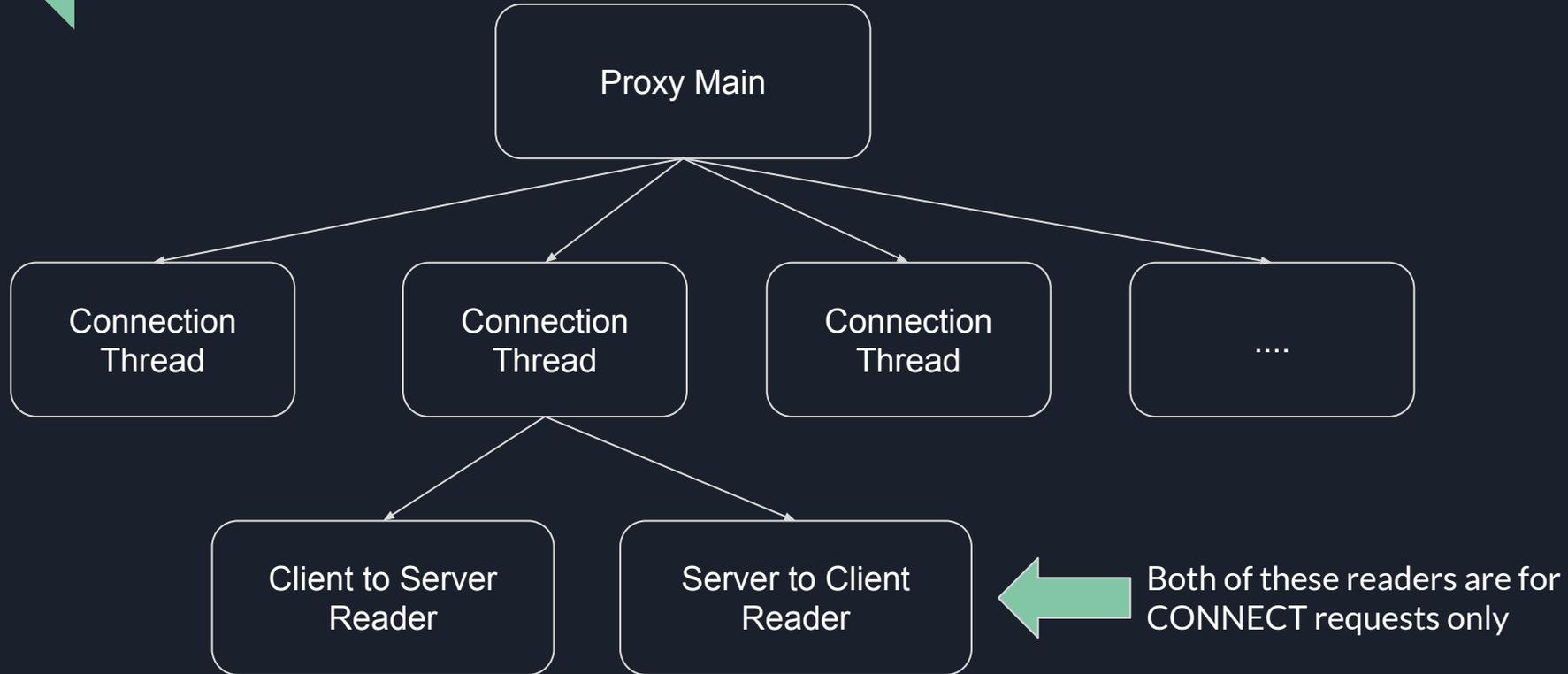
Writes entire buffer to output, even if read(buf) only read 1 byte!



```
byte[] buf = new byte[1024];  
while (true) {  
    int len = input.read(buf);  
    output.write(buf, 0, len);  
}
```

Now, only writes the bytes that were read into buf. Nice!

General Proxy Multithreading Structure





Additional Things to Lookout For

- Make sure you are **not** modifying the headers beyond what the spec requires
 - Even reordering may result in a rejected packet
 - Double carriage return (“\r\n\r\n”) after sending HTTP 200 OK on Connect
 - Parsing headers but not sending headers (or sending headers too many times)
 - Closing sockets prematurely
-
- **Testing:** try loading popular HTTPS websites concurrently and ensure the proxy can do so at a reasonable speed.



Project 3 - HTTP Proxy Port Demo