Chapter 1: intro to computer networks: Delay(end-end)=n(dproc+(dtrans=L/R)+dprop),Instantaneous throughput= rate at which receiving router receives files, Application layer=HTTP,FTP some times referred to as message, Transport Layer=TCP,UDP packet known as segment, Network Layer = moves network packets known as datagrams gets transport layer segment and destination address, Link layer = gets datagram from network layer and delivers it to next router and then passes it back at the next router location packets known as frames, Physical Layer= moves individual bits within a frame from one node to the next, OSI model= Open System Interconnection 7 layers of protocols instead of 5 added 2 after application presentation and session, presentation layer = used for data compression and encryption and data description used to interoperate meaning, session layer = used for delaminating and synchronization of data exchange builds checkpoints and recovery schemes, botnet = a string of malware used to collect and send info back to the infector, Malware= can self replicate itself by sending from an infected host to there friends, viruses = subset of malware require some sort of user input to infect the device, Worms = subset of malware that do not require user input exploites un secure applications running on open network. Dos = uses a large bandwidth connection and sends useless packets to the server clogging its input, DDOS= same as DOS attack but uses multiple sources to spam junk, Ip spoofing = injecting things into a network using a bogus source address, Network core = does not run user applications, central part of communications such as routers orw=switches/network structure, network edge = applications/hosts and clients/ end systems/access networks/links. TCP= reliable communication, ip= used in both udp and tcp, HTTP=, Internet standard = RFS: requrest for comments, IEFT: internet engineering task force. Protocols: Define format, order of msgs sent and received actions taken on msg transmission and receipt. Chapter 2 Application layer: Client Server connection = servers always on, perminant ip data centers for scaling, clients = may be intermittently connected, may have dynamic ip addresses. , P2P = intermediate connection, self-scalability new peers bring new service capacity. Complex management, Processes Communicating = process: program running within a host within same host, two processes communicate using inter-process communication, processes in different hosts communicate by exchanging messages, client process= process that intitiates communication, server process = process that waits to be contacted. sockets = process sends/receives messages to/from its socket analogous to door, relies on transport infrastructure on other side of door to deliver message to socket at receiving process. Addressing processes = to receive messages process must have identifier, many applications can run on the same host. App-Layer protocol defines = types of messagers exchanged (request response) , message syntax: what fields in messages and how fields are delineated, message semantics: meaning of info rules for when and how processes send and respond. Open protocols = defined in RFC’s, allows for interoperability (HTTP,SMTP). Transport service needs = data integraty, throughput ,timing ,security. Internet Transport Protocols = TCP service: reliable, flow control, congestion control, no timing no minimum throughput gaurentee, no security, connection oriented (setup), UDP service: unreliable data transfer, no reliability no flow or congestion control, no timing, no throughput gaurentee, no security, no connection setup. Internet Apps: application,transport protocols (Application protocol, transport Protocol) = email (SMTP,TCP), remote terminal access (telnet,TCP),web(HTTP,TCP), file transfer (FTP,TCP), streaming multimedia ((HTTP,RTP),TCP or UDP), Internet telephony ((SIP,RTP,Proprietary),TCP or UDP), Securing TCP = SSL:provides encrypted TCP connection, Data integrity, end point authentication, SSL is app layer, HTTP = non persistent or persistent defined later Client: browser that requests, receives and displays web objects , server: web server sends objects in response to requests, HTTP Stateless (does not remember previous requests). Non persistant HTTP = takes 2RTT + file transmission per request also overhead for each tcp connection to fetch referenced objects. Browser often opens parallel tcp connections to fetch referenced objects. Persistent HTTP = server leaves connection open after sending response, client sends request as soon as it encounters a referenced object as little as one RTT for all referenced objects. Uploading from input = Post method = web page often includes form intput, input is uploaded to server in entity body. URL method = uses Get method, input is uploaded in URL field of request line. HTTP method types = 1.0: get \n post \n head asks server to leave requested object out of response, 1.1: get,post,head \n put uploads file in entity body to path \n Delete deletes file specified in path. 4 components of cookies = 1. Cookie header line of http response message 2. Cookie header line in next http request message 3. Cookie file kept on users host 4. Backend database at web site. Web caches (proxy server) = client requests without involving origin server requests objects from cache of other server or local cache if not available then cache requests from origin, cache acts as both client and server cache server checks date modified before sending. File transfer protocol (FTP)= client: initiates transfer to or from host server: remote host FTP: RFC 959 FTP server: 2 ports :port 21 port 20 used for data transfer, transfer protocol: 1.ftp client contacts ftp server, 2.client authorized over control connection 3. Client browses remote directory ad sends command, server receives transfer command opens 2nd connection (for file) to client 5. After transfer done server closes data connection 6. Opens another TCP connection to transfer another file, ftp server maintains state: current directory. FTP commands = (all sent as ascii test over control channel) USER username PASS password LIST return list of file in directory RETR filename retrieves file STOR stores file into remote host. Electronic mail = user agnets, mail servers (contains SMTP protocol between mail servers to send email messages), SMTP: simple mail transfer protocol has 3 phases of transfer (handshaking, transfer of messages, closure) uses ppersisten connections, requires message to be in 7bit ascii uses CRLF.CRLF to determine end of message.POP3 protocol = 2 phases (authorization phase and transaction phase) client commands: user , pass server responses: +k –ERR pop 3 allows for keeping copies of messages on diefferent clients and is stateless across sessions. IMAP= keeps all messages on server, allows user to organize messages in folders, keeps user state across sessions, keeps names of folders and mappings between message ids and folder name. DNS=Domain Name System distributed database implemented in hierarchy of many name servers 13 root servers worldwide, all server cache mapping to other servers in TTL disappears after certain time. Update/ notify mechanizms use IETF protocol,.TLD servers = Top-Level domain server, responsible com,org ,edu. Athoreitative DNS server = organization’s own dns server(s) providing authoritative hostname to ip mapping for the orginizations host name. Local DNS server = ISP provided uses cache to give recently used names, acts as proxy forwards query into hierarchy. DNS records (types) = A:name is hostname value is ip address, NS: name is domain value is hostname of authorative sname server for this domain, CNAME: name is alias name for some canonical name (shortened values of real name) value is canonical name, MX: value is name of mailserver associated with name. Clent server vs p2P = P2P is faster because each successive peer brings service copacity. DHT = distributed Hash Table uses key value pairs distributed amongst millions of peers peer queries DHT with key DHT returs value, peer can also insert key value pairs keys must be hashed to get original. Socket = the point where a local application process attaches to the network interface between application and the network, interface defines operations for: creating a socket, attaching a socket to the network, sending and receiving messages through the socket, closing the socket. Socket family = PF\_inet: internet family ,PF\_unix: unix pipleline, ,PF\_packet: direct network access. Chapter 3 transport layer: Transport services and protocols = logical communication, protocols run in end systems, send side: breaks app messages into segments passes to network layer, recv side: reassembles segments into messages, passes to app layer. Transport vs network layer = network: logical communication between hosts, transport: logical communication between processes : relies on, enhances, network layer services. Multiplexing at sender = handle data from multiple sockets, add transport header. Demultiplexing at receiver = use header info to deliver received segments to correct socket. Demultiplexing = hosts receive ip datagrams: each datagram has source ip / destination IP, each datagram carries on transport layer segment, each segment has source destination port number. Hosts uses ip addresses and port numbers to direct segment to appropriate socket, can use connectionless demultiplexing . UDP = User datagram protocol, barebones reliable transfer over udp Is possible if added in application layer application specific error recovery. UDP Checksum= goal: to detect errors in transmitted segment, **sender:** treat segment contents including header fileds as sequence of 16 bit integers, checksum addition is ones compliment of segment contents sender puts checksum values into udp checksum field receiver: compute checksum of received segment, check if computed checksum equals checksum field value:no = error detected yes = no error detected by still could have errors.