- **PARAGRAPH**
    - CHAPTER 2
- **PARAGRAPH**
    - Building Blocks of TCP
- **PARAGRAPH**
    - At the heart of the Internet are two protocols, IP and TCP. The IP, or Internet Protocol, is what provides the host-to-host routing and addressing, and TCP, or Transmission Control Protocol, is what provides the abstraction of a reliable network running over an unreliable channel.
    - TCP/IP is also commonly referred to as the Internet Protocol Suite and was first proposed by Vint Cerf and Bob Kahn in their 1974 paper titled "A Protocol for Packet Network Intercommunication."
- **PARAGRAPH**
    - The original proposal (RFC 675) was revised several times, and in 1981 the v4 specification of TCP/IP was published not as one, but as two separate RFCs:
- **PARAGRAPH**
    - • RFC 791—Internet Protocol
- **PARAGRAPH**
    - • RFC 793—Transmission Control Protocol
- **PARAGRAPH**
    - Since then, there have been a number of enhancements proposed and made to TCP, but the core operation has not changed significantly.

  - TCP quickly replaced previous protocols and is now the protocol of choice for many of the most popular applications: World Wide Web, email, file transfers, and many others.

- **PARAGRAPH**

  - TCP provides an effective abstraction of a reliable network running over an unreliable channel, hiding most of the complexity of network communication from our applications: retransmission of lost data, in-order delivery, congestion control and avoidance, data integrity, and more.

  - When you work with a TCP stream, you are guaranteed that all bytes sent will be identical with bytes received and that they will arrive in the same order to the client.

  - As such, TCP is optimized for accurate delivery, rather than a timely one.

  - This, as it turns out, also creates some challenges when it comes to optimizing for web performance in the browser.

- **PARAGRAPH**

  - The HTTP standard does not specify TCP as the only transport protocol.

  - If we wanted, we could deliver HTTP via a datagram socket (User Datagram Protocol or UDP), or any other transport protocol of our choice, but in practice all HTTP traffic on the

- **PARAGRAPH**

  - Internet today is delivered via TCP due to the many great features it provides out of the box.

- **PARAGRAPH**

  - Because of this, understanding some of the core mechanisms of TCP is essential knowledge for building an optimized web experience.

- Chances are you won't be working with TCP sockets directly in your application, but the design choices you make at the application layer will dictate the performance of TCP and the underlying network over which your application is delivered.

- **PARAGRAPH**
    - Three-Way Handshake

- **PARAGRAPH**
    - All TCP connections begin with a three-way handshake (Figure 2-1).
    - Before the client or the server can exchange any application data, they must agree on starting packet sequence numbers, as well as a number of other connection specific variables, from both sides.
    - The sequence numbers are picked randomly from both sides for security reasons.

- **PARAGRAPH**
    - SYN

- **PARAGRAPH**
    - Client picks a random sequence number x and sends a SYN packet, which may also include additional TCP flags and options.

- **PARAGRAPH**
    - SYN ACK

- **PARAGRAPH**
    - Server increments x by one, picks own random sequence number y, appends its own set of flags and options, and dispatches the response.
- **PARAGRAPH**
    - ACK
- **PARAGRAPH**
    - Client increments both x and y by one and completes the handshake by dispatching the last ACK packet in the handshake.
- **PARAGRAPH**
    - Once the three-way handshake is complete, the application data can begin to flow between the client and the server.
    - The client can send a data packet immediately after the ACK packet, and the server must wait for the ACK before it can dispatch any data.
    - This startup process applies to every TCP connection and carries an important implication for performance of all network applications using TCP: each new connection will have a full roundtrip of latency before any application data can be transferred.
- **PARAGRAPH**
    - For example, if our client is in New York, the server is in London, and we are starting a new TCP connection over a fiber link, then the three-way handshake will take a minimum of 56 milliseconds (Table 1-1): 28 milliseconds to propagate the packet in one direction, after which it must return back to New York.
    - Note that bandwidth of the connection plays no role here.

- Instead, the delay is governed by the latency between the client and the server, which in turn is dominated by the propagation time between New York and London.

- **PARAGRAPH**
    - The delay imposed by the three-way handshake makes new TCP connections expensive to create, and is one of the big reasons why connection reuse is a critical optimization for any application running over TCP.

- **PARAGRAPH**
    - Congestion Avoidance and Control In early 1984, John Nagle documented a condition known as “congestion collapse,”

- **PARAGRAPH**
    - which could affect any network with asymmetric bandwidth capacity between the nodes:

- **PARAGRAPH**
    - Congestion control is a recognized problem in complex networks.
    - We have discovered that the Department of Defense’s Internet Protocol (IP), a pure datagram protocol, and Transmission Control Protocol (TCP), a transport layer protocol, when used together, are subject to unusual congestion problems caused by interactions between the transport and datagram layers.
    - In particular, IP gateways are vulnerable to a phenomenon we call

- **PARAGRAPH**
    - “congestion collapse”, especially when such gateways connect networks of widely different bandwidth…

- **PARAGRAPH**
    - Should the roundtrip time exceed the maximum retransmission interval for any host, that host will begin to introduce more and more copies of the same datagrams into the net.
    - The network is now in serious trouble.
    - Eventually all available buffers in the switching nodes will be full and packets must be dropped.
    - The roundtrip time for packets that are delivered is now at its maximum.
    - Hosts are sending each packet several times, and even-tually some copy of each packet arrives at its destination.
    - This is congestion collapse.
- **PARAGRAPH**
    - This condition is stable.
    - Once the saturation point has been reached, if the algorithm for selecting packets to be dropped is fair, the network will continue to operate in a degraded condition.
- **PARAGRAPH**
    - — John Nagle
- **PARAGRAPH**
    - RFC 896

- **PARAGRAPH**
    - The report concluded that congestion collapse had not yet become a problem for AR-PANET because most nodes had uniform bandwidth, and the backbone had substantial excess capacity.
    - However, neither of these assertions held true for long.
    - In 1986, as the number (5,000+) and the variety of nodes on the network grew, a series of congestion collapse incidents swept throughout the network—in some cases the capacity dropped by a factor of 1,000 and the network became unusable.
- **PARAGRAPH**
    - To address these issues, multiple mechanisms were implemented in TCP to govern the rate with which the data can be sent in both directions: flow control, congestion control, and congestion avoidance.
- **PARAGRAPH**
    - Flow Control
- **PARAGRAPH**
    - Flow control is a mechanism to prevent the sender from overwhelming the receiver with data it may not be able to process—the receiver may be busy, under heavy load, or may only be willing to allocate a fixed amount of buffer space.
    - To address this, each side of the TCP connection advertises (Figure 2-2) its own receive window (rwnd), which communicates the size of the available buffer space to hold the incoming data.
- **PARAGRAPH**

  - When the connection is first established, both sides initiate their rwnd values by using their system default settings.

  - A typical web page will stream the majority of the data from the server to the client, making the client's window the likely bottleneck.

  - However, if a client is streaming large amounts of data to the server, such as in the case of an image or a video upload, then the server receive window may become the limiting factor.

- **PARAGRAPH**

  - If, for any reason, one of the sides is not able to keep up, then it can advertise a smaller window to the sender.

  - If the window reaches zero, then it is treated as a signal that no more data should be sent until the existing data in the buffer has been cleared by the application layer.

  - This workflow continues throughout the lifetime of every TCP connection: each ACK packet carries the latest rwnd value for each side, allowing both sides

- **PARAGRAPH**

  - to dynamically adjust the data flow rate to the capacity and processing speed of the sender and receiver.

- **PARAGRAPH**

  - Slow-Start

- **PARAGRAPH**

  - Despite the presence of flow control in TCP, network congestion collapse became a real issue in the mid to late 1980s.

- The problem was that flow control prevented the sender from overwhelming the receiver, but there was no mechanism to prevent either side from overwhelming the underlying network: neither the sender nor the receiver knows the available bandwidth at the beginning of a new connection, and hence need a mechanism to estimate it and also to adapt their speeds to the continuously changing conditions within the network.

- **PARAGRAPH**
    - To illustrate one example where such an adaptation is beneficial, imagine you are at home and streaming a large video from a remote server that managed to saturate your downlink to deliver the maximum quality experience.
    - Then another user on your home network opens a new connection to download some software updates.
    - All of the sudden, the amount of available downlink bandwidth to the video stream is much less, and the video server must adjust its data rate—otherwise, if it continues at the same rate, the data will simply pile up at some intermediate gateway and packets will be dropped, leading to inefficient use of the network.
- **PARAGRAPH**
    - In 1988, Van Jacobson and Michael J. Karels documented several algorithms to address these problems: slow-start, congestion avoidance, fast retransmit, and fast recovery.
    - All four quickly became a mandatory part of the TCP specification.
    - In fact, it is widely held that it was these updates to TCP that prevented an Internet meltdown in the ’80s and the early ’90s as the traffic continued to grow at an exponential rate.
- **PARAGRAPH**

    - To understand slow-start, it is best to see it in action.

    - So, once again, let us come back to our client, who is located in New York, attempting to retrieve a file from a server in London.

    - First, the three-way handshake is performed, during which both sides advertise their respective receive window (rwnd) sizes within the ACK packets (Figure 2-2).

    - Once the final ACK packet is put on the wire, we can start exchanging application data.

- **PARAGRAPH**

    - The only way to estimate the available capacity between the client and the server is to measure it by exchanging data, and this is precisely what slow-start is designed to do.

- **PARAGRAPH**

    - To start, the server initializes a new congestion window (cwnd) variable per TCP connection and sets its initial value to a conservative, system-specified value (initcwnd on Linux).

- **PARAGRAPH**

    - Congestion window size (cwnd)

- **PARAGRAPH**

    - Sender-side limit on the amount of data the sender can have in flight before receiving an acknowledgment (ACK) from the client.

- **PARAGRAPH**

    - The cwnd variable is not advertised or exchanged between the sender and