به نام خدا

ارائه‌ی درس مدارهای واسط
پروتکل TCP

استاد:
دکتر فصحتی

امیرکسری احمدی ۴۰۱۱۷۰۵۰۷

زمستان ۱۴۰۳

در این مستند به معرفی پروتکل TCP می‌پردازیم و این معرفی را در قالب پاسخ به تعدادی سوال تهیه کردیم.

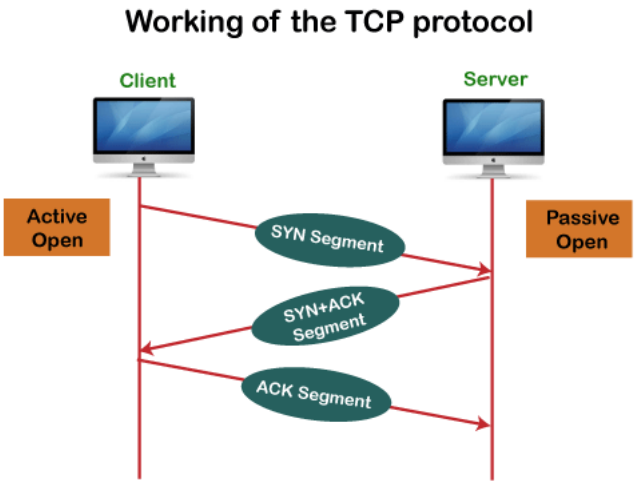
سوال ۱) کاربرد پروتکل و چرایی توسعه این پروتکل را شرح دهید.

کاربرد پروتکل TCP:

پروتکل (Transmission Control Protocol) TCP یکی از پروتکل‌های اصلی در مجموعه‌ی پروتکل‌های اینترنت (TCP/IP) است که وظیفه‌ی انتقال داده‌ها بین دستگاه‌های مختلف در شبکه را بر عهده دارد. TCP یک پروتکل اتصال‌گرا (Connection-Oriented) است، به این معنی که قبل از انتقال داده‌ها، یک اتصال مطمئن بین فرستنده و گیرنده برقرار می‌کند. این پروتکل تضمین می‌کند که داده‌ها به صورت کامل، بدون خطا و به ترتیب صحیح به مقصد برسند. از TCP در برنامه‌هایی مانند مرورگرهای وب (HTTP/HTTPS)، ارسال ایمیل(SMTP) ، انتقال فایل (FTP) و بسیاری از برنامه‌های دیگر که نیاز به انتقال مطمئن داده‌ها دارند، استفاده می‌شود.

چرایی توسعه TCP:

TCP در اوایل دهه‌ی 1970 توسط وینت سرف و باب کان توسعه یافت. هدف اصلی از توسعه TCP، ایجاد یک پروتکل ارتباطی بود که بتواند در شبکه‌های بزرگ و ناهمگن (مانند اینترنت) کار کند و قابلیت اطمینان و یکپارچگی داده‌ها را تضمین کند. در آن زمان، شبکه‌های کامپیوتری در حال گسترش بودند و نیاز به یک پروتکل استاندارد برای انتقال داده‌ها به صورت مطمئن و کارآمد احساس می‌شد. TCP با ارائه‌ی مکانیزم‌هایی مانند کنترل جریان (Flow Control)، کنترل ازدحام (Congestion Control) و تصحیح خطا (Error Correction)، این نیازها را برطرف کرد.



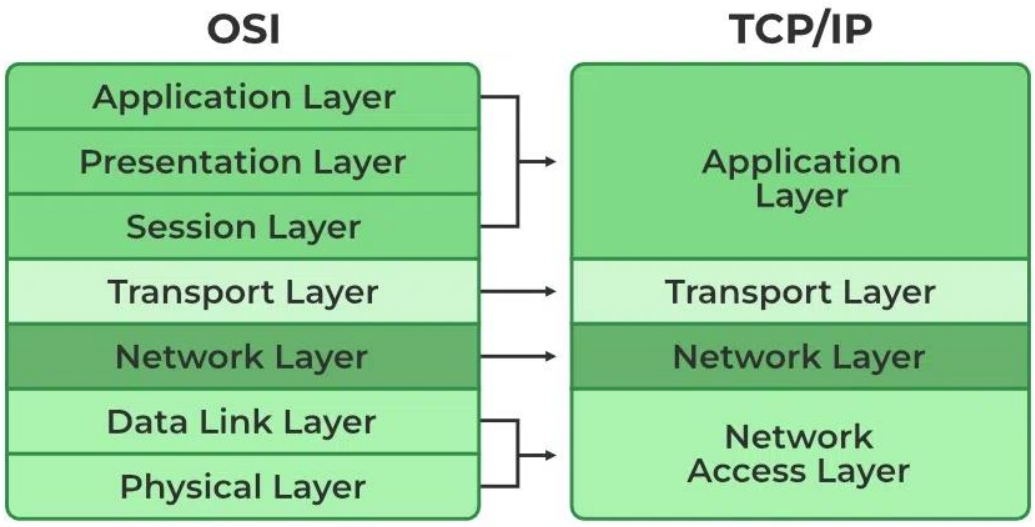
سوال ۲) اتصالات و مدارات لایه فیزیکی این پروتکل با رسم شکل توضیح داده شود. آیا لایه فیزیکی از سیگنالینگ تفاضلی استفاده میکند؟ اگر یا این طور است مدار آن شرح داده شود. اتصالات ضروري اين پروتکل را نام ببريد، آيا اتصالات اختياري هم دارد؟

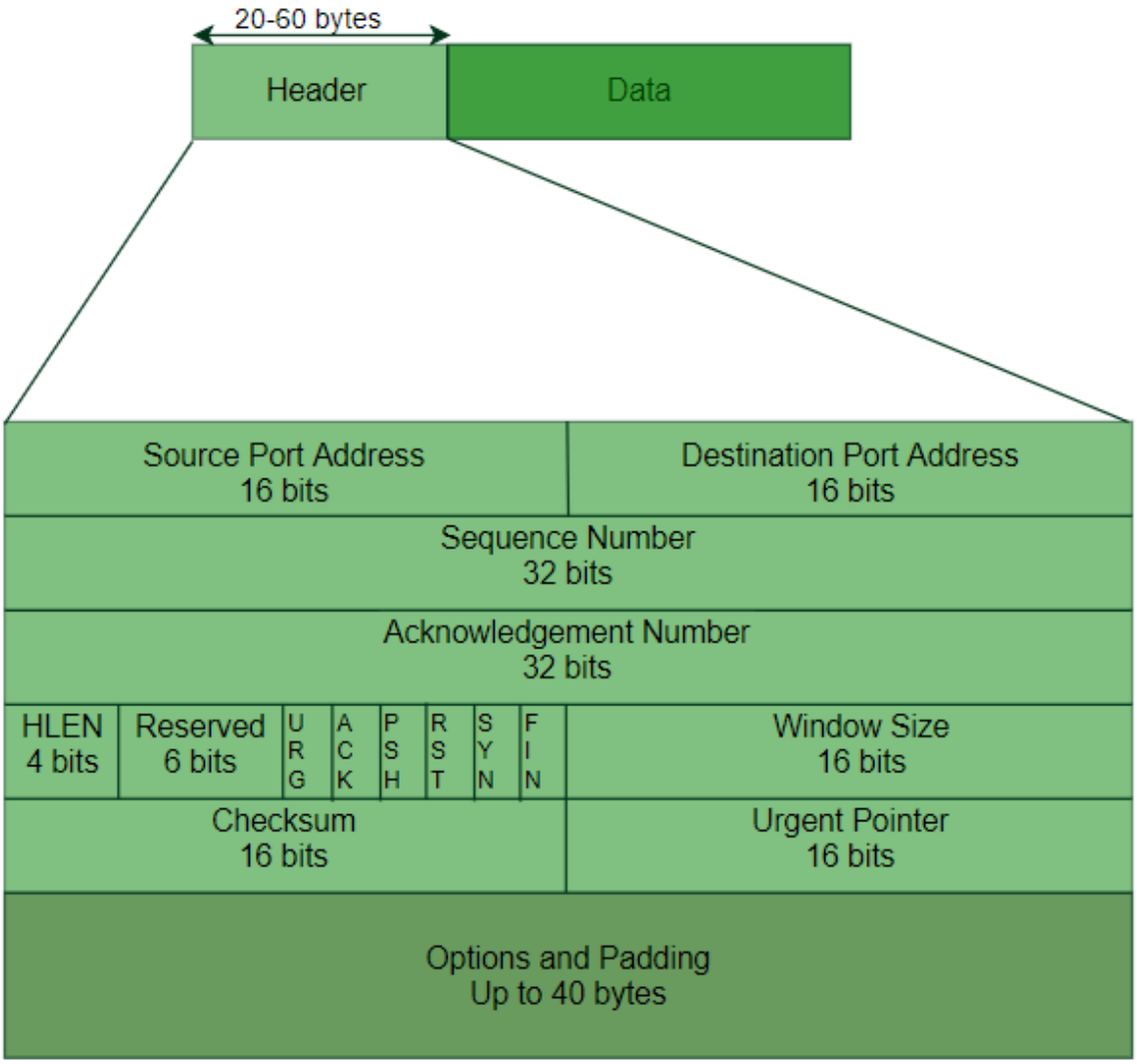
لایه فیزیکی (Physical Layer) پایین‌ترین لایه در مدل OSI و اولین لایه در مدل TCP/IP است. این لایه مسئول انتقال بیت‌های خام (Raw Bits) از طریق یک محیط فیزیکی مانند کابل‌های مسی، فیبر نوری یا

امواج رادیویی است. در مورد TCP، لایه فیزیکی به خود TCP مربوط نمی‌شود، زیرا TCP یک پروتکل لایه انتقال (Transport Layer) است. با این حال، لایه فیزیکی نقش حیاتی در انتقال داده‌هایی که TCP مدیریت می‌کند، ایفا می‌کند.
TCP نقشی در تعیین مکانیزم ارسال پیام‌ها در لایه‌ی فیزیکی ندارد زیرا این پروتکل لایه‌ی انتقال است در نتیجه ممکن است سیگنالینگ تفاضلی استفاده شود اما الزامی نیست.

اتصالات ضروری TCP:

- **اتصال سه‌مرحله‌ای (Three-Way Handshake):** این اتصال برای برقراری یک جلسه ارتباطی بین دو دستگاه استفاده می‌شود. مراحل آن عبارتند از:
  1. SYN: فرستنده یک بسته SYN به گیرنده ارسال می‌کند.
  2. SYN-ACK: گیرنده با یک بسته SYN-ACK پاسخ می‌دهد.
  3. ACK: فرستنده یک بسته ACK ارسال می‌کند و اتصال برقرار می‌شود.
- **اتصالات اختیاری TCP:**
  - **گزینه‌های TCP (TCP Options):** این گزینه‌ها برای پیکربندی پیشرفته‌تر اتصال استفاده می‌شوند، مانند تنظیم اندازه پنجره (Window Scaling) یا فعال‌سازی فشرده‌سازی داده‌ها.
  - **پورت‌های اختیاری:** در برخی موارد، پورت‌های خاصی برای اهداف خاص (مانند پورت‌های غیراستاندارد) استفاده می‌شوند.





سوال ۳) ارتباط در این پروتکل سریال است یا موازی؟ نوع انکودینگ این پروتکل چیست؟ و نحوه تولید سیگنال را با رسم شکل توضیح دهید. روش انتقال آن هم زمان است یا ناهم زمان؟

ارتباط در پروتکل TCP:

ارتباط در پروتکل TCP به صورت **سریال** است. این بدان معناست که داده‌ها به صورت بیت‌به‌بیت و پشت سر هم (یکی پس از دیگری) از طریق یک کانال ارتباطی ارسال می‌شوند.TCP از لایه‌های پایین‌تر شبکه (مانند لایه فیزیکی و لایه پیوند داده) برای انتقال بیت‌ها استفاده می‌کند، که معمولاً به صورت سریال انجام می‌شود.

نوع انکودینگ در TCP:

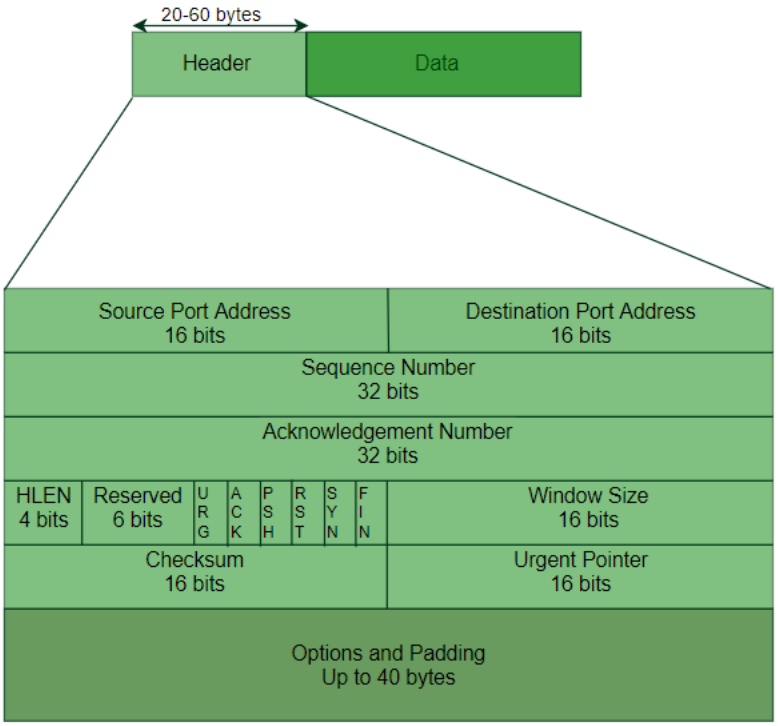
TCP خودش به طور مستقیم مسئول انکودینگ داده‌ها نیست، زیرا این کار توسط لایه‌های پایین‌تر (مانند لایه فیزیکی) انجام می‌شود. با این حال، TCPداده‌ها را در قالب **بایت‌ها (8 بیتی)** سازماندهی می‌کند

و آن‌ها را در **بسته‌های TCP (TCP Segments)** قرار می‌دهد. هر بسته TCP شامل یک هدر (Header) و بخش داده (Payload) است. هدر TCP شامل اطلاعات کنترل‌کننده مانند شماره پورت‌ها، شماره‌های توالی (Sequence Numbers) و چک‌سام (Checksum) است.

در لایه فیزیکی، انکودینگ داده‌ها می‌تواند به روش‌های مختلفی انجام شود، مانند:

- **انکودینگ NRZ (Non-Return-to-Zero):** یک روش ساده برای انکودینگ بیت‌ها.
- **انکودینگ Manchester:** برای هم‌زمان سازی سیگنال‌ها استفاده می‌شود.
- **انکودینگ 8b/10b:** برای انتقال داده‌ها در شبکه‌های پرسرعت مانند Ethernet.

ساختار سگمنت‌های TCP:



نحوه تولید سیگنال:
در لایه فیزیکی، داده‌ها به صورت سیگنال‌های الکتریکی، نوری یا رادیویی تبدیل می‌شوند. در نتیجه مدیریت و نحوه‌ی تولید سیگنال در این پروتکل بررسی نمی‌شود.

روش انتقال:
انتقال داده‌ها در TCP به صورت **هم‌زمان (Synchronous)** است. این بدان معناست که فرستنده و گیرنده از یک ساعت (Clock) مشترک برای هم‌زمان‌سازی استفاده می‌کنند. در لایه فیزیکی، این هم‌زمان‌سازی معمولاً با استفاده از سیگنال‌های ساعت یا تکنیک‌های انکودینگ مانند Manchester

انجام می‌شود. TCP از شماره‌های توالی (Sequence Numbers) و تاییدیه‌ها (Acknowledgements) برای اطمینان از تحویل صحیح و مرتب داده‌ها استفاده می‌کند.

سوال ۴) آیا این پروتکل را میتوان جهت اتصال چندین دستگاه سخت‌افزاری استفاده کرد؟ اگر پاسخ مثبت است، نحوه اتصال آنها را توضیح دهید. همچنین چالش مدیریت برخورد در این پروتکل را توضیح دهید. اگر قابلیت اتصال دو دستگاه سخت افزاري را فقط دارد؛ توضیح دهید که چرا قابلیت اتصال به چند دستگاه/ماژول سخت افزاري براي آن در نظر گرفته نشده است؟

در مورد اتصال چند دستگاه سخت افزاری با کمک گرفتن از بقیه‌ی لایه‌ها پاسخ مثبت است اما اگر تنها به این پروتکل نگاه کنیم جواب خیر است. پروتکل TCP می‌تواند برای اتصال چندین دستگاه سخت‌افزاری استفاده شود، اما این کار به طور مستقیم توسط TCP انجام نمی‌شود. TCP یک پروتکل لایه انتقال (Transport Layer)است که ارتباط نقطه‌به‌نقطه (Point-to-Point) بین دو دستگاه (یک فرستنده و یک گیرنده) را مدیریت می‌کند. برای اتصال چندین دستگاه، معمولاً از پروتکل‌ها و فناوری‌های لایه‌های دیگر (مانند لایه شبکه و لایه پیوند داده) استفاده می‌شود.

نحوه اتصال چندین دستگاه با استفاده از TCP:

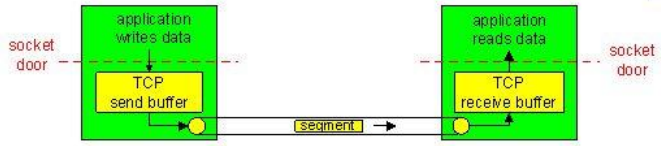
برای اتصال چندین دستگاه سخت‌افزاری با استفاده ازTCP ، معمولاً از یک **سوئیچ شبکه (Switch)** یا **روتر(Router)** استفاده می‌شود. این دستگاه‌ها در لایه شبکه و لایه پیوند داده کار می‌کنند و امکان ارتباط چندین دستگاه را فراهم می‌کنند. در این حالت، هر دستگاه می‌تواند با استفاده از آدرس IP و پورت‌های TCP به دستگاه‌های دیگر متصل شود. به عنوان مثال:

1. **سوئیچ شبکه:** دستگاه‌ها به سوئیچ متصل می‌شوند و سوئیچ بسته‌های داده را بین آن‌ها مسیریابی می‌کند.
2. **روتر:** اگر دستگاه‌ها در شبکه‌های مختلف قرار داشته باشند، روترها بسته‌های TCP را بین شبکه‌ها منتقل می‌کنند.
3. **سرور مرکزی:** یک سرور می‌تواند به عنوان نقطه مرکزی عمل کند و ارتباطات TCP بین چندین دستگاه را مدیریت کند.

TCP: Overview

RFCs: 793, 1122, 1323, 2018, 2581

- point-to-point:
  - one sender, one receiver
- reliable, in-order *byte steam:*
  - no "message boundaries"
- pipelined:
  - TCP congestion and flow control set window size
- *send & receive buffers*



- full duplex data:
  - bi-directional data flow in same connection
  - MSS: maximum segment size
- connection-oriented:
  - handshaking (exchange of control msgs) init's sender, receiver state before data exchange
- flow controlled:
  - sender will not overwhelm receiver

چالش مدیریت برخورد در TCP:

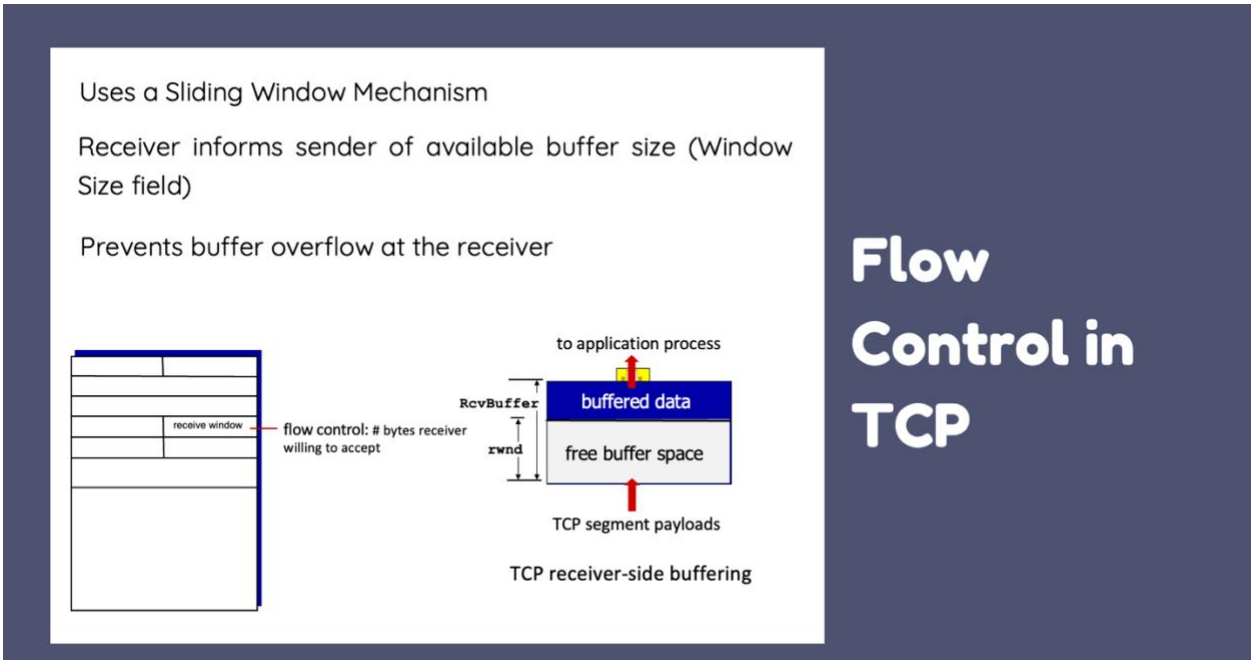
مدیریت برخورد (Collision Management) بیشتر در لایه پیوند داده (Data Link Layer) و پروتکل‌هایی مانند Ethernet مطرح می‌شود. TCP به خودی خود مکانیزم خاصی برای مدیریت برخورد ندارد، زیرا این کار توسط لایه‌های پایین‌تر انجام می‌شود. با این حال، TCP از مکانیزم‌هایی مانند **کنترل ازدحام (Congestion Control)** و **کنترل جریان(Flow Control)** استفاده می‌کند تا از overload شدن شبکه و از دست رفتن بسته‌ها جلوگیری کند.

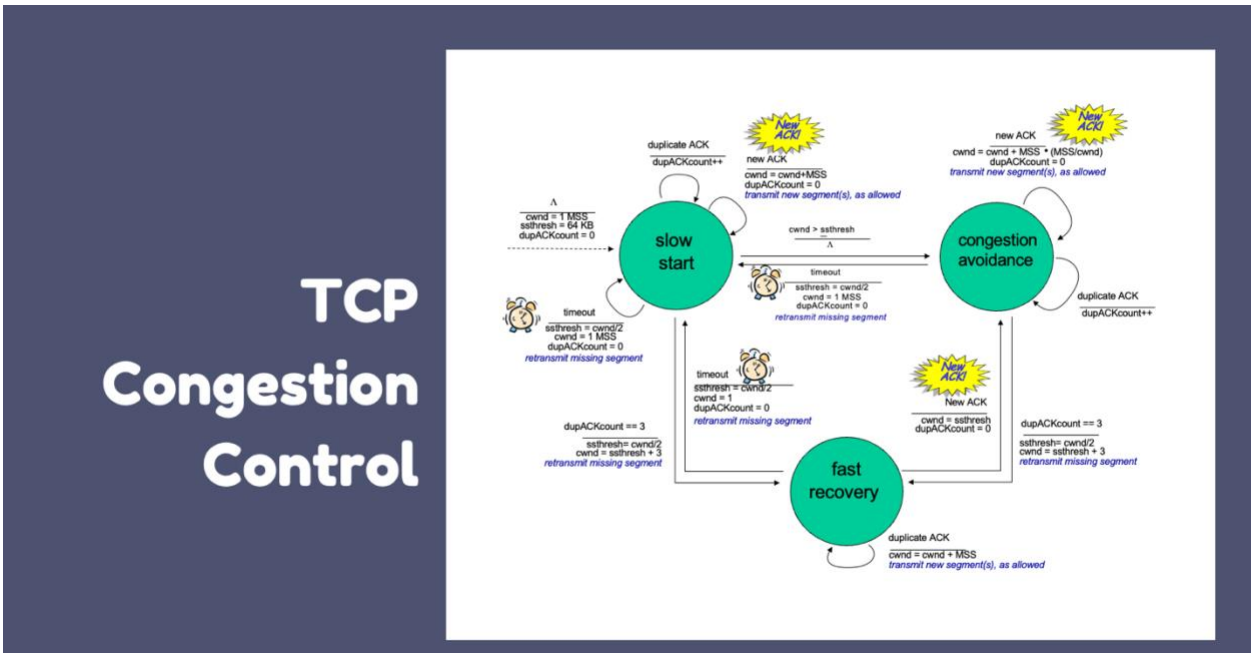
TCP به‌دلیل ماهیت طراحی خود، یک پروتکل **اتصال‌گرا (Connection-Oriented)** و **نقطه‌به‌نقطه (Point-to-Point)** است. این بدان معناست که TCP برای ایجاد یک ارتباط مستقیم و مطمئن بین دو دستگاه طراحی شده است. دلایل اصلی این طراحی عبارتند از:

1. **سادگی و کارایی:** TCP با تمرکز بر ارتباط دو دستگاه، سادگی و کارایی را در مدیریت اتصالات و تضمین تحویل داده‌ها افزایش می‌دهد.
2. **تضمین یکپارچگی داده‌ها:** TCP مکانیزم‌هایی مانند شماره‌های توالی (Sequence Numbers) و تاییدیه‌ها (Acknowledgements) را برای تضمین تحویل صحیح و مرتب داده‌ها استفاده می‌کند. این مکانیزم‌ها در یک ارتباط نقطه‌به‌نقطه به راحتی قابل مدیریت هستند.

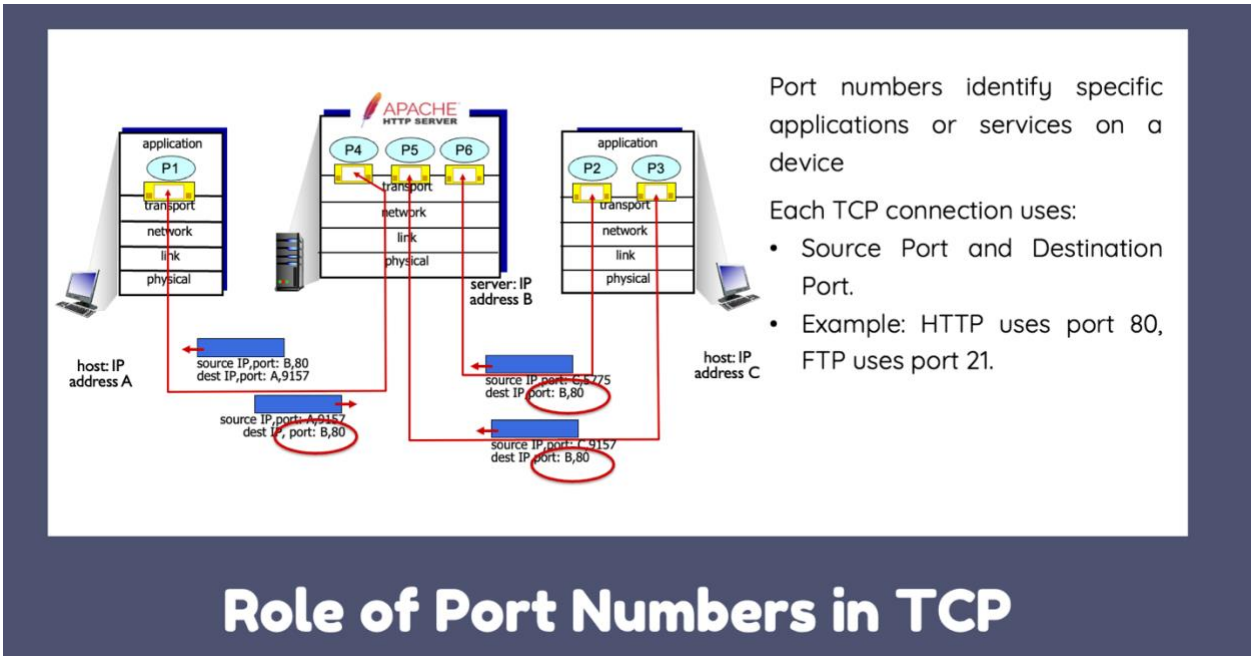
3. **مدیریت ازدحام:** TCP از الگوریتم‌های کنترل ازدحام (مانند TCP Reno یا TCP Cubic) استفاده می‌کند که برای ارتباطات دوطرفه بهینه‌سازی شده‌اند.

برای اتصال چندین دستگاه، از پروتکل‌های لایه‌های دیگر (مانند IP برای آدرس‌دهی و مسیریابی) و دستگاه‌های شبکه (مانند سوئیچ‌ها و روترها) استفاده می‌شود.





سوال ۵) آدرس دهی و مسیریابی در این پروتکل را با رسم شکل توضیح دهید و اگر این پروتکل/گذرگاه شامل این دو ویژگی نیست، با ذکر دلیل توضیح دهید؛ چرا نیاز به این ویژگی در این پروتکل/گذرگاه نیست؟



آدرس‌دهی و مسیریابی در پروتکل TCP:
پروتکل TCP (Transport Control Protocol) یک پروتکل لایه انتقال (Transport Layer) است و **به‌طور مستقیم مسئول آدرس‌دهی و مسیریابی در شبکه نیست**. این وظایف بر عهده لایه‌های پایین‌تر، به ویژه لایه شبکه (Network Layer) با پروتکل IP (Internet Protocol) است. در ادامه توضیحات دقیق‌تر ارائه می‌شود:

۱. آدرس‌دهی در TCP:
- TCP از **پورت‌های منطقی (Port Numbers)** برای شناسایی برنامه‌های کاربردی روی یک دستگاه استفاده می‌کند.
  - هر اتصال TCP با یک جفت **آدرس IP** (برای شناسایی دستگاه) و **پورت** (برای شناسایی برنامه) تعریف می‌شود.
  - مثال:
    - آدرس IP: 192.168.1.10 (دستگاه فرستنده)
    - پورت: 5000 (برنامه روی فرستنده)
    - آدرس IP: 10.0.0.5 (دستگاه گیرنده)
    - پورت: 80 (برنامه روی گیرنده)

- **نقش پورت‌ها:**
  - پورت‌ها امکان چندگانگی (Multiplexing) را فراهم می‌کنند، به طوری که چندین برنامه می‌توانند همزمان از یک دستگاه استفاده کنند.
  - پورت‌های شناخته‌شده (Well-Known Ports) مانند پورت ۸۰ برای HTTP یا پورت ۲۲ برای SSH وجود دارند.

۲ .مسیریابی در TCP:

- TCP **هیچ مکانیزمی برای مسیریابی ندارد** .مسیریابی (انتخاب مسیر مناسب برای رسیدن بسته‌ها به مقصد) توسط پروتکل‌های لایه شبکه (مانندIP) و دستگاه‌های شبکه (مانند روترها) انجام می‌شود.
- TCP تنها مسئول انتقال مطمئن داده‌ها بین دو نقطه انتهایی (Endpoint) است و به مسیر انتقال داده‌ها توجهی ندارد.

چرا TCP به آدرس‌دهی و مسیریابی نیاز ندارد؟

- **تقسیم وظایف در مدل لایه‌ای (TCP/IP یا OSI):**
  - مدل شبکه‌ای به صورت لایه‌ای طراحی شده است تا هر لایه مسئولیت خاصی داشته باشد.
  - لایه انتقال (TCP) روی **تحویل مطمئن داده‌ها** تمرکز دارد، در حالی که آدرس‌دهی و مسیریابی به لایه شبکه (IP) و لایه پیوند داده (Data Link Layer) واگذار شده‌اند.
  - این تقسیم کار باعث ساده‌سازی، انعطاف‌پذیری و کارایی می‌شود.
- **عدم نیاز به افزونگی:**
  - اگر TCP مسیریابی را انجام می‌داد، با عملکرد IP تداخل پیدا می‌کرد و باعث افزونگی غیرضروری می‌شد.

سوال ۶) قابلیت مدیریت جریان داده را توضیح دهید. در این پروتکل نحوه پیاده سازي مدیریت جریان داده را با رسم شکل شرح دهید.

قابلیت مدیریت جریان داده (Flow Control) در TCP:

مدیریت جریان داده در TCP به منظور جلوگیری از ارسال داده با سرعتی بیشتر از توانایی پردازش گیرنده طراحی شده است. این مکانیزم تضمین می‌کند که فرستنده، گیرنده را با حجم غیرقابل مدیریتی از داده‌ها اشباع نکند .TCP برای این کار از **مکانیزم پنجره کشویی (Sliding Window)** و **فیلد پنجره (Window Size)**در هدر خود استفاده می‌کند.

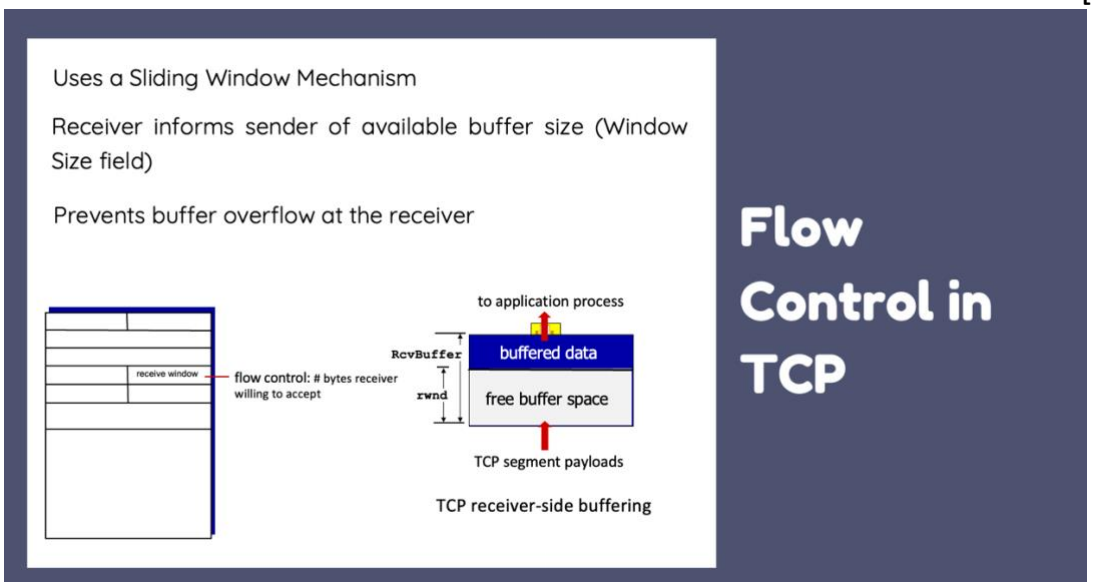
نحوه پیاده‌سازی مدیریت جریان داده:

1. **پنجره کشویی (Sliding Window):**
   - گیرنده ظرفیت بافر خود را به فرستنده اعلام می‌کند (از طریق فیلد **Window Size** در هدر TCP).

- فرستنده تنها مجاز به ارسال داده در محدوده این پنجره است.
- با دریافت تأییدیه (ACK) از گیرنده، پنجره به جلو "می‌لغزد" و فضای جدیدی برای ارسال داده باز می‌شود.

2. **مثال:**
   - اگر گیرنده یک پنجره با اندازه ۳۰۰۰ بایت اعلام کند، فرستنده می‌تواند حداکثر ۳۰۰۰ بایت داده ارسال کند.
   - پس از دریافت ACK برای ۱۰۰۰ بایت اول، پنجره به اندازه ۱۰۰۰ بایت به جلو حرکت می‌کند و فرستنده می‌تواند ۱۰۰۰ بایت جدید ارسال کند.
3. **صفر شدن پنجره (Zero Window):**
   - اگر گیرنده بافر خود را پر کند، پنجره را به اندازه **صفر** اعلام می‌کند.
   - فرستنده تا زمانی که گیرنده یک پنجره جدید (غیرصفر) اعلام نکند، منتظر می‌ماند.

مثال:
فرستنده: [داده‌های ارسال شده | داده‌های تأیید نشده | پنجره آزاد]
گیرنده: [داده‌های دریافت شده | فضای بافر آزاد]
مراحل:
1 .گیرنده Window Size = 3000 بایت را اعلام می‌کند.
2 .فرستنده 3000 بایت ارسال می‌کند.
3 .گیرنده 1000 بایت را پردازش و ACK ارسال می‌کند + Window Size = 2000Byte.
4. پنجره فرستنده به جلو می‌لغزد: [داده‌های تأیید شده | داده‌های تأیید نشده | پنجره جدید (2000 بایت)].



سوال ۷) نحوه تشخیص خطا، در لایه‌هاي متفاوت را با رسم شکل توضیح دهید. دقت فرمایید؛ اکثر پروتکل‌ها تشخیص خطا در لایه داده را دارند. اما بعضی از آنها این امکان را در لایه فیزیکی یا در لایه‌هاي بالاتر هم ایجاد میکنند. شما لازم است این مفهوم را در تمامی لایه‌ها بررسی نمایید.

۱. لایه فیزیکی (Physical Layer)

در این لایه، داده‌ها به صورت بیت‌های خام (۰ و ۱) منتقل می‌شوند. لایه فیزیکی به‌طور پیش‌فرض **فاقد مکانیزم تشخیص خطا** است، زیرا وظیفه اصلی آن انتقال سیگنال‌های الکتریکی/نوری است. با این حال، برخی پروتکل‌ها مانند **کدهای مانچستر (Manchester Encoding)** یا **کدهای خطایابی در کابل‌های فیبر نوری**، از روش‌هایی برای شناسایی خطاهای فیزیکی (مثلاً قطع سیگنال) استفاده می‌کنند
مثال:
در انتقال پی‌سیم، اگر سیگنال به دلیل نویز مخدوش شود، گیرنده ممکن است با استفاده از **الگوریتم‌های تطبیق الگو**، خطا را تشخیص دهد.

۲. لایه داده پیوندی (Data Link Layer)

این لایه اصلی‌ترین لایه برای تشخیص خطا است و از روش‌های زیر استفاده می‌کند:

- **بررسی افزونگی چرخشی (CRC)**: یک مقدار محاسباتی (Checksum) به انتهای فریم اضافه می‌شود. گیرنده با انجام محاسبات CRC روی داده دریافتی، خطا را شناسایی می‌کند.
- **بررسی توازن (Parity Check)**: یک بیت توازن به فریم اضافه می‌شود تا تعداد ۱ها زوج یا فرد شود.

۳ .لایه شبکه (Network Layer)

در این لایه، پروتکل IP از **Checksum در هدر بسته** برای شناسایی خطا در هدر (نه محتوا) استفاده می‌کند. اگر Checksum نادرست باشد، بسته دور انداخته می‌شود.

۴. لایه انتقال (Transport Layer)

- **TCP**: از Checksum برای کل داده و شماره توالی (Sequence Numbers) استفاده می‌کند. اگر Checksum نامعتبر باشد یا شماره توالی گم شود، گیرنده درخواست ارسال مجدد می‌کند.
- **UDP**: Checksum اختیاری است و فقط هدر را بررسی می‌کند.

۵. لایه نشست (Session Layer)

این لایه بیشتر روی مدیریت جلسات ارتباطی (Session) تمرکز دارد، اما برخی پروتکل‌ها مانند **RPC (Remote Procedure Call)** از **بررسی توازن تراکنش‌ها** برای تشخیص خطا در تبادل داده بین جلسات استفاده می‌کنند.

۶. لایه نمایش (Presentation Layer)

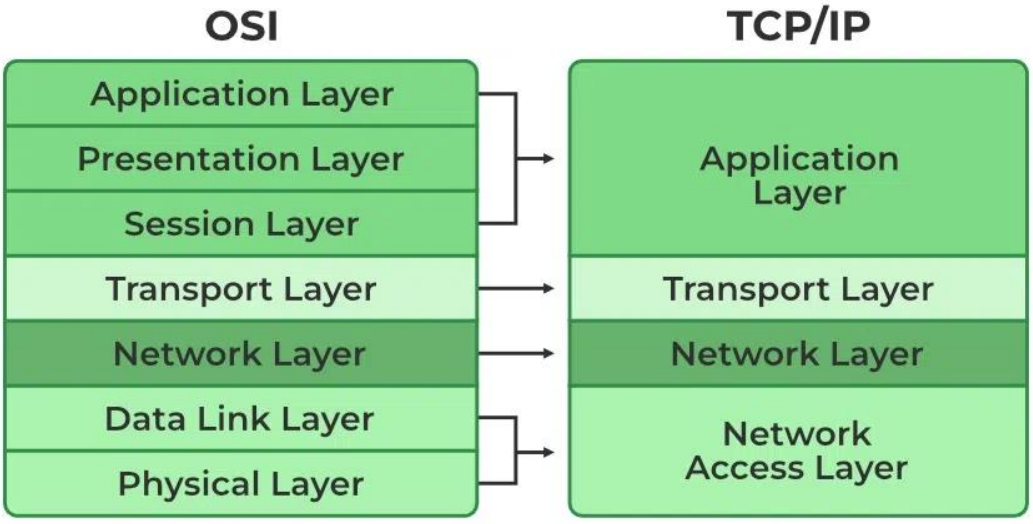
این لایه با تبدیل فرمت داده (مثلاً ASCII به Unicode) و رمزنگاری سروکار دارد. خطاها معمولاً در فرآیند رمزگشایی یا فشرده‌سازی شناسایی می‌شوند.

- **خطا در رمزگشایی:** اگر داده رمزگشایی‌شده با الگوریتم تطابق نداشته باشد، خطا گزارش می‌شود.
- **خطا در فشرده‌سازی**: اگر داده فشرده‌شده غیرقابل بازیابی باشد، خطا رخ می‌دهد.

۷. لایه کاربرد (Application Layer)

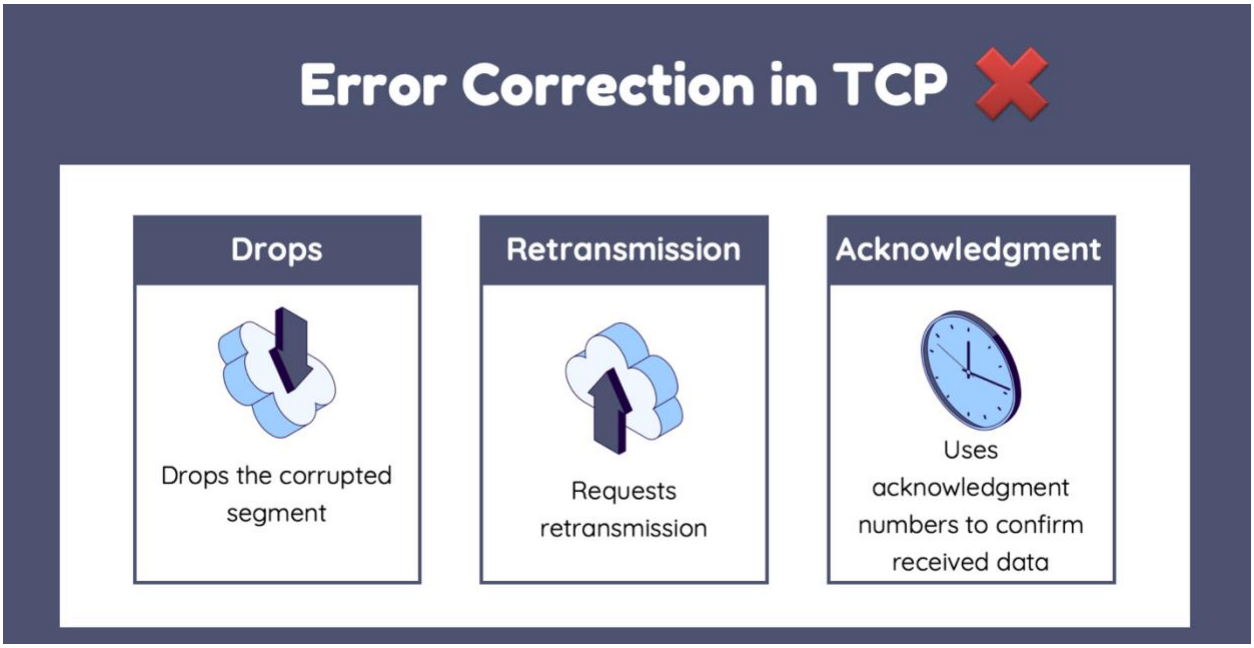
پروتکل‌های این لایه مانند HTTP یا FTP ممکن است از **Checksum داخلی** یا **رمزنگاری** برای تشخیص خطا استفاده کنند.

- **HTTP**: اگر داده دانلودشده با Checksum سرور مطابقت نداشته باشد، خطا نمایش داده می‌شود.
- **FTPS/HTTPS**: خطاهای رمزنگاری (مثلاً گواهی نامعتبر) در این لایه شناسایی می‌شوند.



سوال ۸) آیا در این پروتکل رویکردی براي تصحیح خطا هم داریم؟ اگر پاسخ مثبت است، آن را شرح دهید.

خیر، پروتکل TCP **به‌صورت ذاتی قابلیت تصحیح خطا (Error Correction)** ندارد. یعنی اگر داده‌ای در طول انتقال خراب شود، TCPخطا را **تشخیص** می‌دهد، اما آن را **اصلاح نمی‌کند**.
با این حال، TCP یک پروتکل **قابل اطمینان (Reliable)** است، زیرا با استفاده از مکانیزم **بازفرستادن (Retransmission)**، تضمین می‌کند داده‌ها در نهایت به صورت صحیح و کامل به مقصد برسند.

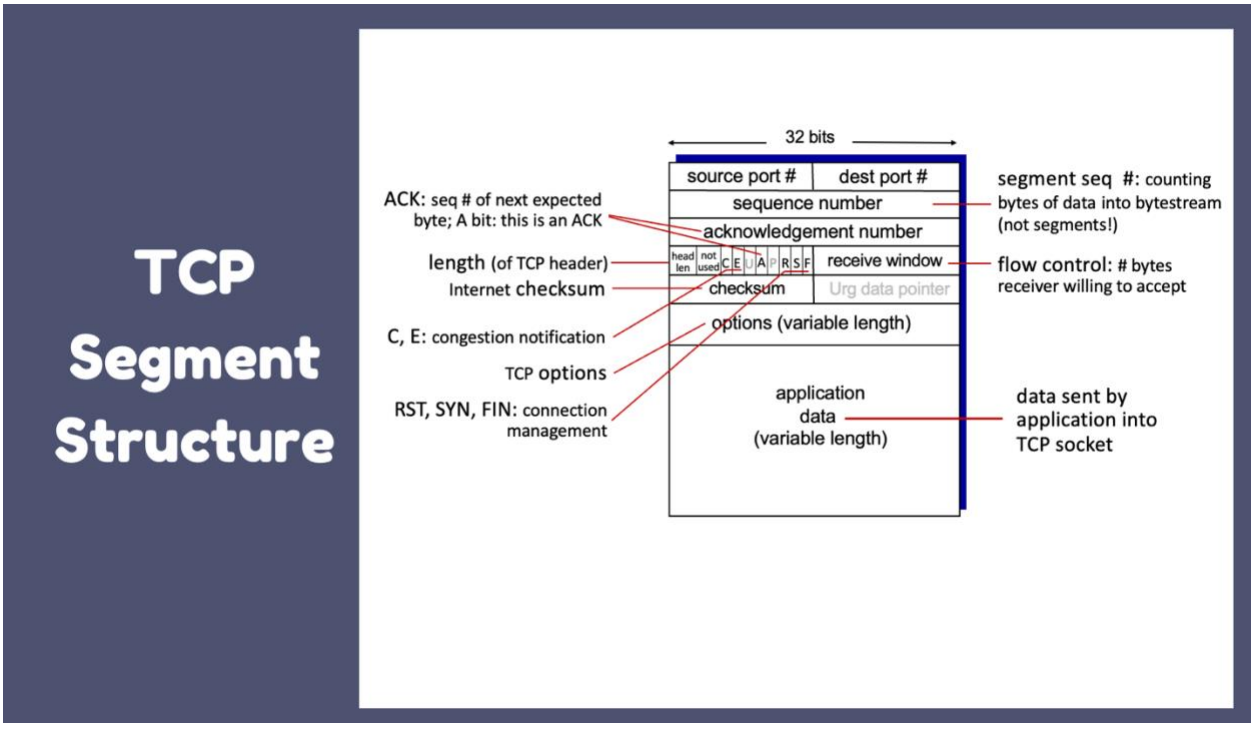


سوال ۹) انواع پیام در این پروتکل را نام ببرید و سپس فرمت هر نوع پیام را با رسم شکل توضیح دهید.

انواع پیام‌ها در پروتکلTCP

TCPاز پیام‌های مختلفی برای **ایجاد اتصال، انتقال داده، کنترل جریان، و قطع اتصال** استفاده می‌کند. این پیام‌ها با استفاده از **پرچم‌ها (Flags)** در هدر TCP تعیین می‌شوند. انواع اصلی پیام‌ها عبارتند از:

1. **پیام**(Synchronize) **SYN** : برای شروع اتصال (Three-Way Handshake).
2. **پیام**(Synchronize-Acknowledge) **SYN-ACK** : پاسخ به SYN.
3. **پیام**(Acknowledge) **ACK** : تأیید دریافت داده.
4. **پیام**(Finish) **FIN** : درخواست پایان اتصال.
5. **پیام**(Reset) **RST** : قطع ناگهانی اتصال.
6. **پیام**(Push) **PSH** : ارسال فوری داده به لایه کاربرد.
7. **پیام**(Urgent) **URG** : داده فوری (کمتر استفاده می‌شود).
8. **پیام داده** (Data Segment): انتقال محتوای اصلی.



منابع:

https://en.wikipedia.org/wiki/Transmission_Control_Protocol
https://gaia.cs.umass.edu/kurose_ross/ppt.php
https://datatracker.ietf.org/doc/html/rfc793