۷ لایه کاربرد

بعد از به سرانجام رساندن همهٔ مقدمات، اکنون به لایه‌ای رسیده‌ایم که تمام کاربردهای شبکه در آن قرار دارد: لایهٔ کاربرد (application layer) . لایه‌های زیرین لایهٔ کاربرد فقط برای سرویس دادن به این لایه هستند، و هیچ کار واقعی برای کاربران انجام نمی‌دهند. در این فصل با کاربردهای واقعی شبکه آشنا خواهید شد.

با این حال در لایهٔ کاربرد هم به پروتکلهای پشتیبانی‌کننده، پروتکلهایی که بار وظایف را بر گردن می‌گیرند، نیاز داریم. بهمین دلیل، برای شروع یکی از این پروتکلها را بررسی خواهیم کرد. این پروتکل که نام آن DNS است، نامگذاری در اینترنت را بر عهده دارد. پس از آن، سه تا از کاربردهای واقعی شبکه را مورد بررسی قرار خواهیم داد: پُست الکترونیک (ایمیل)، تارنمای جهانی (یا باختصار، وب)، و چندرسانه‌ای.

۷.۱ سیستم نام ناحیه - DNS

با اینکه از نظر تئوری برنامه‌ها می‌توانند برای تماس با کامپیوترها، صندوق‌های پستی و منابع دیگر از آدرس شبکهٔ (مثلاً، IP) آنها استفاده کنند، حفظ کردن این قبیل آدرسها برای افراد دشوار است. همچنین اگر آدرس ایمیل «حسن» hasan@128.111.24.41 باشد، و ISP یا سازمان متبوع وی تصمیم بگیرد کامپیوتر سرویس‌دهنده پست الکترونیک خود را به آدرس IP دیگری منتقل کند، آدرس ایمیل «حسن» نیز عوض خواهد شد. بهمین دلیل برای تفکیک نام ماشینها از آدرس آنها، استفاده از نامهای معمولی باب شد. به این ترتیب، آدرس ای. بل «حسن» چیزی شبیه hasan@art.ucsb.edu خواهد شد. با این وجود، کامپیوترها فقط آدرسهای عددی را می‌فهمند، پس باید مکانیزمی برای تبدیل اسامی معمولی به آدرسهای شبکه فراهم کنیم. در این قسمت روش تبدیل نامهای معمولی به آدرس عددی را در اینترنت بررسی خواهیم کرد.

سالها قبل در آرپانت فایلی وجود داشت بنام hosts.txt ، که نام کامپیوترها و آدرس IP آنها در این فایل لیست می‌شد. کامپیوترهای شبکه هر شب این فایل را از جایی که قرار داشت، می‌خواندند و خود را به روز می‌کردند. برای شبکه‌ای با دهها (و یا صدها) کامپیوتر این روش بخوبی کار می‌کرد.

ولی وقتی تعداد کامپیوترهای شبکه از مرز هزاران PC و کامپیوتر بزرگ گذشت، همه دریافتند که این روش دیگر جوابگو نیست. اولین دلیل آن بود که اندازهٔ چنین فایلی بشدت بزرگ می‌شد، ولی از آن مهمتر مشکل نامهای تکراری بود که ضرورت یک مدیریت مرکزی را اجتناب‌ناپذیر می‌کرد (چیزی که بزرگی و بار شبکه آنرا ناممکن می‌کرد). برای غلبه بر این مشکلات بود که DNS (سیستم‌نام‌ناحیه - Domain Name System) اختراع شد.

ایدهٔ اصلی DNS یک روش نامگذاری سلسله مراتبی بر اساس ناحیه‌ها بود، که بصورت یک پایگاه اطلاعاتی

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

توزیع‌یافته پیاده‌سازی می‌شد. هدف اولیهٔ این سیستم تبدیل نام کامپیوترها و آدرسهای ایمیل (پست الکترونیک) به آدرسهای IP بود، ولی می‌توانست کاربردهای دیگری هم داشته باشد. DNS در RFC 1034 و RFC 1035 تعریف شده است.

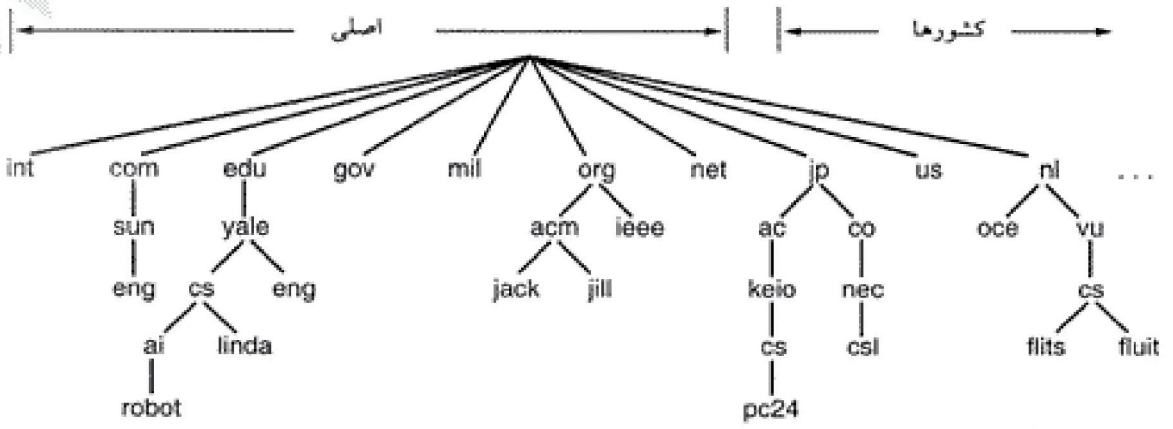
روش کار DNS خیلی خلاصه چنین است: برای تبدیل یک نام به آدرس IP ، برنامه یک تابع کتابخانه‌ای بنام تبدیل‌کننده (resolver) را فراخوانی می‌کند، و نام موردنظر را بصورت پارامتر به آن می‌دهد. تابع *gethostbyname* در شکل ۶-۶ نمونه‌ای از یک تبدیل‌کننده است. تبدیل‌کننده یک بستهٔ UDP به سرویس‌دهنده DNS محلی می‌فرستد، که این DNS آدرس IP معادل نام خواسته شده را یافته و به تبدیل‌کننده برمی‌گرداند، که آن هم بنوبهٔ خود آدرس را به برنامهٔ فراخوانی‌کننده تحویل می‌دهد. برنامه هم پس از بدست آوردن آدرس IP کامپیوتر مقصد، می‌تواند با آن ارتباط TCP برقرار کرده یا بسته‌های UDP به آن بفرستد.

۷-۱-۱ فضای نام DNS

مدیریت مجموعه‌ای بزرگ و دائماً در تغییر از نامها بهیچوجه کار ساده‌ای نیست. در سیستم پُست، مدیریت نامها از طریق اجبار افراد به نوشتن نام کشور، استان (یا ایالت)، شهر، خیابان و شمارهٔ پلاک مقصد انجام می‌شود. با این روش دیگر «پلاک ۷، خیابان آزادی، تهران» هرگز با «پلاک ۷، خیابان آزادی، اصفهان» اشتباه نخواهد شد. DNS هم به همین روش کار می‌کند.

اینترنت به بیش از ۲۰۰ ناحیهٔ سطح بالا (top-level domain) (که هر کدام تعداد زیادی کامپیوتر را در برمی‌گیرند) تقسیم شده است. هر ناحیه به چندین زیرناحیه (subdomain) ، و آنها نیز بنوبهٔ خود به زیرناحیه‌های کوچکتر، تقسیم می‌شوند. این سلسله مراتب را می‌توان بصورت یک درخت نمایش داد (شکل ۷-۱ را ببینید). ناحیه‌هایی که زیرناحیه ندارند، برگهای این درخت را تشکیل می‌دهند. هر یک از این برگ‌ها می‌تواند یک کامپیوتر، یک شبکهٔ کوچک (با چند کامپیوتر)، و یا شرکتی بزرگ (با هزاران کامپیوتر) باشد.

ناحیه‌های سطح بالا بر دو گونه‌اند: عمومی و کشورها. ناحیه‌های عمومی اولیه عبارت بودند از: *com* (مخفف commercial ، تجاری)، *edu* (مخفف educational ، مؤسسات آموزشی)، *gov* (مخفف goverment ، ادارات دولتی)، *int* (مخفف international ، مؤسسات بین‌المللی)، *mil* (مخفف military ، نظامی)، *net* (مخفف network provider ، شرکتهای خدمات شبکه و اینترنت)، و *org* (مخفف organization ، مؤسسات غیرانتفاعی). هر کشور نیز دارای یک ناحیهٔ خاص (در ناحیهٔ کشورها) است، که در استاندارد ISO 3166 تعریف شده است.



شکل ۷-۱. بخشی از فضای نام ناحیه در اینترنت.

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

در نوامبر ۲۰۰۰، ICANN چهار ناحیهٔ سطح بالای جدید را برای مصارف عمومی تصویب کرد: *biz* (مخفف businesses ، مشاغل)، *info* (مخفف information ، شرکتهای اطلاعاتی)، *name* (نام افراد)، و *pro* (مخفف proffession ، صاحبان حِرَف مانند وکلا و پزشکان). علاوه بر آن، سه ناحیهٔ سطح بالای تخصصی نیز معرفی شد، که برخی از صنایع خاص می‌توانند از آنها استفاده کنند؛ این سه ناحیه عبارتند از: *aero* (مخفف aerospace ، صنایع هوا-فضا)، *coop* (مخفف co-operatives ، تعاونی‌ها)، و *museum* (موزه‌ها). به احتمال زیاد در آینده ناحیه‌های سطح بالای دیگری نیز اضافه خواهند شد.

از طرف دیگر هر چه اینترنت بیشتر تجاری می‌شود، بحث و جدل هم بالاتر می‌گیرد. مثلاً همین ناحیهٔ *pro* را در نظر بگیرید. این ناحیه برای صاحبان حِرَف که صلاحیت آنها تأئید شده باشد، در نظر گرفته شده است. اما حرفه‌ای کیست؟ و چه کسی باید صلاحیت‌ها را تأئید کند؟ یک پزشک یا وکیل مسلماً حرفه‌ایست، اما یک عکاس، معلم پیانو، شعبده‌باز، لوله‌کش، آرایشگر، خالکوب، آدمکش حرفه‌ای و یا فاحشه چطور؟ آیا اینها هم حرفه‌اند، و شایستهٔ دریافت ناحیهٔ *pro* ؟ و اگر پاسخ مثبت است، چه کسی صلاحیت داوطلبان را تأئید می‌کند؟

در کل، گرفتن ناحیهٔ سطح-دوم در یک ناحیهٔ سطح بالا ، مانند *name-of-company.com* ، ساده است: تنها کاری که باید کرد مراجعه به پایگاه اطلاعاتی ناحیهٔ سطح بالا (در اینجا *com*) و اطمینان از آزاد بودن نام موردنظر است. اگر مشکلی وجود نداشته باشد، درخواست‌کننده پول کمی بابت هزینهٔ سالیانه پرداخته، و آن نام را بدست می‌آورد. می‌توان به جرأت گفت که، اکنون تمام نامهای بامعنای انگلیسی در ناحیهٔ *com* گرفته شده‌اند (اگر باور ندارید، امتحان کنید!).

نام هر ناحیه بصورت مسیری رو به بالا و به سمت یک ریشه (که نامی ندارد) مشخص می‌شود. اجزای این نام با نقطه(که «دات» تلفظ می‌شود) از هم جدا می‌شوند. برای مثال، نام ناحیهٔ بخش مهندسی شرکت سان میکروسیستمز می‌تواند *eng.sun.com.* باشد (در حالیکه همین نام در سیستم عامل یونیکس بصورت */com/sun/eng* نوشته می‌شود). دقت کنید که با این روش نامگذاری دیگر نام ناحیهٔ بخش مهندسی سان میکروسیستمز با دانشکدهٔ مهندسی دانشگاه ییل (که مثلاً *eng.yale.edu.* است) اشتباه نخواهد شد.

نام ناحیه می‌تواند مطلق (absolute) یا نسبی (relative) باشد. یک نام مطلق همیشه به نقطه ختم می‌شود (مانند *eng.sun.com.*)، در حالیکه نامهای نسبی چنین نیستند. نامهای نسبی بدون توجه به جایی که بکار رفته‌اند، معنی نمی‌دهند. در هر دو حالت، یک نام ناحیه به گرهی خاص در درخت نامها (و تمام گرههای ذیل آن) اشاره می‌کند.

کوچکی یا بزرگی حروف در نام ناحیه بی‌تأثیر است، بعبارت دیگر *edu* ، *Edu* و *EDU* همگی یک معنی می‌دهند. هر جزء از نام ناحیه حداکثر ۶۳ حرف، و کل مسیر نام ناحیه حداکثر ۲۵۵ حرف می‌توانند داشته باشند.

برای قرار دادن یک ناحیه در درخت نامهای ناحیه دو روش وجود دارد. برای مثال، ناحیهٔ *cs.yale.edu* می‌تواند در ذیل شاخهٔ کشور آمریکا (*us*) و بصورت *cs.yale.ct.us* هم ثبت شود. در کل، اغلب سازمانها و شرکتها در آمریکا ترجیح می‌دهند از ناحیه‌های عمومی استفاده کنند، در حالیکه در خارج از آمریکا بیشتر از نام کشور بعنوان ناحیهٔ سطح بالا استفاده می‌شود. هیچ منعی برای ثبت نام ناحیه در شاخه‌های متعدد وجود ندارد، با این حال چنین گرایشی بجز در شرکتهای چندملیتی (مانند شرکت سونی که ناحیه‌های *sony.com* و *sony.nl* را ثبت کرده) وجود ندارد.

هر ناحیه ناحیه‌های ذیل خود را کنترل می‌کند. برای مثال، کشور ژاپن دارای ناحیه‌های *ac.jp* و *co.jp* است، که بترتیب مشابه *edu* و *com* هستند؛ در حالیکه در هلند چنین تقسیم‌بندی وجود ندارد، و تمام ناحیه‌ها ذیل *nl* قرار دارند. برای مثال، ناحیه‌های زیر همگی دانشکده‌های کامپیوتر در کشور مربوطه هستند:

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

۱. cs.yale.edu (دانشگاه ییل، ایالات متحدهٔ آمریکا)
۲. cs.vu.nl (دانشگاه فریژه، هلند)
۳. cs.keio.ac.jp (دانشگاه کی‌یو، ژاپن)

برای ایجاد یک زیرناحیه، مجوز ناحیهٔ بالاتر مورد نیاز است. برای مثال اگر گروه VLSI در دانشکدهٔ کامپیوتر دانشگاه ییل تأسیس شود، و بخواهد به نام *vlsi.cs.yale.edu* شناخته شود، باید مجوزهای لازم را از مسئول *cs.yale.edu* کسب کند. بهمین ترتیب اگر دانشگاه جدیدی، مثلاً دانشگاه شمالی داکوتای جنوبی، تأسیس شود، و بخواهد ناحیهٔ *unsd.edu* را برای خود ثبت کند، باید هماهنگیهای لازم را با مسئول ناحیهٔ *edu* بعمل آورد. قرار دادن مسئولیت زیرناحیه‌ها بر عهدهٔ ناحیهٔ بالاتر باعث می‌شود تا هیچ دو ناحیه‌ای همنام نشوند. همین که یک ناحیه ایجاد شد (مانند، *unsd.edu*)، دیگر می‌تواند بدون نظارت ناحیه‌های بالاتر به ایجاد زیرناحیه‌های دلخواه خود (مثلاً، *cs.unsd.edu*) بپردازد.

نامگذاری ناحیه‌ها امری سازمانی است نه فیزیکی. برای مثال، دانشکده‌های کامپیوتر و مهندسی برق یک دانشگاه می‌توانند ناحیه‌های کاملاً مستقلی داشته باشند، حتی اگر در یک ساختمان باشند و از شبکهٔ LAN واحدی استفاده کنند. از طرف دیگر، بخشهای مختلف یک دانشکده به یک ناحیه تعلق دارند، حتی اگر از نظر فیزیکی از هم جدا باشند.

۷-۱-۲ رکوردهای منابع

هر ناحیه، خواه ناحیه‌ای سطح بالا یا ناحیه‌ای با یک کامپیوتر، دارای تعدادی رکوردمنابع (resource record) است. برای یک کامپیوتر، متداولترین رکورد منبع آدرس IP آن است، اما انواع دیگری از رکوردهای منابع می‌تواند وجود داشته باشد. وقتی یک تبدیل‌کننده نام ناحیه را به DNS می‌دهد، چیزی که دریافت می‌کند تمام رکوردهای منابع وابسته به آن نام است. بنابراین، اصلی‌ترین وظیفهٔ یک DNS تبدیل نام ناحیه به رکوردهای منابع است.

هر رکورد منبع پنج بخش دارد. با اینکه رکوردهای منابع را می‌توان برای کارایی بهتر بصورت باینری در آورد، ولی در اغلب مواقع این رکوردها بصورت متنی (ASCII) - یک رکورد در هر خط - نگهداری می‌شوند. فرمت یک رکورد منبع مانند زیر است:

Domain_name Time_to_live Class Type Value

که در آن *Domain_name* نام ناحیه‌ایست که این رکورد متعلق به آن است. معمولاً هر ناحیه تعداد زیادی رکورد دارد، و در هر پایگاه داده اطلاعات چندین ناحیه نگهداری می‌شود. در نتیجه این فیلد کلید اصلی جستجو در پایگاه دادهٔ DNS است. (ترتیب قرار گرفتن رکوردها در پایگاه داده اهمیتی ندارد.)

فیلد *Time_to_live* مشخص می‌کند که این رکورد چقدر دوام می‌آورد. این فیلد در رکوردهای با دوام مقدار زیادی دارد، مثلاً ۸۶۴۰۰ (تعداد ثانیه‌های یک شبانه‌روز)؛ و بر عکس، اطلاعات کم دوام عمر کوتاهی دارد، مثلاً ۶۰ (یک دقیقه). در بحث حافظهٔ نهان به این موضوع برخواهیم گشت.

فیلد سوم هر رکورد منبع *Class* است. برای اطلاعات اینترنتی این فیلد همیشه *IN* است؛ برای اطلاعات غیراینترنتی از کُدهای دیگری هم می‌توان استفاده کرد (که البته بندرت دیده می‌شوند).

فیلد *Type* نوع رکورد منبع را مشخص می‌کند. مهمترین انواع رکوردهای منابع را در شکل ۷-۲ ملاحظه می‌کنید.

رکورد *SOA* نام منبع اصلی اطلاعات منطقه (zone) ، آدرس ایمیل سرپرست ناحیه، شماره سریال منحصر به فرد آن، و مشخصات دیگر ناحیه را مشخص می‌کند.

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

نوع	مفهوم	مقدار
SOA	Start of Authority	پارامترهای منطقه
A	IP address of a host	عدد صحیح ۳۲ بیتی
MX	Mail exchange	تقدم دریافت ایمیل
NS	Name Server	نام سرویس دهنده ناحیه
CNAME	Canonical name	نام ناحیه
PTR	Pointer	نام مستعار برای آدرس IP
HINFO	Host description	مشخصات CPU و سیستم عامل
TXT	Text	متن تفسیر نشده

شکل ۷-۲. انواع رکوردهای منابع اصلی DNS برای IPv4 .

مهمترین نوع رکورد منبع، رکورد *A* (آدرس) است. هر رکورد *A* یک آدرس IP ۳۲ بیتی را در خود نگه می‌دارد. هر کامپیوتر اینترنت باید حداقل یک آدرس IP داشته باشد، تا کامپیوترهای دیگر بتوانند با آن تماس بگیرند. برخی از کامپیوترها دو یا چند آدرس IP دارند، که برای هر آدرس IP چنین کامپیوتری باید یک رکورد *A* وجود داشته باشد. DNS را می‌توان بگونه‌ای پیکربندی کرد که در میان این رکوردها بچرخد، یعنی برای اولین درخواست اولین رکورد را برگرداند، برای درخواست دوم دومین رکورد را، و الی‌آخر.

رکورد مهم بعدی رکورد *MX* است. این رکورد آدرس سرویس‌دهنده پُست الکترونیک (ایمیل) ناحیه را مشخص می‌کند. اختصاص یک نوع رکورد خاص به سرویس‌دهنده پست الکترونیک بدین خاطر است که تمام کامپیوترها چنین قابلیتی (دریافت ایمیل) ندارند. برای مثال، اگر کسی بخواهد به *bill@microsoft.com* ایمیل بفرستد، باید آدرس سرویس‌دهنده پست الکترونیک *microsoft.com* را پیدا کند. این اطلاعات را رکورد *MX* عرضه می‌کند.

رکورد *NS* سرویس‌دهنده‌نام (name server) را مشخص می‌کند. برای مثال، معمولاً هر پایگاه دادهٔ DNS یک رکورد NS برای ناحیه‌های سطح بالا دارد. (باز هم به این موضوع برمی‌گردیم.)

از رکورد *CNAME* می‌توان برای ایجاد **نامهای‌مستعار** (alias) استفاده کرد. برای مثال، فرض کنید دوستی بنام پاول در دانشکدهٔ مهندسی کامپیوتر دانشگاه MIT دارید، و می‌خواهید برای وی ایمیل بفرستید. فقط می‌دانید که نام وی در شبکهٔ این دانشگاه *paul* است، اما آدرس ایمیل کامل او را ندارید. حدس می‌زنید که ناحیهٔ دانشکدهٔ مزبور *cs.mit.edu* باشد، اما مسئولین این دانشکده (شاید از روی کج‌سلیقگی) نام *lcs.mit.edu* را برای ناحیهٔ خود را انتخاب کرده‌اند. اگر ایمیلی به آدرس *paul@cs.mit.edu* بفرستید، مسلماً برگشت خواهد خورد؛ ولی اگر مسئولین این دانشکده کمی هوشیاری بخرج دهند، با تعریف یک نام مستعار می‌توانند تا حدی اشتباه خود را جبران کنند - برای این منظور می‌توان از یک رکورد *CNAME* مانند زیر استفاده کرد:

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

رکورد *PTR* هم، مانند *CNAME* ، به یک نام دیگر اشاره می‌کند. اما بر خلاف *CNAME* (که در واقع یک ماکرو است)، رکورد *PTR* یک نوع دادهٔ معمولی DNS است که تفسیر آن به محتویات این رکورد بستگی دارد. در عمل، تقریباً همیشه از این رکورد برای **جستجوی‌معکوس** (reverse lookup) - تبدیل آدرس IP به نام ماشین - استفاده می‌شود.

رکورد *HINFO* اطلاعات مربوط به نوع ماشین و سیستم عامل آنرا برمی‌گرداند. از رکورد *TXT* هم می‌توان

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

برای برگرداندن اطلاعات متنی اضافی به کاربران استفاده کرد. رکوردهای *HINFO* و *TXT* فقط برای راحتی کاربران تعبیه شده‌اند، و الزامی نیستند. اغلب اوقات چنین اطلاعاتی وجود ندارد، و اگر هم وجود داشته باشد، نمی‌توان به آن کاملاً اطمینان کرد.

آخرین فیلد رکورد منبع، فیلد *Value* (مقدار) است. این فیلد می‌تواند یک عدد، نام ناحیه، یا یک رشتهٔ متنی باشد. طرز وارد کردن این مقدار به نوع آن بستگی دارد (در شکل ۷-۲ توضیح کوتاهی دربارهٔ نوع هر فیلد آورده شده است).

برای آشنایی بیشتر با اطلاعاتی که در پایگاه دادهٔ DNS یک ناحیه می‌توان یافت، شکل ۷-۳ را ببینید. در این شکل قسمتی از پایگاه دادهٔ نیمه‌فرضی ناحیه‌ای بنام *cs.vu.nl* (شکل ۷-۱) نشان داده شده است. در این پایگاه داده هفت نوع رکورد منبع وجود دارد.

اولین خط غیرتوضیحی شکل ۷-۳ مقداری اطلاعات اولیه دربارهٔ ناحیهٔ *cs.vu.nl* می‌دهد، که فعلاً با آنها کاری نداریم. دو خط بعدی مقداری اطلاعات متنی دربارهٔ این ناحیه (و محل آن) می‌دهند. پس از آن دو کامپیوتری که مسئول دریافت ایمیل‌های ناحیهٔ *cs.vu.nl* هستند، مشخص شده‌اند. اولین کامپیوتری که ایمیل‌ها باید به آن فرستاده شوند، *zephyr* نام دارد؛ و اگر *zaphyr* جواب نداد، نوبت به *top* می‌رسد.

بعد از یک خط خالی (که فقط برای خواناتر کردن فایل است)، رکوردهایی آمده‌اند که می‌گویند *flits* یک کامپیوتر Sun با سیستم عامل UNIX است، و دو آدرس IP دارد (130.37.16.112 و 192.31.231.165). پس از آن سه ماشین برای دریافت ایمیل‌های *flits.cs.vu.nl* مشخص شده است: اولین آنها طبیعتاً خود *flits* است، ولی

```
; Authoritative data for cs.vu.nl
cs.vu.nl.       86400   IN  SOA     star.boss (952771,7200,7200,2419200,86400)
cs.vu.nl.       86400   IN  TXT     "Divisie Wiskunde en Informatica."
cs.vu.nl.       86400   IN  TXT     "Vrije 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       130.37.16.112
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.vu.nl

rowboat                 IN  A       130.37.56.201
                        IN  MX      1 rowboat
                        IN  MX      2 zephyr
                        IN  HINFO   Sun Unix

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
```

شکل ۷-۳. قسمتی از پایگاه دادهٔ DNS در ناحیهٔ *cs.vu.nl* .

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

اگر این ماشین خاموش بود، *zephyr* و *top* گزینه‌های بعدی خواهند بود. بعد از آن یک نام مستعار برای ماشین *star.cs.vu.nl* تعریف شده است: *www.cs.vu.nl* . با تعریف این نام مستعار، کاربران می‌توانند بدون دغدغهٔ تغییر آدرس صفحهٔ وب *cs.vu.nl* به آن مراجعه کنند. همین کار برای *ftp.cs.vu.nl* نیز انجام شده است.

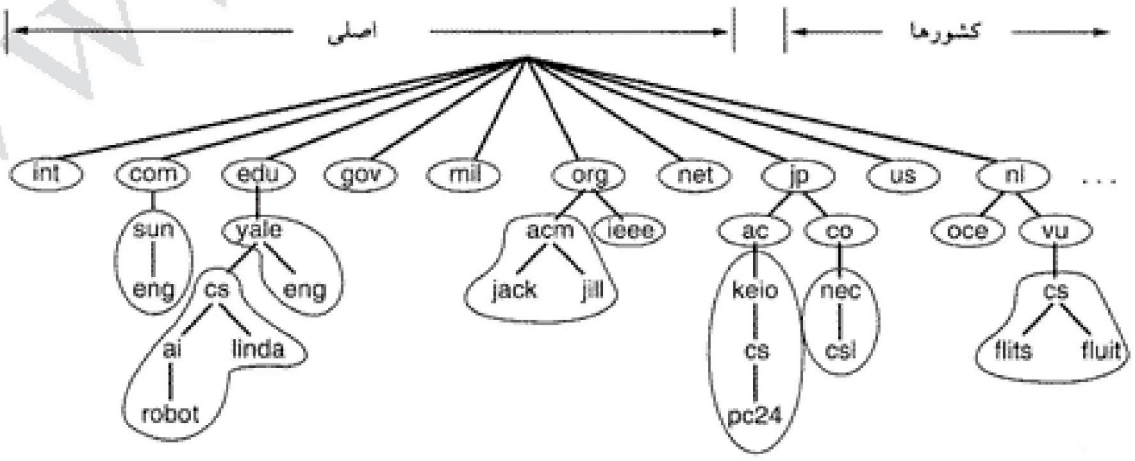
در چهار خط بعدی رکوردهای منبع کامپیوتری بنام *rowboat.cs.vu.nl* تعریف شده‌اند. این اطلاعات شامل آدرس IP ، محل دریافت ایمیل (اولیه و ثانویه)، و اطلاعاتی دربارهٔ خود ماشین است. پس از آن یک کامپیوتر MAC (بدون قابلیت دریافت ایمیل)، و بدنبال آن یک چاپگر لیزری که به اینترنت متصل است، تعریف شده‌اند.

چیزی که در اینجا نشان داده نشده (و در واقع در این فایل هم نیست)، آدرس IP ناحیه‌های سطح بالا است. از آنجائیکه این کار در حوزهٔ مسئولیت ناحیهٔ *cs.vu.nl* نیست، در این فایل هم رکوردی برای آن وجود ندارد. این قبیل اطلاعات را کامپیوترهایی بنام **سرویس‌دهنده ریشه** (root server) ارائه می‌کنند، که با هر بار اجرای سرویس‌دهنده DNS آدرس آنها در حافظه حافظهٔ نهان DNS بار می‌شود. تعداد سرویس‌دهنده‌های ریشه در حدود ۱۲ تاست که در سراسر دنیا پراکنده‌اند، و آدرس IP تمام ناحیه‌های سطح بالا را دارند. بنابراین، اگر ماشینی آدرس IP حداقل یکی از این سرویس‌دهنده‌های ریشه را داشته باشد، می‌تواند نام هر ناحیه‌ای را پیدا کند.

۳-۱-۷ سرویس‌دهنده نام

از نظر تئوری، برای نگهداری تمام اطلاعات DNS و پاسخ دادن به درخواست‌ها یک سرویس‌دهنده DNS کافیست. اما در عمل، بار کاری چنین کامپیوتری آنقدر سنگین خواهد شد که عملاً آنرا بلااستفاده می‌کند. علاوه بر آن، اگر این کامپیوتر از کار بیفتد، تمام اینترنت هم با آن به خراب خواهد رفت.

برای اجتناب از چنین وضعیتی، فضای نام DNS به چندین منطقه (zone) با مرزهای مشخص و غیرمشترک تقسیم شده است. در شکل ۷-۴ یکی از راههای تقسیم فضای نام شکل ۷-۱ را مشاهده می‌کنید. هر منطقه شامل بخشی از درخت DNS است، و سرویس‌دهنده‌های نام آنرا در خود نگه می‌دارد. معمولاً هر منطقه دارای یک **سرویس‌دهنده نام**(name server) اولیه است که اطلاعاتش را از فایلی روی دیسک خود می‌گیرد، و یک یا چند سرویس‌دهنده نام ثانویه نیز دارد که آنها اطلاعات خود را از سرویس‌دهنده نام اولیه می‌گیرند. برای بالا بردن ضریب اطمینان، می‌توان تعدادی از سرویس‌دهنده‌های نام یک منطقه را خارج از آن منطقه مستقر کرد.



شکل ۷-۴. فضای نام DNS به منطقه‌های مختلف تقسیم شده است.

تعیین مرزهای یک منطقه بر عهدهٔ سرپرست آن است. تصمیم‌گیری در این باره تا حد زیادی به تعداد سرویس‌دهنده‌های نام منطقه و محل استقرار آنها بستگی دارد. برای مثال، در شکل ۷-۴، دانشگاه ییل دارای یک

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

سرویس‌دهنده نام برای ناحیه‌های *yale.edu* و *eng.yale.edu* است، اما ناحیهٔ *cs.yale.edu* در منطقهٔ دیگری قرار دارد. چنین تصمیماتی بیشتر به تمایل ناحیه‌ها برای کنترل مستقیم منطقهٔ خود بستگی دارد. در مثال فوق، ناحیهٔ *cs.yale.edu* منطقه‌ای مستقل است، در حالیکه *eng.yale.edu* چنین نیست.

وقتی یک تبدیل‌کننده می‌خواهد آدرس ناحیه‌ای را بداند، ابتدا درخواست خود را به سرویس‌دهنده‌های نام محلی خود می‌دهد. اگر این ناحیه در محدودهٔ قانونی سرویس‌دهنده نام مزبور بود (مانند *ai.cs.yale.edu* که در قلمرو *cs.yale.edu* است)، سرویس‌دهنده نام رکوردهای منبع معتبر را به آن برمی‌گرداند. یک **رکورد معتبر** (authoritative record) رکوردی است که مستقیماً از سرپرست ناحیه منشأ می‌گیرد، و بنابراین همیشه صحیح و معتبر است (بر خلاف **رکوردهای ذخیره‌شده** – cached record – که تاریخ اعتبار آنها می‌تواند منقضی شده باشد).

ولی اگر آن ناحیه در قلمرو سرویس‌دهنده‌های محلی نباشد، سرویس‌دهنده نام این درخواست را به سرویس‌دهنده نام سطح بالای ناحیهٔ مزبور می‌فرستد. برای روشنتر شدن مطلب، به مثالی در شکل ۷-۵ توجه کنید. در اینجا یک تبدیل‌کننده در ناحیهٔ *flits.cs.vu.nl* می‌خواهد آدرس IP ماشین سرویس‌دهنده ناحیهٔ *linda.cs.yale.edu* را بداند. در مرحلهٔ ۱، این تبدیل‌کننده درخواست خود را به سرویس‌دهنده نام محلی، یعنی *cs.vu.nl* ، می‌فرستد. این درخواست شامل نام ناحیهٔ موردنظر (رکوردهای نوع *A* و کلاس *IN*) می‌باشد.

شکل ۷-۵. مراحل جستجوی نام ناحیه.

اجازه دهید فرض کنیم سرویس‌دهنده نام محلی تا بحال چیزی از ناحیهٔ *linda.cs.yale.edu* نشنیده و دربارهٔ آن هیچ اطلاعاتی ندارد. این سرویس‌دهنده می‌تواند از همسایه‌های خود در این باره پرس و جو کند، ولی اگر آنها هم بی‌اطلاع بودند، یک بستهٔ UDP به سرویس‌دهنده نام *edu* یعنی *edu-server.net* (که آدرس آنرا در حافظه‌اش دارد) می‌فرستد (شکل ۷-۵ را ببینید). احتمال کمی هست که این سرویس‌دهنده آدرس *linda.cs.yale.edu* (یا حتی *cs.yale.edu*) را بداند، ولی حتماً بچه‌های خودش را می‌شناسد، پس درخواست را به سرویس‌دهنده نام *yale.edu* منتقل می‌کند (مرحلهٔ ۳). سرویس‌دهنده *yale.edu* هم درخواست را به *cs.yale.edu* هدایت می‌کند (مرحلهٔ ۴)، که باید اطلاعات معتبر را در اختیار داشته باشد. از آنجائیکه این مسیر از مشتریهای مختلفی عبور کرده، رکوردهای درخواستی هم باید از همان مسیر به *flits.cs.vu.nl* برگردند (مراحل ۵ تا ۸).

وقتی این رکوردها به *cs.vu.nl* رسید، در حافظهٔ نهان آنجا ذخیره می‌شود تا در دفعات بعد مورد استفاده قرار گیرد. ولی این اطلاعات معتبر نیستند، چون هر تغییری که در *cs.yale.edu* داده شود بطور خودکار در حافظهٔ نهان DNS هایی که این رکوردها در آنجا وجود دارد، پخش نخواهد شد. بهمین دلیل، آیتمهای حافظهٔ نهان نباید عمری طولانی داشته باشند؛ و علت قرار دادن فیلد *Time_to_live* در رکوردهای منبع نیز همین است. این فیلد به سرویس‌دهنده نام می‌گوید که تا چه مدتی می‌تواند رکورد را در حافظهٔ نهان خود نگه دارد. اگر یک ماشین IP خود را سالها حفظ می‌کند، نگه داشتن آن برای ۱ روز در حافظهٔ نهان DNS چندان غیرمنطقی نیست. اما اطلاعات ناپایدارتر را بهتر است بیش از چند ثانیه (یا حداکثر ۱ دقیقه) در حافظهٔ نهان نگه نداریم.

به پرس و جوهایی که به طریق بالا عمل می‌کنند، **جستجوی بازگشتی** (recursive query) می‌گویند، چون هر سرویس‌دهنده‌ای که اطلاعات خواسته شده را نداشته باشد، آنرا به سرویس‌دهنده بالاتر هدایت کرده و جواب را

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

باز می‌گرداند. روش جستجوی دیگری نیز وجود دارد: سرویس‌دهنده‌ای که اطلاعات درخواست شده را ندارد، خود به سرویس‌دهنده بالاتر مراجعه نمی‌کند، بلکه آدرس آنرا به درخواست کننده برمی‌گرداند. برخی از سرویس‌دهنده‌های DNS قادر به انجام جستجوی بازگشتی نیستند، و همیشه آدرس سرویس‌دهنده بعدی را برمی‌گردانند.

اگر یک مشتری DNS در زمان مقرر پاسخ خود را دریافت نکند، سراغ سرویس‌دهنده DNS بعدی خواهد رفت. این اتفاق بیشتر در مواردی رخ می‌دهد که سرویس‌دهنده DNS بدلایلی از مدار خارج شده باشد.

با اینکه DNS کار ساده‌ای انجام می‌دهد (تبدیل نام به آدرس IP)، اما کارکرد صحیح آن در اینترنت اهمیت حیاتی دارد. DNS هیچ کمکی برای یافتن افراد، منابع، سرویسها، اشیاء و چیزهایی از این قبیل نمی‌تواند بکند؛ برای این کارها سرویس دیگری بنام LDAP (پروتکل سبک دسترسی دایرکتوری – Light-weight Directory Access Protocol) تعریف شده است. این سرویس شکل ساده شدهٔ سرویس دایرکتوری OSI X.500 است، که در استاندارد RFC 2251 تشریح شده است. در LDAP (که می‌توان آنرا «دفتر تلفن اینترنتی» بشمار آورد)، اطلاعات بصورت درختی منظم شده‌اند، و امکانات فراوانی برای جستجو در این درخت تعبیه شده است. در این کتاب بیش از این دربارهٔ LDAP صحبت نخواهیم کرد؛ برای کسب اطلاعات بیشتر می‌توانید به (Weltman and Dahbura, 2000) مراجعه کنید.

۲-۷ پُست الکترونیک

بیش از دو دهه است که پُست الکترونیک، یا آنطور که هوادارانش می‌گویند ایمیل (e-mail)، در صحنه حضور دارد. تا سال ۱۹۹۰، این سرویس بیشتر در دانشگاهها و مراکز علمی وجود داشت، ولی وقتی در این سال بصورت سرویسی عمومی در آمد، با چنان سرعتی رشد کرد که در طی یک دهه تعداد نامه‌های الکترونیکی فرستاده شده از تعداد نامه‌های کاغذی فراتر رفت.

ایمیل، مانند سایر روشهای ارتباطی، دارای قواعد و شیوه‌های خاص خود است. جاذبهٔ ایمیل بسیار بالاست، بطوریکه حتی آنهائیکه بندرت نامه‌های معمولی می‌نویسند، در نوشتن نامه‌های الکترونیکی (حتی به مقامات رسمی و سطح بالا) تردیدی بخود راه نمی‌دهند.

نامه‌های الکترونیکی پُر از کلماتیست که قبلاً در هیچ کجا دیده نشده‌اند: BTW (By The Way – راستی)، ROTFL (Rolling On The Floor Laughing – از خنده غش کردم)، و IMHO (In My Humble Opinion – به نظر من ناقابل) از آن نمونه‌اند. بسیاری افراد نیز در ایمیلهای خود از علائم خاصی موسوم به خندانک (smiley) یا احساس‌نما (emoticon) استفاده می‌کنند. در شکل ۷-۶ تعدادی از معروفترین این خندانک‌ها (و معنای آنها) را می‌بینید. اگر می‌خواهید بهتر متوجه معنای این علائم شوید، کتاب را ۹۰° در جهت

مفهوم	خندانک	مفهوم	خندانک	مفهوم	خندانک
دماغ گنده	:-)	عمو لینکن	:-| =	من خوشحالم	:-)
غبغب بزرگ	:-))	عمو سام	:-) =	من غمگین / ناراحتم	:-(
سبیلو	:-{)	بابا نوئل	*:-)>	من بی تفاوتم	:-|
ژولیده مو	#:-)	کودن / احمق	:-)___>	من چشمک می زنم	;-)
عینکی	8-)	استرالیایی	:-)	من خمیازه می کشم	(O)-:
با هوش	C:-)	مرد فلکی	X-) :-	حالم به هم خورد	(*)-:

شکل ۷-۶. چند خندانک. اینها جزء امتحان نهایی نیستند :-)

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

عقربه‌های ساعت بچرخانید. برای دیدن تعداد زیادی از این قبیل خندانک‌ها به (Sanderson and Dougherty, 1993) مراجعه کنید.

اولین سیستم ایمیل فقط یک پروتکل سادهٔ انتقال فایل (file transfer) بود، که آدرس گیرنده در خط اول پیام (فایل) نوشته می‌شد. با گذشت زمان محدودیت‌های این روش آشکارتر شد، که برخی از آنها عبارت بودند از:

۱. فرستادن یک پیام به چند نفر مشکل بود. این اشکال بیشتر مدیران را آزار می‌داد، چون آنها میل داشتند پیامهای خود را به تمام افراد زیردست خود بفرستند.
۲. پیامها هیچگونه ساختار داخلی نداشتند، و بهمین دلیل پردازش کامپیوتری آنها مشکل بود. برای مثال، اگر پیامی از طرف یک شخص واسطه هدایت یا فرستاده می‌شد، استخراج قسمت هدایت شده مشکل بود.
۳. فرستندهٔ نامه هرگز نمی‌توانست بداند پیامش به گیرنده رسیده یا نه.
۴. اگر کسی قصد داشت برای مدتی به مرخصی برود و می‌خواست در این مدت نامه‌های وارده به دست منشی‌اش برسد، کار ساده‌ای نبود.
۵. واسط کاربر (جایی که نامه را می‌نوشت) با قسمت ارسال نامه یکپارچه نبود. کاربر باید ابتدا نامه را می‌نوشت، و برای ارسال آن برنامهٔ ادیتور را ترک می‌کرد، و به قسمت انتقال فایل می‌رفت.
۶. نامه‌ها فقط متن بود؛ ارسال تصویر، طرح، صدا و مانند آنها ممکن نبود.

بتدریج سیستمهای ایمیل بهتری عرضه شد. در سال ۱۹۸۲، آرپانت سیستم ایمیل پیشنهادی خود را در RFC 821 (پروتکل انتقال) و RFC 822 (فرمت پیام) ارائه کرد. این پیشنهادها با تغییراتی اندک با عنوان RFC 2821 و RFC 2822 به استاندارد اینترنت تبدیل شد - اما هنوز هم استاندارد ایمیل اینترنت را با نام RFC 822 می‌شناسند.

در ۱۹۸۴، CCITT توصیه‌ای بنام X.400 ارائه کرد. بعد از دو دهه، اغلب سیستمهای ایمیل همچنان به RFC 822 پایبند هستند، و X.400 عملاً کنار گذاشته شده است. این که چگونه سیستمی به عظمت X.400 که تمام مقامات رسمی استاندارد، شرکتهای مخابرات سراسر دنیا، دولتها و بسیاری از صنایع کامپیوتری پشتیبان آن بودند، مغلوب سیستمی که چند دانشجوی کامپیوتر آنرا نوشته‌اند، می‌شود بیشتر به داستان داوود و گولیات شبیه است.

علت موفقیت RFC 822 خوبی آن نبود، بلکه این X.400 بود که چنان پیچیده و بد طراحی شده بود که پیاده‌سازی آن را عملاً غیرممکن می‌کرد. انتخاب بین یک سیستم ساده ولی کاری (مانند RFC 822) و سیستمی فوق‌العاده جالب که در عمل کار نمی‌کرد (مانند X.400) چندان دشوار نبود. این عبرت تاریخ است.

۱-۲-۷ معماری و سرویسها

در این قسمت خواهید دید که یک سیستم ایمیل چه کاری می‌تواند انجام دهد، و سازماندهی آن چگونه است. هر سیستم ایمیل دارای دو زیرسیستم است: **عامل‌کاربر** (user agent) ، که به افراد اجازه می‌دهد پیامهای خود را بفرستند و پیامهای رسیده را بخوانند، و **عامل‌انتقال‌پیام** (message transfer agent) ، که پیامها را به دست گیرنده می‌رساند. عامل کاربر برنامه‌ایست (با ظاهر معمولی) روی کامپیوتر محلی کاربر، که با سیستم ایمیل بر هم کنش دارد. در حالیکه عامل انتقال پیام معمولاً یک **سرویس** (service یا daemon) است که در پس‌زمینه اجرا می‌شود، و وظیفهٔ آن انتقال پیام در سیستم ایمیل است.

یک سیستم ایمیل، معمولاً، پنج کارکرد اصلی دارد، که آنها را در زیر توضیح می‌دهیم.

تصنیف: نوشتن پیام و جواب آن. با اینکه از هر ادیتوری می‌توان برای نوشتن پیامها استفاده کرد، ولی اغلب سیستمهای ایمیل دارای ادیتور خاص خود هستند که بسیاری از کارها (از جمله نوشتن آدرس، و سرآیند ایمیل) را بطور خودکار انجام می‌دهند. برای مثال، وقتی می‌خواهید به یک نامه جواب بدهید، سیستم ایمیل می‌تواند آدرس فرستندهٔ نامه را بطور خودکار استخراج کرده و در فیلد گیرندهٔ جوابیه (reply) قرار دهد.

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

انتقال: فرستادن پیام از فرستنده به گیرنده. این فرآیند سه مرحله دارد: تماس با ماشین گیرنده (یا یک ماشین واسط)، فرستادن پیام، و قطع ارتباط. سیستم ایمیل این کارها را بطور خودکار و بدون دخالت کاربر انجام می‌دهد.

گزارش‌دهی: مطلع کردن کاربر از سرنوشت پیام فرستاده شده. نامه تحویل شد؟ گیرنده آنرا قبول نکرد؟ در راه گم شد؟ در برخی مواقع اطمینان از رسیدن نامه بدست گیرنده اهمیت حیاتی و تبعات قانونی دارد (مانند احضاریه‌های دادگاه).

نمایش: پیامهای رسیده باید بگونه‌ای مناسب در معرض دید کاربر قرار گیرند، تا وی بتواند براحتی آنها را بخواند. گاهی لازم است برای خواندن محتویات برخی نامه‌ها (مانند نامه‌هایی که پیوست صوتی یا تصویری دارند) از برنامه‌های کمکی استفاده شود. برخی از سیستمهای ایمیل نیز نامه‌ها را بگونه‌ای خاص فرمت کرده و نمایش می‌دهند.

بایگانی: نگهداری نامه‌های رسیده. سرنوشت پیامهای رسیده متفاوت است: برخی پیامها حتی قبل از خوانده شدن دور انداخته می‌شوند، برخی فقط یک بار ارزش خواندن دارند، و برخی دیگر را باید حتماً ذخیره کرد. یک سیستم ایمیل باید بتواند نامه‌ها را به انحاء مختلف پردازش کند.

علاوه بر این سرویسهای اصلی، برخی از سیستمهای ایمیل (مخصوصاً سیستمهای رسمی) دارای ویژگیهای پیشرفتهٔ دیگری نیز هستند، که در زیر به برخی از آنها اشاره می‌کنیم.

وقتی یک فرد از محلی نقل مکان می‌کند (یا برای مدتی به مأموریت می‌رود)، باید بتوان پیامهای وی را به محل جدید **هدایت‌کرد** (forward) . سیستم ایمیل باید بتواند این کار را بصورت خودکار انجام دهد.

در اکثر سیستمها کاربران اجازه دارند برای ذخیره کردن پیامهای خود **صندوق‌پستی** (mailbox) داشته باشند. سیستم ایمیل باید فرمانهایی برای ایجاد، مدیریت و یا از بین بردن این صندوق‌ها داشته باشد.

اغلب مدیران نیاز دارند تا یک پیام را به افراد متعددی (کارمندان، مشتریان، یا شرکتهای طرف قرارداد) بفرستند. از اینجا بود که ایدهٔ **لیست‌پستی** (mailing list) – که در واقع لیستی از آدرسهای ایمیل است – پیدا شد. وقتی پیامی به یک لیست پستی فرستاده می‌شود، تمام افراد لیست کپی‌های کاملاً یکسانی از آن پیام دریافت خواهند کرد.

ویژگیهای دیگر عبارتند از: کپی (CC) ، کپی ناشناس (BCC) ، ایمیل با اولویت زیاد، ایمیل سرّی (رمز شده)، گیرندهٔ جانشین (وقتی گیرندهٔ اصلی در دسترس نبود)، و تحویل نامه‌های رئیس به منشی.

امروزه ایمیل کاربرد گسترده‌ای برای ارتباطات داخلی شرکتها و سازمانها دارد. ایمیل می‌تواند گروه گسترده‌ای از افراد (حتی آنهایی که بسیار از یکدیگر دور هستند) را در یک پروژه گرد آورد. ایمیل با حذف اکثر تمایزها (مانند مقام، سن، و جنسیت) باعث تمرکز روی اهداف می‌شود. با ایمیل، ایدهٔ درخشان یک کارآموز ساده اهمیت فزونتری نسبت به ایده‌های (اکثراً احمقانهٔ) مدیر عامل خواهد یافت.

ایدهٔ کلیدی در سیستم ایمیل تمایز بین پاکت و محتویات نامه است. پاکت نامه پیام را در خود جای می‌دهد، و شامل اطلاعاتی از قبیل آدرس گیرنده، اولویت و سطح امنیتی آن می‌شود، که معمولاً ارتباطی با محتویات نامه ندارند. عامل انتقال پیام از اطلاعات این پاکت برای جابجایی صحیح پیام استفاده می‌کند (درست مثل ادارهٔ پُست).

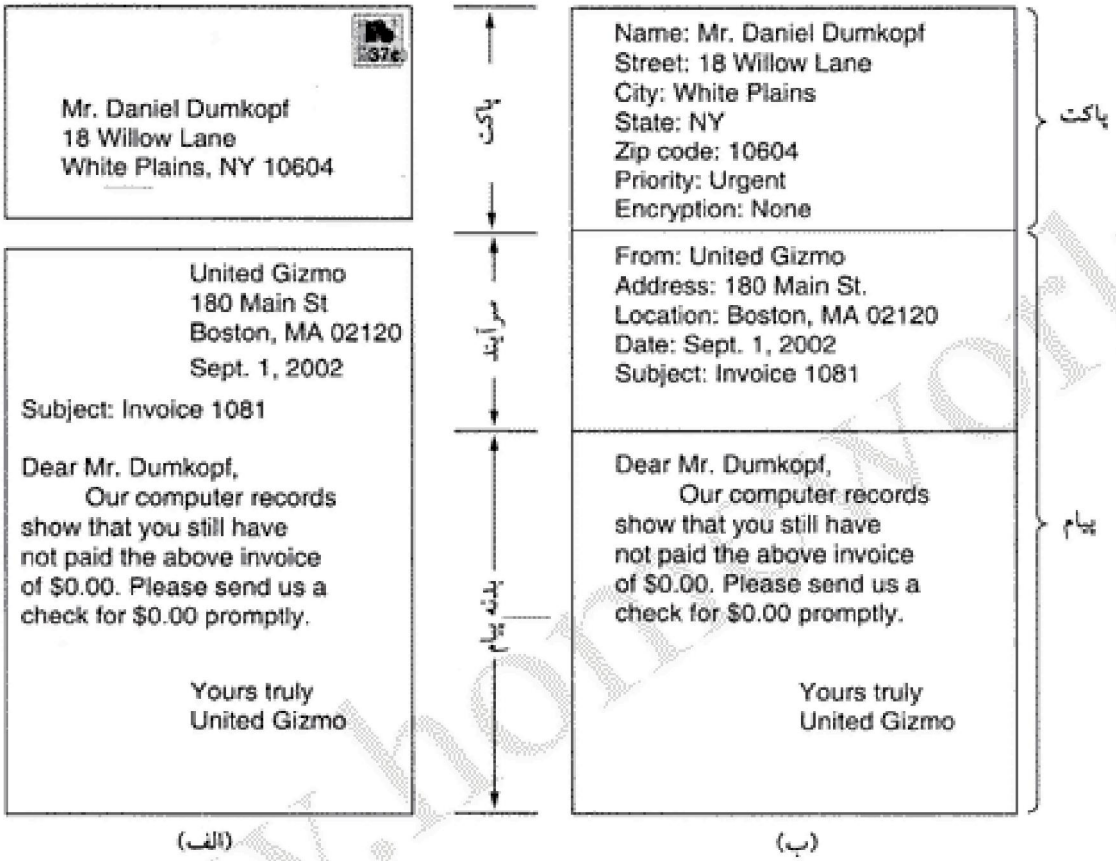
محتویات پاکت دو بخش دارد: سرآیند (header) و بدنه (body) . سرآیند شامل اطلاعات کنترلیست که عامل کاربر از آنها برای کار خود استفاده می‌کند. بدنه بخشی از پیام است که به گیرنده مربوط می‌شود. در شکل ۷-۷ رابطهٔ پاکت و پیام نشان داده شده است.

۷-۲-۲ عامل کاربر

همانطور که گفتیم، سیستمهای ایمیل دو بخش اصلی دارند: عامل کاربر، و عامل انتقال پیام. در این قسمت بخش

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

اول (یعنی، عامل کاربر) را بررسی خواهیم کرد. عامل کاربر یک برنامهٔ معمولیست (و گاهی به آن نامه‌خوان - mail reader - نیز می‌گویند)، که می‌تواند در نوشتن پیامها، خواندن نامه‌های رسیده، پاسخ به آنها و کارهایی از این قبیل به کاربر کمک کند. طیف وسیع و متنوعی از برنامه‌های عامل کاربر وجود دارد، اما کارکرد اصلی آنها بسیار شبیه یکدیگر است.



شکل ۷-۷. پاکت و پیام در (الف) پُست کاغذی، (ب) پُست الکترونیک.

فرستادن ایمیل

برای فرستادن یک ایمیل، کاربر باید ابتدا متن نامه را نوشته و آدرس گیرنده را هم مشخص کند. برای نوشتن نامه می‌توان از ادیتورها یا واژه‌پردازهای مستقل، و یا ادیتوری که به همراه عامل کاربر می‌آید، استفاده کرد. آدرس گیرنده باید با فرمتی باشد که برای عامل کاربر آشناست؛ بسیاری از آنها آدرسها را با فرمت *user@dns-address* می‌پذیرند. با فرمت نامهای DNS در ابتدای همین فصل آشنا شدید.

غیر از آدرسهای DNS فرمتهای دیگری هم برای آدرسهای ایمیل وجود دارد، که بویژه نوع X.400 آن قابل توجه است. آدرسهای X.400 تفاوت قابل توجهی با آدرسهای DNS دارند. هر آدرس X.400 مجموعه ایست از زوجهای *value* = *attribute* که با / از هم جدا می‌شوند:

/C=US/ST=MASSACHUSETTS/L=CAMBRIDGE/PA=360 MEMORIAL DR./CN=KEN SMITH/

در این آدرس بترتیب از چپ براست کشور (C) ، ایالت (ST) ، محل (L) ، آدرس (PA) ، و نام شخص (CN) نوشته می‌شود. مشخصات دیگری (مانند شرکت و شغل) نیز وجود دارد، که با استفاده از آنها می‌توانید پیام خود را حتی به فردی که آدرس ایمیل او را نمی‌دانید، بفرستید. با اینکه آدرسهای X.400 بسیار غریب‌تر از نامهای

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

DNS هستند، اما اغلب سیستمهای ایمیل اجازه می‌دهند تا هر آدرس یک نام مستعار ساده (موسوم به nickname) داشته باشد. بدین‌ترتیب دیگر لازم نیست کاربر آدرس ایمیل (حتی X.400) را بطور کامل وارد کند.

اغلب سیستمهای ایمیل از لیست پُستی پشتیبانی می‌کنند، بنابراین می‌توان یک نامه را یکباره برای تعداد زیادی از افراد فرستاد. اگر لیست پستی بصورت محلی نگهداری شود، ایمیل در همان مبدأ بین افراد آن لیست توزیع خواهد شد. ولی اگر لیست پستی در محل دیگری نگهداری شود، پیام در آنجا پخش می‌شود. برای مثال، اگر گروهی از طرفداران حیات وحش یک لیست پستی بنام *birders* در دانشگاه آریزونا داشته باشند *(birders@meadowlark.arizona.edu)* ، نامه‌ای که به این آدرس فرستاده شده باشد، فقط بعد از رسیدن به دانشگاه آریزونا بین افراد آن توزیع خواهد شد. آنهایی که در یک لیست پستی قرار دارند، بهیچوجه متوجه این مطلب نخواهند شد؛ آنها هم مانند سایر افراد پیامهایی از یک فرستندهٔ مشخص دریافت می‌کنند.

خواندن ایمیل

معمولاً، وقتی عامل کاربر کار خود را شروع می‌کند، قبل از هر چیز صندوق پستی کاربر را چک کرده و نامه‌های رسیده را از آنجا برمی‌دارد. پس از آن، تعداد نامه‌های رسیده را به کاربر اعلام کرده، و احتمالاً آنها را بصورت خلاصه (شامل اطلاعاتی از قبیل فرستنده، تاریخ دریافت، و موضوع نامه) به کاربر نمایش می‌دهد، و سپس منتظر اقدام بعدی کاربر می‌ماند.

بعنوان مثال، یک سناریوی ساده را در زیر بررسی می‌کنیم. بعد از شروع برنامهٔ عامل کاربر، اولین اقدام معمولاً نمایش خلاصه‌ای از نامه‌های رسیده است. این خلاصه می‌تواند چیزی شبیه شکل ۷-۸ باشد: هر خط مشخصات یک پیام را نشان می‌دهد. در این مثال، هشت پیام در صندوق پستی وجود دارد.

#	پرچم	بایت	فرستنده	موضوع
1	K	1030	asw	Changes to MINIX
2	KA	6348	trudy	Not all Trudys are nasty
3	K F	4519	Amy N. Wong	Request for information
4		1236	bal	Bioinformatics
5		104110	kaashoek	Material on peer-to-peer
6		1223	Frank	Re: Will you review a grant proposal
7		3110	guido	Our paper has been accepted
8		1204	dmr	Re: My student's visit

شکل ۷-۸ نمایش محتویات صندوق پستی.

در هر خط اطلاعات مختلفی دیده می‌شود. در یک سیستم ایمیل ساده این اطلاعات ثابت است، ولی در سیستمهای پیچیده‌تر کاربر می‌تواند نحوهٔ نمایش اطلاعات را بنا به میل خود تغییر دهد (و این تنظیمات را در فایلی بنام پروفایل کاربر – user profile – ذخیره کند). در مثال بالا، اولین فیلد شمارهٔ پیام است. فیلد دوم، *Flags* ، مقادیر مختلفی می‌تواند بگیرد: *K* یعنی این پیام آرشیوی است (جدید نیست و قبلاً خوانده شده)؛ *A* یعنی به این پیام پاسخ داده شده است؛ *F* یعنی این پیام به فرد دیگری هدایت شده است (forward) . پرچمهای دیگری نیز در این فیلد می‌تواند وجود داشته باشد، که هر کدام معنای خاص خود را دارند.

فیلد سوم طول پیام، و فیلد چهارم فرستندهٔ آنرا نشان می‌دهند. از آنجائیکه این فیلد از نامهٔ رسیده استخراج

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

می‌شود، محتویات آن به نامهٔ فرستاده شده بستگی دارد. و بالاخره، در فیلد **موضوع** خلاصه‌ای از محتویات نامه را می‌بینید. سعی کنید همیشه نامه‌هایتان «موضوع» داشته باشد، چون تجربه نشان داده که نامه‌های بدون موضوع براحتی نادیده گرفته می‌شوند.

بعد از خواندن خلاصهٔ نامه، کاربر می‌توان اقدامات مختلفی روی آن انجام دهد: نامه را باز کند، آنرا حذف کند، به نامه جواب دهد، آنرا برای کس دیگری بفرستد، و مانند آن. برای هر یک از این اعمال، فرمان خاصی در برنامهٔ عامل کاربر وجود دارد.

سیستمهای ایمیل امروزی خیلی بیش از انتقال فایل ساده‌اند. مدیریت حجم زیادی از نامه‌ها با برنامه‌های امروزی کار چندان دشواری نیست؛ و برای افرادی که در سال هزاران ایمیل رد و بدل می‌کنند، این مزیت کوچکی نیست.

۷-۲-۳ فرمت پیامها

اجازه دهید کمی هم دربارهٔ فرمت پیامهای ایمیل صحبت کنیم. ابتدا به ایمیلهای متنی ساده با فرمت RFC 822 می‌پردازیم، و پس از آن دربارهٔ الحاقات چندرسانه‌ای در RFC 822 توضیح خواهیم داد.

RFC 822

هر پیام یک پاکت ساده (که مشخصات آن در RFC 821 آمده) دارد، بعلاوهٔ تعدادی سرآیند، یک خط خالی، و بعد از آن متن یا بدنهٔ پیام. هر فیلد سرآیند عبارتست از یک خط متن ASCII مشتمل بر: نام فیلد، علامت : ، و مقدار فیلد (که البته برخی از فیلدها می‌توانند مقدار نداشته باشند). در استاندارد RFC 822 که بیش از دو دهه از عمر آن می‌گذرد، تمایز آشکاری بین فیلدهای پاکت نامه و فیلدهای سرآیند وجود ندارد. با اینکه در RFC 2822 این مشکل مرتفع شده، ولی بعلت رواج گستردهٔ آن در عمل چنین اتفاقی نیفتاده است. در حالت عادی، عامل انتقال پیام با استخراج اطلاعات از نامه‌ای که از عامل کاربر دریافت می‌کند، پاکت نامه را می‌سازد، و بهمین دلیل پاکت و نامه با هم مخلوط می‌شوند.

مهمترین فیلدهای سرآیند که نقش مهمی در انتقال پیام دارند، در شکل ۷-۹ نشان داده شده است. در فیلد *TO:* آدرس DNS گیرندهٔ اصلی پیام نوشته می‌شود. نوشتن چندین گیرنده در این فیلد مجاز است. در فیلد *Cc:* آدرس گیرنده‌های ثانویهٔ پیام نوشته می‌شود. از نظر تحویل پیام در سیستم ایمیل، تفاوتی بین گیرنده‌های اصلی و ثانویه وجود ندارد؛ این فیلد بیشتر برای انسانها مهم است تا برای ماشین. اصطلاح *Cc* (کپی کربنی) قدری قدیمی است، چون در کامپیوترها چیزی بنام کاغذ کپی وجود ندارد، اما این اصطلاح دیگر کاملاً جا افتاده است. فیلد *Bcc:* (کپی کربنی ناپیدا) شبیه *Cc:* است، با این تفاوت که این فیلد در نامه‌های کپی شده حذف می‌شود، و گیرندهٔ نامه از هویت سایر گیرنده‌ها (و حتی وجود چنین گیرنده‌هایی) مطلع نخواهد شد.

سرآیند	مفهوم
To:	آدرس ایمیل گیرنده های اصلی
Cc:	آدرس ایمیل گیرنده های ثانویه (کپی)
Bcc:	آدرس ایمیل گیرنده های کپی‌های ناشناس
From:	فرستنده پیام
Sender:	آدرس ایمیل فرستنده
Received:	هر عامل انتقال واسطه در بین راه مشخصات خود را اضافه می کند
Return-Path:	می توان از آن برای مشخص کردن مسیر برگشت به فرستنده استفاده کرد

شکل ۷-۹. فیلدهای سرآیند RFC 822 برای انتقال پیام.

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

دو فیلد بعدی، یعنی *From:* و *Sender:* ، بترتیب نویسنده و فرستندهٔ نامه را مشخص می‌کنند. این دو الزاماً یکی نیستند (ولی اغلب چنین است). برای مثال، نویسندهٔ نامه می‌تواند مدیر عامل باشد *(From:)* ، ولی منشی شرکت آنرا بفرستد *(Sender:)* . فیلد *From:* حتماً باید پُر شود، ولی فیلد *Sender:* (اگر با *From:* یکی باشد) می‌تواند خالی رها شود. اگر احتمال می‌دهید نامه بدست گیرنده نمی‌رسد و برگشت می‌خورد، حتماً فیلد *Sender:* را پُر کنید، چون نامه‌های غیرقابل‌تحویل به این آدرس برگشت داده می‌شوند.

وقتی یک نامه از واسطه‌های مختلفی عبور می‌کند تا به دست گیرنده برسد، نام هر واسطه در یک فیلد *Received:* جداگانه نوشته می‌شود. در این فیلد نام عامل گیرنده، تاریخ و زمان دریافت پیام، و اطلاعات دیگر ثبت می‌شود. از این اطلاعات می‌توان برای رفع اشکالاتی که در طی تحویل نامه‌ها پیش می‌آید، استفاده کرد.

فیلد *Return-Path:* که توسط آخرین عامل انتقال پیام اضافه می‌شود، نشان می‌دهد که مسیر برگشت به فرستنده چگونه است. از نظر تئوری این اطلاعات باید شامل تمام سرآیندهای *Received:* (بجز صندوق پستی فرستنده) باشد، ولی بندرت چنین است و معمولاً فقط آدرس فرستنده در آن نوشته می‌شود.

علاوه بر فیلدهای شکل ۷-۹، پیامهای RFC 822 دارای سرآیندهای دیگری نیز هستند که به بیشتر کار عامل کاربر یا شخص گیرنده می‌آیند. در شکل ۷-۱۰ برخی از این سرآیندها را می‌بینید. کارکرد اغلب این فیلدها از روی نامشان پیداست، و نیازی به توضیح زیاد ندارند.

سرآیند	مفهوم
Date:	زمان و تاریخ ارسال
Reply-To:	آدرس ایمیل برای پاسخ نامه
Message-Id:	عدد منحصر بفرد شناسایی پیام
In-Reply-To:	شماره پیامی که این پاسخ پیام پاسخ آن است
References:	سایر شماره های مربوطه
Keywords:	کلمات کلیدی انتخاب شده توسط کاربر
Subject:	موضوع پیام

شکل ۷-۱۰. برخی از فیلدهای سرآیند RFC 822 .

فیلد *Reply-To:* برای مواقعیست که نویسنده و گیرندهٔ نامه هیچکدام نمی‌خواهند گیرندهٔ پاسخ نامه باشند. برای مثال، وقتی مدیر بازاریابی نامه‌ای دربارهٔ محصولات جدید شرکت به یک مشتری می‌نویسد، و منشی هم آنرا می‌فرستد، فیلد *Reply-To:* می‌تواند به آدرس قسمت فروش شرکت، که پاسخگوی سفارشات هستند، اشاره کند. این فیلد برای مواردی که فرستنده دو آدرس ایمیل دارد، و مایل است پاسخها را از طریق آدرس دیگرش بگیرد، نیز مفید است.

استاندارد RFC 822 اجازه می‌دهد تا کاربران سرآیندهای دلخواهشان را به نامه‌ها اضافه کنند، مشروط باینکه این سرآیندها با *X-* شروع شوند (هیچیک از سرآیندهای رسمی این استاندارد با *X-* شروع نمی‌شوند، و در آینده نیز نخواهند شد). این قبیل سرآیندها می‌توانند اطلاعات اضافی را با خود حمل کنند.

بعد از سرآیند، بدنهٔ نامه می‌آید. کاربران می‌توانند هر چیزی در این بدنه بنویسند. برخی افراد در انتهای نامه‌هایشان اختتامیه‌های ماهرانه‌ای می‌آورند، مانند اشکال کارتونی جالب و خنده‌دار، کلمات قصار مشاهیر، و یا عبارات قانونی سلب مسئولیت (مثلاً، «شرکت فلان و بهمان هیچگونه مسئولیتی را دربارهٔ محتویات این نامه نمی‌پذیرد.»).

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

MIME - الحاقات چندمنظورهٔ پُست اینترنت

در روزهای اولیهٔ آرپانت، ایمیل‌ها فقط متن ساده بود که به زبان انگلیسی و با فرمت ASCII نوشته می‌شد. استاندارد RFC 822 با این وضعیت هیچ مشکلی نداشت: کاربر می‌توانست هر چیزی که می‌خواست در بدنهٔ نامه بنویسد. اما این روش دیگر برای دنیای امروز کافی نیست. برخی از مشکلات ذاتی این سیستم عبارتند از:

۱. ارسال پیام به زبانهایی که اِعراب دارند (مانند فرانسه و آلمانی).
۲. ارسال پیام به زبانهای غیرلاتین (مانند عربی و روسی).
۳. ارسال پیام به زبانهای غیرالفبایی (مانند چینی و ژاپنی).
۴. ارسال پیامهایی که اصلاً متن نیستند (مانند صدا و تصویر).

راه حل این مشکلات در RFC 1341 ارئه شد، و بعدها در RFC 2045-49 به روز در آمد. این راه حل که MIME (الحاقات‌چندمنظورهٔ‌پُست‌اینترنت – Multipurpose Internet Mail Extensions) نام دارد، امروزه بطور گسترده‌ای رواج یافته است.

ایدهٔ اصلی MIME عبارتست از: ادامهٔ استفاده از فرمت RFC 822 ، و اضافه کردن ساختاری جدید به بدنهٔ پیام و تعریف قواعد درج پیامهای غیرمتنی. پیامهای MIME با برنامه‌ها و پروتکلهای موجود ایمیل کاملاً سازگارند، چون از استاندارد RFC 822 تخطی نمی‌کنند. فقط برنامه‌های عامل کاربر باید عوض شوند، که این کار را هم کاربران می‌توانند براحتی انجام دهند.

MIME پنج سرآیند جدید تعریف می‌کند، که آنها را در شکل ۷-۱۱ مشاهده می‌کنید. اولین سرآیند به عامل کاربر دریافت کنندهٔ پیام می‌گوید که با یک پیام MIME سروکار دارد، و ویرایش آن هم اعلام می‌شود. هر پیامی که سرآیند *MIME-Version:* نداشته باشد، متن ساده تلقی شده و به همان طریق پردازش می‌شود.

سرآیند	مفهوم
MIME-Version:	ویرایش MIME پیام
Content-Description:	جمله ای درباره محتویات پیام
Content-Id:	عدد منحصر بفرد شناسایی محتویات پیام
Content-Transfer-Encoding:	نحوه کدگذاری پیام
Content-Type:	نوع و فرمت محتویات پیام

شکل ۷-۱۱. سرآیندهای اضافی MIME در RFC 822 .

سرآیند *Content-Description:* یک عبارت متنی است که می‌گوید چه چیزی در پیام وجود دارد. از روی این سرآیند است که گیرنده تشخیص می‌دهد آیا محتویات پیام ارزش خواندن دارد یا خیر. برای مثال، اگر سرآیند *Content-Description:* بگوید: «این عکس یک موش است»، و گیرنده علاقه‌ای به دیدن عکس موش نداشته باشد، برنامه پیام را دور انداخته و تلاشی برای نمایش این عکس نخواهد کرد.

سرآیند *Content-Id:* محتویات پیام را مشخص می‌کند. فرمت این سرآیند مانند *Message-Id:* است.

سرآیند *Content-Transfer-Encoding:* نحوهٔ کُد شدن محتویات پیام (برای گذر از شبکه‌هایی که فقط به متن ساده اجازهٔ عبور می‌دهند) را مشخص می‌کند. پنج نوع کُدگذاری پیام وجود دارد. نوع اول فقط متن سادهٔ ASCII است. کاراکترهای ASCII هفت‌بیتی هستند، و تمام پروتکلهای ایمیل می‌توانند خطوط متن را (مشروط بر اینکه هر خط از ۱۰۰۰ حرف تجاوز نکند) منتقل کنند.

نوع دوم در واقع همان نوع قبلیست، که فقط از کاراکترهای ASCII ۸بیتی (از 0 تا 255) استفاده می‌کند.

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

برخی از بخشهای اینترنت از این کُد برای ارسال متن (و نمایش کاراکترهای خاص) استفاده می‌کنند. این کُدگذاری در پروتکلهای ایمیل اینترنتی مجاز نیست (و این تعریف هم باعث مجاز شدن آن نمی‌شود)، ولی حداقل توضیح می‌دهد که اشکال کار از کجاست. پیامهای دارای کُدگذاری ۸بیتی هم محدود به خطهای ۱۰۰۰ حرفی هستند.

بدتر از آن پیامهایی هستند که کُدگذاری باینری دارند. اینها فایلهای باینری هستند، که نه تنها از تمام حالات ممکنهٔ ۸ بیت استفاده می‌کنند، بلکه به محدودیت ۱۰۰۰ بایت نیز وقعی نمی‌گذارند. برنامه‌های اجرایی در این دسته قرار می‌گیرند. هیچ تضمینی وجود ندارد که پیامهای باینری درست به مقصد برسند، ولی بسیاری افراد همچنان کار خودشان را می‌کنند.

یکی از روشهای کُدگذاری صحیح پیامهای باینری روش کُدگذاری base64 (که گاهی ASCII armor نیز گفته می‌شود) است. در این روش، هر دستهٔ ۲۴ بیتی به چهار واحد ۶بیتی تقسیم شده، و هر واحد بعنوان یک کاراکتر معتبر ASCII فرستاده می‌شود. در این روش "A" معادل 0 است، "B" معادل 1 ، ... و بالاخره "Z" معادل 26 ؛ پس از آن ۲۶ حرف کوچک انگلیسی می‌آیند، و پس از آن ارقام 0 تا 9 ؛ 62 و 63 نیز بترتیب معادل + و / هستند. توالیهای == و = بترتیب نشان می‌دهند که آخرین گروه ۸ یا ۱۶ بیتی است. کاراکترهای «برگشت سر خط» (carriage return) و «خط بعدی» (line feed) نیز بکلی نادیده گرفته می‌شود، بنابراین می‌توان برای کوتاه کردن خطوط از آنها استفاده کرد. با این روش می‌توان فایلهای باینری را بطور صحیح ارسال کرد.

برای پیامهایی که تقریباً بطور کامل ASCII هستند و فقط چند کاراکتر غیر ASCII دارند، کُدگذاری base64 کارایی مطلوبی ندارد. برای این قبیل پیامها از کُدگذاری quoted-printable استفاده می‌شود. این در واقع همان روش ASCII ۷بیتی است، که در آن کاراکترهای بالای 127 با علامت = و یک عدد هگزادسیمال دو رقمی مشخص می‌شوند.

اطلاعات باینری همواره باید با یکی از روشهای base64 یا quoted-printable فرستاده شوند. اگر دلیل موجهی برای استفاده نکردن از هر یک از روشها وجود داشته باشد، می‌توان از کُدگذاریهای خاص (که با سرآیند *Content-Transfer-Encoding:* مشخص می‌شوند) استفاده کرد.

آخرین آیتم شکل ۷-۱۱ در واقع مهمترین سرآیند MIME است. این سرآیند خصلت واقعی محتویات پیام را مشخص می‌کند. در RFC 2045 هفت نوع (type) تعریف شده، که هر کدام از آنها می‌توانند چندین زیرنوع (subtype) داشته باشند. نوع و زیرنوع با یک / از هم جدا می‌شوند:

```
Content-Type: video/mpeg
```

زیرنوع باید صریحاً در سرآیند مشخص شود: هیچ مقداری بعنوان پیش‌فرض وجود ندارد. لیست اولیهٔ نوع‌ها و زیرنوع‌های RFC 2045 را در شکل ۷-۱۲ ملاحظه می‌کنید. از آن زمان تاکنون آیتمهای دیگری اضافه شده است، و در آینده باز هم اضافه خواهد شد.

اجازه دهید این لیست را مختصراً بررسی کنیم. نوع *text* همان متن ASCII ساده است. زیرنوع *text/plain* به گیرنده می‌گوید پیام را بهمان صورتی که دریافت کرده (بدون هیچ فرمت یا پردازشی) نمایش دهد. این گزینه اجازه می‌دهد تا پیامهای معمولی بدون هیچ اقدام اضافه‌ای به پیامهای MIME تبدیل شوند.

زیرنوع *text/enriched* اجازه می‌دهد تا از زبانهای علامتگذاری ساده برای فرمت کردن متن (تعیین فونت، اندازه، رنگ و صفحه‌بندی) استفاده شود. این زیرنوع زیرمجموعه‌ای از SGML (زبان علامتگذاری عمومی استاندارد – که زبان استاندارد وب یعنی HTML نیز جزیی از آن محسوب می‌شود) است. برای مثال پیام

```
The <bold> time </bold> has come the <italic> walrus </italic> said ...
```

به این صورت به نمایش در خواهد آمد:

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

نوع	زیرنوع	توضیح
Text	Plain	متن فرمت نشده
	Enriched	متن با فرمت ساده
Image	Gif	تصاویر با فرمت GIF
	Jpeg	تصاویر با فرمت JPEG
Audio	Basic	صدا
Video	Mpeg	تنظیم با فرمتMPEG
Application	Octet-stream	توالی بایت تفسیر نشده
	Postscript	سند چاپی با فرمت پست اسکریپت
Message	Rfc822	پیام RFC 822
	Partial	سند چند تکه شده
	External-body	پیام باید از اینترنت گرفته شود
Multipart	Mixed	پیامی با چند بخش مستقل
	Alternative	یک پیام با فرمتهای مختلف
	Parallel	بخشهای پیام باید همزمان دیده شوند
	Digest	هر بخش یک پیام RFC 822 کامل است

شکل ۷-۱۲. نوع‌ها و زیرنوع‌های MIME تعریف شده در RFC 2045 .

The **time** has come the *walrus* said ...

فرمت کردن پیام بطور کامل بر عهدهٔ سیستم گیرنده است، و نباید انتظار داشته باشید چنین پیامی در تمام سیستمها یکسان (و آنطوری که شما تصور می‌کنید) دیده شود. گاهی یک سیستم معنای کاری را که شما خواسته‌اید نمی‌فهمد، و بجای آن کار دیگری انجام می‌دهد.

بعد از رواج وب، زیرنوع جدیدی بنام *text/html* (در RFC 2854) اضافه شد، که اجازه می‌داد صفحات وب از طریق ایمیل‌های RFC 822 فرستاده شوند. در RFC 3023 نیز زیرنوع دیگری (*text/xml*) برای ارسال پیامهای XML تعریف شده است. در ادامهٔ این فصل با HTML و XML بیشتر آشنا خواهید شد.

نوع MIME بعدی *image* است، که برای ارسال تصاویر ثابت بکار می‌رود. امروزه فرمتهای بسیار متنوعی برای ذخیره و ارسال کردن تصاویر (با فشرده‌سازی یا بدون آن) وجود دارد، که از میان آنها فرمتهای GIF و JPEG بسیار معروفند و تقریباً تمام مرورگرهای اینترنت از آنها پشتیبانی می‌کنند. (البته بعدها فرمتهای دیگری نیز به این لیست اضافه شد.)

نوع‌های *audio* و *video* بترتیب برای ارسال صدا و تصاویر متحرک هستند. توجه داشته باشید که نوع *video* فقط برای ارسال اطلاعات تصویری است، و اگر فایل شما دارای تراک صوتی هم باشد، باید آنها را جداگانه بفرستید. اولین فرمت ویدئویی که قابلیت ارسال از طریق ایمیل را پیداکرد، فرمت MPEG (گروه تخصصی تصاویر متحرک - Moving Picture Expert Group) بود؛ بعدها فرمتهای دیگر نیز اضافه شد. علاوه بر زیرنوع *audio/basic* ، در RFC 3003 زیرنوع دیگری (*audio/mpeg*) برای ارسال فایلهای صوتی MP3 تعریف شده است.

هر نوع فایل باینری دیگری که در دستجات بالا قرار نگیرد (و سیستم ایمیل نداند آنرا چگونه پردازش کند)، در ذیل نوع *application* دسته‌بندی خواهد شد. زیرنوع *octet-stream* فقط استریمی از بایتهاست (و سیستم هیچگونه تفسیری روی آنها نخواهد کرد). عامل کاربر بعد از دریافت این استریم، تنها کاری که می‌تواند انجام دهد ذخیره کردن آن بصورت یک فایل است - پس باید نام فایل را از کاربر بگیرد. پردازشهای بعدی نیز بر عهدهٔ کاربر است.

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

زیرنوع تعریف شدهٔ دیگر *postscript* است، که به زبان پُست اسکریپت (زبان توصیف صفحات چاپی، از شرکت Adobe) مربوط می‌شود. بسیاری از چاپگرهای امروزی دارای مفسرهای پُست اسکریپت هستند. با اینکه عامل کاربر می‌تواند تفسیر فایلهای پُست اسکریپت را بر عهدهٔ برنامه‌های خارجی بگذارد، اما این کار خالی از خطر نیست. پُست اسکریپت یک زبان برنامه‌نویسی کامل است، و یک برنامه‌نویس (خودآزار) می‌تواند با صرف وقت کافی حتی یک کامپایلر C یا سیستم مدیریت پایگاه داده با آن بنویسد. عامل کاربر برای نمایش محتویات فایل پُست‌اسکریپت، در واقع برنامه‌ای راکه در دل پیام فرستاده شده اجرا می‌کند. چنین برنامه‌ای حتی می‌تواند (در کنار نمایش یک متن ساده) فایلهای کاربر را تغییر داده، یا آنها را پاک کند (و یا دهها کار ناجور دیگر انجام دهد).

نوع *message* اجازه می‌دهد تا یک پیام را بطور کامل در دل پیام دیگر جای دهیم. این نوع بویژه برای هدایت پیامها (message forwarding) مفید است. برای قرار دادن یک ایمیل RFC 822 در دل پیام دیگر، می‌توان از زیرنوع *message/rfc822* استفاده کرد.

با زیرنوع *partial* می‌توان یک پیام را چند تکه کرده، و هر تکه را در دل یک ایمیل جداگانه قرار داد (که این برای پیامهای خیلی بزرگ مناسب است). در مقصد سیستم ایمیل می‌تواند با استفاده از پارامترهای این زیرنوع، پیام تکه‌تکه شده را دوباره به هم بچسباند.

بالاخره، از زیرنوع *external-body* می‌توان برای پیامهای بسیار بزرگ (مانند فیلمهای ویدئویی) استفاده کرد؛ یعنی بجای قرار دادن فایل MPEG در دل پیام، آدرس FTP آن نامه در نوشته می‌شود، و عامل کاربر می‌تواند در موقع نیاز به این آدرس مراجعه کرده و فایل را بخواند. این زیرنوع بخصوص در مواردی که بخواهیم فایل بزرگی را برای یک لیست پستی بفرستیم، بسیار ایده‌آل است، چون احتمال اینکه همهٔ اعضاء لیست پستی این فایل را بخواهند چندان زیاد نیست (فرستادن آگهی‌های تبلیغاتی یکی از این موارد است).

نوع آخر *multipart* است، که اجازه می‌دهد یک پیام چندین بخش داشته باشد (بخشهایی که ابتدای و انتهای آنها کاملاً مشخص است). در زیرنوع *mixed* این بخشها می‌توانند کاملاً متفاوت باشند، بدون اینکه نیازی به ساختار اضافی وجود داشته باشد. در بسیاری از برنامه‌های ایمیل یک پیام ساده می‌تواند چندین پیوست (attachment) از انواع مختلف داشته باشد، که این پیوستها با استفاده از نوع *multipart* فرستاده می‌شوند.

بر خلاف *multipart* ، در زیرنوع *alternative* یک پیام واحد به چندین فرمت (مانند متن ساده، متن فرمت‌دار، و یا پست اسکریپت) فرستاده می‌شود، که عامل کاربر گیرنده می‌تواند بسته به امکانات خود از یکی از این انواع استفاده کند. این انواع باید از ساده‌ترین به پیچیده‌ترین مرتب شوند، تا حتی کاربران غیر MIME بتوانند پیامها را بخوانند.

از زیرنوع *alternative* برای ارسال پیام به زبانهای مختلف هم می‌توان استفاده کرد. (سنگ روزِتا را می‌توان یکی از قدیمیترین پیامهای *multipart/alternative* دانست - سنگ روزِتا سنگ‌نبشته‌ای است که توسط سپاهیان ناپلئون در منطقه‌ای به همین نام در مصر کشف شد، و در آن یک پیام واحد به زبانهای هیروگلیف و یونانی نوشته شده بود؛ با استفاده از همین سنگ‌نبشته بود که شامپولیون باستان‌شناس فرانسوی توانست معمای خط هیروگلیف را بعد از قرنها کشف کند.)

در شکل ۷-۱۳ یک پیام *multipart* را ملاحظه می‌کنید. در این مثال، یک پیام تبریک تولد به دو صورت متن و صوت فرستاده شده است. اگر کامپیوتر گیرنده کارت صوتی داشته باشد، برنامهٔ عامل کاربر فایل صوتی *birthday.snd* را از شبکه گرفته، و پخش می‌کند. اما اگر چنین نباشد، فقط متن شعر را نمایش خواهد داد. دقت کنید که بخشهای پیام با دو - (خط تیره) و رشته‌ای از حروف (که نرم‌افزار آنها را تولید کرده، و در قسمت *boundary* مشخص شده) از هم جدا می‌شوند.

همچنین توجه کنید که سرآیند *Content-Type* در سه نقطه از این پیام آمده است. در بالای پیام، سرآیند

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي


```
From: elinor@abcd.com
To: carolyn@xyz.com
MIME-Version: 1.0
Message-Id: <0704760941.AA00747@abcd.com>
Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm
Subject: Earth orbits sun integral number of times

This is the preamble. The user agent ignores it. Have a nice day.

--qwertyuiopasdfghjklzxcvbnm
Content-Type: text/enriched

Happy birthday to you
Happy birthday to you
Happy birthday dear <bold> Carolyn </bold>
Happy birthday to you

--qwertyuiopasdfghjklzxcvbnm
Content-Type: message/external-body;
        access-type="anon-ftp";
        site="bicycle.abcd.com";
        directory="pub";
        name="birthday.snd"

content-type: audio/basic
content-transfer-encoding: base64
--qwertyuiopasdfghjklzxcvbnm--
```

شکل ۷-۱۳. یک پیام *multipart*، محتوی متن و صوت.

Content-Type می‌گوید که این یک پیام *multipart/alternative* است. در دو *Content-Type* بعدی نوع و زیرنوع هر بخش مشخص می‌شود. در آخرین *Content-Type* نیز کُدگذاری فایل صوتی را مشخص کرده‌ایم، چون هر چیزی که متن ASCII ۷بیتی نباشد، باید دارای کُدگذاری مشخص باشد.

پیامهای *multipart* دارای دو زیرنوع دیگر نیز می‌توانند باشند. از زیرنوع *parallel* وقتی استفاده می‌کنیم که بخواهیم تمام بخشهای پیام همزمان «مشاهده» شوند. برای مثال، فیلمهای ویدئویی دارای کانالهای تصویری و صوتی مجزا هستند، که باید همزمان پخش شوند (نه بدنبال هم).

زیرنوع *digest* وقتی بکار برده می‌شود که بخواهیم تعدادی پیام را در یک پیام مرکب بسته‌بندی کنیم. برای مثال، در **گروههای مباحثه** (discussion group) اینترنتی معمولاً چندین پیام که از اعضای مختلف جمع‌آوری شده، در یک پیام *multipart/digest* به سایر اعضای گروه فرستاده می‌شود.

۷-۲-۴ انتقال پیام

سیستم **انتقال پیام** (message transfer) با ارسال پیام از فرستنده به گیرنده سروکار دارد. ساده‌ترین راه برای این کار، برقراری یک اتصال مستقیم از ماشین فرستنده به ماشین گیرنده، و انتقال پیام است. بعد از بررسی این روش، به مواردی می‌پردازیم که چنین کاری امکان ندارد، و سپس برای آن موارد نیز راه‌حل‌هایی نشان خواهیم داد.

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

SMTP - پروتکل سادهٔ انتقال نامه

در اینترنت، انتقال ایمیل با برقراری یک اتصال TCP از ماشین مبدأ به پورت 25 ماشین مقصد صورت می‌گیرد. برنامه‌ای که به این پورت گوش می‌کند، دیمون SMTP (پروتکل سادهٔ انتقال نامه - Simple Mail Transfer Protocol) نام دارد. این دیمون اتصالات ورودی را پذیرفته، و پیامها را در صندوق پستی مربوطه کپی می‌کند. اگر گیرنده نتواند پیامی را تحویل گیرد، یک گزارش خطا حاوی اولین بخش از پیام مزبور به فرستنده برمی‌گرداند.

SMTP یک پروتکل سادهٔ ASCII است. بعد از برقراری اتصال TCP ، ماشین فرستنده (که نقش مشتری را بازی می‌کند) منتظر می‌ماند تا ماشین گیرنده (که نقش سرویس‌دهنده را بازی خواهد کرد) شروع به صحبت کند. در شروع، سرویس‌دهنده یک خط متن فرستاده و ضمن معرفی خود، اعلام می‌کند که آیا آماده دریافت ایمیل هست یا خیر. اگر سرویس‌دهنده آماده نباشد، مشتری ارتباط را قطع کرده، و بعداً دوباره سعی خواهد کرد.

اگر سرویس‌دهنده آمادهٔ دریافت باشد، مشتری اعلام می‌کند که پیام از چه کسی می‌آید و به چه کسی باید تحویل شود. اگر چنین دریافت‌کننده‌ای در ماشین سرویس‌دهنده وجود داشته باشد، سرویس‌دهنده از مشتری می‌خواهد که پیام را بفرستد. پس از آن که مشتری پیام را فرستاد، سرویس‌دهنده دریافت آن را تصدیق می‌کند. سرویس‌دهنده هیچ تلاشی برای چک کردن جمع تطبیقی انجام نخواهد داد، چون TCP سالم بودن ارتباط را تضمین می‌کند. اگر باز هم پیامی وجود داشته باشد، پس از آن فرستاده می‌شود. وقتی تمام ایمیل‌ها (در هر دو جهت) مبادله شد، ارتباط قطع می‌شود. در شکل ۷-۱۴ مکالمهٔ سرویس‌دهنده و مشتری (منجمله کُدهای عددی SMTP) برای ارسال ایمیل شکل ۷-۱۳ را ملاحظه می‌کنید. خطوطی که توسط مشتری ارسال شده‌اند، را با C: ، و آنهایی که توسط سرویس‌دهنده فرستاده شده‌اند، را با S: مشخص کرده‌ایم.

کمی توضیح دربارهٔ شکل ۷-۱۴ می‌تواند مفید باشد. اولین پیام مشتری *HELO* است. این کلمه در واقع مخفف چهارحرفی «سلام» *(HELLO)* است، و پیداست که از همهٔ چهارحرفی‌ها به آن شبیه‌تر است. اینکه چرا باید از کلمه‌های چهارحرفی استفاده کنیم، در غبار زمان گم شده، ولی این رسم همچنان پا برجاست.

از آنجائیکه فقط یک گیرنده وجود دارد، در اینجا فقط یک دستور *RCPT* دیده می‌شود، ولی می‌توان در آن واحد یک پیام را به چندین گیرنده فرستاد (که قبول یا رد درخواست برای هر یک جداگانه انجام خواهد شد).

با اینکه دستورات چهارحرفی از طرف مشتری ثابت و مشخص هستند، پاسخ سرویس‌دهنده فقط یک سری کُدهای عددی است. در واقع همین کُدهاست که اهمیت دارد، و هر سیستم می‌تواند برای هر کُد توضیح خاص خود را داشته باشد.

اجازه دهید برای درک بهتر پروتکل SMTP ، کمی دربارهٔ فرآیند کار توضیح دهیم. قبل از هر چیز سراغ کامپیوتری بروید که به اینترنت دسترسی دارد. سپس دستور زیر را در خط فرمان وارد کنید (در این مثال فرض کرده‌ایم سیستم عامل یونیکس است)

```
telnet mail.isp.com 25
```

(بجای mail.isp.com آدرس IP یا نام DNS سیستم ایمیل خود را وارد کنید.) در سیستمهای ویندوز، پنجرهٔ Start|Run را باز کرده و دستور فوق را در آنجا وارد کنید. این دستور یک اتصال telnet (یعنی، TCP) با پورت 25 ماشین مشخص شده برقرار می‌کند. پورت 25 پورت SMTP است (شکل ۶-۲۷ را ببینید). پاسخی که دریافت خواهید کرد، احتمالاً چیزی شبیه زیر است:

```
Trying 192.30.200.66...
Connection to mail.isp.com
Escape character is '^]'
220 mail.isp.com Smail #74 ready at Thu, 25 Mar 2003 13:26 +0200
```

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي


```
                S: 220 xyz.com SMTP service ready
C: HELO abcd.com
                S: 250 xyz.com says hello to abcd.com
C: MAIL FROM: <elinor@abcd.com>
                S: 250 sender ok
C: RCPT TO: <carolyn@xyz.com>
                S: 250 recipient ok
C: DATA
                S: 354 Send mail; end with "." on a line by itself
C: From: elinor@abcd.com
C: To: carolyn@xyz.com
C: MIME-Version: 1.0
C: Message-Id: <0704760941.AA00747@abcd.com>
C: Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm
C: Subject: Earth orbits sun integral number of times
C:
C: This is the preamble. The user agent ignores it. Have a nice day.
C:
C: --qwertyuiopasdfghjklzxcvbnm
C: Content-Type: text/enriched
C:
C: Happy birthday to you
C: Happy birthday to you
C: Happy birthday dear <bold> Carolyn </bold>
C: Happy birthday to you
C:
C: --qwertyuiopasdfghjklzxcvbnm
C: Content-Type: message/external-body;
C:         access-type="anon-ftp";
C:         site="bicycle.abcd.com";
C:         directory="pub";
C:         name="birthday.snd"
C:
C: content-type: audio/basic
C: content-transfer-encoding: base64
C: --qwertyuiopasdfghjklzxcvbnm
C: .
                S: 250 message accepted
C: QUIT
                S: 221 xyz.com closing connection
```

شکل ۷-۱۴. انتقال یک پیام از *elinor@abcd.com* به *carolyn@xyz.com* .

سه خط اول مربوط به برنامهٔ telnet هستند، و می‌گویند چه اتفاقی در حال افتادن است. خط آخر از سرویس‌دهنده SMTP آمده، و آمادگی آنرا برای دریافت ایمیل از طرف شما اعلام می‌کند. برای اینکه ببینید سرویس‌دهنده ایمیل چه فرمانهایی را قبول می‌کند، دستور زیر را وارد کنید

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

HELP

از اینجا به بعد با چیزی شبیه شکل ۷-۱۴ روبرو خواهید بود. پروتکلهای ASCII در اینترنت بسیار رایج هستند، چون تست و دیباگ آنها بسیار ساده است. فرمان دادن به این پروتکلها بسیار آسان است، و پاسخ آنها را نیز براحتی می‌توان درک کرد.

با اینکه پروتکل SMTP بخوبی تعریف شده است، اما برخی مشکلات کوچک نیز می‌تواند بروز کند. یکی از این مشکلات طول پیام است: در برخی از سیستمهای ایمیل قدیمی طول پیام نباید از 64 KB تجاوز کند. مشکل دیگر زمانهای متفاوت انتظار برای پاسخ در دو سمت مقابل است. اگر زمان انتظار برای دریافت پاسخ (timeout) در سمت سرویس‌دهنده و مشتری متفاوت باشد، این احتمال هست که یکی از آنها تسلیم شده (در حالیکه دیگری هنوز منتظر است) و ارتباط را بطور نامنتظره قطع کند. مشکل دیگر بروز توفانهای ایمیل است. اگر ماشین ۱ دارای یک لیست پستی بنام A ، و ماشین ۲ دارای یک لیست پستی بنام B باشند، و هر لیست عضو لیست مقابل باشد، توفانی پایان‌ناپذیر از ایمیلهای تکراری به راه خواهد افتاد (که فقط با دخالت سرپرست سیستم می‌تواند قطع شود).

برای حل این قبیل مشکلات، ویرایش گسترش‌یافتهٔ SMTP (موسوم به ESMTP) در RFC 2821 تعریف شده است. اگر یک مشتری بخواهد بجای SMTP از این پروتکل استفاده کند، باید در شروع کار بجای *HELO* دستور *EHLO* را بکار ببرد. اگر دستور *EHLO* از طرف سرویس‌دهنده پذیرفته نشود، مشتری می‌فهمد که با یک سرویس‌دهنده *SMTP* معمولی سروکار دارد و باید از همان روش سابق استفاده کند. اما اگر *EHLO* پذیرفته شد، مشتری اجازه دارد دستورات این پروتکل را بکار ببرد.

۵-۲-۷ تحویل نهایی

تا اینجا فرض ما بر این بود که تمام کاربران شبکه ماشینهایی با قابلیت ارسال و دریافت ایمیل دارند. همانطور که دیدید، فرستنده باید یک اتصال TCP به ماشین گیرنده برقرار کرده، و ایمیل خود را به آن بفرستد. این روش سالها بخوبی کار می‌کرد، چون تمام ماشینهای آرپانت (و بعدها اینترنت) کامپیوترهایی بودند که همیشه روی خط بودند و می‌توانستند اتصالات TCP را بپذیرند.

این وضعیت با ظهور کاربرانی که برای دسترسی اینترنت مجبور بودند از طریق مودم و با واسطهٔ یک ISP اقدام کنند، تغییر کرد. مشکل این بود که: اگر در لحظه‌ای که Elinor می‌خواهد برای Carolyn ایمیل بفرستد، Carolyn روی خط نباشد، چه خواهد شد؟ در این حالت Elinor قادر به برقراری ارتباط TCP با Carolyn نیست، و نمی‌تواند پروتکل SMTP را اجرا کند.

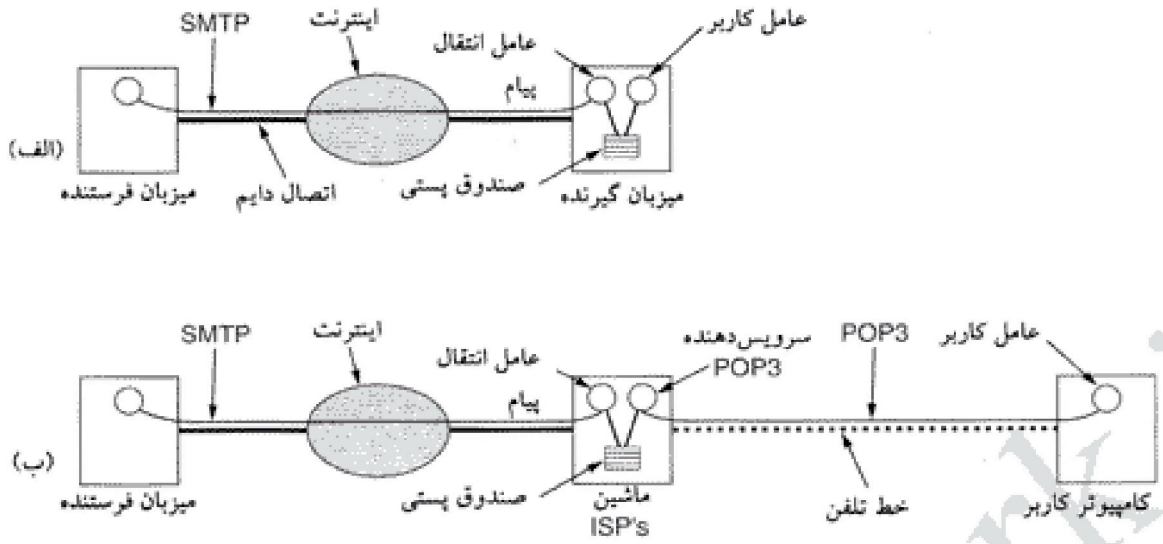
ساده‌ترین راه‌حل آن است که یک عامل انتقال پیام در محل ISP ایجاد شود، و ایمیلها را پس از دریافت از فرستنده در صندوق پستی گیرنده (که در همان ISP قرار دارد) ذخیره کند. از آنجائیکه این عامل انتقال پیام می‌تواند ۲۴ ساعته روی خط باشد، همیشه می‌توان به آن ایمیل فرستاد.

POP3

متأسفانه این راه‌حل یک مشکل کوچک دارد: کاربران چگونه باید ایمیل‌های خود را از عامل انتقال مستقر در ISP بگیرند؟ برای حل این مسئله به پروتکل جدیدی نیاز داریم که PC کاربر را به یک عامل انتقال پیام (از ISP) تبدیل کند. یکی از این پروتکل‌ها، که در RFC 1939 تعریف شده، POP3 (**پروتکل دفترپستی‌ویرایش ۳** – Post Office Protocol Version 3) نام دارد.

به شکل ۷-۱۵ نگاه کنید؛ در تصویر (الف) وضعیتی که باید وجود داشته باشد، را می‌بینید: فرستنده و گیرنده

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي



شکل ۷-۱۵. (الف) وضعیتی که فرستنده و گیرنده دسترسی دائم به اینترنت دارند، و عامل کاربر و عامل انتقال پیام هر دو روی یک ماشین اجرا می‌شوند. (ب) خواندن ایمیل در حالتی که گیرنده از طریق تلفن و با واسطهٔ یک ISP به اینترنت وصل می‌شود.

هر دو دائماً روی خط هستند. در شکل ۷-۱۵ (ب) اوضاع به این خوبی نیست: فرستنده همیشه روی خط است، ولی گیرنده چنین نیست.

پروتکل POP3 کار خود را زمانی شروع می‌کند که کاربر برنامهٔ ایمیل‌خوان را باز می‌کند. برنامهٔ ایمیل‌خوان با ISP تماس گرفته (البته اگر این تماس از قبل وجود نداشته باشد)، و یک اتصال TCP به پورت 110 ماشین عامل انتقال پیام برقرار می‌کند. بعد از برقراری ارتباط، پروتکل POP3 سه مرحله را طی می‌کند:

۱. احراز هویت (authorization) .
۲. تراکنش (transaction) .
۳. به روز در آوردن (update) .

در مرحلهٔ احراز هویت کاربر باید هویت واقعی خود را به عامل انتقال پیام بشناساند. در مرحلهٔ تراکنش، کاربر ایمیل‌های خود را از صندوق پستی خوانده، و آنها را برای حذف شدن علامتگذاری می‌کند. این ایمیل‌ها در مرحلهٔ بعد (به روز در آوردن) حذف می‌شوند.

برای دیدن مراحل کار، فرمان زیر را اجرا کنید:

```
telnet mail.isp.com 110
```

(که در آن *mail.isp.com* نام DNS سرویس‌دهنده ایمیل ISP است.) برنامهٔ telnet یک اتصال TCP با پورت 110 (که سرویس‌دهنده POP3 به آن گوش می‌کند) برقرار می‌سازد. بعد از برقراری ارتباط، سرویس‌دهنده یک پیام ASCII فرستاده، و آمادگی خود را اعلام می‌کند. این پیام معمولاً با +OK شروع می‌شود، و جملهٔ کوتاه دیگری بعد از آن می‌آید. در شکل ۷-۱۶ مکالمهٔ یک مشتری با سرویس‌دهنده POP3 را ملاحظه می‌کنید (در اینجا هم پیامهای مشتری را با C: و پاسخهای سرویس‌دهنده را با S: مشخص کرده‌ایم).

در مرحلهٔ احراز هویت، مشتری ابتدا نام‌کاربر (username) و سپس کلمهٔ‌عبور (password) خود را می‌فرستد. اگر مشتری بتواند این مرحله را با موفقیت پشت سر گذارد، می‌تواند با ارسال دستور *LIST* فهرستی از پیامهای موجود در صندوق پستی خود را دریافت کند (یک پیام در هر خط، که طول آنها نیز مشخص شده است).

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

پایان این لیست با یک نقطه (.) مشخص می‌شود.

```
                S: +OK POP3 server ready
C: USER carolyn
                S: +OK
C: PASS vegetables
                S: +OK login successful
C: LIST
                S: 1 2505
                S: 2 14302
                S: 3 8122
                S: .
C: RETR 1
                S: (sends message 1)
C: DELE 1
C: RETR 2
                S: (sends message 2)
C: DELE 2
C: RETR 3
                S: (sends message 3)
C: DELE 3
C: QUIT
                S: +OK POP3 server disconnecting
```

شکل ۷-۱۶. استفاده از POP3 برای گرفتن پیامهای ایمیل.

پس از آن مشتری می‌تواند برای دریافت پیامها از فرمان *RETR* استفاده کرده، و آنها را با *DELE* برای حذف علامتگذاری کند. وقتی تمام پیامها دریافت (و احتمالاً برای حذف علامتگذاری) شدند، مشتری می‌تواند با فرمان *QUIT* وارد مرحلهٔ بعدی (به روز در آوردن) شود. بعد از آن که سرویس‌دهنده پیامهای علامتگذاری شده را حذف کرد، یک پیام تصدیق به مشتری فرستاده و ارتباط TCP را قطع می‌کند.

با اینکه پروتکل POP3 می‌تواند ایمیل‌ها را بصورت دسته‌جمعی یا تکی گرفته و آنها را روی سرویس‌دهنده‌باقی بگذارد، اغلب برنامه‌های ایمیل همه چیز را از روی سرویس‌دهنده خوانده و سپس صندوق پستی را خالی می‌کنند. در این حالت تنها کپی پیامها در دست خود کاربر است، و اگر روزی کامپیوتر وی صدمه ببیند، همهٔ ایمیل‌ها از بین خواهند رفت.

اجازه دهید یک بار دیگر روش ارسال و دریافت ایمیل را برای کاربران ISP بطور خلاصه مرور کنیم. Elinor در برنامهٔ ایمیل خود (یعنی، عامل کاربر) یک نامه برای Carolyn می‌نویسد، و روی دکمهٔ Send کلیک می‌کند. برنامهٔ ایمیل این پیام را گرفته و به عامل انتقال پیام ISP (که Elinor مشتری آن است) تحویل می‌دهد. عامل انتقال پیام آدرس گیرندهٔ نامه (*carolyn@xyz.com*) را خوانده، و از یک DNS کمک می‌گیرد تا رکورد *MX* ناحیهٔ *xyz.com* را پیدا کند. درپاسخ به این درخواست، DNS نام DNS سرویس‌دهنده ایمیل ناحیهٔ *xyz.com* را برمی‌گرداند. عامل انتقال پیام دوباره به کمک همان DNS آدرس IP این سرویس‌دهنده را درخواست می‌کند. پس از دریافت این آدرس IP ، عامل انتقال پیام یک اتصال TCP به پورت 25 (سرویس‌دهنده SMTP) برقرار می‌کند، و با استفاده از دستورات SMTP (مانند آنچه در شکل ۷-۱۴ دیدید) پیام را به صندوق پستی Carolyn فرستاده، و سپس ارتباط TCP را قطع می‌کند.

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

پس از مدتی، Carolyn (از خواب بیدار شده و) PC خود را روشن کرده، و مثل همیشه قبل از هر کاری به ISP وصل می‌شود تا ایمیل‌هایش را چک کند. برنامهٔ ایمیل در بدو شروع یک اتصال TCP به پورت 110 سرویس‌دهنده ایمیل ISP (سرویس‌دهنده POP3) برقرار می‌کند. نام DNS یا آدرس IP این ماشین در همان بدو عضویت Carolyn در ISP در اختیار وی قرار داده می‌شود، تا آنرا در برنامهٔ ایمیل خود وارد کند (و معمولاً نیازی به پا در میانی DNS نیست). پس از برقراری اتصال به پورت 110 ، برنامهٔ ایمیل پروتکل POP3 را اجرا کرده، و (مانند آنچه در شکل ۷-۱۶ دیدید) پیامهای رسیده را از صندوق پستی Carolyn خوانده و در کامپیوتر وی ذخیره می‌کند (در پایان کار هم، اتصال TCP را قطع می‌کند). در اینجا می‌توان ارتباط تلفنی با ISP را هم قطع کرد (که البته نباید در این کار عجله کنید، چون برای فرستادن پاسخ نامه‌ها باز هم به آن احتیاج پیدا خواهد کرد).

IMAP

برای کاربری که فقط با یک ISP کار می‌کند و همیشه هم از یک PC به آن وصل می‌شود، POP3 بهترین گزینه است (بخصوص که ساده و کارآمد هم هست). ولی در دنیای کامپیوتر اصلی بدیهیست که می‌گوید، «وقتی چیزی دارد خوب کار می‌کند، بلافاصله یکی پیدا می‌شود که بیشتر می‌خواهد.» برای ایمیل هم همین اتفاق افتاد. برای مثال، خیلی از مردم هستند که فقط یک آدرس ایمیل دارند، و می‌خواهند هر کجا که هستند (در خانه، در مدرسه، در محل کار، و یا در سفر) با همان آدرس کار کنند. با اینکه POP3 می‌تواند از عهدهٔ این کار بر آید، اما کاربر بزودی متوجه می‌شود که ایمیل‌هایش روی چندین کامپیوتر پراکنده شده است (کامپیوترهایی که شاید بعضی از آنها حتی متعلق به وی نباشد).

این مشکل POP3 منجر به ارائهٔ راه‌حلی بنام IMAP (پروتکل دسترسی پیام‌اینترنتی – Internet Message Access Protocol) شد، که در RFC 2060 تعریف شده است. بر خلاف POP3 که اساساً فرض می‌کند کاربر تمام پیامهایش را به کامپیوتر خود منتقل کرده و سپس ارتباط با اینترنت را قطع می‌کند، IMAP پیامها را برای همیشه روی کامپیوتر سرویس‌دهنده نگه داشته و آنها را در چندین صندوق پستی حفظ می‌کند. IMAP دارای مکانیزمهای پیشرفته‌ای برای خواندن پیامهاست، که حتی اجازه می‌دهند کاربر فقط بخشهایی از یک پیام بخواند؛ فقط تصور کنید که یک پیام مهم برای کاربر بیچارهٔ ما – که با یک مودم کُند عهد بوق به اینترنت وصل می‌شود – آمده، و این پیام ضمن داشتن متنی مهم دارای یک پیوست بزرگ نیز هست – IMAP به کاربر ما اجازه می‌دهد تا فقط متن پیام را خوانده و از خیر پیوست آن بگذرد. از آنجائیکه در IMAP فرض بر آن است که پیامها به کامپیوتر کاربر منتقل نمی‌شوند، مکانیزمهایی برای نوشتن ایمیل، از بین بردن آنها، و یا مدیریت پیامهای رسیده (مانند دسته‌بندی آنها بر حسب فرستنده) روی سرویس‌دهنده ایمیل در نظر گرفته شده است.

یکی از قابلیتهای جالب IMAP دسته‌بندی و نمایش پیامهای رسیده بر حسب فرستندهٔ ایمیل – یا ویژگیهای دیگر – است (بر خلاف شکل ۷-۸ که تمام ایمیل‌ها پشت سر هم ردیف می‌شوند). IMAP (برخلاف POP3) فقط پروتکلی برای دریافت ایمیل نیست، بلکه می‌تواند ارسال نامه‌ها را هم انجام دهد. روش کار IMAP بسیار شبیه POP3 (شکل ۷-۱۶) است، با این تفاوت که دستورات بسیار متنوعتری دارد. سرویس‌دهنده IMAP به پورت 143 گوش می‌کند. در شکل ۷-۱۷ مقایسه‌ای بین POP3 و IMAP آورده شده است. اما همین جا باید تذکر داد که تمام ISP ها (و همچنین برنامه‌های ایمیل) از هر دو پروتکل پشتیبانی نمی‌کنند (هنگام انتخاب ISP و برنامهٔ ایمیل دقت کنید که از چه پروتکلهایی پشتیبانی می‌کنند).

امکانات سیستم تحویل نامه

اغلب سیستمهای ایمیل، صرفنظر از اینکه از POP3 یا IMAP برای گرفتن نامه‌ها استفاده کنید، امکاناتی برای

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

پردازش پیامها دارند. یکی از این امکانات فیلتر کردن پیامهاست. این فیلترها قواعدی هستند که روی ایمیلهای رسیده عمل می‌کنند. هر قاعده یک شرط دارد و عملی را انجام می‌دهد. برای مثال، یک قاعده می‌تواند بگوید «اگر پیامی از رئیس رسید، آنرا در صندوق شماره ۱ قرار بده.» یا «اگر پیام از گروهی مشخصی از دوستان بود، آنرا در صندوق شماره ۲ قرار بده.» و یا «اگر کلمهٔ خاصی در موضوع پیام بود، آنرا بکلی دور بینداز.»

ویژگی	POP3	IMAP
سطح تعریف پروتکل	RFC 1939	RFC 2060
پورت	110	143
محل ذخیره شدن ایمیل	کاربرPC	سرویس دهنده
محل خوانده شدن ایمیل	خارج خط	روی خط
زمان اتصال	کم	زیاد
استفاده از منابع سرویس دهنده	حداقل	گسترده
صندوق پستی های متعدد	خیر	بلی
مسئول گرفتن پشتیبان	کاربر	ISP
مناسب برای کاربران سیار	خیر	بلی
کنترل بار کردن محتویات	کم	زیاد
بار کردن قسمتی از پیامها	خیر	بلی
مشکل محدودیت دیسک	خیر	گاهی
پیاده سازی ساده است	بلی	خیر
پشتیبانی گسترده	بلی	در حال رشد

شکل ۷-۱۷. مقایسه‌ای بین POP3 و IMAP .

برخی از ISP ها فیلترهایی دارند که بطور خودکار ایمیلهای رسیده را به دو دستهٔ «مهم» و «هرز» (spam یا junk - نامه‌هایی که فقط بدرد سطل آشغال می‌خورند) تقسیم کرده، و آنها را در صندوقهای مخصوص قرار می‌دهند. این فیلترها ابتدا با چک کردن فرستندهٔ نامه‌ها شروع می‌کنند (تا مطمئن شوند از مردم‌آزارهای حرفه‌ای نباشند). اگر چند صد کاربر یک ISP نامه‌هایی با یک موضوع دریافت کنند، فرستندهٔ آن باحتمال زیاد یک هرزنویس است. برای تشخیص این قبیل نامه‌ها تکنیکهای دیگری نیز وجود دارد.

یک دیگر از امکانات سیستمهای تحویل ایمیل هدایت (موقتی) نامه‌های رسیده برای یک کاربر به آدرسی دیگر است. این آدرس حتی می‌تواند شماره‌ای در یک سیستم فراخوان (pager) باشد، که در این حالت کاربر بلافاصله بعد از دریافت هر ایمیل موضوع آنرا در دستگاه فراخوان خود مشاهده خواهد کرد.

امکان دیگر پاسخ خودکار در صورت عدم حضور در محل است (که به آن دیمون تعطیلات - vacation daemon - می‌گویند). با این ویژگی فرد می‌تواند هنگام رفتن به تعطیلات یا مأموریت بطور خودکار به ایمیلهای رسیده پاسخی مانند زیر بدهد:

سلام، من تا ۲۴ ام آگوست در تعطیلات تابستانی هستم. امیدوارم شما هم تعطیلات خوبی داشته باشد.

در این قبیل پیامها حتی می‌توانید مشخص کنید که در صورت اضطرار با کجا باید تماس گرفته شود. در برخی از سیستمهای ایمیل، دیمون تعطیلات می‌تواند تشخیص دهد که قبلاً برای چه کسانی پیام فرستاده، تا از ارسال پیام تکراری برای آنها اجتناب شود. برخی از این دیمون‌ها حتی می‌توانند تشخیص دهند که نامهٔ وارده به یک لیست پستی فرستاده شده، که در این صورت از فرستادن جواب آماده خودداری خواهند کرد.

نویسندهٔ این کتاب شخصاً با یک سیستم پاسخ خودکار فوق‌العاده جالب از جانب شخصی روبرو شده است،

www.homework.ir مشاوره و انجام پروژه هاي دانشجويي دانلود فصلهايي از كتب دانشگاهي

که ادعا می‌کرد روزی ۶۰۰ ایمیل می‌گیرد. (اجازه دهید برای حفظ حریم شخصی افراد، این فرد را با نام مستعار جان معرفی کنیم.)

جان یک روبات ایمیل در کامپیوتر خود نصب کرده که تمام پیامهای رسیده را چک می‌کند. اگر این ایمیل از کسی باشد که قبلاً با جان تماس نداشته (و بعبارت دیگر تازه‌وارد باشد)، پیام خودکاری برای وی ارسال می‌شود که (ضمن عذرخواهی از عدم امکان پاسخ فردی) سندی حاوی تمام اطلاعات موردنیاز (از جمله آدرس، شماره تلفن، شماره فکس، و طریقهٔ تماس با شرکت جان) در آن آمده است. از اطلاعات مفصل دیگری که جان دربارهٔ خود در این پاسخ خودکار آورده حرفی نمی‌زنم، اما بنظر می‌رسد که روش جان یکی از نمونه‌های افراط در استفاده از امکانات باشد.

پُست وب

آخرین مبحثی که در سیستمهای ایمیل می‌توان به آن اشاره کرد، **پُست‌وب** (Webmail) است. همانطور که شاید قبلاً دیده باشید، برخی از سایتهای معروف و بزرگ مانند هات‌میل (Hotmail.com) و یاهو (Yahoo.com) سرویسهای ایمیل مجانی در اختیار بازدیدکنندگان قرار می‌دهند. در این سیستمها یک عامل انتقال پیام معمولی وجود دارد، که به پورت 25 (پورت SMTP) گوش می‌کند. برای وصل شدن به، مثلاً، هات‌میل باید ابتدا رکورد *MX* آنرا بدست آورید؛ در یک سیستم یونیکس می‌توانید از دستور زیر استفاده کنید:

```
host -a -v hotmail.com
```

پس از آن، با فرض اینکه سرویس‌دهنده ایمیل هات‌میل *mx10.hotmail.com* نام داشته باشد، می‌توانید با دستور

```
telnet mx10.hotmail.com 25
```

یک اتصال TCP به پورت 25 آن برقرار کرده، و از فرمانهای SMTP برای ارسال پیامهای خود استفاده کنید. (تا اینجا که چیز غیرعادی وجود ندارد؛ فقط مواظب باشید که سر این کامپیوترها خیلی شلوغ است، و معمولاً باید چند بار سعی کنید تا بتوانید با آنها تماس بگیرید.)

بخش جالب در این سرویسها قسمت تحویل نامه است. معمولاً وقتی وارد صفحهٔ وب سرویس ایمیل می‌شوید، فرمی ظاهر می‌شود که باید نام‌کاربر و کلمهٔ عبور خود را در آن وارد کنید. وقتی دکمهٔ Sign In را کلیک می‌کنید، این اطلاعات به سرویس‌دهنده فرستاده شده و در آنجا با اطلاعات موجود تطبیق داده می‌شود. اگر هویت شما تأئید شود، سرویس‌دهنده صندوق پستی را یافته و محتویات آنرا بصورت یک صفحه‌ای شبیه شکل ۷-۸، ولی با فرمت HTML ، نمایش می‌دهد. اغلب امکانات یک برنامهٔ ایمیل (مانند خواندن نامه، نوشتن نامه، و حذف آنها) در این صفحهٔ وب نیز وجود دارد.

۳.۷ تارنمای جهانی - وب

تارنمای جهانی (World Wide Web) ، یا بطور مختصر **وب**، ساختاریست برای دسترسی به **سندهای پیوندشده** (linked documents) در اینترنت. ظرف ده سال، وب از یک ابزار ارتباطی فیزیکدانها به چیزی تبدیل شده که بسیاری از مردم آنرا همان «اینترنت» می‌دانند. علت اصلی محبوبیت وب در ظاهر گرافیکی آن ریشه دارد، که باعث شده تا میلیونها کاربر تازه‌کار بتوانند بسادگی از آن استفاده کنند. حجم فوق‌العادهٔ اطلاعات موجود در وب را نیز می‌توان یکی دیگر از علل موفقیت آن دانست.

وب (که به WWW نیز معروف است) به سال ۱۹۸۹ در مرکز اروپایی فیزیک هسته‌ای موسوم به CERN متولد شد. در این مرکز دهها تیم تحقیقاتی از سراسر اروپا به کار روی فرضیه‌های فیزیک ذرات مشغول هستند.