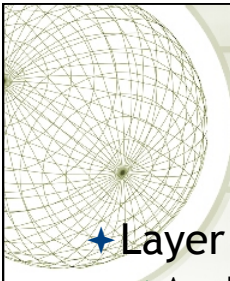


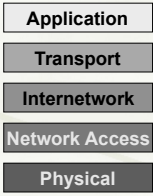
# Simple View of Protocols

Info 341 Networking and Distributed Applications

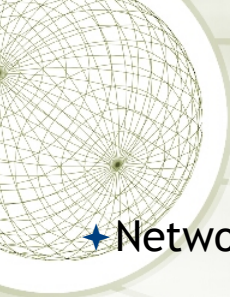


# Context

- ★ Layer 5
  - ◆ Applications
  - ◆ HTTP, SMTP, POP, ...
- ★ Layer 4
  - ◆ end-to-end communication
  - ◆ UDP, TCP



Internet Reference Model



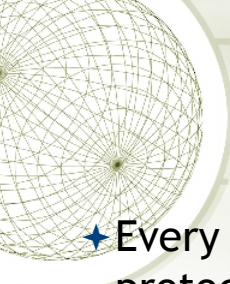
## *Protocols*

- ★ Network applications rely on protocols
- ★ What is a protocol?



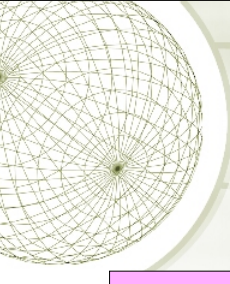
## *Protocols*

- ★ An agreed upon mechanism for communicating
- ★ A defined standard for communicating



## *Protocols*

- ★ Every Client/Server must use the protocol the same way
- ★ Peer-to-peer both client and server must use the protocol the same way
- ★ Consider HTTP



## *Protocol Schema*


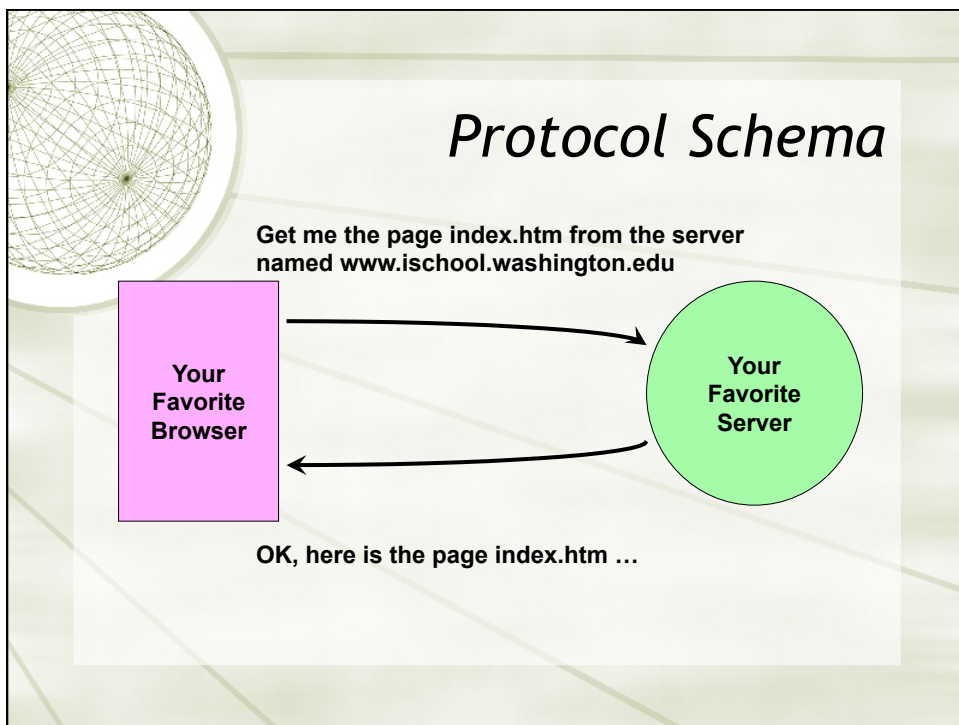
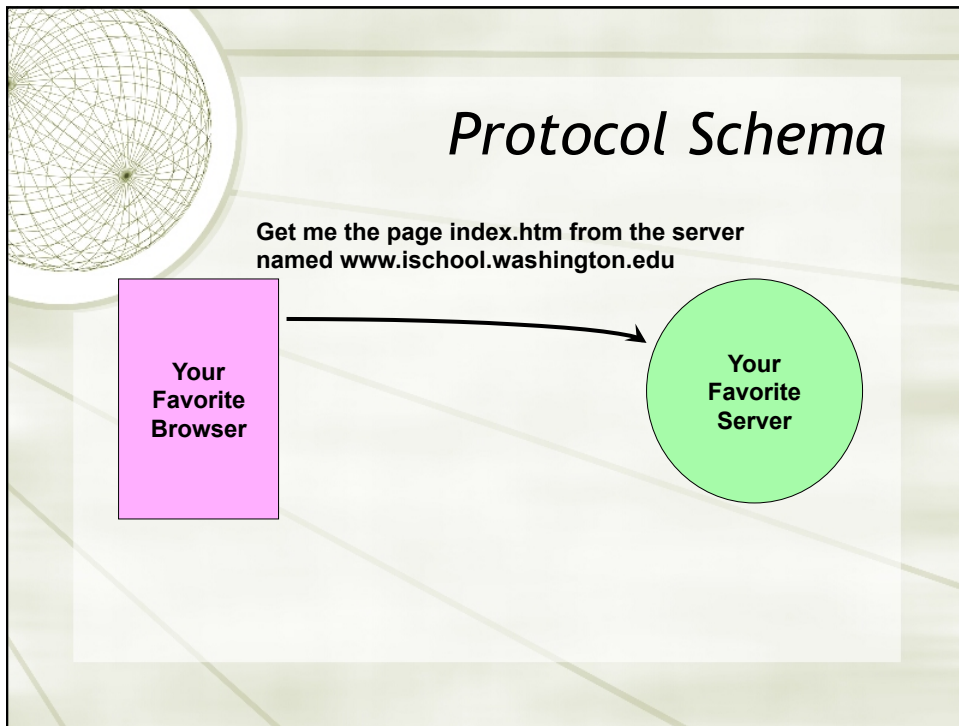
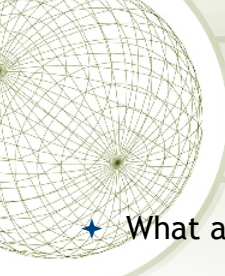


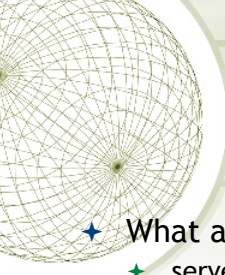
Diagram illustrating a protocol schema. On the left is a pink rectangle labeled "Your Favorite Browser". On the right is a green circle labeled "Your Favorite Server".





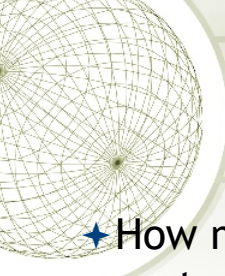
## Protocol Problems

- ★ What are the possible problems?



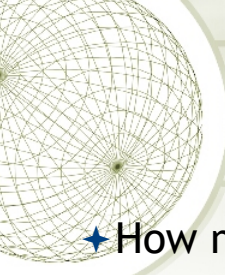
## Protocol Problems

- ★ What are the possible problems?
  - ★ server crash or internal error
  - ★ page not there or unknown
  - ★ page access restricted
  - ★ page not really 'mine' (moved, proxy)
  - ★ content (page) can expire
- ★ The protocol must specify how to handle these situations and many more



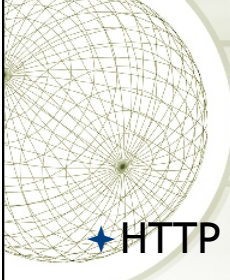
## *Protocol Specification*

- ★ How might you define/specify a protocol?




## *Protocol Specification*

- ★ How might you define/specify a protocol?
  - ★ English text
  - ★ State Transition Diagram
  - ★ Algebraic (math)



## Consider HTTP

- ★ HTTP Protocol
  - ◆ HTTP Request format
  - ◆ HTTP Response format
    - ◆ Common response codes
  - ◆ HTTP Commands



## HTTP Protocol: Request

```


Request = Request-Line
          *(( general-header
            | request-header
            | entity-header ) CRLF)
          CRLF
          [ message-body ]

Request-Line = Method SP Request-URI SP HTTP-Version CRLF

Method = "OPTIONS"
        | "GET"
        | "HEAD"
        | "POST"
        | "PUT"
        | "DELETE"
        | "TRACE"
        | "CONNECT"
        | extension-method
        extension-method = token

Request-URI = "*" | absoluteURI | abs_path | authority

HTTP-Version = "HTTP/1.0" | "HTTP/1.1"
  
```

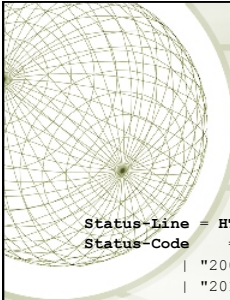


## HTTP Protocol: Response

```

Response = Status-Line
          *(( general-header
             | response-header
             | entity-header ) CRLF)
          CRLF
          [ message-body ]

Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase
             CRLF
  
```

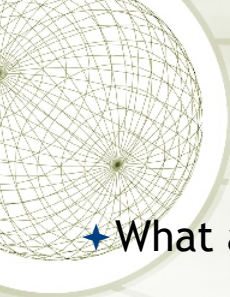


## HTTP Protocol: Status Codes

```

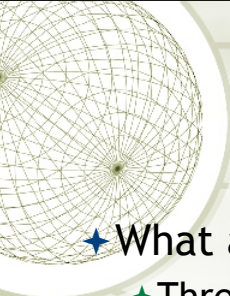
Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase CRLF
Status-Code =
| "200" ; OK
| "201" ; Created
| "202" ; Accepted
| "203" ; Non-Authoritative Information
| "204" ; No Content
| "301" ; Moved Permanently
| "307" ; Temporary Redirect
| "400" ; Bad Request
| "401" ; Unauthorized
| "402" ; Payment Required
| "403" ; Forbidden
| "404" ; Not Found
| "500" ; Internal Server Error
| "501" ; Not Implemented
| "502" ; Bad Gateway
| extension-code
extension-code = 3DIGIT
  
```





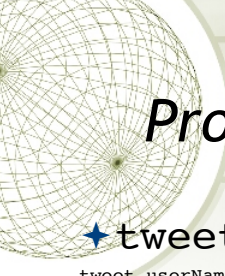
## *Project Part 3 Protocol*

- ★ What are the protocol commands?




## *Project Part 3 Protocol*

- ★ What are the protocol commands?
  - ★ Three possible commands
    - ★ tweet
    - ★ follow
    - ★ profile



## Project Protocol Commands

- ★ **tweet**  
tweet userName#userID#StatusMessageString\n
  - ★ Asynchronous, no reply
- ★ **follow**  
follow [remove] userName#userID\n
  - ★ Asynchronous, no reply
- ★ **profile**  
profile request userID | <profile data>\n
  - ★ Synchronous, reply required



## Service Type

- ★ Service Type for Part 3
  - ★ twit
  - ★ “\_twit.\_tcp”