بسمه تعالی

بررسی پروتکل UDP

استاد

دکتر امین فصحتی

نویسنده

سهند اسماعیل‌زاده

دانشگاه صنعتی شریف

پاییز ۱۴۰۳

فهرست

پروتکل UDP

مقدمه

پروتکل User Datagram Protocol (UDP) به‌منظور فراهم کردن یک حالت دیتاگرام برای ارتباطات رایانه‌ای مبتنی بر بسته در محیطی از شبکه‌های رایانه‌ای به‌هم‌پیوسته تعریف شده است. این پروتکل فرض می‌کند که پروتکل اینترنت (IP) به‌عنوان پروتکل زیربنایی استفاده می‌شود. UDP روشی را برای برنامه‌های کاربردی فراهم می‌کند تا پیام‌ها را با حداقل مکانیزم پروتکلی به برنامه‌های دیگر ارسال کنند. این پروتکل به صورت تراکنشی بوده و تحویل داده و جلوگیری از تکرار داده‌ها را تضمین نمی‌کند. برای درک بهتر این پروتکل ابتدا به بررسی لایه انتقال می‌پردازیم و سپس جزئیات این پروتکل را بررسی می‌کنیم.

لایه انتقال: هدف و وظایف

لایه انتقال (لایه ۴ مدل OSI) به‌عنوان پلی بین خدمات لایه کاربرد و عملکردهای شبکه‌ای لایه‌های پایین‌تر عمل می‌کند. این لایه مسئول اطمینان از رسیدن داده‌های ارسال‌شده از یک برنامه مبدأ به برنامه مقصد، به‌صورت قابل اعتماد و کارآمد است.

برخی از وظایف لایه انتقال

1. **بخش‌بندی و بازسازی (Segmentation and Reassembly):** این لایه پیام‌های بزرگ را به بخش‌های کوچکتر تقسیم کرده و در مقصد مجدداً آن‌ها را سرهم‌بندی می‌کند.
2. **ارتباط انتها به انتها (End-to-End Communication):** این لایه ارتباط منطقی بین فرآیندهای در حال اجرا روی میزبان‌های مختلف را فراهم می‌کند.
3. **چندپخشی و تفکیک (Multiplexing and Demultiplexing):** با اختصاص شناسه‌های منحصربه‌فرد **(پورت‌ها)**، امکان برقراری ارتباط همزمان چندین برنامه از طریق یک اتصال شبکه فراهم می‌شود.
4. **تشخیص و اصلاح خطا (Error Detection and Correction):** برخی از پروتکل‌های لایه انتقال (مانند **TCP**) دارای مکانیزم‌هایی برای تشخیص و اصلاح خطاهای انتقال هستند.
5. **کنترل جریان (Flow Control):** جلوگیری از ارسال بیش از حد داده توسط فرستنده به‌گونه‌ای که گیرنده قادر به پردازش آن باشد.

6. **مدیریت ارتباط و قابلیت اطمینان (Reliability and Connection Management):** پروتکل‌هایی مانند **TCP** ارتباطات را مدیریت کرده، دریافت داده‌ها را تأیید کرده و در صورت از دست رفتن بسته‌ها، مجدداً آن‌ها را ارسال می‌کنند.

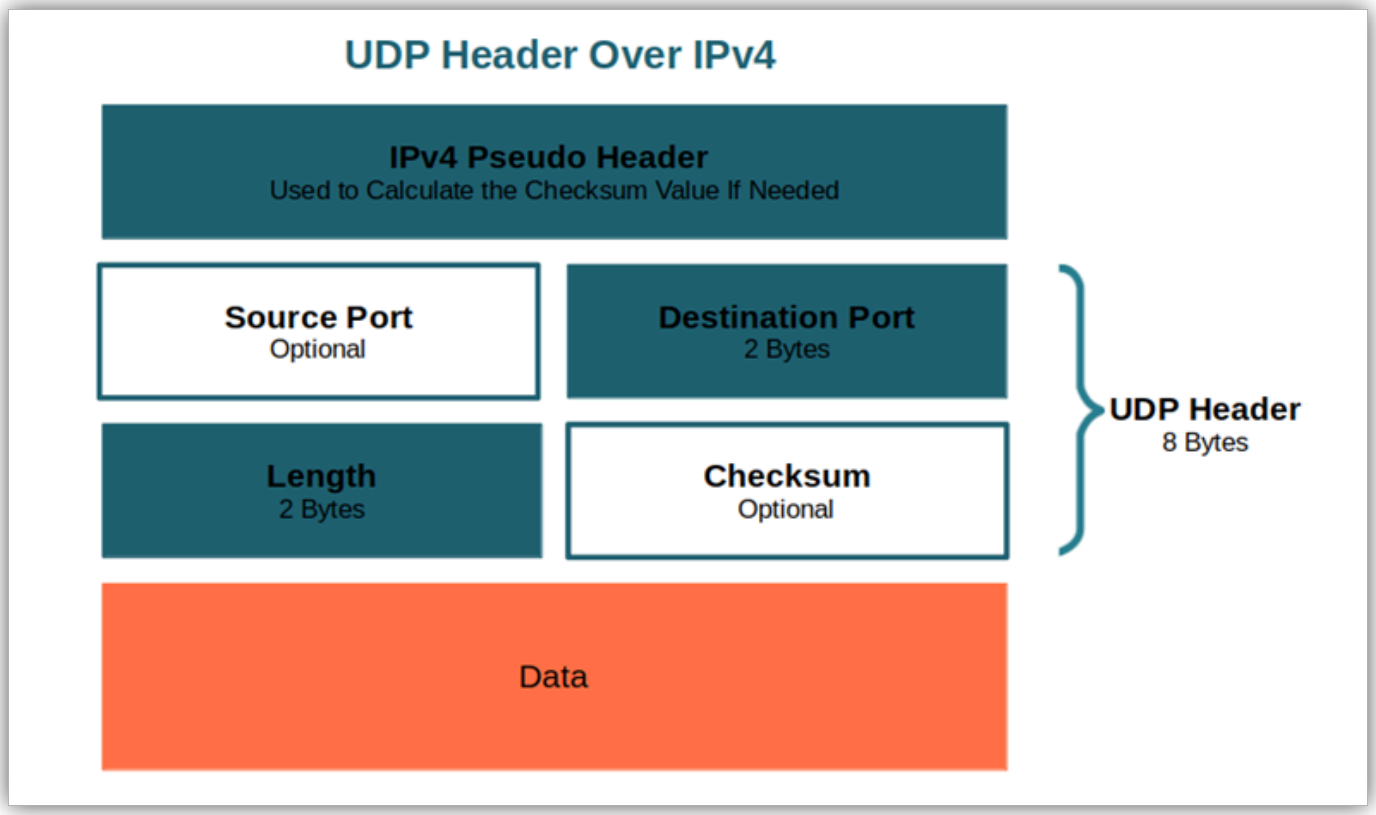
معرفی پروتکل UDP

UDP یکی از پروتکل‌های اصلی لایه انتقال در کنار TCP است. برخلاف TCP که بر قابلیت اطمینان تأکید دارد، UDP یک پروتکل بدون اتصال (Connectionless) و سبک‌وزن است که برای سرعت و کارایی طراحی شده است.

ویژگی‌های UDP

- **ارتباط بدون اتصال:** نیازی به برقراری ارتباط اولیه بین فرستنده و گیرنده ندارد. هر بسته (دیتاگرام) به‌طور مستقل ارسال می‌شود.
- **عدم تضمین تحویل:** UDP مکانیزم‌های تأیید دریافت، ارسال مجدد و اصلاح خطا را ارائه نمی‌دهد. در صورت از دست رفتن یک بسته، به‌طور خودکار مجدداً ارسال نمی‌شود.
- **بار اضافی کم (Low Overhead):** به دلیل عدم مدیریت اتصال و قابلیت اطمینان، UDP سربار کمی دارد و سریع‌تر از TCP عمل می‌کند.
- **تحویل به بهترین نحو (Best-Effort Delivery):** تضمینی برای رسیدن بسته‌ها به‌ترتیب یا بدون از دست رفتن وجود ندارد.
- **چک‌سام برای تشخیص خطا:** UDP شامل یک مقدار چک‌سام (Checksum) ساده برای تشخیص خطاها است، اما مکانیزم اصلاح خطا ندارد.
- پشتیبانی از پخش (Broadcast) و چندپخشی (Multicast): امکان ارسال داده به چند گیرنده به‌صورت همزمان را فراهم می‌کند.

ساختار هدر UDP



فیلدهای هدر UDP

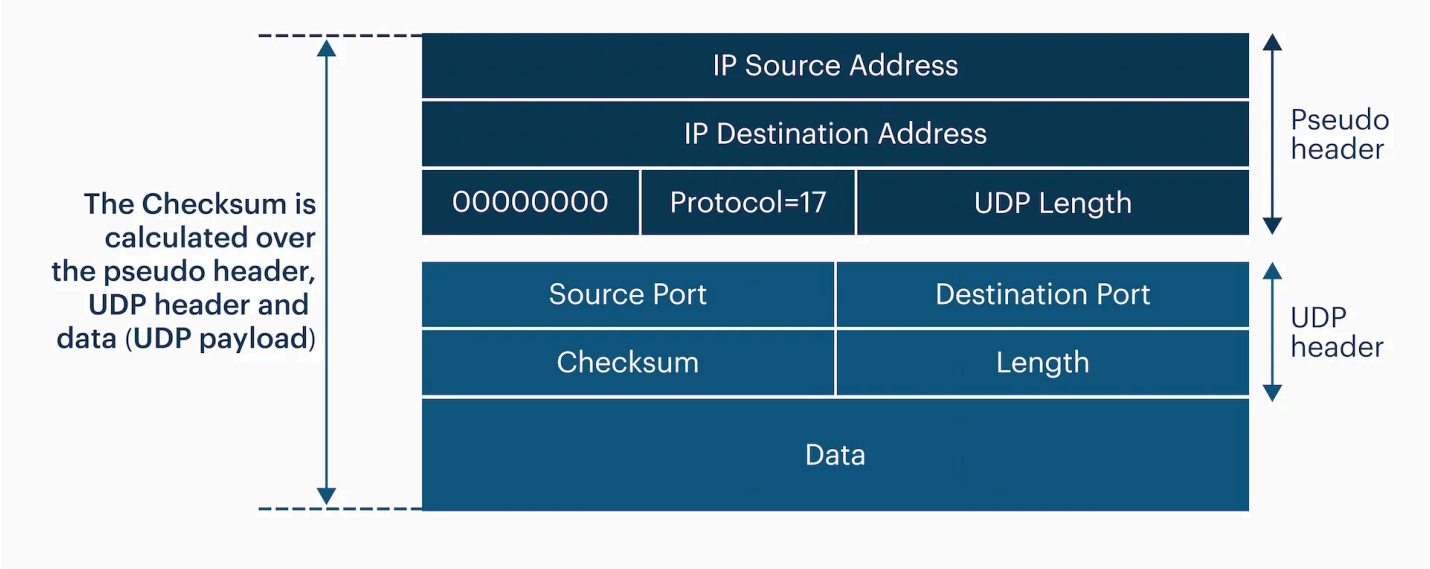
پورت مبدأ (Source Port): یک فیلد اختیاری است. در صورت استفاده، نشان‌دهنده پورت فرستنده است. اگر استفاده نشود، مقدار صفر در آن درج می‌شود.

پورت مقصد (Destination Port): نشان‌دهنده پورت مقصد برای بسته است.

طول (Length): اندازه کل دیتاگرام شامل هدر و داده را مشخص می‌کند. حداقل مقدار آن ۸ ماکسیمم آن ۶۵۵۳۵ بایت است.

چک‌سام (Checksum): برای تشخیص خطا در داده‌ها استفاده می‌شود. این مقدار از هدر UDP، داده و یک هدر کاذب (Pseudo Header) از IP محاسبه می‌شود.

سربرگ کاذب برای محاسبه چک‌سام



برای محاسبه Checksum، سربرگ کاذب که به‌صورت مفهومی قبل از سربرگ UDP قرار می‌گیرد، شامل آدرس مبدأ، آدرس مقصد، پروتکل و طول UDP است. این اطلاعات از دیتاگرام‌های ارسال‌شده به مسیر اشتباه محافظت می‌کند. محاسبه Checksum در UDP به این صورت انجام می‌شود که ابتدا داده‌های هدر UDP، داده‌های بخش کاربردی (Payload) و سربرگ کاذب (Pseudo-header) به بخش‌های ۱۶ بیتی تقسیم می‌شوند. سپس این بخش‌ها با استفاده از جمع مکمل یک (One's Complement Sum) با یکدیگر جمع می‌شوند. در صورت ایجاد حمل (Carry) از جمع، این مقدار به کم‌ارزش‌ترین بیت اضافه می‌شود. در نهایت، مکمل یک از نتیجه نهایی گرفته شده و مقدار به‌دست‌آمده به‌عنوان مقدار Checksum در سربرگ UDP ذخیره می‌شود. هنگام دریافت بسته، گیرنده همین محاسبات را روی کل بسته، شامل مقدار Checksum دریافت‌شده، انجام می‌دهد. اگر نتیجه همه یک (0xFFFF) باشد، بسته معتبر است، در غیر این صورت، خطا در انتقال رخ داده است.

اگر مقدار چکسام محاسبه‌شده صفر باشد، مقدار 0xFFFF ارسال می‌شود. مقدار ۰ به این معنی است که فرستنده چکسام را محاسبه نکرده است (برای اشکال‌زدایی یا برای پروتکل‌های سطح بالاتر که نیازی به آن ندارند).

چرا UDP را به جای TCP انتخاب کنیم؟

UDP در مواردی که سرعت و سربار کم از قابلیت اطمینان مهم‌تر است، استفاده می‌شود. در حالی که TCP دارای کنترل خطا، ارسال مجدد و کنترل جریان است، این ویژگی‌ها باعث افزایش تأخیر می‌شوند.

موارد استفاده از UDP

- **نیاز به تأخیر کم:** UDP در تماس‌های صوتی (VoIP)، پخش زنده و بازی‌های آنلاین که سرعت انتقال داده حیاتی است، استفاده می‌شود.
- **کاهش سربار:** ساختار سبک‌وزن UDP باعث کاهش مصرف پهنای باند و منابع پردازشی می‌شود.
- **پشتیبانی از پخش و چندپخشی:** برای IPTV و کشف دستگاه‌های شبکه مناسب است.
- **مدیریت قابلیت اطمینان در سطح نرم‌افزار:** پروتکل‌هایی مانند QUIC و RTP قابلیت اطمینان را در لایه برنامه مدیریت می‌کنند.
- **ارتباطات بدون وضعیت (Stateless Communication):** درخواست‌های DNS و نظارت SNMP نیازی به اتصال مداوم ندارند و از UDP استفاده می‌کنند.

پروتکل‌های مبتنی بر UDP و کاربردهای آن‌ها

پروتکل‌های زیر بر پایه UDP ساخته شده‌اند و هر کدام از آن‌ها ویژگی مورد نیاز کاربرد خود را به آن اضافه می‌کنند.

دسته‌بندی	پروتکل	کاربرد
شبکه و خدمات اینترنت	DNS	تبدیل نام دامنه به آدرس IP
	DHCP	تخصیص پویای آدرس IP
	SNMP	مدیریت و مانیتورینگ شبکه
	NTP	همگام‌سازی زمان در شبکه
ارتباطات بلادرنگ	VoIP & SIP	تماس‌های صوتی با تأخیر کم
	WebRTC	تماس ویدیویی و صوتی مبتنی بر مرورگر
پخش و رسانه	RTP & RTSP	پخش زنده و استریم محتوای چندرسانه‌ای
	Multicast IPTV	توزیع بهینه محتوای ویدیویی بر بستر شبکه
بازی‌های آنلاین و انتقال فایل	Game Servers	بازی‌های چندنفره بلادرنگ
	TFTP	پروتکل انتقال فایل سبک
VPN و امنیت	IKE	امنیت و تونل‌سازی در **IPSec VPNs**
	WireGuard	تونل‌سازی ایمن و پرسرعت
	QUIC	انتقال سریع داده‌های وب (HTTP/3)

چگونه پروتکل‌ها قابلیت اطمینان را به UDP اضافه می‌کنند؟

برای افزایش قابلیت اطمینان UDP، پروتکل‌هایی که بر پایه UDP ساخته می‌شوند، از مکانیزم‌هایی مانند تأیید دریافت (ACK)، ارسال مجدد بسته‌های گمشده، شماره‌گذاری بسته‌ها برای حفظ ترتیب، و کنترل ازدحام استفاده می‌کنند. این ویژگی‌ها باعث می‌شوند که ارتباطات مبتنی بر UDP، که ذاتاً غیرقابل اطمینان هستند، بتوانند داده‌ها را به شکلی مطمئن‌تر و پایدارتر انتقال دهند. به این ترتیب، می‌توان از سرعت و کارایی UDP بهره برد، در حالی که مشکلاتی مانند از بین رفتن بسته‌ها و ترتیب نامنظم آن‌ها کاهش می‌یابد.

به‌عنوان مثال، QUIC که توسط گوگل توسعه داده شده و در HTTP/3 مورد استفاده قرار می‌گیرد، یک پروتکل انتقال مبتنی بر UDP است که قابلیت اطمینان را با استفاده از تأیید انتخابی (SACK)، ارسال مجدد هوشمند بسته‌ها، چندجریانی، و رمزنگاری پیش‌فرض تأمین می‌کند. برخلاف TCP، که تأخیر بیشتری در برقراری اتصال اولیه دارد، QUIC از RTT-0 برای کاهش زمان تأخیر استفاده می‌کند. این ویژگی‌ها باعث شده‌اند که QUIC در کاربردهایی مانند مرور وب، پخش ویدیو، و بازی‌های آنلاین عملکرد بهتری نسبت به TCP داشته باشد.

نتیجه‌گیری

UDP یک پروتکل کلیدی در لایه انتقال است که سرعت و کارایی را بر قابلیت اطمینان ترجیح می‌دهد. درک نقاط قوت و محدودیت‌های UDP به مهندسان شبکه و توسعه‌دهندگان کمک می‌کند تا پروتکل مناسب را برای نیازهای خاص خود انتخاب کنند و تعادل بین سرعت، قابلیت اطمینان و استفاده از منابع را برقرار کنند.