آموزش مفاهیم شبکه به زبان ساده

آشنایی با پروتکل‌ها و مسیریابی شبکه

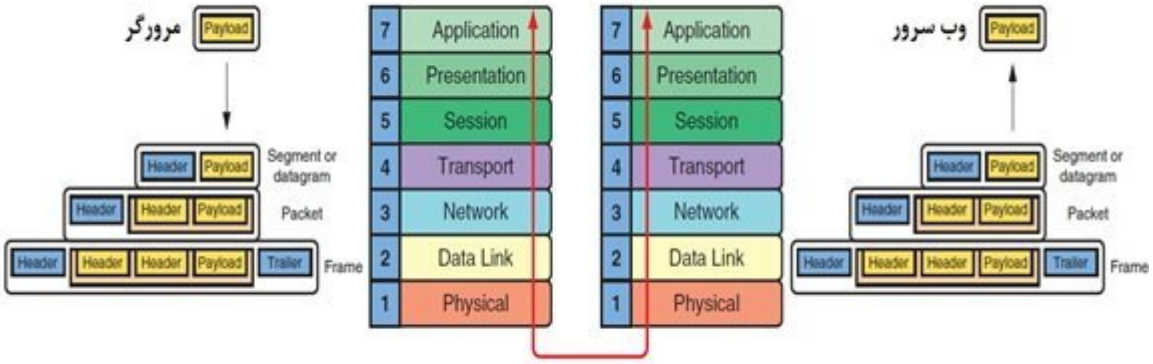
در این مقاله قصد داریم درباره نحوه عملکرد پروتکل‌ها در لایه‌های مختلف و پیام‌هایی که ایجاد می‌کنند اطلاعات بیشتری به دست آورده، با نحوه کار روترها آشنا شده و نیم نگاهی به لایه شبکه داشته باشیم.

این مطلب بخشی از سری آموزش‌های نتورک پلاس است که پیشتر در سایت شبکه منتشر شده است.
برای مشاهده فهرست و خرید کتاب Network+ راهنمای شبکه‌ها اینجا کلیک کنید.

در مقاله‌های آغازین آموزش نتورک‌پلاس به شما گفتیم یک پروتکل نقش حاکمیتی داشته و به کامپیوترهای تحت یک شبکه اعلام می‌کند که دستورالعمل‌ها و داده‌ها را بر مبنای چه روشی انتقال دهند، در ادامه با تجهیزات زیرساختی شبکه آشنا شدیم و اطلاعاتی درباره نحوه کار لایه‌های شبکه، کاربرد، انتقال و پیوند داده به دست آورده و خواندیم که لایه‌ها چگونه به تجهیزات شبکه کمک می‌کنند انواع مختلفی از آدرس‌ها را برای ارسال درست داده‌ها برای دستگاه‌ها یا پردازه‌ها به خدمت گیرند. در ادامه با وظایف لایه‌های مختلف مدل OSI همچون قالب‌بندی، آدرس‌دهی و کشف خطاها اطلاعاتی آشنا شدیم. در بطن همه این فعالیت‌ها و اتفاقاتی که به آن‌ها اشاره داشتیم پروتکل‌ها قرار دارند. اکنون قصد داریم درباره نحوه عملکرد پروتکل‌ها در لایه‌های مختلف و پیام‌هایی که ایجاد می‌کنند اطلاعات بیشتری به دست آورده، با نحوه کار روترها آشنا شده و نیم نگاهی به لایه شبکه داشته باشیم.

پروتکل‌های اصلی TCP/IP

TCP/IP را مجموعه‌ای از پروتکل‌ها یا استانداردهایی همچون ARP، DUP، IP(IPv4/Ipv6)، TCP به انضمام سایر پروتکل‌ها به وجود آورده‌اند. اما چه اتفاقی برای پیام‌های سرآیند در لایه انتقال رخ می‌دهد؟ اجازه دهید خلاصه‌ای از آن‌چه درباره سرآیندها و دنباله فریم آموختید را در قالب شکل زیر به تصویر بکشیم.



هر لایه داده ها و آدرس های خود را برای انتقال اطلاعات به سمت دستگاه موردنظر به لایه های متناظر اضافه می کند

لایه‌های 7، 6 و 5 دستورالعمل‌ها و داده‌هایی هستند که بار داده شناخته می‌شوند. بارداده‌هایی که یک برنامه کاربردی در حال اجرا روی مبدا آن‌ها را تولید می‌کند. به‌طور مثال، در شکل بالا بارداده توسط مرورگر ایجاد شده است، در ادامه بالاترین لایه مدل OSI بار داده به سمت دو لایه بعدی انتقال می‌دهد.

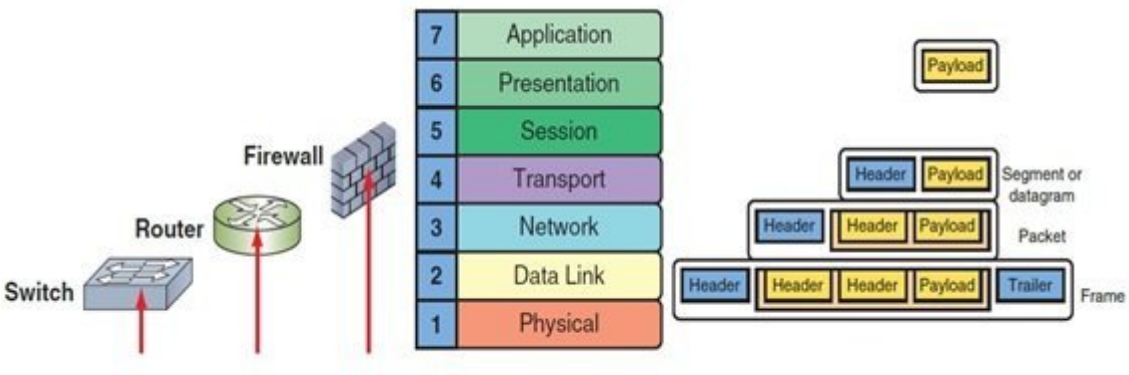
لایه 4- یک **پروتکل** لایه انتقال است که معمولا هر یک از دو **پروتکل** TCP یا UDP از آن استفاده کرده و یک سرآیند به بار داده اضافه می‌کنند. این سرآیند شامل شماره پورتی است که مشخص می‌کند روی یک میزبان چه برنامه‌ای قرار است بسته‌ها را دریافت کند. کل پیام در ادامه به یک سگمنت (زمانی که از TCP استفاده شود) یا دیتاگرام (زمانی که از UDP استفاده می‌شود) تبدیل می‌شود.

لایه 3- لایه شبکه سرآیند خود را به سگمنت یا دیتاگرامی که به برایش ارسال شده اضافه می‌کند. این سرآیند آدرس آی‌پی مقصد و پیامی که پاکت نامیده می‌شود را مشخص می‌کند.

لایه 2- این بسته به سمت لایه پیوند داده روی کارت شبکه هدایت می‌شود. در ادامه بسته با سرآیند و دنباله فریم کپسوله شده و یک فریم ایجاد می‌شود. این فریم لایه شامل یک آدرس فیزیکی است که برای پیدا کردن یک گره روی شبکه محلی استفاده می‌شود.

لایه 1- لایه فیزیکی روی کارت شبکه فریم را دریافت کرده و فرآیند انتقال ملموس روی شبکه را انجام می‌دهد.

میزبان دریافت‌کننده پیام در هر لایه پیام را از حالت کپسوله خارج کرده و سپس اطلاعات بارداده‌ای که دریافت می‌کند را نشان می‌دهد. یک پیام در فرآیند انتقال ممکن است از دستگاه‌های مختلفی همچون سوییچ‌ها و روترها عبور کند. دستگاه‌های اتصال‌دهنده، دستگاه‌های ویژه‌ای هستند که به دو یا چند شبکه یا بخش‌های مختلفی که درون یک شبکه قرار دارند اجازه می‌دهند به یکدیگر متصل شده و به تبادل داده‌ها بپردازند. هر دستگاهی که یک پیام را خوانده و آن‌را پردازش می‌کند با بالاترین سرآیند لایه OSI شناخته می‌شود. به‌طور مثال اگر یک سوییچ سرآیند لایه پیوند را خوانده و پردازش کند، اما فرآیند انتقال پیام را بدون خواندن سرآیندهای لایه بالاتر انجام دهد، به نام سوییچ لایه 2 شناخته می‌شود. به عبارت دیگر، روتری که سرآیند لایه شبکه را خوانده، پردازش کرده و آن‌را به سرآیند لایه انتقال تحویل می‌دهد به نام دستگاه لایه 3 از آن نام برده می‌شود. شکل زیر این مسئله را نشان می‌دهد.



پروتکل کنترل انتقال (TCP) سرنام Transmission Control Protocol

اگر به خاطر داشته باشید، به شما گفتیم **پروتکل** TCP در لایه انتقال از مدل OSI استفاده می‌شود و سرویس‌هایی قابل اعتماد برای تحویل داده‌ها ارائه می‌کند. اجازه دهید برای روشن شدن مطلب پروتکل TCP را با یک تماس تلفنی مقایسه کنیم تا سه ویژگی کاربردی TCP که ضمنات می‌دهند داده‌ها به شکل درستی انتقال پیدا می‌کنند را به خوبی درک کنیم.

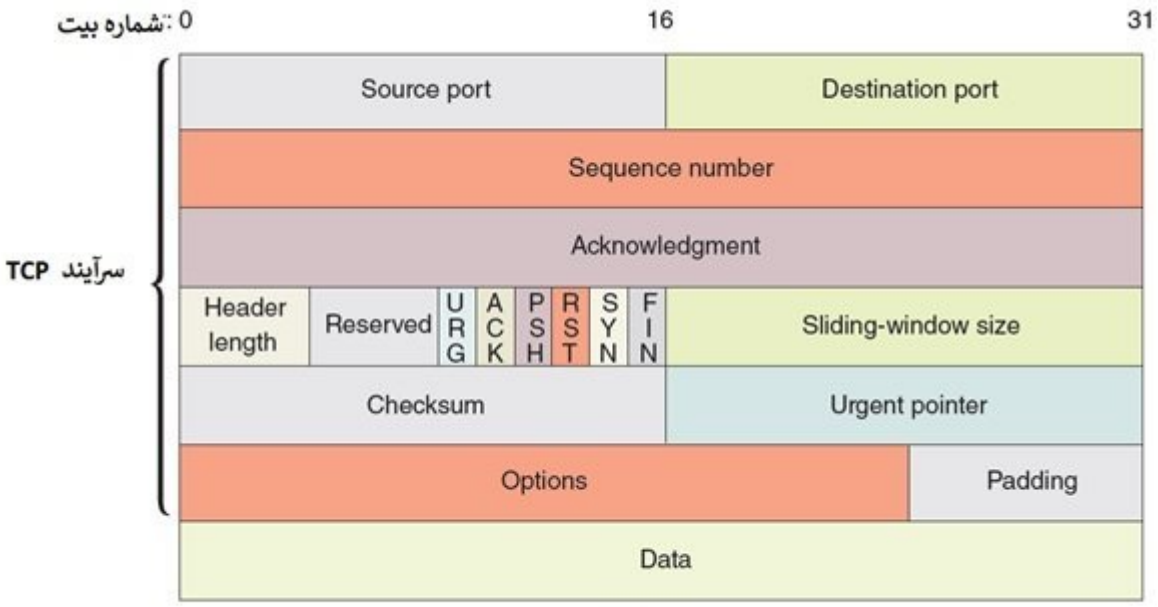
اتصال‌گرایی- قبل از آن‌که TCP داده‌ها را انتقال دهد، این **پروتکل** مطمئن می‌شود که یک ارتباط یا نشستی ایجاد شده است. شبیه به حالتی که یک تماس تلفنی برقرار می‌کنید و پیش از آن‌که صحبت‌های خود را شروع کنید با گفتن سلام مطمئن می‌شود در آن سوی خط فردی در حال گوش دادن است. TCP از یک فرآیند سه مرحله‌ای که به نام دست‌دادن سه طرفه از آن نام برده می‌شود برای برقراری یک ارتباط TCP استفاده می‌کند. TCP تنها پس از آن‌که ارتباطی را برقرار کرد، در ادامه فرآیند انتقال واقعی داده‌ها را آغاز می‌کند. به‌طور مثال پاسخ‌گویی به محاوره‌ای که درباره دسترسی به یک صفحه وب مبتنی بر HTTP ارسال شده است.

توالی و بررسی‌ها- زمانی که یک تماس تلفنی برقرار می‌کنید، ممکن است از شخص مقابل سوال کنید که آیا صدای شما را به درستی می‌شنود و در صورت لزوم جمله‌ای به زبان می‌آورید. مشابه همین حالت در دنیای شبکه رخ می‌دهد، TCP یک رشته کاراکتری که بررسی‌کننده‌ها (checksum) نام دارند را ارسال می‌کند. در طرف دیگر پروتکل TCP میزبان رشته مشابهی را تولید می‌کند. اگر فرآیند تطابق دو رشته با شکست روبرو شود، میزبان از مبدا درخواست می‌کند داده‌ها را دومرتبه ارسال کند. علاوه بر این، به دلیل این‌که پیام‌ها همیشه به همان ترتیبی که ساخته شده‌اند به مقصد نمی‌رسند، TCP یک شماره ترتیب زمانی را برای مشخص کردن هر سگمنت برای میزبان ارسال می‌کند. اگر ضرورتی داشته باشد، در مقصد سگمنت‌هایی که دریافت شده‌اند دومرتبه مرتب می‌شوند.

کنترل جریان- در آن سوی خط اگر فردی که با او صحبت می‌کنید، به درستی متوجه حرف‌های شما نشود، مجبور هستید آهنگ صحبت کردن خود را کند کنید تا واژه‌ها به درستی شنیده شوند. مشابه چنین حالتی در دنیای شبکه کنترل جریان نام دارد. کنترل جریان فرآیندی است که ضمانت می‌کند پیام‌ها با نرخ درستی در حال انتقال هستند. نرخ انتقال داده‌ها بر مبنای سرعتی که گیرنده قادر به دریافت داده‌ها است تنظیم می‌شود. به‌طور مثال، فرض کنید که دریافت‌کننده اعلام می‌دارد که بافر او قادر به مدیریت 4000 بایت است. در این حالت فرستنده فرآیند انتقال را بر مبنای 4000 بایت تنظیم کرده و به انتقال یک یا چند بسته کوتاه همراه با حالت توقف ادامه داده و پیش از فرستادن بسته‌های بعدی کمی مکث می‌کند تا داده‌ها به درستی انتقال پیدا کنند. TCP همه عناصر دخیل در این فرآیند همچون دست دادن سه طرفه، بررسی‌کننده‌ها، توالی و کنترل جریان را با ارسال داده‌ها به فیلدهایی در سرآیند TCP در ابتدای یک سگمنت TCP مدیریت می‌کند.

فیلدهای درون یک سگمنت TCP

شکل زیر فهرستی از آیتم‌هایی را نشان می‌دهد که فیلد نامیده شده و درون یک سگمنت TCP قرار دارند. هر بلوک نشان داده شده در شکل بیان‌گر مجموعه‌ای از بیت‌ها بوده و هر سطر بیان‌گر 32 بیت است. تصویر زیر یک سگمنت TCP را نشان می‌دهد که همه فیلدها به جزء آخرین مورد فیلد داده‌ای بوده و بخشی از سرآیند TCP هستند. محتوای فیلد داده‌ای پیامی است که لایه بالای لایه انتقال آن‌را ارسال کرده است.



یک سگمنت TCP

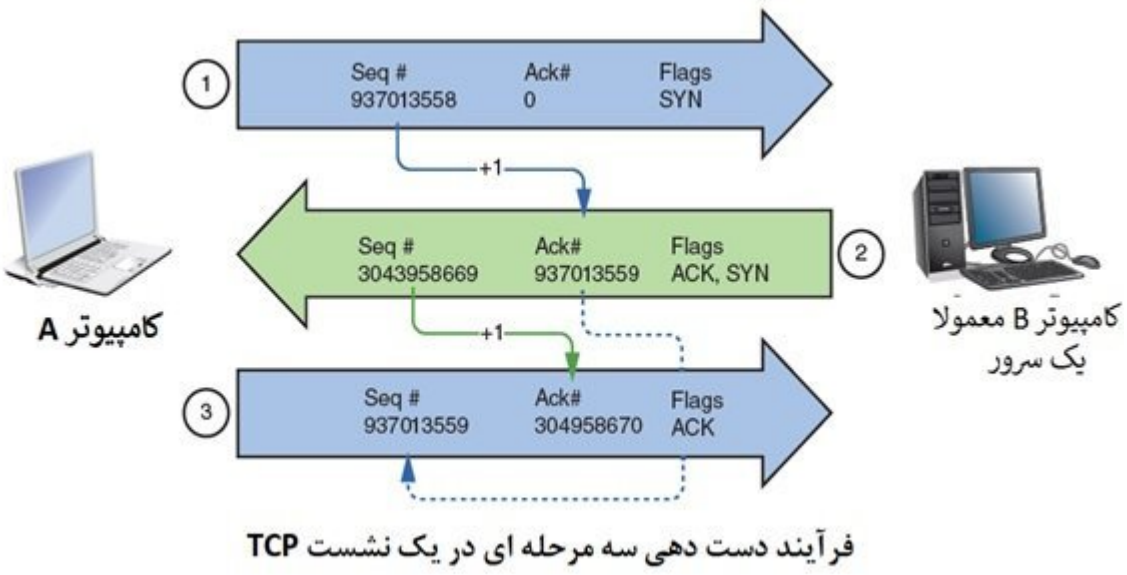
نکته: سرآیندها در گروه‌های 32 ساخته می‌شوند که به آن‌ها words گفته می‌شود. هر word (کلمه) شامل 4 بایت است که بلوک نامیده شده و هر کدام 8 بیت هستند. توضیح فوق به این دلیل آورده شد که سرآیندهای نشان داده شده در تصویر بالا همگی در گروه‌های 32 بیتی نشان داده شده‌اند. توضیح هر یک از فیلدهای تصویر بالا در جدول زیر ارائه شده است. دقت کنید فیلد داده‌ای در انتهای تصویر بخشی از سرآیند TCP نیست. زمانی که سگمنت TCP به سمت پایین یعنی لایه شبکه می‌رود، کل سگمنت به بخش داده‌ای یک سگمنت آی‌پی تبدیل می‌شود. در ادامه این بارداده درون یک بسته آی‌پی کپسوله می‌شود.

فیلد		طول	عملکرد
سرآیند	Source port	bits 16	پورتی در گره مبدا را نشان می‌دهد. یک پورت شماره‌ای است که برای شناسایی یک پردازه روی یک میزبان استفاده می‌شود. پورت به یک پردازه اجازه می‌دهد داده‌هایی را دریافت کرده یا ارسال کند.
	Destination port	bits 16	پورتی در یک گره مقصد را نشان می‌دهد
	Sequence number	bits 16	موقعیت سگمنت داده در یک استریم از سگمنت‌های داده‌ای ارسال شده را نشان می‌دهد. به عبارت ساده‌تر شماره ترتیب آخرین بایت درون فید داده از بسته جاری را نشان می‌دهد.
	Acknowledgment number	bits 32	تایید دریافت اطلاعات از طریق یک پیام بازگشتی برای فرستنده.
	TCP header length	bits 4	طول سرآیند TCP را در بایت نشان می‌دهد. سرآیند می‌تواند حداقل 20 بایت و حداکثر 60 بایت باشد. این فیلد با نام Data offset هم شناخته می‌شود.
	Reserved	bits 6	فیلدی که برای استفاده‌های بعدی رزرو شده است نشان می‌دهد.
	Flags	bits 6	مجموعه‌ای از شش فیلد 1 بیتی یا پرچم‌هایی است که برای شناسایی حالت‌های ویژه سایر فیلدها در یک سرآیند سیگنالی را ارسال می‌کند. فرستنده می‌تواند از فلگ‌های زیر در زمان ارسال داده‌ها استفاده کند: URG: اگر 1 باشد، اعلام می‌دارد که فیلد اشاره‌گر Urgent درون سگمنت اطلاعاتی قرار داده که گیرنده باید آن‌را پردازش کند. اگر 0 باشد، گیرنده از فیلد اشاره‌گر Urgent صرفنظر خواهد کرد. ACK: اگر 1 باشد، به معنای آن است که فیلد Acknowledgment درون سگمنت اطلاعاتی برای گیرنده قرار داده است. اگر 0 باشد، گیرنده از فیلد Acknowledgment خواهد کرد. PSH: اگر به 1 باشد، داده‌ها باید بدون بافر شدن برای یک برنامه ارسال شوند. RST: اگر 1 باشد، فرستنده درخواست کرده تا ارتباط ریست شود. SYN: اگر 1 باشد، فرستنده درخواست یک همگام‌سازی شماره‌های توالی میان دو گره را ارسال کرده است. این کد نشان می‌دهد که هیچ بارداده‌ای درون سگمنت وجود ندارد، و فیلد Acknowledgment number در پاسخ باید 1 واحد افزایش پیدا کند. اگر هر دو فیلد ACK و 1 SYN باشند، هر دو طرف آماده برقراری ارتباط هستند. FIN: اگر 1 باشد، سگمنت آخرین بسته بوده و ارتباط باید بسته شود.
	Sliding-window (size (or window	bits 16	نشان می‌دهد که فرستنده می‌تواند چند بایت را قبل از تأیید گیرنده برای او ارسال کند. این فیلد جریان ارسال داده‌ها را کنترل کرده و مانع از پر شدن بیش از اندازه بافر گیرنده می‌شود.
	Checksum	bits 16	به گره دریافت‌کننده اجازه می‌دهد که تصمیم‌گیری کند که آیا سگمنت TCP می‌تواند در مدت زمان انتقال از بین برود یا خیر
	Urgent pointer	bits 16	بیان‌گر مکانی در یک فیلد داده‌ای است که موقعیت داده‌های مهم درون بسته را مشخص می‌کند.
	Options	32–0 bits	گزینه‌های خاصی را ارائه می‌کند. به‌طور مثال حداکثر اندازه سگمنت که یک شبکه قادر به مدیریت آن است.
	Padding	متغیر	فیلد padding در TCP برای اطمینان از اینکه سرآیند TCP به پایان رسیده استفاده شده و اطمینان می‌دهد که طول سرآیند TCP دارای بیت‌های 32 بیتی است. به عبارت دیگر؛ اطمینان می‌دهد که بخش داده‌ای یک بسته از کراه 32 بیتی آغاز شده و هیچ داده‌ای درون بسته از دست نخواهد رفت.

عملکرد	طول	فیلد	
شامل اطلاعات ارسال شده توسط میزبان است. فیلد داده بخشی از سرآیند TCP نیست، بلکه درون سرآیند TCP کپسوله شده است. اندازه فیلد داده بستگی به میزان داده‌هایی دارد که باید انتقال داده شوند، محدودیت‌هایی که روی اندازه سگمنت TCP اعمال می‌شود به نوع شبکه بستگی دارد.	متغیر	متغیر	

دست دادن سه مرحله‌ای در TCP

دست دادن سه مرحله‌ای به معنای شروع نشست/جلسه‌ای است که پیش از آن‌که TCP داده‌های واقعی را انتقال دهد ایجاد می‌شود. برای درک بهتر این موضوع به زمانی که فرد جدیدی را ملاقات می‌کنید فکر کنید. شما ابتدا دست خود را جلو می‌برید، اما مطمئن نیستید که فرد مقابل به شما پاسخ خواهد داد. اگر شخص مقابل دست خود را جلو بیاورد، شما دو نفر با یکدیگر دست داده و گفت‌وگو را آغاز می‌کنید. شکل زیر انتقال سه مرحله‌ای در یک فرآیند دست‌دهی TCP را نشان می‌دهد.



فرآیند دست دهی سه مرحله ای در یک نشست TCP

جزییات مراحل نشان داده شده در تصویر بالا به شرح زیر است:

مرحله 1، SYN (درخواست برای یک ارتباط) کامپیوتر A پیامی برای کامپیوتر B همراه با اطلاعات زیر ارسال می‌کند.

- در فیلد Sequence number، کامپیوتر A یک عدد تصادفی برای همگام‌سازی ارتباط انتخاب و ارسال می‌کند. در شکل بالا این عدد 937013558 است.
- بیت SYN به 1 تظیم شده است که نشان می‌دهد فلگ SYN فعال شده است. فعال بودن این فلگ نشان می‌دهد که هر دو طرف آماده هستند یک ارتباط را برقرار کنند. کامپیوتر A دست خود را به نشانی برقراری ارتباط برای کامپیوتر B دراز می‌کند تا ببیند آیا پاسخی دریافت می‌کند یا خیر.
- بیت ACK در حالت کلی در اولین انتقال به 0 تنظیم می‌شود، زیرا هنوز هیچ اطلاعاتی از کامپیوتر B برای تایید وجود ندارد.

مرحله 2: SYN/ACK (پاسخ به یک درخواست)- زمانی که کامپیوتر B این پیام را دریافت می‌کند با سگمنتی که حاوی اطلاعات زیر است پاسخ می‌دهد:

- بیت‌های ACK و SYN هر دو به 1 تنظیم می‌شوند. این‌کار به زبان ما می‌شود: "بله، من این‌جا حضور دارم و در حال گوش دادن هستم."
- فیلد Acknowledgment number حاوی عددی است برابر با یک شماره توالی که کامپیوتر A قبلا ارسال کرده است. (به‌علاوه 1). آن‌چنان‌که در شکل بالا نشان داده شده است، کامپیوتر B مقدار 937013559 را

ارسال کرده است. به این ترتیب، کامپیوتر B سیگنالی برای کامپیوتر A ارسال می‌کند که به معنای درخواست برقراری ارتباط است. اکنون کامپیوتر B انتظار دارد تا کامپیوتر A دومرتبه با شماره ترتیبی 937013559 به او پاسخ دهد.
- در فیلد Sequence number، کامپیوتر B شماره تصادفی خود را ارسال می‌کند. در تصویر بالا این شماره برابر با 3043959669 است.

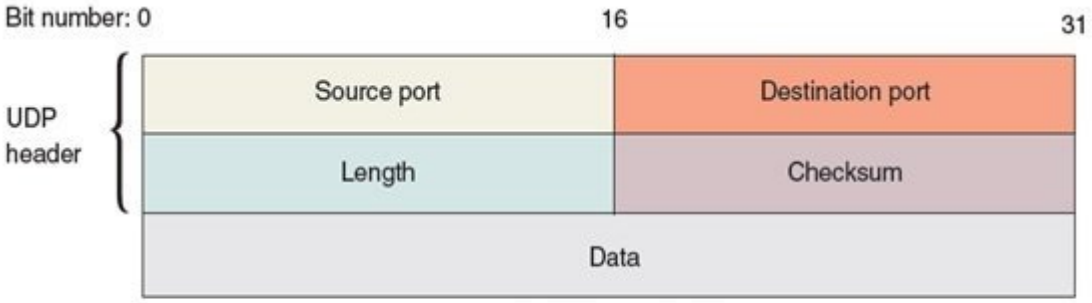
مرحله 3: ACK (اتصال برقرار شد)- کامپیوتر A سگمنتی که حایل اطلاعات زیر است را منتشر می‌کند.

- Sequence number برابر با 937013559 است، زیرا این شماره‌ای است که کامپیوتر B انتظار دارد آن‌را دریافت کند.
- فیلد Acknowledgment number برابر با شماره توالی کامپیوتر B به‌علاوه 1 است. در این مثال این شماره برابر با 3043959670 است.
- بیت ACK به 1 تنظیم شده است. این ارتباط در حال حاضر برقرار شده و در پیام بعدی، کامپیوتر A شروع به ارسال داده‌ها خواهد کرد.

تا این نقطه، هیچ بارداده‌ای در هیچ‌یک از پیام‌های سه مرحله‌ای ضمیمه نشده و تعداد توالی‌ها در هر مرحله 1 واحد افزایش پیدا کرده‌اند. پس از این سه مرحله انتقال، بارداده یا داده‌ها ارسال می‌شود. این‌کار می‌تواند در قالب یک پیام واحد برای حجم کوچکی از داده‌ها از قبیل درخواست برای یک صفحه وب، یا در قالب پیام‌های چندگانه شکسته شده از قبیل ارسال ارسال داده‌هایی که متعلق به یک صفحه وب هستند انجام شود. در این مرحله تعداد توالی‌ها با تعداد بیت‌های موجود در هر سگمنت دریافت شده افزایش پیدا می‌کنند تا مشخص شود طول پیام دریافتی به شکل صحیحی افزایش پیدا کرده است. در شکل بالا، کامپیوتر A پیام بعدی را ارسال خواهد کرد که شامل بارداده‌ای (همچون یک درخواست HTTP) از یک لایه بالاتر است. فرض کنید کامپیوتر A درخواست دسترسی به یک صفحه وب را در قالب یک پیام ارسال کند، چهارمین پیام در این نشست اندازه‌ای برابر با 725 بیت خواهد داشت. کامپیوتر B این پیام را دریافت کرده، تعداد بیت‌ها را شمارش کرده و 725 بیت به شماره توالی پیام دریافت شده یعنی 937013559 اضافه می‌کند. شماره جدید برابر با 937014284 خواهد بود که شماره تایید پیام بازگشتی خواهد بود. (که پنجمین پیام در این نشست خواهد بود.) دو میزبان ارتباط را به همین روش ادامه خواهند داد تا وقتی که کامپیوتر A سگمنتی که بیت FIN آن برابر با 1 است را ارسال کند. یک بودن این بیت نشان می‌دهد که انتقال داده‌ها به پایان رسیده است.

پروتکل بسته داده کاربر (UDP) سرنام User Datagram Protocol

پروتکل بسته داده کاربر موسوم به UDP از یک مدل انتقال ساده بدون ارتباط استفاده کرده که در آن هیچ ارتباط دست‌دهی وجود ندارد، در نتیجه پروتکل قابل اعتمادی نیست. اصطلاح غیر قابل اعتماد بودن به معنای آن نیست که پروتکل UDP بی مصرف بوده و نباید استفاده شود، بلکه منظور این است که این پروتکل هیچ‌گونه تضمینی بابت تحویل داده‌ها ارائه نکرده و پیش از آن‌که فرآیند انتقال داده‌ها آغاز شود هیچ‌گونه اتصالی برقرار نمی‌کند. همان‌گونه که گفتیم پروتکل UDP هیچ‌گونه مکانیزم دست‌دهی در زمان انتشار، تایید دریافت داده‌های منتقل شده، بررسی خطاها، توالی یا کنترل جریان نداشته و به همین دلیل سرعت و کارایی بالاتری نسبت به TCP دارد. عملکرد پروتکل UDP را به جای آن‌که شبیه به یک تماس تلفنی تشریح کنیم، باید شبیه به یک برنامه رادیویی تصور کنیم که سیگنال خود را برای هر کسی که در حال گوش دادن است ارسال می‌کند. UDP برای زمانی که حجم بالایی از داده‌ها باید به سرعت انتقال پیدا کند؛ همچون انتقال داده‌های صوتی یا ویدویی روی اینترنت مناسب است. این پروتکل همچنین برای رسیدگی به درخواست‌های کوچک همچون سامانه نام دامنه یا شرایطی که داده‌ها تغییر پیدا کرده و سرعت نقش مهمی در تکمیل یک پروسه دارد استفاده می‌شود. بازی‌های آنلاین مبتنی بر شبکه از جمله این موارد هستند.
در مقایسه با 10 فیلد سرآیند TCP، سرآیند DUP فقط شامل چهار فیلد پورت مبدا، پورت مقصد، اندازه و Checksum است. دقت کنید که فیلد Checksum این پروتکل به شکل اختیار در شبکه‌های مبتنی بر IPv4 استفاده می‌شود، اما برای تبادلات شبکه‌های مبتنی بر IPv6 ضروری است. شکل زیر دیتاگرام این پروتکل را نشان می‌دهد.



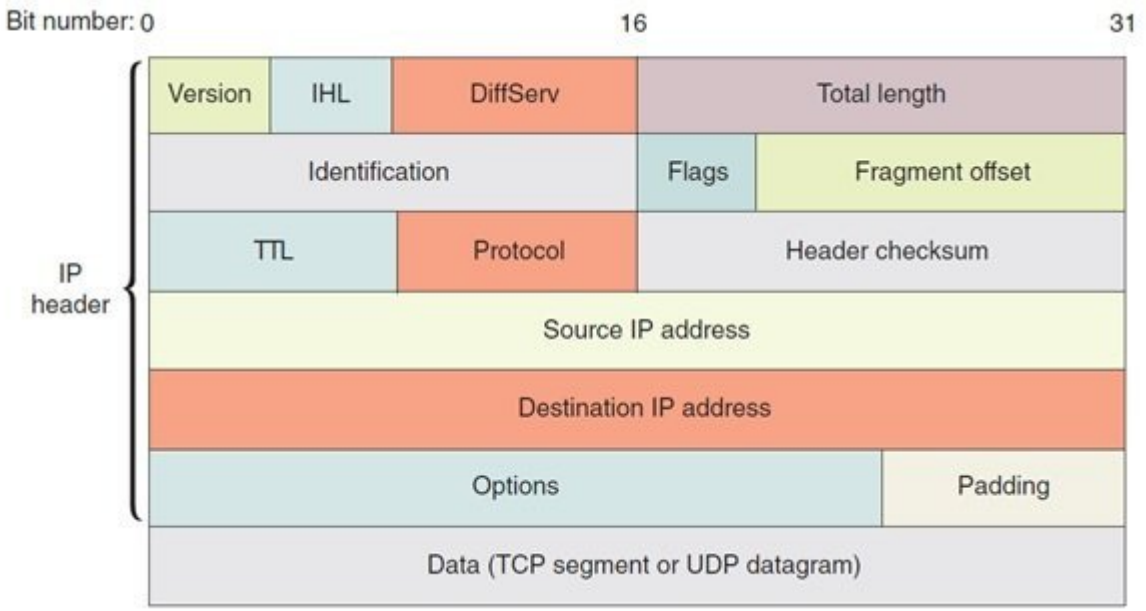
یک دیتاگرام UDP

پروتکل IP سرنام Internet Protocol

پروتکل IP به لایه شبکه در مدل OSI تعلق دارد. این پروتکل مشخص می‌کند که داده‌ها به چه مکانی باید تحویل داده شده و همچنین آدرس‌های آی‌پی مبدا و مقصد را مشخص می‌کند. IP پروتکلی است که TCP/IP را به شبکه متصل می‌کند. به عبارت دقیق‌تر به این پروتکل اجازه می‌دهد از میان شبکه‌های محلی مختلف با اتکا بر روترها عبور کند. همان‌طور که پیش‌تر گفتیم، در لایه شبکه از مدل OSI، داده‌ها درون بسته‌هایی سازمان‌دهی می‌شوند. یک بسته آی‌پی شامل اطلاعات مهمی است که روترها برای انتقال داده‌ها میان سگمنت‌های مختلف شبکه‌های محلی به آن‌ها نیاز دارند. IP یک پروتکل بدون اتصال است، به این معنی که IP نشستی برای ارسال بسته‌های خود منتشر نمی‌کند. هر بسته آی‌پی به‌طور جداگانه از سایر بسته‌هایی که درون مجموعه خودش قرار دارد ارسال می‌شود، در نتیجه برخی از پیام‌ها ممکن است از مسیرهایی متفاوت از دیگری برای رسیدن به مقصد استفاده کنند. تایید این مسئله که آیا آی‌پی پیامی را به میزبان درستی تحویل داده است یا خیر بر عهده پروتکل TCP است. IP با اتکا بر پروتکل TCP یا UDP مطمئن می‌شود که هر پیام به برنامه درستی که روی میزبانی در حال اجرا است، تحویل داده شده است. همان‌گونه که عنوان شد، دو نسخه از پروتکل IP روی شبکه‌های امروزی استفاده می‌شود. IPv4 که اولین بار در سال 1981 معرفی شد و هنوز هم به عنوان استانداردی روی بیشتر شبکه‌ها استفاده می‌شود و IPv6 که در سال 1998 معرفی شد که امنیت بهتر، تنظیمات اولویت‌بندی بهتر، تنظیمات پیکربندی خودکار بهتر و آدرس‌های آی‌پی اضافی‌تر را ارائه می‌کند. بیشتر برنامه‌ها، سرورها، کلاینت‌ها و دستگاه‌های تحت شبکه از IPv6 پشتیبانی می‌کنند. با این حال، هزینه ارتقا زیرساخت‌ها به IPv6 برای بسیاری از سازمان‌ها سنگین بوده و در نتیجه بیشتر سازمان‌ها ترجیح می‌دهند از IPv4 استفاده کنند. به عنوان یک تکنسین شبکه، شما باید اطلاعات کافی در مورد هر دو نسخه این پروتکل به دست آورید. ابتدا اجازه دهید ببینیم بسته‌های IPv4 چگونه ساخته شده و پس از آن به سراغ بسته‌های IPv6 برویم.

بسته‌های IPv4

شکل زیر یک بسته IPv4 را نشان می‌دهد.



توضیح فیلدهای درون تصویر بالا در جدول زیر آماده است. دقت کنید که فیلد داده‌ها در سطر پایین به سرآیند IPv4 تعلق ندارد.

فیلد		طول	عملکرد
سرآیند	Version	bits 4	نسخه پروتکل IP را مشخص می‌کند. به‌طور مثال IPv4 یا Ipv6. یک ایستگاه کاری به فیلد فوق نگاه کرده تا بررسی کند که آیا می‌تواند داده‌های وارد شونده را بخواند یا خیر. اگر موفق نشود بسته را برگشت می‌کند.
	IHL (Internet (header length	bits 4	اندازه سرآیند آی‌پی را در واحد بایت‌ها نشان می‌دهد. این سرآیند می‌تواند حداقل 20 بایت و حداکثر 60 بایت باشد. این فیلد همچنین Data offset نیز نامیده می‌شود، زیرا افست شروع بسته را تاوقتی که داده‌ها از سوی بسته حمل شوند را مشخص می‌کند.
	DiffServ (Differentiated (services	bits 8	برای روترها سطح اولویت‌بندی بسته‌هایی که قرار است پردازش شوند را مشخص می‌کنند.
	Total length	bits 16	طول کل بسته آی‌پی را در واحد بایت مشخص کرده و شامل سرآیند و داده است. یک بسته آی‌پی شامل سرآیند و داده بوده و اندازه آن نباید از 65353 بایت تجاوز کند.
	Identification	bits 16	برخی مواردروترها و میزبان‌های مجبور به شکستن یک دیتاگرام به بسته‌های کوچک‌تر هستند. در این حالت ماشین مقصد مجبور به بازسازی بسته‌ها است. زمانی که یک دیتاگرام واحد شکسته می‌شود، باید ویژگی وجود داشته باشد تا مقصد بتواند بسته‌های دریافتی را بازسازی کرده و آن‌ها را از میان سایر بسته‌های دیتاگرام جدا کند. این فیلد و دو فیلد بعد Flags و Fragment offset به بازسازی بسته‌هایی که جدا دریافت شده‌اند کمک می‌کنند.
	Flags	bits 3	مشخص می‌کند که آیا یک یام شکسته شده و اگر شکسته شده است، آیا بسته‌ای که دریافت شده آخرین قطعه شکسته شده است یا خیر. اولین بیت برای استفاده در آینده رزرو شده است.
	Fragment offset	bits 13	مشخص می‌کند که بسته شکسته شده به چه مکانی در یک مجموعه وارد شده تعلق دارد.
	TTL (Time to (Live	bits 8	حداکثر مدت زمانی را مشخص می‌کند که یک بسته می‌تواند روی یک شبکه پیش از آن‌که از دست برود باقی بماند. درست است که این فیلد واحدی از زمان را نشان می‌دهد، اما روی شبکه‌های مدرن این فیلد تعداد فدعاتی که یک بسته می‌تواند از طریق یک روتر فوروارد شده یا حداکثر تعداد دفعاتی که بسته از هر روتر می‌تواند عبور کند را نشان می‌دهد. مقدار TTL برای هر بسته متفاوت بوده و قابل پیکربندی است. به‌طور معمول این مقدار به 32 یا 64 تنظیم می‌شود. هر بار که بسته‌ای از یک روتر عبور می‌کند، TTL یک واحد کاهش پیدا می‌کند. زمانی که یک روتر یک بسته را با TTL عادل با 0 دریافت می‌کند، آن‌را بسته را حذف کرده و یک پیام اتمام زمان پیام TTL را از طریق پروتکل ICMP برای مبدا ارسال می‌کند.
	Protocol	bits 8	نوع پروتکلی که بسته را دریافت می‌کند را مشخص می‌کند. (به‌طور مثال TCP، UDP یا ICMP)
	Header checksum	bits 16	به میزبان دریافت کننده بسته اجازه می‌دهد تا محاسبه کند که آیا سرآیند آی‌پی در هنگام دریافت بسته خراب شده است یا خیر. اگر فرآیند تطابق و ارزیابی وضعیت بسته‌های دریافتی درست نباشد به معنای آن است که بسته از دست رفته است.
	Source IP address	bits 32	آدرس آی‌پی مبدا را مشخص می‌کند.
	Destination IP address	bits 32	آدرس آی‌پی مقصد را مشخص می‌کد.
	Options	Variable	شامل اطلاعات زمانی و مسیریابی اختیاری است.
	Padding	Variable	شامل بیت‌هایی است که اطمینان می‌دهند که سرآیند دارای بیت‌های 32 است.
	Data	Variable	شامل داده‌هایی است که اساسا از طرف مبدا ارسال شده‌اند و همچنین شامل هر سرآیندی است که از لایه‌های بالاتر دریافت شده‌اند. فیلد داده‌ای بخشی از سرآیند آی‌پی نیست و درون سرآیند آی‌پی کپسوله می‌شود.

تاریخ انتشار:
10 آبان 1398

نشانی منبع:
https://www.shabakeh-mag.com/networking-technology/16171/%D8%A2%D8%B4%D9%86%D8%A7%DB%8C%DB%8C-%D8%A8%D8%A7-%D9%BE%D8%B1%D9%88%D8%AA%DA%A9%D9%84%E2%80%8C%D9%87%D8%A7-%D9%88-%D9%85%D8%B3%DB%8C%D8%B1%DB%8C%D8%A7%D8%A8%DB%8C-%D8%B4%D8%A8%DA%A9%D9%87