WWW.IRANMEET.COM WWW.IRANMEET.COM

IPv6

اگرچه ممکن است مکانیزمهای CIDR و NAT چند سالی دیگر به دوام نسخهٔ چهارم IP کمک کنند ولی تقریباً بر همه آشکار شده که نفسهای پروتکل IP در شکل کنونی آن، به شماره افتاده است. مضاف بر مشکلات فنی IP، برخی از موارد پشت صحنه و زمینه‌ای دیگر نیز مطرح است. در سالهای اولیه، از اینترنت عموماً در دانشگاهها، صنایع پیشرفته و دولت ایالات متحده (خصوصاً وزارت دفاع) استفاده می‌شد. با گرایش بسیار زیاد مردم به اینترنت که از اواسط دههٔ نود شروع شد، گروههای مختلفی از افراد به آن رو آوردند؛ افرادی که نیازها و انتظارات متفاوتی داشتند. یکی از موارد آنست که افراد باکامپیوترهای بی‌سیم قابل حمل برای در ارتباط بودن با محل استقرار دائمی خود (ایستگاههای خانگی) می‌خواهند از اینترنت بهره بگیرند. مورد دیگر آن که با همگرایی قریب‌الوقوع صنایع کامپیوتر و مخابرات و صنایع تولید بازی و ابزار تفریح، دیری نخواهد پایید که حتی دستگاههای تلفن و تلویزیون در دنیا، به عنوان گرهی از اینترنت، به آن خواهد پیوست و در آن زمان میلیاردها ماشین، از صدا و تصویر بهره خواهند گرفت. با در نظر داشتن چنین چشم‌اندازی ، IP بوضوح نیازمند تغییرات اساسی است و باید انعطاف بیشتری داشته باشد.

IETF که چنین افقی را پیش روی خود می‌دید در اوان ۱۹۹۰ کار را بر روی نسخهٔ جدیدی از پروتکل IP شروع کرد که در آن فضای آدرس هرگز با کمبود مواجه نشود و مشکلات عدیده‌ای را حل کند؛ قابلیت انعطاف بیشتری داشته باشد و در ضمن کارآمدتر باشد. اهداف عمدهٔ IPv6 عبارت بودند از:

۱. پشتیبانی از میلیاردها ماشین میزبان حتی در صورتی که تخصیص فضای آدرس ناکارآمد و با اسراف انجام شود.
۲. کاهش اندازهٔ جداول مسیریابی
۳. ساده‌سازی پروتکل به منظور افزایش سرعت پردازش مسیریابها
۴. ارائه امنیت بهتر در مقایسه با نسخهٔ فعلی IP (شامل احراز هویت و سرّی ماندن داده‌ها)
۵. توجه بیشتر به نوع خدمات و QoS ، به ویژه برای داده‌های بی‌درنگ
۶. کمک به فرآیند ارسال چندپخشی از طریق توصیف حوزه‌ها (Scopes)
۷. فراهم آوردن امکان جابجایی ماشینهای میزبان بدون تغییر در آدرس
۸. امکان ایجاد تغییر و پیشرفت در آینده
۹. امکان همزیستی پروتکلهای جدید و قدیم در طی سالها

برای توسعهٔ پروتکلی که تمام نیازهای فوق‌الذکر را برآورده نماید، IETF با انتشار RFC 1550 و در طی یک فراخوان، خواستار پیشنهادات دیگران در این خصوص شد. ۲۱ پیشنهاد دریافت گردید که اغلب آنها جامع نبودند. تا دسامبر ۱۹۹۲ فقط هفت طرح پیشنهادی قابل توجه در دستور کار قرار داشت. این طرحهای پیشنهادی از اصلاحات جزیی در نسخهٔ فعلی IP تا پیشنهاد دور انداختن آن و جایگزینی با یک پروتکل کاملاً متفاوت را شامل می‌شد.

یک پیشنهاد آن بود که TCP بر روی CLNP اجرا شود؛ پروتکلی که با آدرسهای ۱۶۰ بیتی فضای آدرس‌دهی نامحدود و جاویدان را فراهم کرده بود و دو پروتکل عمده و مهم لایهٔ شبکه را متحد و یکنواخت می‌کرد. ولیکن بسیاری افراد احساس کردند که پذیرش آن مهر تأییدی است بر این ادعا که هر کاری که OSI انجام داده صحیح تلقی می‌شود، داعیه‌ای که لااقل در حوزهٔ اینترنت به دلایل خاص نادرست است. الگوی CLNP بسیار شبیه به IP بود و این دو، تفاوت چندانی با هم ندارند. در آخر نیز طرحی انتخاب شد که تفاوت بسیار زیادی با IP و از آن بیشتر با CLNP دارد. ضربهٔ دیگری که CLNP خورد از آنجا بود که پشتیبانی ضعیفی از «نوع خدمات» (Type of Service) می‌کرد، خصوصیتی که برای انتقال کارآمد داده‌های چند رسانه‌ای به شدت نیاز بود.

سه تا از بهترین طرحهای پیشنهادی در ژورنال IEEE Network منتشر شد.[۱] پس از مباحثات فراوان، بازبینی ، ارزیابی موقعیت و سنجش استقبال عمومی، نسخه‌ای ترکیبی از طرحهای پیشنهادی Deering و Francis که SIPP[۲] نامیده می‌شد به عنوان طرح برگزیده معرفی و با عنوان IPv6 معرفی گردید.

IPv6 بخوبی اهداف مورد نظر را برآورده می‌کند: ویژگیهای خوب IP را نگه داشته، ویژگیهای بد را کنار گذاشته یا کمرنگ کرده و ویژگیهای جدیدی به آن افزوده است. بطور کلی IPv6 با IPv4 سازگار نیست ولی با تمام پروتکلهای جانبی اینترنت مثل TCP ، UDP ، ICMP ، IGMP ، OSPF ، BGP و DNS سازگار است. (البته ممکن است به دلیل آنکه آدرسها طولانی‌تر شده‌اند نیاز به اندکی تغییر داشته باشند.) ویژگیهای اساسی IPv6 در زیر تشریح شده است. برای آگاهی بیشتر در خصوص آن به RFCهای 2460 تا 2466 مراجعه نمایید.

اولین و مهمترین ویژگی آنست که IPv6 آدرسهای بسیار طولانی‌تری نسبت به IPv4 دارد. این آدرسها ۱۶ بایت طول دارند و دقیقاً مشکلی را حل

---

۱. Deering, 1993; Francis, 1993; and Katz and Ford, 1993

۲. Simple Internet Protocol Plus

Ramin.samad@yahoo.com

WWW.IRANMEET.COM WWW.IRANMEET.COM

کرده که به همان دلیل طراحی شد: یعنی تقریباً فضای نامحدودی از آدرسهای IP را فراهم آورده است. در این خصوص بیشتر صحبت خواهیم کرد.

دومین پیشرفت عمدهٔ IPv6 ساده‌سازی سرآیند آنست. این سرآیند جمعاً هفت فیلد دارد (در مقابل سیزده فیلد در IPv4). این تغییر، امکان آنرا فراهم آورده که مسیریاب بسته‌ها را سریعتر پردازش نماید و ظرفیت مفید مسیریاب را افزایش و تأخیر را کاهش دهد. در ادامه مختصراً به سرآیند خواهیم پرداخت.

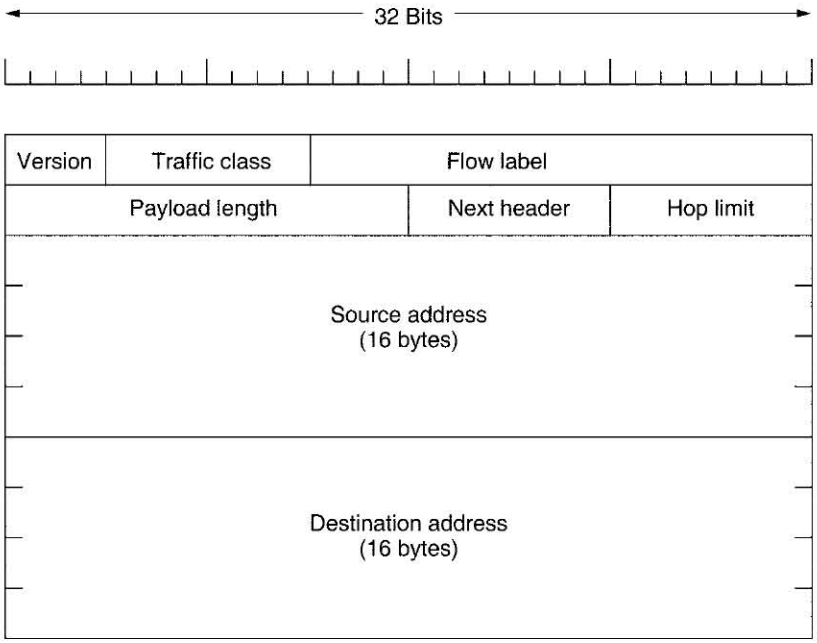
سومین بهبود عمدهٔ آن پشتیبانی از گزینه‌های اختیاری (Options) است. این تغییر برای سرآیند جدید حیاتی بود چراکه برخی از فیلدهایی که در نسخهٔ قبلی وجودشان الزامی است در نسخهٔ فعلی اختیاریند. مضاف بر این، روش درج گزینه‌ها در این فیلد متفاوت از قبل است و اجازه می‌دهد مسیریابها بتوانند به سادگی از گزینه‌هایی که برایشان مهم نیست رد شوند. [یعنی در نسخهٔ جدید، دسترسی تصادفی و مستقیم به گزینه‌ها ممکن شده است.] این ویژگی سرعت پردازش بسته‌ها را افزایش می‌دهد.

چهارمین موضوعی که IPv6 در آن پیشرفت عمده‌ای داشته است، «**امنیت**» است. IETF با انبوهی از گزارشهای مطبوعاتی در خصوص نابغه‌های دوازده ساله‌ای مواجه بود که باکامپیوترهای شخصی خود، به شبکه‌های بانکی یا پایگاههای نظامی در سراسر اینترنت، نفوذ کرده بودند و جوّ شدیدی براه افتاده بود که باید برای بهبود امنیت شبکه‌ها کاری کرد. در نسخهٔ جدید IP ، «احراز هویت» (Authentication) و «حفظ امنیت اطلاعات» (Privacy) جزو ویژگیهای کلیدی به شمار می‌رود. البته این ویژگیها بعداً به IPv4 نیز افزوده شد [با عنوان IPsec]، فلذا در حال حاضر این دو ، در زمینهٔ امنیت تفاوت چندانی با هم ندارند.

موضوع آخر آنکه در نسخهٔ جدید به «کیفیت خدمات» (QoS) دقت بیشتری شده است. قبل از آن نیز تلاشهای جَسته وگریخته‌ای در این رابطه انجام شده بود ولیکن با رشد کاربردهای چند رسانه‌ای در اینترنت، این موضوع جدی‌تر و حساس‌تر به نظر می‌رسید.

سرآیند اصلی IPv6

سرآیند اصلی IPv6 در شکل ۱ نشان داده شده است. فیلد Version (شمارهٔ نسخهٔ پروتکل) برای IPv6 همیشه ۶ است. (کما اینکه برای IPv4 نیز همیشه ۴ است!). در دوران گذار از IPv4 به نسخهٔ جدید که ممکن است یک دهه طول بکشد، مسیریابها قادرند با بررسی این فیلد تشخیص بدهند که با چه نوع بسته‌ای روبرو هستند. البته از آنجایی که بررسی این فیلد به چندین دستورالعمل اجرایی CPU نیاز دارد و این کار زمان مفید پردازش هر بسته را هدر می‌دهد لذا در بسیاری از پیاده‌سازیهای عملی، برای اجتناب از این زمان تلفاتی، تشخیص آنکه یک بسته از نوع IPv4 است یا IPv6 ، با استفاده از فیلد خاصی در سرآیند لایهٔ پیوند داده‌ها بر عهدهٔ سخت‌افزار گذاشته شده است.[1] بدین ترتیب بسته‌ها براساس نوعشان مستقیماً به نرم‌افزار مناسب در لایهٔ شبکه هدایت می‌شوند. البته این الزام که لایهٔ پیوند داده از جزئیات نوع بسته‌های لایهٔ شبکه آگاه باشد با این اصل اساسی که «هر لایه نباید از معنای بیتهایی که از لایهٔ بالاتر تحویل او می‌شود، آگاه باشد» در تناقض است. بدون شک بحث و مناقشه بین طرفداران ایده‌های «انجام اصولگرایانه و صحیح کار» و «تسریع کار» به شدت ادامه خواهد داشت.



۱. مثلاً در فریم اترنت فیلد Type نوع بستهٔ درون فریم را تعیین می‌کند. م

Ramin.samad@yahoo.com

WWW.IRANMEET.COM WWW.IRANMEET.COM

فیلد Traffic Class (کلاس ترافیک) برای تشخیص تفاوت بسته‌ها از لحاظ نیازمندیهای تحویل بی‌درنگ و QoS درخواستی، بکار می‌آید. فیلدی با همین منظور از ابتدا در IP وجود داشت ولیکن استفاده از آن به صورت پراکنده و سلیقه‌ای بر روی مسیریابها پیاده‌سازی شد و اغلب مسیریابها آن را نادیده می‌گرفتند. اکنون تجربیات گذشته چراغ راهی شده تا بتوان بهترین راه و روش تحویل بسته‌های اطلاعات چندرسانه‌ای را تعیین کرد.

فیلد Flow Label (برچسب جریان) همچنان آزمایشی است ولی کاربرد مورد نظر آن، این بوده که بتوان یک «**شبه اتصال**» (Pseudoconnection) بین مبداء و مقصد، با ویژگیها و نیازمندیهای خاصّ ایجاد کرد. به عنوان مثال، جریانی از بسته‌ها که از یک پروسه در مبداء خاص تولید و به سوی یک پروسه بر روی مقصد خاص روانه می‌شوند احتمالاً نیاز به تضمین تأخیر محدود و مشخص دارد و در نتیجه باید پهنای باند لازم را رزرو کرد. در چنین مواردی می‌توان پیشاپیش یک «جریان» (Flow) با مشخصات درخواستی تنظیم کرد و به آن یک شناسه اختصاص داد. هر گاه مسیریاب بسته‌ای دریافت کند و فیلد Flow Label آن غیرصفر باشد، با مراجعه به جداول درونی خود تشخیص می‌دهد که با این بسته چگونه رفتار کند. در حقیقت استفاده از مفهوم «جریان»[۱] در IPv6، تلاشی است برای رسیدن به قابلیت انعطاف در زیرشبکه‌های دیتاگرام و تضمین کیفیت خدمات در زیرشبکه‌های مدار مجازی.

هویت هر «جریان» برحسب آدرس مبداء، آدرس مقصد و شمارهٔ جریان (برچسب جریان) مشخص می‌شود فلذا بین دو مبداء و مقصد در شبکه می‌توان بطور همزمان چندین «جریان» فعال تنظیم کرد. همچنین در این روش حتی اگر دو جریان متفاوت با شمارهٔ جریان یکسان از دو ماشین میزبان مختلف تولید و از مسیریابهای مشابهی عبور کنند، مسیریابها به کمک آدرس مبداء و مقصد قادر به تشخیص آنها خواهند بود. انتظار آنست که «برچسبهای جریان» به جای آنکه به صورت ترتیبی و از ۱ شروع شوند به صورت کاملاً تصادفی انتخاب گردند تا مسیریاب بتواند آنها را در Hash Table خود درج کند.[۲]

فیلد Payload Length (طول قسمت حمل داده) مشخص می‌کند که پس از سرآیند ۴۰ بایتی در شکل ۱ چند بایت داده قرار گرفته است. همین فیلد در IPv4 با نام Total Length وجود داشت. تغییر نام به آن دلیل بوده که در نسخهٔ جدید ، سرآیند جزو طول بسته به حساب نمی‌آید بلکه فقط اندازه بخش حمل داده تعیین می‌شود.

فیلد Next Header ساختار بسته را سبکبار کرده است! دلیل آنکه سرآیند بسته ساده شده آنست که می‌توان در صورت لزوم سرآیند اضافی و انتخابی داشت. این فیلد مشخص می‌کند که پس از سرآیند ۴۰ بایتی کدامیک از سرآیندهای ششگانهٔ اضافی قرار گرفته است (در صورت وجود). اگر سرآیند اخیر، آخرین سرآیند بستهٔ IP باشد، این فیلد مشخص می‌کند که کدام پروسه در لایهٔ انتقال محتوای بسته را تحویل خواهد گرفت (مثلاً TCP ، UDP و نظائر آن).[۳]

کاربرد فیلد Hop Limit آنست که بسته‌ها عمر محدودی داشته باشند. این فیلد در عمل مشابه با فیلد Time to Live (زمان حیات بسته) در IPv4 است یعنی به ازای عبور بسته از یک مسیریاب، یک واحد از مقدار آن کاسته می‌شود. در تئوری، مبنایی که IPv4 برای این فیلد در نظر داشت، «زمان بر مبنای ثانیه» بود در حالی که هیچ مسیریابی از چنین مبنایی استفاده نمی‌کند (بلکه به ازای هر گام یک واحد از آن می‌کاهد) لذا نام این فیلد را به گونه‌ای عوض کردند که عملکرد واقعی آن را نشان بدهد. هر گاه مقدار این فیلد در یک بسته به صفر برسد آن بسته حذف خواهد شد.

در ادامه فیلدهای Source Address و Destination Address (آدرس مبداء و مقصد) قرار گرفته‌اند. در طرح پیشنهادی آقای Deering آدرسها ۸ بایتی انتخاب شده بودند در حالی که در مراحل بازبینی دیگران احساس کردند که شاید این فضای آدرس نیز در خلال چند دهه، IPv6 را نیز با کمبود فضای آدرس مواجه کند، در حالی که با آدرسهای ۱۶ بایتی هرگز چنین کمبودی رخ نخواهد داد. برخی از افراد معتقد بودند که آدرسهای ۱۶ بایتی بیش از حد بزرگ هستند در حالی برخی دیگر اعتقاد داشتند باید از آدرسهای ۲۰ بایتی استفاده شود تا با پروتکل دیتاگرام پیشنهادی OSI سازگار باشد. گروه دیگری نیز به آدرسهای با طول متغیر گرایش داشتند. پس از بحث و جدل فراوان، به این نتیجه رسیدند که آدرسهای با طول ثابت ۱۶ بایتی بهترین انتخاب است.

با توجه به طول زیاد آدرسهای IP، نماد جدیدی برای نوشتن آنها پیشنهاد شد. این آدرسها به صورت هشت **گروه** که با علامت : از هم جدا شده، نوشته می‌شوند. هر گروه نیز به صورت چهار رقم هگزادیسمال نمایش داده‌می‌شود:

---

۱. برای آشنایی با مفهوم جریان رجوع کنید به بخش ۵-۴-۱

۲. به عبارت دیگر مسیریاب انتظار دارد این شماره‌ها را Hash کند لذا این شماره‌ها نباید متوالی باشند.

۳. به عبارت دیگر محتوای این فیلد به صورت بازگشتی سرآیندهای بعدی را مشخص می‌کند تا نهایتاً به سرآیند آخر برسد که نوع بستهٔ لایهٔ انتقال را تعیین می‌نماید.

Ramin.samad@yahoo.com

WWW.IRANMEET.COM WWW.IRANMEET.COM

8000:0000:0000:0000:0123:4567:89AB:CDEF

از آنجایی که در آدرسها، تعداد ارقام صفر زیاد است، سه نوع بهینه‌سازی مجاز شمرده شده: صفرهای سمت چپ در هر گروه نوشته نمی‌شوند یعنی 0123 به صورت 123 نشان داده می‌شود؛ دوم آنکه اگر یک یا چند گروه شانزده بیتی تماماً صفر باشند با یک زوج علامت :: نشان داده می‌شود. بنابراین آدرس مثال بالا به صورت زیر نوشته خواهد شد:

8000::123:4567:89AB:CDEF

نهایتاً آنکه آدرسهای IPv4 را می‌توان با یک جفت :: و سپس آدرس نقطه‌دار قدیمی، نشان داد:

::192.31.20.46

شاید لازم به گفتن نباشد که آدرسهای شانزده بایتی، فضایی معادل 2^{128} آدرس هستند که چنین فضایی تقریباً معادل 3×10^{38} آدرس است. اگر کل کرهٔ زمین شامل خشکیها و دریاها پر از کامپیوتر شوند باز هم IPv6 می‌تواند برای هر مترمربع 7×10^{23} آدرس IP فراهم کند. دانشجویان رشته شیمی می‌دانند که این عدد حتی از عدد آووگادرو نیز بزرگ است. [عدد آووگادرو $6/02\times10^{23}$ است]. چون در نظر نبوده که حتی به مولکولهای سطح زمین آدرس بدهیم آدرسهای IP شانزده بایتی، بهیچوجه کم نخواهد آمد!!

در عمل از فضای آدرس IP، بخوبی استفاده نخواهد شد. (دقیقاً همانند فضای شماره‌های تلفن که مثلاً فضای شماره‌های تلفن منهتن با پیش شمارهٔ ۲۱۲ پر شده ولی فضای شماره‌های ویومینگ (Wyoming) با پیش شمارهٔ ۳۰۷ تقریباً خالی مانده است.) دو نفر به نامهای Durand و Huitema در سند RFC 3194 محاسبه کرده‌اند که با ایده گرفتن از تخصیص شماره‌های تلفن و حتی در بدبینانه‌ترین حالت ممکن، باز هم می‌توان برای هر مترمربع از کرهٔ زمین، ۱۰۰۰ آدرس IP کنار گذاشت. در حالت کلی نیز می‌توان تریلیونها آدرس IP برای هر مترمربع از زمین در نظر گرفت. کوتاه سخن آنکه، در آینده هیچگاه به مشکل فضای آدرس برنخواهیم خورد.

مقایسهٔ سرآیند IPv4 با سرآیند IPv6 از این دیدگاه که چه فیلدی و چرا حذف شده است، آموزنده خواهد بود: فیلد IHL حذف شده زیرا سرآیند بسته‌های IPv6 طول ثابتی دارد. فیلد «پروتکل» وجود ندارد چرا که فیلد Next Header مشخص می‌کند که پس از آخرین سرآیند چه بستهٔ دیگری آمده است (بستهٔ TCP، UDP یا نظائر آن).

تمام فیلدهایی که در ارتباط با «قطعه‌قطعه‌سازی» بسته‌ها در IPv4 تعریف شده بود در IPv6 حذف گردیده است زیرا پروتکل اخیر راهکار دیگری برای مکانیزم قطعه‌قطعه سازی برگزیده است. انتظار آنست که تمام ماشینهای سازگار با IPv6 بتوانند به صورت خودکار و پویا اندازهٔ دیتاگرامها را تعیین کنند و بدین نحو نیاز به قطعه‌قطعه شدن بسته‌ها کمتر اتفاق می‌افتد. همچنین حداقل طول بسته‌ای که هر ماشین موظف به پذیرش آنست از ۵۷۶ بایت به ۱۲۸۰ بایت افزایش یافته تا بتوان یک قطعه دادهٔ ۱۰۲۴ بایتی را به همراه تعداد زیادی سرآیند (۲۵۶ بایت)، بدون نیاز به قطعه‌قطعه شدن ارسال و دریافت کرد. مضاف بر این، وقتی ماشینی یک بستهٔ بیش از حد بزرگ IPv6 را ارسال می‌دارد مسیریاب ناتوان از هدایت آن، به جای قطعه‌قطعه کردن بسته آن را حذف کرده و پیام خطایی را باز می‌گرداند. این پیام به ماشین میزبان تفهیم می‌کند که باید بسته‌هایش را بشکند. البته اگر ماشین میزبان خودش بسته‌ها را با اندازهٔ مناسب ارسال کند کارآمدتر از آنست که بسته‌ها در طول مسیر شکسته شود.

فیلد Checksum نیز حذف شد زیرا محاسبهٔ آن کارآیی و سرعت پردازش بسته‌ها را به نحو چشمگیری کاهش خواهد داد. با توجه به قابلیت اعتماد شبکه‌های کنونی و با در نظر داشتن این حقیقت که در لایهٔ پیوند داده و لایهٔ انتقال نیز (بطور مجزا) صحت داده‌ها بررسی می‌شود، محاسبهٔ یک کد کشف خطای دیگر مثل checksum در مقایسه با کاهش کارآیی ارزشی ندارد. حذف این ویژگیهای زائد، IPv6 را به پروتکل متعادل و جمع و جور تبدیل کرده است. بدین ترتیب IPv6 به اهداف مورد نظر خود که همانا انعطاف، سرعت و فضای بزرگ آدرس بوده، نائل شده است.

سرآیندهای اضافی (سرآیندهای توسیع یا Extension Header)

گاهی به برخی از فیلدهای حذف شدهٔ IPv4 نیاز می‌شود و به همین منظور در IPv6 مفهوم جدیدی به نام «سرآیندهای توسیع» معرفی شده است. این سرآیندهای اختیاری برای افزودن اطلاعات به هر بسته بکار می‌آیند ولیکن روش کدینگ (و جاسازی) آنها کارآمد و سریع است. شش نوع مختلف سرآیند توسیع که تاکنون معرفی شده، در شکل ۲ فهرست گردیده است. هر کدام از این سرآیندها اختیارند ولیکن اگر به بیش از یک سرآیند نیاز باشد باید بطور پیاپی، پس از سرآیند ثابت و ترجیحاً به ترتیب فهرست، قرار بگیرند.

نام سرآیند توسیع (سرآیند اضافی)	توصیف عملکرد
Hop-by-hop options	حاوی اطلاعات گوناگون برای مسیریابها
Destination options	اطلاعات اضافی برای مقصد
Routing	فهرست ناکاملی از مسیریابها که بسته باید از آنها بگذرد.
Fragmentation	مدیریت قطعات دیتاگرام
Authentication	بررسی هویت فرستنده
Encrypted security payload	اطلاعاتی در خصوص محتوا[illegible]

Ramin.samad@yahoo.com

برخی از سرآیندها دارای قالب ثابتی هستند در حالی که برخی دیگر تعداد متغیری فیلد با طول متفاوت دارند. به همین دلیل هر آیتم در قالب سه تایی **(نوع، طول، مقدار)**[1] سازماندهی و کُد می‌شود. فیلد Type مشخص می‌کند که نوع گزینه چیست. مقدار فیلد **نوع** (Type) به نحوی انتخاب شده است که دو بیت ابتدایی آن به مسیریابهایی که نمی‌دانند آن گزینه را چگونه پردازش کنند، راه و چگونگی کار را نشان می‌دهند. این راهکارها عبارتند از: (۱) گزینهٔ مربوط را نادیده بگیر (۲) بسته را حذف کن (۳) بسته را حذف و یک بستهٔ ICMP برگردان (۴) بسته را حذف کن و یک بستهٔ ICMP برگردان ولیکن بستهٔ ICMP را برای آدرسهای «چندپخشی» (Multicast) نفرست. (تا یک بستهٔ چندپخشی اشتباه، منجر به تولید میلیونها گزارش ICMP نشود.)

فیلد یک بایتی **طول** (Length) مشخص می‌کند که فیلد **مقدار** (Value) چند بایتی است. (صفر تا ۲۵۵ بایت). فیلد **مقدار** (Value) در برگیرنده اطلاعات مورد نیاز است و حداکثر می‌تواند ۲۵۵ بایت باشد.

«سرآیند توسیع Hop-by-Hop» (گام به گام) برای حمل اطلاعاتی کاربرد دارد که مسیریابهای واقع بر مسیر باید آنها را بررسی نمایند. قبلاً یکی از گزینه‌ها را معرفی کردیم: پشتیبانی از دیتاگرامهایی با طول بیش از ۶۴کیلوبایت. قالب این سرآیند در شکل ۳ نشان داده شده است. وقتی از این سرآیند استفاده می‌شود باید فیلد Payload Length (در سرآیند اصلی) به صفر مقداردهی شود.

Next header	0	194	4
Jumbo payload length			

همانند تمام سرآیندهای اختیاری دیگر، این سرآیند نیز با فیلدی یک بایتی به نام Next Header شروع می‌شود و مشخص می‌کند که سرآیند بعدی از چه نوع است. پس از این بایت، بایت دیگری قرار گرفته که طول سرآیند Hop-by-Hop را بر مبنای بایت تعیین می‌کند ولیکن در مقدار آن، ۸ بایت ابتدایی (که وجود آن الزامی است) لحاظ نمی‌شود. تمام سرآیندهای توسیع دیگر نیز به همین نحو شروع می‌شوند. در ادامه فیلدی دو بایتی آمده که بایت اول مشخص می‌کند که این گزینه (Option) قرار است اندازهٔ دیتاگرام را تعریف کند (کد ۱۹۴) و بایت بعدی مشخص می‌کند که اندازهٔ دیتاگرام یک شمارهٔ چهار بایتی است. چهار بایت آخر این سرآیند، طول دیتاگرام را مشخص می‌کند. اندازهٔ زیر ۶۵۵۳۶ مجاز نیست و منجر به حذف بسته در اولین مسیریاب و بازگشت پیام خطای ICMP خواهد شد. دیتاگرامهایی که از این سرآیند اختیاری (یعنی Hop-by-Hop Header) استفاده کرده‌اند اصطلاحاً Jumbogram (دیتاگرام عظیم) نامیده می‌شوند. [بدین ترتیب در IPv6 می‌توان قطعات دادهٔ بسیار بزرگ را به کمک سرآیند فوق به صورت یکجا ارسال کرد.] کاربرد جامبوگرامها در سوپرکامپیوترها که باید چندین گیگابایت اطلاعات را از طریق اینترنت منتقل کنند، بسیار حیاتی است.

«سرآیند توسیع Destination Options» برای درج فیلدهایی در نظر گرفته شده که صرفاً توسط ماشین مقصد پردازش و تفسیر می‌شوند. در نسخهٔ اولیهٔ IPv6، مقدار این گزینه پوچ در نظر گرفته شده و کاربردی نداشته است. وجود چنین فیلدی برای آن بوده که نرم‌افزار ماشینهای میزبان و مسیریابها چنین سرآیندی را به رسمیت بشناسند تا اگر روزگاری به آن نیاز شد، شرایط مهیا باشد وگرنه باید پروتکل عوض شود.

«سرآیند توسیع Routing»، فهرست مسیربایهایی را مشخص می‌نماید که بسته باید در راه رسیدن به مقصد از آنها عبور کند. این گزینه شباهت زیادی به گزینهٔ Loose Source Routing در IPv4 دارد. فهرست آدرسهایی که در این سرآیند مشخص شده باید در طول مسیر و به ترتیب ملاقات شوند ولی این امکان وجود دارد که مسیریابهایی هم که آدرس آنها در فهرست نیست مابین مسیر باشند. قالب سرآیند Routing در شکل ۴ مشخص شده است.

Next header	Header extension length	Routing type	Segments left
Type-specific data			

۱. (Type,Length,Value)

WWW.IRANMEET.COM WWW.IRANMEET.COM

چهار بایت اول از این سرآیند، شامل چهار فیلد یک بایتی است: دو فیلد Next Header و Header Extension Length را قبلاً تعریف کردیم. فیلد Routing Type، ساختار مابقی سرآیند را مشخص می‌نماید: مقدار صفر مشخص کنندهٔ آنست که پس از کلمهٔ چهار بایتی اول، یک کلمهٔ چهار بایتی دیگر قرار گرفته و پس از آن آدرس IPv6 (یعنی آدرسهای ۱۲۸ بیتی) مسیریابها قرار می‌گیرد. به غیر از این ساختار، فعلاً ساختار دیگری تعریف نشده مگر آنکه در آینده چیز جدیدی ابداع شود. فیلد آخر یعنی Segment Left تعداد آدرسهایی را مشخص می‌کند که هنوز ملاقات نشده‌اند. مقدار اولیهٔ این فیلد معادل با تعداد مسیریابهایی است که آدرس آنها در فهرست مورد نظر درج شده است و به ازای ملاقات در هر مسیریاب که آدرس آن در فهرست آمده یک واحد از این فیلد کاسته می‌شود. وقتی مقدار این فیلد در بسته به صفر برسد بسته روال طبیعی طی مسیر خود را از سر می‌گیرد بدون آنکه اجبار به عبور از مسیر خاصی داشته باشد. معمولاً در چنین لحظه‌ای بسته به مقصد خود نزدیک شده است.

سرآیند Fragmentation مشابه با IPv4، با مسئله قطعه‌قطعه سازی بسته‌ها سر و کار دارد. در این سرآیند نیز فیلدهای «شمارهٔ شناسایی دیتاگرام»، «شمارهٔ قطعه» و یک بیت MF تعریف شده‌اند که بیت MF مشخص می‌کند که آیا قطعهٔ جاری آخرین قطعهٔ دیتاگرام است یا آنکه قطعات دیگری در ادامه وجود دارند. البته در IPv6 برخلاف IPv4، فقط ماشین مبداء می‌تواند بسته‌ای را قطعه‌قطعه کند و مسیریابهای واقع بر روی مسیر قادر به چنین کاری نیستند. اگرچه این موضوع از لحاظ فلسفی یک واپسگرایی محسوب می‌شود ولی در عوض کار مسیریابها را ساده‌تر کرده و فرآیند مسیریابی سریعتر خواهد شد. همانگونه که قبلاً اشاره کردیم هر گاه یک مسیریاب با بسته‌ای بیش از حد بزرگ مواجه گردد آن را حذف کرده و بستهٔ ICMP (حامل پیغام خطا و اطلاعات مفید دیگر) به مبداء آن بر می‌گرداند. اطلاعات ارسالی به مبداء بسته، امکان آنرا می‌دهد که به کمک این سرآیند، بسته را به قطعات کوچکتر تقسیم و آنها را از نو ارسال کند.

سرآیند Authentication (سرآیند احراز هویت) مکانیزمی را فراهم آورده تا گیرنده بتواند از هویت فرستندهٔ بسته مطمئن شود. سرآیند Encrypted Security Payload اجازه می‌دهد تا محتوای بسته رمزنگاری شود و بدین ترتیب فقط گیرندهٔ مورد نظر قادر به خواندن آنست. این گونه سرآیندها برای انجام مأموریت خود از تکنیکهای رمزنگاری بهره می‌گیرند.

اختلاف نظرها و مناقشات

نظر به آنکه فرآیند طراحی IPv6، «باز» بوده و افراد درگیر در طراحی، بر عقاید خود تأکید داشته‌اند فلذا شگفت‌آور نیست که بسیاری از گزینه‌های انتخابی در IPv6 متناقض باشند. در زیر اجمالاً برخی از آنها را بررسی خواهیم کرد. برای آگاهی از جزئیات ماجرا به RFCهای مربوطه مراجعه نمائید.

قبلاً اشاره کردیم که بحث و جدل گسترده‌ای پیرامون طول آدرسها وجود داشت و توافق نهایی آن بود که آدرسها با طول ثابت و ۱۶ بایتی باشند.

جدل دیگری بر سر حداکثر تعداد گام (Hop Limit) در گرفت. یک گروه احساس می‌کرد که محدود کردن حداکثر تعداد گام (Hop) به ۲۵۵ یک اشتباه محض است چرا که اگرچه در آن زمان حداکثر طول مسیرها عموماً از ۳۲ تجاوز نمی‌کرد ولی مدعی بودند که ممکن است ده سال بعد مسیرها طولانی‌تر از ۲۵۵ باشند. استدلال آنها این بود که فضای آدرس ۱۶ بایتی آینده‌نگری بیش از اندازه و در عوض مقدار کم Hop Count، کوته‌نظری است. از دیدگاه آنها بزرگترین اشتباه یک دانشمند کامپیوتر، آنست که برای هر فیلدی، تعداد بیت کمی در نظر بگیرد.

پاسخ گروه مقابل آن بود که افزایش بی‌مورد فضای هر فیلد منجر به تشکیل یک سرآیند حجیم خواهد شد. همچنین استدلال دیگرشان آن بود که وظیفهٔ فیلد Hop Count جلوگیری از سرگردانی بسته‌ها به مدت طولانی است و ۶۵۵۳۵ گام بیش از حد زیاد است. استدلال آخر آنکه با رشد اینترنت، لینکهای بسیار طولانی ساخته می‌شوند و این امکان فراهم می‌شود که برای رسیدن از یک کشور به کشور دیگر به کمتر از ده گام نیاز باشد. اگر یک بسته برای رسیدن از مبداء به مقصد مجبور شود از ۱۲۵ مسیریاب بین‌المللی بگذرد، ستون فقرات این شبکهٔ بین‌المللی در جایی اشکال دارد! بدین ترتیب طرفداران فیلد ۸ بیتی در عقیدهٔ خود پیروز شدند.

یکی دیگر از بحثهای داغ بر سر حداکثر طول بسته‌ها بود. سوپرکامپیوترها به بسته‌هایی با طول بیش از ۶۴ کیلوبایت احتیاج داشتند. وقتی یک سوپرکامپیوتر شروع به ارسال می‌کند و به کار خود مشغول می‌شود نباید به ازای هر ۶۴ کیلوبایت یکبار متوقف شود. استدلال گروه مخالف آن بود که اگر یک بستهٔ یک مگابایتی در طول مسیر به یک خط T1 برسد آن خط به مدت حداقل ۵ ثانیه اشغال شده و کاربران دیگری که در این خط سهیم هستند با تأخیر قابل توجهی روبرو خواهند شد.[1] توافق نهایی بدینجا ختم شد که بسته‌های معمولی حداکثر ۶۴ کیلوبایتی باشند ولی به کمک سرآیند اختیاری Hop-by-Hop بتوان جامبوگرامهایی با هر طول دلخواه ارسال کرد.

موضوع سوم مناقشه، حذف فیلد IPv4 checksum (کد تشخیص خطاهای احتمالی در سرآیند) بود. برخی از افراد حذف این فیلد را مشابه با برداشتن ترمزهای یک خودرو می‌دانستند که اگرچه ماشین را سبکبار و سریعتر می‌کند ولی اگر اتفاق غیرمترقبه‌ای رخ بدهد مشکل جدی بوجود می‌آید. استدلال گروه مقابل آن بود که هر برنامهٔ کاربردی که نگران صحت داده‌های خود است باید از پروتکلی در لایهٔ انتقال بهره بگیرد که داده‌ها را از

---

۱. نرخ ارسال خط T1 را 1.544 مگابیت در نظر بگیرید.

Ramin.samad@yahoo.com

WWW.IRANMEET.COM WWW.IRANMEET.COM

لحاظ سلامت بررسی می‌کند. لذا اضافه کردن کد کنترلی دیگر به لایهٔ IP برای کشف خطا (در حالی که هر بسته یکبار هم در لایهٔ پیوند داده بررسی می‌شود) بیهوده و زائد است. مضاف بر آن، تجربه نشان داده بود که محاسبهٔ جمع کنترلی (Checksum) در IPv4 هزینهٔ بالایی دارد. در این مناقشه نیز طرفداران حذف کد کشف خطا پیروز شدند.

موضوع دیگر، بحث ماشینهای همراه بود: وقتی یک کامپیوتر قابل حمل، در نیمی از کل دنیا حرکت می‌کند (مثلاً درون هواپیما)، آیا می‌تواند با همان آدرس IPv6 قبلی، کار خود را ادامه بدهد یا آنکه مجبور به استفاده از ساختار «عامل خانگی» و «عامل خارجی» است؟ ماشینهای همراه مشکل «عدم تقارن» را به سیستم مسیریابی تحمیل می‌کنند: یک کامپیوتر همراه و کوچک براحتی قادر به شنیدن سیگنال قوی منتشره از مسیریاب ثابت خود هست ولی مسیریاب ثابت براحتی قادر به احساس سیگنال ضعیف ارسال شده توسط کامپیوتر همراه نیست. در نتیجه برخی از افراد گرایش داشتند که در IPv6 از ماشینهای همراه حمایت شود ولی تمام این تلاشها به دلیل آنکه بر روی هیچیک از طرحهای پیشنهادی توافقی بدست نیامد، با شکست مواجه گردید.

شاید بزرگترین مناقشه بر سر موضوع «امنیت» بود: همه بر این اصل که «امنیت لازم است» اشتراک نظر داشتند. دعوا بر سر چگونگی رسیدن به امنیت و محل پرداختن به آن بود. اولین محل پرداختن به امنیت لایهٔ شبکه است. استدلال موافقین مبنی بر آن بود که پیاده‌سازی امنیت در لایهٔ شبکه، سرویسی استاندارد فراهم می‌کند که تمام برنامه‌های کاربردی بدون هیچگونه برنامه‌ریزی قبلی می‌توانند از آنها بهره بگیرند. استدلال مخالفین نیز آن بود که برنامه‌های کاربردی امن، عموماً به هیچ مکانیزمی کمتر از رمزنگاری انتهابه‌انتها (End-to-End Encryption) احتیاج ندارند، به نحوی که پروسهٔ مبداء خودش داده‌های ارسالی خود را رمز کرده و پروسهٔ مقصد آنها را از رمز خارج کند. هر چیزی کمتر از این، می‌تواند کاربر را با خطراتی مواجه کند که از اشکالات امنیتی لایهٔ شبکه ناشی می‌شود و هیچ تقصیری از او نیست. پاسخ به این استدلال آن بود که کاربر می‌تواند امنیت لایهٔ IP را نادیده بگیرد و کار خودش را انجام بدهد! پاسخ نهایی مخالفین نیز آن بود که افرادی که به عملکرد صحیح شبکه (در خصوص امنیت) اعتماد ندارند چرا باید هزینهٔ پیاده‌سازی سنگین و کندی IP را بپردازند!!

یکی دیگر از جنبه‌های مربوط به امنیت این حقیقت بود که بسیاری از کشورها قوانین سخت‌گیرانه‌ای در مورد صادرات محصولات مرتبط با رمزنگاری وضع کرده‌اند. از مثالهای بارز می‌توان به فرانسه و عراق اشاره کرد که حتی استفاده از رمزنگاری در داخل را نیز محدود کرده‌اند و عموم افراد نمی‌توانند چیزی را از پلیس مخفی نگه دارند. در نتیجه هر گونه پیاده‌سازی از IP که از روشهای رمزنگاری قوی استفاده می‌کند مجوز صدور از ایالات متحده (و بسیاری از کشورهای دیگر) را نخواهد گرفت. پیاده‌سازی دو نرم‌افزار یکی برای کاربرد داخلی و یکی برای صادرات، موضوعی است که عرضه‌کنندگان صنعت کامپیوتر با آن مخالفند.

موضوعی که پیرامون آن هیچ اختلاف نظر پیش نیامد آن بود که نمی‌توان انتظار داشت صبح روز یکشنبه IPv4 را در اینترنت از کار انداخت و صبح دوشنبه IPv6 را روشن نمود. در عوض مسیریابها و ماشینهایی که به IPv6 مجهز می‌شوند به مثابهٔ جزایر مستقل با استفاده از تونل با یکدیگر مبادلهٔ داده می‌کنند و با افزایش این جزایر، در یکدیگر ادغام شده و جزیرهٔ بزرگتری پدید می‌آید. در نهایت تمام این جزایر به هم می‌پیوندند و اینترنت کاملاً متحول می‌شود.

سرمایه‌گذاری حجیمی که از قبل بر روی مسیریابهای مبتنی بر IPv4 صورت گرفته، فرآیند تغییر و تحول اینترنت را سالها به تأخیر می‌اندازد. به همین دلیل تلاش زیادی صورت می‌گیرد تا این اطمینان حاصل شود که گذار از IPv4 به IPv6 حتی‌الامکان بدون زحمت و گرفتاری انجام گیرد. برای کسب آگاهی بیشتر در خصوص IPv6 به مرجع (Loshin, 1999) مراجعه نمایید.

NAT[1]: ترجمهٔ آدرسهای شبکه

آدرسهای IP کمیاب و ارزشمند هستند: یک ISP ممکن است یک بلوک آدرس با الگوی 16/ (همان کلاس B سابق) و توانایی آدرس‌دهی ۶۵۵۳۴ ماشین میزبان، داشته باشد. اگر تعداد مشتریان این ISP از این تعداد بیشتر شود مشکل بهم می‌زند. برای مشتریان خانگی که از طریق خطوط تلفن متصل می‌شوند، راه حل این مشکل آن است که وقتی مشتری شماره‌گیری کرد و وارد شد به او موقتاً یک آدرس IP پویا اختصاص داده شود و پس از پایان نشست و قطع ارتباط، این آدرس پس گرفته شود. در این روش شبکه‌ای با آدرس 16/ (کلاس B) می‌تواند حداکثر ۶۵۵۳۴ کاربر فعّال داشته باشد که این تعداد حتی برای یک ISP با چند صدهزار مشتری نیز کفایت می‌کند. به محض آنکه یک نشست خاتمه یافت آدرس IP منتسب شدهٔ قبلی، به تماس گیرندهٔ بعدی داده می‌شود. این استراتژی اگرچه برای یک ISP با تعداد متوسطی از کاربران خانگی به خوبی کار می‌کند ولی برای ISPهایی که به مشتریان اداری خدمات می‌دهند مفید نیست.

مشکل از اینجا ناشی می‌شود که مشتریان اداری انتظار دارند که حداقل در ساعات کاری روز خطی دائم و فعال (On-Line) داشته باشند. امروزه، چه دفاتر کوچک اداری مثل یک آژانس مسافرتی با سه کارمند و چه شرکتهای بزرگ که دارای تعداد زیادی کامپیوتر و شبکهٔ محلی هستند، نیاز به خط

---

۱. Network Address Translation

Ramin.samad@yahoo.com

WWW.IRANMEET.COM WWW.IRANMEET.COM

دائم و فعال دارند. برخی از این کامپیوترها، PCکارمندان و برخی دیگر مثلاً سرویس دهنده‌های وب هستند. عموماً در هر LAN یک مسیریاب وجود دارد که از طریق یک خط اجاره‌ای (Leased) به ISP متصل شده است. چنین ساختاری متضمن آن است هر کامپیوتر آدرس IP خود را داشته باشد و به طور روزانه تغییر نکند. در نتیجه، تعداد کل کامپیوترهایی که در اختیار مشتریان اداری است نباید از تعداد آدرسهای IP متعلق به ISP بیشتر شود. برای آدرس 16/، حداکثر تعداد کامپیوترها ۶۵۵۳۴ است. برای یک ISP با دهها هزار مشتری اداری، این فضا سریعاً اشباع می‌شود.

آنچه که مشکل را حادّتر می‌کند آن است که روزبه‌روز بر تعداد مشترکین اینترنت از طریق مودمهای کابلی ADSL افزوده می‌شود. ویژگی چنین سرویسی عبارت است از: (۱) کاربر یک آدرس IP دائم و ثابت می‌گیرد. (۲) هزینهٔ شماره‌گیری و اتصال ندارد (مگر یک هزینهٔ ثابت ماهانه) بدین ترتیب اینگونه کاربران همیشه در شبکه حضور دارند. این موضوع، مشکل کمبود آدرسهای IP را افزایش می‌دهد. در اینجا تخصیص موقت آدرسهای IP (شبیه به مکانیزمی که برای کاربران تلفنی داشتیم) عملی نیست.

حتی از این هم پیچیده‌تر آنکه ممکن است کاربران ADSL و اینترنت کابلی دارای دو یا چند کامپیوتر در خانه باشند و تمام اعضای خانواده از طریق همین خط فعال و مشترک به ISP متصل شوند. یک راه حل آن است که تمام PCها از طریق یک LAN به هم متصل شده و با یک مسیریاب به ISP وصل شوند. از دیدگاه ISP شبکهٔ این خانواده فرقی با یک دفتر اداری کوچک ندارد.

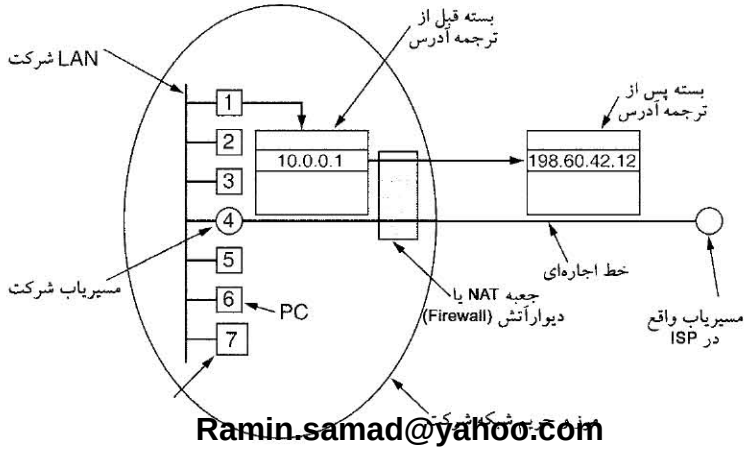
مشکل کمبود آدرسهای IP یک مسئله تئوریک نیست که در آینده‌ای دور رخ بدهد. همین الان با این مشکل مواجه هستیم. راه حل طولانی مدت و همیشگی این مشکل آنست که به سوی IPv6 (که آدرسهای آن ۱۲۸ بیتی است) حرکت نماییم ولیکن گذار از نسخهٔ ۴ به نسخهٔ ۶ به آهستگی صورت می‌گیرد و سالها طول می‌کشد تا این تغییر به طور کامل انجام شود. در نتیجه بسیاری افراد احساس کردند که به یک راه حل سریع و کوتاه مدت نیاز است. این راه حل سریع، NAT (ترجمهٔ آدرس شبکه) است که در RFC 3022 تشریح شده و در ادامه آنرا مختصراً بررسی می‌نماییم. برای کسب آگاهی بیشتر به مرجع (Dutcher, 2001) مراجعه نمایید.

ایدهٔ اصلی در NAT آن است که به هر شرکت یک یا تعداد کمی آدرس IP معتبر و جهانی اختصاص بدهیم. درون این شرکت، هر کامپیوتر دارای یک آدرس IP یکتا است که برای مسیریابی ترافیک داخلی بکار می‌آید. با این حال وقتی بسته‌ای بخواهد شرکت را ترک کرده و به ISP برود باید قبل از خروج ترجمهٔ آدرس صورت بگیرد. برای آنکه این روش ممکن باشد سه محدوده از فضای آدرس IP جهت بکارگیری در شبکه‌های داخلی، به صورت «خصوصی» (Private) تعریف شده است و شرکتها می‌توانند به صورت دلخواه از آنها استفاده کنند. [نیازی به ثبت جهانی آنها نیست.] تنها قانون آن است که هیچ بسته‌ای نباید با چنین آدرسی بر روی اینترنت ظاهر شود. این سه محدودهٔ رزرو شده عبارتند از:

10.0.0.0	- 10.255.255.255/8	(16,777,216 Hosts)
172.16.0.0	- 172.31.255.255/12	(1,048,576 Hosts)
192.168.0.0	- 192.168.255.255/16	(65535 Hosts)

در محدودهٔ اول ۱۶۷۷۷۲۱۶ آدرس (به استثنای ۰ و ۱-) در دسترس است و عموماً اکثر شرکتها از آن استفاده می‌کنند هر چند نیازی به چنین تعداد آدرسی نداشته باشند.

عملکرد NAT در شکل ۵ نشان داده شده است. ماشینها در درون شبکه دارای یک آدرس یکتا به فرم 10.x.y.z هستند. ولیکن وقتی بسته‌ای بخواهد مرز شرکت را ترک کند ابتدا باید از درون یک «جعبهٔ NAT» (NAT BOX) عبور کرده و آدرس مبداء آن با آدرس IP حقیقی شرکت جانشین شود. مثلاً در شکل ۵ آدرس 10.0.0.1 با آدرس 198.60.42.1 عوض شده است. اغلب «جعبهٔ NAT» در یک «دیوار آتش» (Firewall) ادغام می‌شود تا این ابزار ضمن ترجمهٔ آدرس، امنیت شبکه را نیز با نظارت دقیق بر ورود و خروج اطلاعات تضمین نماید. در فصل ۸ مفهوم «دیوار آتش» را بررسی خواهیم کرد. همچنین می‌توان «جعبهٔ NAT» را در مسیریاب شرکت قرار داد. اکثر مسیریابهای امروزی از فرآیند NAT پشتیبانی می‌کنند.



Ramin.samad@yahoo.com

WWW.IRANMEET.COM WWW.IRANMEET.COM

تا اینجا جزئیات کمی از فرآیند NAT مطرح کرده‌ایم: وقتی پاسخ یک بسته بر می‌گردد (مثلاً از سرویس‌دهندهٔ وب) طبعاً آدرس ماشین گیرندهٔ پاسخ، 192.60.42.1 است. سؤال این است که جعبهٔ NAT از کجا بداند که آدرس کدام ماشین داخلی را به جای آن قرار بدهد؟ مسئلهٔ اصلی در NAT همین نکته است. اگر فیلدی اضافی در سرآیند بستهٔ IP وجود داشت می‌شد از آن برای درج آدرس واقعی گیرندهٔ بسته بهره گرفت ولیکن در سرآیند بسته تنها یک بیت بلااستفاده مانده است. همچنین می‌توان یک گزینهٔ جدید [در فضای فیلد اختیاری Option] تعریف کرد تا آدرس حقیقی ماشین مبداء بسته را نگاه دارد ولی انجام این کار مستلزم آن است که کُد نرم‌افزار IP در تمام ماشینها و در کل اینترنت تغییر کند تا گزینهٔ جدید به رسمیت شناخته شده و به درستی تعبیر شود. این راه حل نیز فرآیند زمان‌بری است و مشکل را در کوتاه مدت حل نخواهد کرد.

آنچه که بطور واقعی اتفاق می‌افتد به نحو ذیل است: طراحان NAT بدین نتیجه رسیده بودند که اغلب بسته‌های IP در درون فیلد دادهٔ خود یک بستهٔ TCP یا UDP حمل می‌کنند. هرگاه در فصل ششم TCP و UDP را بررسی کردیم، خواهید دید که هر دوی این پروتکلها دارای سرآیندی برای بسته‌های خود هستند که دو فیلد «**شمارهٔ پورت مبداء**» و «**شمارهٔ پورت مقصد**» جزو آنهاست. در زیر اگرچه تمرکز ما بر پورتهای TCP است ولی همین قضیه برای پورتهای UDP نیز صادق است. این پورتها که اعداد صحیح ۱۶ بیتی هستند، مشخص می‌کنند که اتصال TCP از چه پروسه‌ای شروع و به چه پروسه‌ای ختم می‌شود. این شمارهٔ پورتها فیلدهای لازم برای عملکرد NAT را فراهم آورده‌اند.

هرگاه یک پروسه بخواهد یک اتصال TCP با یک پروسهٔ راه دور برقرار کند یک شمارهٔ پورت بلااستفاده برای خود بر می‌گزیند. این پورت، اصطلاحاً «پورت مبداء» نام دارد و به کد برنامه TCP تفهیم می‌کند که باید بسته‌های ورودی با این شمارهٔ پورت را برای او بفرستد. همچنین هر پروسه یک شمارهٔ پورت مقصد تعیین می‌کند تا مشخص شود که بسته‌ها باید به کدام پروسه در ماشین مقصد تحویل شود. شمارهٔ پورتهای صفر تا ۱۰۲۳ برای سرویس دهنده‌های مشهور رزرو شده است. به عنوان مثال پورت شمارهٔ ۸۰ توسط سرویس دهنده‌های وب بکار گرفته شده است، لذا برنامه‌های مشتری (Client) براحتی با آنها ایجاد ارتباط می‌کنند. کوتاه سخن آنکه، هر پیام خروجی از TCP دارای شمارهٔ پورت مبداء و شماره پورت مقصد است و این دو شمارهٔ پورت هویت پروسه‌های طرفین ارتباط را مشخص می‌نمایند.

تمثیلی از یک نمونه می‌تواند به فهم شماره‌های پورت کمک کند: یک شرکت را در نظر بگیرید که دارای یک شماره تلفن اصلی و واحد است. وقتی افراد با این شماره تماس می‌گیرند بلافاصله اپراتور مربوطه از آنها سؤال می‌کند که کدام شمارهٔ داخلی مدّنظر آنهاست؛ سپس خط داخلی را وصل می‌کند. شمارهٔ اصلی به مثابهٔ آدرس IP است و شماره‌های داخلی، مشابه با شماره پورت هستند. پورتها، ۱۶ بیت آدرس اضافی دیگر هستند که هویت پروسهٔ گیرندهٔ بسته‌ها را مشخص می‌نمایند.

با استفاده از فیلد «شمارهٔ پورت مبداء» می‌توان مشکل نگاشت آدرسها در NAT را حل کرد. هر گاه بسته‌ای برای خروج از شبکه به NAT وارد شود، آدرس مبداء آن که به شکل 10.x.y.z است با آدرس IP حقیقی و معتبر شرکت عوض می‌شود. مضاف بر این فیلد شمارهٔ پورت مبداء (TCP Source Port) با عددی عوض می‌شود که در حقیقت این عدد اندیس جدول ترجمهٔ آدرس در جعبهٔ NAT است. هر یک از درایه‌های این جدول، آدرس IP اصلی و همچنین شمارهٔ پورت واقعی آن بسته را نگه می‌دارند. در آخر کد کشف خطای بستهٔ TCP و بستهٔ IP از نو محاسبه و در بسته قرار داده می‌شود. [چرا که هم فیلد شمارهٔ پورت و هم فیلد آدرس IP مبداء در جعبهٔ NAT عوض می‌شود. -م] عوض کردن مقدار فیلد پورت مبداء (Source Port) الزامی است چراکه ممکن است بطور همزمان از دو ماشین به آدرسهای 10.0.0.1 و 10.0.0.2 یک اتصال TCP اتفاقاً با شمارهٔ پورت مبداء یکسان (مثلاً ۵۰۰۰) ایجاد شود، لذا شمارهٔ پورت مبداء نمی‌تواند هویت واقعی پروسهٔ ارسال کنندهٔ بسته‌ها را مشخص کند.[۱]

وقتی بسته‌ای از طریق ISP به جعبهٔ NAT وارد می‌شود ابتدا مقدار فیلد پورت مبداء استخراج شده و از آن به عنوان اندیس جدول نگاشت در جعبهٔ NAT استفاده می‌شود. پس از پیدا شدن درایهٔ متناظر، آدرس IP داخلی و شمارهٔ اصلی پورت مبداء بسته استخراج شده و در درون بسته قرار می‌گیرد. سپس این بسته از طریق مسیریاب داخلی شرکت، برای تحویل به آدرس 10.x.y.z مسیر طبیعی خود را طی می‌کند.

همچنین از NAT می‌توان برای تخفیف مشکل کمبود آدرس IP برای کاربران ADSL و کاربران کابلی بهره گرفت. هر گاه ISP بخواهد به هر یک از این کاربران آدرسی انتساب بدهد، از آدرسی در فضای 10.x.y.z بهره می‌گیرد. قبل از آنکه بسته‌های ماشین کاربران، ISP را ترک کنند و به اینترنت وارد شوند باید ابتدا وارد جعبهٔ NAT شده و آدرس غیرحقیقی و محلی آنها به آدرس واقعی متعلق به ISP نگاشته شود. در مسیر برگشت، عکس فرآیند نگاشت انجام می‌شود. بدین نحو از دیدگاه اینترنت، این ISP (و کاربران ADSL یا کابلی آن) دقیقاً مثل یک شرکت بزرگ به نظر می‌رسند، هر چند تعداد آدرسهای واقعی و معتبر ISP ناچیز است.

اگرچه این روش مشکل کمبود آدرسهای IP را حل می‌کند ولیکن بسیاری از افراد در جامعه اینترنت از آن به عنوان کاری بی‌ارزش و مردود یاد می‌کنند. برخی از مخالفتهای آنان را به اختصار ارائه می‌نماییم. اول آنکه NAT مدل معماری IP را نقض می‌کند چرا که در این مدل بیان شده که آدرس IP به صورت یکتا ماشینی واحد را در کل جهان مشخص می‌نماید. ساختار تمام نرم‌افزارهای اینترنت با تکیه بر این واقعیت بنیان گذاشته شده است. با

---

۱. بعبارت روشنتر چون تمام بسته‌ها در برگشت آدرس IP یکسانی دارند فلذا این آدرس پورت هر بسته است که هویت گیرندهٔ واقعی بسته را مشخص می‌کند و طبعاً NAT باید در هنگام خروج بسته‌ها ضمن عوض کردن شمارهٔ پورت، یکتا بودن آنرا تضمین کند. -م

Ramin.samad@yahoo.com

WWW.IRANMEET.COM WWW.IRANMEET.COM

NAT ممکن است هزاران ماشین از آدرس 10.0.0.1 استفاده کنند (و می‌کنند).

دوم آنکه NAT اینترنت را از حالت «بدون‌اتصال» به شبکه‌ای «اتصال‌گرا» تبدیل می‌نماید. مسئله اینجاست که جعبهٔ NAT باید اطلاعاتی را در خصوص نگاشت اتصالهایی که از آن می‌گذرند در خود نگاه دارد. نگهداری وضعیت هر اتصال ویژگی شبکه‌های اتصال‌گراست و سنخیتی با شبکه‌های بدون‌اتصال ندارد. اگر جعبهٔ NAT به ناگاه از کار بیفتد و جدول نگاشت آن از دست برود تمام اتصالات TCP برقرار شده از دست می‌رود. بدین ترتیب با وجود NAT، اینترنت به یک شبکهٔ آسیب‌پذیر مدار مجازی تبدیل می‌شود.

سوم آنکه، NAT اصول بنیانی لایه‌بندی پروتکلها را نقض می‌کند: لایهٔ k نباید هیچ تصوری از آنچه که لایهٔ k+1 در فیلد حمل داده از بستهٔ او قرار می‌دهد، داشته باشد یا در آن دخالتی کند.[1] اصل اساسی در معماری لایه‌به‌لایه آنست که تمام لایه‌ها مستقل از دیگری باشند. اگر مثلاً زمانی TCP به نسخهٔ TCP-2 ارتقاء یابد و سرآیند بسته‌ها تغییر کنند (مثلاً شمارهٔ پورتها ۳۲ بیتی شوند)، NAT از کار خواهد افتاد. ایدهٔ اصلی در پروتکلهای لایه‌ای آن بوده که تغییر در یک لایه نیازی به تغییر در لایه‌های دیگر نداشته باشد در حالیکه NAT این عدم وابستگی را از بین می‌برد.

چهارم آنکه پروسه‌های اینترنت مجبور به استفاده از TCP یا UDP نیستند. اگر فرضاً کاربری بر روی ماشین A تصمیم بگیرد برای محاوره و مبادلهٔ داده با کاربری بر روی ماشین B از یک پروتکل جدید در لایهٔ انتقال استفاده کند (مثلاً برای کاربردهای چند رسانه‌ای)، وجود NAT منجر به عدم کارکرد آن برنامهٔ کاربردی خواهد شد چرا که NAT نخواهد توانست فیلد پورت مبداء را به درستی پیدا کرده و از آن استفاده نماید.

پنجم آنکه برخی از برنامه‌های کاربردی آدرس IP ماشین خود را در متن اطلاعات ارسالی قرار می‌دهند. گیرنده نیز این آدرس را استخراج کرده و از آن در جایی استفاده می‌کند. از آنجایی که NAT چیزی در مورد این آدرسهای مخفی نمی‌داند فلذا قادر به تغییر آنها نبوده و هرگونه تلاش برای استفاده از این آدرسهای ناصحیح (در ماشین راه دور) با شکست مواجه می‌شود. FTP، یعنی استاندارد انتقال فایل در اینترنت به همین ترتیب عمل می‌کند و با وجود NAT از کار می‌افتد مگر آنکه اقدامات احتیاطی خاصی به عمل آید. به دلیل مشابه، پروتکل H.323 که برای تلفن اینترنتی کاربرد دارد (و در فصل هفتم به معرفی آن خواهیم پرداخت) با وجود NAT کار نخواهد کرد. البته می‌توان با تغییرات اصلاحی در NAT آن را بکار گرفت ولی این که با معرفی هر برنامهٔ کاربردی جدید مجبور به اصلاح NAT شویم، اصلاً ایدهٔ جالبی نیست.

ششم آنکه چون فیلد آدرس پورت مبداء، ۱۶ بیتی است، حداکثر ۶۵۵۳۶ ماشین را می‌توان به یک آدرس IP واحد نگاشت. این تعداد حقیقتاً مقدار کمی است (گذشته از آن، ۴۰۹۶ شمارهٔ پورت نیز برای کاربردهای خاص کنار گذاشته شده‌اند)، ولیکن اگر تعداد آدرسهای IP معتبر و جهانی که در اختیار ISP قرار دارد، بیش از یکی باشد به ازای هر یک می‌توان ۶۱۴۴۰ ماشین را با آدرسهای غیرحقیقی مدیریت کرد.

این مشکلات بهمراه مسائل دیگر NAT، در RFC 2993 تشریح شده است. عموماً مخالفین NAT می‌گویند که با حل نازیبا و موقتی مسئله کمبود آدرسهای IP، اصرار بر روی راه حل واقعی و نهایی که همانا رفتن به طرف IPv6 است، کم می‌شود و تعویق انداختن در این تغییر و تحول اصلاً خوب نیست!

---

۱. بعبارتی از دیدگاه لایهٔ k آنچه که از لایهٔ k+1 می‌رسد تعدادی بایت خام است. م

Ramin.samad@yahoo.com