

## گزارش تحقیق درباره‌ی رابطه‌ی میان DNS و پروتوکلهای TCP و UDP

تهیه و تنظیم: مبین خیبری

شماره دانشجویی: 994421017

استاد راهنما: دکتر تاجبخش

### چکیده:

در این گزارش قصد داریم به رابطه‌ی میان پروتوکلهای UDP و TCP با سرویس DNS بپردازیم. می‌دانیم که پروتکل DNS بطور پیش فرض از UDP برای ارسال درخواست و گرفتن جواب استفاده می‌کند. با توجه به این مسئله می‌خواهیم به دو پرسش زیر پاسخ بدهیم:

1. چرا از UDP استفاده می‌کند؟ دلیل خاصی دارد؟
2. از TCP می‌توان برای DNS Resolve استفاده کرد؟ اگر بله، کجاها و چرا؟

در این بخش ابتدا به معرفی مفاهیم TCP، UDP و DNS پرداخته و سپس به مرور به پرسش‌های بالا پاسخ می‌دهیم.

### پروتکل UDP چیست؟

UDP، سرواژه عبارت User Datagram Protocol، یک پروتکل ارتباطی است که برای کاربردهای حساس به زمان مانند VoD، پخش زنده و جست‌وجوی DNS در اینترنت استفاده می‌شود. UDP همانند TCP و سایر پروتکل‌های ارتباطی، پیام‌های برنامه‌های کاربردی را دریافت کرده و به تعداد بسته می‌شکند و سپس آن بسته‌ها را در شبکه به سمت مقصد ارسال می‌کند.

این پروتکل که در کنار پروتکل TCP از پرکاربردترین پروتکل‌های انتقال در بستر اینترنت به حساب می‌آید، به خاطر تشکیل ندادن اتصال قبل از انتقال داده سرعت ارتباطات را بسیار افزایش می‌دهد. همین سرعت بالای انتقال دلیل استفاده از UDP برای مصرف‌های حساس به زمان شده است. البته باید در نظر داشت که تشکیل ندادن اتصال به منظور افزایش سرعت انتقال داده‌ها، باعث خواهد شد که بسته‌ها در حین انتقال گم شوند و کیفیت تحت تاثیر این اتفاق قرار گیرد.

### کاربردهای UDP

UDP برای کاربردهایی استفاده می‌شود که در آن از دست دادن بسته‌های یا به هم خوردن ترتیب آن‌ها اهمیت کم‌تری نسبت به صبر کردن برای رسیدن بسته‌ها دارد. به عنوان مثال برای ارسال صوت و فیلم آنلاین از این پروتکل استفاده می‌شود، چرا که از یک سو این کاربردها حساس به زمان هستند و نیاز به انتقال سریع داده‌ها بسیار پررنگ است و از سوی دیگر در طراحی آن‌ها قابلیت تحمل از دست دادن

داده‌ها در نظر گرفته شده است. یک نمونه‌ی کاربردی دیگر از استفاده از پروتکل UDP، سیستم Voice over IP یا VoIP است. VoIP مبنای کار بسیاری از سیستم‌های تلفنی بر پایه‌ی اینترنت است که در آن، یک تماس تلفنی کم کیفیت ولی بدون تاخیر نسبت به یک تماس بسیار باکیفیت ولی با تاخیر زیاد مناسب‌تر است. به دلایلی مشابه، برای بازی‌های آنلاین نیز استفاده از UDP گزینه مناسبی است.

### تفاوت پروتکل UDP و TCP

UDP یک روش استاندارد انتقال داده بین دو دستگاه در شبکه است. این پروتکل مکانیزم انتقال را بسیار ساده می‌کند؛ چرا که بدون ایجاد اتصال و فرایندی زمان‌گیر مانند Handshake در TCP، انتقال اطلاعات آغاز می‌شود. از سوی دیگر، در UDP الزامی برای حفظ ترتیب بسته‌ها و بررسی صحت بسته‌های دریافت شده وجود ندارد. این موارد در کنار هم باعث می‌شوند تا انتقال یک فایل یکسان در UDP نسبت به TCP با سرعت بیشتری انجام شود.

UDP یک روش استاندارد انتقال داده بین دو دستگاه در شبکه است. این پروتکل مکانیزم انتقال را بسیار ساده می‌کند؛ چرا که بدون ایجاد اتصال و فرایندی زمان‌گیر مانند Handshake در TCP، انتقال اطلاعات آغاز می‌شود. از سوی دیگر، در UDP الزامی برای حفظ ترتیب بسته‌ها و بررسی صحت بسته‌های دریافت شده وجود ندارد. این موارد در کنار هم باعث می‌شوند تا انتقال یک فایل یکسان در UDP نسبت به TCP با سرعت بیشتری انجام شود.

### پروتکل TCP چیست؟

پروتکل TCP، سرواژه‌ی عبارت Transmission Control Protocol به معنی پروتکل کنترل انتقال، یک پروتکل ارتباطی دوطرفه است که انتقال پیام بین تجهیزات در یک شبکه را ممکن می‌کند. TCP مهم‌ترین و پرکاربردترین پروتکل ارتباطی شبکه به حساب می‌آید. این پروتکل پیام‌ها را از برنامه‌های کاربردی دریافت می‌کند و آن‌ها را به بسته‌های جداگانه‌ای می‌شکند که می‌توانند در شبکه به‌وسیله‌ی سوئیچ‌ها و روترها جابجا شوند و به مقصد برسند. TCP این بسته‌ها را شماره‌گذاری می‌کند تا در مقصد بتواند ترتیب درست آن‌ها را تشخیص دهد و با کنارهم گذاشتن محتوای بسته‌ها، پیام را به شکل کامل به برنامه‌ی کاربردی در دستگاه مقصد تحویل دهد.

### پروتکل TCP چگونه کار می‌کند؟

تصور کنید می‌خواهیم از سرور ایمیل یک ایمیل را ارسال کنیم. پروتکل TCP پیاده‌سازی شده در سرور، آن ایمیل را به تعدادی بسته تقسیم می‌کند، آن‌ها را شماره‌گذاری می‌کند و به پروتکل IP تحویل می‌دهد تا پروتکل IP آن‌ها را به دست مقصد مورد نظر برساند. در مقصد نیز پروتکل IP این بسته‌ها را تحویل پروتکل TCP می‌دهد و در نهایت پروتکل TCP آن‌ها را به هم متصل می‌کند و به برنامه‌ی ایمیل می‌رساند. این بسته‌های ارسالی گرچه به مقصدی یکسان ارسال می‌شوند ولی ممکن است با توجه به وضعیت شبکه، اختلالات و کندی‌ها مسیرهای متفاوتی را طی کنند و با ترتیب متفاوتی به مقصد برسند. پروتکل TCP وظیفه دارد بسته‌ها را با ترتیبی صحیح به برنامه‌ی کاربردی در مقصد ارایه دهد.

TCP یک پروتکل مبتنی بر اتصال است به این معنا که تا زمان اتمام تبادل پیام بین دو دستگاه (که ممکن است شامل تعداد زیادی بسته باشد) اتصال بین آنها باید برقرار بماند و بسته‌های مربوطه در همان اتصال جابجا شوند. درست برخلاف پروتکل IP که در آن هر واحد داده به شکل مستقل آدرس دهی می‌شود و از دستگاه مبدا به مقصد می‌رسد. به این ترتیب، وجود پروتکلی مانند TCP ضرورت پیدا می‌کند تا بتوانیم ترتیب پیام‌ها را حفظ کنیم.

## Handshaking در TCP

برای ایجاد اتصال در پروتکل TCP یک Handshake سه مرحله‌ای (Three-way Handshaking) انجام می‌شود. ابتدا مبدا، یک پیام SYN به مقصد ارسال می‌کند تا مکالمه شروع شود. سپس مقصد پیام SYN/ACK را به مبدا ارسال می‌کند تا موافقت خود را برای ایجاد مکالمه ابراز کند و در نهایت نیز مبدا یک پیام ACK به مقصد می‌فرستد تا پس از آن فرایند انتقال پیام آغاز شود.

برای اتمام اتصال نیز یک Handshake چهار مرحله‌ای (Four-way Handshaking) لازم است. هر کدام از طرفین مکالمه می‌تواند آغازگر فرایند اتمام اتصال باشند. به این ترتیب، طرف آغازکنندهی فرایند، یک پیام FIN به طرف دیگر ارسال می‌کند و طرف دیگر نیز با ACK به آن پاسخ می‌دهد و همچنین یک پیام FIN دیگر را به آن طرف اول می‌فرستد. در ادامه طرف اول با ACK به طرف مقابل پاسخ می‌دهد و پس از مدت زمانی مشخص، اتصال را قطع می‌کند. طرف مقابل نیز با دریافت ACK، اتصال را از سمت خود خاتمه می‌دهد.

## تفاوت‌های دیگر پروتکل TCP با پروتکل UDP

هر دو پروتکل TCP و UDP وظیفه‌ای مشابه، یعنی انتقال Packetها را برعهده دارند. هرچند تفاوت‌هایی میان این دو پروتکل وجود دارد که باعث می‌شود در هر نوع استفاده‌ای، یکی از این دو پروتکل انتقال مورد استفاده قرار گیرد. پروتکل TCP از نظرهای زیر با UDP متفاوت است:

- حفظ ترتیب بسته‌ها: پروتکل TCP در مقصد، ترتیب بسته‌ها را اصلاح کرده و آنها را همانند ترتیب ارسالی قرار می‌دهد. در پروتکل UDP تضمینی برای حفظ ترتیب بسته‌ها وجود ندارد.
- ارسال مجدد بسته‌های گم شده: در مسیر ارتباطی ممکن است تعدادی از پیام‌ها گم شوند TCP. گم شدن بسته‌ها را تشخیص می‌دهد و مجدداً آنها را ارسال می‌کند. در حالی که پروتکل UDP از بسته‌های گم شده صرف نظر می‌کند.
- حفظ صحت پیام: پروتکل TCP با استفاده از روش‌های خاصی وجود خطا در بسته‌ی دریافت شده را شناسایی و آنها را مجدداً ارسال می‌کند.

به شکل کلی استفاده از پروتکل UDP در مواردی توصیه می‌شود که سرعت ارسال اهمیت بالایی داشته باشد ولی ترتیب و صحت ارسال داده‌ها از اهمیت کمتری برخوردار باشد. موارد استفاده‌ی TCP بیش‌تر به کاربردهایی برمی‌گردد که اولویت اصلی، سرعت ارسال و دریافت نباشد.

## DNS چیست؟

DNS مانند یک دفترچه تلفن برای اینترنت است. همانطور که شما برای تماس با دیگران به جای بخاطر سپردن شماره‌ی آن‌ها، از دفترچه تلفن استفاده می‌کنید، DNS نیز مانند یک دفترچه تلفن عمل می‌کند و نیازی به حفظ کردن آدرس IP ها نیست. همانطور که می‌دانید، کامپیوترها برای اتصال به یکدیگر از اعداد یا همان IP آدرس‌ها استفاده می‌کنند.

Domain Name System فهرست توزیع شده‌ای است که نام دامنه قابل خواندن توسط انسان مانند [www.respina.net](http://www.respina.net) را به اعداد خوانا برای کامپیوترها یعنی IP آدرس تبدیل می‌کند. برعکس این نیز در مورد DNS صدق می‌کند، یعنی DNS سیستمی است که نام دامنه وب را سازماندهی می‌کند و آن‌ها را برای همه کسانی که می‌خواهند به شبکه وصل شوند، قابل فهم‌تر می‌کند.

## DNS چگونه کار می‌کند؟

هنگامی که از سایتی بازدید می‌کنید، کامپیوتر شما یک سری مراحل را برای تبدیل آدرس وب قابل خواندن انسان به یک آدرس IP قابل خواندن ماشین دنبال می‌کند. این اتفاق هر بار که از یک نام دامنه استفاده می‌کنید، چه در حال مشاهده وبسایتی باشید، چه در حال ارسال ایمیل و یا گوش دادن به ایستگاه‌های رادیویی اینترنتی باشید، رخ می‌دهد.

هر سایت نامگذاری شده‌ای می‌تواند با بیش از یک آدرس IP مطابقت داشته باشد. در حقیقت، برخی سایت‌ها صدها یا بیشتر آدرس IP دارند که با یک نام دامنه واحد مطابقت دارند. در نتیجه به سیستم DNS نیاز است تا آدرس‌های IP را به نام دامنه قابل خواندن افراد تبدیل کند، چرا که به خاطر سپردن تعداد زیادی عدد دشوارتر از یک نام دامنه ثابت است.

اگر فقط یک دایرکتوری برای سایت وجود داشته باشد، آن‌گاه وقتی تعداد درخواست‌ها برای بازدید از یک سایت زیاد شود، مدت زمان زیادی طول می‌کشد تا به درخواست شما پاسخی داده شود. در عوض، اطلاعات DNS در سرورهای زیادی به اشتراک گذاشته می‌شود، اما به صورت محلی نیز در کامپیوتر مشتریان ذخیره می‌شود. این احتمال وجود دارد که شما چندین بار در روز از یک سایت بازدید کنید. با ذخیره شدن در کش دیگر نیازی به هر بار حل و فصل کردن نام دامنه با آدرس IP نیست. در نتیجه تعداد دفعاتی که لازم است از DNS استفاده شود، کمتر از تعداد دفعاتی است که شما یک سایت را در مرورگر جستجو می‌کنید.

DNS از یک پایگاه داده سلسله مراتبی استفاده می‌کند که حاوی اطلاعاتی در مورد نام دامنه است. فرض کنید شما در مرورگر خود نام دامنه سایتی را وارد می‌کنید. اولین کاری که کامپیوتر شما انجام خواهد داد، ارسال درخواست به سرور DNS محلی سیستم عامل است تا بررسی کند که آیا پاسخ مورد نیاز شما در حافظه نهان (Cache) کامپیوتر ذخیره شده است یا خیر. اگر در حافظه پنهان یافت نشد، درخواست شما از طریق اینترنت به یک یا چند سرور دی ان اس ارسال می‌شود که به‌طور کلی توسط ارائه‌دهنده خدمات اینترنت شما با آن‌ها ارتباط برقرار می‌شود. اگر اطلاعات لازم در این سرورهای DNS یافت نشود، درخواست به سرورهای خارجی دیگر ارسال می‌شود.

## مزایای DNS چیست؟

اصلی‌ترین مزیت سیستم DNS این است که استفاده از اینترنت را بسیار تسهیل می‌کند. در صورتی که برای بازدید از سایت‌ها لازم بود که تمام آدرس‌های IP که می‌خواستیم به آن‌ها دسترسی داشته باشیم را حفظ باشیم، بسیار سنگین و دشوار می‌شد. با استفاده از آن دیگر نیازی به حفظ کردن این رشته اعداد نیست و برای دسته‌بندی، بایگانی و کمک به موتورهای جستجو مناسب است.

یکی دیگر از مزیت‌های قابل توجه ثبات آن است. به دلایل مختلف، ممکن است آدرس‌های IP تغییر کنند، بنابراین اگر می‌خواهید به یک وبسایت دسترسی پیدا کنید، نه تنها باید آدرس IP آن را بدانید بلکه این اطلاعات نیز باید به روز باشد. سیستم DNS وظیفه دارد تا آدرس‌های IP را به روشی بسیار سریع و ثابت، به روز کند و دسترسی ما به وبسایت‌ها را آسان کند.

DNS می‌تواند امنیت زیرساخت را ارتقا بخشد، همچنین می‌تواند به روزرسانی‌های ایمن پویا را فراهم کند. قابل اطمینان‌تر است و می‌تواند پیام‌ها را با خرابی صفر به کاربران تحویل دهد. این سیستم شما را قادر می‌سازد تا عملکرد فنی سرویس دیتابیس را مشخص کنید. همچنین می‌تواند پروتکل DNS، مشخصات دقیق ساختار داده‌ها و مبادلات ارتباطی داده مورد استفاده در DNS را تعریف کند. در واقع DNS به عنوان نوعی توازن بار یا یک لایه اضافی امنیتی استفاده می‌شود.

## معایب DNS چیست؟

در کنار تمام مزیت‌ها و کاربردهای DNS، معایبی نیز برای آن وجود دارد. یکی از اصلی‌ترین معایب آن DNS Attacks است که در آن مهاجم آدرس واقعی را با یک آدرس جعلی به منظور کلاهبرداری جایگزین می‌کند و با فریب کاربران آن‌ها را بدون اطلاع به آدرس‌های مخرب هدایت می‌کند. معمولاً هدف از این کار گرفتن اطلاعات بانکی یا سایر داده‌های مهم و حساس کاربران است.

اگر بدافزار تنظیمات سرور DNS شما را تغییر داده باشد، با وارد کردن URL ممکن است شما را به یک وبسایت کاملاً متفاوت یا به وبسایتی که به نظر می‌رسد مانند وبسایت بانکی شما باشد منتقل کند. ممکن است نام کاربری و رمز عبور شما را ضبط کند و اطلاعاتی که برای دسترسی به حساب بانکی شما مورد نیاز باشد را به دست افراد سوءاستفاده‌گر برساند.

بدافزارها برخی از سرورهای DNS را می‌ربایند تا شما را از وبسایت‌های محبوب و پربازدید به وبسایت‌های ویروسی جعلی و پر از تبلیغات هدایت کنند و این دیدگاه غلط را به وجود می‌آورند که برای حذف ویروس‌ها از کامپیوتر خود، لازم است برنامه‌هایی که در واقع مخرب و ویروسی هستند را دانلود و نصب کنید.

برای جلوگیری از چنین مشکلاتی، لازم است که برنامه‌های آنتی ویروس معتبر را بر روی سیستم خود نصب کنید و از ورود به سایت‌هایی که ظاهر متفاوتی با وبسایت درخواستی شما دارند پرهیز کنید. همچنین از وارد کردن اطلاعات شخصی و بانکی خود در سایت‌های نامعتبر خودداری کنید.

## DNS با کدامیک کار میکند UDP یا TCP ؟!

یکی از متداولترین سوالاتی که اکثر افراد از خودشان می‌پرسند، این است که DNS با کدامیک از پروتکل‌ها کار می‌کند TCP یا UDP؟

ممکن است وقتی جواب این سوال را ندانید، از روی حدس و گمان بگویید هیچکدام، که اشتباه است. زیرا از هر دو استفاده می‌کند هم TCP و هم UDP.

هر دو پروتکل کاملاً با هم متفاوت هستند، TCP پروتکل اتصال گرا (امن) است و UDP بدون اتصال است.

DNS از TCP برای انتقال Zone فایل‌ها بر روی پورت ۵۳ استفاده می‌کند:

DNS از معماری master/slave استفاده می‌کند، یکی از آنها name server اصلی هست که تمام داده‌ها در آن وجود دارد و مابقی، داده‌های تکراری است که از name server اصلی منتقل شده.

از آنجا که هیچ تناقضی بین zone فایل‌ها نمی‌تواند وجود داشته باشد، بنابراین برای انتقال این zone فایل‌ها، DNS از TCP برای برقراری اتصال استفاده می‌کند، که این اطمینان را حاصل می‌کند که zone فایل‌ها بصورت کاملاً امن منتقل شده‌اند.

DNS برای جواب دادن به query ها از UDP بر روی پورت ۵۳ استفاده می‌کند:

DNS از UDP برای کاربردهای معمولی نظیر پاسخ‌گویی به query کلاینت‌ها استفاده می‌کند. زمانیکه یک کلاینت از DNS server یک نام به IP و یا یک IP به نام را می‌پرسد، سپس DNS از پروتکل UDP برای پاسخ‌گویی به آن query استفاده می‌کند. دلیلی که از پروتکل UDP برای این مقصود استفاده می‌شود اینست که UDP اتصال گرا نیست، بنابراین سریع و سبک عمل می‌کند و نتیجه query را به سرعت به کلاینت ارسال می‌کند و در مقایسه با TCP زمان کمتری را صرف این کار می‌کند.

البته در صورت نیاز DNS می‌تواند برای پاسخ‌گویی به query ها از TCP هم استفاده کند، اما معمولاً استفاده از UDP بخاطر سرعت بالای آن ارجحیت دارد.

## چرا DNS روی هر دو پروتکل TCP و UDP کار می‌کند؟

همه ما از اهمیت DNS و DNS Server در شبکه آگاه هستیم. سرویسی که اسم دستگاه‌ها را به آدرس IP و برعکس، تبدیل می‌کند. زمانی که به دنبال کامپیوتر یا دستگاه خاصی در شبکه می‌گردیم، این DNS است که کار پیدا کردن دستگاه را برای ما انجام می‌دهد.

وقتی در اینترنت آدرس [www.ITPro.ir](http://www.ITPro.ir) را وارد می‌کنیم، پشت پرده پردازش‌هایی انجام می‌شود که DNS عهده دار آنها است. ولی آیا تا به حال توجه کرده ایم که DNS هم با پروتکل TCP کار می‌کند و هم با پروتکل UDP؟ چطور این امر ممکن است؟ چرا DNS با هر دو پروتکل TCP و UDP کار می‌کند؟ برای فهمیدن پاسخ این پرسش، لطفاً تا پایان این مقاله با من همراه باشید.

همانطور که میدانید، TCP یک پروتکل اتصال گرا می باشد. TCP الزام می کند که داده در مقصد، سالم و پایدار باشد. برای چک کردن درستی و سالم بودن داده، TCP با میزبان (مبدأ) ارتباط برقرار می کند. این در حالی است که UDP، یک پروتکل غیر اتصال گرا بوده و مستلزم پایداری داده در مقصد نیست و هیچ گونه ارتباطی با میزبان برای چک کردن درستی داده، برقرار نمی کند. بسته های UDP از نظر اندازه، کوچک تر از بسته های TCP هستند.

هر بسته UDP حداکثر 512 بایت حجم دارد. بنابراین هر نرم افزاری که بخواهد حجم داده بالاتری از 512 بایت منتقل کند، از TCP استفاده می کند. از UDP برای انتقال اطلاعات کوچک استفاده می شود، در حالی که می بایست از TCP برای انتقال اطلاعات بیشتر از 512 بایت استفاده شود. UDP به دلیل حجم کمتر خود و اینکه دارای مکانیزمی برای چک کردن درستی داده نیست، سریعتر از TCP می باشد.

### **DNS از پروتکل TCP روی پورت 53 برای انجام Zone Transfer استفاده می کند.**

DNS از ساختار Master & Slave (یا ارباب و برده) استفاده می کند که در این ساختار، شما یک Name Server معتبر و مورد تأیید (Authoritative) دارید که همان Master بوده و تمامی ورودی ها را در اختیار دارد و بقیه سرور ها که همان Slave ها هستند

با استفاده از Replication (یا یکسان سازی اطلاعات)، فایل های Zone ها را از این سرور معتبر دریافت کرده و به DNS Query ها، رسیدگی می کنند. از آنجایی که می بایست فایل های Zone ها همیشه پایدار و سالم باشند، DNS از TCP به عنوان پروتکل ارتباطی برای انتقال فایل های Zone ها استفاده می کند که از این طریق، از صحت انتقال فایل ها اطمینان حاصل می کند.

### **DNS از پروتکل UDP روی پورت 53 برای انجام DNS Query استفاده می کند.**

DNS از پروتکل UDP برای انجام کارهای اولیه، مانند جواب دادن به پرسش های کلاینت ها، استفاده می کند. زمانی که یک کلاینت از DNS برای تبدیل اسم به IP و یا IP به اسم سوال می پرسد، DNS از پروتکل UDP برای جواب دادن به کلاینت استفاده می کند.

دلیل استفاده از UDP این است که UDP یک پروتکل غیر اتصال گرا بوده و بنابراین، سبک و سریع می باشد که باعث می شود ارسال جواب به کلاینت در زمان به مراتب کوتاه تری نسبت به TCP، انجام شود. اگر کلاینت پاسخی از DNS دریافت نکرد، پس از طی دوره زمانی 3 تا 5 ثانیه می بایست درخواست خود را این بار از طریق TCP مجدداً ارسال کند.

در عین حال، DNS می تواند در صورت نیاز روی TCP نیز کار کند تا بتواند به پرسش های کلاینت ها، جواب دهد. اما همیشه UDP به خاطر سرعت بیشترش، مقدم در نظر گرفته می شود. نکته ای که باید در نظر گرفته شود این است که جواب های DNS (DNS Answers) که حجمشان بیشتر از 512 بایت است، از طریق TCP ارسال می شوند.

(منظور جواب هایی است که DNS Server به کلاینت ها ارسال می کند) اگر DNS Server پرسشی دریافت کند که جواب آن بیشتر از 512 بایت باشد، DNS Server از کلاینت درخواست کننده می

خواهد که درخواست خود را از طریق TCP ارسال کند. این امر به خاطر محدودیت های حجمی و کیفی پروتکل UDP می باشد.

نکته : زمانی که حجم بسته UDP از 512 بایت فراتر رود، بسته بریده می شود ( کوتاه می شود که به این امر، Truncate اطلاق می شود). لازم به ذکر است که زمانی که بسته کوتاه می شود، یک Truncated Bit در Header این بسته قرار می گیرد که مشخص می کند بسته بریده شده است. زمانی که کلاینت DNS بسته ای با عنوان Truncated Bit دریافت می کند، متوجه می شود که داده از 512 بایت بیشتر است و دریافت بسته با پروتکل UDP امکان پذیر نیست. بنابراین کلاینت به TCP روی می آورد و همان درخواست را از طریق TCP ارسال می کند.

پاورقی: EDNS یا مکانیزم توسعه برای DNS (Extension Mechanism for DNS) اجازه می دهد که جواب هایی که حجمشان بیشتر از 512 بایت است، برای ارسال توسط پروتکل UDP مجددا بسته بندی شوند. لازم به ذکر است که EDNS قادر به ارسال بسته ها با حجم بیشتر از 512 بایت روی پروتکل UDP می باشد.

DNS یک پروتکل در لایه 7 یا لایه Application است و همانطور که می دانید این لایه از مدل مرجع OSI، لایه ای است که پروتکل های کاربردی نظیر HTTP و FTP و حتی همین DNS در این لایه قرار دارند. در دوره نتورک پلاس و در لینک بالا، در مورد TCP و UDP صحبت کردیم و دیدیم که TCP قابل اعتماد است و در ازای هر بسته اطلاعاتی که ارسال می کند یک تاییدیه برای سورس هم می فرستد و نتیجتا سرعت آن کندتر از UDP می باشد. در مقابل UDP غیرقابل اعتماد است و تاییدیه ای بابت ارسال بسته های اطلاعاتی برای سورس ارسال نمی کند که اطمینان حاصل شود بسته اطلاعاتی صحیح سالم و بدون کم و کاستی به مقصد رسیده است ولی سرعت آن بالاتر است. حال چرا باید DNS که از Component های مهم شبکه است، ترافیک آن از نوع UDP باشد؟

حقایق جالبی در مورد TCP و UDP در لایه transport وجود دارد که بیایید با هم بدان نگاهی بیندازیم. UDP همانطور که گفته شد، سریع تر از TCP عمل می کند و ترافیک DNS از آنجایی که از UDP استفاده می کند، فاکتوری بسیار مهم است. سرورهای DNS به دلیل همین UDP بودن ترافیک، ارتباط یا Connection را حفظ نمی کنند و Connection less هستند.

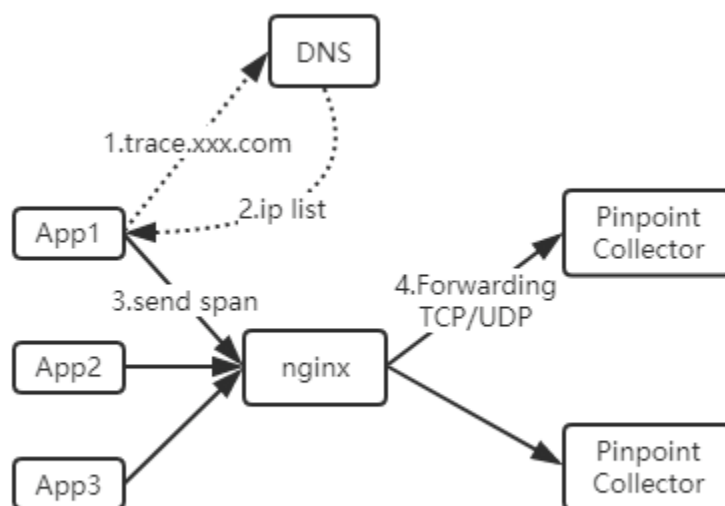
درخواست های DNS به طور کلی بسیار کوچک و مناسب در بخش های UDP است ( UDP segments).

UDP قابل اعتماد نیست، اما قابلیت اطمینان در application layer می تواند اضافه شود. یک برنامه می تواند از UDP استفاده کند و با استفاده از زمانبندی یا همان Timeout قابلیت اطمینان را بدست بیاورد و در لایه application layer مجددا ارسال شود.



اجازه دهید کمی بهتر و ساده تر در این مورد صحبت کنیم و برای درک بهتر موضوع از یک مثال استفاده خواهیم کرد. فرض کنید برای دریافت شماره تماس شرکت، سازمان، بیمارستان یا خلاصه هر جای دیگری با 118 تماس می گیرید، خب، آیا پس از برقراری ارتباط با اپراتور و درخواست شماره تماس، تا ابد منتظر پاسخ از طرف اپراتور می مانید، قطعاً که خیر، اگر در دریافت پاسخ معطل شوید، تماس را قطع و مجدداً تماس با 118 می گیرید. در حقیقت هدف شما از برقراری تماس با 118، دریافت شماره آن جایی است که با آن کار دارید نه خود 118.

در شبکه هم به همین صورت است، وقتی درخواستی به سمت DNS سرور می رود، DNS باید در 2 ثانیه این درخواست را Respond کند وگرنه درخواست از بین می رود. کلاینتهایی که به منظور name resolution به سرخ dns می روند، تا ابد که منتظر دریافت پاسخ از سوی DNS نخواهند ماند و چنانچه درخواست بی جواب ماند، دوباره درخواست را برای dns ارسال می کنند و تا 2 ثانیه منتظر جواب می مانند. هدف کلاینتها از برقراری ارتباط با DNS در واقع خود dns نیست بلکه دریافت hostname و یا ip address سیستمی است که قرار است با آن ارتباط گرفته و به نحوی نقل و انتقال اطلاعات داشته باشند.



### دستور Resolve-DnsName در پاورشل

پاورشل دستورات مختلفی را به خود اختصاص داده است که این دستورها رده های سختی متفاوتی را دارند. بعضی از آن ها آسان تر و بعضی دیگر سخت تر و پیچیدگی های خاص خودشان را دارند. یکی از این دستورها Resolve-DnsName می باشد که قبل از پرداختن به آموزش آن بهتر است با Nslookup آشنایی پیدا کنید. Nslookup یک ابزار خط فرمان است که در رزولیشنی تقریباً با dns یکسان است.

Resolve-DnsName نسخه ای می باشد که مدرن تر است Nslookup می باشد. این دستور به منظور جستجوی نام میزبان استفاده می شود که در ادامه به این دستور خواهیم پرداخت. با استفاده از Resolve-DnsName می توان به انجام کارهایی همچون زیر پرداخت.

### پرس و جو استاندارد

اگر می خواهید یک پرس و جو استاندارد را در محیط پاورشل داشته باشید، می توان نام میزبان مورد نظر خود را مشخص کنید و سپس اقدام به اجرای دستور زیر کنید. لازم نیست یک جدول قالب را مشخص کنید اما بودن آن نیز اطلاعات مفیدی را در اختیار شما قرار بدهد. برای این کار می توانید دستور زیر را اجرا کنید:



```
PS C:\> Resolve-DnsName sid-500.com | Format-Table -AutoSize
```

Name	Type	TTL	Section	IPAddress
sid-500.com	A	300	Answer	192.0.78.24
sid-500.com	A	300	Answer	192.0.78.25

Resolve-DnsName parspack.com | Format-Table -AutoSize

### پرس و جو بدون پرونده میزبان

در ابتدا به چیزی که باید توجه کنید این است که میزبان شما پرونده میزبان و حافظه نهان DNS را خواستار می باشد. اگر فایل میزبان و حافظه نهان شما به صورت عدم بازگشت در آمد، شما با استفاده از سرور DNS می توانید آن را انتقال دهید. برای اینکه از این کار جلوگیری کنید، ضروری است که اسکریپت Resolve-DnsName را با استفاده از پارامتر -NoHostFile اجرا کنید.

Resolve-DnsName parspack.com -NoHostsFile

### پرس و جو تنها در حالت نهان

برای اینکه این دستور را به شما نشان دهیم، Dns Client Cache را پاک می کنیم و سپس با استفاده از parspack.com پرس و جوهای لازم را انجام می دهیم. در صورت انجام این کار باید با یک خطا مواجه شوید.

```
Administrator: Windows PowerShell
PS C:\> Clear-DnsClientCache
PS C:\> Resolve-DnsName sid-500.com -CacheOnly
Resolve-DnsName : sid-500.com : DNS record does not exist
At line:1 char:1
+ Resolve-DnsName sid-500.com -CacheOnly
+ ~~~~~
+ CategoryInfo          : ResourceUnavailable: (sid-500.com:String) [Resolve-DnsName], Win32Exception
+ FullyQualifiedErrorId : RECORD_DOES_NOT_EXIST,Microsoft.DnsClient.Commands.ResolveDnsName
```

Clear-DnsClientCache

Resolve-DnsName parspack.com -CacheOnly

### مشخص نمودن سرور DNS

Resolve-DnsName بدون احتیاج به هیچگونه پارامتری با سرور DNS ارتباط برقرار می کند. این سرور در تنظیمات کارت شبکه شما پیکربندی شده است.

```
Administrator: Windows PowerShell
PS C:\> Resolve-DnsName sid-500.com -Server 8.8.8.8 | Format-List

Name       : sid-500.com
Type       : A
TTL        : 299
DataLength : 4
Section    : Answer
IPAddress  : 192.0.78.24
```

Resolve-DnsName parspack.com -Server 8.8.8.8 | Format-List

### سوابق AAAA (تنها برای IPv6)

به منظور دستیابی به سوابق AAAA از دستورالعمل زیر استفاده می کنیم:

```
Administrator: Windows PowerShell
PS C:\> Resolve-DnsName facebook.com -Type AAAA | Format-List

Name       : facebook.com
Type       : AAAA
TTL        : 259
DataLength : 16
Section    : Answer
IPAddress  : 2a03:2880:f107:83:face:b00c:0:25de
```

Resolve-DnsName facebook.com -Type AAAA | Format-List

### دستور LLMNR Only

به منظور استفاده از پیوندهای محلی Multicast از پارامتری به نام LLMNROnly استفاده می کنیم. این دستور تنها با رایانه هایی که لینک های محلی را به اشتراک می گذارند، کار می کنند. همچنین My Computer و Dc01 با یک دیگر پیوند یکسانی دارند. دستور زیر به خوبی نشان دهنده دستورات بالا می باشد:

```
Administrator: Windows PowerShell
PS C:\> Resolve-DnsName dc01 -LlmnrOnly | Format-Table -AutoSize

Name Type TTL Section IPAddress
----
DC01 AAAA 30 Answer 2a02:8388:b01:3700:18d0:b8:9b30:a3fc
DC01 AAAA 30 Answer fe80::18d0:b8:9b30:a3fc
DC01 A 30 Answer 192.168.0.100
```

Resolve-DnsName dc01 -LlmnrOnly | Format-Table -AutoSize

امکان بروز خطا



```
Administrator: Windows PowerShell
PS C:\> Resolve-DnsName sid-500.com -LlmnrOnly | Format-Table -AutoSize
Resolve-DnsName : sid-500.com : DNS record does not exist
At line:1 char:1
+ Resolve-DnsName sid-500.com -LlmnrOnly | Format-Table -AutoSize
+ ~~~~~
+ CategoryInfo          : ResourceUnavailable: (sid-500.com:String) [Resolve-DnsName], Win32Exception
+ FullyQualifiedErrorId : RECORD_DOES_NOT_EXIST,Microsoft.DnsClient.Commands.ResolveDnsName
```

### وضوح نام سه گانه

دستور زیر برای وضوح نام سه گانه و یا به اصطلاح Triple Name Resolution به کار می رود:

```
"parspack.com","facebook.com","cnn.com" | Resolve-DnsName -Type A |
Format-Table -AutoSize
```

همچنین برای دستور بالا راه حل دیگری نیز وجود دارد و آن استفاده از nslookup همراه با -Foreach Object می باشد. استفاده از این راه حل دارای دستورالعمل ذیل می باشد:

```
"parspack.com","facebook.com","cnn.com" | ForEach-Object {nslookup $_}
```

### اجرای Resolve-DnsName با استفاده از پرونده های هاست

اگر می خواهید تمام اسامی که در پرونده هایتان وجود دارند را بازیابی کنید، باید از Get-Content استفاده می کنیم.

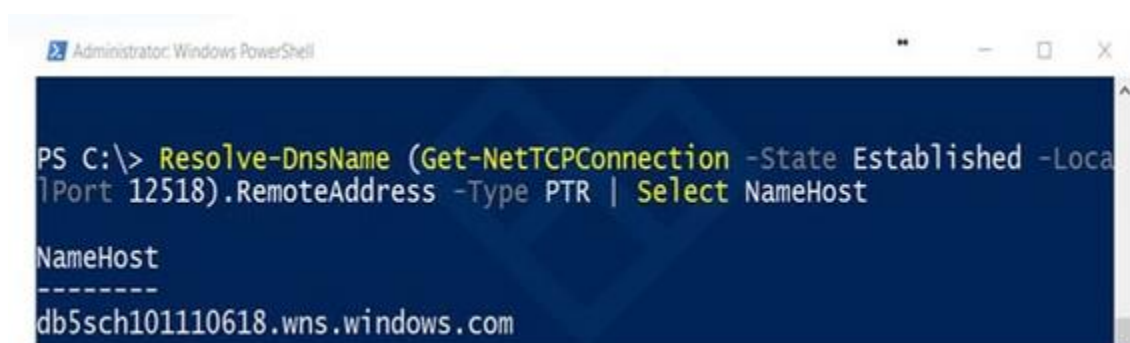


## Resolving DNS Names با استفاده از جدول اتصال TCP

برای توضیحی دقیق تر باید گفت که Get-NetTCPConnection اتصالات فعلی به همراه آدرس IP را می دهد. به همین علت باید تمامی اتصالات را چک کنید و سپس یکی از آن ها را انتخاب کنید و در نهایت Resolve-DnsName را بر روی آن به اجرا برسانید. دستور زیر به خوبی نشان دهنده جملات بالا است:

### Get-NetTCPConnection

دستور بالا اتصال به ۴۰.۷۷.۲۲۹.۴۵ را به ما می دهد. همچنین به پورت محلی ۱۲۵۱۸ دست می یابیم. سپس دستور زیر را اجرا می کنیم:



```
Administrator: Windows PowerShell
PS C:\> Resolve-DnsName (Get-NetTCPConnection -State Established -LocalPort 12518).RemoteAddress -Type PTR | Select NameHost
NameHost
-----
db5sch101110618.wns.windows.com
```

Resolve-DnsName (Get-NetTCPConnection -State Established -LocalPort 12518).RemoteAddress -Type PTR | Select-Object NameHost

Resolve-DnsName یکی از دستورات مفیدی است که می توانید در ابزار قدرتمند و قابل ارتقا پاورشل آن را پیدا کنید و با بهره گیری از آن اطلاعات جامعی را به دست آورید. پایان.