

۱) سرویس دهنده نامهای موزه^۱

در فصول گذشته به کرات صحبت از آدرسهای IP شد و کلاسهای آدرس و الگوهای زیر شبکه، مورد بررسی قرار گرفتند و لیکن آدرسهایی که در دنیای واقعی بعنوان آدرسهای اینترنت دیده می‌شود، نمادی متفاوت و شبیه به مثالهای زیر دارد:

www.ibm.com

cs.ucsb.edu

آنچه مسلم است از آدرسهای با قالب و ساختار بالا فقط بخاطر راحتی کاربرد و سادگی در بخاطر سپردن استفاده شده است و قطعاً در هنگام برقراری یک ارتباط و مبادله داده، این آدرسهای نمادین بایستی به معادل عددی آن (یعنی عدد ۳۲ بیتی آدرس IP) ترجمه شود؛ به این آدرسهای نمادین "نام حوزه" گفته می‌شود. بخاطر داشته باشید که آدرسهای نمادین فوق می‌تواند با مقدار آدرس IP آن تعویض شود؛ یعنی آدرسهای زیر نیز در دنیای اینترنت معتبرند ولی به دلیل سختی در به خاطر سپردن و استفاده از آنها، به ندرت استفاده می‌شود:

alavi@203.41.3.81

http://131.11.70.14

حال سوال اینجاست که وقتی ماشینی تمایل دارد داده‌هایی را به ماشین دیگری با آدرس:

cs.rayan.ir

بفرستد چگونه می‌تواند قبل از ارسال پیام، آدرس IP معادل با این آدرس نمادین را پیدا کند در حالی که دهها میلیون از این گونه آدرسها روی اینترنت تعریف شده اند.

به غیر از آدرسهای نمادین همانند مثالهای فوق، "شبکه‌های خودمختار" میتوانند برای ماشینهای خود از روشهای نامگذاری اختصاصی استفاده کنند. در این گونه شبکه‌ها یک ماشین می‌تواند دارای دو نام متفاوت باشد، مثل:

kiti.cs.ucsb.edu

hpss_kiti_510

آدرس نمادین اول در کل شبکه اینترنت معتبر است ولی نام دوم فقط درون شبکه‌ای قابل شناسایی است که آن ماشین متعلق به آن است.

^۱ Domain Name System

در سالهای نخستین راه اندازی شبکه ARPANet راه حلی بسیار ساده برای ترجمه نامهای نمادین به آدرس IP وجود داشت و آن تعریف تمام نامها و آدرسهای IP معادل ، در یک فایل بنام hosts.txt بود. در آن تاریخ تعداد ماشینهای میزبان زیاد نبود و حجم چنین فایللی ، چندان بزرگ نمی شد. هر ماشین میزبان ساعت 24.00 هر شب این فایل را تازه سازی و به روز می کرد تا هر گونه تغییر احتمالی و تعریف آدرسهای جدید اعمال شود. تابع مترجم نام روی هر ماشین میزبان برای ترجمه یک نام نمادین مستقیماً به این فایل مراجعه می کرد و معادل IP آن را استخراج کرده و به برنامه کاربردی برمی گرداند.

بدیهی است که داشتن یک فایل متمرکز و قرار دادن تمام آدرسها و معادل آدرس IP در آن ، امروزه با حجم میلیونی آدرسها در اینترنت امکان پذیر نیست و سالهاست از روشی برای تبدیل آدرسهای نمادین به آدرسهای IP استفاده می شود که DNS نام دارد. (امروزه روش متمرکز فقط برای شبکه های داخلی و کوچک و برای سیستمهای نامگذاری خاص استفاده میشود)

DNS یا "سیستم نامگذاری حوزه" ، روشی سلسله مراتبی است که بانک اطلاعاتی مربوط به نامهای نمادین و معادل IP آنها را روی کل شبکه اینترنت توزیع کرده است و هر ایستگاه می تواند در یک روال منظم و سلسله مراتبی آدرس IP معادل با ایستگاه مورد نظرش را در نقطه ای از شبکه پیدا کند؛ این سیستم در سال ۱۹۸۴ معرفی شد. در DNS ، کل آدرسهای اینترنت درون بانکهای اطلاعاتی توزیع شده ای هستند که هیچ تمرکزی روی نقطه ای خاص از شبکه ندارند. روش ترجمه نام بدین صورت است که وقتی یک برنامه کاربردی مجبور است برای برقراری یک ارتباط ، معادل آدرس IP از یک ماشین با نامی مثل cs.ucsب.edu را بدست بیاورد ، قبل از هر کاری یک تابع کتابخانه ای^۱ را صدا می زند؛ به این تابع کتابخانه ای "تابع تحلیلگر نام"^۲ گفته می شود. تابع تحلیلگر نام ، یک آدرس نمادین را که بایستی ترجمه شود ، بعنوان پارامتر ورودی پذیرفته و سپس یک بسته درخواست^۳ به روش UDP تولید کرده و به آدرس یک سرویس دهنده DNS (که به صورت پیش فرض مشخص می باشد) ، ارسال می کند. همه ماشینهای میزبان ، حداقل باید آدرس IP از یک سرویس دهنده

^۱ Library Function
^۲ Name Resolver
^۳ Query Packet

DNS را در اختیار داشته باشند. این "سرویس دهنده محلی"^۱ پس از جستجو، آدرس IP معادل با یک نام نمادین را بر می گرداند. "تابع تحلیگر نام" نیز آن آدرس IP را به برنامه کاربردی تحویل می دهد. با پیدا شدن آدرس IP، برنامه کاربردی می تواند عملیات مورد نظرش را ادامه بدهد.

حال باید روشهای جستجو را در سیستم DNS بررسی کنیم:

همانگونه که اشاره شد بانک اطلاعاتی که اسامی اینترنت را با معادل IP آنها در خود دارد متمرکز نیست بلکه روی کل اینترنت توزیع شده است. حال باید دید اسامی اینترنت چگونه سازماندهی می شود تا نهایتاً بتوان روش جستجو روی یک بانک اطلاعاتی توزیع شده را توضیح داد. اسامی نمادین زیر را در نظر بگیرید:

www.president.ir

www.bristol.edu

khatami@president.ir

elvis@hware.cs.mit.edu

بدیهی است که نامهای حوزه همانند مثالهای بالا بدون مسمی و دلیل انتخاب نمی شوند بلکه اطلاعاتی ارزشمند برای جستجو در بانک اطلاعاتی توزیع شده نامهای نمادین در خود دارند. بگونه ای که مشهود است یک نام حوزه از چند بخش مجزا که با علامت "." از هم تفکیک شده، تشکیل می شود. هر کدام از این بخشها که "سطح" نام دارد به یک قسمت از بانک اطلاعاتی توزیع شده اشاره می نماید که به محدودتر شدن فضای جستجو کمک می کند.^۲

برای تحلیل یک نام حوزه، سطوح از سمت راست به چپ تفکیک می شوند و در یک روند سلسله مراتبی، سرویس دهنده متناظر با آن سطح پیدا می شود.

فعالاً از بالاترین سطح که در سمت راست نام حوزه قرار می گیرد (مثل com)، .edu، .net و...) شروع می کنیم. نامهای حوزه به هفت منطقه عمومی و حدود صد و اندی منطقه کشوری تقسیم بندی شده است. حوزه بدین معناست که شما با یک

^۱ Local DNS Server

^۲ نامهای حوزه شباهت ویژه ای به سیستم سلسله مراتبی ذخیره سازی فایلها روی یک ماشین دارد؛ به عنوان مثال اگر شما تمام فایلها را فقط روی یک شاخه ذخیره می کردید جستجو و یافتن یک فایل از بین آنها، کاری بسیار پر زحمت و وقتگیر می شد درحالی که تقسیم فضای دیسک به شاخه های متعدد و دسته بندی و ذخیره فایلها روی شاخه های متفاوت این مشکل را حل می کند.

نگاه ساده به انتهای آدرس نمادین ، می‌توانید ماهیت آن نام و سرویس دهنده متناظر با آن را حدس بزنید. یعنی اگر انتهای نامهای حوزه متفاوت باشد منطقه جستجو برای یافتن آدرس IP معادل نیز متفاوت خواهد بود.

هفت حوزه عمومی که همه آنها سه حرفی هستند عبارتند از:

◀ **.com** . صاحب این نام جزو موسسات اقتصادی و تجاری به شمار می‌آید.^۱

www.sony.com

◀ **.edu** . صاحب این نام جزو موسسات علمی یا دانشگاهی به شمار می‌آید.^۲

www.sharif.edu

◀ **.gov** . این مجموعه از نامها برای آژانسهای دولتی آمریکا اختصاص داده شده است.^۳

www.whitehouse.gov

◀ **.int** . صاحب این نام یکی از سازمانهای بین المللی (مثل یونسکو ، یونیسف ، فائو و ...) محسوب می‌شود.^۴

www.unicef.int

◀ **.mil** . صاحب این نام یکی از سازمانهای نظامی دنیا به شمار می‌آید.^۵

◀ **.net** . صاحب نام جزو یکی از "ارائه‌دهندگان خدمات شبکه"^۶ به شمار می‌رود.

www.pegah.net

◀ **.org** . صاحب نام جزو یکی سازمانهای عام‌المنفعه و غیرانتفاعی محسوب می‌شوند.^۷

www.ieee.org

بنابراین اگر در انتهای نام یک آدرس یکی از حالت‌های هفتگانه فوق را دیدید می‌توانید سریعاً به ماهیت آن آدرس و منطقه جستجو پی ببرید.

نامهای حوزه بسیار زیادی در اینترنت تعریف شده اند که هیچیک از حوزه‌های سه حرفی هفتگانه را در انتهای آنها نمی‌بینید. معمولاً در انتهای این آدرسها یک رشته

^۱ مخفف commercial

^۲ مخفف educational

^۳ مخفف government

^۴ مخفف international

^۵ مخفف military

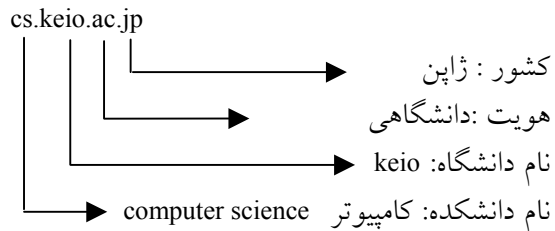
^۶ مخفف Network Service Provider

^۷ مخفف organization

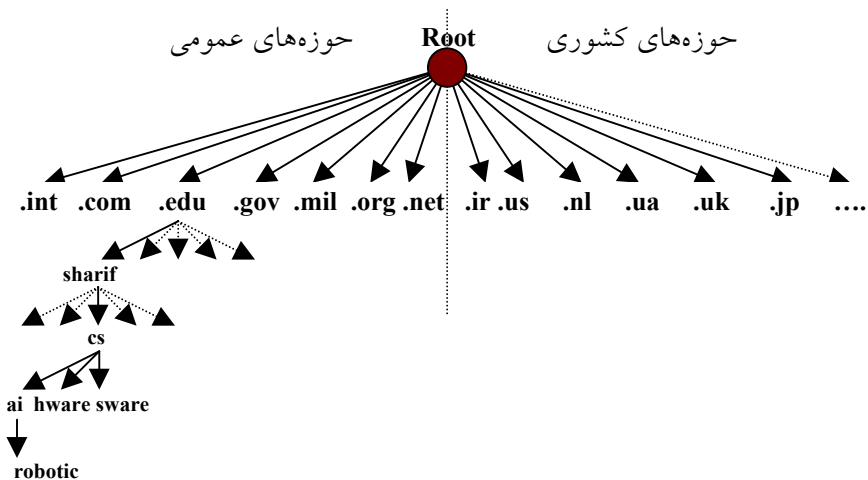
دو حرفی مثل **.ir** یا **.nl** قرار گرفته است. این رشته دو حرفی مخفف نام کشوری است که آن آدرس و ماشین صاحب آن نام، در آن کشور واقع است. مثل:

.ca ←	کانادا	.ir ←	ایران
.jp ←	ژاپن	.nl ←	هلند
.us ←	امارات متحده	.us ←	ایالات متحده

هر حوزه می‌تواند به زیر حوزه‌های کوچکتری تقسیم شود. بعنوان مثال نامهای مربوط به حوزه ژاپن با مخفف **.jp** به دو حوزه کوچکتر تقسیم می‌شود: **.ac.jp** و **.co.jp** که اولی یک مؤسسه علمی و دانشگاهی و دومی یک مؤسسه بازرگانی یا تجاری را در ژاپن تعیین می‌نماید؛ یعنی محل جستجو برای ترجمه یک نام متفاوت خواهد بود. بعنوان مثال آدرس زیر بسادگی قابل تحلیل است:



برای سادگی در درک حوزه و زیرحوزه به نمودار زیر دقت کنید:



در شکل قبل چگونگی شکسته شدن یک آدرس به زیرحوزه‌های کوچکتر به تصویر کشیده شده است.

با این ساختار برای ترجمه یک نام حوزه مثل `robotic.ai.cs.sharif.edu`، عملیات از فایلی به نام "ریشه" شروع می‌شود؛ سپس آدرس ماشینی که فایل راهنمای `edu` در آنجا واقع شده بدست می‌آید. با مراجعه به چنین فایلی مجدداً آدرس ماشینی که فایل راهنمای `sharif.edu` در آنجا قرار دارد به دست می‌آید؛ این روند تا رسیدن به آدرس IP معادل ادامه می‌یابد. این عملیات در چند مرحله محدود تکرار می‌شود و با توجه به آنکه از پروتکل UDP استفاده می‌شود، تاخیر بحرانی نخواهد داشت.

دقت کنید شما نمی‌توانید هر نام دلخواه را برای شرکت یا سازمان خودتان انتخاب نمایید بلکه برای اینکار باید نام مورد نظر را ثبت^۱ نمایید؛ در غیر این صورت چنین آدرسی در اینترنت هویت نخواهد داشت. برای ثبت آدرس باید به سایتهای ثبت‌کننده نام حوزه مراجعه کرده و تقاضای ثبت آدرس نمایید.^۲ این مؤسسات ضمن ثبت و درج نام در بانک اطلاعاتی یکی از حوزه‌های سطح بالا، تضمین خواهند کرد که نام انتخابیتان در کل شبکه اینترنت منحصر بفرد باشد.

۱۲) روشهای جستجو در سرویس دهنده‌های نام

همانگونه که اشاره شد اسامی نمادین در شبکه اینترنت که خود در قالب حوزه‌ها و زیرحوزه‌ها سازماندهی شده اند در یک فایل متمرکز ذخیره نمی‌شوند بلکه روی کل شبکه اینترنت توزیع شده اند، به همین دلیل برای ترجمه یک نام به آدرس IP ممکن است چندین مرحله "پرس و جو" صورت بگیرد تا یک آدرس پیدا شود. طبیعی است که یک پرس و جو برای تبدیل یک نام حوزه همیشه موفقیت آمیز نباشد و ممکن است به پرس و جوهای بیشتری نیاز شود یا حتی ممکن است یک آدرس نمادین اشتباه باشد و هیچ معادل IP نداشته باشد.

سه روش برای پرس و جوی نام در سرویس‌دهنده‌های نام وجود دارد:

← پرس و جوی تکراری^۳

^۱ Register

^۲ سایتهای ثبت نام حوزه، متعددند ولی مشهورترین آنها www.internic.net می‌باشد که هزینه‌ای را نیز دریافت

می‌کند.

^۳ Iterative Query

◀ پرس و جوی بازگشتی^۱

◀ پرس و جوی معکوس^۲

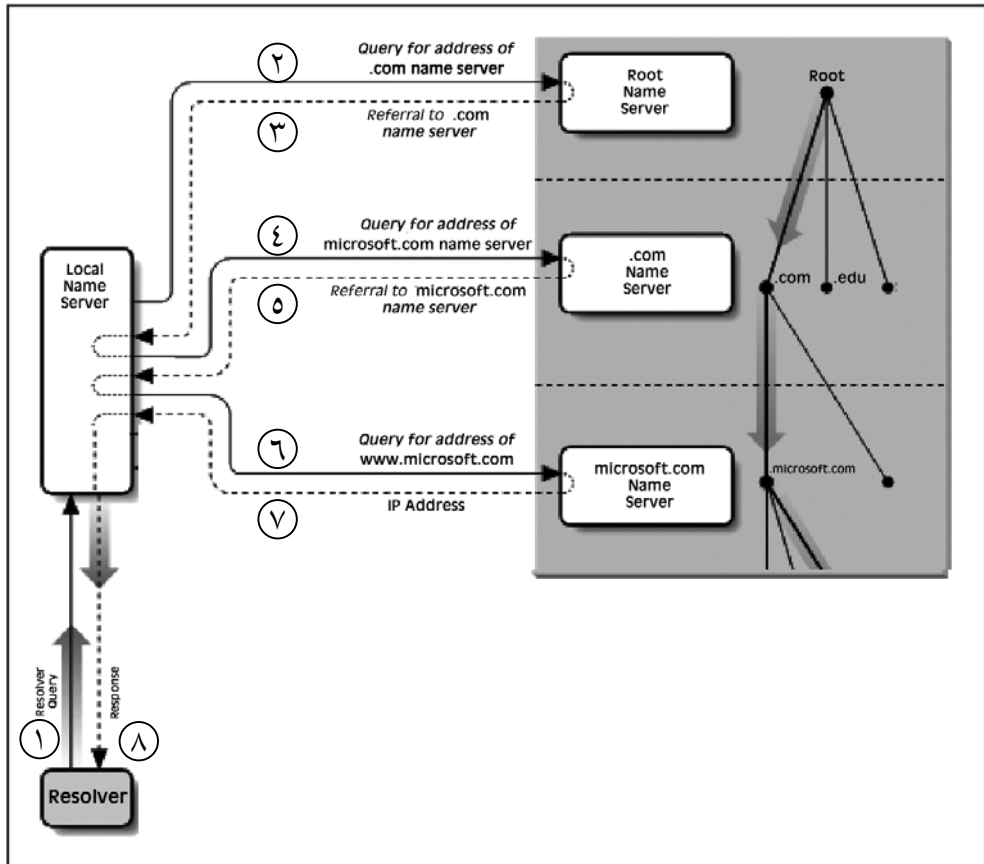
این سه روش را بررسی می‌کنیم:

۲-۱) پرس و جوی تکراری

در پرس و جوی تکراری قسمت اعظم تلاش برای تبدیل یک نام بر عهده سرویس دهنده محلی است؛ این DNS حداقل به آدرس ماشین Root، به عنوان نقطه شروع نیاز دارد. وقتی یک تقاضای ترجمه آدرس به سرویس دهنده محلی ارسال می‌شود در صورتی که قادر به ترجمه نام به معادل IP آن باشد، معادل آدرس IP نام مورد نظر را به تقاضا کننده برمی‌گرداند. (این حالت وقتی است که سرویس دهنده محلی قبلاً آن نام را ترجمه و در یک فایل ذخیره کرده باشد). در غیر این صورت سرویس دهنده محلی خودش یک تقاضا برای DNS سطح بالا ارسال می‌کند. این سرویس دهنده، آدرس ماشینی را که می‌تواند برای ترجمه نام مورد نظر مفید باشد، به سرویس دهنده محلی معرفی می‌کند؛ سرویس دهنده محلی مجدداً یک تقاضا به ماشین معرفی شده در مرحله قبل ارسال می‌کند. در این حالت هم سرویس دهنده نام می‌تواند در صورت یافتن آدرس IP معادل با آن نام حوزه، آنرا ترجمه کند و یا آنکه آدرس سرویس دهنده سطح پایتتری را به او برگرداند. این روند ادامه می‌یابد تا DNS نهائی نام مورد نظر را به آدرس IP ترجمه نماید. برای درک بهتر از روند کار به شکل (۱-۶) دقت کنید. در این مثال فرض شده است که یک برنامه کاربردی با فراخوانی "تابع تحلیلگر نام"، تقاضای ترجمه نام `www.microsoft.com` را می‌نماید. مراحل که انجام میشود به شرح ذیل است:

- در مرحله اول برنامه کاربردی با فراخوانی "تابع تحلیل نام"، تقاضای ترجمه آدرس `www.microsoft.com` را برای سرویس دهنده محلی ارسال کرده و منتظر می‌ماند.
- در مرحله دوم، سرویس دهنده محلی از سرویس دهنده Root (که حوزه‌های متفاوت را تفکیک می‌کند) آدرس ماشینی یک DNS که متوالی حوزه `.com` است را سؤال می‌کند.
- در مرحله سوم، آدرس سرویس دهنده مربوط به حوزه `.com` برمی‌گردد.

^۱ Recursive Query
^۲ Reverse Query



شکل (۱-۶) روند ترجمه نام به روش پرس و جوی تکراری

- در مرحله چهارم، سرویس دهنده محلی، از ماشین معرفی شده در مرحله قبلی، آدرس سرویس دهنده مربوط به حوزه `microsoft.com` را سؤال می‌نماید.
- در مرحله پنجم فهرستی از سرویس دهنده‌های DNS مربوط به `microsoft.com` برمی‌گردد.
- در مرحله ششم، سرویس دهنده محلی تقاضای ترجمه آدرس نمادین `www.microsoft.com` را از DNS متعلق به حوزه `microsoft.com` می‌کند.
- در مرحله هفتم، معادل آدرس IP نام `www.microsoft.com` برمی‌گردد.
- در مرحله هشتم، آدرس IP خواسته شده در اختیار برنامه کاربردی قرار می‌گیرد.

”تابع سیستمی تحلیل نام“ در فصل برنامه‌نویسی تحت شبکه معرفی خواهد شد.

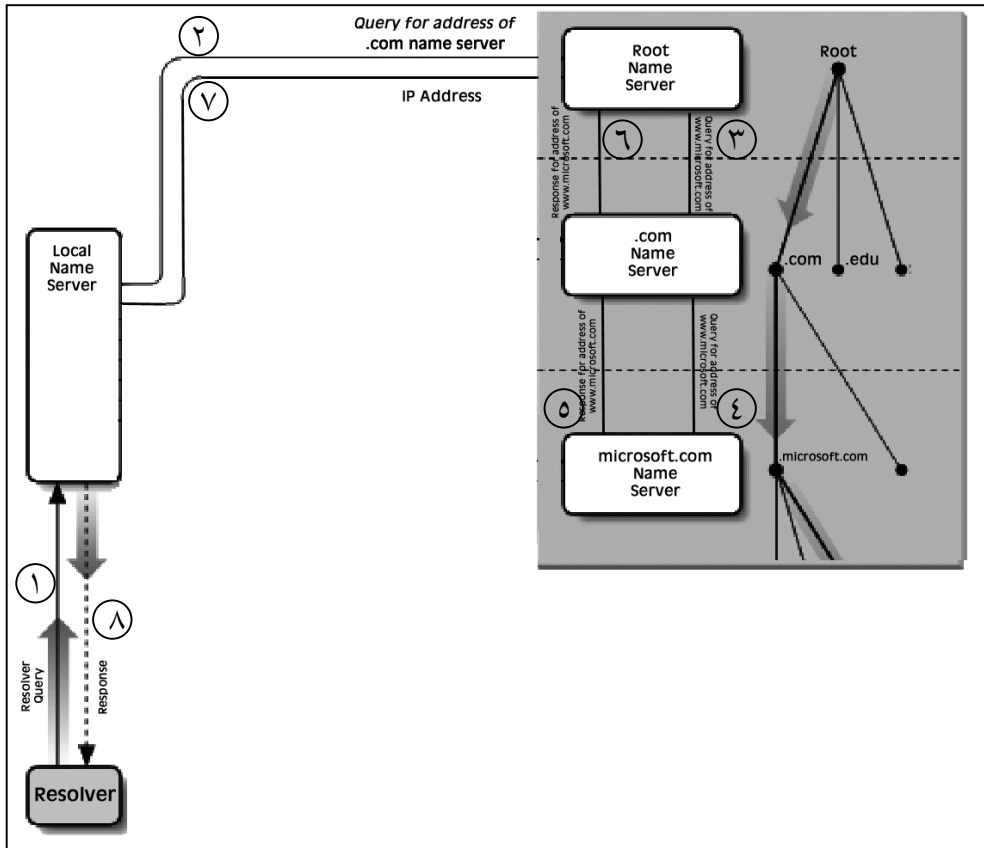
۷-۲) پرس و جوی بازگشتی

در این روش هر گاه برنامه‌ای بخواهد آدرس IP معادل یک نام مثل cs.yale.edu را بدست آورد بگونه‌ای که قبلاً اشاره شد، "تابع سیستمی تحلیل نام" را فراخوانی می‌کند. این تابع یک ماشین را بعنوان سرویس دهنده محلی از قبل می‌شناسد و بنابراین تقاضای تبدیل نام را به روش UDP برای آن ارسال کرده و منتظر جواب می‌ماند (پاسخ نهایی DNS طبیعتاً باید یک آدرس ۳۲ بیتی معادل آدرس IP یک ماشین باشد) دو حالت ممکن است اتفاق بیفتد:

- ممکن است در بانک اطلاعاتی مربوط به سرویس دهنده محلی، آدرس IP معادل با آن نام از قبل وجود داشته و بالطبع به سرعت مقدار معادل IP آن برمی‌گردد.
- ممکن است در بانک اطلاعاتی سرویس دهنده محلی، معادل IP آن نام وجود نداشته باشد. مثلاً سرویس دهنده محلی در بانک اطلاعاتی خودش معادل آدرس IP نام cs.mit.edu را نداشته و طبیعتاً نمی‌تواند آن را ترجمه کند. در چنین حالتی سرویس دهنده محلی موظف است بدون آنکه به تقاضا دهنده خبر بدهد، خودش رأساً به سرویس دهنده سطح بالاتر تقاضای ترجمه آدرس بدهد. در این حالت هم DNS سطح بالاتر بهمین نحو ترجمه آدرس را پیگیری می‌کند یعنی اگر معادل IP آن نام را داشته باشد آنرا برمی‌گرداند و در غیر اینصورت خودش از سرویس دهنده سطح پایینتر تقاضای ترجمه آن نام را می‌نماید و این مراحل تکرار میشود. در روش پرس و جوی بازگشتی ماشین سرویس دهنده محلی این مراحل متوالی را نمی‌بیند و هیچ کاری جز ارسال تقاضای ترجمه یک آدرس برعهده ندارد و پس از ارسال تقاضا برای سرویس دهنده سطح بالا منتظر خواهد ماند. باز هم تکرار می‌کنیم، روشی که DNS برای ترجمه آدرس بکار می‌برد می‌تواند بدون اتصال (UDP) باشد که این کار به سرعت عمل ترجمه آدرس می‌افزاید.

دقت کنید که در روش پرس و جوی تکراری نسبت به روش پرس و جوی بازگشتی، حجم عمده عملیات برعهده سرویس دهنده DNS محلی است و مدیریت خطاها و پیگیری روند کار ساده‌تر خواهد بود و روش منطقی‌تری برای بکارگیری در شبکه اینترنت محسوب می‌شود. روش پرس و جوی بازگشتی برای شبکه‌های کوچک کاربرد دارد.

برای درک بیشتر این روش به شکل (۲-۶) دقت کنید.



شکل (۲-۶) روند ترجمه نام به روش پرس و جوی بازگشتی

۱۲-۳ پرس و جویهای معکوس

فرض کنید حالتی بوجود بیاید که یک سرویس دهنده DNS، آدرس IP یک ماشین را بداند ولی نام نمادین معادل با آن را نداند. بعنوان مثال DNS مایل است بداند که چه نامی در شبکه اینترنت معادل با `195.13.42.7` می باشد. در چنین حالتی مسئله کمی حادتر به نظر می رسد، چرا که برای ترجمه نامهای نمادین، چون این نامها دارای حوزه و زیرحوزه هستند، تحلیل آدرسها ساده است ولی ترجمه آدرس IP به معادل نام حوزه، از چنین روابطی تبعیت نمی کند؛ عبارت بهتر هیچ ارتباط مستقیم و متناظری بین آدرسهای IP و اسامی انتخاب شده در اینترنت وجود ندارد.

برای یافتن نامهای متناظر با یک آدرس IP باید یک جستجوی کامل و در عین حال وقتگیر انجام بشود.

روش کاربدین صورت است که سرویس دهنده محلی یک تقاضا برای DNS متناظر با شبکه‌ای که مشخصه آن در آدرس IP، مشخص شده، ارسال می‌کند.^۱ بعنوان مثال آدرس IP شبکه‌ای را 138.14.7.13 در نظر بگیرید، آدرس کلاس B و مشخصه آن 138.14.0.0 است. زمانی که موسسه‌ای یک کلاس IP ثبت می‌دهد یک سرویس دهنده DNS، متناظر با شبکه خود ایجاد کرده و آنرا نیز معرفی می‌کند. سرویس دهنده محلی بایستی آدرس DNS متناظر با شبکه 138.14.0.0 را پیدا کرده و سپس برای آن یک تقاضا ارسال کند. DNS مربوط به این شبکه، براساس زیرشبکه‌هایی که دارد این سؤال را از طریق سرویس دهنده‌های متناظر با هر زیرشبکه پیگیری می‌کند. (چون هر زیر شبکه یک سرویس دهنده DNS مخصوص به خود دارد) نهایتاً یک نام نمادین حوزه معادل با آن آدرس IP برخواهد گشت.

۱۳) ساختار بانک اطلاعاتی سرویس دهنده‌های نام

یک سرویس دهنده نام در دو قسمت سازماندهی می‌شود:

- پروسه سرویس دهنده: یک برنامه اجرایی است که تقاضاهای ترجمه نام از ماشینهای دیگر را گرفته و پردازش می‌کند و پاسخ مناسب را برای تقاضا دهنده برمی‌گرداند. قالب هر تقاضا در شبکه اینترنت، استاندارد و مشخص است تا هر ماشینی فارغ از ساختار و سیستم عامل، بتواند تقاضا بدهد و پاسخ دریافت کند. قالب استاندارد تقاضا و پاسخ را در بخش بعدی بررسی خواهیم کرد.
- بانک اطلاعاتی: در این بانک اطلاعاتی داده‌های لازم برای تحلیل یک نام نمادین، ذخیره می‌شود. هر سرویس دهنده می‌تواند بنابر روش مورد نظر خود، این بانک اطلاعاتی را ایجاد کرده و از آن استفاده کند. بهمین دلیل ساختار این بانک اطلاعاتی در سرویس دهنده‌های گوناگون، اندکی متفاوت است ولی تقریباً همه آنها از اصول یکسانی پیروی می‌کنند. این بانک اطلاعاتی "بانک رکوردهای منبع"^۲ نام دارد که به اختصار "فایل RR" گفته میشود. برای بالا بردن سرعت جستجو در این بانک

^۱ یعنی فیلد NetID در آدرس IP

^۲ Resource Records

اطلاعاتی، این فایل معمولاً در حافظه اصلی نگهداری می‌شود. (به همین دلیل برای ماشینهای سرویس دهنده نام، حافظه زیاد و سریع پیشنهاد می‌شود.)

بایستی به این نکته دقت داشته باشید که وقتی یک سرویس دهنده محلی نام، از طریق پرس و جوهای تکراری، یک آدرس نمادین را یکبار به آدرس IP ترجمه می‌کند نتیجه را در این فایل نگهداری می‌نماید تا آنکه در تقاضاهای بعدی بتواند از آن استفاده کند و مراحل و تغییر پرس و جو تکرار نشود. البته این عمل اشکالی را خواهد داشت که باید برای آن تمهیدی اندیشیده شود:

اگر نام حوزه و معادل IP آن در یک فایل، ثابت و همیشگی باشد، تغییرات احتمالی و تعویض آدرسها چگونه در این فایل اعمال می‌شود؟ مثلاً هر گاه آدرس `www.sbw.com` و معادل IP آن `197.140.11.3` در فایل RR ذخیره شود، چه تضمینی وجود دارد که چند روز بعد معادل IP آن بدلیل خاصی به `197.140.12.16` تغییر نکند.

به همین دلیل هر رکورد درون فایل RR دارای زمان اعتبار است و پس از انقضای زمان باید از آن فایل حذف شده یا آنکه با پرس و جو مجدد به‌هنگام گردد. البته در فایل RR فقط نامهای نمادین و آدرسهای IP درج نمی‌شود بلکه آیتمهای دیگری هم وجود دارد که به سرویس دهنده نام برای حل بهینه تبدیل آدرسها، کمک می‌کند.

در حقیقت وقتی تقاضای تحلیل یک نام نمادین، به یک سرویس دهنده نام ارسال می‌شود ابتدا درون این فایل، روی تک تک رکوردها جستجو انجام می‌شود و در صورت موفق بودن عمل جستجو، رکورد مربوطه به متقاضی بر خواهد گشت.

فایل RR یک فایل کاملاً متنی است (یعنی به راحتی و با یک ویرایشگر میتوان آن را مشاهده کرد یا تغییر داد). هر رکورد درون این فایل معمولاً دارای پنج فیلد است:

Domain Name	Time to live	Class	Type	Value
-------------	--------------	-------	------	-------

به گونه‌ای که اشاره شد چون الزامی به استاندارد بودن این فایل وجود نداشته در برخی از سرویس دهنده‌های نام، ساختار هر رکورد بصورت زیر سازماندهی شده است:

Domain Name	Type	Class	Time to Live	Length	Value
-------------	------	-------	--------------	--------	-------

◀ **Domain Name**: در این قسمت نام حوزه یا نام مربوط به یک ماشین (نام نمادین) قرار می‌گیرد. دقت کنید که چندین رکورد می‌تواند وجود داشته باشد که نام

نمادین آنها یکسان باشد (چراکه ممکن است بقیه فیلدها متفاوت باشند) بهمین دلیل این فیلد، فیلدی "منحصر به فرد"^۱ نیست.

◀ **Time to Live**: این گزینه نشان می‌دهد که رکورد تا چه مدت (بر حسب ثانیه) معتبر و قابل استناد است. با این گزینه مشکل عنوان شده در ابتدای این بخش حل می‌شود چراکه هر رکورد یک زمان اعتبار دارد که پس از منقضی شدن زمان، باید از فایل RR حذف شده یا آنکه به‌هنگام شود. معمولاً در این فیلد مقدار ۸۶۴۰۰ قرار گیرد که معادل یک شبانه روز بر حسب ثانیه می‌باشد.

◀ **Class**: این فیلد مشخص می‌کند که ماهیت نام نمادین مربوط به چه شبکه‌ای است، چرا که هر سرویس دهنده نام می‌تواند به غیر تعریف اسامی مبتنی بر شبکه اینترنت، روش نامگذاری خاص خود را در شبکه محلی خود نیز اعمال کند. اگر رکوردی مربوط به یک نام در شبکه اینترنت باشد، در این فیلد رشته دوحرفی IN قرار می‌گیرد؛ این مشخصه نشان می‌دهد که این رکورد، در ارتباط با تعریف یک نام حوزه روی شبکه اینترنت است. دو نوع رکورد دیگر با کلاسهای CHAOS و Hesiod نیز تعریف شده‌اند که به آنها نخواهیم پرداخت زیرا به شبکه اینترنت مربوط نمی‌شوند.

◀ **Type**: این فیلد نوع رکورد و معنای آن را مشخص می‌کند. مهمترین مقادیری که در این فیلد قرار می‌گیرد در جدول (۳-۶) فهرست شده است. به گونه‌ای که از جدول مشخص است در این فیلد میتواند یک گزینه حرفی یا معادل عددی آن قرار بگیرد ولی برای سادگی ویرایش و تغییر، از گزینه حرفی استفاده میشود. مثال:

```
ns.nic.ddn.mil 99999999 IN A 192.112.36.4
```

برای آنکه راحتتر بتوانیم این گزینه‌ها را تعریف کنیم، به قسمتی از یک فایل RR در یک سرویس دهنده نام که در مورد نام حوزه cs.vu.nl رکوردهائی را ذخیره کرده است در جدول (۴-۶) دقت کنید.

لازم به تذکر است هر خطی که در فایل RR با علامت ; شروع شود به عنوان خط توضیح تلقی شده و نادیده گرفته می‌شود. (خطوط خالی و فواصل نیز مهم نیست)

^۱ Unique

<i>Number</i>	<i>Code</i>	<i>Description</i>
1	A	Network address
2	NS	Authoritative name server
3	MD	Mail destination; now replaced by MX
4	MF	Mail forwarder; now replaced by MX
5	CNAME	Canonical alias name
6	SOA	Start of zone authority
7	MB	Mailbox domain name
8	MG	Mailbox member
9	MR	Mail rename domain
10	NULL	Null resource record
11	WKS	Well-known service
12	PTR	Pointer to a domain name
13	HINFO	Host information
14	MINFO	Mailbox information
15	MX	Mail exchange
16	TXT	Text strings
17	RP	Responsible person
18	AFSDB	AFS-type services
19	X.25	X.25 address
20	ISDN	ISDN address
21	RT	Route through

جدول (۳-۶) انواع رکوردها در بانک اطلاعاتی سرویس دهنده نام

```

;Authoritative data for cs.vu.nl
cs.vu.nl. 86400 IN SOA  star boss (952771,7200,2419200,86400)
cs.vu.nl. 86400 IN TXT  "Faculteit wiskunde en informatica"
cs.vu.nl. 86400 IN TXT  "Virje universiteit Amsterdam"
cs.vu.nl. 86400 IN MX   1 zephyr.cs.vu.nl.
cs.vu.nl. 86400 IN MX   2 top .cs.vu.nl.

flits.cs vu.nl. 86400 IN HINFO  SUN UNIX
flits.cs vu.nl. 86400 IN  A      130.37.231.165
flits.cs vu.nl. 86400 IN  A      192.31.231.165
flits.cs vu.nl. 86400 IN  MX     1 flits.cs.vu.nl
flits.cs vu.nl. 86400 IN  MX     2 zephyr .cs.vu.nl
flits.cs vu.nl. 86400 IN  MX     3 top.cs.vu.nl
www.cs.vu.nl. 86400 IN CNAME  star.cs.vu.nl
ftp.cs.vu.nl. 86400 IN CNAME  zephyr.cs. vy.nl

rowboat          IN  A      130.37.56.201
                 IN  MX     1 rowboat
                 IN  MX     2 zephyr

little-sister    IN  A      130.37.62.23
                 IN  HINFO  Mac MacOS

laserjet         IN  A      192.31.231.216
                 IN  HINFO  "HP LaserJet IIISi Proprietary"

```

جدول (۶-۴) قسمتی حقیقی از فایل RR در یک سرویس دهنده نام (در دانشگاه ویرجی هلند)

♦ SOA^۱: یکسری اطلاعات ابتدائی پیرامون "ناحیه آدرس نمادین"^۲، یک شماره سریال، مدیر مسئول (مسئول تعریف اسامی) و مهلت اعتبار، ارائه می‌کند. در حقیقت این رکورد آغاز تمام رکوردهائی است که در مورد یک ناحیه آدرس مثل cs.vu.nl تعریف می‌شود.

; named.hosts files;

;Start Of Authority RR

tpci.com. IN SOA merlin.tpci.com

root.merlin.tpci.com (2 ; Serial number

7200 ;Refresh (2 hrs)

3600 ;Retry (1 hr)

151200 ;Expire (1 week)

86400) ;min TTL

در مثال فوق رکورد SOA، آغاز تعریف تمام رکوردهائی است که به نحوی در حوزه tpci.com تعریف می‌شوند. بعنوان مثال، تمام اسامی که دو سطح آخر نام حوزه آنها tpci.com است، باید در این "ناحیه" تعریف شوند:

artemis.tpci.com. IN A 143.23.25.7

merlin.tpci.com. IN A 143.23.25.9

pepper.tpci.com. IN A 143.23.25.72

ماشینهای سرویس دهنده نام و همچنین ماشین دریافت‌کننده نامه‌های الکترونیکی که با آدرسهای بصورت *person@tpci.com* ارسال میشوند باید در این حوزه تعریف شوند.

♦ A^۳: این نوع رکورد که مهمترین نوع بشمار می‌رود معادل آدرس IP نامی را که در فیلد اول آمده است، تعیین می‌کند. چون یک ماشین می‌تواند بیش از یک آدرس IP داشته باشد لذا به ازای یک آدرس نمادین که در فیلد Domain Name مشخص شده می‌تواند چندین رکورد وجود داشته باشد. ذکر این نکته ضروری است که هرگاه

^۱ Start of Authority

^۲ Zone

^۳ IP Address

جای فیلدی خالی بماند مقدار آن همان مقدار فیلد در سطر قبلی در نظر گرفته می شود.

artemis.tpci.com. 86400 IN A 143.23.25.7

♦ انواع **MD** و **MF** هر دو امروزه بلااستفاده شده اند و بجای آنها از نوع **MX** استفاده میشود. این دو گزینه برای آدرسهای پست الکترونیکی کاربرد دارند.

♦ **MX**^۱: این نوع رکورد برای آن است که یک ماشین تمام نامه‌های الکترونیکی که با آدرسهای مختلف از ماشینهای متفاوتی می‌رسند را دریافت کند. بعنوان مثال فرض کنید که شخصی دارای یک آدرس پست الکترونیکی با نام حوزه cs.vu.nl دارد ولی پس از مدتی به جای دیگری منتقل می‌شود که نام حوزه آن electrobrain.com است ولی تمایل دارد تمام نامه‌های او با نام حوزه electrobrain.com به همان آدرس قبلی او ارسال شود. او از مسئول حوزه com. خواهد خواست که این کار را برای او با ثبت رکورد زیر در فایل RR انجام بدهد:

electrobrain.com 86400 IN MX mailserver.cs.vu.nl

در حقیقت با این رکورد پس از تعریف و ثبت یک آدرس همانند electrobrain.com تمام نامه‌هایی را که با آدرسهای نظیر person@electrobrain.com ارسال می‌شوند، به ماشین دیگری ارجاع داده می‌شود؛ چراکه صاحب این نام (مثلاً) نخواسته است سیستم سرویس دهنده پست الکترونیکی برای خودش ایجاد نماید و آنرا به ماشینی دیگر ارجاع داده است.

♦ **NS**^۲: رکوردهای از این نوع، یک ماشین سرویس دهنده نام، ویژه یک حوزه را معرفی می‌کند. (برای تکرار جستجو توسط تقاضا دهنده) احتمالاً کاربرد حیاتی این رکورد را احساس کرده‌اید چراکه در یک روند پرس و جوی تکراری، وقتی از یک ماشین سرویس دهنده نام، تقاضای تحلیل یک نام حوزه می‌شود، یا معادل IP آن نام را بر می‌گرداند یا آنکه آدرس یک ماشین دیگر که می‌تواند آن نام را تحلیل کند برگشت داده می‌شود. مثلاً در فایل RR از یک سرویس دهنده حوزه com.، وجود رکوردهایی مثل رکورد زیر کاملاً ضروری است:

^۱ Mail Exchange
^۲ Name Server

```
yahoo.com 86400 IN NS ns.yahoo.com
```

```
ns.yahoo.com 86400 IN A 143.231.221.22
```

این دو رکورد بدین معناست که تمام نامهایی که به yahoo.com ختم می‌شوند در ماشینی با آدرس 143.231.221.22 قابل تحلیل هستند. این رکوردها مبنای یک پرس‌وجوی تکراری هستند.

به گونه‌ای که اشاره شد هر سرویس دهنده برای آنکه قادر به شروع عملیات پرس‌وجوی تکراری باشد، باید حداقل آدرس سرویس دهنده‌های ROOT را داشته باشد. بنابراین وجود رکوردهایی مثل رکوردهای زیر در یک فایل RR الزامی است:

```
; servers for the root domain;
```

```
. 99999999 IN NS ns.nic.ddn.mil.
```

```
99999999 IN NS ns.nasa.gov.
```

```
99999999 IN NS ns.internic.net
```

```
; servers by address
```

```
;
```

```
ns.nic.ddn.mil 99999999 IN A 192.112.36.4
```

```
ns.nasa.gov 99999999 IN A 192.52.195.10
```

```
ns.internic.net 99999999 IN A 198.41.0.4
```

(دقت کنید در سرویس‌دهنده نام، حوزه‌ای که با . مشخص می‌شود به معنای بالاترین حوزه نام یا همان ریشه تلقی می‌شود.^{۱)}

◆ **CNAME**^{۲)}: نامهای مستعار و راحتتر را برای یک آدرس تعیین می‌کند. بعنوان مثال شاید کسی حدسی بزند آدرس شبکه دانشگاه کامپیوتر در دانشگاه MIT بصورت cs.mit.edu است ولی نمیداند آدرس اصلی آن به صورت les.mit.edu می‌باشد، که کمتر قابل حدس زدن و بخاطر سپردن است؛ در چنین حالتی با رکورد زیر می‌توان نام cs.mit.edu را بعنوان نامی مشابه با نام اصلی تعریف کرد:

```
cs.mit.edu 86400 IN CNAME les.mit.edu
```

^{۱)} Top Level Domain
^{۲)} Canonical Name

از طرفی اسامی نمادین گاهی طولانی هستند (مثل www.altavista.com) و صاحبان این نامها ترجیح می دهند برای از دست ندادن مراجعین کم حوصله، اسامی کوتاه معادل با نام اصلی داشته باشند. (مثل www.av.com) در چنین حالاتی این نوع از رکورد بکار می آید:

```
www.av.com 86400 IN CNAME www.altavista.com
```

♦ **PTR^۱**: این نوع رکورد هم شبیه قبلی است با این تفاوت که بجای قرار دادن آدرس نمادین معادل با www.av.edu در مثال فوق دقیقاً آدرس IP آنرا قرار می دهد:

```
www.av.com 86400 IN A 203.145.11.121
```

```
www.av.com 86400 IN PTR 203.145.11.121
```

در مجموع استفاده از گزینه CNAME بهتر است چراکه اگر آدرس IP معادل با یک نام عوض شد فقط یک رکورد باید اصلاح شود در حالی که اگر از گزینه PTR استفاده شود تمام رکوردها باید اصلاح شوند.

♦ **HINFO^۲**: رکوردهائی از این دسته، ارزش عملیاتی برای تعیین و تبدیل آدرس ندارند بلکه سیستم عامل و نوع ماشین متناظر با یک آدرس یا نام را تعریف می کند:

```
flits.cs.vu.nl 86400 IN HINFO SUN UNIX
```

یعنی ماشینی با نام flits.cs.vu.nl، دارای سیستم عامل یونیکس و سخت افزار آن سازگار با مینی کامپیوترهای نوع Sun است.

♦ **MINFO**: همانند گزینه قبلی با این نوع رکورد می توان اطلاعاتی در مورد سیستم پست الکترونیکی متناظر با یک حوزه ارائه کرد.

♦ **MG**: با این گزینه مدیر مسئول شبکه می تواند افراد حقیقی را به عنوان دارنده صندوق پستی در یک حوزه تعریف نماید. به عنوان مثال اگر شما یک آدرس پست الکترونیکی با نام ali@refah.com داشته باشید، طبیعتاً باید نام ali در فایل RR از سرویس دهنده refah.com تعریف شده باشد.

^۱ Pointer
^۲ Host Information

♦ **TXT**: این نوع رکورد ارزش عملیاتی ندارد و توضیحاتی را در مورد صاحب این نام و هویت آن ارائه میکند. تفاوتی این رکورد با خطوط توضیح (که با ; شروع می‌شوند) آنست که این رکوردها می‌تواند در اختیار متقاضی آن قرار بگیرد.

cs.mit.edu. 86400 IN TXT "Faculty of computer science"

cs.mit.edu. 86400 IN TXT "Massachusetts Institute of Technology"

♦ **WKS**^۱: با استفاده از این نوع رکورد، گذشته از تعریف یک ماشین و آدرس IP آن، سرویسهای معروفی را که آن ماشین ارائه می‌دهد، معرفی می‌شود.

tpci_sco.tpci.com IN WKS 143.23.1.34. FTP TCP SMTP TELNET

♦ **NULL**: استفاده از این گزینه رکورد را غیرعملیاتی و بی‌ارزش مینماید و برای زمانی کاربرد دارد که ترجیح داده شود بجای حذف، رکوردی تبدیل به یک رکورد بی ارزش شود.

۱-۳) رکوردهای پرس‌وجوی معکوس

بگونه‌ای که اشاره شد هرگاه یک سرویس‌دهنده، آدرس IP یک ماشین را بداند ولی نام نمادین معادل با آن را نداند به پرس‌وجوی معکوس متوسل می‌شود. برای آنکه یک سرویس‌دهنده نام، بتواند نام نمادین معادل با یک آدرس IP را برگرداند باید در رکوردهای موجود در فایل RR به دنبال آن آدرس بگردد. رکوردهائی که در پرس‌وجوی معکوس جستجو می‌شوند باید در اولین فیلد، آدرس IP یک ماشین یا یک شبکه را تعریف کنند (البته در ادامه آن باید رشته IN-ADDR-ARPA درج شده باشد). مثال زیر گویای کل کاری است که این رکوردها برای پرس‌وجوی معکوس انجام می‌دهند:

23.1.45.143.IN-ADDR-ARPA. IN PTR TPCI_HPWS_4.TPCI.COM

1.23.64.147.IN-ADDR-ARPA. IN PTR TPCI_SERVER.MERLIN.COM

3.12.6.123.IN-ADDR-ARPA. IN PTR BEAST.BEAST.COM

23.143.IN-ADDR-ARPA. IN PTR MERLINGATEWAY.MERLIN.COM

^۱ Well-Known Services

۱۴ قالب پیام در سرویس دهنده‌های نام

در بخش قبلی اشاره شد که اگرچه ساختار فایل‌های RR می‌تواند در سرویس دهنده‌های متفاوت اندکی متفاوت باشد ولی قالب "پیامهای پرس و جو"^۱ که بین سرویس دهنده‌ها مبادله می‌شوند باید استاندارد باشد تا تمام این سیستمها بتوانند آنها را "تجزیه و تحلیل"^۲ کرده و اطلاعات لازم را از آن استخراج کنند.

ساختار پیامهایی که بین سرویس دهنده‌های نام مبادله میشود در شکل (۵-۶) به تصویر کشیده شده است.

Header
Question
Answer
Authority
Additional

شکل (۵-۶) ساختار پیامهای سرویس دهنده نام

همانگونه که از شکل مشخص می‌شود هر پیام دارای پنج بخش زیر است که آنها را به ترتیب در بخشهای بعدی توضیح خواهیم داد:

- ◆ بخش سرآیند پیام
- ◆ بخش پرسش
- ◆ بخش پاسخ
- ◆ بخش اطلاعات ناحیه
- ◆ بخش اطلاعات اضافی

هر یک از بخشهای پنج‌گانه فوق دارای چندین زیربخش است که باید عملکرد آنها بررسی شود:

۱۴-۱) فیلدهای بخش سرآیند

در شکل (۶-۶) فیلدهای متفاوت بخش سرآیند از یک پیام نشان داده شده است. عملکرد و معنای هر فیلد به شرح زیر است:

۱۵	۱۴	۱۳	۱۲	۱۱	۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	۰
ID															
QR	Opcode		AA	TC	RD	RA	Unused				RCODE				
QDCOUNT															
ANCOUNT															
NSCOUNT															
ARCOUNT															

شکل (۶-۶) ساختار "بخش سرآیند" از پیامهای سرویس دهنده نام

◀ فیلد **ID**: وقتی یک سرویس دهنده پرسشی را در قالب یک پیام برای سرویس دهنده دیگر ارسال میکند یک شماره منحصر به فرد برای آن برمیگزیند تا بتواند از بین پاسخهای رسیده پاسخ متناظر با هر پرسش را تشخیص بدهد. (تعداد پیامهای پرسش و پاسخ زیاد است)

◀ فیلد **QR**: این فیلد تکبیتی مشخص میکند که پیام از نوع پرسشی (0) است یا از نوع پاسخ (1) می باشد.

◀ فیلد **Opcode**: این فیلد چهاربیتی نوع پیام را تعریف می کند:

- 0000: قالب پیام از نوع پرس و جوی معمولی است.
- 0001: قالب پیام از نوع پرس و جوی معکوس است.
- 0010: قالب پیام از نوع پرس و جو در مورد وضعیت سرویس دهنده است.
- 0011 تا 1111: فعلاً بدون استفاده هستند.

◀ فیلد **AA**: این فیلد تکبیتی که فقط برای پیامهای پاسخ معنا دارد مشخص میکند که محتوای پاسخ توسط سرویس دهنده مسئول و تعریف کننده یک نام صادر شده است (0) یا

آنکه محتوای پاسخ معرفی یک سرویس دهنده دیگر است که میتواند در ترجمه نام کمک نماید.

◀ فیلد **TC**: ۱ بودن این فیلد تکبیتی مشخص میکند که محتوای پاسخ به دلیل زیاد بودن تعداد رکوردهای ارسالی، کامل نیست و تعدادی از رکوردهای پاسخ حذف شده‌اند.

◀ فیلد **RD**: این فیلد تکبیتی که فقط برای پیامهای پرسش معنا دارد از یک سرویس دهنده نام تقاضا میکند که در صورت امکان برای ترجمه یک نام از روش "پرس و جوی بازگشتی" استفاده کرده و نتیجه نهایی را برگرداند. (روش پرس و جوی بازگشتی رادر ابتدای این فصل معرفی کردیم)

◀ فیلد **RA**: این فیلد تکبیتی که فقط برای پیامهای پاسخ معنا دارد به یک سرویس دهنده نام اعلام می‌کند که آیا قادر است پرس و جوی بازگشتی انجام بدهد (1) یا خیر (0).

◀ بیتهای **Unused**: این سه بیت فعلاً هیچ کاربردی نداشته و در تمام پیامها باید صفر شوند.

◀ فیلد **RCODE**: این فیلد چهاربیتی نیز در پیامهای نوع پاسخ کاربرد دارد و نتیجه رسیدگی به یک پیام پرسش را تعیین می‌کند:

- 0000: مشکلی وجود ندارد و پاسخ لازم ضمیمه پیام است.
- 0001: سرویس دهنده موفق نشده است تا نوع تقاضا را تشخیص بدهد.
- 0010: سرویس دهنده فعلاً مشکلی داخلی دارد.
- 0011: نام تقاضا شده وجود ندارد.
- 0100: سرویس دهنده از پاسخ به تقاضای ارسالی ناتوان است و آنرا نمی‌پذیرد.
- 0101: سرویس دهنده از پاسخ به تقاضای ارسالی به دلیل مسائل امنیتی معذور است.
- 0110 تا 1111: تعریف نشده و بلااستفاده هستند.

◀ فیلد **QDCOUNT**: این فیلد ۱۶ بیتی مشخص می‌کند در بخش "تقاضا" از قالب پیام، چه تعداد رکورد پرسش وجود دارد. (به قالب یک پیام در بخش قبلی دقت کنید)

◀ فیلد ANCOUNT: این فیلد ۱۶ بیتی مشخص می‌کند در بخش "پرسش" از قالب پیام، چه تعداد رکورد RR ارسال شده است.

◀ فیلد NSCOUNT: این فیلد ۱۶ بیتی مشخص می‌کند در بخش "اطلاعات ناحیه" از قالب پیام، چه تعداد رکورد ضمیمه شده است. (به قالب یک پیام در بخش قبلی دقت کنید)

◀ فیلد ARCOUNT: این فیلد ۱۶ بیتی مشخص می‌کند در بخش "اطلاعات اضافی" چه تعداد رکورد ضمیمه شده است.

۱۴-۷) فیلدهای بخش پرسش

در شکل (۶-۷) فیلدهای متفاوت در بخش پرسش از قالب یک پیام نشان داده شده است. عملکرد و معنای هر فیلد به شرح زیر است:

۱۵	۱۴	۱۳	۱۲	۱۱	۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	۰
QNAME															
QTYPE															
QCLASS															

شکل (۶-۷) ساختار "بخش پرسش" از پیامهای سرویس دهنده نام

◀ فیلد QNAME: در این فیلد نامی که باید به آدرس IP ترجمه شود و یا اطلاعاتی در مورد آن به دست آید قرار می‌گیرد. این نام در قالب کاراکترهای متوالی ۸ بیتی ذخیره شده و انتهای آن با کد ۱۰ مشخص می‌شود.

◀ فیلد QTYPE: در این فیلد نوع رکورد درخواستی مشخص می‌شود. (مثلاً نوع SOA، HINFO و ...). تمام انواع رکوردها که در جدول (۶-۳) معرفی شده‌اند میتواند تقاضا داده شود؛ فقط دقت کنید که در این فیلد ۱۶ بیتی کد مربوط به نوع رکورد قرار می‌گیرد نه نام نمادین آن. (مثلاً ۱ برای A، ۵ برای CNAME، ۱۳ برای HINFO) نکته مهم در این فیلد آنست که اگر مقدار آن ۲۵۵ باشد پرسش کننده از سرویس دهنده تقاضا می‌کند که تمام

رکوردهای موجود در فایل RR که به نحوی با نام حوزه مشخص شده در فیلد QNAME مربوط است، در پاسخ ارسال شود.

◀ فیلد QCLASS: در این فیلد، کلاس آدرس درخواستی مشخص می‌شود و به گونه‌ای که اشاره شد برای آدرسهای شبکه اینترنت از نوع IN است. (به بخش قبلی مراجعه کنید)

۳-۱۴) فیلدهای تعریف شده در بخشهای پاسخ، اطلاعات نامیه و بخش اطلاعات اضافی

وقتی یک پیام پرسش به یک سرویس دهنده نام ارسال می‌شود، آن سرویس دهنده درون فایل RR به جستجوی رکوردهای متناظر با نام حوزه درخواستی می‌پردازد و در صورتی که رکورد یا رکوردهایی یافت شدند باید تمام آنها برای سؤال کننده ارسال شود. این رکوردها بطور متوالی در یک پیام جاسازی می‌شوند. بار دیگر در شکل (۵-۶) به ساختار پیامهایی که بین سرویس دهنده‌های نام مبادله میشود دقت کنید. سه بخش آخر که فقط در پیامهای پاسخ وجود دارند دارای قالب یکسانی هستند و محل قرارگرفتن رکوردهای پیدا شده در خصوص یک نام حوزه محسوب می‌شود. ساختار این بخشها در شکل (۸-۶) مشخص شده‌اند.

۱۵	۱۴	۱۳	۱۲	۱۱	۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	۰
NAME															
TYPE															
CLASS															
TTL															
RDLENGTH															
RDATA															

شکل (۸-۶) ساختار سه بخش آخر از پیامهای سرویس دهنده نام

◀ فیلد NAME: در این فیلد نام حوزه قرار میگیرد. این نام در قالب کاراکترهای متوالی ۸ بیتی ذخیره شده و آخر نام با کد ۱۰ مشخص می‌شود.

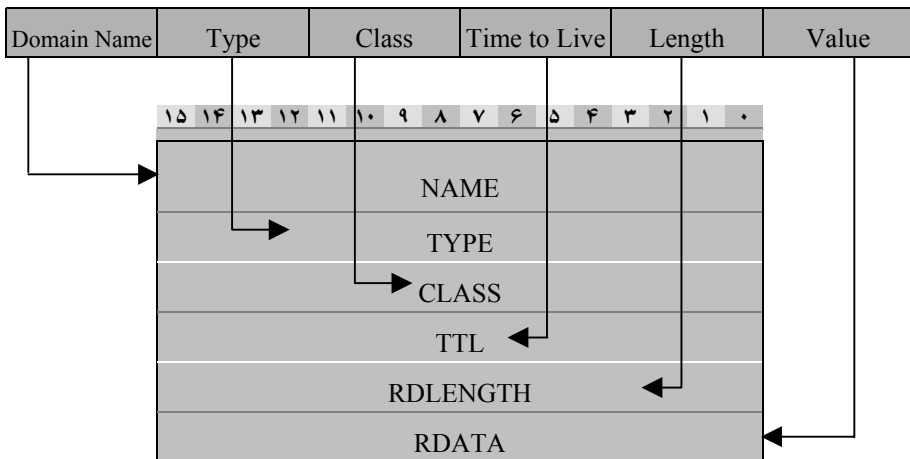
◀ فیلد **TYPE**: در این فیلد نوع رکورد مشخص می‌شود. انواع رکوردها که در جدول (۳-۶) معرفی شده‌اند؛ فقط دقت کنید که در این فیلد ۱۶ بیتی کد مربوط به نوع رکورد قرار می‌گیرد نه نام نمادین آن. (مثلاً ۱ برای A، ۵ برای CNAME، ۱۳ برای HINFO)

◀ فیلد **CLASS**: در این فیلد کلاس نام حوزه مشخص می‌شود و به گونه‌ای که اشاره شد برای آدرسهای شبکه اینترنت از نوع IN است.

◀ فیلد **RDLENGTH**: در این فیلد طول داده‌های موجود در فیلد RDATA بر حسب تعداد کاراکتر مشخص می‌شود.

◀ فیلد **RDATA**: در این فیلد رشته کاراکتری متناظر با نام حوزه قرار می‌گیرد.

برای درک ساده‌تر از فیلدهای فوق مجدداً ساختار رکوردهای فایل RR را در نظر بگیرید. وقتی رکوردی قرار است در پاسخ به یک پرسش ارسال شود طبق شکل (۹-۶) در "بخش پاسخ" از یک پیام جاسازی می‌شود.



شکل (۹-۶) جاسازی یک رکورد در یک پیام ارسالی از سرویس‌دهنده نام

۵) مقدمه‌ای بر مدیریت شبکه

در اوایل پیدایش آرپانت در صورت بروز هرگونه مشکل در شبکه، به علت کم بودن تعداد مسیریابها و ایستگاهها، امکان بررسی همه آنها وجود داشت و با انجام برخی آزمایشهای فیزیکی توسط یک فرد خبره مشکل رفع می‌شد. ولی زمانی که آرپانت به اینترنت جهانی توسعه یافت به علت گستردگی آن، این روش اشکالزدایی امکانپذیر نبود و راهکار بهتری را برای مدیریت شبکه اقتضا می‌کرد. در شبکه‌های قدیمی پروتکل‌های مدیریت و نظارت بر شبکه بخشی از لایه پیوند داده بودند. مدیر شبکه می‌توانست برای مشخص شدن مشکل، مسیریابها را مورد سؤال قرار دهد^۱ و یا مسیرها را آزمایش^۲ کرده و یا تغییر بدهد و همچنین هر یک از واسطه‌های ارتباطی شبکه^۳ را بصورت فیزیکی بررسی نماید و در صورت رفع مشکل، به مسیریابها دستور ادامه عملیات را بدهد. برخلاف شبکه‌های محلی که از سخت‌افزار همگون تشکیل می‌شوند، شبکه اینترنت مبتنی بر TCP/IP دارای یک پروتکل پیوند داده واحد نمی‌باشد و شامل شبکه‌های ناهمگونی است که از طریق "دروازه‌ها"^۴ به هم ارتباط یافته‌اند، بنابراین مدیریت یک شبکه محلی با مدیریت شبکه گسترده‌ای همانند اینترنت کاملاً متفاوت خواهد بود.

نظارت بر وضعیت شبکه و اجزای آن و همچنین توانایی اعمال مدیریت بر روی ماشینهای میزبان و اجزای یک زیرشبکه (شامل مسیریابها، پلها و ...) از ملزومات شبکه اینترنت محسوب می‌شود. در این راستا پروتکل‌های مدیریت شبکه بوجود آمده‌اند. بدلیل ناهمگونی سخت‌افزار ارتباطی شبکه‌ها، لاجرم نرم‌افزارهای مدیریت شبکه باید در لایه کاربرد پیاده‌سازی شوند. پیاده‌سازی نرم افزار مدیریت در لایه کاربرد باعث می‌شود پروتکل‌های مدیریت، مستقل از سخت‌افزار شبکه طراحی گردند که در اینصورت مدیر شبکه می‌تواند با انواع مسیریابها و ماشینهای میزبان به یک روش مشابه ارتباط برقرار نماید و پروتکل مدیریت شبکه برای تمام اجزای آن یکسان و واحد باشد. از معایب پیاده‌سازی نرم افزار مدیریت در لایه کاربرد، آن است که قابلیت اطمینان آن پائین می‌آید چراکه به هنگام بروز هرگونه مشکل پشته

^۱ توسط برنامه‌های کمکی همانند ping

^۲ توسط برنامه‌هایی مثل tracert (مخفف trace route)

^۳ کارت شبکه یا NIC (مخفف Network Interface Card)

^۴ در گذشته به مسیریابهایی که دو شبکه با پروتکل‌های متفاوت را به هم متصل می‌کردند دروازه (Gateway) گفته می‌شد و این اصطلاح هنوز هم بجای مسیریاب استفاده می‌شود.

TCP/IP در یکی از عناصر شبکه، قبل از آنکه بتوان راهی را برای کشف و رفع عیب آن ارائه کرد، نرم افزار مدیریت را از کار خواهد انداخت. در ادامه به معرفی مختصر یکی از پروتکل‌های مدیریت شبکه (SNMP^۱) که در اینترنت مورد استفاده قرار می‌گیرد، خواهیم پرداخت.

۵-۱) معماری پروتکل‌های مدیریت شبکه

بطور کلی مسئله مدیریت شبکه به دو بخش عمده تقسیم شده و برای هر بخش، استاندارد مخصوص به آن تعریف شده است:

بخش اول، تعریف استاندارد مبادله اطلاعات لازم برای نظارت و مدیریت شبکه بین ماشینها و مدیر شبکه می‌باشد. بدین معنا که یک نرم افزار مشتری که در یک ماشین میزبان قرار دارد چگونه با ماشینی که نقش مدیر شبکه را بر عهده دارد، ارتباط برقرار نماید، از آن اطلاعات دریافت کند و یا باعث تغییر اطلاعات آن شود.

بخش دوم، شامل تعریف استاندارد نظارت^۲ و کنترل و همچنین تعریف اطلاعات مدیریتی^۳ می‌باشد. در این بخش از استاندارد معین می‌شود که چه اطلاعاتی برای نظارت و مدیریت شبکه لازم است و این اطلاعات با چه اسامی و چه قالبی باید جمع‌آوری و نگهداری و به‌هنگام‌سازی شوند.

تاکنون برای مدیریت شبکه، راهکارها و استانداردهای مختلفی تعریف شده است. اولین استانداردهای مدیریت در شبکه اینترنت توسط IETF در RFC-1068 و RFC-1028 تعریف شدند که بدلیل نقائص و قابلیت اطمینان پایین، طول عمرشان زیاد نبود. در ماه مه ۱۹۹۰ نسخه اول پروتکل مدیریت ساده شبکه^۴ منتشر شد که بعداً با مشخص شدن عیوب آن، نسخه اصلاح شده و پیشرفته‌تر آن به نام SNMPv2 ارائه گردیده و به عنوان استاندارد مدیریت در شبکه اینترنت مطرح شد. از جمله استانداردهای مدیریت شبکه می‌توان به CMOT^۴ و RMON^۵ نیز اشاره کرد.

^۱ Simple Network Management Protocol

^۲ Monitoring

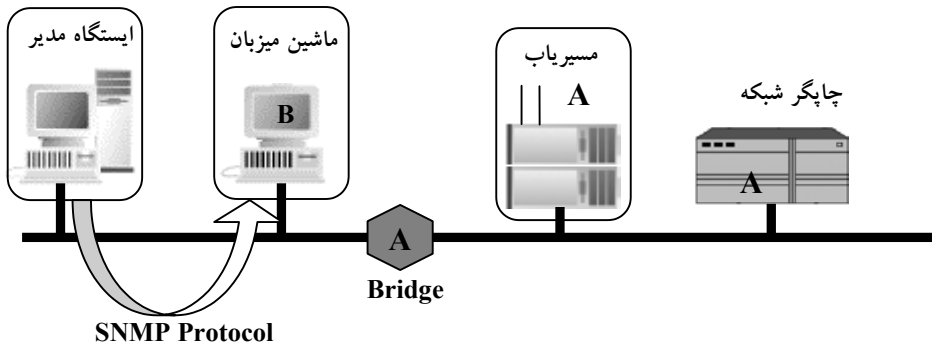
^۳ Management Information

^۴ Common standard Management information protocol Over TCP/IP

^۵ Remote Monitor

۵-۲) مدل SNMP

در مدل SNMP کلیه عناصر یک شبکه خودمختار به چهار رده زیر تقسیم‌بندی می‌شوند. در شکل (۱۰-۶) این تقسیم‌بندی نشان داده شده است:



شکل (۱۰-۶) اجزای مدل مدیریت در SNMP

◀ **نودهای تحت مدیریت**^۱: شامل ماشینهای میزبان، مسیریابها، پلها، چاپگرها و هر ماشین دیگری که بتواند اطلاعاتی از وضعیت خود، به ایستگاههای مدیر ارسال نماید و از فرامین آنها تبعیت کند. یک نود تحت مدیریت باید قادر به اجرای پروسه کاربردی SNMP باشد. در این حالت به آن ایستگاه **نمایندگی SNMP**^۲ گفته می‌شود. هر نود تحت مدیریت ممکن است در کنترل چند ایستگاه مدیریت باشد که هر یک از این ایستگاههای مدیر، سطوح دسترسی متفاوتی به آن ایستگاه دارند.

◀ **ایستگاههای مدیریت**^۳: این ایستگاهها مراکز مدیریت شبکه می‌باشند و معمولاً کامپیوترهای همه‌منظوره‌ای هستند که نرم‌افزار لازم برای مدیریت بر روی آنها نصب شده است. این ایستگاهها با نمایندگیها (نودهای تحت مدیریت) ارتباط برقرار کرده، دستوراتی را صادر و پاسخهایی را دریافت می‌کنند. ممکن است نرم‌افزار مدیریت، دارای رابط گرافیکی^۴ باشد که مسئول شبکه به سادگی وضعیت شبکه را نظارت کند.

^۱ Managed Nodes

^۲ SNMP Agent

^۳ Management Stations

^۴ Graphical User Interface

« اطلاعات مدیریت : هر ایستگاه یک یا چند ”متغیر وضعیت“ را در حافظه سازماندهی و نگهداری می‌کند که این متغیرها وضعیت فعلی آنرا توصیف می‌کنند. در ادبیات پروتکل SNMP ، این متغیرها ”اشیاء“ نامیده شده‌اند.

« قرارداد مدیریت : روشی است استاندارد و مستقل که بر اساس آن ، ایستگاه مدیر با نمایندگیها ارتباط برقرار می‌کند و قادر است حالت اشیاء (متغیرهای وضعیت) آنها را تقاضا کرده و در صورت لزوم آنها را تغییر دهد.

هرگاه در شبکه واقعه پیش‌بینی نشده‌ای رخ بدهد و یکی از نمایندگیها متوجه شود ، وقوع آن را به تمام ایستگاههای مدیریت اطلاع می‌دهد. این گزارش به دلایل تاریخی ”تله“ (TRAP) نامیده می‌شود. از آنجایی که ارتباط ایستگاه مدیریت با گره‌های تحت مدیریت ، قابل اعتماد نیست^۱ ، بهمین دلیل در فواصل زمانی مشخص ، نودهای تحت مدیریت توسط ایستگاه مدیریت سرکشی^۲ می‌شوند؛ به سرکشی ایستگاههای مدیر ”سرکشی مستقیم تله“^۳ اطلاق می‌شود. برای ماشینها و شبکه‌های قدیمی یا هر ماشینی که قادر به اجرای پروسه^۴ SNMP نیستند از ”نماینده‌گی وکالت“^۵ استفاده می‌شود ، یعنی سیستمی که قادر است به هر نحو آنها را مدیریت کند ، باید اطلاعات لازم را از آنها اخذ و سازماندهی کرده و در قالب پروتکل SNMP در اختیار ایستگاههای مدیر قرار دهد.

امنیت و قابلیت اطمینان^۵ در شبکه حساسیت ویژه‌ای دارد؛ از آنجایی که یک ایستگاه مدیر می‌تواند اطلاعات زیادی از گره‌های تحت مدیریت بدست آورده و حتی آنها را از کار بیاندازد ، لذا قبل از برقراری هرگونه ارتباط و مبادله اطلاعات ، هر یک از نمایندگیها باید به نوعی متقاعد گردند که تقاضاها و فرمانها از طرف یک ایستگاه مدیریت مجاز صادر شده است. برای این منظور در SNMPv2 کلمه رمز و روشهای رمزنگاری جدید استفاده می‌شود ، زیرا بدون احراز هویت صادر کننده پیام ، یک کاربر اخلاک‌گر قادر خواهد بود پیامهای مدیریتی را بصورت مصنوعی تولید و منتشر نماید ، یا فرمان از کار افتادن بخشی از اجزای شبکه را صادر نماید.

^۱ عدم اطمینان از آنجایی ناشی می‌شود که هنگام ارسال هر گونه گزارش به ایستگاههای مدیریت ، هیچگونه پیغام تصدیق (Ack) بر نمی‌گردد و یک ایستگاه پس از ارسال اطلاعات مدیریتی هیچ اطلاعی از سرنوشت آنها نخواهد داشت.

^۲ Pulling

^۳ Trap Directed Polling

^۴ Proxy Agent

^۵ Security/Reliability

۳-۵) استانداردهای مدیریت داده

همانطور که اشاره شد، هر نمود تحت مدیریت، باید دارای مجموعه استانداردی از متغیرها برای توصیف وضعیت خود (از قبیل میزان ترافیک ورودی و خروجی، نرخ خرابی بسته‌های داده، وضعیت اجزای مرتبط و ...) باشد. بر این اساس "استاندارد مدیریت داده" در شبکه بوجود آمده است. به مجموعه اطلاعات مدیریتی و ساختار پیاده‌سازی آن "پایگاه داده اطلاعات مدیریتی"^۱ (MIB) اطلاق می‌شود.

دو سازمانی که SNMP و CMOT را ابداع نمودند، یک استاندارد مدیریت داده در شبکه به نام MIB پیشنهاد کرده‌اند که در واقع یک بانک اطلاعاتی جهت ذخیره سازی اشیاء و متغیرها می‌باشد. استاندارد MIB مستقل از پروتکل‌های مدیریت شبکه تعریف شده است که این نکته باعث شده، امکان تغییر پروتکل مدیریت، بدون نیاز به تغییر MIB وجود داشته ولی عیب اساسی آنست که پروتکل‌های مدیریت شبکه باید از اطلاعات مدیریتی یکسان استفاده کنند. MIB دارای ۱۰ گروه از اشیاء است که در جدول (۶-۱۱) فهرست شده‌اند. برای تعریف و کدگذاری استاندارد اشیاء نیز از یک زبان به نام ASN.1^۲ استفاده می‌شود که در ادامه به معرفی آن می‌پردازیم.

نام گروه	شماره شیئی	نوع متغیرهای وضعیتی که هر شیئی نگهداری می‌کند
System	7	Name, location and description of the equipment
Intefaces	23	Network interfaces and their measured traffic
AT	3	Address translation (deprecated)
IP	42	IP packet statistics
ICMP	26	Statistics about ICMP messages received
TCP	19	TCP algorithms, parameters and statistics
UDP	6	UDP traffic statistics
EGP	20	Exterior gateway protocol traffic statistics
Transmission	0	Reserved for media specific MIBs
SNMP	29	SNMP traffic statistics

جدول (۶-۱۱) گروه‌های اشیاء MIB-II در اینترنت

^۱ Management Information Base
^۲ Abstract Syntax Notation 1

۴-۵) زبان توصیفی ASN.1

هسته اصلی پروتکل SNMP، مجموعه‌ای از اشیاء (متغیرهای وضعیت) است که می‌تواند توسط ایستگاههای مدیریت، خوانده یا نوشته شوند. برای آنکه تمام ایستگاههای شبکه با هر گونه اختلاف بنیادی از لحاظ سخت‌افزار و نرم‌افزار، قادر به ارتباط با مدیر شبکه باشند، باید ساختار اشیاء دقیقاً استاندارد باشد. زبان توصیفی ASN.1، استاندارد است که با آن اشیاء و متغیرهای حالت تعریف می‌شوند. همانند زبان C که قادرید طبق یک روال استاندارد متغیرها و استراکچرها را تعریف نمایید، ASN.1 نیز یک روش برای تعریف^۱ متغیرها و اشیاء ارائه کرده است. نکته دوم آنست که روش ارسال این اشیاء روی کانال باید دقیقاً تبیین شود؛ چیزی که در زبان C به آن نیاز نبوده است ولی در SNMP حیاتی است. به عنوان مثال شما نمی‌دانید یک استراکچر با دو متغیر از نوع long، چگونه و به چه ترتیبی روی کانال ارسال می‌شود. (اول کدام بایت از کدام متغیر) یا مثلاً در C نمی‌توانید یک متغیر ۱۳ بیتی تعریف کنید، در حالی که این کار در ASN.1 ممکن است.

اطلاق کلمه شیئی به متغیرهای حالت، پیشنهاد خوبی نیست زیرا در مفهوم شیئی‌گرایی، یک شیئی مجموعه‌ای از متغیرهای حالت و متودهاست، در حالی که در ASN.1، اشیاء هیچ متودی ندارند. اطلاق "متغیر حالت" بجای شیئی مناسبتر است ولی چاره‌ای جز تبعیت از ادبیات استاندارد ASN.1 نداریم.

ASN.1 که بخشی از آن توسط SNMP مورد استفاده قرار می‌گیرد، دارای دو مجموعه استاندارد می‌باشد:

- ◀ یک نوع زبان توصیف اشیاء که توسط کاربر قابل استفاده است.
- ◀ یک روش کدگذاری^۲ برای مبادله اطلاعات بین ایستگاههایی که از پروتکل SNMP پشتیبانی می‌کنند.

استاندارد ASN.1، هرگونه ابهامی را در شکل و محتوای اشیاء از بین می‌برد. به عنوان مثال برای یک شیئی می‌توان نوع و محدوده مقادیر آن را تعریف کرد که این دقت در بیان متغیرها، با توجه به وجود کامپیوترهای متنوع در شبکه اهمیت بیشتری پیدا می‌کند.^۳ همچنین این زبان باعث آسانتر شدن پیاده‌سازی پروتکل‌های مدیریت

^۱ Declaration
^۲ Encoding Rule

^۳ به عنوان مثال یک متغیر صحیح (Integer) در ماشینهای متفاوت می‌تواند طول متفاوتی داشته باشد. (۲ بایت یا ۴ بایت) بنابراین در هنگام تعریف اشیاء به نحوی که فارغ از ساختار ماشین باشند، توصیف یک شیئی باید کاملاً دقیق باشد.

شبکه می‌گردد چراکه براساس زبان ASN.1 داده‌ها و متغیرها در یک پیغام قابل رمزگذاری هستند. (رمزگذاری به معنای تبیین قالب ارسال اشیاء -Encoding-)

با استفاده از این زبان کاربر اشیاء اولیه را تعریف کرده و سپس اشیاء پیچیده‌تری را به کمک آنها ایجاد می‌کند. در حقیقت تعریف اشیاء و متغیرها در ASN.1 همانند تعریف متغیرها و استراکچرها در زبان C البته با نماد و اصول متفاوت می‌باشد. چند قاعده را در نامگذاری اشیاء باید در نظر گرفت:

◀ انواع داده‌ای قبل تعریف شده با حروف بزرگ نوشته می‌شوند. (مثل INTEGER) در جدول (۶-۱۲) برخی از این انواع از پیش تعریف شده فهرست شده است. (در SNMP انواع داده‌ای اعشاری و بولین مورد استفاده نیستند.)

◀ انواع داده که توسط کاربر تعریف می‌شود اگرچه می‌تواند با حروف بزرگ نوشته شود ولی باید حداقل دارای یک حرف غیر بزرگ نیز باشد.

◀ نام شناسه‌ها^۱ با حروف کوچک شروع می‌شود ولی می‌تواند در ادامه حاوی حروف بزرگ نیز باشند.

◀ خطوط توضیح با دو کاراکتر -- شروع شده و تا انتهای خط یا -- بعدی ادامه می‌یابند.

مثلاً در زیر یک متغیر صحیح با محدوده خاص تعریف شده است:

```
Packetsize ::= INTEGER (0..1023)
```

◀ **Object Identifier** روشی برای شناسایی اشیاء است. هر شیئی باید بطور منحصر بفرد قابل شناسایی باشد. مکانیزیم شناسایی اشیاء، استفاده از درخت استانداردها می‌باشد. همانطور که در شکل (۶-۱۳) مشاهده می‌شود در سطح فوقانی این درخت سازمانهای تعیین استاندارد وجود دارند که بخشی از زیردرخت های آن به SNMPv2 اختصاص داده شده است. هر نود در درخت دارای برچسب و کد مخصوص به خود است. برای توصیف اشیاء می‌توان از برچسب یا کدهای درخت ASN استفاده کرد. مثلاً برای نامگذاری متغیر **IpInReceive** که با کد شماره ۳ در زیرگروه **ip** قرار دارد می‌توانیم از روشهای زیر استفاده کنیم:

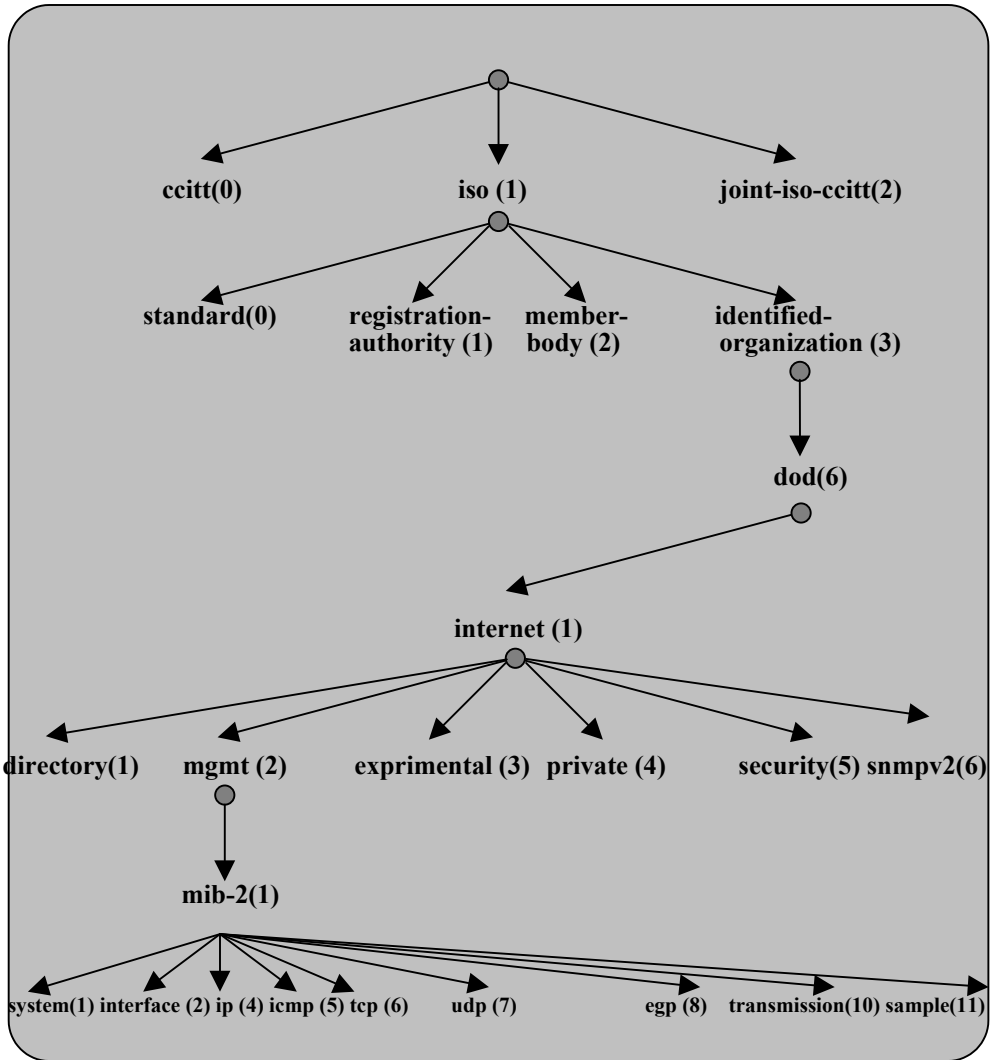
```
{ iso.org.dod.internet.mgmt.mib.ip.IpInReceives }
```

```
یا {1.3.6.1.2.1.4.3}
```

^۱ Identifiers

Primitive type	Meaning	Code
INTEGER	Arbitrary length integer	2
BIT STRING	A string of 0 or more bits	3
OCTET STRING	A string of 0 or more signed bytes	4
NULL	A place holder	5
OBJECT IDENTIFIER	An officially defined data type	6

جدول (۶-۱۲) داده های از پیش تعریف شده در ASN.1 که در SNMP استفاده می شود.



شکل (۶-۱۳) بخشی از درخت نامگذاری اشیاء در ASN.1

مثالی از تعریف یک عدد صحیح و مقداردهی اولیه بر اساس زبان ASN:
 count INTEGER ::= 100
 count متغیری صحیح با مقدار اولیه ۱۰۰ تعریف شده است.

در **ASN.1** چندین روش برای تعریف انواع جدید داده وجود دارد. **SEQUENCE** لیستی مرتب از انواع داده را همانند رکورد در پاسکال بوجود می‌آورد. **SEQUENCE OF** یک آرایه تک‌بُعدی از یک نوع داده خاص را تعریف می‌کند. در اینجا به یک مثال کلی و ذکر چند نکته می‌پردازیم: در گروه **ip** از **MIB** یک شیء به نام **ipAddrTable** تعریف شده است که شامل آرایه‌ای از مشخصه‌های آدرس **IP** می‌باشد و کد ۲۰ به آن تخصیص یافته است. تعریف این شیء بر اساس **ASN.1** بصورت زیر است:

```
IpAddrTable ::= SEQUENCE OF IpAddrEntry
                └──────────┘
                آرایه‌ای از
```

```
IpAddrEntry ::= SEQUENCE {
    ipAdEntAddr      IPAddress,
    ipAdEntIfIndex  INTEGER,
    ipAdEntNetMask  IPAddress,
    ipAdEntBcastAddr IPAddress,
    ipAdEntReasmMaxSize INTEGER(0.. 65535)
}
```

بر طبق تعریف بالا **IpAddrTable** آرایه‌ای از اشیاء **IpAddrEntry** است. شیء **IpAddrEntry** خودش یک رکورد حالت با پنج متغیر می‌باشد. از این پنج متغیر، سه تای آنها از نوع **IPAddress** تعریف شده که عملاً یک عدد صحیح ۴ بایتی بدون علامت است. در **ASN** اینگونه از انواع را می‌توانید به صورت زیر تعریف کنید:

```
IPAddress ::= [APPLICATION 1] INTEGER (0..4294967295)
```

بدین معنا که نوع **IPAddress** با کد شماره ۱، عددی صحیح و ۴ بایتی با محدوده تغییر از ۰ تا $2^{32}-1$ است. از آن به بعد **IPAddress** یک نوع شناخته شده محسوب می‌شود.

در **ASN** برخلاف اغلب زبانها به جای اندیس در آرایه، از پسوند در انتهای نام متغیر استفاده می‌شود.

۵-۵) نمونه انتقال در ASN.1

در ASN.1، روش تبدیل متغیرها و داده‌ها به دنباله‌ای از بایتها، نحوه کدگذاری و نیز دیکود دنباله بایتها، دقیقاً مشخص شده است؛ این روش به نام BER^۱ مشهور است. در این روش، قالب هر شیئی که روی کانال ارسال می‌شود، اعم از مقادیر اولیه و استراکچرها شامل چهار فیلد زیر است:

Identifier	Data Length Field	Data	End-of-Content Flag
------------	-------------------	------	---------------------

◀ شناسه (نوع یا برحسب): این فیلد خود دارای سه قسمت می‌باشد که دو بیت با ارزش آن، نوع شناسه را مشخص می‌کند. فیلد یک بیتی بعدی مشخص می‌کند که نوع متغیر ابتدایی^۲ است (0) و یا ساخته شده^۳ (1) می‌باشد. ۵ بیت باقیمانده برای تعیین شماره نوع متغیر یا شیئی است. شماره انواع متغیر در جدول (۱۲-۶) آمده است. در زیر قالب بایت شناسه یعنی اولین بیتی که در استاندارد ASN.1 روی کانال انتقال می‌یابد، مشخص شده است:

2 Bits	1 Bit	5 Bits
Tag	Type	Number
00 Universal	0 Primitive Type	
01 Application	1 Constructed Type	
10 Context specific		
11 Private		

◀ طول فیلد داده‌ها بر حسب بایت: اگر طول بایتهای داده کمتر از ۱۲۸ باشد مقدار آن در بایتهای که بیت با ارزش آن صفر است نوشته می‌شود. برای داده‌های با طول بیشتر از ۱۲۸، این فیلد چند بیتی است، بطوریکه بیت با ارزش بایت اول و هفت بیت کم ارزش دارای حداکثر مقدار ۱۲۷ می‌باشد. مثلاً اگر طول داده ۱۰۰۰ بایت باشد، ابتدا در بایت اول مقدار ۱۳۰ قرار

^۱ Basic Encoding Rules

^۲ Primitive

^۳ Constructed Type

می‌گیرد تا نشان بدهد دو بایت بعدی دارای مقدار هستند (۲+۱۲۸) سپس در دو بایت بعدی معادل دودویی عدد ۱۰۰۰ قرار می‌گیرد.

◀ **فیلد داده:** داده‌های از نوع صحیح به صورت مکمل ۲ کد می‌شوند. رشته‌های بیتی به همان شکل خودشان کدگذاری و از بیت پرارزش به کم ارزش روی کانال ارسال می‌شوند. چون ممکن است تعداد بیت‌های رشته ضربی از ۸ نباشد، بایت اول که قبل از رشته ارسال می‌شود، تعداد بیت‌های بلااستفاده بایت آخر از رشته بیتی را مشخص می‌کند؛ مثلاً برای رشته ۹ بیتی "۰۱۰۰۱۱۱۱۱"، رشته زیر انتقال می‌یابد:

00000111	01001111	10000000
07	4F	80

برای انتقال شناسه‌های اشیاء از دنباله اعداد صحیح استفاده می‌شود. مثلاً برای تعریف "شیئی اینترنت" با دقت به درخت ASN.1، باید رشته {۶ و ۳ و ۱} ارسال شود.

◀ **نشانهگر انتهای داده:** پایان داده‌ها را مشخص می‌کند. در پروتکل SNMP از این فیلد استفاده نمی‌شود، چراکه فیلد طول داده‌ها حتماً باید ذکر شود و از اینرو دیگر نیازی به نشانهگر انتهای داده نیست. در زیر مثالهایی از نحوه کدگذاری برای داده‌ها و اشیاء دیده می‌شود:

♦ ارسال عدد صحیح ۴۹ با قاعده BER

Integer 49	Tag	Type Code	Length	Value
	000	00010	00000001	00110001

♦ ارسال رشته سه بیتی 110 با قاعده BER

Bit String 110	Tag	Type Code	Length	Value	Value
	000	00011	00000010	00000101	11000000

♦ ارسال رشته دو بیتی "xy" با قاعده BER

String "xy"	Tag	Type Code	Length	Value	Value
	000	00100	00000010	01111000	01111001

♦ ارسال بایت NULL با قاعده BER

Tag	Type Code	Length
	00000101	

NULL

♦ ارسال شیء Internet Object در درخت ASN با قاعده BER

Tag	Type Code	Length	Value	Value	Value
	00000110				

Internet Object

۴-۵) ساختار اطلاعات مدیریتی SMI^۱

با توجه به اینکه متغیرهای وضعیت در یک ایستگاه بسیار متنوعند لذا در SMI زیرمجموعه مهمی از اشیاء در ASN.1، تحت نام ساختار اطلاعات تعریف شده که برای تشکیل ساختمان داده‌های لازم و مقاردهی آنها کاربرد دارد. این استاندارد شامل چهار ماکرو و هشت نوع داده جدید است. مثلاً اگرچه متغیر Counter و متغیر Gauge هر دو یک عدد ۳۲ بیتی هستند ولی یک تفاوت بنیادی دارند: متغیر Counter پس از رسیدن به مقدار $2^{32}-1$ ، صفر خواهد شد ولی متغیر Gauge پس از رسیدن به مقدار $2^{32}-1$ ، دیگر نمی‌شمارد تا زمانیکه یا صفر شود یا از مقدار آن کاسته شود.

برای اینکه پروتکل مدیریت شبکه ساده باشد محدودیتهایی بر روی متغیرهای بانک اطلاعات مدیریتی اعمال شده است. SMI ضمن تعریف انواع داده جدید امکان استفاده از آنها را در تعریف اشیاء جدید فراهم کرده است. مهمتر از همه اینکه SMI نحوه رجوع به جداول مقادیر، مانند جدول مسیریابی را شرح می‌دهد. در جدول (۱۴-۶) انواع داده را که در ساختار مدیریت اطلاعات توسط استاندارد SMI تعریف شده است، مشاهده می‌کنید.

^۱ Structure for Management Information

Name	Type	Bytes	Meaninig
INTEGER	Numeric	4	integre (32 bits in current implementaions
Counter32	Numeric	4	Unsigned 32-bit counter that wraps
Gauge32	Numeric	4	Unsigned value that does not wrap
Integer32	Numeric	4	32 Bits, even on a 64-bit CPU
UInteger32	Numeric	4	Like Inetger 32,but unsigned
Counter64	Numeric	8	A 64 bit counter
TimeTicks	Numeric	4	In hundredths of a second since some epoch
BIT STRING	Numeric	4	Bit map of 1 to 32 bits
OCTET STRING	String	>= 0	Variable length byte string
Opaque	String	>= 0	Obfolete; for backward compatibility only
Object Identifier	String	>0	A list of integers form Fig 3
IpAddress	String	4	A dotted decimal interner address
Nsap Address	String	<22	An OSI NSAP address

شکل (۱۴-۶) انواع داده جدید تعریف شده در SMI برای استفاده در SNMP

۵-۷) پروتکل ساده مدیریت شبکه (SNMP)

به گونه‌ای که اشاره شد مدیریت شبکه درخواستی را به نمایندگی می‌فرستد و اطلاعاتی را از وی خواسته و یا در صدد تغییر حالت آن برمی‌آید. در SNMP حدود ۱۷۵ متغیر حالت و وضعیت تعریف شده که هر یک به نحوی برای نظارت و مدیریت شبکه بکار می‌آید. ارسال داده‌های حالت نیز براساس استاندارد انتقال ASN.1 انجام می‌شود. (البته ممکن است خطاهای متعددی مانند عدم وجود متغیر گزارش گردد).

در یک پروتکل مدیریت شبکه، ممکن است انواع مختلفی از دستورات موجود باشد که این باعث پیچیدگی زیاد می‌شود چراکه برای هر نوع عملیاتی، نیاز به اضافه کردن دستورات جدید خواهد بود. در SNMP از یک روش جالب استفاده می‌شود، به این ترتیب که تمامی عملیات و فرمانها در قالب روش واکنشی و ذخیره متغیرهای حالت انجام می‌شود. مثلاً برای بوت کردن دوباره یک ایستگاه تحت مدیریت، از تغییر مقدار در یک متغیر ویژه استفاده می‌شود. پروتکل SNMP مجموعاً دارای هفت فرمان است. شش فرمان از هفت فرمان موجود در جدول (۱۵-۶) آمده است. فرمان Get-request برای درخواست مقدار یک یا چند متغیر حالت است.

فرمان **get-next-request** امکان درخواست متغیر بعدی را می‌دهد. فرمان **get_bulk_request** برای واکنشی جداول بزرگ است. فرمان **set-request** برای بهنگام‌سازی متغیرهای حالت می‌باشد. با فرمان **inform-request** یک مدیر به مدیر دیگر متغیری را که در حال مدیریت آن است اعلام می‌کند و فرمان **SnmpV2-trap** جهت گزارش یک رخداد از طرف یکی از ایستگاهها به مدیر، کاربرد دارد.

Message	Description
Get-request	Requests the value of one or more variables
Get-next-request	Request the variable following this one
Get-bulk-request	Fetches a large table
Set-request	Updates one or more variables
Inform-request	Manager to manager message describing local MIB
SnmpV2-trap	Agent-to-manager trap report

جدول (۱۵-۶) انواع پیغامهای SNMP

همانطور که گفته شد در ASN برای مشخص کردن اعضای آرایه به جای اندیس از یک پسوند که به دنبال شناسه شیئی می‌آید، استفاده می‌شود. حال مسئله این است که ایستگاه مدیریت باید بتواند عناصر موجود در جدول را واکنشی نماید و برای اینکار باید پسوندها را به دست آورد. دستور **get-next-request** متغیر بعدی در جدول را براساس نام متغیری که به آن ارسال می‌شود، واکنشی می‌کند و بطور پیش‌فرض در صورتیکه نام متغیر بدون پسوند فرستاده شود عنصر اول در جدول واکنشی می‌شود و اگر نام متغیر بطور کامل ارسال شود، عنصر بعدی موجود در جدول را برمی‌گرداند. پروسهٔ SNMP در ایستگاه تحت مدیریت، بر اساس کد ارسال شده، اولین متغیر حالتی را که کد آن از لحاظ ترتیب در جدول بزرگتر از کد موردنظر است، برمی‌گرداند. با این روش می‌توان عناصر یک جدول را به ترتیب واکنشی کرد.

پیغامهای SNMP دارای طول مشخصی می‌باشند و رمزگشایی آنها توسط افراد عادی ساده نیست. یک پیغام SNMP شامل سه بخش مهم می‌باشد؛ شماره نسخهٔ

پروتکل SNMP، یک شناسه که گروه ایستگاههای تحت نظارت یک مدیر را مشخص می‌کند؛ و بخش داده که به چند واحد داده تقسیم می‌شود. هر واحد داده پروتکل شامل یک درخواست از طرف ایستگاه مدیریت و یک پاسخ از سوی ایستگاه تحت مدیریت می‌باشد. در زیر قالب پیغام به زبان ASN نمایش داده شده است:

SNMP-Message ::=

```

SEQUENCE {
    version INTEGER {
        version-1 (0)
    },
    community
        OCTET STRING,
    data
        ANY
}

```

پروتکل SNMP بر خلاف اسم آن - "پروتکل ساده مدیریت شبکه" - چندان هم ساده نیست و مستندات آن ۶۰۰ صفحه را در بر می‌گیرد. بسیاری از منتقدان معتقدند هرگاه سازمان استاندارد جهانی ISO قدم در استاندارد کردن یک پروتکل بر می‌دارد به دلیل وسواسی که در کامل بودن آن استاندارد دارد، چنان آنرا پیچیده می‌کند که پیاده‌سازی، استفاده و فهم آن مشکل می‌شود. ASN.1 نیز از این قاعده مستثنی نیست و کلاً پیچیده و مبهم است. علاقمندانی که به رسالت پروتکل‌های مدیریت شبکه همانند SNMP و زبان توصیف اشیاء مثل استاندارد ASN پی برده‌اند برای درک کامل جزئیات آنها، باید به مراجع معرفی شده مراجعه کنند.

۶) مراجع این فصل

مجموعه مراجع زیر می‌توانند برای دست آوردن جزئیات دقیق و تحقیق جامع در مورد مفاهیم معرفی شده در این فصل مفید واقع شوند.

	"Computer Networks" , Andrew S.Tanenbaum, Third Edition, Prentice-Hall, 1996.
	"Internetworking with TCP/IP", Commer D.E. ,Prentice-Hall, 1996.
RFC 1101	"DNS Encoding of Network Names and Other Types," Mockapetris, P.V.; 1989
RFC 1035	"Domain Names—Implementation and Specification," Mockapetris, P.V.; 1987
RFC 1034	"Domain Names—Concepts and Facilities," Mockapetris, P.V.; 1987
RFC 1033	"Domain Administrators Operations Guide," Lottor, M.; 1987
RFC 1032	"Domain Administrators Guide," Stahl, M.K.; 1987
RFC 974	"Mail Routing and the Domain System," Partridge, C.; 1986
RFC 920	"Domain Requirements," Postel, J.B.; Reynolds, J.K.; 1984
RFC 799	"Internet Name Domains," Mills, D.L.; 1981
RFC 1157	Simple Network Management Protocol (SNMP). J.D. Case, M. Fedor, M.L. Schoffstall, C. Davin. May-01-1990
RFC 1161	SNMP over OSI. M.T. Rose. Jun-01-1990.
RFC 1187	Bulk Table Retrieval with the SNMP. M.T. Rose, K. McCloghrie, J.R. Davin. Oct-01-1990.
RFC 1902	Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2). SNMPv2 Working Group, J. Case, K. McCloghrie, M. Rose & S. Waldbusser. January 1996.
RFC 2578	Structure of Management Information Version 2 (SMIv2). K. McCloghrie, D. Perkins, J. Schoenwaelder. April 1999.