فصل دوم

لایه کاربرد

انجمن فناوری اطلاعات
INFORMATION TECHNOLOGY FORUM
www.it98.ir

کاملترین مجموعه نمونه سوالات بهمراه پاسخنامه تستی و تشریحی، جزوه، اسلاید، حل تمرین، گزارش آزمایشگاه و کارآموزی، پروژه و... در سایت:
www.it98.ir
مرجع دانشجویان رشته کامپیوتر دانشگاه پیام نور

کاربردهای شبکه، مفهوم و اصل شبکه کامپیوتری می باشند .اگر ما هیچ کاربرد مفیدی را تصور ننماییم،آنگاه ما هیچ نیازی به پروتکل های شبکه برای پشتیبانی از آنها نداریم.در طول ۴۰ سال گذشته کاربردهای خلاقانه ای برای شبکه ابداع شده است.در ابتدا شامل کاربردهای متنی برای ارسال نامه،دسترسی به کامپیوترهای راه دور،انتقال فایل و چت متنی بود.در میانه دهه ۹۰ کاربردهای) IMجذابی نظیر وب ابداع شدند.وب گردی،جستجو وتجارت الکترونیک.در پایان هزاره دو کاربرد جذاب دیگر پیام رسانهای فوری (بوجود آمدند و شامل کاربردهای موفق صوتی و تصویری (تلفن اینترنتی) اشتراک گذاری P2Pبا لیست دوستان و اشتراک گذاریهای ویدئو،رادیوی اینترنتی و ویدئو کنفرانسها می باشند.

و در نهایت با افزایش دسترسی های خانگی به تکنولوژی باند گسترده و دسترسی روز افزون به وایرلس ،انتظار کاربردهای مهیج دیگری نیز می‌رود.

در این فصل با جنبه های مفهومی و پیاده سازی کاربردهای شبکه آشنا خواهیم شد.مفاهیم کلیدی لایه کاربرد،سرویس‌های مورد نیاز این لایه ،مشتری‌ها و سرورها، پروسس‌ها، رابطه‌های لایه انتقال، وب، ایمیل، DNS ، P2P توزیع فایل ، تلفن اینترنتی .

سپس پیشرفت کاربرد شبکه را از لحاظ TCP و UDP بررسی می‌کنیم.بطور نمونه ، سوکت API را می‌خوانیم و بعد کاربردهای مشتری-سرویس دهنده ساده در جاوا را قدم به قدم بررسی می‌کنیم.بعنوان مثال ، بررسی می‌کنیم که چگونه یک سرویس دهنده وب ساده می‌تواند در جاوا ساخته شود .همچنین چندین تکلیف برنامه نویسی سوکت جذاب و جالب را در آخر این فصل آورده‌ایم.

لایه کاربرد یک نقطه شروع خوب برای مطالعه پروتکل‌ها است . ما با بسیاری از کاربردهای پروتکل‌هایی که می‌خواهیم بخوانیم آشنا شده‌ایم.این آشنایی به ما درک خوبی درباره اینکه پروتکل‌ها بطور کلی چه چیزی هستند می‌دهد و بسیاری از مشکلات مشابه را که دوباره هنگام مطالعه لایه‌های انتقال ، شبکه و پروتکل‌های لایه ارتباط می‌بینیم ،معرفی می‌کند.

۱-۲ : قواعدی کلی از کاربردهای شبکه

با معرفی مثالهای کاربردهای متداول شبکه شروع می‌کنیم :

- پست الکترونیکی
- وب
- پیام رسانی فوری
- دسترسی به یک کامپیوتر از راه دور مانند Telnet و SSH
- اشتراک گذاری فایلP2P
- انتقال قایل مابین دو کاربر بر روی دو کامپیوتر مانندFTP
- بازیهای شبکه ای چند کاربره
- استریم ویدئو کلیپ‌ها
- تلفن اینترنتی
- پخش ویدئو کنفرانس‌های زنده

ما بسیاری از این کاربردها را با جزئیات مختلفشان در این کتاب بحث می‌کنیم.برای مثال در این فصل ایمیل،وب و اشتراک گذاری فایلP2P را پوشش خواهیم داد.در فصل ۷ این کتاب که شبکه های چند رسانه‌ای را بررسی خواهیم کرد ویدئو استریم و تلفن اینترنتی را پوشش می‌دهیم.

www.it98.ir

wWw.it98.ir

www.it98.ir

فرض کنید شما ایده جدیدی برای برنامه شبکه در ذهن دارید.شاید این ایده خدمتی به بشریت یا خوشحال کردن استاد و یا بدست آوردن پول و یا برای سرگرمی باشد . برای پیاده سازی آن بایستی یک برنامه بنویسید که بر روی دو سیستم پایانی مختلف اجرا شده و از طریق شبکه با هم مرتبط شوند.

برای مثال کاربرد وب شامل دو برنامه مجزا می‌باشد .برنامه مرورگر که بر روی ماشین میزبان کاربر نصب و اجرا می‌شود(کامپیوتر رومیزی ، لپ‌تاپ ، پد ، موبایل) و سرور وب که بر روی ماشین سرویس دهنده وب نصب و اجرا می‌شود.بعنوان مثالی دیگر در اشتراک گذاری P2P ، برنامه‌ای وجود دارد که بر روی هر عضو اجتماع اشتراک‌گذاران فایل نصب می‌شود . در این حالت این برنامه ممکن است یکسان و یا مشابه باشند.

بدین‌ترتیب هنگامی که کاربر جدید خود را پیشرفت می‌دهید شما احتیاج به نوشتن نرم‌افزاری دارید که روی چندین ماشین اجرا شود.مهمترین نکته این است که شما هیچ نرم‌افزار و برنامه‌ای که بر روی دستگاههای هسته شبکه اجرا شوند نمی‌نویسید.همانطور که می‌دانیم دستگاههای زیرساخت شبکه در لایه کاربرد عمل نمی‌کنند و بجای آن آنها در لایه‌های پایین‌تر (لایه شبکه و پایین‌تر) عمل می‌نمایند.این طراحی پایه – معروف به نرم‌افزار کاربرد محدود شونده به سیستم‌های مقصد – که در شکل ۱-۲ نشان داده شده است ، پیشرفت سریع و استقرار آرایه‌ای وسیعی از کاربردهای اینترنت را آسان کرده است .

۱-۱-۲ معماریهای کاربرد شبکه :

هنگامیکه یک کاربرد شبکه می‌سازید ،ابتدا لازم است در مورد معماری کاربرد تصمیم‌گیری کنید.لازم است بدانیم که قبل از کدنویسی بایستی یک طرح معماری مناسب برای برنامه خود داشته باشید.معماری کاربرد چیز متفاوتی است از معماری شبکه (بطور مثال معماری ۵ لایه‌ای اینترنت که قبلا بحث شد). معماری شبکه قبلا طرح‌ریزی و ثابت (فیکس شده) می‌باشد از منظر توسعه دهنده کاربرد و یکسری خدمات ویژه برای کاربردها تهیه می‌کند. معماری کاربرد برای طراح برنامه است و نحوه ساختار بندی برنامه روی سیستم‌های پایانی را نشان می‌دهد. در این موضوع طراح اصولا یکی از سه طرح از قبل آماده شده را انتخاب می‌نماید . معماری مشتری-سرویس دهنده ، معماری نظیر به نظیر ، یک پیوند از معماری نظیر به نظر و مشتری – سرویس دهنده .

در معماری مشتری-سرویس دهنده حتما میزبانی روشن بنام سرویس دهنده وجود دارد که به درخواست‌های رسیده از طرف مشتریان سرویس می‌دهد.میزبانهای مشتریان می‌توانند گاهی روشن و یا همیشه روشن باشند.یک مثال کلاسیک از این موضوع برنامه وب می‌باشد برای هر سرویس دهنده وب همیشه روشن است که از جستجوگرهای در حال اجرای میزبانان مشتری درخواست خدمات می‌کنند.در این معماری مشتریها مستقیما با هم ارتباط ندارند . برای مثال در کاربرد وب دو جستجوگر بطور مستقیم با هم در ارتباط نیستند .دیگر خصوصیت معماری مشتری-سرویس دهنده اینست که سرویس دهنده یک آدرس ثابت و مشخص که IP نامیده می‌شود دارد.بدلیل اینکه سرویس دهنده یک آدرس ثابت و مشخص دارد و همچنین سرویس دهنده همیشه روشن است ، یک مشتری همیشه می‌تواند با سرویس دهنده بوسیله ارسال یک بسته به آدرس سرویس دهنده، ارتباط برقرار کند.بعضی از کاربردهای مطرح‌تر با معماری مشتری-سرویس دهنده شامل وب ،انتقال فایل، اتصال از راه دور و ایمیل می‌شود.این معماری مشتری-سرویس دهنده در شکل ۲-۲(a) نشان داده شده است .

اغلب در برنامه‌های مشتری-سرویس دهنده یک میزبان سرویس دهنده برای پاسخگویی به تقاضاهای تمام مشتریان کافی نیست.برای مثال وب سایت‌های شبکه‌ای اجتماعی که به سرعت رشد می‌نمایند از مجموعه‌ای از میزبانها(Cluster Of hosts) بنام مزرعه سرویس دهنده(Server farm) استفاده می‌شود که معمولا برای ساخت یک سرویس دهنده مجازی قدرتمند در معماری مشتری-سروی دهنده استفاده می‌شوند.

www.it98.ir

www.it98.ir

موتورهای جستجو معروف(گوگل)، تجارتهای الکترونیکی (آمازون و e-bay) ایمیلهای وبی (yahoo mail) شبکه‌های اجتماعی(My space,facebook) و اشتراک‌گذاریهای ویدئو (youtube) بصورت infrastructure intensive هستند و پرهزینه می‌باشند.

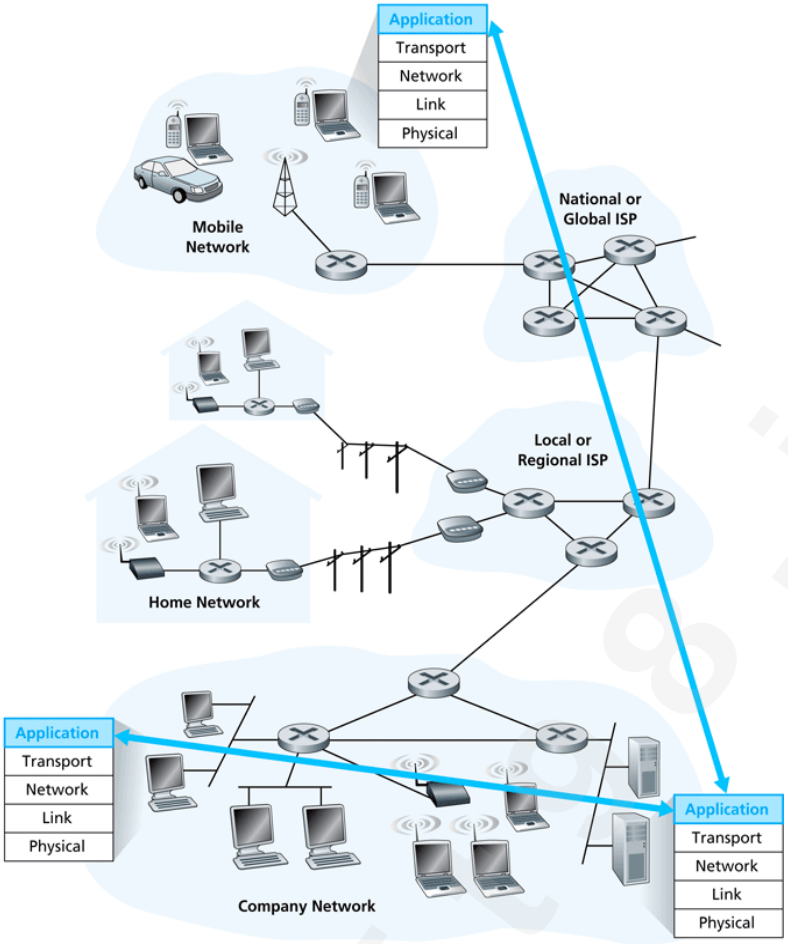


Figure 2.1 ♦ Communication for a network application takes place between end systems at the application layer.

در معماری P2P اتکای حداقل و یا هیچ به سرورهای زیر ساخت وجود دارد. بجای آن برنامه‌ها مستقیما با نظیرهایی که بطور متناوب متصل هستند ارتباط برقرار می‌کنند.این نظیرها توسط کاربران کنترل می‌شوند و بر روی میزبانهای رومیزی ،لپ‌تاپ می‌باشند. از آنجاییکه ارتباط آنها از طریق هیچ سروری انجام نمی‌شود به‌نام Peer-to-Peer معروف هستند. برنامه‌های توزیع فایل (نظیر Bit torrent)، اشتراک‌گذاری/جستجوی فایل (emule) ،تلفن اینترنتی (Skype) و تلویزیون روی IP (PPLive) برنامه‌های معروفی هستند از این معماری.برخی برنامه‌های کاربردی ممکن است از معماری ترکیبی استفاده نمایند. برای مثال بسیاری از برنامه‌های IM از سرویس دهنده‌ها برای دنبال کردن آدرس IP کاربران کمک می‌گیرند اما پیام‌های کاربران مستقیما بین میزبانها ارسال می‌شود.(بدون عبور از طریق سرویس دهنده).یک مثال خوب از یک کاربردی که دارای معماری P2P کامل است Gnutella(Guntella 2004) می‌باشد .در Gnutella هر میزبان می‌تواند درخواست فایل کند و فایل را ارسال کند.سوال برای پیدا کردن مکان فایل ، به پرسش‌ها پاسخ دهد و پرسش‌ها را نیز ارسال کند . ما Gnutella را بطور مفصل در بخش ۶-۲ توضیح می‌دهیم. معماری P2P در شکل (b)۲-۲ مشخص شده است .

از ویژگیهای مهم معماری P2P ،ویژگی خود مقیاس پذیری است (self scalability) بطور مثال در اشتراک‌گذاریهای فایل P2P گرچه هر عضو جدید با کاری را به مجموعه اضافه می‌کند اما در عوض خود او هم ظرفیت سرویس برای توزیع فایل را فراهم

wWw.it98.ir

www.it98.ir

می‌آورد.اما بخاطر ماهیت کاملا باز و توزیع شده این معماری چالشی برای امنیت می‌باشند.بدین ترتیب ،هنگامیکه هر عضو حجم بار (work laod) را بوسیله فایلهای درخواست شده تولید و منتشر خواهد کرد ،هر عضو همچنین ظرفیت خدمات را برای سیستم بوسیله پاسخ داده به درخواست‌های دیگر نظیرها اضافه می‌کند.بدین ترتیب اساسا،اشتراک‌گذاری فایل P2P ذاتا مقیاس پذیر (scalable) است.هر عضو اضافه فقط درخواست را اضافه می‌کند اما همچنین خدمات را نیز افزایش می‌دهد.

از طرف دیگر،بدلیل توزیع بیش از حد و ماهیت نامتمرکز کاربردهای P2P ،می‌توانند برای اداره شدن مشکل باشند. برای مثال یک عضو که تنها کپی یک فایل مهم را دارد و آن عضو می‌تواند در هر زمانی از ارتباط خارج شود.این یک سوال آزاد را باقی می‌گذارد که آیا ممکن است که راه حلهای P2P صنعتی برای کاربردهای سرمایه‌گذاری ساخته شود یا خیر ؟

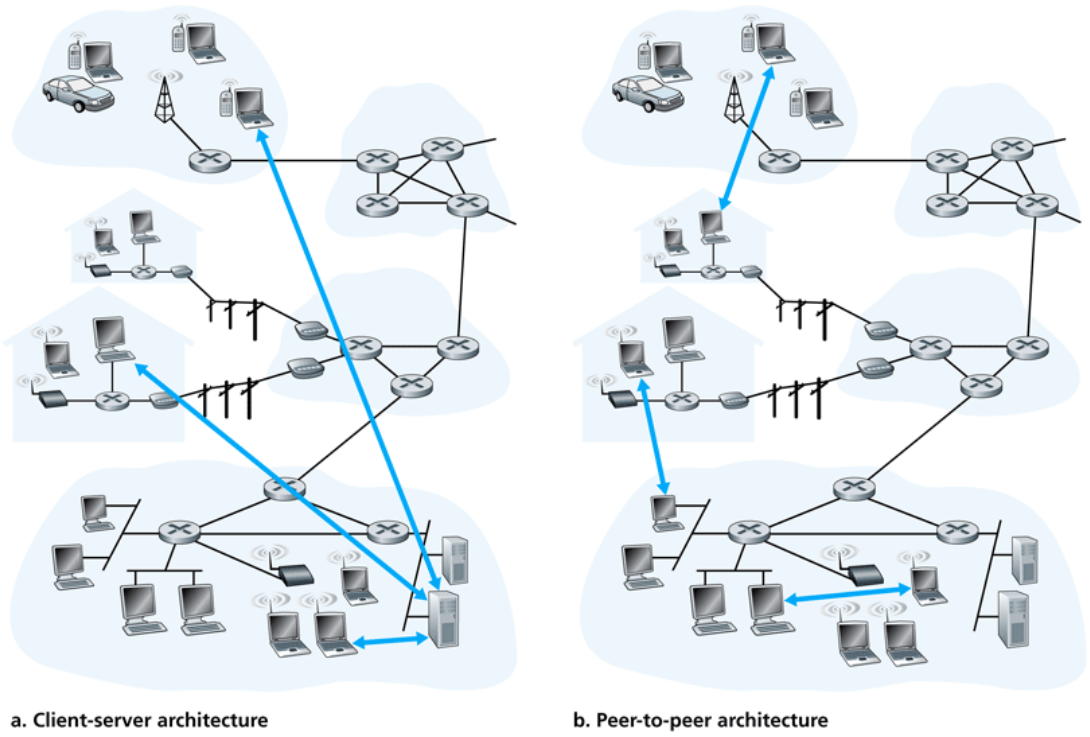


Figure 2.2 ♦ (a) Client-server architecture; (b) P2P architecture.

مشتری-سرویس دهنده و نظیر به نظیر دو معماری رایج در کاربردهای شبکه هستند . هر چند، بسیاری از کاربردها بصورت پیوندی از معماری مشتری-سرویس دهنده و نظیر به نظیر سازماندهی شده‌اند . بعنوان مثال Napster که اکنون از بین رفته است که اولین برنامه اشتراک‌گذاری فایل MP3 پرکاربرد بود. Napster ، P2P است بدین صورت که فایلهای MP3 بطور مستقیم در میان عضوها تغییر کرده‌اند، بدون عبور اختصاصی از میان سرویس دهنده‌های همیشه روشن . اما Napster همچنین C/S نیز است بدین صورت که یک جفت از یک سرویس دهنده مرکزی برای مشخص کردن اینکه کدام جفت‌ها در حال حاضر یک طراحی فایل MP3 دارد می‌پرسد.(معماری Napster بصورت کاملتر در بخش ۶-۲ توضیح داده می‌شود.)برنامه دیگر که از یک معماری پیوندی استفاده می‌کند پیام فوری(Instant Message) است . در پیام دادن فوری گفتگو بین دو کاربر بطور نمونه P2P است ، یعنی اینکه متن بین دو کاربر از میان یک سرویس دهنده رابط همیشه روشن عبور نمی‌کند . هر چند ، هنگامیکه آلیس برنامه پیام دادن فوری خود را اجرا می‌کند ، خود را در یک سرویس دهنده مرکزی ثبت نام می‌کند و هنگامی که باب می‌خواهد با فردی در لیست

www.it98.ir

www.it98.ir

دوستان خود صحبت کند مشتری پیام دادن فوری او به سرویس دهنده مرکزی برای فهمیدن کدامیک از دوستان در حال حاضر روی خط و در دسترس هستند ارتباط برقرار می‌کند.

۲-۱-۲ ارتباط بین پروسس ها

قبل از نوشتن برنامه لازم است بدانیم که برنامه‌ها چگونه در سیستم‌های نهایی اجرا شده و با هم ارتباط برقرار می‌نمایند . در اصطلاح سیستم عامل به آن برنامه نمی‌گویند بلکه پروسس نامیده می‌شود . در حالت ساده پروسس یک برنامه‌ای است که بر روی ماشین نهایی اجرا می‌شود . زمانیکه پروسس‌ها در یک ماشین نهایی در حال اجرا می‌باشند از طریق مکانیزم ارتباط بین پروسس Interprocess Communication با همدیگر ارتباط برقرار می‌نمایند .

پروسس‌های اجرا شونده بر روی دو سیستم نهایی از طریق پیام از طریق شبکه با هم ارتباط برقرار می‌نمایند . پروسس ارسال کننده پیامی را تولید کرده و به شبکه می‌دهد و پروسس دریافت کننده آن پیام را از شبکه گرفته و در صورت لزوم جواب را با پیامی پاسخ خواهد داد.

پروسس‌های مشتری و سرویس دهنده

در یک کاربرد شبکه یک جفت پروسس وجود دارد که پیام را به همدیگر ارسال می‌دارند ، برای مثال پروسس مرورگر مشتری با پروسس سرویس دهنده وب تبادل پیام می‌نماید .

در یک ارتباط P2P ، پروسسی که اقدام به دریافت فایل می‌نماید ، پروسس مشتری نام می‌گیرد و (شروع کننده ارتباط) و پروسسی که فایل را ارسال می‌کند ، پروسس سرویس دهنده نامیده می‌شود .(پروسسی که منتظر ارتباط می‌ماند .)

اینترفیس بین پروسس و شبکه کامپیوتری

در نوشته بالا ، بسیاری از برنامه‌ها از جفت‌های پردازش‌های ارتباطی هستند . با دو پردازش در هر جفت پیام ارسالی به یکدیگر . هر پیام از یک پردازش به دیگری باید از میان شبکه اصولی ارسال شود . پروسس‌ها (مشتری و سرویس دهنده) اقدام به ارسال پیام از طریق اینترفیس نرم‌افزاری بنام سوکت می‌نمایند. اگر سوکت را خانه فرض نماییم ، سوکت مشابه درب خانه می‌باشد . اگر پروسسی بخواهد پیامی را به پروسس دیگری بفرستد آنرا به درب (سوکت) هل می‌دهد و به درون شبکه این پروسس ارسال کننده فرض می‌کند که یک زیرساخت انتقالی موجود است که پیام را از درب تحویل گرفته و به سمت مقصد انتقال می‌دهد ، به محض رسیدن پیام به میزبان مقصد پیام از طریق درب پروسس گیرنده (سوکت) به پروسس مقصد تحویل داده می‌شود .

شکل ۳-۲ ارتباط سوکتی بین دو پروسس را از طریق اینترنت نمایش دهد .در این شکل فرض شده است که پروتکل انتقال زیرین، پروتکل اینترنتی TCP می‌باشد. در شکل مشاهده می‌شود که سوکت اینترفیس بین لایه کاربرد و لایه انتقال می‌باشد و تحت عنوان API (App.Programming Interface) معروف است . بنابراین سوکت یک API است و توسعه دهنده کاربرد کنترل کاملی روی سمت لایه کاربرد دارد. اما کنترل جزیی روی سمت لایه انتقال آن دارد . کنترل‌های موجود در سمت لایه انتقال عبارتند از (۱) نوع پروتکل انتقال و (۲) توانایی تنظیم برخی پارامترهای لایه انتقال شامل اندازه بافر و حداکثر اندازه سگمنت می‌باشد.بعد از انتخاب پروتکل انتقالی ، برنامه کاربردی می‌تواند از سرویس‌های ارائه شده توسط آن پروتکل استفاده نماید ، سوکت‌ها را بصورت دقیقتر در بخش‌های ۷-۲ و ۸-۲ موشکافی می‌کنیم .

آدرس‌دهی پروسس

www.it98.ir

www.it98.ir

برای یک پروسس در یک میزبان برای ارسال یک پیام به یک پروسس در میزبان دیگر ، پروسس ارسالی باید پروسس گیرنده را شناسایی کند .برای شناسایی کردن پروسس گیرنده ، دو بخش از اطلاعات باید شناسایی شود : (۱) اسم یا آدرس میزبان و (۲) یک شناسایی کننده که پروسس گیرنده را در میزبان مقصد تعیین کند .

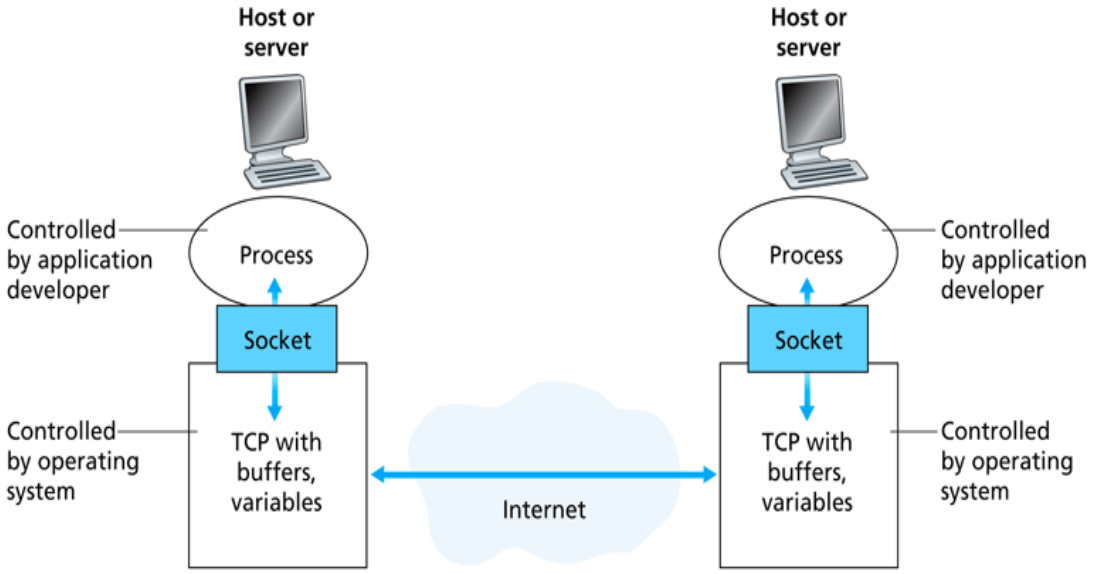


Figure 2.3 ♦ Application processes, sockets, and underlying transport protocol

اجازه بدهید ابتدا آدرس‌های میزبان را بررسی کنیم . در کاربردهای اینترنتی ، میزبان مقصد بوسیله آدرسIP خود شناسایی خواهد شد . آدرس‌های IP را بصورت کاملتر در فصل ۴ بحث خواهیم کرد . اکنون این کافی است که بدانید آدرسIP یک کمیت ۳۲ بیتی است که بصورت منحصر بفرد و بی‌مانند میزبان را شناسایی می‌کند (بطور دقیقتر ، بصورت منحصر بفرد شبکه واسطه که آن میزبان را به اینترنت متصل کرده را شناسایی می کند. از زمانی که آدرسهای IP به هر میزبان اینترنت همگانی متصل شده‌اند باید بصورت سراسری بی‌مانند باشند ، وضیفه آدرس‌های IP باید بدقت اداره شود ، همانطور که در فصل ۴ بحث کرده‌ایم .)

بعلاوه دانستن آدرس میزبان برای اینکه یک پیام کجا فرستاده شود ، میزبان فرستنده باید همچنین پروسس گیرنده اجرایی در میزبان را شناسایی کند . این اطلاعات به این دلیل لازم است چون بطور عمومی یک میزبان می‌تواند بسیاری از کاربردهای شبکه را راه‌اندازی کند . یک شماره پورت (Port Number) به این منظور بکار برده می‌شود . برنامه‌های پرکاربرد با شماره پورت‌های ویژه‌ای مشخص می‌شوند . برای مثال یک سرویس دهنده وب بوسیله پورت ۸۰ شناسایی می‌شود . یک پروسس سرویس دهنده پستی (با استفاده از پروتکSMPT) بوسیله پورت ۲۵ شناسایی می‌شود . یک لیست از شماره پورتهای معروف برای همه پروتکلهای استاندارد اینترنتی در http://www.lona.org پیدا می‌شود . هنگامیکه یک توسعه دهنده یک کاربرد شبکه جدید را می‌سازد کاربرد باید با یک شماره پورت جدید مشخص شود . شماره پورت‌ها را در فصل۳ مفصل می‌گوئیم .

۳-۲-۱ پروتکل های لایه کاربرد

تاکنون ما دیده ایم که چگونه پروسس های شبکه از طریق سوکت باهم ارتباط برقرار می نمایند .اما پیام ها چگونه شکل می گیرند ؟معنی فیلدهای درون پیام چیست ؟چه زمانی پروسس پیام را ارسال میدارد ؟این سوالات ما را به قلمرو پروتکل های لایه

www.it98.ir

کاربرد می برد . یک پروتکل لایه کاربرد مشخص می کند چگونه یک پروسس کاربرد ، بر روی سیستم های پایان مختلف اجرا می شود ،پیام ها را به هم ارسال می کنند . بطور نمونه یک پروتکل لایه کاربرد مشخص می کند :

- نوع پیام های منتقل شده ، برای مثال پیام های درخواست و پیام های پاسخ
- ترکیب و ساختار انواع پیام ها،نظیر فیلدها در پیام

معنی هر فیلد (چه اطلاعاتی را منتقل می نماید)

قوانینی برای تعیین اینکه چه زمانی و چگونه پروسس پیام را ارسال و پاسخ میدهد .

مشخصات برخی از پروتکل های لایه کاربرد در RFCها آورده می شود و بنابراین در دسترس عموم می باشد .برای مثال پروتکل لایه کاربرد وب،HTTP، در RFC2616 آمده است .بنابراین اگر یک مرورگر قوانین HTTP RFC را رعایت کند قادر به دریافت صفحات وب از هر سرویس دهنده وب (که آنهم قوانین HTTP RFC را بایستی رعایت کند)می باشد . برخی دیگر از پروتکل های لایه کاربرد اختصاصی بوده و در دسترس عموم نمی باشد.

برای مثال بسیاری از سیستم های اشتراک گذاری فایل P2P از پروتکل های اختصاصی لایه کاربرد استفاده می نمایند .

مهم است بین کاربرد شبکه(Netwok Application) و پروتکل های لایه کاربرد تمایز قائل شد . پروتکل لایه کاربرد تنها بخشی از کاربرد شبکه می باشد . (بخش مهم از آن)بعنوان مثال کاربرد وب (Web application) کاربردی مشتری – سرویس دهنده است. شامل بخش های زیر :

استاندارد فرمت اسناد وبی ، مرورگرهای وب ، سرویس دهنده های وب ، پروتکل HTTP .پروتکل لایه کاربرد وب ، HTTP ، قالب و ترتیب پیام هایی که بین جستجوگر و سرویس دهنده وب در انتقال هستند را مشخص می کند . بعنوان یک مثال دیگر ، کاربرد ایمیل اینترنتی را ملاحظه کنید .نامه الکترونیکی اینترنتی همچنین ترکیب های بسیاری دارد ، شامل سرویس دهنده های پستی که جایگاه صندوق پستی های کاربر می شود؛خوانندگان نامه که به کاربران اجازه می دهند که پیام ها را بخوانند و یا ایجاد کنند ؛ یک استاندارد برای شناسایی ساختار یک پیام ایمیل و پروتکل لایه کاربرد که مشخص می کند چگونه پیام بین سرویس دهنده ها حرکت می کنند ، چگونه پیام بین سرویس دهنده ها و نامه خوانها جرکت می کنند و چگونه محتوای قسمت های ویژه ای از پیام ایمیل (برای مثال ، یک عنوان پیام ایمیل) باید تفسیر و ترجمه شود . اصل و مبدا پروتکل لایه کاربرد برای نامه الکترونیکی SMTP است (پروتکل انتقال نامه ساده :Simple Mail Transfer Protocol –[RFC 2821] بدین ترتیب اصل و مبدا پروتکل لایه کاربرد ایمیل،SMTP تنها بخشی از (البته یک بخش عمده) کاربرد ایمیل است .

۴-۱-۲ سرویسهای انتقالی موجود برای کاربردها

در بسیاری از شبکه‌ها از جمله اینترنت پروتکل لایه انتقال شامل سرویس‌های مختلفی می‌باشد . هنگام فراخوانی آنها بایستی نوع پروتکل در دسترسی را انتخاب نمایید. کاربرد سمت ارسالی پیام‌ها را از میان در عبور می‌دهد . در سمت دیگر ، پروتکل لایه کاربرد مسئولیت بدست آوردن و کسب پیام‌ها از طرف در پروسس گیرنده دارد .

بسیاری از شبکه‌ها ، مانند اینترنت بیشتر از یک پروتکل لایه کاربرد تهیه و آماده می‌کنند .سوال اینست که چگونه شما این انتخاب را انجام می‌دهید ؟ جواب این سوال همانند اینست که شما برای رفتن از جایی به جای دیگر می‌توانید از ماشین ، قطار ، هواپیما و ... استفاده نمایید . شما باید یکی را انتخاب کنید و هر حالت انتقال خدمات متفاوتی را به شما ارائه می‌دهد .در واقع آن سرویس را انتخاب می‌نماییم که بهترین مطابقت را با خواسته‌های ما داشته باشد . (مثلا قطار یک سوار شدن و پیاده شدن راحت را ارائه می‌دهد در حالیکه هواپیما مسیر را کوتاهتر می‌کند .)

www.it98.ir

wWw.it98.ir

www.it98.ir

چه سرویس‌هایی را یک پروتکل لایه انتقال می‌تواند به کاربر ارائه دهد :سرویس انتقال مطمئن داده ،سرویس انتقال غیرمطمئن داده ،توان عملیاتی (درخواست یک سرعت خاصی با گارانتی آن)

انتقال داده مطمئن

بعضی کاربردها مانند پست الکترونیکی پیام دهی فوری،انتقال فایل ، دسترسی میزبان راه دور ، انتقال سند وب و کاربردهای مالی انتقال داده کاملا قابل اعتماد نیاز دارد ، یعنی اینکه بدون از دست دادن داده‌ها بطور نمونه یک خسارت از داده فایل یا داده‌ای در یک انتقال مالی ، می‌تواند یک پیامد بسیار بدی داشته باشد.(برای بانک و همچنین مشتری). دیگر کاربردهای نوسان تلفات، برجسته‌ترین کاربردهای چند رسانه‌ای مانند یک صوتی/تصویری زمان حقیقی یا صوتی/تصویری نگهداری شده ، می‌تواند مقداری از تلفات داده را تحمل و پشتیبانی کند . در این کاربردهای چندرسانه‌ای ، داده تلف شده ممکن است یک خرابی تصادفی کوچک در اجرای برنامه صوتی/تصویری نتیجه کند – نه یک اختلال بحرانی . تاثیرات این خسارت در کیفیت کاربرد و مقدار واقعی خسارت کوچک تحمل‌پذیر قویا به کاربرد و طرح و نقشه کدگذاری استفاده شده بستگی دارد .

پهنای باند

بعضی از کاربردها باید قابلیت برای انتقال داده و یک سرعت ویژه در نتیجه تاثیرگذار باشد . بطور مثال در کاربرد تلفن اینترنتی ممکن است برنامه درخواست سرعت 32 kb/s را برای ارسال صدا داشته باشد . در نتیجه باید قابل ارسال داده به درون شبکه باشد و داده را به کاربرد گیرنده در این سرعت تحویل داده باشد.اگر این مقدار پهنای باند در دسترس نباشد ، کاربرد احتیاج به رمزدهی در یک سرعت متفاوت دارد . برنامه‌هایی که نیاز به توان عملیاتی خاصی برای انجام کار هستند تحت عنوان کاربدهای حساس به پهنای باند انمیده می‌شود . همانند کاربردهای مالتی مدیا، در حال حاضر بسیاری از کاربردهای چندرسانه‌ای حساس به پهنای باند هستند ، اما کاربردهای چندرسانه‌ای آینده تکنیکهای رمزدهی تطابقی برای رمزگذاری در یک سرعت که مطابق با پهنای باند در دسترس امروزی است. برنامه‌های دیگر که حساس به پهنای باند نیستند ولی می‌توانند تحت سرعتهای مختلف کار نمایند " برنامه‌های الاستیک" نامیده می‌شوند .همانند ایمیل ، انتقال فایل ، وب –البته اگر سرعت بیشتر باشد بهتر است!

زمانبندی

آخرین سرویس مورد نیاز زمانبندی است.یک پروتکل لایه انتقال می‌تواند تضمین (گارانتی) زمانی(تاخیری) بدهد.گارانتی زمانی می‌تواند به شکل‌های مختلفی باشد.بطور مثال زمانیکه فرستنده بیتی را به سمت سوکت ارسال می‌دارد ، انتظار دارد در کمتر از 100 ms به مقصد برسد .چنین تضمینی برای کاربردهای تعاملی بلادرنگ مورد نظر است . نظیر تلفن اینترنتی ، محیط‌های مجازی ، کنفرانس راه دور و بازیهای چند نفره تاخیرهای طولانی باعث عدم طبیعی بودن محاوره خواهد شد.برای کاربردهای زمان غیرحقیقی ، تاخیر کمتر همیشه نسبت به تاخیر بیشتر پرارزش‌تر است ، ولی هیچ پیوند فشرده‌ای روی تاخیرات پایان به پایان (end-to-end) قرار ندارد .

شکل ۲۰۴ خلاصه‌ای از قابلیت اطمینان ، پهنای باند و زمانبندی مورد نیاز بعضی از کاربردهای اینترنتی فوری و پرکاربرد را نشان می‌دهد . شکل ۲۰۴ فقط تعداد کمی از کلیدهای مورد نیاز برنامه‌های اینترنتی که بسیار پرکاربرد هستند را توضیح می‌دهد . هدف ما این نیست که یک دسته‌بندی کلی فراهم کنیم اما ساده کردن شناختن بعضی از مهمترین محورهایی که کاربردهای شبکه نیاز دارند می‌توان دسته بندی کرد .

www.it98.ir

www.it98.ir

امنیت

در نهایت پروتکل انتقال می‌تواند برای کاربرد سرویس‌های امنیتی فراهم آورد برای مثال در میزبان فرستنده پروتکل انتقال تمام داده‌های ارسالی را به رمز درآورده و به فرستنده تحویل دهد و در سمت گیرنده آنها را از رمز درآورد . چنین سرویسی می‌تواند "محرمانه بودن" را فراهم آورد . (Confidentiality) سرویس‌های دیگری نظیر "جامعیت" (integrity) و یا "اعتبارسنجی نقطه پایانی" را فراهم آورد.

۲-۱-۵ سرویسهای فراهم شده توسط پروتکل انتقال اینترنت

در اینترنت دو نوع پروتکل انتقالی ارائه می‌شود : TCP و UDP . هر کدام مجموعه سرویس‌های مختلفی را ارائه می‌دهند . هنگامیکه شما (بعنوان نویسنده یک نرم‌افزار) یک کاربرد شبکه جدید می‌سازید برای اینترنت یکی از اولین تصمیماتتان اینست که برای ساختن باید از UDP استفاده کنید یا TCP .

سرویسهای TCP

مدل سرویس TCP شامل سرویس "اتصال‌گرا" و " انتقال مطمئن داده" می‌باشد .

Application	Data Loss	Bandwidth	Time-Sensitive
File transfer	No loss	Elastic	No
E-mail	No loss	Elastic	No
Web documents	No loss	Elastic (few kbps)	No
Internet telephony/ Video conferencing	Loss-tolerant	Audio: few kbps–1Mbps Video: 10 kbps–5 Mbps	Yes: 100s of msec
Stored audio/video	Loss-tolerant	Same as above	Yes: few seconds
Interactive games	Loss-tolerant	Few kbps–10 kbps	Yes: 100s of msec
Instant messaging	No loss	Elastic	Yes and no

Figure 2.4 ♦ Requirements of selected network applications

سرویس اتصال‌گرا : TCP مشتری و سرویس دهنده مبادله اطلاعات کنترلی لایه انتقال را با هر کدام قبل از اینکه پیام‌های سطح کاربرد شروع به جریان کنند دارد . این روش دست تکانی مشتری و سرویس دهنده را باخبر می‌کند و به آنها اجازه می‌دهد که برای یک هجوم از بسته‌ها آماده شوند . بعد از فاز دست تکانی ، یک ارتباط TCP بین سوکت‌های دو پروسس وجود دارد . ارتباط یک ارتباط تمام دو طرفه در آن دو پروسس است که می‌تواند پیام‌ها را به یکدیگر از طریق ارتباط در یک زمان ارسال کند . هتنگامیکه برنامه پیام‌های ارسالی را به پایان رساند ، باید ارتباط را قطع کند . سرویس بعنوان یک سرویس اتصال‌واگرا نسبت به یک سرویس اتصال اشاره می‌کند به همین دلیل که هر دو پروسس در یک حالت بسیار ضعیف ارتباط دارند . در فصل ۳ در مورد سرویس اتصال‌واگرا توضیح خواهیم داد و مطالعه می‌کنیم که چگونه پیاده‌سازی می‌شود .

wWw.it98.ir

www.it98.ir

سرویس انتقال مطمئن : پروسس‌های ارتباطی می‌توانند بهTCP برای تحویل دادن تمام داده‌های ارسال شده درون خطا اعتماد کند و با دستور درست هنگامیکه یک طرف کاربرد یک جریانی از بایت‌ها را به درون سوکت هدایت می‌کند ، می‌تواند روی TCP برای تحویل همان جریان بایت‌ها به سوکت گیرنده حساب و اعتماد کند ، بدون گم شدن یا تکرار کردن بایت‌ها .

همانگونه که در فصل ۳ خواهیم دید ، TCP همچنین شامل مکانیزم کنترل ازدحام نیز می‌باشد . که زمانیکه شبکه در حالت ازدحام و شلوغی است پروسس‌ها (مشتری یا سرویس دهنده) را برای ارسال داده محدود می‌نماید . همچنین تلاش می‌کند که هر اتصالTCP را به قسمت منصفانه پهنای باند شبکه خودش محدود کند .

محدود کردن سرعت انتقال می‌تواند تاثیرات بسیار مضری بر روی برنامه‌های ویدئویی و صوتی زمان حقیقی که یک حداقل محدودیت پهنای باند مورد نیاز دارند ، داشته باشد . بعلاوه ، برنامه‌های زمان حقیقی کمتر نوسان و اشتباه دارند و احتیاج به یک سرویس انتقال کاملا قابل اعتماد ندارند . برای همین دلیل ، توسعه دهندگان برنامه‌های زمان حقیقی معمولا برنامه هایشان را از طریق UDP نسبت به TCP اجرا می‌کنند .

خلاصه ای از خدمات تهیه شده بوسیلهTCP را گفتیم ، اکنون می‌خواهیم مقداری در مورد خدماتی که TCP تهیه نمی‌کند صحبت کنیم . اولا TCP یک حداقل سرعت انتقال را تضمین نمی‌کند . بطور نمونه ، یک پروسس ارسال برای فرستادن در هر سرعت دلخواهی اجازه ندارد . بجای آن سرعت ارسالی بوسیله کنترل ازدحام TCP تنظیم خواهد شد که ممکن است فرستنده را برای ارسال در یک سرعت میانگین پایین مجبور کند . دوما TCP هیچ ضمانت تاخیر را نمی‌دهد.بطور نمونه هنگامیکه یک پروسس ارسال داده را به درون یک سوکت TCP هدایت می‌کند ، داده نهایتا به پروسس گیرنده خواهد رسید ، اما TCP بطور قطع هیچ محدودیتی در مورد اینکه چقدر طول می‌کشد که داده به آنجا برسد را تضمین نمی‌کند . همانطور که همه ما تجربه کرده‌ایم با "world wide wait" یک شخص می‌تواند گاهی اوقات ده‌ها ثانیه یا حتی دقیقه برای تحول یک پیام توسطTCP (مثلا شامل یک فایلHTML باشد) از سرویس دهنده وب به مشتری وب منتظر بماند . بطور خلاصه ، TCP تحویل دادن همه داده‌ها را تضمین می‌کند ، اما تضمین برای سرعت تحویل و یا تاخیرات موجود نمی‌کند .

سرویس‌های UDP :

UDP یک سرویس بدون حاشیه‌ای است ، پروتکلی سبک و با حداقل سرویس می‌باشد . بدون اتصال بوده بنابراین هیچ دست تکانی برای قبل از اینکه دو پروسس شروع به ارتباط کنند ، وجود ندارد و انتقال مطمئن داده را تضمین نمی‌نماید و ممکن است داده‌های ارسالی حتی به ترتیب هم به مقصد نرسند و مکانیزم کنترل ازدحام هم ندارد و بنابراین طرف ارسال کننده می‌تواند داده‌ها را با هر سرعتی که مایل است به سمت شبکه ارسال نماید .

برخی برنامه‌های بلادرنگ که می‌توانند مقداری خطا را تحمل نمایند اما تاخیر زمانی برای آنها مشکل‌ساز است از این سرویس استفاده نمایند . در نتیجه به کنترل ازدحام TCP و نامه‌های اضافه صدمه می‌زند .مانند TCP ، UDP نیز ضمانتی در قبال تاخیر نمی‌دهد .

شکل ۵-۲ نشان می‌دهد که پروتکل‌های انتقال بوسیله بعضی از برنامه‌های پرکاربرد استفاده می‌شوند .می‌بینیم که ایمیل ، دسترسی به خروجی راه دور ، وب انتقال فایل ، همه از TCP استفاده می‌کنند . این برنامه اول از همه TCP را انتخاب خواهند کرد زیراTCP یک سرویس انتقال داده قابل اعتماد را فراهم می‌کند و تضمین می‌کند که همه داده در نهایت به مقصدشان می‌رسند . همچنین می‌بینیم که تلفن اینترنتی بطور نمونه بوسیله UDP اجرا می‌شود . هر طرف از کاربرد تلفن اینترنتی احتیاج به ارسال داده در طول شبکه در سرعت حداقل یکسان دارد (فایل صوتی زمان حقیقی در شکل ۴-۲ می‌بینید) ؛ این بیشتر به نظر می‌رسد با

www.it98.ir

UDP باشد تا با TCP . همچنین ، کاربردهای تلفن اینترنتی دارای نوسان تلفات هستند ، بنابراین احتیاجی به سرویس انتقال داده قابل اعتماد توسط TCP ندارد.با نبودن سرویس‌های توان عملیاتی و زمانی ، آیا برنامه‌هایی که نیاز به این دو سرویس دارند قادر به اجرا نخواهند بود ؟ پاسخ منفی است . در اینترنت سالهاست که برنامه‌های حساس به زمان در حال اجرا هستند . این برنامه‌ها طوری طراحی شده‌اند که بتوانند بدون تضمین‌های زمانی و پهنای باند بتوانند بر مشکلات فائق ایند و سرویس‌های قابل قبولی را ارئه دهند .در مورد چندین از این طراحی‌ها در فصل ۷ توضیح می‌دهیم . با این حال ، طراحی هوشمندانه محدودیتهای خود را هنگامیکه تاخیر زیاد است دارد ، که این حالت در اینترنت عمومی اغلب است . بطور خلاصه ، اینترنت امروزی می‌تواند اغلب خدمات در خور مناسبی برای برنامه‌های حساس به زمان را تهیه و آماده کند ، اما نمی‌تواند هیچ تضمینی در مورد پهنای باند و یا زمانبندی بدهد ، در فصل ۷ ، همچنین بحث خواهیم کرد که مدلهای سرویس اینترنتی فوری که خدمات جدید را فراهم می‌کنند ، شامل ضمانت سرویس تاخیر برای برنامه‌های حساس با زمان

Application	Application-Layer Protocol	Underlying Transport Protocol
Electronic mail	SMTP [RFC 2821]	TCP
Remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
File transfer	FTP [RFC 959]	TCP
Streaming multimedia	HTTP (e.g., YouTube), RTP	TCP or UDP
Internet telephony	SIP, RTP, or proprietary (e.g., Skype)	Typically UDP

Figure 2.5 ♦ Popular Internet applications, their application-layer protocols, and their underlying transport protocols

www.it98.ir

www.it98.ir

۶-۱-۲ برنامه‌های شبکه پوشش داده شده در این کتاب

قلمرو عمومی و خصوصی جدید کاربردهای اینترنت هر روزه در حال توسعه و پیشرفت هستند . بجای پوشش دادن تعداد زیادی از برنامه‌های اینترنت در یک حالت دایره‌المعارفی ، تصمیم گرفتیم بر روی تعداد کمی از برنامه‌ها که هم مهم و هم فراگیر هستند تمرکز کنیم . در این فصل پنج برنامه مهم را مورد بحث قرار می‌دهیم : وب ، انتقال فایل ، پست الکترونیکی ، سرویس دایرکتوری و اشتراک‌گذاری فایل P2P . ابتدا در مورد وب بحث می‌کنیم . نه فقط بخاطر اینکه این یک برنامه پرکاربرد وسیع است ، همچنین به این دلیل که پروتکل لایه کاربرد آن، HTTP ، ساده و یادگیری آن آسان است . بعد از وب ، بطور خلاصه FTP را بررسی می‌کنیم، به این دلیل که یک پیوند زیبا با HTTP ایجاد می‌کند. سپس پست الکترونیکی را مورد بحث قرار می‌دهیم. اولین برتامه بسیار جذاب اینترنت . ایمیل بسیار پیچیده‌تر از وب در این حالت است زیرا نه فقط از یکی بلکه از چندین پروتکل لایه کاربرد باید استفاده کند . بعد از ایمیل ، به بحث در مورد DNS می‌پردازیم که یک سرویس دایرکتوری برای اینترنت آماده می‌کند . بیشتر کاربران بطور مستقیم با DNS ارتباط برقرار نمی‌کنند ؛ بجای آن کاربران DNS را بصورت غیرمستقیم از میان دیگر برنامه‌ها احضار می‌کنند (که شامل وب ، انتقال فایل و پست الکترونیکی می‌شود). DNS به زیبایی نشان می‌دهد که چگونه یک بخش از عملکرد شبکه میانی (هسته) (انتقال اسم شبکه به آدرس شبکه) می‌تواند در لایه کاربرد در لایه کاربرد در اینترنت پیاده‌سازی شود . در پایان اشتراک‌گذاری فایلP2P را مطالعه می‌کنیم ، که بوسیله بعضی معیارها بیشترین کاربرد گروه برنامه‌ها در اینترنت امروزی را دارد .

۲-۲ وب و HTTP :

تا سال ۱۹۹۰ اینترنت اساسا بصورت مداوم توسط محققان ،دانش آموزان و دانشجویان برای وصل شدن به میزبانان دور دست برای انتقال قایل‌ها از میزبانان مبدا به میزبانان دوردست و بلعکس برای دریافت و ارسال اخبار و دریافت و ارسال نامه‌های الکترونیکی استفاده می‌شد . با اینکه این کاربردها بسیار مفید بودند ، اینترنت در خارج از دانشگاه‌ها و موسسه‌های تحقیق کاملا ناشناخته بود . سپس در اوایل ۱۹۹۰ یک شاخه جدید کاربرد بر روی کار آمد –شبکه گسترده جهانی (world wide web) [بارنز لی ۱۹۹۴] . شبکه کاربردی از اینترنت است که مورد توجه مردم عامه قرار گرفت . شبکه بصورت مهیجی چگونگی برخورد انسان‌ها در داخل و خارج از محیط کارشان را تغییر داد . شبکه اینترنت را از فقط یکی از بسیار داده شبکه‌ها بودند بالاتر برد و تبدیل به تنها داده شبکه شده . (شامل شبکه‌های آنلاین مانند Prodigy , American online,compuserve ، و شبکه داده‌های ملی مانند minitel , transpac در فرانسه و قالب شبکه‌های ذخیره)

تاریخ با رسیدن به فناوریهای ارتباط الکترونیکی به صورت تکه تکه است که برخوردها و اثرات اجتماعی اصلی و مهمی را دارا بود . اولین اینچنین فناوری تلفن بود که در سال ۱۸۷۰ اختراع شد . تلفن به دو نفر اجازه می‌داد با هم بصورت شفاهی در زمان حقیقی بدون اینکه در یک مکان فیزیکی یکسان باشند ارتباط برقرار کنند . فناوری ارتباط الکترونیکی قابل توجه بعدی پخش رادیو/تلویزیون بود که در سال ۱۹۲۰ و ۱۹۳۰ ظهور کرد .

پخش رادیو/تلویزیون اجازه می‌داد انسانها مقادیر وسیعی از اطلاعات صوتی و تصویری دریافت کنند. اینترنت هنوز یکی دیگر از فناوری ارتباط الکترونیکی با اثرات اجتماعی مهم است . ایمیل و شبکه، دو نمونه از کاربردهای بسیار عامه‌پسند شبکه ، کاملا چگونگی زندگی و کار انسان را متحول کرده است . در مورد ایمیل در قسمت ۴-۲ بحث می‌کنیم . در این بخش شبکه و پروتکل لایه کاربردی شبکه را پوشش می‌دهیم .

شاید بیشترین چیزی که برای کاربران جاذبه داشته باشد عملیات شبکه بصورت درخواستی است . کاربران چیزی دریافت می‌کنند ، چیزیکه در زمانیکه می‌خواهند . برخلاف پخش رادیو/تلویزیون که کاربران را مجبور می‌کنند که خود را تنظیم کنند با زمانیکه تهیه

www.it98.ir

www.it98.ir

کننده اطلاعات ، اطلاعات را قابل دسترس و فراهم کند . بعلاوه درخواستی بودن ، شبکه دارای دیگر ویژگی‌های شگفت انگیز دیگری است که انسانها دوست دارند و لذت می‌برند . شبکه برای هر شخص بسیار ساده است تا بتواند اطلاعات قابل دسترسی را در فضای شبکه ایجاد کند . هر کس می‌تواند با یک هزینه بسیار ناچیز ناشر شود . فوق رابط‌ها و موتورهای جستجوگر به ما کمک می‌کنند تا در میان اقیانوس سایت‌های شبکه دریانوردی کنیم . گرافیک‌ها احساسات ما را تحریک می‌کنند . اشکال ریز برنامه‌های جاوا ، و بسیار ابزار و وسایل دیگر ما را فعال می‌کنند برای ارتباط برقرار کردن با صفحات و سایت‌ها . و بیشتر و بیشتر ، شبکه یک لیست واسطه برای انبار کردن و گذاشتن مقادیر وسیعی از برنامه‌های صوتی و تصویری در اینترنت را فراهم می‌کنند –چند رسانه‌ای که می‌تواند به صورت اختیاری و درخواستی در دسترس باشد .

۱-۲-۲ نگاهی بر HTTP

پروتکل (قرارداد) انتقال فوق متن‌ها (HTTP) ، پروتکل لایه کاربرد شبکه ، در قلب شبکه قرار دارد . HTTP در [RFC 1945] و [RFC 2616] تعریف شده است . HTTP در دو برنامه پیاده سازی شده است . یک برنامه مشتری و یک برنامه سرویس دهنده . برنامه مشتری و برنامه سرویس دهنده بوسیله سیستم های مقصد متفاوتی اجرا می شوند ، و بوسیله معادل کردن پیام های HTTP با هم در ارتباط هستند . HTTP ساختار این پیام ها و چگونگی مبادله کردن پیام های مشتری و سرویس دهنده را تشخیص می دهد . قبل از اینکه جزییاتHTTP را توضیح دهیم ، باید مرروی بر بعضی از اطلاعات شبکه داشته باشیم .

یک صفحه شبکه (web page) (همچنین مدرک نیز گفته می شود) شاکل اهدافی می شود . یک هدف بصورت ساده یک فایل است .مانند یک فایلHTML ، یک عکس GIF ، یک عکس ، یک برنامه ریزی جاوا ، یک فایل صوتی و ... که قابل آدرس دهی توسط یکURL تکی است . بیشتر صفحات شبکه شامل یک فایل HTML پایه و چندین اهداف ارجاع شده هستند . برای مثال ، اگر یک صفحه شبکه حاوی متنHTML وپنج عکس JPEG باشد ، بنابراین صفحه شش هدف دارد : فایل HTML پایه به اضافه پنج عکس . فایلHTML پایه دیگر اهداف درون صفحه را با اهداف URLها ارجاع می دهد . هر URL دو جزء دارد : نام میزبان سرویس دهنده که مکان های هدف است و نام مسیر هدف . برای مثال URL :

http://www.someschool.edu/somedepartment/picture.gif

www.someschool.edu برای یک نام میزبان دارد و somedepartment/picture.gif/ برای یک نام مسیر . یک مرورگر (Browser) یک عامل و نماینده کاربرد برای شبکه است ؛ که صفحه شبکه درخواست شده را برای کاربر نمایش می دهد و پیکربندی و قالب آنها را فراهم می کند . بدلیل اینکه جستجوگرهای شبکه جنبه مشتری http را نیز پیاده سازی می کنند ، دریافت شبکه ، از کلمه های جستجوگر و مشتری بطور متغیر استفاده می کنیم .

جستجوگرهای شبکه عمومی و معروف شامل Netscape Communication و Microsoft Internet می شوند . یک سرویس دهنده شبکه ، اهداف شبکه را در خود جای می دهد ، هر کدام که بوسیله URL قابل آدرس دهی باشند . سرویس دهنده های شبکه همچنین جنبه سرویس دهندهHTTP را نیز پیاده سازی می کنند . جستجوگرهای عمومی و معروف شبکه شامل Apache , microsot internet information server می شوند . http مشخص و تعریف می کند که چگونه مشتریان شبکه (مانند جستجوگرها) صفحات شبکه را درخواست می دهند از جزییات سرویس دهنده های شبکه و چگونه سرویس دهنده ها صفحات شبکه را به مشتریان انتقال می دهند . در مورد اثر متقابل مشتری و سرویس دهنده ها بعدا بحث می کنیم ، اما نظریه عمومی در شکل ۶-۲ روشن بیان شده است . زمانیکه یک کاربر یک صفحه شبکه را تقاضا می کند (برای مثال ، بر روی فوق رابط ها کلیک می کند) ،جستجوگر پیام درخواست برای اهداف این صفحه را به سرویس دهنده ارسال می کند . سرویس دهنده درخواست را دریافت می کند و بوسیله پیام های پاسخ http که شامل اهداف می شود ، پاسخ می دهد . تا سال ۱۹۹۷ کاملا همه جستجوگران

www.it98.ir

wWw.it98.ir

www.it98.ir

و سرویس دهنده های شبکه نسخه http/1.0 را پیاده سازی می کردند ، که بنام RFC 1945 تعریف می شده به اوایا ۱۹۹۸ ، سرویس دهنده های شبکه و جستجوگران شروع به پیاده سازی نسخهHTTP/101 کردند ، که بنام RFC 2616 تعریف می شد . HTTP 101 سازگاری عکس باHTTP/1.0 دارد . یک سروی دهنده شبکه اجرایی ۱۰۱ می تواند ارتباط برقرار کند با یک جستجوگر اجرایی ۱٫۰ و یک جستجوگر اجرایی ۱۰۱ می تواند با یک سرویس دهنده اجرایی ۱٫۰ ارتباط برقرار کند . بدلیل اینکه اکنون HTTP/101 فراگیر است ، پس از این هنگامیکه به HTTP رجوع می کنیم به HTTP 101 رجوع کرده ایم .

HTTP ازTCP بعنوان پروتکل انتقال اصولی خود استفاده می کند . HTTP مشتری ابتدا یک ارتباط ابتدایی TCP با سرویس دهنده برقرار می کند . وقتی که ارتباط محقق شد ، جستجوگر و پردازش گران سرویس دهندهTCP از میان سوکت های واسطه دسترسی می یابند . همانطور که در بخش ۱-۲ توضیح دادیم در طرف مشتری سوکت واسطه دری است میان پردازش سرویس دهنده و ارتباطTCP .

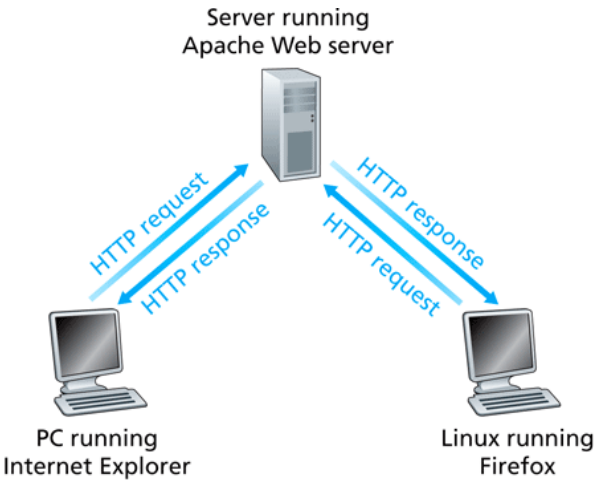


Figure 2.6 ♦ HTTP request-response behavior

مشتری پیام های درخواستHTTP را بدون سوکت واسطه خود ارسال می کند و پیام های پاسخ HTTP را از سوکت واسطه خود دریافت می کند . بطور مشابه سرویس دهندهHTTP پیام های ئرخواست را از سوکت واسطه خود دریافت می کند و پیام های واسطه را به درون سوکت واسطه خود ارسال می کند . هنگامیکه واسطه یک پیام به سوکت واسطه خود ارسال می کند ، پیام از دسترس مشتری خارج است و در دسترس TCP قرار می گیرد . به یاد آوردن از قسمت ۱-۲ کهTCP یک سرویس انتقال اطلاعات قابل اعتماد به HTTP فراهم می کند . این بدین معنی است که هر پیام درخواست HTTP منتشر شده بوسیله یک پردازش مشتری سرانجام دست نخورده و سالم به سرویس دهنده می رسد . بصورت مشابه هر پیام پاسخ HTTP منتشر شده بوسیله پردازش سرویس دهنده سرانجام دست نخورده و سالم به مشتری می رسد .

در اینجا ما یکی از مهمترین فواید معماری لایه دار را می بینیم . HTTP نگران کم شدن و از دست دادن اطلاعات و جزئیات چگونه TCP از تلفات و سفارش‌دهی مجدد اطلاعات در محدوده شبکه را بازیابی می‌کند. این وظیفه TCP و پروتکل‌ها در لایه‌های پایین‌تر دستهٔ پروتکل است.

این وظیفهٔ مهم برای یادداشت که سرویس‌دهنده فایل‌های درخواست شده را به مشتریان بدون ذخیره کردن هیچ وضعیتی از اطلاعات دربارهٔ مشتری ارسال کننده. اگر یک مشتری دیگر برای یک هدف مشابه در یک بازه زمانی بسیار کوتاه دوباره درخواست کند و سرویس‌دهنده با گفتن اینکه این هدف چند لحظه پیش به مشتری ارسال شده است پاسخ نمی‌گوید و به‌جای آن سرویس-

www.it98.ir

www.it98.ir

دهنده مجدداً هدف را ارسال می‌کند، به‌گونه‌ای که کاملاً فراموش کرده چند لحظه پیش این کار را کرده است. بدلیل اینکه یک سرویس دهنده HTTP هیچ اطلاعاتی درباره مشتری نگهداری نمی‌کند، HTTP را پروتکل بدون تابعیت (بدون حالت) می‌گویند.

همچنین باید خاطر نشان کرد که شبکه از معماری مشتری-سرویس‌دهنده استفاده می‌کند. همانگونه که در بخش ۱-۲ توضیح دادیم، یک شبکه سرویس‌دهنده همیشه روشن است، با یک IP آدرس ثابت و به درخواست‌ها از ملیون‌ها جستجوگر متفاوت فعال سرویس‌دهی می‌کند.

۲,۲,۲ ارتباطات ناپایدار و پایدار:

HTTP می‌تواند هر دو ارتباطات پایدار و ناپایدار استفاده می‌کند. اگرچه HTTP ارتباطات پایدار را بعنوان حالت پیش فرض خود استفاده می‌کند و HTTP مشتری و سرویس‌دهنده می‌توانند همچنین بجای ارتباطات ناپابدار استفاده می‌شوند. هر دو ارتباطات HTTP پایدار و ناپایدار را در این بند فرعی آزمایش می‌کنیم.

ارتباطات ناپایدار

اجازه بدهید قدم به قدم انتقال دادن یک صفحه شبکه از سرویس دهنده به مشتری برای اتصالات ناپیدار را برررسی کنیم. فرض کنید صفحه شامل یک فایل HTML پایه و ۱۰ عکس JPEG باشد و همه این ۱۱ هدف در یک سرویس دهنده مشترک قرار داشته باشند. فرض کنید URL برای فایل پایه HTML برابر است با :

http : // www.someschool.edu/some deprtment / hom.index

این اتفاقات می افتد :

۱. پردازش مشتری HTTP یک ارتباط TCP به سرویس دهنده www.someschool.edu بر روی پرت شماره ۸۰ که شماره پرت پیش فرض برای HTTP است را آغاز می کند.

۲. HTTP مشتری یک پیام درخواست HTTP به سرویس دهنده توسط سوکت وابسته خو.د با اتصال TCP ارسال می کنمد. پیام درخواست شامل اسم مسیر است. /some department/home.index (درباره جزئیات پیام HTTP در پایین بحث می کنیم).

۳. پردازش سرویس دهنده HTTP پیام درخواست را به وسیله سوکت وابسته خود و با اتصال دریافت می کند ، اهداف را از حافظه کمکی خود بازیابی می کند. (Disk یا RAM) /some Department/ home. Index. و هدف را درون یک پیام پاسخ HTTP قرار می دهد و پیام پاسخ را به مشتری توسط سوکت خود ارسال می کند.

۴. پرداز سرویس دهنده HTTP به TCP می گوید که اتصال TCP را قطع کند. (اما TCP بطور کامل ارتباط را تا زمانی که مطمئن شود که مشتری پیام پاسخ را کامل و دست نخورده دریافت کرده است، پایان نمی دهد).

۵. مشتری HTTP پیام پاسخ را دریافت می کند. اتصال TCP خاتمه می یابد. پیام که نشان می دهد که هدف ذخیره شده یک فایل HTML است. مشتری فایل را از پیام پاسخ استخراج می کند، فایل HTML را آزمایش می کند و مراجع ۱۰ هدف JPEG را پیدا می کند.

۶. هر ۴ مرحله اول برای هر هدف JPEG ارجا داده شده تکرار می شود.

wWw.it98.ir

www.it98.ir

هنگامی که جستجوگر صفحه وب را دریافت می کند ، صفحه را برای کاربر نمایش می دهد. در جستجوگر متفاوت ممکن است یک صفحه شبکه را به روش های مختلفی تفسیر کند (همان نمایش به کاربرد). HTTP هیچ کاری با چگونگی تفسیر نمایش یک صفحه شبکه به وسیله مشتری ندارد. ویزگی های HTTP (RFC 1945) و (RFC2616) تنها با پروتکل ارتباط میان برنامه HTTP مشتری و برنامه HTTP سرویس دهنده شناسایی می شود.

مراحل بالا استفاده از ارتباطات ناپیدار را بیان می کند، جایی که هر اتصال TCP بعد از ارسال که هر اتصال TCP یک پیام درخواست و یک پیام پاسخ را ردو بدل می کند. بدین ترتیب ، در این مثال، وقتی یک کاربرد صفحه شبکه را درخواست می کند، اتصال TCP تولید شده است.

در مراحل بالا توضیح داده شده، بطور عمده درباره آیا اینکه مشتری JPEG10 از روی اتصالات TCP سری بدست می آورد. آیا اینکه بعضی از JPEG ها از روی اتصالات TPC موازی بدست می آورد، مبهم است. قطعا کاربران می توانند جستجوگرهای پیشرفته ای را برای کنترل درجه تقارن و موازات طرااحی کنند.

در حالت پیش فرضشان، بیشتر جستجوگرهای ۵ تا ۱۰ اتصال TCP موازی را باز می کنند، و هر یک از این اتصالات ترتیب یکی از نقل و انتقالات و درخواست و پاسخ را می دهد. اگر کاربر ترجیح دهد ، مقدار ۱۰۰ ماکزیممم اتصالات موازی به روی عدد۷ تنظیم شود، در این حالت ۱۰ اتصال به صورت موازی برقرار می شوند. همانطور که در فصل بعد خواهیم دید، استفاده از اتصالات موازی زمان پاسخ گویی را کاهش می دهد.

قبل از اینکه ادامه دهیم، اجازه دهید محاسبه برگشت نامه را برای تخمین مقدار زمانی که از شروع جریان از آن موقع که مشتری فایل پایه HTML را درخواست می دهد تا تمام فایل به وسیله مشتری دریافت شود انجام دهیم. در پایان این محاسبه، زمان دور گردشی (RTT) را بدست می آوریم، و این زمانی است که برای یک ارسال بسته کوچک از مشتری به سرویس دهنده و سپس برگشت به مشتری لازم است. RTT شامل تاخیرات انتشار بسته، (این تاخیرات در قسمت ۱,۶ بحث شده است) است.

اکنون در نظر می گیریم چه اتفاقاتی هنگامی که یک کاربر بر روی فوق ارتباط کلیک می کند می افتد. همانگونه که در شکل ۲,۷ نشان داده شده است، این سبب می شود که جستجوگر یک اتصال TCP میان جستجوگر و سرویس دهنده شبکه آغاز کند.

و این شامل یک ارتباط (دست تکانی) سه مسیره می شود- مشتری یک بخش کوچک TCP را به سرویس دهنده ارسال می کند، یرویس دهنده شناسایی و تایید می کند و با یک قطعه کوچک TCP پاسخ می دهد و در نهایت تاثیرات را به سرویس دهنده برگشت می دهد. دو قسمت اول ارتباط سه مسیره یک RTT زمان می برد. بعد از کامل شدن دو قسمت اول از ارتباط، مشتری، ترکیب شده پیام درخواست HTTP با قسمت سوم ارتباط سه مسیره را (شناسائیات و تاییدات) به درون اتصال TCP ارسال می کند. هنگامی که پیام درخواستبه سرویس دهنده می رسد ، سرویس دهنده فایل رلا HTML را به درون اتصال TCP ارسال می کند. این درخواست/ پاسخ HTTP یک RTT دیگر زمان مصرف می کند. بدین ترتیب، تقریبا کل زمان پاسخ RTT۲ به زمان انتقال فایل HTML به سرویس دهنده.

اتصالات پایدار

اتصالات ناپایدار کمبودها و نقص هایی داشت. اولا، اتصال مدل جدید باید برای هر هدف درخواست شده برقرار و نگهداری شده باشد. برای هر کدام از این اتصالات ،

www.it98.ir

www.it98.ir

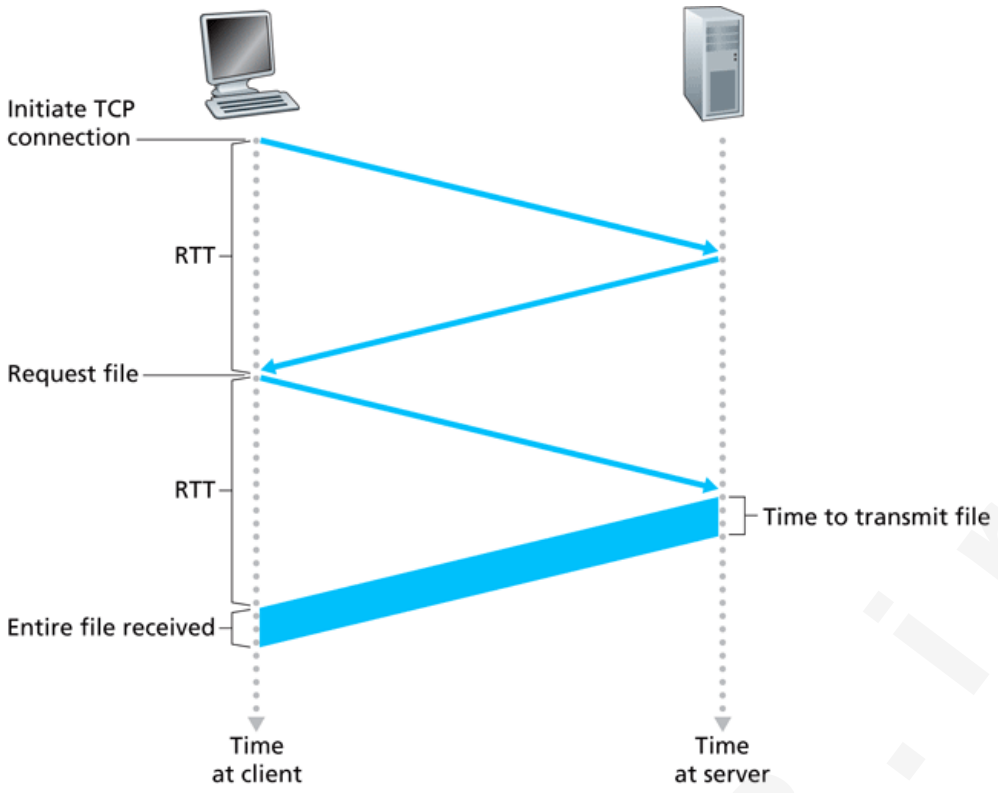


Figure 2.7 ♦ Back-of-the-envelope calculation for the time needed to request and receive an HTML file

ذخایر TCP باید تخصیص یافته باشند و متغییرات TCP باید هم در سرویس دهنده و هم در مشتری نگه داشته شود. این می تواند یک بار مسئولیت جدی بر روی سرویس دهنده شبکه قرار دهد، که این ممکن است با سرویس دهی به درخواست هم زمان صدها مشتری متفاوت باشد. دوما، همانطور که توضیح دادیم، هر هدف متحمل یک تحویل تاخیر دو RTT می شود – یک RTT برای برقراری اتصال RTT و یک RTT برای درخواست و دریافت یک هدف.

با اتصال پایدار، سرویس دهنده اتصال TCP را بعد از ارسال یک پاسخ باز نگه می دارد. پیامد درخواست ها و پاسخ ها میان مشتری و سرویس دهنده یکسان به وسیله یک اتصال مشابه می تواند ارسال شود. مخصوصا، یک صفحه شبکه کامل (در مثال بالا، فایل پایه HTML و۱۰ عکس) می تواند به وسیله یک اتصال TCP پایدار سرویس دهنده به مشتری یکسان توسط یک اتصال TCP پایدار تک ارسال شود. بطور نمونه، سرویس دهنده HTTP یک اتصال را می بندد زمانی که برای یک زمان مشخص استفاده نداشته باشد.

دو نسخه اتصال پایدار وجود دارد : بدون تغییر مسیردهی و با تغییر مسیردهی. برای نسخه بدون تغییر مسیر دهی، مشتری یک درخواست جدید فقط هنگامی که درخواست قبلی رسیده باشد صادر می کند. در این حالت مشتری یک RTT برای اینکه درخواست و دریافت هریک از اهداف ارجاع شده تجربه می کند.(۱۰عکس در مثال بالا). اگرچه این بهبود نسبت به دو RTT برای هر هدف ناپایدار است. تاخیر RTT می تواند با تغییر مسیردهی بیشتر کاهش پیدا کند. یکی دیگر از نامطلوبی های بدون تغییر مسیر دهی این است که بعد از اینکه سرویس دهنده یک هدف را به وسیله اتصال TCP پایدار ارسال کرد، اتصال بیهوده کار می کند- هیچ عملی انجام نمی دهد- زمانی که برای رسیدن درخواست بعدی منتظر می ماند. این بیهودگی منابع سرویس دهنده را هدر می‌دهد.

wWw.it98.ir

www.it98.ir

حالت پیش فرض HTTP اتصالات پایدار با تغییر مسیردهی را استفاده می کند. با تغییر مسیردهی، مشتری HTTP یک درخواست به محض اینکه با یک مرجع برخورد می کند، صادر می کند. بدین ترتیب مشتری HTTP می تواند درخواست های پشت به پشت برای اهداف ارجاع شده ایجاد کند.یعنی اینکه می تواند یک درخواست جدید قبل از دریافت کردن یک پاسخ به درخواست قبلی ایجاد کند. هنگامی که سرویس دهنده درخواست های پشت به پشت را دریافت می کند، اهداف را پشت به پشت ارسال می کند. با تغییر مسیر دهی این ممکن است که فقط یک RTT برای همه اهداف ارجاع شده زمان مصرف شود. بعلاوه، اتصالTCP تغییر مسیر داده شده برای یک کسر زمانی کمتری بیهوده باقی می ماند. ما بطور کمی نحوه کار و اجرای اتصالات پایدار و ناپیدار را در تمرینات فصل های ۲ و ۳ مقایسه خواهیم کرد. و همچنین شما را به دیدن دعوت می کنیم (هایدمان ۱۹۹۷، نیلسن ۱۹۹۷).

۳-۳-۲ قالب پیام های HTTP

HTTP بویژه (RFC2616) شامل شناسایی قالب های پیام HTTP می شود. دو نوع از پیام های HTTP وجود دارد، پیام های درخواست و پیام های پاسخ ، هر دوی آنها در پایین بحث شده است.

پیام درخواست HTTP

در پایین یک نمونه از پیام درخواست HTTP را آورده ایم :

```
GET/Soxed/page.html HTTP/1.1
HOST :www. Someschocl. Edu
Connection : close
User-agent : mozila/4.0
ACCEPT –languge : fr
```

یک یک نگاه دقیق به این پیام درخواست ساده می توانیم چیزهای بسیاری یاد بگیریم. همه ، می بینیم که با کد اسکی معمولی نوشته شده است. بنابراین سواد کامپیوتری معمولی انسان می تواند این را بخواند. دوم ، می بینیم که پیام شامل ۵ خط است، هر کدام بوسیله یک carriage return و یک line feed همراهی می شود. خط آخر به وسیله یک و یک اضافه همراهی می شود. اگرچه این پیام درخواست مخصوص ۵ خط دارد، یک پیام درخواست می تواند بیشتر از این ها و یا حتی یک خط داشته باشد. اولین خط از پیام درخواست مسج را خط درخواست (request line) می گویند : خط های بعدی را خط های عنوان (healer line) می گویند. خط درخواست سه حوزه دارد : حوزه روش، حوزه url، حوزه نسخهhttp.

حوزه روش می تواند چندید ارزش متفاوت را داشته باشد، شامل HEAD,POST,GET .

روش GET هنگامی که جستجوگر یک هدف را درخواست می دهد استفاده می شود، با شناسایی هدف درخواست شده در حوزه URL. در این مثال ، جستجوگر هدف زیر را درخواست کرده است: /somedir/page.html . این نسخه خود توصیفی است و در این مثال، جستجوگر نسخه HTTP/101 را پیاده سازی کرده است.

اکنون نگاهی به خطوط عنوان در این مثال می اندازیم. خط هدف HOST : www.somechocl.edu مشخص می کند میزبان هر کدام هدف دارد. شما ممکن است فکر کنید که این خط عنوان لازم نباشد ، همانطور که یک اتصال TCP در میزبان هم اکنون قرار دارد. اما همانطور که در قسمت ۲,۲,۶ خواهیم دید، اطلاعات تهیه شده به وسیله خط عنوان میزبان به وسیله حافظه

www.it98.ir

www.it98.ir

پنهانی پرکسی شبکه لازم است. با در بر داشتن خط عنوان اتصال : بسته (connection : close) جستجوگر به سرویس دهنده می گوید که نمی خواهد با اتصالات پایدار جهت بیافتد، می خواهد که سرویس دهنده اتصال را بعد از ارسال هدف درخواست شده ببندد. خط عنوان user- agent نماینده کاربر را تعیین می کند. که همان نوع جستجوگر است که درخواست به سرویس دهنده را ایجاد می کند. در اینجا نماینده کاربر mozila/4.0. یک جستجوگر netscape است. این خط عنوان به دلیل اینکه سرویس دهنده در واقع می تواند نسخه های متفاوتی از هدف مشابه برای مدل های مختلفی از نماینده کاربر ارسال کند. مفید و سودمند است. (هر یک از نسخه ها به وسیله url مشابه آدرس داده می شوند). در آخر، عنوان ACCEPT- LANUAGE نشان می دهد که کاربر ترجیح می دهد که یک دستوری از هدف را دریافت کنند و اگر همچنین هدفی در سرویس دهنده وجود داشته باشد؛ در غیر این صورت سرویس دهنده باید نسخه پیش فرض خود را ارسال کند. عنوان ACCEPT-LANGUAGE تنها یکی از بسیار عنوان های قابل استفاده و در دسترس درون HTTP است.

به یک مثال دیگر نگاه می کنیم؛ اکنون اجازه دهید به قالب عمومی یک پیام درخواست نگاهی بیاندازیم، همانگونه که در شکل 2.8 نشان داده شده است. می بینیم که قالب عمومی بسیار زیاد از مثال قبلی مان پیروی می کند. ممکن است متوجه شده باشید ، هرچند بعد از خطوط عنوان (Line feed carrge return و اضافه) یک ماهیت (entity body) وجود دارد. ماهیت بدنه با روش GET خالی است، اما با روش POST مورد استفاده قرار می گیرد. یک مشتری HTTP معمولا از روش POST هنگامی که کاربر یک فرم را پر می کند ، استفاده می کند.- برای مثال، هنگامی که یک

کاربر کلمات جستجو را برای موتور جستجو فراهم می کند- بای یک پیام POST ، کاربر همچنان یک صفحه شبکه از سرویس دهنده را درخواست می کند، اما محتوای خاص صفحه شبکه به چیزی که کاربر درون زمینه فراهم کرده است بستگی دارد. اگر مقدار POST است ، سپس ماهیت بدنه شامل چیزهایی که کاربر درون زمینه فرم وارد کرده است می شود.

ما بی مبالات می بودیم اگر اشاره نکرده بودیم که یک درخواست انتشار یافته با یک فرم لازم نیست استفاده می‌شود. از روش POST استفاده شود. در عوض ، فرم های HTML معمولا روش GET استفاده می شود و شامل داده وارد شده (در حوزه نرم) درون URL درخواست شده می شود. برای مثال اگر یک فرم روش GET را استفاده کند، دو حوزه دارد، و درون دو حوزه bananas , monkeys وارد شده باشد و بنابراین URL چنین ساختاری خواهد داشت :

www.some site.com/animal search?monkeys8 bananas.

درون صفحه بندی روزانه شبکه تان ، احتمال متوجه URLهای گسترش یافته از این مدل گردیده اید. روش HEAD شبیه به روش GET می باشد. وقتی که یک سرویس دهنده یک درخواست با روش HEAD دریافت کند، با یک پیام HTTP پاسخ می دهد اما هدف درخواست شده را از قلم می اندازد. کاربرد توسعه یافته معمولا از روش HEAD برای اشکال زدایی استفاده می کند. ویژگی HTTP/1.0 فقط برای سه مدل از روش ها کاربرد دارد : HEAD, POST , GET بعلاوه، برای این سه روش ، ویژگی – اما HTTP برای چندین روش اضافه کاربرد دارد. که شامل DELETEPUT می شود. روش PUT غالبا صرف ربط با ابزار انتشار شبکه استفاده می شود. این به کاربر اجازه می دهد تا یک هدف را در مسیر ویژه (بطور مستقیم) در یک سرویس دهنده شبکه ویزه بارگذاری کند. روش PUT همچنین به وسیله کاربردهایی که احتیاج به بارگذاری اهداف به سرویس دهنده شبکه دارند مورد استفاده قرار می گیرند. روش DELETE به یک کاربر اجازه می دهد ، یا یک کاربر ، که یک هدف را از یک سرویس دهنده شبکه پاک کند.

www.it98.ir

wWw.it98.ir

www.it98.ir

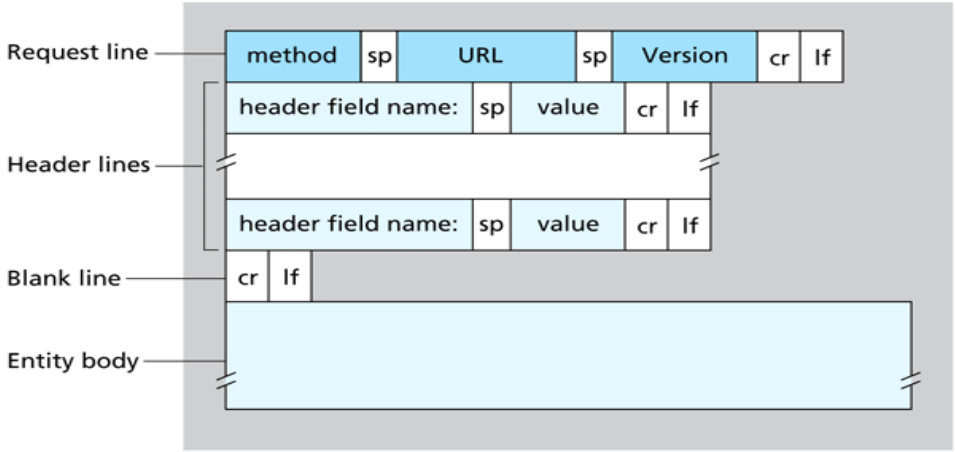


Figure 2.8 ♦ General format of an HTTP request message

پیام پاسخ HTTP

در پایین ما یک پیام پاسخ نمونه را آماده کرده ایم . این پیام پاسخ می تواند پاسخی به پیام درخواست مثالی که بحث کردیم باشد.

http/1.1 200 ok
connection : close
date : thu , 03 jul 2003 12 :00:15 GMT
server : apache/ 1.3.0 (unix)
last – modified : sun, 5may 2003 04: 23:24 GMT
content – type : text/html
(data data data data …..)

بیایید یک نگاه دقیق به این پیام بیندازیم. این پیام سه قسمت دارد :» یک خط وضعی ابتدایی (STATUS LINE) ، ۶ تا خط عنوان (HEADER LINEES)، و سپس ماهیت بدنه (entity body) . ماهیت بدنه پیام است- شامل هدف درخواست شده خود است(بوسیله data data data data پاسخ داده می شود). خط وضعی سه حوزه دارد : حوزه نسخه پروتکل (protocol versem rild) ، یک کد وضعی و یک پیام وضعی متقابل. در این مثال، خط وضعی نشان می دهد که سرویس دهنده از http/1.1 استفاده می کند و اینکه همه چیز ok است. (یعنی اینکه سرویس دهنده پیدا شده ، و ارسال کرده و هدف را درخواست داده).

اکنون اجازه دهید نگاهی به خطوط عنوان کنیم. سرویس دهنده خط عنوان connection : close را استفاده می کند. برای اینکه مشتری بگوید که قصد دارد اتصال TCP را بعد از ارسال پیام قطع کند. خط عنوان : DATE زمان و تاریخ موقعی پاسخ HTTP ساخته و به وسیله سرویس دهنده ارسال شده بود را نشان می دهد.

یادآوری می‌کنیم که این زمانی که هدف درستت شده یا آخرین اصلاحات رویش انجام شده است، نیست. بین زمانی است که سرویس دهنده هدف را از فایل سیستمی خود بازیابی کرده ، هدف را درون پیام پاسخ وارد کرده، و پیام پاسخ را ارسال کرده. خط عنوان The server: نشان می دهد که پیام به وسیله سرویس دهنده شبکه APACHE انتشار یافته است و این متشابه خط عنوان SER- AGENT در پیام پاسخ HTTP است خط عنوان LAST- MODIFIED نشان می دهد که چه در زمان و تاریخ هدف ساخته شده یا آخرین اصلاحات بر رویش انجام شده است. عنوان Last- modified که به زود جزئیات آن را نیز می

www.it98.ir

www.it98.ir

گوییم، شاخصی برای ذخیره سازی هدف است. خط عنوان content- length نشان می دهد که هدف درون ماهیت بدنه که HTML است. (نوع هدف معمولا به وسیله عنوان :content- type مشخص می شود و نه به وسیله گسترش فایل)

به یک مثال دیگر نگاه می کنیم، اجازه دهید اکنون قالب عمومی پیام پاسخ را بررسی کنیم که در شکل ۲,۹ نشان داده شده است. قالب عمومی پیام پاسخ همانند مثال قبلی پیام پاسخ است. فقط چند چند در مورد کدهای وضعی و عبارتشان اضافه می کنیم. کد وضعی عبارت وابسته نتیجه درخواست را نشان می دهد. بعضی کدهای وضعی و عبارات وابسته عبارتند از :

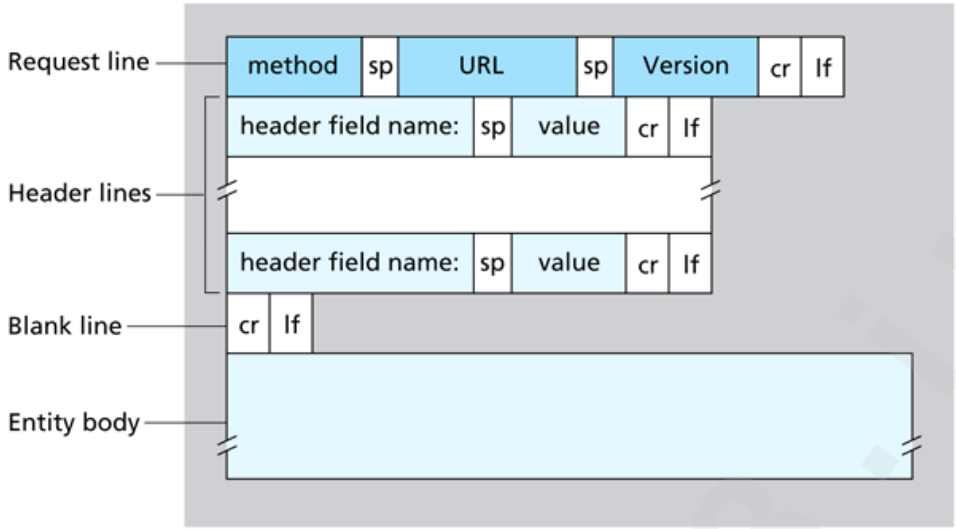


Figure 2.8 ♦ General format of an HTTP request message

- 200 ok : درخواست موفقیت آمیز بود و اطلاعات در پاسخ بازگشت داده شده است.
- 301 moved permanently : هدف درخواست شده بطور پایدار حرکت کرده است. URL جدید در عنوان Location پیام پاسخ مشخص شده است. نرم افزار مشتری بصورت اتوماتیک URL جدید را بازیابی می کند.
- UOO Bad Request : این یک کد خطای عمومی است که نشان می دهد درخواست به وسیله سرویس دهنده قابل فهم نیست.
- Uou noT Found : سند درخواست شده در این سرویس دهنده وجود ندارد.

505 HTTP Version Not supported : نسخه پروتکل http درخواست شده به وسیله سرویس دهنده پشتیبانی نمی شود.

چگونه شما می خواهید که یک پیام پاسخ HTTP واقعی را ببینید؟

این بسیار درخواست شده و بسیار ساده است که انجام شود. ابتدا Telnet برون سرویس دهنده شبکه مورد علاقه خود سپس در یک خط پیام درخواست برای بعضی اهداف که در سرویس دهنده قرار دارد را بنویسید. برای مثال اگر شما به یک فرمان فوری دسترسی دارید، بنویسید:

telnet cis. Poly . ed 80
GET/-ROSS/HTTP/1.1
HOST :cis.poly.edu

(carriage return را دوباره بعد از نوشتن آخرین خط فشار دهید). این یک اتصال TCP را به پورت ۸۰ میزبان cis.poly.edu باز می کند و سپس پیام درخواست http را ارسال می کند. شما باید یک پیام پاسخ که شامل فایل پایه html صفحه مبدا پروفسور راس (ross) است را ببینید.

wWw.it98.ir

www.it98.ir

اگر شما ترجیح دادید فقط خطوط پیام http را ببینید و خود هدف را دریافت نکنید، get را با head جایگزین کنید. در پایان ، /ross - / /banana-/ جایگزین کنید و ببینید چه مدل از پیام پاسخ شما دریافت می کنید. در این قسمت در مورد تعداد خطوط عنوان کع می تواند در محدوده درخواستhttp و پیام پاسخ استفاده شود. بحث می کنییم. ویزگی http بسیار از خطوط عنوان که می تواند به وسیله جستجوگرها، سرویس دهنده های شبکه و سرویس دهنده های ذخیره شبکه مورد استفاده قرار گیرد را مشخص می کند. ما فقط اندکی از آنان را مورد بحث قرار می دهیم. تعدادی را در ادامه مورد بحث قرار دادیمو تعداد اندکی دیگر را هنگامی که در مورد ذخیره سازی شبه و در قسمت ۲,۲,۶ بحث می کنیم. یک بحث بسیار خواندنی و قابل فهم در پروتکل http که شامل عنوان هایش و کدهای وضعی اش می شود در (kris hnamurty 2001) گنجانده شده است. همچنین برای وسعت دیدتان می توانید (lvotonen 1998) را ببینید. یک معرفی بسیار خوب برای وب نیز است (Yeager 1996).

چگونه یک سرویس دهنده وب چگونه تصمیم می گیرد که کدامیک از خطوط عنوان درون یک پیام پاسخ قرار دارد؟

یک جستجوگر باید خطوط عنوان را یه عنوان دستورالعمل نوشته و نسخه(برای مثال ، یک جستجوگر HTTP/1.5 هیچ یک از خطوط عنوام ۱,۱ را انتشار نخواهد داد). شکل بندی جستجوگر به وسیله کاربر (برای مثال، زبان توضیح داده شده) ، و آیا جستجوگر به تازگی ذخیره سازی شده است، اما ممکن است از تاریخ آن گذشته باشد، و نسخه هدف انتشار خواهد داد. سرویس دهنده های وب شبیه به هم رفتار می کنند : در آن ها محصولات ، نسخه ها، و شکل بندی مختلفی و همه چیزهایی را تحت تاثیر قرار می دهد وجود دارد که درآن خطوط عنوان درون پیام های پاسخ قرار دارند.

۲,۲,۴. برهمکنش کاربر – سرویس دهنده : کوکی ها (COOKIES)

در بالا اشاره کردیم که یک سرویس دهنده HTTP بدون تابعیت و حالت است. این سادگی طراحی سرویس دهنده ها به مهندسان اجازه می دهد که سرویس دهنده ها وب با کارایی بالا که بتواند هزاران اتصال TCP همزمان را ترتیب دهد و بر عهده بگیرد را طراحی و آماده کنند. هرچند، این معمولا برای یک وب سایت برای شناسایی کاربران قابل طراحی است هم بخاطر اینکه سرویس دهنده هاا می خواهند دسترسی کاربر را محدود کنند و یا هم می خواهند محتوای سرویس دهنده را به عنوان تابعی از هویت کاربر داشته باشد. به همبن منظور HTTP از کوکی ها استفاده می کند.

کوکی ها در RFC 2159 تعریف شده اند، و به سایت ها اجازه می دهند که مسیر کاربران را نگهداری کند.اگرچه همه سایت ها از کوکی ها استفاده نمی کنند، بسیاری از شاخه های اصلی (مانند YAHOO) ، تجارت الکترونیکی (مانند AMAZON) و تبلیغاتی (مانندDOUBLE CLICK) بصورت وسیع از کوکی ها استفاده می کنند.

فن آوری کوکی ۴ جزء دارد :

۱. یک خط عنوان کوکی در پیام پاسخ HTTP.
۲. یک خط عنوان کوکی در پیام درخواست HTTP.
۳. یک فایل کوکی نگهداری شده در سیستم مقصد کاربر و اداره شده توسط جستجوگر کاربر
۴. یک بانک اطلاعات bank-end در وب سایت.

بیایید در قالب یک مثال ببینیم کوکی ها چگونه استفاده می‌شوند. فرض می شود سوزان شخصی است که همیشه به وسیله Internet Explorer از کامپیوتر خانگی خود به شبکه دسترسی دارد، با یک سایت تجارت الکترونیکی برا اولین بار ارتباط برقرار می کند، وب سایت یک شماره شناسایی منحصر به فرد و یک ورودی به بانک اطلاعات bank – end خود که با شماره شناسایی شاخص شده است را درست می کند. سپس سرویس‌دهنده به جستجوگر سوزان پاسخ می دهد،

www.it98.ir

www.it98.ir

که شامل یک عنوان set-coolie پاسخ http است. که حاوی عد شناسایی است. برای مثال خط عنوان ممکن است بصورت زیر باشد :

Set- cookie : 1678453

وقتی که جستجوگر سوزان پیام پاسخ http را دریافت می کند و جستجوگر عنوان :set- cooki را می بیند. جستجوگر سپس یک خط را به فایل مخصوص کوکی اضافه می کند که جستجوگر آنرا اداره می کند. این خط شامل اسم میزبان سرویس دهنده و عدد شناسایی در عنوان :set- cookie می شود. هنگامی که سوزان به جستجو ادامه می دهد این سایت تجارت الکترونیکی هر دفعه که صفحه وب درخواست کند، جستجوگرش فایل کوکی اش را در نظر می گیرد، شناسایی اش را برای این سایت استخراج می کند و یک خط عنوان کوکی درون درخواست HTTP قرار می دهد که شامل شماره شناسایی است. به ویژه هر یک از درخواست های HTTP او برای سرویس دهنده تجارت الکترونیکی شامل خط عنوان است. :

COKIE :1678453

در این حالت وب سایت قادر است فعالیت های سوزان در وب سایت را میزبانی کند.اگرچه وب سایت لازم نیست با هم سوزان را بداند، ولی می داند دقیقا کاربر ۱۶۷۸۴۵۳ کدام صفحات را بازید کرده است، و در چه زمانی. سایت تجارت الکترونیکی سپس می تواند استفاده از کوکی ها برای یک سرویس خرید الکترونیکی فراهم و مهیا کند- در طول یک دوره خاص با سایت ، آن سایت می تواند یک لیست کامل از تمام خریدهای سوزان ایجاد کند، که بدین صورت او می تواند پول آنها را بصورت کلی در پایان آن دوره پرداخت کند.

اگر سوزان به سایت باز گردد، مثلا یک هفته بعد، جستجوگر او به گذاشتن خط عنوان COOKIE : 167853 در پیام های درخواست ادامه خواهد داد. سایت تجارت الکترونیکی می تواند محصولاتی به سوزان با توجه به صفحات وبی که در گذشته از سایت دیدن کرده بود را ارائه و پیشنهاد دهد. اگر سوزان همچنین خود را در سایت ثبت نام کند – با استفاده از اسم کامل ، آدرس ایمیل، کدپستی و اطلاعات کارت اعتباری سایت تجارت الکترونیکی در بانک اطلاعاتی او این اطلاعات را نیز وارد می کند.، در نتیجه اسم سوزان با عدد شناسایی او در رابطه هستند (و همه صفحاتی که او از سایت در گذشته بازدید کرده است). بعضی از سایت های تجارت الکترونیکی خرید کردن با یک کلیک را فراهم می کنند – وقتی که سوزان یک کالا را در طول یک بازدید متوالی برای خرید انتخاب می کند، لازم نیست مجددا اسم، شماره کارت اعتباری و آدرس خود را وارد کنید.

از این بحث متوجه می شویم که کوکی ها می توانند برای شناسایی یک کاربر مورد استفاده قرار گیرند. اولین بار که یک کاربر یک سایت را بازدید می کند، کاربر می تواند یک کاربر شناسایی راآماده کند(ممکن است اسم او باشد). جستجوگر سپس یک عنوان کوکی را به سرویس دهنده در طول مدت هم بازدیدهای متوالی از سایت ارسال می کند، در نتیجه کاربر برای سرویس دهنده شناسایی می شود. همچنین دانستیم از این بحث که کوکی ها می توانند برای ساخت ک لایه session کاربر در بالای http بدون تابعیت به کار گرفته شود. بطور مثال زمانی که یک کاربر به یک ایمیل کاربردی وب پایه متصل می شود جستجوگر اطلاعات کوکی را به سرویس دهنده ارسال می کند، سرویس دهنده اجازه می دهد که کاربر را از همه کاربران session با کاربرد شناسایی کند.

اگرچه کوکی ها معمولا کار خرید اینترنتی برای کاربران را ساده کرده اند، اما آنها بسیار بحث برانگیز باقی می مانند زیرا آنها همچنین می توانند بعنوان یک تخلف در مالکیت شخصی یک کاربر در نظر گرفته شوند.

همانطور دیدیم، استفاده از یک ترکیب از کوکی ها و اطلاعات حساب ذخیره شده کاربر، یک وب سایت می تواند چیزهای بسیار در مورد کاربر و احتمالا چیزهایی که به اشخاص می شوند ارائه دهد و بفروشد را بداند. بعلاوه، کوکی ها همچنین می توانند برای جمع آوری اطلاعات در مورد یک رفتار مخصوص کاربرد در طول استفاده تعداد زیادی از وب سایت ها مورد استفاده قرار گیرند. قطعا صفحات وب که بندهای تبلیغاتی را نمایش می دهند ، پیام درخواستHTTP را برای بدست آوردن بندهای تبلیغاتی (که می تواند

wWw.it98.ir

www.it98.ir

www.it98.ir

GIF یا JPEG باشند) از سرویس دهنده یک شرکت تبلیغاتی استفاده کنند. هر کدام از درخواست ها به سرویس دهنده HTTP شرکت تبلیغاتی می تواند حاوی یک کوکی شد که بوسیله شرکت تبلیغاتی اداره می شود. بدلیل اینکه بسیاری از شرکت های تبلیغاتی اینترنتی در بسیاری از وب سایت ها بندهای تبلیغاتی خود را قرار می دهند، شرکت می تواند یک پرونده در مورد طرح های یک جستجو کننده مشخصی در میان چندین سایت بسازد.

ما به خوانندگان کوکب های HTTP وضعیت مشتری پایدار را پیشنهاد می کنیم (NETSCAPE COOLE19999) که بسیار گسترده گفته شده اما یک معرفی کوکی های خواندنی است.

۲,۲,۵ محتوای HTTP

در این فصل بطور ضمنی فرض داشتیم که داده ها یی که در پیام ها پاسخ HTTP حمل می شوند اهدافی از صفحات وب هستند، یعنی اینکه فایل های JPEG, EGIF,HTML، Java applets ریز برنامه های و غیره هستند. ما HTTP را در متن وب برای اینکه بتوانیم یک مثال کاربردی محکم و آشنا بزنیم. اما کم کاری کرده ایم. اگر اشاره نکنیم به اینکه http اغلب برای رد و بدل بسیاری دیگر از انواع فایل‌ها نیز استفاده می شوند.

برای مثال، http غالبا در کاربردهای تجارت الکترونیکی برای رد و بدل فایل های XML از یک ماشین به ماشین دیگر مورد استفاده قرار می گیرند، با وجود اینکه هیچ کدام از این ماشین ها درگیر یک جستجوگر یا یک کاربر نیستند. بانک ها اغلب XML را برای ساختار اطلاعات بانکداری استفاده می کنند.(مانند اطلاعات حساب مشتری)، ماشین های بانکداری غالبا از http برای تغییر این اطلاعات نباشد xml استفاده می کنند. (بحث در موردxml در خارج از مقوله این کتاب است. اینجا فقط می گوییم که یک سند xml عمومی ساختمان داده را فراهم می کند و یک تعریف از معنای داده : بطور عمومی علائم قالب بندی را مانند htm آماده و فراهم نمی کند).HTTP همچنین برای رد و بدل XML صوتی، WML و دیگر انواع فایل XML مورد استفاده قرار می گیرد.بعلاوه، همانطور که در پایان این فصل خواهیم دید، HTTP غالبا بعنوان پروتکل فایل رد و بدل برای به اشتراک گذاری فایل P_2P مورد استفاده قرار می گیرد. و همانطور که در فصل ۷ خواهیم دید ، HTTP مکررا برای درجه بندی بسته های صوتی و تصویری ذخیره شده استفاده می شود.

۲,۲,۶ . ذخیره سازی وب

یک ذخیره وب – سرویس دهنده پروکسی نیز گفته می شود- ماهیت یک شبکه است که درخواست های HTTP را از طرف مبداء سرویس دهنده وب تایید و تصدیق می کند. ذخیره وب لوح ذخیره سازی خود را دارد و در این لوح کپی های درخواست اهداف جدید را نگه داری می کند. همانطور که شکل ۲,۱۰ نشان دادهش ده است، یک جستجوگر کاربر می تواند اینگونه پیکربندی شود که تمام درخواست های HTTP کاربر ابتدا به سمت ذخیره وب هدایت می شوند.

هنگامی که یک جستجوگر پیکربندی شد، هر درخواست جستجوگر برای یک هدف ابتدا به ذخیره وب هدایت می‌شود. بعنوان مثال، فرض کنید یک جستجوگر هدف http://www.someschocol.edu/compus.g.f را درخواست می کند. این اتفاقات رخ می دهد.

۱.جستجوگر یک اتصال TCP را به ذخیره وب برقرار می کند.و یک در خواست HTTP را برای هدف به ذخیره وب ارسال می کند.

۲. ذخیره وب بررسی می کند تا ببیند آیا یک کپی از هدف بطور موضعی ذخیره دارد یا خیر. اگر داشت، ذخیره وب هدف را در محدوده یک پیام پاسخ HTTP به جستجوگر مشتری می فرستد.

www.it98.ir

www.it98.ir

۳. اگر ذخیره وب هدف را نداشت ، وب ذخیره یک اتصال TCP را به سرویس دهنده مبدا باز می کند، که همان WW.SOME.SCHOOL.EDU است. ذخیره وب سپس یک درخوات http برای هدف به درون اتصال tcp می فرستد. بعد از دریافت کردن این درخواست ، سرویس دهنده مبدا هدف را در محدوده یک پاسخ http به ذخیره وب ارسال می کند.

۴. هنگامی که ذخیره وب هدف را دریافت کرده یک کپی از آنرا در محو آن موضعی خود نگه می دارد و یک کپی در محدوده یک پیام پاسخ HTTPبه جستجوگر مشتری ارسال می کند. (بخاطر وجود داشتن اتصال tcp بین جستجوگر مشتری و ذخیره وب)

یادآوری می شود که یک ذخیره هم یک سرویس دهنده و هم یک مشتری در یک زمان است. هنگامی که درخواست ها را از جایی دریافت می کند و به یک جستجوگر پاسخ می دهد، یک سرویس دهنده است. زمانی که درخواست ها را به جایی می فرستد و پاسخ از یک سرویس دهنده مبدا دریافت می کند یک مشتری است.

بطور عمومی یک ذخیره وب به وسیله یک isp خریداری و نصب خواهد. برای مثال، یک دانشگاه ممکن است یک ذخیره را روی محوطه شبکه خود نصب و راه اندازی کند و تمام جستجوگران محوطه را به سمت دستیابی به ذخیره طراحی کند. یا یک isp اصلی (مانند aaol) ممکن است یک یا چند ذخیره در شبکه خود راه اندازی کند و جستجوگران ارسال شده و شکل گرفته خود را به سمت ذخار نصب و راه‌اندازی شده مجددا طراحی کند.

ذخیره سازی وب بصورت آرایشی به دو دلیل دیده شده است. اولا، یک ذخیره وب می تواند به طور اساسی زمان پاسخ به یک مشتر را کاهش دهد، مخصوصا اگر عرض پنهای باند بین مشتری و سرویس دهنده بسیار کمتر از عرض پهنای باند بین مشتری و ذخیره است. اگر یک اتصال سرعت بالا بین مشتری و ذخیره وجود دارد، که غالبا نیز وجود دارد، و اگر ذخیره هدف درخواست شده دارد، بنابراین ذخیره قادر خواهد بود هدف را به سرعت تحول مشتری دهد. دوما ، همانطور که در یک مثال بصور واضح خواهیم دید، ذخیره وب ها می تواند بطور اساس ی ترافیک بر روی دسترسی به اتصال سازمان به اینترنت را کاهش دهد بوسیله کاهش دادن ترافیک ، سازمان (برای مثال ، یک شرکت یا یک دانشگاه) لازم نیست پهنای باند را به سرعت تقویت کند، در نتیجه هزینه ها را کاهش می دهد. از این گذشته ذخایر وب ها می توانند به طور اساسی ترافیک وب را درون اینترنت بعنوان یک whole کاهش دهند در نتیجه کارایی تمام کاربردها را بهبود می بخشند.

برای بدست آوردن یک فهم بهتر از فواید ذخیره ها ، اجازه دهید یک مثال را که در متن شکل ۲,۱۱ است را بررسی کنیم. این کل دو شبکه را نشان می دهد – یک شبکه سازمانی و بقیه اینترنت عمومی. شبکه سازمانی یک LAN سرعت بالا است. یک مرورگر در شبکه سازمانی و یک مرورگر در اینترنت به وسیله یک اتصال 1.5Mbps با هم در ارتباط هستند. سرویس دهنده مبدا به اینترنت متصل شده اما در تمام دنیا قرار دارد. فرض کنید که متوسط سایز هدف ۱۰۰۰۰۰ بایت است و اینکه میزان متوسط درخواست جستجوگرهای سازمانی به سرویس دهنده مبدا ۱۵ درخواست در هر ثانیه است. فرض کنید که پیام های درخواست http بطور ناچیز کوچک هستند و از این رو هیچ ترافیکی در شبکه ها و در دسترسی به اتصال (از مرورگر سازمانی به مرورگر اینترنت) ایجاد نمی کنند. همچنین فرض کنید مقدار زمانی را صرف کرده از موقعی که مرورگر در طرف اینترنت دسترسی به اتصال در شکل ۲,۱۱ یک پیام http را ارسال کند(در قالب یک IP داده نگاشت) تا زمانی که پاسخ را دریافت کند.(عموما در قالب IP داده نگاشت های بسیار) ۲ ثانیه بصورت متوسط است. بصورت غیر رسمی به این آخرین تاخیر زمانی ذکر شده ، تاخیر اینترنتی می گوییم.

زمان کل پاسخگویی – که عبارت از زمانی که درخواست جستجوگر یک هدف تا دریافت هدف خود – برابر جمع تاخیر LAN، تا خیر دسترسی (عبارت است از، تاخیر بن دو مرورگر) و تاخیر اینترنتی اکنون می خواهیم یک محاسبات بسیار خام و خالص برای تخمین این تاخیر انجام دهیم. شدت ترافیک در LAN(بخش ۱,۶ دیده شود)

برابر است با :

$$(15\ \text{REQUESTS/SEC}) - (100\ \text{kbite/requests})/(10\text{mbps}) = 0.15$$

wWw.it98.ir

www.it98.ir

در صورتی که شدت ترافیک در دسترسی به اتصال (از مرورگر اینترنت تا مرورگر سازمانی) برابر است با :

$$(15\ \text{REQUESTS/SEC}) - (100\text{kbits/request})/(1.5\text{mbps}) = 1$$

یک شدت ترافیک ۰,۱۵ روی یک LAN بطور عمومی حداکثر ده ها میلی ثانیه تاخیر نتیجه می دهد : از این رو ما می توانیم از تأخیر LAN چشم پوشی کنیم.

به هر حال همانطور که در بخش ۱,۶ گفته شد هنگامی که دت ترافیک به ۱ نزدیک می شود (در حالت دسترسی به اتصال در شکل ۲,۱۱) ، تاخیر در یک اتصال بسیار بزرگ و با رشد نامحدود می شود. بدین ترتیب متوسط زمان پاسخ گویی به درخواست های مناب مناسب تقریبا چندین دقیقه می شود اگر بیشتر نشود که این برای کاربران سازمانی غیر قابل قبول است. به طور واضح باید اقداماتی انجام شود.

یک راه حل ممکن افزایش میزان دسترسی از 15Mbps به مثلا 100Mbps است. این کار شدت ترافیک در خطوط اتصال را تا ۰,۱۵ پایین می اورد، که می توان از تاخیر بین دو مرورگر ناچیز در نظر گرفته شود . در این حالت، کل زمان پاسخ گویی تقریبا ۲ ثانیه خواهد بود، که عبارتست از تاخیر اینترنتی. ولی این راه حل بدین معنی است که سازمان باید دسترسی به اتصال خود را از 15Mbps به 100Mbps تقویت کند که می تواند بسیار پر هزینه باشد.

اکنون راه حل تناوبی بدون تقویت کردن دسترسی به اتصال را در نظر میگیریم اما به جای ان یک ذخیره دب در شبکه سازمانی، نصب و راه اندازی می کنیم. این راه حل در شکل ۲,۱۲ نشان داده شده است. Hit rate- تجزیه در خواست هایی که مناسب هستند به وسیله یک ذخیره – به طور نمونه در عمل بین ۰,۲ تا ۰,۷ متغیر است.

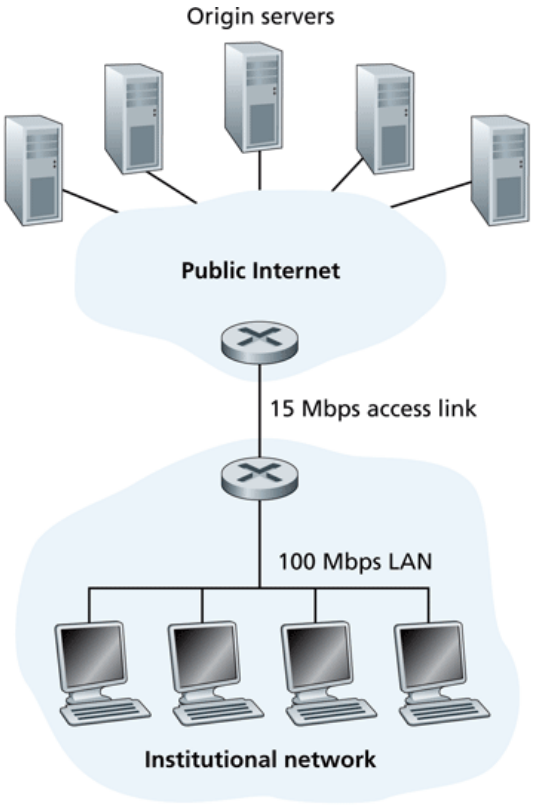


Figure 2.12 ♦ Bottleneck between an institutional network and the Internet

برای بهتر نشان دادن منظور، به طور مثال فرض کنید که ذخیره یک نرخ برخورد ۰,۴ را برای سازمان مهیا کند.به دلیل اینکه مشتری ها و ذخیره به یک LAN سرعت بالای یکسان متصل هستند،۴۰٪ از درخواست ها تقریبا به سرعت انجام خواهند شد،مثلا در حدود ۱۰ میلی ثانیه ،به وسیله ذخیره.با این حال ۶۰٪ باقیمانده در خواست ها همچنان نیاز به انجام دهی توسط سرویس دهنده های مبدا هستند.اما فقط با ارسال این ۶۰٪ از اهداف در خواست شده به وسیله اتصال دسترسی،شدت ترافیک روی خطوط

www.it98.ir

www.it98.ir

دسترسی از ۱,۰ به ۰,۶ کاهش خواهد یافت.به طور نمونه یک شدت ترافیک کمتر از ۰,۸ برای یک تاخیر کوچک مانند ده ها میلی ثانیه روی اتصال 15Mbps مطابقت دارد.این تاخیر در مقایسه با ۲ ثانیه تاخیر اینترنتی قابل چشم پوشی خواهد بود.با این ملاحظات، بنابراین متوسط تاخیر برابر است با :

0.4(0.01seconds) + 0.6(2.01seconds)

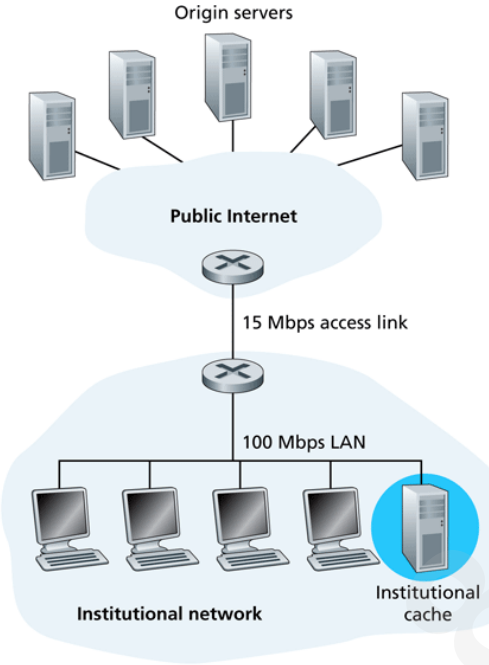


Figure 2.13 ♦ Adding a cache to the institutional network

که یک مقدار به طور جزیی بیشتر از ۱,۲ ثانیه است.بدین ترتیب این راه حل دوم حتی یک زمان پاسخ کمتر نسبت به راه حل اول فراهم میکند و همچنین احتیاجی نیست که سازمان اتصال به اینترنت خود را تقویت کند.البته کاری که سازمان باید یکند این نیست که یک وب ذخیره خریداری و نصب کند. اما هزینه ان بسیار کم است.

Get۲,۲,۶ شرطی

اگر چه ذخیره کردن میتواند زمان پاسخ گویی به در خواست کاربر را کاهش دهد، اما یک مشکل جدید ایجاد می کند .کپی یک هدف قرار گرفته در ذخیره ممکن است کهنه و قدیمی باشد .به عبارت دیگر هدف قرار داده شده در سرویس دهنده شبکه ممکن است از زمانی که کپی در مشتری ذخیره سازی شده باشد،تغیر کرده باشد.خوشبختانه HTTP یک ساختاری دارد که به یک ذخیره اجازه می دهد تا به اهداف خود رسیدگی کند که ایا به روز هستند یا خیر . این ساختارGET شرطی گفته می شود .یک پیام در خواست HTTP یک پیام GET شرطی گفته می شود اگر (۱) پیام در خواست از روش GET استفاده کند و (۲) اگر پیام در خواست شامل یک خط عنوان :If-Modified-Since باشد.

برای نشان دادن چگونگی کار GET شرطی، یک مثال گفته می شود.اولا، از طرف یک جستجوگر در خواست کننده ، یک پروکسی ذخیره یک پیام در خواست به سرویس دهنده وب ارسال می کند:

```
GET  / fruit / kiwi . gif  HTTP / 1.1
Host : www. exotiquecuisine.com
```

دوما، سرویس دهنده وب یک پیام پاسخ با هدف در خواست شده به ذخیره ارسال می کند:

```
HTTP / 1.1   200   ok
Data :Thu   7Jul 2007   15:39:29
```


wWw.it98.ir

www.it98.ir

Server : Apache /1.3.0 (Unix)
Last-Modified :Mon 4 Jul 2007 09:23:24
Content-Type : image / gif
(data data data data…)

cache هدف را به سمت جستجو گر در خواست کننده می فرستد اما هدف را در خود نیز ذخیره سازی می کند .مسئله مهم اینست که cache همچنین تاریخ اخرین تغییرات و اصلاحات هدف را نیز نگهداری می کند. سوما، یک هفته بعد یک جستجو گر دیگر همان هدف را به وسیله cache در خواست می کند،و هدف همچنان در cache وجود دارد. از زمانیکه احتمالا هدف در سرویس دهنده وب در هفته گذشته تغییر کرده است،cache یک بررسی به روز بودن به وسیله ارسال یک GET شرطی انجام می دهد. به ویژهcache متن زیر را ارسال می کند:

GET / fruit / kiwi.gif HTTP / 1.1
Host : www.exotiquecuisine.com
If – modified –since : wed 4 Jul 2007 09:23:24

می بینیم که در پاسخ به دریافت شرطی، همچنان سرویس دهنده وب یک پیام پاسخ ارسال می کند اما شامل هدف در خواست شده در پیام پاسخ نمی باشد. هدف در خواست شده شامل تنها پهنای باند را هدر می دهد و زمان پاسخ گویی به در خواست کاربر را افزایش می دهد، به ویژه اگر هدف بزرگ باشد.یاداوری می شود که این اخرین پیام پاسخ در خط وضعی 304 Not Modified قرار دارد، که به نهانگاه می گوید که می تواند ادامه دهد و کپی نهانگاهی خود هدف را به جستجو گر در خواست کننده ارسال کند.

۲,۳ انتقال فایلFTP

در یک جلسه FTP نمونه ،کاربر در مقابل یک میزبان نشسته است (میزبان محلی) و می خواهد فایل ها را به و یا از یک میزبان راه دور انتقال دهد. برای اینکه کاربر به حساب راه دور دسترسی یابد ،کاربر باید یک کد شناسایی کاربر و رمز عبور فراهم کند .بعد از فراهم کردن این اطلاعات برای کسب اجازه عبور کاربر می تواند فایل ها را از سیستم فایل محلی به سیستم فایل راه دور انتقال دهد و بر عکس. همانطور که در شکل۲,۱۴ نشان داده شده است،کاربر با FTPبه وسیله یک نماینده کاربر FTP ارتباط برقرار می کند.کاربر ابتدا نام میزبان ،میزبان راه دور را مهیا می کند ، پردازش مشتری FTP باعث می شود که در میزبان محلی یک اتصال TCPبا پردازش سرویس دهنده FTP در میزبان راه دور برقرار شود.کاربر سپس کد شناسایی کاربر و کلمه عبور را مهیا می کند که با ارسال از طریق اتصال TCP به عنوان بخشی از دستوراتFTP دریافت می کند.هنگامی که سرویس دهنده به کاربر اجازه می دهد،کاربر یک یا چند فایل کپی را از درون سیستم محلی به درون سیستم راه دور ذخیره میکند(یا بلعکس).

HTTPوFTP هر دو پروتکل انتقال فایل هستند و هر دو خیلی خصوصیات یکسانی دارند، برای مثال هر دو بر روی TCP اجرا می شوند. هر چند دو پروتکل لایه کاربرد تفاوت های مهمی با هم دارند.

www.it98.ir

www.it98.ir

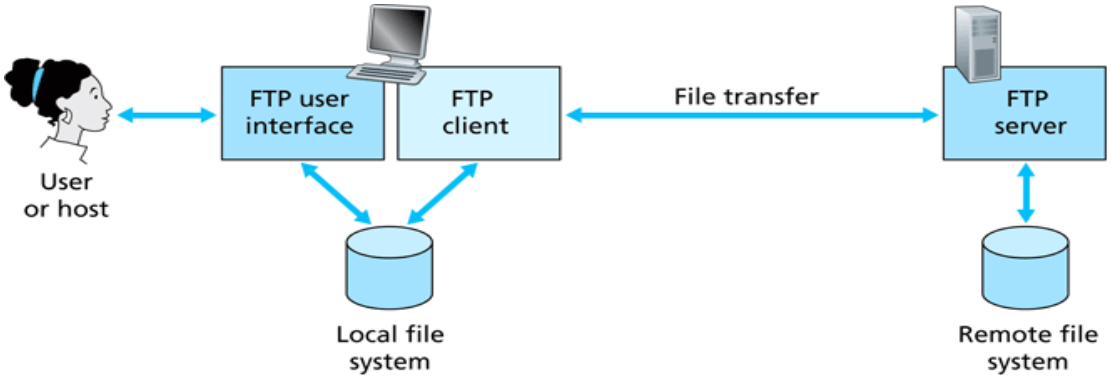


Figure 2.14 ♦ FTP moves files between local and remote file systems

برجسته ترین تفاوت اینست که FTPاز دو اتصال موازیTCP برای انتقال یک فایل استفاده می کند،یک ارتباط کنترلی (Control connection) و یک ارتباط داده ای(data connection).ارتباط کنترلی برای ارسال اطلاعات کنترلی بین دو میزبان استفاده می شود واطلاعاتی مانند کد شناسایی کاربرد ،کلمه عبوردستورات برای تغییر دیرکتوری راه دور،ودستورات برای فایل های "get" و"put".ارتباط داده ای در واقع برای ارسال یک فایل استفاده می شود.به دلیل اینکه یک اتصال کنترلی جدا استفاده می کنند ،گفته می شود که FTPاطلاعاتی کنترلی خود را خارج از باند(out-of-band) ارسال می کند.در بخش ۷ خواهیم دید که پروتکل RTSP که برای کنترل کردن انتقال رسانه پیوسته مانند فایل های صوتی و تصویری استفاده می شود ،همچنین اطلاعات کنترلی خود را خارج از باند نیز ارسال می کنند.HTTPخطوط عنوان پاسخ و در خواست را به درون اتصال TCPمشابه که خودش فایل انتقال یافته را حمل می کند ارسال می کند.به همین دلیل گفته می شودHTTP اطلاعات کنترلی خود را در باند (in-band) ارسال میکند. در قسمت بعد خواهیم دید که SMTP پروتکل اصلی برای برنامه الکترونیکی،همچنین اطلاعات کنترلی را در باند ارسال می کند. ارتباط کنترلی و داده ایFTPدر شکل ۲,۱۴ نشان داده شده است.

هنگامی که یک کاربر یک دوره FTP با یک میزبان راه دور را شروع می کند.طرف مشتری FTP (کاربر) ابتدا یک ارتباط FTP کنترلی با طرف سرویس دهنده (میزبان راه دور) در پورت شماره ۲۱ سرویس دهنده بر قرار کند. طرف مشتری FTP همچنین به وسیله ارتباط کنترلی دستورات را برای تغییر دایر کتوری راه دور ارسال می کند. هنگامی که طرف سرویس دهنده بوسیله ارتباط کنترلی یک دستور برای انتقال یک فایل (از یا به میزبان راه دور) دریافت کرد.طرف سرویس دهنده یک ارتباط داده ای TCP با طرف مشتری برقرار می کند. اگر، در طول جلسه یکسان، کاربر بخواهد یک فایل دیگر را انقال دهد ، FTP ارتباط داده ای دیگری را باز می کند. زیرا FTP دقیقا یک فایل بوسیله ارتباط داده ای می فرستد و سپس ارتباط داده ای را می بندد. بدین ترتیب با FTP ارتباط کنترلی در کل مدت جلسه کاربر باز نگر می دارد. اما یک ارتباط داده ای جدید برا هر فایل انتقال یافته در محدوده جلسه ایجاد می کند (بدین معنی که ، ارتباط داده ناپیدار است).

کل یک جلسه سرویس دهنده FTP باید حالت و وضعیت مربوط به کاربر را نگهدار کند. به ط.ر عمودی، سرویس دهنده باید به ارتباط کنترلی را با یک صاحب کاربر ویژه کمک کند

www.it98.ir

wWw.it98.ir

www.it98.ir

Figure 2.15 ♦ Control and data connections

، و سرویس دهنده باید مسیر دایرکتوری جاری کاربر را بعنوان user wonders در مورد tree دایرکتوری راه دور نگهداری کند. نگه داشتن مسیر این حالت اطلاعات برای هرجلسه (session) کاربر در جریان عمدتا متحمل جمع از دوره هایی می شود که FTP می تواند همزمان نگهداری کند، از طرف دیگر HTTP بدون وضعیت و حالت است – یعنی اینکه نباید مسیر حالت هیچ کاربری را نگهداری کنند.

۲,۳,۱- فرامین FTP و پاسخ‌ها

این قسمت را با خلاصه ای از فرامین رایج FTP به پایان می بریم. فرامین، از مشتری به سرویس دهنده و پاسخ ها، از سرویس دهنده به مشتری از میان ارتباط کنترلی در قالب کد اسکی ۷ بیتی ارسال می وند. بدین ترتیب مانند فرامین HTTP ، فرامینFTP بوسیله انسان ها قابل خواندن است. هر دستور شامل ۴ خصوصیت حروف بزرگ کد اسکی می باشد. بعضی از فرامین بسیار رایج در پایین آمده است.

- ♦ USER USERNAME : برای ارسال کد شناسایی کاربر به سرویس دهنده استفاده می شوند.
- ♦ PASS Password : برای ارسال کلمه عبور و کاربر به سرویس دهنده استفاده می شود.
- ♦ Reterfilename : برای بازیابی (عبارتتست از get) یک فایل از دایرکتوری جاری میزبان راه دور استفاده می شود. راه اندازی میزبان راه دور برای آغاز یک ارتباط داده ای و برای ارسال فایل درخواست شده بوسیله ارتباط داده ای
- ♦ STOR file name : برای ذخیره سازی (عبارت است ازput) یک فایل درون دایرکتوری جاری میزبان راه دور استفاده می شود.
- ♦ List : برای درخواست به سرویس دهنده برای برگشت دادن یک لیست از همه فایل ها در دایرکتوری جاری استفاده می شود. لیست فایل ها بوسیله یک (جدیدی و ناپایدار) ارتباط داده ای بجای ارتباط tcp کنترلی ارسال می شود.

در این جا به طور نمونه یک تطابق یک به یک میان دستور که کاربر صادر می کند و دستور FTP ارسال شده از میان ارتباط کنترلی وجود دارد. هر دستور با یک پاسخ همراه است، از سرویس دهنده به مشتری ارسال شده . پاسخ ها اعداد سه رقمی هستند. با یک پیام اختیاری همراه شده با عدد این از لحاظ ساختار شبیه به کد وضعیت و عبارت درون خط وضعیت پیام پاسخ HTTP است.

www.it98.ir

مخترعین HTTP عمدا این شباهت را درون پیام های پاسخ HTTP نیز شامل شده اند. بعضی از پاسخ های نمونه، همراه با پیام محتمل شان، در ادامه آمده است :

- 331 username ok, password required
- 125 Data connection already open, transfer starting
- 425 cant open data connection
- 452 error writing file

خوانندگانی که مایلند در مورد دیگر فرامین FAT و پاسخ بدانند به خواندن RFC959 دعوت می‌کنیم.

۲,۴. نامه الکترونیکی در اینترنت

نامه الکترونیکی (ایمیل) از ابتدای اینترنت وجود داشته است. این بیشترین کاربرد رایج اینترنت زمانی که در اول کار خود قرار داشت، بود. و در مرور زمان و سال ها بسیار بسیار ماهرانه تر و پیچیده تر و قدرتمندتر شد و به این روند تکمیلی خود ادامه داد. این یکی از مهمترین خواست کاربردهای اینترنت در تاریخ است.

به عنوان یک نامه پستی معمولی، ایمیل یک ارتباط آسنکرون متوسط است. انسان ها و پیام تا را وقتی که برایشان مناسب است ارسال می کنند و می خواسته- بدون اینکه با برنامه های دیگران بخواهند هماهنگ کنند. در مقابل نامه پستی ، نامه الکترونیکی سریع ات، توزیع آن آسان است، و ارزان است. ایمیل های مدرن بسیار ویز"ی های قدرتمندی دارد. با استفاده از لیست های پستی ، پیام های ایمیل و Spam می توان به هزاران گیرنده در یک زمان ارسال شود. پیام های ایمیل های امروزی معمولا شامل ضمیمه ها، فوق ارتباط ها، متن با قالب بندی HTML و عکس ها هستند. بطور کلی ، ایمیل یک متن مرکزی بوده است، اما همچنین می تواند بعنوان یک سکو برای پیغام دهی صوتی و تصویری آسنکرون استفاده شود.

در این بخش ما پروتکل های لایه کاربر را که در مطلب ایمیل اینترنتی هستند را بررسی می کنیم. اما قبل از اینکه بخواهیم دارد بحث گسترده این پروتکل ها شویم، اجازه دهید یک نگاه سطح بالا به سیستم نامه اینترنتی و اجزا کلیدش بیاندازیم.

شکل ۲,۱۵ یک نگرش سطح بالا را از سیستم پستی ارائه می دهد. از این دیاگرام می فهمیم که این سیستم سه جزء دارد: نمایندگان کاربر، سرویس دهنده های پستیو پروتکل انتقال نامه ساده. اکنون هر کدام از این اجا را در قالب متن یک ارسال کننده ،آلیس، که یک پیام ایمیل را به یک گیرنده ارسال می کند، یعنی باب. نمایندگان کاربر به کاربران اجازه می دهند که پیام ها را بخوانند، پاسخ دهند ، ارسال کنند و ذخیره کنند(نمایندگان کاربر برای نامه های الکترونیکی گاهی اوقات خوانندگان نامه گفته می شوند، هر چند ما به طور کلی از این اصطلاح در این کتاب اجتناب می کنیم).

هنگامی که آلیس تالیف پیام خود را به پایان رسانده نماینده کاربر پیام را به سوریس دهنده پتی اش ارسال می کند، جایی که پیام در صف خارج کننده پیام سرویس دهنده نامه قرار می گیرد. هنگامی که باب می خواهد یک پیام را بخواند ، نماینده کاربر او پیام را از صنودوق پستی اس در سرویس دهنده پستی اش بازیابی می کند. در اواخر ۱۹۹۰ رابط کاربر گرافیکی (GUI) نمایندگان کاربر بسیار مطرح شد، که به کاربران اجازه می داد که پیام های چند رسانه ایی را بیشتر تالیف کنند.

به تازگی ، ادورا NETSCAPES MESSENGER ,MICROSOFT OUT LOOK در میان مطرح ترین نمایندگان کاربر GUI برای ایمیل هستند. در آنجا همچنین بسیاری کاربر امیل های مبتنی بر متن درون حوزه عمومی یک واسطه می گذراند که شامل ELM, PINE, EMAIL می شود.

www.it98.ir

wWw.it98.ir

www.it98.ir

نامه از هسته زیر ساختار ایمیل سرویس دهی می شود. هر گیرنده، مانند باب، یک صندوق پستی (MAIL BOX) در یکی از سرویس دهنده های پستی دارد. صندوق پستی باب پیام هایی که به او ارسال شده است اداره و نگهداری می کند. یک پیام نمونه سفر خود را از درون نماینده کاربر ارسال کننده آغاز می کند، حرکت به سمت سرویس دهنده پستی ارسال کننده و حرکت به سمت سرویس دهنده گیرنده ، جایی که در صندوق پستی گیرنده ذخیره می شود. هنگامی که باب می خواهد به پیام های درون صندوق پستی خود دسترسی یابد. سرویس دهنده پستی شامل مجوز عبور صندوق پستی خود می شود(با نام کاربری و کلمه عبور) . سرویس دهنده آلیس نیز همچنین باید به خوابی ها در سرویس دهنده باب نیز رسیدگی کند. اگر سرویس دهنده آلیس نتواند نامه را به سرویس دهنده باب تحویل دهد، سرویس دهنده آلیس نامه را در یک ردیف پیام نگهداری می کند و تلاش می کند که بعدا پیام را انتقال دهد. هر تلاش مجدد معمولا هر ۳۰ دقیقه یا بیشتر انجام می شود، اگر بعد از چند روز نیز موفق به انجام این کار شد، سرویس دهنده پیام را پاک می کند و با یک پیام ایمیل به فرستنده (آلیس) اطلاع می دهد.

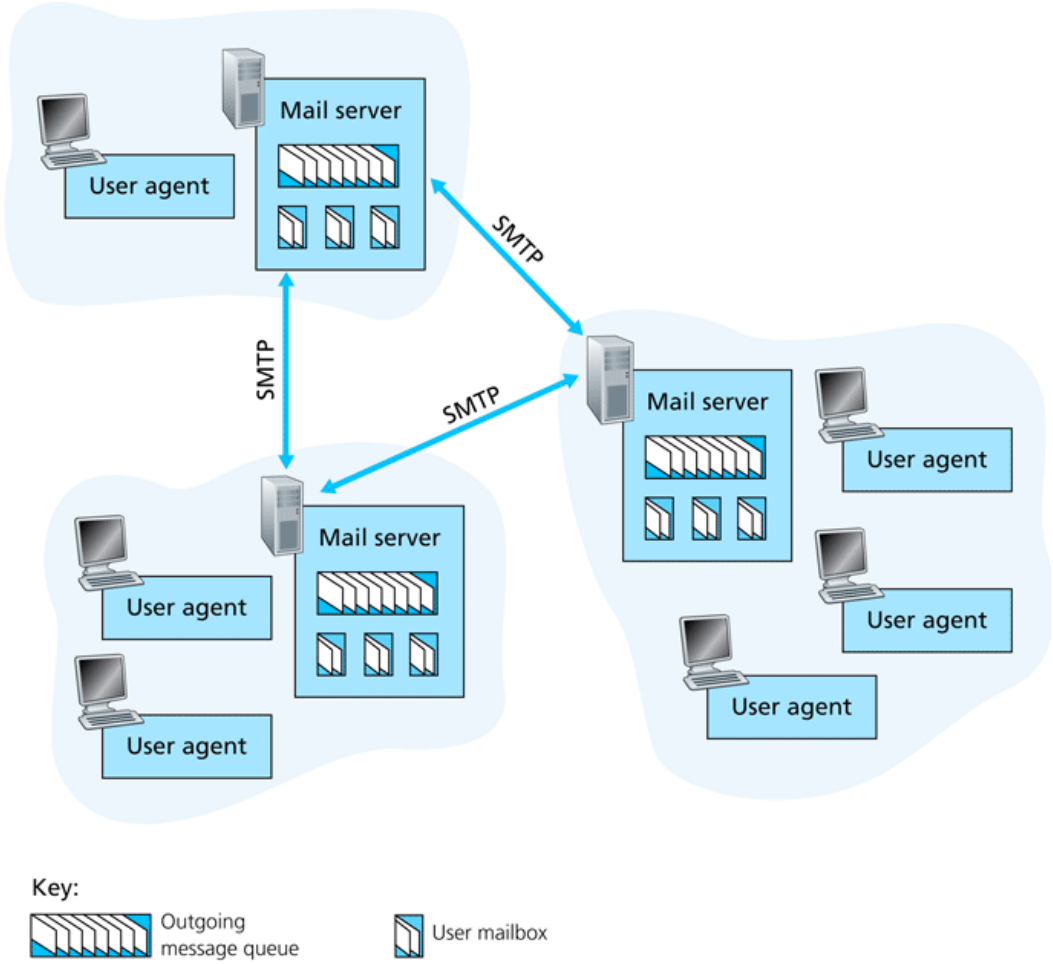


Figure 2.16 ♦ A high-level view of the Internet e-mail system

SMTP مدیر پروتکل لایه کاربر برای نامه الکترونیکی اینترنتی است. SMTP یک خدمات انتقال داده فایل احتمال TCP را برای انتقال نامه از سرویس دهنده پستی فرستنده به سرویس دهنده گیرنده استفاده می کند. بسیاری از پروتکل های لایه کاربرد، SMTP دو طرف دارد: یک طرف مشتری،که بر روی سرویس دهنده پستی فرستنده اجرا می شود و یک طرف سرویس دهنده که بر روی سرویس دهنده پستی گیرنده اجرا می شود. هر دو طرف مشتری و سرویس دهنده ، بعنوان یک SMTP مشتری عمل می

www.it98.ir

www.it98.ir

کنند. هنگامی که یک سرویس دهنده پستی نامه را از دیگر سرویس دهنده های پستی دریافت کرد مانند یک سرویس دهنده SMTP عمل می کند.

۲,۴,۱. SMTP

SMTP در RFC2821 معرفی شده است. در قلب نامه الکترونیکی اینترنتی است. همانطور که در بالا اشاره شد، SMTP پیام ها را از سرویس دهنده های پستی فرستنده به سرویس دهنده های پستی گیرنده انتقال می دهد.SMTP بسیار قدیمی تر از HTTP است. (SMTP RFC اصلی به تاریخ 1982 بر می گردد و SMTP به قبل از آن). اگرچه SMTP خواص عالی بسیاری دارد، با توجه به اینکه حضورش در همه جا در اینترنت ثابت شده ، با این حال ای یک فن آوری ارثی است که خصوصیات قدیمی بخصوصی را دارد. برای مثال SMPT بدنه کلیه پیام های پست شده را (نه فقط عناوین) به کد اسکی ۷ بیتی ساده محدود می کند. این محدودیت در اوایل دهه ۱۹۸۰ زمانی احساس شد که ظرفیت انتقالات کمتر شد و هیچ کس ضمیمه های بزرگ و عکس بزرگ و فایل های صوتی، تصویری بزرگ را پست نمی کرد. اما امروزه در دوران چند رسانه ای، محدودیت ۷ بیتی یک معضل است.- این لازم به این است که داده چند رسانه ای بانیری به کداسکی قبل از اینکه بوسیلهSMTP ارسال شود رمزی شود و این همچنین مستلزم این است که پیام اسکی متناظر به باینری بعد از انتقال SMTP برگردد. بیاد آورید از بخش ۲۰۲ که Http مستلزم داده چند رسانه ای برای به رمز کردن کد اسکی قبل از انتقال نبود.

برای نشان دادن عملیات پایه ای SMTP یک سناریو را قدم به قدم بررسی کنیم. فرض کنید آلیس بخواهد که یک پیام اسکی ساده را به باب ارسال کند.

۱. آلیس نماینده کاربر خود را برای ایمیل احضار می کند، آدرس باب را آماده می کند (برای مثال bob@someschool.edu،) پیام را می نویسد، نماینده کاربر را برای ارسال پیام مطلع می کند.
۲. نماینده کاربر آلیس پیام را به سرویس دهنده پستی اش ارسال می کند، جایی که پیام در یک ردیف از پیام قرار می گیرد.
۳. طرف مشتریSMTP ، در سرویس دهنده پستی الیس اجرا می شود. پیام را در ردیف پیام می بینید. یک اتصال TCP به یک سرویس دهنده SMTP باز می کند، در سرویس دهنده پستی باب اجرا می شود.
۴. بعد از مقداری ارتباط ابتدایی SMTP مشتری SMTP پیام آلیس را به درون اتصال TCP ارسال می کند.
۵. در سرویس دهنده پستی باب، طرف سرویس دهنده SMTP پیام را دریافت می کند. سرویس دهنده پستی باب سپس پیام را در صندوق پستی باب قرار می دهد.
۶. باب نماینده کاربر خود را برای خواندن پیام در وقت مناسب احضار می کند.

سناریو بصورت خلاصه در شکل ۲,۱۷ آمده است.

این مهم است گفته شود که SMTP بطور عادی از سرویس دهنده های پستی میانجی برای ارسال نامه استفاده نمی کنند. حتی زمانی که ۲ سرویس دهنده پستی در دو نقطه کاملا مخالف دنیا قرار داشته باشند. اگر سرویس دهنده آلیس در هنگ کنگ و سرویس دهنده باب در سنت لوئیس باشند اتصال TCP یک اتصال مستقیم بین سرویس دهنده های هنگ کنگ و سنت لوئیس است. بطور کلی، اگر سرویس دهنده پستی باب خراب شود، پیام در سرویس دهنده پستی آلیس باقی می ماند و برای تلاش جدید منتظر می ماند – پیام در چند سرویس دهنده پستی میانجی قرار نخواهد گرفت.

www.it98.ir

wWw.it98.ir

www.it98.ir

بگذارید اکنون نگاهی دقیق کنیم که چگونه SMTP یک پیام را از یک سرویس دهنده پستی فرستاده به یک سرویس دهنده پستی گیرنده انتقال میدهد. خواهیم دید که پروتکل SMTP شباهت های بسیاری با پروتکل هایی که برای اثر متقابل برخورد رودر روی انسلان ها استفاده می شود، دارد.

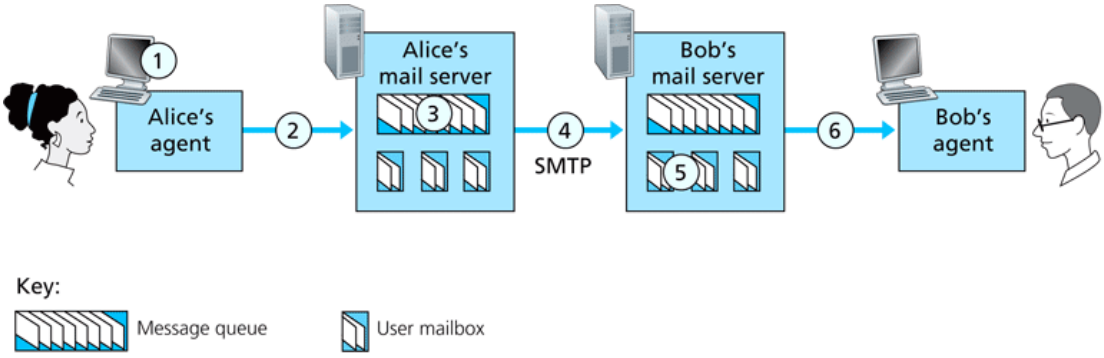


Figure 2.17 ♦ Alice sends a message to Bob

اولا،SMTP مشتری (روی میزبان سرویس دهنده پستی فرستنده اجرا می شود) یک اتصال TCP روی پورت ۲۵ به SMTP سرویس دهنده برقرار می کند. هنگامی که این اتصال برقرار شد، سرویس دهنده و مشتری برخی توافق نمایی لایه کاربرد را انجام می دهند- مانند انسان ها که اغلب خود را قبل از انتقال اطلاعات از یکی به دیگری معرفی می کنند. سرویس دهنده و مشتری SMTP خود را قبل از انتقال اطلاعات معرفی می کنند. در طول این حوزه توافق نمایی SMTP مشتری SMTP آدرس ایمیل فرستنده (شخصی که پیام را منتشر می کند) و آدرس ایمیل گیرنده را تعیین می کند. هنگامی که SMTP مشتری و سرویس دهنده خودشان را به یکدیگر معرفی کردند مشتری پیام را ارسال می کند. SMTP می تواند به روی قابل اعتماد بودن سرویس انتقال داده TCP برای دادن پیام به سرویس دهنده بدون خطا حساب کند. مشتری پس از این پردازش را بوسیله اتصال TCP یکسان تکرار می کند اگر پیام های دیگری برای ارسال به سرویس دهنده داشته باشد و در غیر این صورت به TCP اطلاع می دهد ارتباط را ببندد.

اجازه دهید نگاه دیگر به یک مثال رونوشت پیام های مبادله شده بین یک مشتری SMPT (C) و یک سرویس دهنده SMPT (S) بیاندازیم. نام میزبان مشتری CREPES.FR است و نام میزبان سرویس دهنده hamburyer.edu است. خطوط کد اسکی با C: شروع شده دقیقا خط هایی هستند که سرویس دهنده به درون سوکت TCP خود ارسال می کند. رونوشت بعدی به محض اینکه ارتباط TCP برقرار شد شروع می شود.

S : 220 hamburger. Edu
C :helo crepes.fr
S : 250 hello crepes. Fr , pleased to meet you
C : mail from <alice @ crepes.fr>
S : 250 alice @ crepes. Fr…. Sender ok
C :rcpt to : <@ hamburger. edu>
S : 250 bob @ hamburger. Edu….recipient ok
C : DATA
S : 354 enter mail , end with "0" ona line by itself
C : How about pickles?
C : 0
S : 250 message accepted for delivery

www.it98.ir

www.it98.ir

C : QUIT
S : 221 ham burger. Edu closing connection

در مثال بالا یک پیام (Do you like ketchup? , How about pickles) از سرویس دهنده پستی crepes.fr به سرویس دهنده پستی hamburger.edu ارسال می کند. در قسمتی از مکالمه ، مشتری ۵ فرمان را بکار برده : HELO (مخفف برای HELLO) ، MAIL, FROM , RCPT , DATA, QUIT . این فرامین خود توصیفی هستند. مشتری همچنین یک خط شامل یک نقطه تنها ارسال می کند که پایان پیام را به سرویس دهنده نشان می دهد. (در جارگن(زبان فنی) کد اسکی، هر پیام با CRLF.CRLF پایان می یابد، جایی که CR و LF به ترتیب برای Carriage return و line feed می ایستند). سرویس دهنده پاسخ ها را به هر دستور صادر می کند، با هر پاسخ یک کد پاسخ و مقداری توضیح به زبان انگلیسی (اختیاری) داریم. اشاره کردیم که smtp اتصالات پایدار استفا ده می کند. اگر سرویس دهنده پستی فرستنده چندین پیام برای ارسال به سرویس دهنده پستی گیرنده یکسان دارد، همه پیام ها را می تواند بوسیله ارتباط TCP یکسان بفرستد. برای هر پیام،مشتری پردازش را با یک mail from. Crepes جدید شروع می کند، آخر پیام را با یک نقطه تنها مشخص می کند و quit را فقط بعد از همه پیام های ارسال شده ، صادر می کند.

بسیار پیشنهاد شده که شده شما از Telnet برای انجام دادن یک مکالمه مستقیم با سرویس دهنده smtp استفاده کنید برای این کار بنویسید :

Tel net server name 25

جایی کهserver name نام یک سرویس دهنده پستی محلی است. هنگامی که این کار را می کنید شما بصورت ساده یک اتصال TCP بین میزبان محلی تان و سرویس دهنده پستی برقرار می کنید. بعد از نوشتن این خط شما باید به سرعت پاسخ 220 را از سرویس دهنده دریافت کنید. پس فرامین MAIL FROM ,HELO, SMTP,QUIT, CRLF,CRLF,DATA,RCPTTO را در زمان مناسب صادر کنید. همچنین بسیار پیشنهاد می شود که Programming Assignment2 (تکلیف برنامه نویسی ۲) در پایان این فصل انجام دهید. در آن تکلیف، شما یک نماینده کاربر ساده را که طرف مشتری smtp را پیاده سازی می کند، خواهید ساخت. همچنین به شما اجازه ارسال یک پیام ایمیل به یک گیرنده اختیاری بوسیله یک سرویس دهنده پستی خواهد داد.

۲,۴,۲ مقایسه با HTTP

اجازه دهید اکنون به طور خلاصه SMTP را با HTTP مقایسه کنیم. هر دو پروتکل برای انتقال فایل ها در یک میزبان به دیگری استفاده می شوند :

HTTP فایل ها را از یک سرویس دهنده وب به یک سرویس دهنده شبکه (معمولا یک جستجوگر) انتقال می دهد و SMTP فایل ها را (منظور همان پیام های ایمیل است) را ازیک سرویس دهنده پستی به سرویس دهنده پستی دیگر انتقال می دهد. در هنگام انتقال دادن فایل ها ، هر دو SMTP و HTTP پایدار از اتصالات پایدار استفاده می کنند. بدین ترتیب هر دو پروتکل خصوصیات های یکسانی دارند. هر چند تفاوت های مهمی نیز دارند. اولا HTTP اساسا یک پروتکل کشتی است. یک نفر اطلاعات را روی یک سرویس دهنده وب بارگذاری می کند و کاربران از HTTP برای بیرون کشیدن اطلاعات از سرویس دهنده در وقت مناسب خود استفاده می کنند. بطور نمونه، ارتباط TCP بوسیله ماشینی آغاز شده است که می خواهد فایل را دریافت کند – از طرف دیگر، SMTP اساسا یک پروتکل فشار است. سرویس دهنده پستی سرویس دهنده فایل را برای سرویس دهنده پستی گیرنده حرکت می دهد. بطور نمونه ارتباط TCP بوسیله یک ماشینی که می خواهد فایل را بفرستد آغاز شده است.

wWw.it98.ir

دومین تفاوت ، که پیش از این اشاره کردیم ، این است که SMTP برای هر پیام مستلزم این است که شامل پیکره هر پیام می شود، در قالب کد اسکی ۷ بیتی باشد. اگر پیام محتوای علائم باشد که در کد اسکی ۷ بیتی وجود ندارد (مانند، علائم فرانسوی با لهجه) یا محتوای داده باینری (مانند یک فایل عکس) باشد، در نتیجه پیام باید به کد اسکی ۷ بیتی رمز دهی شود و تفسیر کند. داده HTTP این محدودیت را شامل نمی شود.

سومین تفاوت مهم چگونگی ترتیب دادن یک فایل حاوی متن و عکی (همراه با دیگر انواع رسانه ای ممکن دیگر) است. همانطور که در بخش ۲,۲ یاد گرفتیم، HTTP هر هدف را در پیام پاسخ HTTP خودش بصورت بسته در می آورد و نگهداری می کند. پست الکترونیکی که در ادامه بیشتر بحث می‌کنیم، تمام اهداف پیام را در یک پیام جای می‌دهد.

۲,۴,۳ قالب‌های پیام پستی و MIME

هنگامی که آلیس یک نامه پستی معمولی را برای باب نوشت ، ممکن است تمام اطلاعات عنوانی جانبی را در بالای نامه آورده باشد، مانند آدرس باب ، آدرس خودش ، و تاریخ. بطور مشابه، هنگامی که یک پیام ایمیل از یک شخص به دیگری ارسال شده است ، یک عنوان شامل اطلاعات جانبی که مقدمه بدنه خود پیام شد. این اطلاعات جانبی به صورت متوالی و پشت سرهم در خطوط عنوان جای می گرفتند. که در RFC822 تعریف شده اند. خطوط عنوان و بدنه پیام بوسیله یک خط خالی جدا شده بودند(یعنی به وسیله CRLF) . RFC822 قالب دقیق را برای خطوط عنوان نامه بخوبی ترجمه و تفسیر معنایی شان مشخص می کند.HTTP هر عنوان که حاوی متن قابل خواندنی می شود، شامل یک کلمه کلیدی همراهی شده بوسیله یک COLON همراهی شده بوسیله یک ارزش می شود. بعضی از کلمه های کلیدی التزامی و بعضی اختیاری هستند. هرعنوان باید یک خط عنوان From: و یک خط عنوان TO: را داشته باشد و یک عنوان ممکن است شامل یک خط عنوان subject: بخوبی دیگر خطوط عنوان اختیاری شود. مهم است که خاطرنشان کنیم که این خطوط عنوان از فرامین smtp که در بخش ۲,۴,۱ متفاوت است. (حتی اگر شامل بعضی از کلمات رایج مانند to from باشد) فرامین در آن بخش قسمتی از پروتکل توافق نمایی smtp بود : خطوط عنوان بررسی شده در این بخش قسمتی از خود پیام پستی هستند.

www.it98.ir

www.it98.ir

From :alice @crepes. Fr

To : bob @hamburger. Edu
Subject : searching for the meaming of life

بعد از عنوان پیام و یک خط خالی می آید و سپس پیکره پیام (دراسکی) می آید شما باید از elnet برای ارسال یک پیام به یک سرویس دهنده پستی که محتوی تعدادی خطوط عنوان است استفاده کنید، شامل خط عنوان :subject برای این کار telnet server name 25 را صادر کنید.

گسترش mime برای داده بدون کد اسکی

هنگامی که عنوان پیام در RFC822 تعریف شدن برای ارسال کد اسکی مناسب هستند. آنها برای پیام های چندرسانه ای به قدرت کافی قدرتمند نیستند. (برای مثال، پیام های باعکس ، صوتی ، تصویری) یا برای انتقال قالب های کد غیر اسکی (برای مثال علائم استفاده شده بوسیله زبان های غیر از انگلیسی). برای ارسال محتوایی به جز کد اسکی ، نماینده کاربر فرستنده با عناوین اضافی دیگری را در پیام شامل شود. این عناوین اضافی در RFC2046, RFC2045 تعریف شده اند. پست الکترونیکی چند منظوره گسترش یافته (MIME) در RFC822 تعریف شده است.

دو عنوان کلیدی MII برای پشتیبانی فایل های چند رسانه ای عنوان CONTENT-TYPE: و عنوان content- ransfer- encodtty: هستند. عنوان content- type به نماینده کاربر گیرنده اجازه می دهد تا یک عمل مناسب روی پیام انجام دهد. برای مثال، با نشان دادن اینکه بدنه پیام شامل یک عکس JPEG است، نماینده کاربر گیرنده می تواند بدنه پیام را به یک JPEG کم حجم عادی هدایت کند. برای فهمیدن احتیاج عنوان content- transfer – enecodry بیاد آورید که پیام های کد غیر اسکی باید به یک قالب اسکی رمزدهی شوند که باعث سردرگمی smtp نشوند. عنوان content – transfer- enccodry کاربر کاربر گیرنده را خبردار می کند که بدنه پیام به کداسکی رمزدهی شده است و نوع رمز دهی استفاده شده را تععین می کند. بدین ترتیب هنگامی که یک نماینده کاربر یک پیام با عنوان را دریافت می کند. ابتدا از عنوان content – transfer- encodry: برای تبدیل بدنه پیام به قالب غیر اسکی اصلی خود استفاده می کند و سپس از عنوان content- type: برای معلوم کردن اینکه چه عملی باید روی بدنه پیام انجام شود، استفاده می کند.

نگاهی به یک مثال واقعی و مملوس می اندازیم. فرض کنید آلیس می خواهد یک عکس JPEG را برای باب ارسال کند. برای اینکار ، آلیس نماینده کاربر خود را برای ایمیل فراخوانی می کند. آدرس ایمیل باب را مشخص می کند و موضوع نامه را تعیین می کند و عکس JPEG را به درون بدنه پیام نامه وارد می کند. (بسته به کارگیری آلیس از نماینده کاربر او ممکن است عکس را به صورت ضمیمه در پیام وارد کند). هنگامی که آلیس نوشتن پیام خود را تمام کرده بر روی ارسال (SEND) کلیک می کند. نماینده کاربرد آلیس سپس یک پیام MIME را منتشر می کند. که ممکن است چیزی شبیه این باشد :

From :alice @crepes. Fr
To : bob @hamburger. Edu
Subject : picture of yummy crepe
MIME – Version : 1.0
Content – transfer- encoding : base 64
Content – type : image / jpeg
(base 64 encoded data ……
……
…….base 64 encoded data)

wWw.it98.ir

www.it98.ir

از پیام MIME بالا مشاهده می کنیم که نماینده کاربر آلیس عکس JPEG را با استفاده از رمزگذاریBASE64 ، رمز دهی کرده است. این یکی از چند روش های رمز گذاری استاندارد شده در MIME برای تبدیل به یک کد اسکی ۷ بیتی قابل قبول برای شبکه است. دیگر روش کد گذاری رایج :content- transfer- encoddy است که بطور کلی برای تبدیل یک پیام اسکی بیتی (ممکن است شامل علائم غیرانگلیسی باشد) به اسکی ۷ بیتی است. هنگامی که باب نامه خود را با نماینده کاربرش می خواند، نماینده کاربر و روی این پیام MIME مشابه عمل می کند.

هنگامی که نماینده کاربر باب خط عنوان content- transfer- encody: base64 را مشاهده می کنند، به رمز دهی بدنه پیام با base64- enconded ادامه می دهد. پیام همچنین شامل یک خط عنوان content- transfer-encondy:base64 را مشاهده می کنند، به رمز دهی بدنه پیام بابase64- encoded ادامه می دهد. پیام همچنین شامل یک خط عنوان content- type : image/jpeg می شود. این به نماینده کاربر باب نشان می دهد که بدنه پیام باید jpeg کم حجم باشد. در پایان، مسج شامل عنوان MIME- VERSION است که البته نسخه MIME نشان می دهد که در حال استفاده و کار است. گفتنی است که پیام در هر صورت قالب استانداردrfc822/smtp را همراه دارد. بطور نمونه، بعد از عنوان نامه یک خط خالی و سپس بدنه پیام است.

پیام دریافت شده

اگر به دیگر نوع خطوط عنوان که بوسیله سرویس دهنده گیرنده SMTP وارد شده اند اشاره می کنیم کم کاری کرده ایم. سرویس دهنده گیرنده، بر اساس دریافت یک پیام با RFC822 و خطوط عنوان MIME یک خط عنوان : Received را بالای پیام اضافه می کند : این خط عنوان نام سرویس دهنده SMTP را که پیام را از.... ارسال رده یا نام سرویس دهنده SMTP که پیام را بوسیله.... دریافت کرده و زمانی را که سرویس دهنده گیرنده پیام را دریافت کرده مشخص می کند. بدین ترتیب پیام بوسیله کاربر مقصد مانند شکل زیر دیده می‌شود :

Received : from crepes – fr by hamburger. Edu, 12 oct 98 15:27:39
GMT From :alice @crepes. Fr
To : bob @hamburger. Edu
Subject : picture of yummy crepe
MIME – Version : 1.0
Content – transfer- encoding : base 64
Content – type : image / jpeg
base 64 encoded data ……
……
…….base

تقریبا تمامی کسانی که نامه الکترونیکی استفاده کرده باشند خط عنوان :RECEIVED (همراه با دیگر خطوط عنوان)جلوی پیام های ایمیل دیده اند. (این خط معمولا بطور مستقیم بر روی صفحه مانیتور یا هنگامی که پیام به یک چاپگر ارسال شده است ، دیده می شود.). ممکن است شما متوجه شده باشید که یک پیام تنها گاهی اوقات چندین خط عنوان :Received و یک خط عنوان پیچیده تر Return- path دارد. این بخاطر این است که که یک پیام ممکن است به بیش از یک سرویس دهنده SMTP در مسیر بین فرستنده و گیرنده فرستاده شود. برای مثال، اگر باب سرویس دهنده ایمیل خود را hamburger.edu گفته باشد که تمام پیام هایش را به sushi.jp ارسال کند، سپس خواندن پیام بوسیله نماینده کاربر باب با چیزی شبیه این شروع خواهد شد :

Received : from hambureger – edu by sush . jp , 3jul 01 15 :30:01 GMT

www.it98.ir

www.it98.ir

Received : from crepes. Fr by hamburger.Edu, 3jul 01 15:17:39 GMT

این خطوط عنوان نماینده کاربر گیرنده را هنگامی که SMTP پیام از سرویس دهنده پستی آلیس به سرویس دهنده پستی باب تحویل دهد، پیام در صندوق پستی باب قرار خواهد گرفت.

۲,۴,۴ پروتکل دسترسی Mail

هنگامی‌که SMTP پیام را از سرویس‌دهنده ایمیل آلیس به سرویس‌دهنده ایمیل باب دریافت می‌کند، پیام در صندوق پستی باب قرار دارد. در تمام این بحث بطر ضمنی فرض کرده ایم که باب نامه اش را بوسیله واردشدن به میزبان سرویس دهنده و سپس اجرای یک نامه خوان که در میزبان اجرا می شود می خواند. تا اوایل سال ۱۹۹۰ این روش استاندارد اینکار بود. اما امروزه، دسترسی به نامه از معماری مشتری – سرویس دهنده استفاده می کند- بطور نمونه کاربر ایمیل را با یک مشتری می خواند که در سیستم کاربر اجرا می شود(end system)، برای مثال ، روی یک شرکت کامپیوتری، یک لب تاپ و یا یک PDA . بوسیله اجرای یک نامه مشتری در یک کامپیوتر محلی، کاربران از ویژگی های بسیار قدرتمند لذت می برند،که شامل قابلیت تماشای پیام های چند رسانه ای و ضمیمه ها می شود.

قابل ذکر است که باب (گیرنده) نماینده کاربر خود را روی کامپیوتر محلی خود اجرا می کند، این طبیعی است که به جایگزینی یک سرویس دهنده پستی در PC محل خود بخوبی رسیدگی کند. با این راه دسترسی، سرویس دهنده پستی آلیس بصورت مستقیم با کامپیوتر باب مکالمه خواهد کرد. هر چند یک مشکل به این دسترسی وجود دارد. یادآوری می شود که یک سرویس دهنده پستی صندوق های پستی را اداره می کند و طرف های مشتری و سرویس دهنده SMTP را راه اندازی می کند. اگر سرویس دهنده پستی باب قرار بود در کامپیوتر محلی اش قرار داشته باشد بنابراین کامپیوتر باب همیشه باید روشن می بود و متصل به اینترنت، برای اینکه به نامه جدید دست یابد، که ممکن است هر زمانی برسد. این برای بسیاری از کاربران اینترنت غیر عملی است.بجای آن ، یک کاربر نمونه یک نماینده کاربر در کامپیوتر محلی اجرا می کند. اما به ذخیره صندوق پستی اش از طریق یک سرویس دهنده پستی به اشتراک گذاشته همیشه روشن دسترسی می یابد. این سرویس دهنده پیام با دیگر کاربران به اشتراک گذاشته شده است و بطور نمونه بوسیله ISP کاربر نگهداری می شود. (برای مثال دانشگاه یا کارخانه).

اکنون اجازه دهید مسیر یک پیام ایمیل را از زمانی که از طرف آلیس به باب ارسال شده است طی می کند را بررسی کنیم. ما یاد گرفتیم که در بعضی از نقاط مسیر پیام ایمیل احتیاج دارد سرویس دهنده پستی باب ذخیره شود. این می تواند به آسانی بوسیله اینکه نماینده کاربر آلیس مستقیما پیام را به سرویس دهنده پستی باب انجام شود. و این کار نیز می تواند با SMTP انجام شود- قطعا SMTP برای قرار دادن ایمیل از یک میزبان به دیگری طراحی شده است. هرچند به طور نمونه نماینده کاربر فرستنده بطور مستقیم با نماینده کاربر گیرنده صحبت و گفتگو نمی کند. سیمای آن، همانطور که در شکل ۲,۱۸ نشان داده شده است ، نماینده کاربر آلیس از SMTP برای قرار دادن پیام ایمیل به درون سرویس دهنده پستی خودش استفاده می کنند. سپس سرویس دهنده پستی آلیس SMTP را برای (بعنوان یک SMTP مشتری) رساندم پیام ایمیل به سرویس دهنده پستی باب استفاده می کند. چرا دو مرحله روش؟ اساساً بدلیل اینکه بدون رساندن از طریق سرویس دهنده پستی آلیس، نماینده کاربر آلیس هیچ گونه مراجعه ای (RECOURSE) به یک مقصد غیر قابل دسترسی سرویس دهنده پستی ندارد. با اولین ذخیره گذاری کردن آلیس در سرویس دهنده پستی خودش ، سرویس دهنده پستی آلیس می تواند بطور مداوم برای ارسال پیام به سرویس دهنده پستی باب تلاش کند، مثلا هر ۳۰ دقیقه، تا زمانی که سرویس دهنده پستی باب عملی و کاربردی شود(و اگر سرویس دهنده پستی آلیس خراب باشد، در نتیجه او مراجعه شاکیانه به مدیر سیستم خود دارد.). RFC.SMTP مشخص می کند که چگونه فرامین SMTP می توان برای رساندن یک پیام از میان چندین سرویس دهنده SMTP استفاده شود.

wWw.it98.ir

www.it98.ir

ولی همچنان یک قسمت گمشده معما وجود دارد. چگونه یک گیرنده مانند باب یک نماینده کاربر را در کامپیوتر محلی خود، برای دستیابی به پیام هایش ، که در یک سرویس دهنده پستی در محدوده ISP باب قرار دارند، اجرا می کند. گفتنی است که نماینده کاربر باب نمی تواند از SMTP برای بدست آوردن پیام ها استفاده کند، زیرا بدست آوردن پیام ها یک عملیات کششی است. در صورتی که SMTP یک پروتکل گذاشتنی است.این معما به وسیله معرفی یک پروتکل دسترسی به نامه ویژه که پیام ها را از سرویس دهنده پستی باب به PC محلی اش انتقال می دهد ، کامل می شود. در حال حاضر تعدادی پروتکل دسترسی به نامه پر کاربرد وجود دارد. شامل پروتکل اداره پست – نسخه ۳ (Post office protocol-version3) ، پروتکل دسترسی به نامه اینترنتی(IMAP) (Internet Mail Access Protocal) و http است.

شکل ۲,۱۸ خلاصه ای از پروتکل هایی را آماده کرده است که برای نامه اینترنتی استفاده می شوند : SMTP برای انتقال نامه از سرویس دهنده پستی فرستنده به سرویس دهنده پستی گیرنده استفاده می شود و SMTP نیز همچنین برای انتقال نامه از کاربر فرستنده به سرویس دهنده پستی فرستنده استفاده می شود. یک پروتکل دسترسی به نامه باشد POP_3 برای انتقال نامه از سرویس دهنده پستی گیرنده به نماینده کاربر گیرنده استفاده می شود.

POP3

POP3 یک پروتکل دسترسی نامه بسیار ساده است. در RFC1934 تعریف شده است، که کوتاه و کاملا قابل خواندن است. به دلیل سادگی بسیار زیاد پروتکل ،عملکردش نسبتا محدود شده است. POP3 هنگامی که نماینده کاربر (مشتری) یک ارتباط TCP به سرویس دهنده پستی در پورت ۱۱۰ باز می کند، شروع بکار می کند. با برقرار شدن ارتباط TCP ،POP3 در سه مرحله وفاز پیشرفت می کند : اجازه دادن، تبادل، و به روز کردن. در طول مرحله اول، اجازه دادن، نماینده کاربر یک کلمه کاربری و کلمه عبور (بطور روشن) برای اجازه عبور کاربر ارسال می کند.در طول مرحله دوم ، تبادل، نماینده کاربر پیام ها را دریافت می کند و همچنین در طول این مرحله، نماینده کاربر می تواند پیام ها را با پاک کردن علامت گذاری کند. علامت های پاک کردن را از بین ببرد و نامه های آماری رابه دست آورد.

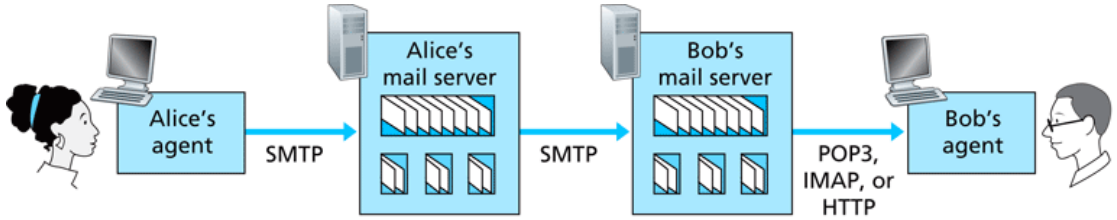


Figure 2.18 ♦ E-mail protocols and their communicating entities

مرحله سوم، به روز کردن ،بعد از اینکه مشتری فرمان quit را صادر کرد، اتفاق می افتد و جلسهpop3 به پایان می رسد.در این زمان،سرویس دهنده پستی پیام هایی را که برای پاک شدن علامت گذاری شده بودند را پاک می کند.

در یک تبادل pop3 نماینده کاربر فرامین را صادر می کند و سرویس دهنده به هر دستور با یک جواب پاسخ می دهد.دو حالت ممکن پاسخ دادن وجود دارد:ok+ (معمولا با داده server – to – client می اید)،به وسیله سرویس دهنده برای نشان دادن اینکه دستور قبلی درست بوده استفاده می شود و ERR– به وسیله سرویس دهنده برای نشان دادن این که یک چیز اشتباه است با دستور قبلی،استفاده می شود.

www.it98.ir

www.it98.ir

مرحله اجازه دادن دو دستور اصلی دارد: User<user name>, Pass< password>، برای نشان دادن این دو دستور،پیشنهاد می کنیم که شما به طور مستقیم بر روی یک سرویس دهنده pop3، Telnet کنید،از پورت ۱۱۰ استفاده کنید و این فرامین را صادر کنید. فرض کنید که mail server اسم سرویس دهنده پستی شما است.

شما چیزی شبیه این خواهید دید:

```
telnet mail server 110
+ok  pop3  server  ready
User bob
+ok
Pass hungry
+ok  user  success fully logged on
```

اگر شما یک دستور را اشتباه بنویسید، سرویس دهنده pop3 با یک پیام ERR– جواب می دهد.اکنون نگاهی به مرحله تبادل می اندازیم. یک نماینده کاربرpop3را می تواند طوری طراحی کند (به وسیله کاربر) که برای "گرفتن و حذف کردن" و یا برای "گرفتن و نگه داشتن " استفاده شود.ترتیب فرامین صادر شده یک نماینده کاربرpop3بستگی دارد که کدامیک از دو حالت نماینده کاربر کار میکند و جواب می دهد.در حالت گرفتن و حذف کردن (download and delete) ،نماینده کاربر فرامین dele,retr,list را صادر می کند.به عنوان مثال فرض کنید کاربر ۲ پیام در صندوق پستی اش دارد.در مکالمه پائین،:C (بجای مشتری) نماینده کاربر است و :S(بجای سرویس دهنده) سرویس دهنده پستی است.تبادل چیزی شبیه این خواهد بود:

```
C:list
S:1  498
```

www.it98.ir

wWw.it98.ir

www.it98.ir

S: 2 912
S: .
C: retr 1
S: (blah blah …
S:……
S: …….blah)
S: .
C: dele 1
C: retr 2
S: (blah blah …
S: ……...
S: ……blah)
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off

در ابتدا کاربر از سرور پستی (mail) درخواست لیست کردن اندازه ی هر کدام از پیغام ها را می دهد. پس از آن کاربرد هر پیغام را بازخوانی یا حذف می کند. به یاد داشته باشید که بعد از مرحله مجاز شدن (Authoritaim) کاربر از ۴ فرمان استفاده می کند. List , retr, dtle , quit . دستور این فرمان ها در RFC 1939 تعریف شده است. بعد از پردازش زمانquit سرور POP3 وارد مرحله بروز رسانی می گردد و پیغام ۱ و ۲ را از صندوق پستی پاک می‌کند.

مشکلی که در حدود چنین مدل دانلود – حذفی وجود دارد این است که گیرنده ، باب (Bob) ، ممکن است خواهان دسترسی به صندوق پیغام های خود از چندین دستگاه باشد؛ برای مثال، کامپیوتر محل کارش، کامپیوتر خانگی اش و کامپیوتر قابل حمل اش. حالت دانلود – حذف (download- deleted) صندوق پیغام های باب را به این سه دستگاه تقسیم می کند؛ و به ویژه، اگر باب اول یک پیغام را از طریق کامپیوتر محل کار خود بخواند، او قادر به خواندن درباره ی آن پیغام از طریق کامپیوتر قابل حمل خود در خانه نیست. در حالت دانلود – نگهداری (downlode- keep) : کاربر پیغام را بعد از دانلود کردن آنان در سرپرستی رها می کند. در این حالت باب می تواند از دستگاه های مختلف پیغام خود را بخواند، او می تواند به یک پیغام چه در محل کار چه در یمبداء در خانه دسترسی داشته باشد.

در طول یک مرحله POP3 بین کاربر (سرور) سرور POP3 مقداری اطلاعات state را نگهداری می کند. به ویژه پیغام هایی که توسط کاربر برای حذف انتخاب شده است را در نظر دارد. با این حال، سرور POP3 اطلاعات مربوط به حالت را بین جلسه های POP3 جابجا نمی کند. نبودن اطلاعات state بین جلسات pop3 به شکل عمده ای کاربر سرور POP3 را ساده می کند.

IMAP

با دسترسی به POP3 زمانی که باب (BOb) پیغام هایش را بر روی دستگاه خود دانلود کرد. او می تواند پوشه های پستی درست کند و پیغام های دانلود شده را به آنها منتقل کند. بعد از این کار باب می تواند پیغام ها را حذف کند یا ببین پوشه ها جابجا کند و یا آنها را جستجو کند. (با اسم فرستنده یا موضوع).

www.it98.ir

www.it98.ir

اما این نمونه پوشه ها وپیغام ها در دستگاه خودی – برای کاربری که دستگاه خود را مرتبا تغییر می دهد باعث مشکل می شود. زیرا او مایل است که یک سلسله از پوشه ها را بر روی یک سرور قابل دسترسی از راه دور قرار دهد که از طریق هر دستگاهی بتوان به آن دسترسی داشت. این امر در POP3 محقق نمی شود. پروتوکل POP3 هیچگونه امکاناتی برای کاربر جهت ساختن پوشه های از راه دور و ریختن پیغام ها در پوشه ها فراهم نمی کند.

برای حل این مشکل و مشکلات دیگر، پرتوکل IMAP که در RFC2060 تعریف شده است، اختراع شد. همانند IMAP, POP3 یک پروتوکل دسترسی به پیغام است. این پروتوکل دارای قابلیت های بسیار بیشتر نسبت به POP3 است ولی در عین حال بسیار پیچیده تر می باشد. یک سرور IMAP هر پیغام را به یک پوشه واگذار می کند، زمانی که یک پیغام برای اولین بار به سرور می رسد، آن پیغام شامل پوشه های inbox گیرنده است. گیرنده پس از دریافت می تواند پیغام را به یک پوشه جدید منتقل کند، پیغام را بخواند، آنرا حذف کند و بسیاری کارهای دیگر. پروتوکل IMAP فرمان هایی را برای بوجود آوردن پوشه های پیغام از پوشه ای به پوشه ی دیگر فراهم می کند. IMAP همچنین امکان جستجوی پوشه های دور از دسترس را برای همخوانی با شرایط ویژه فراهم می کند. توجه داشته باشید که برخلاف POP3 یک سرور IMAP اطلاعات حالت (state) کاربر را بین جلسات IMAP حفظ می کند. برای مثال، اسم پوشه ها و اینکه کدام پیغام ها به کدام پوشه ها تعلق دارد. ویژگی دیگر IMAP آن است که فرمان هایی دارد که به کاربر اجازه می دهد که به اجزای پیغام دسترسی داشته باشد. برای مثال یک کاربر می تواند فقط به سر تیتر یک پیغام یا تنها بخشی از یک پیغام چند بخشی MIME را دسترسی داشته باشد. این ویژگی زمانی که ارتباط بین کاربر و سرور پستی دارای پهنای اندکی است مفید می باشد. (برای مثال ، ارتباط بی سیم یا ارتباط کم سرعت مودم). در حالت ارتباط با پهنای باند کم، ممکن است کاربرد تمایلی به دانلود کامل پیغام در صندوق پیغام های خود نداشته باشد به ویژه پیغام های طولانی که در برخی موارد دارای کلیپ های صوتی و تصویری نیز هستند. شما می توانید در سایت رسمی IMAP بطور کامل در مورد آن مطالعه کنید (IMAP 2004).

پست الکترونیک تحت وب

روزانه به تعداد افرادی که از طریق مرورگرهای خود پست الکترونیک (e-mail) می فرستند یا به آن دسترسی دارند افزوده می شود. در اواسط دهه ۹۰، Hotmail دسترسی به e-mail از طریق وب را معرفی کرد. در حال حاضر e-mail تحت وب توسط هر سایت ISP یا دانشگاه یا شرکتی ارائه می گردد. به این روش ، کاربر از یک مرورگر معمولی استفاده می کند، و کاربر به وسیله HTTP با صندوق پستی خود در ارتباط است. زمانی که یک گیرنده مثل Bob می خواهد به یک پیغام در صندوق خود دسترسی داشته باشد. E-mail مورد نظر سرور پستی Bob به مرورگر و نه از طریق پروتکل های POP3 و IMAP بلکه از طریق پروتوکل HTTP فرستاده می شود. زمانی که یک فرستنده مثل Alice ، می خواهد یک پیغام بفرستد، آن e-mail از مرورگر او به سرور پستی او نه از طریق dMTP بلکه از طریق HTTP فرستاده می شود. سرور پستی Alice ، با این حال، هنوز هم پیغام هایی را با استفاده از SMTP به سرورهای پستی دیگر می فرستد یا دریافت می کند.

این راه حل، برای دسترسی به پیغام ها برای کاربران متحرک بسیار مفید است. کاربرد برای دریافت یا ارسال پیغام فقط نیاز به یک مرورگر دارد. این مرورگر می تواند در یک کافی نت، خانه یک دوست ؟، یک PDA یا یک اتاق مجهز به تلویزیون اینترنتی باشد. با این که با استفاده از IMAP، کاربردهای می توانند پیغام های خود را در سلسله ای از پوشه ها روی سرور منظم کنند. در واقع، بسیاری از کاربرهای پست الکترونیکی تحت وب از یک سرور IMAP برای کارایی پوشه ها استفاده می کند. در این حالت ، سرور IMAP از پروتکل IMAP استفاده می‌کنند.

wWw.it98.ir

www.it98.ir

۲,۵ سرویس راهنمای اینترنت DNS-The Internets Direcyory Servic

ما انسان ها به روش های گوناگونی شناسایی می شویم. برای ثمال ما می توانیم توسط نام شناسنامه ای شناخته شویم، می توانیم توسط شماره تامینی اجتماعی ملی شناسایی شویم و یا توسط کد ملی. اگر چه روش های مختلفی برای شناسایی ما وجود دارد ولی در هر جا ممکن است برخی مناسب تر باشند. برای مثال کامپیوترها در IRS (شرکگت به نام جمع کننده مالیات درآمریکا) ترجیح ما از شماره ملی ثابت به جای اسم در شناسنامه استفاده کند. از طرفی دیگر، مردم عادی بیشتر ترجیح دادند از اسم‌های در شناسنامه استفاده کنند تا شماره ملی (واقعا شما می توانید تصور کنید که می گویند "سلام اسم من ۹۸۷۵-۶۷-۱۳۲ است. اینم همسرم ۱۱۴۶-۸۷-۱۷۸ هستش").

همانند انسان ها، ماشین های میزبان نیز می توانند به روش های گوناگونی شناسایی شوند، یکی از آنها نام میزبان است. (host name) همانند www.yahoo.com .cnn.com ، gaia.cs.umass.edu ، cis. Poly.edu

که نام های قابل حفظ کردنی هستند و مناسب برای انسان می باشند. و ممکن است کلی اطلاعات درباره محل خود داشته باشند. (مثلا نام جزبانی مانند www.eurecom.ir به ما می گوید احتمالا ماشینی میزبان در فرانسه است). از آنجایی که نام میزبان ها دارای تعداد کارکترهایی متغیری است برای پردازش توسط مسیریاب‌ها دشوار است. برای همین منظور میزبان ها توسط آدرس های تحت عنون "آدرس IP " شناسایی می شوند.

به طور خلاصه (بحث کامل در فصل چهارم خواهد بود) آدرس IP شامل ۴ بایت است و دارای ساختار سلسله مرابی خاصی است. یک آدرس IP می تواند شبیه .121.7.106.83 FHAN جایی که هر کدام از دوره ها جداگانه نشان دهنده یکی از بایت ها در نمادگذاری ده دهی از ۰ تا ۲۵۵ است. یک IP آدرس به صورت سلسله مراتبی است بخاطر اینکه هنگامی که ما آدرس IP را از چپ به راست بررسی می کنیم،اطلاعات ویژه بیشتر و بیشتری در موردمحل و مکان میزبان در اینرنت بدست می اوریم. به طور مثابه، به زمانی که یک آدرس پستی را از پایین به بالا بررسی می‌کنیم، اطلاعات بیشتری راجع به اینکه آدرس کجا قرار دارد بدست می‌آوریم.

۲,۵,۱ سرویس های ارائه شده توسط DNS

گفته شده که دو روش برای شناسایی ماشین های میزبان وجود دارد با نام میزبان و با آدرس IP. مردم بیشتر ترجیح می دهند از شناسه های حفظ کردنی نام میزبان استفاده کنند، در حالیکه روترها ترجیح می دهند از آدرس های IP ساخت یافته ، سلسله مراتبی و با طول ثابت استفاده نمایند برای حل این مشکل ما نیاز به یک سرویس راهنما که نام میزبان را به آدرس IP تبدیل نماید ، داریم. به این کار وظیفه سیستم نام دامنه (DNS) می باشد.

گفته شده که دو روش برای شناسایی ماشین های میزبان وجود دارد با نام میزبان و با آدرس IP. مردم بیشتر ترجیح می دهند از شناسه های حفظ کردنی نام میزبان استفاده کنند، در حالیکه روترها ترجیح می دهند از آدرس های IP ساخت یافته ، سلسله مراتبی و با طول ثابت استفاده نمایند برای حل این مشکل ما نیاز به یک سرویس راهنما که نام میزبان را به آدرس IP تبدیل نماید ، داریم. به این کار وظیفه سیستم نام دامنه (DNS) می باشد. DNS یک سیستم ۱- پایگاه داده توزیع شده است در سرویس دهنده های سلسله مراتبی DNS، ۲- یک پروتکل لایه- کاربرد که اجازه می دهد میزبان ها بتوانند از پایگاه داده توزیع شده سوال نمایند. سرویس دهنده های DNS عموما ماشین های یونیکس هستند که نرم افزار BIND را اجرا می کنند. پروتکل DNS بر روی UPP و با استفاده از پورت ۵۳ اجرامی شود.

DNS عموما توسط پوتکل های لایه کاربرد دیگر نظیر HTTP ، MSTP ، FTP ، ... برای تبدیل نام های میزبانی که کاربر وارد می کند به آدرس IP مورد استفاده قرار میگیرد. به عنوان مثال توجه کنید زمانی که توسط مرورگر درخواست آدرس

www.it98.ir

www.it98.ir

WWW.SAME SCHOOL.EDU را می نماید چه اتفاقی رخ خواهد داد. برای اینکه میزبان کابر قادر به ارسال یک پیام درخواست HTTP به سرویس دهنده سایت باشد. ابتدا میزبان کاربر باید IP آدرس سایت را به دست آورد. بصورت زیر انجام شده است:

۱- بخش مشتری سیستم DNS بر روی ماشینی کاربر اجرا می شود.

۲- مرورگر نام میزبان یعنی WWW.SAME SCHOOL.EDU را از آدرس (URL) استخراج کرده و ان را به برنامه DNS سمت مشتری تحویل می دهد.

۳- برنامه بخش مشتری DNS یک درخواست (Guery) شامل نام میزبان راربه سرویس دهنده DNS ارسال می نماید.

۴- مشتری DNS سرانجام پاسخ (Reply) را دریافت می دارد که شامل آدرس IP برای این نام میزبان می باشد.

۵- بعد از دریافت آدرس IP از سرویس دهنده DNS توسط مرورگر، آن اقدام به ایجاد و یک ارتباط TCP با پروسس سرویس دهنده HTTP با شماره بورت ۸۰ و آدرس IP در یافت شده می نماید.

همانطور که ملاحضه شده DNS باعث اضافه نشدن تاخیر- گاهی اوقات قابل توجه- به برنامه های اینترنتی که از آن استفاده می کنند می شود. خوشبختانه برای کاهش این ترافیک DNS، آن را در نزدیکی سرویس دهنده هی DNS ، CACHE می نماید تا باعث کاهش متوسط تاخیر DNS بشود.

DNS همچنین برخی سرویس‌های مهم را هم فراهم می‌آورد.

- **نام‌های مستعار میزبان:** میزبانی با یک نام پیچیده می تواند یک یا تعدادی نام مستعار داشته باشد. برای یک نام میزبان همانند Replay1.west-coast.interprice.com به طور مثال می توانیم دو نام interprise.com و www.enterprise.com داشته باشیم. در این حالت نام اصل تحت عنوان نام رسمی و قانونی است. معمولا نام های مستعار راحت تر قابل حفظ کردن می باشند. DNS می تواند به وسیله یک برنامه برای بدست آوردن نام رسمی و قانونی برای یک نامستعار میزبان و ذخیره شده یه خوبی IP آدرس میزبان احضار شود.

نام‌های مستعار ایمیل: به دلایل مشخص، این بسیار مطلوب است که آدرس های ایمیل قابل حفظ تر باشد. برای مثال، اگر باب یک حساب در HOTMAIL دارد. ادرس ایمیل باب به سادگی BOB@HOTMAIL.COM می تواند باشد. هر چند نام میزبان سرویس دهنده پستی Hotmail بسیار پیچیده و کمتر قابل حفظ نسبت به آسانی hotmail.com خواهند بود (برای ثمال، نام رسمی و قانونی می تواند چیزی شبیه Replay1.west-coast.hotmail.com) باشد. Dns می تواند به وسیله یک برنامه پستی برای بدست اوردن نام رسمی و قانونی برای نام میزبان مستعار بخوبی IP آدرس میزبان احضار شود. در حقیقت، ثبت MX (به پایین مراجعه شود) به یک سرویس دهنده پستی شرکت و سرویس دهنده وب اجازه برای داشتن نام میزبان های یکسان (مستعار) می دهندو برای مثال، یک سرویس دهنده وب شرکت و سرویس دهنده پستی می تواند هر INTERPRISE.COM نامیده شوند. به طور مثال نام مستعار HOTMAIL برای نام رسمی Replay1.west-coast.hotmail.com انتخاب شده است.

توزیع بار: DNS همچنین برای اجرای توزیع بار در میان سرویس دهنده هاتکرار شده، مانند سرویس وب تکرار شده، مورد استفاده قرار می گیرند.سایت های بزرگ نظیر CNN.COM دارای چندین سرور می باشند که هر کدام روی یک ماشینی با آدرس IP متفاوتی در حال اجرا هستند. برای سرویس دهنده های وب تکرار شده، یک ردیف از IP آدرس ها بدین صورت با یک نام رسمی و قانونی ارتباط خواهند داشت. پایگاه داده DNS حاوی این ردیف از IP آدرس ها است. هنگامی که مشتری ها از DNS یک پرسش برای یک نام نگاشته یک ردیف از ادرس می کند، سرویس دهنده با ردیفی کامل از IP آدرس پاسخ می دهد، اما آدرس های

wWw.it98.ir

www.it98.ir

درخواستی را در محدوده هر جواب می چرخاند. برای اینکه یک مشتری به طور نمونه پیام درخواست HTTP خود را برای IP آدرس هایی که در ابتدا این ردیف لیست شده اند را ارسال می کند. چرخش DNS ترافیک را میان سرویس دهنده های تکرار شده توزیع می کند. چرخش DNS همچنین برای ایمیل استفاده می شود، بدین ترتیب چندین سرویس دهنده پستی می توان نام مستعار یکسانی داشته باشند. به تازگی محتوای شرکت ها توزیع مانند Akami [Akami 2004] از DNS در راه های پیچیده تر برای مهیا کردن توزیع محتوای وب استفاده کرده اند (فصل هفت را ببینید.)

DNS در RFC1034 و RFC1035 تعیین شده است و در چندین RFC اضافی به روز شده است. این یک سیستم پیچیده است و ما فقط بر اساس جوانب عملیاتش در اینجا بحث می کنیم. خوانندگانی که مشتاق هستند به این RFC و کتاب Abtiz و liv [Abtiz 1993] مراجعه کنند. همچنین مجله های قدیمی [mockaptris] که یک توضیح خوب از چیزها و چراهای DNS آماده کرده است را ببینید. زمانیکه درخواستی به CNN.COM می آید می توان توسط پایگاه داد. DNS هر بار یکی از این ماشین ها را برگرداند تا بتواند توزیع با روی آنها انجام شود.

۲,۵,۲ مروری بر نحوه کارکرد DNS

ما اکنون یک دید کلی سطح بالا از چگونگی کار DNS ارائه کردیم. بحث ما بر روی سرویس تبدیل نام میزبان به IP آدرس ها متمرکز خواهد شد. فرض کنید یک برنامه کابردی (به طور مثال مرورگر وب و یا برنامه ایمیل خوان) که بر روی ماشین میزبان کابر در حال اجرا است نیاز به ترجمه نام میزبان به آدرس IP دارد. در آن صورت برنامه کاربردی می تواند با فراخوانی بخش سمت مشتری DNS ، و با دادن نام میزبان درخواست IP را نماید این کار می تواند با فراخوانی تابعی با نام Gethostgname () انجام شود. در بخش ۲,۷ ، نشان خواهیم داد چگونه یک برنامه جاوا می تواند DNS را احضار کند. DNS درون میزبان کابر ارسال یک پیام پرسش را به درون شبکه به عهده می گیرد. تمام درخواست ها و پاسخ های مربوط به DNS از طریق دیتاگرام UDP به بورت ۵۳ انجام میگیرد. بعد از چند میلی ثانیه تا چند ثانیه DNS در میزبان کابر یک پیام پاسخ DNS را که نگاشت مطلوب را آماده می کند دریافت می کند.این نگاشت سپس به برنامه احضار می‌شود. بدین ترتیب از منظر برنامه احضار شده در میزبان کاربر، DNS یک جعبهٔ سیاه است که یک سرویس ترجمه ساده و آسان را فراهم می‌کند. سرویس DNS یک سیستم پیچیده می باشد و شامل تعداد زیادی سرور DNS می باشد که در حقیقت این جعبه سیاه که پیاده سازی نشده در سرتاسر جهان توزیع شده است، به خوبی یک پروتکل لایه کاربرد که مشخص می کند چگونه سرویس دهنده های DNS و میزبانان سوال کنند ارتباط برقرار می کنند.

یک طرح ساده برای پیاده سازی DNS می تواند شامل یک سرورDNS باشد که شامل تمام نگاشت (نام میزبان ها با آدرس IP) می باشد. در این طرح متمرکز مشتری به طور ساده تمام درخواست های خود را به یک سرور مرکزی ارسال کرده و پاسخ را دریافت می دارد اگر چه سادگی طرح جذاب است اما برای اینترنت امروزی نامناسب است. مشکل طرح متمرکز عبارتند از :

بوجود آمدن یک نقطه خرابی: اگر این سرور DNS خراب شود. کل اینترنت از کار خواهد افتاد.

حجم ترافیک : بایستی تمام درخواست های DNS که از طریق HTTP و ایمیل می آیند را شامل و صدها میلیون می باشد رسیدگی نماید.

فاصله پایگاه داده مرکزی: یک سرور DNS نمی تواند نزدیک به تمام مشتری ها باشد.اگر ما سرویس دهنده DNS تنها در شهر نیویورک قرار دهیم، بنابراین تمام پرس ها از استرالیا باید به طرف دیگر جهان سفر کنند، احتمالاً آن هم به وسیله خطوط ارتباط محدود کند که این منجر به یک تاخیر زمانی قابل ملاحضه می شود.

www.it98.ir

www.it98.ir

نگهداری: بایستی شامل رکوردهایی برای تمام میزبان های اینترنت باشد که می تواند حجم بسیار عظیمی باشد و بایستی برای هر میزبان جدید هم بروز شود.

به طور خلاصه DNS با یک سرور متمرکز، مقیاس پذیر نیست (به راحتی نمی توان آن را افزایش و گسترش داد). بنابراین این مسیر را به صورت توزیع شده طراحی نمودند. در حقیقت DNS یک مثال بسیار خوب برای چگونه یک پایگاه داده توزیع شده می تواند در اینترنت پیاده سازی شده است.

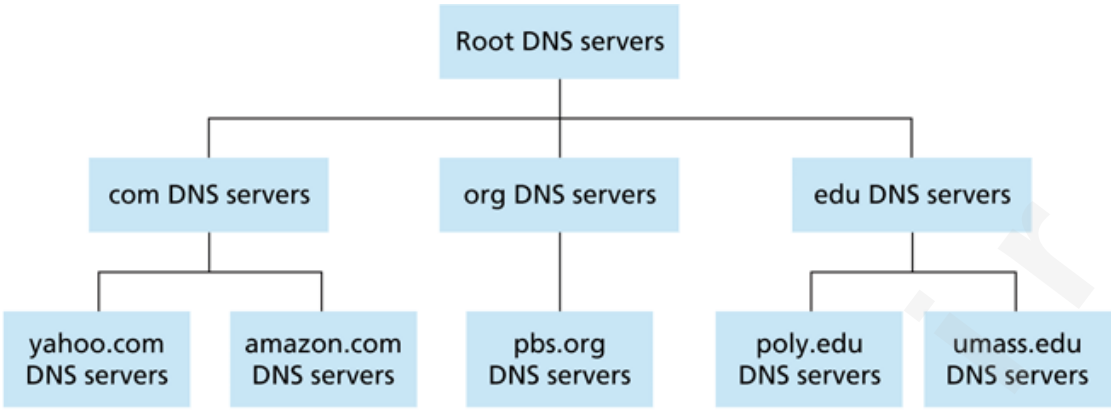


Figure 2.19 ♦ Portion of the hierarchy of DNS servers

یک پایگاه داده توزیع شده سلسله مراتبی

برای رسیدن به اهداف مقیاس پذیری، DNS از تعداد زیادی سرور که به روش سلسله مراتبی و توزیع شده در سراسر جهان می باشد، استفاده می نماید.عمل نگاشت تمام میزبان ها توسط یک سرور DNS انجام نمی گیرد بلکه به جای آن این عمل نگاشت بین تمام سرورها توزیع شده است.در تقریب اول سه کلاس سرور DNS وجود دارد؛ سرورهای DNS ریشه ROOT SERVERS، سرورهای DNS حوزه سطح بالا (TLD) Top- level domain، و سرورهای DNS مسئول که به صورت سلسه مراتبی زیر سازماندهی شده‌اند. برای درک تعامل بین این کلاس ها، فرض کنید یک مشتری DNS بخواهد آدرس IP برای نام میزبان WWW.AMAZON.COM را بدست آورد.۱) مشتری با سرور ریشه جهت دریافت TLD برای COM تماس می گیرد.۲) سپس مشتری با سرور TLD (مربوط به COM) تماس گرفته و آدرس سرور مسئول AMA7ON.COM را درخواست می نماید.۳) سپس با این سرور مسئول اخیر تماس گرفته و آدرس IP برای WWW.AMAZON.COM در یافت می نماید.

سرورهای DNS ریشه: در اینترنت ۱۳ عدد از آنها موجود می باشد. (با برچسب های A/2 تا M) بیشتر آنها در ایالات متحده می باشند. اگر ما به آنها به عنوان یک سرور نگاه می کنیم اما واقعا مجموعه ای سرور تکثیر شده می باشند، هم برای بالا بردن امنیت و هم قابلیت اطمینان. نقشه فوریه ۲۰۰۴ سرویس دهنده های DNS ریشه در شکل ۲۰۱۹ نشان داده شده اند. یک بیت از سرویس دهنده های DNS ریشه در حال حاضر در دسترس به وسیله [Root-server2004]

سرورهای با محدودهٔ سطح بالا TLD: مسئول حوزه های سطح بالا نظیر GOV,EDU,ORG,COM تمام حوزه های سطح بالای کشور TR,UK,VS,DP,IR و ... می باشد. به عنوان نمونه شرکت NETWORK SOLVTION سرورهای TLD برای COM را نگهداری می نماید. و شرکت EDUCAUSE سرورهای TLD برای EDU را نگهداری می نمایند.

سرورهای DNS مسئول: هر سازمانی با میزبان های قابل دسترسی به صورت عمومی (مانند سرورهای وب و سرورهای ایمیل) بر روی اینترنت بایستی رکوردهای با سطح دسترسی عمومی را برای نگاشت نام میزبان ها به آدرس IP آنها داشته باشند.یک سرور DNS مسئول محلی برای نگهداری این رکوردها می باشد. شرکت می توانند سرورهای مسئول خود را پیاده سازی نمایند و یا اینکه

wWw.it98.ir

www.it98.ir

در ازای پرداخت هزینه ای رکوردهای خود را بر روی سرورهای شرکت های مربوطه قرار دهند. بیشتر دانشگاههای شرکت های بزرگ سرورهای اصل و ثانویه DNS مسئول خود را نگهداری و پشتیبانی می نمایند.

سرورهای ریشه، TLD و مسئول، همگی جزو سلسله مراتب سرورهای DNS می باشند که در شکل ۲,۱۸ نشان داده شده است. سرور مهم دیگری نیز وجو دارد تحت نام "سرور DNSمحلی" که جزو سلسله مراتب DNSنیست اما در مرکز معماری DNS قرار دارد. هر ISP ، نظیر دانشگاه، شرکت ها و ... دارای یک سرور DNS محلی می باشند (که تحت عنوان سرور نام پیش فرض می‌باشد). زمانیکه ماشین میزبانی به ISP منتقل می شود، ISP برای آن آدرس یک و یا چند سرور DNS محلی را فراهم می اورد. (معمولا از طریق سروریس DHCP) می توانید به راحتی به این آدرس سرور DNS محلی با دسترسی به پنجره وضعیت شبکه دسترسی داشته باشید. این سرور DNS محلی به صورت "نزدیک به" ماشین میزبان می باشد. یک سرویس دهنده DNS محلی ممکن است روی LAN یک همانند میزبان باشد؛ برای یک ISP محلی، بطور نمونه از میزبان بوسیلهٔ تعداد اندکی مرورگر جدا شده است. زمانیکه ماشین میزبان یک درخواست DNS را تولید می کند،

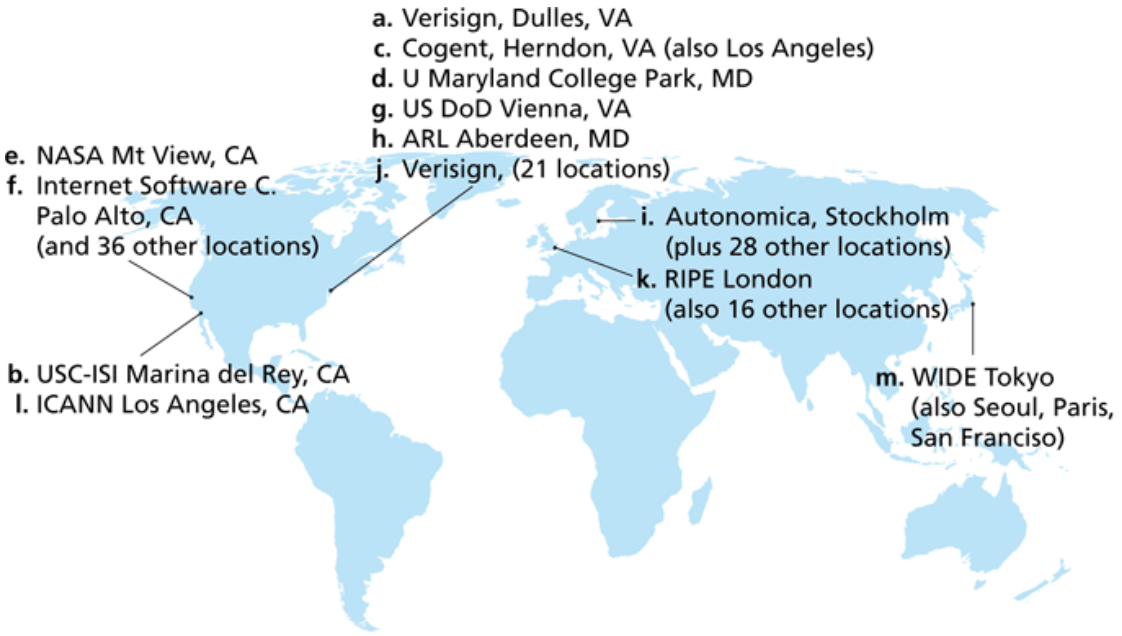


Figure 2.20 ♦ DNS root servers in 2007 (name, organization, location)

این درخواست به سرورDNS محلی فرستاده می شود و آن شبیه به یک پروکسی رخواست را به سلسله مراتب DNS ارسال رد و جواب را به ماشین میزبان بر می گرداند.

نگاهی به مثال ۵۵ می اندازیم. فرض کنید میزبان IP.CIS.POLY.EDU آدرس GAIA.CS.UMASS.EDU را درست کند. همچنین فرض کنید که سرویس دهنده DNS محلی پلی کلینیک DNS.UMASSEDU گفته می شود. همانطور که در شکل ۲,۲۰ نشان داده شده است، CIS.POLY.EDU میزبان ابتدا یک پیام پرسش، DNS محلی پیام پرسش حاوی اسم مزبان برای تبدیل شدن به اسم GAIA.CS.UMASS.EDU است. سروریس دهنده DNS محلی پیام پرسش رابه یک سرویس دهنده DNS ریشه ارسال می کند. سرویس دهنده DNS ریشهیک یادداشت از پیوند EDU بر می دارد و به سرویس دهنده DNS محلی یک لیست از IPآدرس ها برای سرویس دهنده های TLD که مسئول EDU هستند بازگت می دهد. پس سرویس

www.it98.ir

www.it98.ir

دهنده DNS محلی پیام پرسش را به یکی از این سرویس دهنده های TLD مجددا ارسال می کند. سرویس دهنده TLD یادداشتی از پیوند UMASS.EDU بر می دارد و

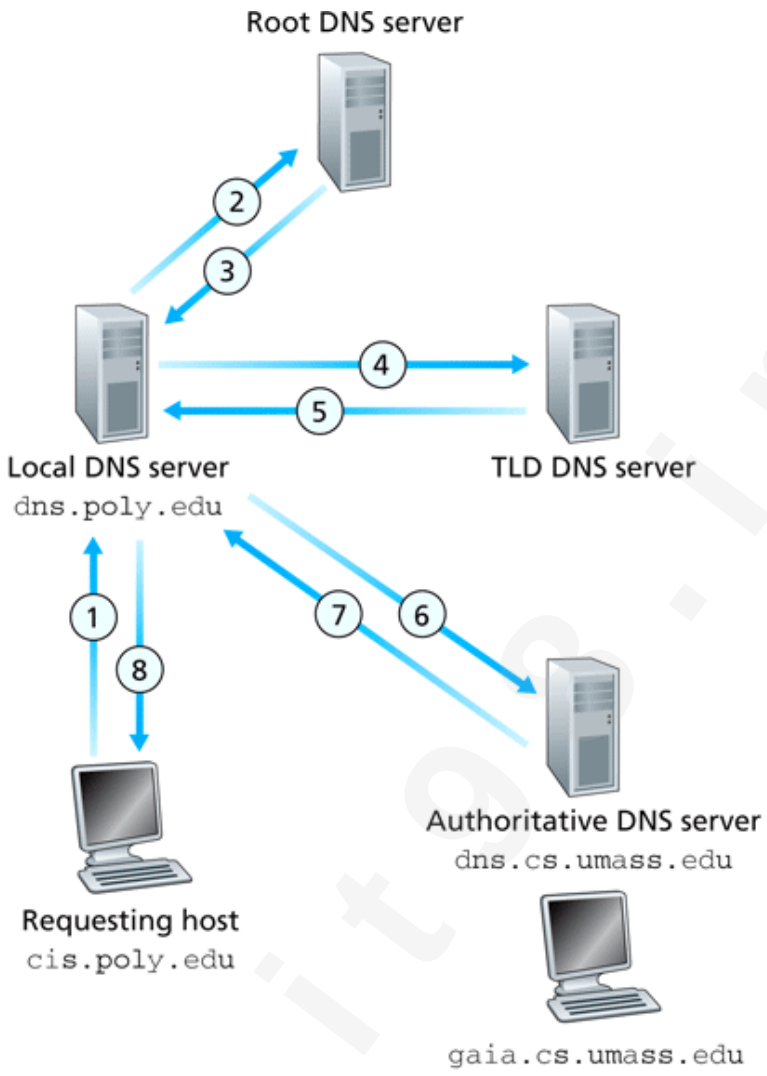


Figure 2.21 ♦ Interaction of the various DNS servers

با IP آدرس سرویس دهنده DNS مسئول برای دانشگاه ماساچوست که به نام DNS.UMASS.EDU مجددا ارسال می کند، که با IP آدرس GAIA.CS.UMASS.EDU پاسخ می دهد. گفتنی است که در این مثال، برای بدست اوردن نگاشت برای یک نام میزبان، هشت پیام DNS ارسال شده است : ۴ پیام پرسش و ۴ پیام پاسخ. بزودی خواهیم چگونه حافظه پنهانی DNS این ترافیک پرسش را کاهش می دهد.

در مثال قبل فرض شده است که سرور TLD، سرور مسئول را می شناسد، اما معمولا در عمل اینطور نیست و بجای آن سرور TLDسرور DNS میانی دیگری را می داند که آن سرور DNS مسئول را می شناسد. برای مثال، دوباره فرض کنید که دانشگاه ماساچوست یک سرویس دهنده DNS برای دانشگاه دارد که DNS.UMASS.EDU گفته می شود. همچنین فرض کنید که هر قسمتی از دانشگاه ماساچوست سرویس دهنده DNS خود را دارد و هر کدام از آن سرویس دهنده های DNS اداری تمام میزبانان DNS.UMASS.EDU و یک پرسش برای یک مزبان با نام میزبان پایانی با CS.UMASS.EDU دریافت می کند، IP آدرس DNS.CS.UMASS.EDU را به DNS.POLY.EDU بازگشت می دهد، که برای تمام نام میزبانان پایانی

wWw.it98.ir

www.it98.ir

CS.UMASS.EDU معتبر است. سرویس دهنده DNS محلی DNS.POLY.EDU سپس پرسش را به سرویس دهنده DNS مسئول ارسال می کند که نگاشت مطلوب را به سرویس دهنده DNS محلی بازگشت می دهد، که نوبت مجدداً نگاشت را به میزبان متقاضی باز میگرداند. در این حالت مجموع کل پیام های ارسال شده برابر با ۱۰ می شود.

درخواست ارائه شده تا کنون تحت عنوان "درخواست تکراری" مطرح است. نوع دیگری هم وجود دارد تحت عنوان "درخواست بازگشتی" که مراحل کار آن در شکل ۲,۲۲ نشان داده شده است.

پرسش که از CIS.POLY.EDU به POLY.EDU شده یک درخواست بازگشتی است، به این دلیل که پرسش از DNS.POLY.EDU برای به دست آوردن نگاشت از طرف آن درخواست می کند. اما سه درخواست متوالی و پشت سر هم از زمانی که همه پاسخ ها به طور مستقیم به DNS.POLY.EDU بازگشت داده شده اند، تکرار می شوند. فرضیه ای، هر درخواست DNS می تواند "تکراری" یا " گردشی" باشد. برای مثال شکل ۲,۲۱ یک پرسش DNS زنجیره ای برای که برای تمام درخواست ها گردشی است را نشان می دهد. در عمل، پرسش نها به طور نمونه طرح در شکل ۲,۲۰ را دنبال می کنند. در خواست از میزبان متقاضی به سرویس دهنده DNS محلی "بازگشتی" لاست و بقیه پرسش ها "تکراری" .

کش کردن

جهت بالا بردن کارایی و کاهش تاخیر درخواست های DNS (سرور DNS محلی) ، زمانی که جوابی از سلسله مراتب دریافت می دارد آن را در حافظه خود نگهداری می نماید. در صورتی که مجددا همان تقاضا به آن رسد بجای دریافت از سلسله مراتب DNS، آن را از حافظه خود بر میگرداند.

نظریه پشت کش کردن DNS بسیار ساده است. در یک زنجیره درخواستی، هنگامی که یک سرویس دهنده DNS یک پاسخ DNS را دریافت می کند (حاوی مثلا یک نگاشت برای یک نام میزبان به یک IP آدرس) ، اطلاعات درون پاسخ را می تواند در حافظه محلی خود کش کند.

برای مثال در شکل ۲,۲۰، هر زمان سرویس دهنده DNS محلی Dns.poly.edu یک پاسخ از بعضی سرویس دهنده های dns دریافت کند، می تواند هر اطلاعات موجود در پاسخ را کش کند. اگر یک جفت نام میزبان/ IP آدرس در یک سرویس دهنده DNS کش شده باشد و درخواست دیگری به سرویس دهنده DNS برای نام میزبان برسد، سرویس دهنده DNS می تواند IP آدرس مطلوب را فراهم کند.

www.it98.ir

www.it98.ir

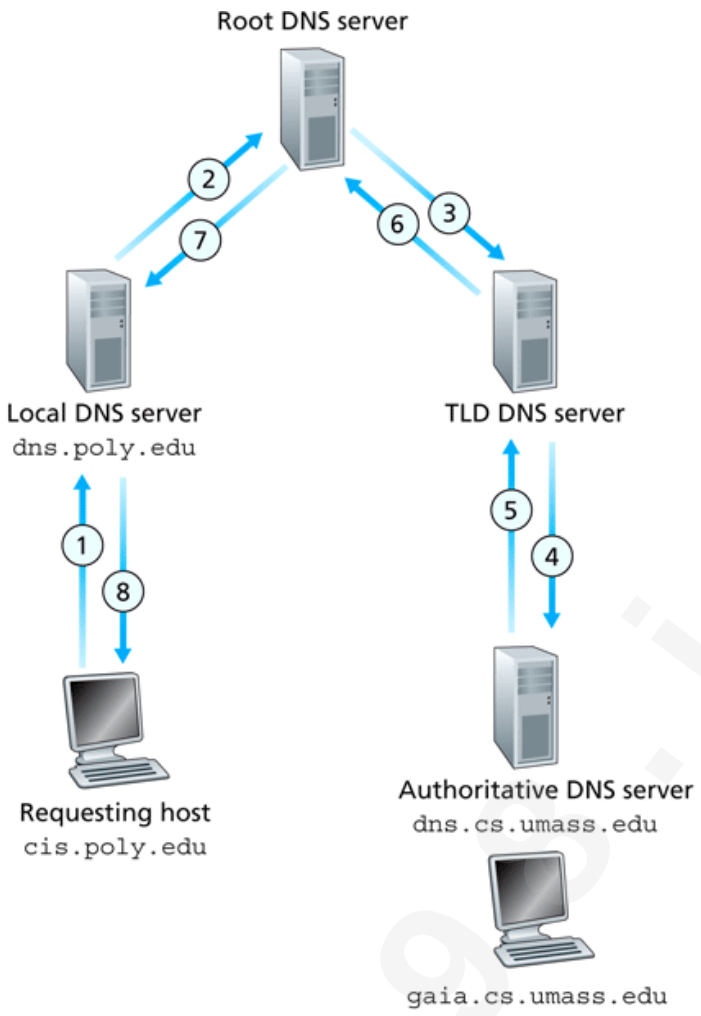


Figure 2.22 ♦ Recursive queries in DNS

حتی اگر برای نام میزبان معتبر نباشد. به دلیل اینکه میزبانان و نگاشت ها بین نام میزبان و IP ادرس ها به هیچ وجه دائمی نیستند و سرویس دهنده های DNS اطلاعات ذخیره شده را بعد از یک مدتی از زمان (معمولا روی دو روز تنظیم شده اند) دور می اندازد.

به عنوان مثال، فرض کنید میزبان OPRICOT.POLY.EDU از DNS.POLY.EDU برای IP آدرس برای نام میزبان CNN.COM درخواست کند. به علاوه، فرض کنید که چند ساعت بعد، میزبان دستگاه پلی کلینیک دیگری، مثلا KIWI.POLY.FR همچنین در DNS.POLY.EDU با نام میزبان یکسان درخواست کند. بخاطر کش کردن، سرویس دهنده DNS محلی قادر خواهد بود به سرعت IP آدرس CNN.COM برای این دومین میزبان متقاضی بدون اینکه بخواهد از سرویس دهنده DNS دیگری درخواست کند، بازگشت دهد. یک سرویس دهنده CNS محلی همچنین می تواند IP آدرس های سرویس دهنده TLD را ذخیره سازی کند. در نتیجه به DNS محلی اجازه می دهد که سرویس دهنده‌های DNS ریشه در یک زنجیر درخواست دور نزند و نادیده بگیرد (این اغلب اتفاق می‌افتد).

۲،۵،۳ رکوردهای DNS وپیام ها

سرورهای DNS که بر روی هم پایگاه داد، توزیع شده DNS را پیاده سازی م نماید. رکوردهای منبع (RR) را نگهداری می نماید. هر رکورد (RR) شامل نگشات های نام میزبان به ادرس IP می باشد. هر پیام پاسخ حاوی یک یا تعددی رکورد منبع است.

wWw.it98.ir

www.it98.ir

در این بخش بعدی خلاصه ای کامل از رکوردهای منبع DNS و پیام آماده کرده ایم. جزئیات بیشتر را می توانید در [Abtiz1993] یا RFCS DNS پیدا کنید.

یک رکورد منبع ۴ بخش بود و به صورت زیر است .

نام میزبان- مقدار آدرس- نوع زمان عمر (NAME , VALUE , TYPE , TTL)

TTL تحت عنوان زمان عمر بسته است و نشان می دهد که این رکورد چقدر بایستی از حافظه نگهداری (CAGHE) شود. در مثال رکوردهای داده شده زیر، بخش TTL را نادیه گرفته ایم. معنی NDME و VALUE به TYPE بستگی دارد.

TYPE = A : به معنی این است که NAME نام میزبان است و VALUE آدرس IP است. (REPLAY1.BAR.FOO.COM, 143.37.93.126,A) در این ترتیب رکورد TYPE A نام میزبان استاندارد را برای نگشات IP آدرس مهیا می کند. مانند مثال بالا که یک TYPE A است.

TYPE=NS : NAME یک نام حوزه است. (مثلا FOO.COM) و VALUE نام میزبان سرور DNS مسئول می باشد که آدرس IP میزبان های حوزه را می داند. این رکورد برای مسیریابی درخواست های DNS در زنجیره درخواست های DNS می باشد. برای مثال (FOO.COM , DNS. FOO. COM, NS)C NAME =TYPE ، VALUE یک نام رسمی برای نام میزبان NAME است. این رکورد می تواند میزبانان متقاضی اسم رسمی برای یک نام میزبان را مهیا می کند.

VALUE , TYPE=MX نام رسمی سرور ایمیل است که دارای نام مستعار میزبان NAME است.

(FOO.COM, MAIL.BAR.FOO.COM,MX) رکوردهای MX به نام میزبانان سرویس دهنده های پستی اجازه می دهد نام مستعار ساده داشته باشند. گفتنی که به وسیله استفاده از رکورد MX، یک شرکت می تواند نام مستعار مشابه برای سرویس دهنده پستی خود و برای یکی دیگر از سرویس دهنده ها خود داشته باشند (مانند سرویس دهنه خود وب) برای به دست اوردن نام رسمی برای سرویس دهنده های پستی، یک کشتری DNS برای یک رکورد MX درخواست خواهد کرد. برای به دست آوردن اسم مستعار برای دیگر سروریس دهنده ها، مشتری DNS برای رکورد C NAME درخواست خواهد کرد.

اگر سرور DNS برای یک نام میزبان مسئول باشد، حتما حاوی رکوردی با TYPE=A می باشد. (حتی اگر سرویس دهنده DNS نام میزبان مسئول هم نباشد ممکن است حاوی یک TYPE A در حافظه پنهان خود باشد) اگر مسئول نباشد، شامل رکوردی با TYPE=NS می باشد که به سروری اشاره می کند که آن مسئول است و حاوی رکورد با نوع A باشد که IPآدرس سرویس دهنده DNS را در فیلد VALUE رکورد DNS فراهم می کند.

به طور مثال فرض کنید یک سرور TLD مربوط به EDU برای میزبان GALA.CSUMASS.EDU سپس در همین سرور (EDU TLD) رکوردی است که آدرس IP مربوط : DNS.UMASS.EDU را دارا باشد. (DNS.UMASS.EDU, 128.11.40.11,A)

پیام های DNS

قبلا اشاره کرده بودیم که ما دو نوع پیام DNS تحت نام های درخواست و پاسخ داریم. هر نوع پیام دارای یک قالب مشابه شکل زیر می باشند. که در شکل ۲,۲۲ نشان داده شده است. معنای متغیر در یک پیام DNS درپایین آمده است.

* ۱۲ بایت اول (بخش هدر) که شامل تعدادی فیلد می باشد.اولین فیلد عدد ۱۶ بیتی که شناسه پرسش است.این شناسه در پیام پاسخ کپی می شود که نشان دهنده جواب به کدام پرسش است. تعدادی پرچم در فیلد پرچم وجود دارد. بیت پرچم نشانده پرسش

www.it98.ir

www.it98.ir

و یا پاسخ بودن پیام می باشد.۱ بیت پرچم مسئول بودن مانی که سرور مسئولی جواب را تولید می کند برابر با یک می شود.۲ بیت پرچم نشان دهنده بازگشتی بودن برای نوع درخواست می باشد.هنگامی یک مشتری (میزبان یا سرویس دهندهDNS) که سرویس دهنده DNS عمل بازگشتی را هنگامی که رکوردی ندارد، برابر ۱ می شود. ۱ بیت فیلد در دسترس بازگشتی در یک پاسخ اگر سرویس دهنده DNS بازگشت را پشتیبانی کند برابر ۱ می شود.در این بخش همچنین ۴ فیلد تعداد نیز می باشد که نشان دهنده هر نوع بخش می باشد.

* بخش سوال- شامل اطلاعاتی راجع به پرسشی که ساخته شده می باشد.این بخش شامل فیلد نام که شامل نامی است که پرسش می شود.فیلد نوع که نوع پرسش را مشخص می نماید.برای مثال . یک آدرس میزبان میانی با یک نام (TYPE A) یا سرویس دهنده پستی برای یک نام (TYPE MX).

* بخش پاسخ، شامل رکوردهای منبع(RR) برای نام اولیه درخواست شده می باشد. گفته شود که در هر رکوردهای منبع (RR) ، TYPE [برای مثال :MX, A) ، VALUE و TTL قرار دارد. یک پاسخ می تواند حاوی تعداد RR باشد که نشان دهنده آدرس های مختلف برای یک نام میزبان باشد. (برای مثال، برای سرویس دهنده های وب تکرار شده، که به زودی در این بخش بحث می کنیم).

* بخش مسئول که شامل رکوردهایی برای سرورهای مسئول دیگر می باشد.

* بخش اضافی شامل رکوردهای مفید دیگر می باشد. برای مثال، فیلد جواب در یک پاسخ برای یک پرسش MX حاوی یک رکورد منبع (RR) که یک نام میزبان قانونی برای یک سرویس دهنده پستی فراهم میکند، است. بخش اضافی حاوی یک رکورد TYPE A که IP آدرس را برای نام میزبان قانونی سرویس دهنده پستی فراهم می کند، است.

چگونه شما دوست دارید یک پیام درخواست DNS را مستقیماً از میزبانی که با آن کار می کنید به تعدادی سرویس دهنده DNS بفرستید ؟ توسط دستور NSLOOKUP چه به عنوان دستور ویندوزی و یا از طریق سایت می توان با سرورهای سلسله مراتب DNS ارتباط برقرار نمود. برای مثال : از یک میزبان ویندوز، COMMAND PROMPT را اجرا و برنامه NSLOOKUP را به وسیله به سادگی نوشتن "NS LOOKUP" احضار می کنید. بعد از احضارNS LOOKUP شما می توانید یک درخواست DNS به هر سرویس دهنده DNS بفرستید، NS LOOKUP رکوردهای شامل شده در پاسخ را نمایش می دهد (درقالبی که توسط انسان قابل خواندن باشد) به عنوان یک جانشین برای اجرای NS LOOKUP از میزبان شخصی خودتان، می توانید یکی از بسیار وب سایت که به شما اجازه می دهد تا NS LOOKUP را از راه دور به خدمت بگیرید، دیدن کنید. (فقط بنویسید "NS LOOKUP " در یک موتور جستجوگر و شما یکی از این سایت را خواهید آورد.

اضافه کردن رکوردها به پایگاه داده DNS

بحث بالا بر روی اینکه چگونه رکوردها از پایگاه داده DNS بازیابی خواهند شد متمرکز شده بود. سوال جواب این است که این رکوردها قبلا چگونه در این پایگاه داده قرار گرفته اند. فرض کنید شما شرکتی ایجاد کرده اید با نام NETWORK UTOPIU . اولین چیزی که مایل به انجام آن هستید ثبت دامنه ای با نام حوزهETWORK UTOPIU.COM توسط یک ثبت کننده (REGISTRAR) نام حوزه.

یک ثبت کننده یک واحد تجاری است که یکتا بودن نام حوزه شما را چک کرد و می توان نام حوزه شما را وارد پایگاه داده DNS بنماید وبرای این کار مقدار پول از شما می گیرد.تا قبل از سال ۱۹۹۹، تنها ثبت کننده "NETWORK SOLVTIONS" بود که حق انحصاری در ثبت نام های حوزه COM و NET و ORG داشت.

wWw.it98.ir

Identification	Flags	12 bytes
Number of questions	Number of answer RRs	
Number of authority RRs	Number of additional RRs	
Questions (variable number of questions)		Name, type fields for a query
Answers (variable number of resource records)		RRs in response to query
Authority (variable number of resource records)		Records for authoritative servers
Additional information (variable number of resource records)		Additional "helpful" info that may be used

Figure 2.23 ♦ DNS message format

www.it98.ir

www.it98.ir

اما هم اکنون شرکت های متعدد ثبت کننده در رقابت برای کسب مشتری هستند. و تحت نظارت ICANN (INTERNET CORPOTATION FOR ASSIGNED NAME AND NUMBERS) عمل می نمایند. لیست کامل این شرکت ها در سایت HTTP://WWW.INTERNIC.NET موجود است.

زمانیکه شما نام حوزه NETWORK UTOPIA.COM را با یک شرکت ثبت کننده ثبت می کنید، بایستی نام و آدرس IP مربوط به سرورهای DNS اصلی و ثانویه خود را نیز ارائه نمائید.فرض کنید که نام و آدرس IP آنها به صورت زیر باشد. , DNS1.NETWORKUTOPIA.COM 212.212.212.1DNS2.NETWORKUTOPIA.COM 212.212.212.1 برای هر کدام از این دو نوع سرور DNS مسئول، ثبت کننده بایستی یک رکورد از نوع NS و یک رکورد از نوع A در سرورهای DNS TLD ایجاد نماید. به صورت زیر : به ویژه، برای سرویس دهنده مسئول اولیه ثبت کننده دورکورد منبع در این آمده را درون سیستم DNS وارد خواهد کرد.

(NETWORKUTOPIA.COM,DNS1.NETWORKUTOPIA.COM,NS)
(DNS1.NETWORKUTOPIA.COM 212.212.212.1, A)

شما همچنین مطمئن باشید که یک رکورد از نوع A برای مرور وب خود یعنی WWW.NETWORKUTOPIA.COM و یک نوع رکورد از نوع MX برای سرور ایمیل خود MAIL.NETWORKUTOPIA.COM در سرور DNS مسئول خود را وارد نمایئید. بعد از انجام دادن این عملیات مردم می تواند وب سایت شما را دیده و به شرکت شما ایمیل ارسال نمائید.

از بحث مان در مورد DNS بوسیله اثبات اینکه این بیانیه درست است، نتیجه گیری می کنیم و این اثبات همچنین کمک به متحد کردن چیزهایی که از مورد DNS یاد گرفته ایم، می کند. فرض کنید آلیس رادر استرالیا می خواهد صفحه وب WWW.NETWORKUTOPIA.COM را مشاهده کند. هانطور که گفتیم، میزبان او ابتدا یک درخواست DNS محلی اش ارسال خواهد کرد. سپس سرویس دهنده DNS محلی تماس با یک سرویس دهنده TLD.COM خواهد گرفت. سرویس دهنده DNSمحلی همچنین باید تماس بایک سرویس دهنده NDS ریشه بگیرد اگر آدرس یک سرویس دهنده TLD.COM ذخیره نشده باشد). این سرویس دهنده TLD حاوی TYPES , TYPE A رکوردهای منبع لست شده در بالا است، زیرا ثبت کننده این رکورد های منبع را به درون همه سرویس دهنده های TLD COM وارد کرده بود. سرویس دهنده TLD COM یک پاسخ به سرویس دهنده DNS محلی الیس می فرستد، با پاسخ دو رکورد منبع وجود دارد. سپس سرویس دهنده DNS محلی یک درخواست DNS به 212.212.212.1 ارسال می کندو برای رکورد TYPE A به WWW.NETWORKUTOPIA مکاتبه میکند. این رکورد IP آدرس مطلوب سرویس دهنده وب را مانند 212.212.71.4 فراهم می کند که سرویس دهنده DNS محلی به میزبان آلیس بازگشت می دهد. جستجو گر آلیس می تواند اکنون یک تماس TCP به میزبان 212.212.71.4 آغاز کند و یک درخواست HTTP به وسیله ارتباط بفرستد.

۲,۶ کاربردهای نظیر به نظیر

یک مولفه بحرانی بسیاری از کاربردهای PZP، اندیس کردن اطلاعات- یک نگاشت اطلاعات به محل میزبان ها- می باشد. در چنین کاربردهایی، جفت ها به طور دینامیک اندیس را به روز کرده و جستجو می کنند. از آنجایی که مفهوم " یک نگاشت اطلاعات به محل میزبان ها " ممکن است کمی انتزاعی به نظر برسد، اجازه دهید به مثال های مشخصی بپردازیم. در یک سیستم اشتراک گذاری فایل PZP، عموما یک تعداد بسیار زیادی جفت شرکت کننده وجود دارد که هر جفت فایل هایی برای به اشتراک گذاری دارد، شامل MP3، ویدئو، تصاویر و نرم افزار. در چنین سیستمی یک اندیس وجود دارد که به طور دینامیک فایل هایی که هر جفت

wWw.it98.ir

www.it98.ir

برای اشتراک گذاری آماده کرده است دنبال می نماید. برای هر کپی از هر فایل که توسط اجتماع جفت ها اشتراک گذاری شده است. اندیس یک رکوردی که نگاشت اطلاعات مربوط به کپی (برای مثال یک آهنگ MP3 است، عنوان آهنگ و نظایر آن) به آدرس IP آن جفت که کپی را دارد، نگهداری می نماید.

این اندیس به طور دینامیک با ورود و یا خروج هر جفت و یا با بدست آوردن یک کپی جدیدی از فایل، به روز می شود. برای مثال زمانی که یک جفت به سیستم ملحق می شود، فایل های خود را به اندیس اطلاع می دهد. زمانی که یک کابر خاص، آلیس، می خواهد فایل خاصی را بدست آورد، او اندیس را برای بافتی محل کپی آن فایل، مورد جستجو قرار می دهد. بعد از اینکه جفتی که دارای چنین کپی از فایل است را پیدا نمود، او می تواند اقدام به دریافت فایل از آن جفت بنماید. زمانی که او این فایل را به طور کامل بدست آورد، اندیس به روز رسانی می شود تا اینکه شامل کپی آلیس از این فایل هم باشد.

در کاربردهای پیام رسانی فوری (IM)، اندیس وجود دارد که نگاشت نام کاربران به محل آنها (آدس های IP) می باشد. برای درک اهمیت وجود اندیس در چنین کاربردهایی به دو کاربر آلیس زیبا و باب خوش تیپ که هر کدام درگیت دوستان دیگری هستند، توجه نمائید. وقتی که باب برنامه مشتری پیام رسان فوری خود را روی میزبانی با آدرس IP معادل X شروع می نماید، برنامه مشتری او به اندیس اعلام می نماید که باب با آدرس X روی خواست. بعداً زمانیکه آلیس برنامه سمت مشتری پیام رسان خود را اجرا می کند، از آنجایی که باب درگیت دوستان او می باشد، برنامه سمت مشتری او، اندیس را برای دریافت اینکه آیا باب روی خط است یا نه جستجو می نماید و متوجه می شود که باب روی خط بود و دارای آدرس X می باشد سپس آلیس می تواند یک ارتباط مستقیم TCP با میزبان با آدرس X بنماید و شروع به پیام رسانی فوری با باب بنماید.

علاوه بر پیام رسانی فوری، بسیاری از کاربردهای امروزی از اندیس را برای دنبال کردن حضور فردی استفاده می نمایند به طور مثال در سیستم تلفن اینترنتی (بخش ۲/۶/۳ را نگاه کنید)

به طور خلاصه اشاره کردیم که پروتکل بیت تورنت که منحصراً یک پروتکل توزیع فایل است و کارکردهای اندیس کردن و جستجو کردن را ارائه می دهد. در ادامه ما سه روش را که برای سازماندهی و جستجوی اندیس در یک اجتماع جفت ها است مورد بحث قرار خواهیم داد. برای تجسم بهتر ما در اینجا جستجو برای یک فایل در سیستم اشتراک گذاری P2P را بررسی خواهیم نمود اما تاکید می نمائیم که این بحث معادل اعمال جستجو برای هر نوع اطلاعاتی در یک اجتماع P2P می باشد.

اندیس متمرکز

یکی از سر راست ترین روشها برای یافتن یک فیال فراهم آوردن یک اندیس متمرکز می باشد، همانطوریکه توسط Napster ، اولین کاربرد اشتراک گذاری فایل P2P تجاری با مقیاس بزرگ، انجام شد. در این طرح سرویس اندیس توسط یک سرور بزرگ (و یا مزرعه سرویس) فراهم می شود. همانطوریکه در شکل ۲/۲۳ نشان داده شده است، زمانیکه یک کاربر برنامه اشتراک گذاری فایل P2P را اجرا می نماید، برنامه به سرویس اندیس در مورد آدرس IP خود و نام فایل هایی که برای اشتراک گذاری آماده کرده است، اطلاع می دهد. (برای مثال عنوان تمام MP3s های خود). سرور اندیس این اطلاعات را برای هر جفت که فعال شده باشد جمع آوری نموده و بنابراین یک اندیس دینامیک و متمرکزی ایجاد کرده که نگاشت هر کپی از فایل با آدرس های IP را نگاشت می نماید. توجه داشته باشید که سیستم اشتراک گذاری فایل به ورت P2P با اندیس متمرکز در واقع یک معماری ترکیبی P2P و مشتری سرویس دهنده می باشد.

www.it98.ir

www.it98.ir

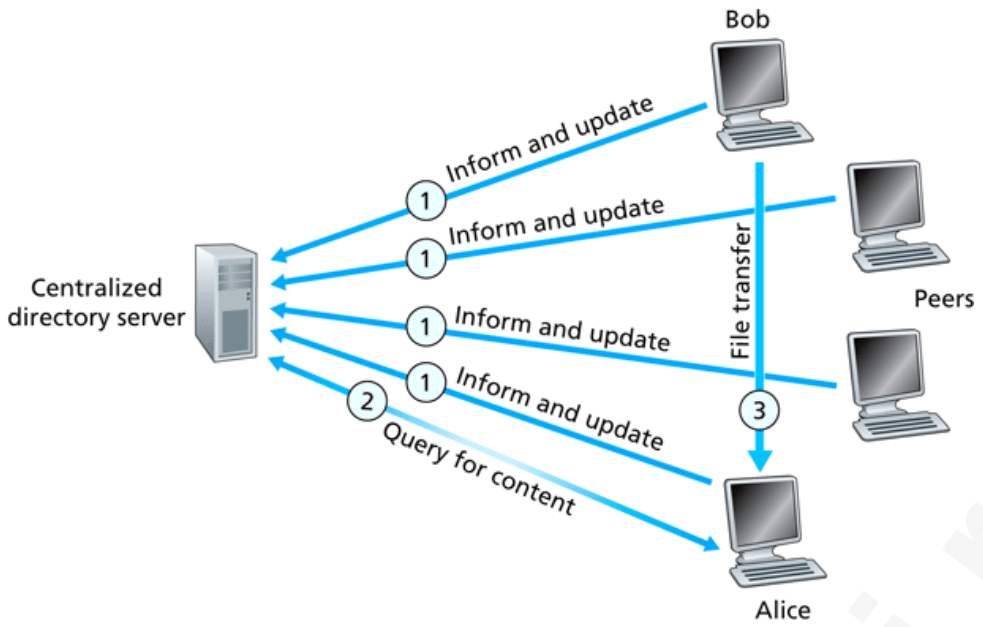


Figure 2.27 ♦ Centralized index

چنین معماری مرکبی در بسیاری از کاربردهای امروزی نظیر برنامه های پیام رسانی فوری یافت می شود. استفاده از یک اندیس متمرکز برای یافتن اطلاعات از نظر مفهومی سه راست بوده اما دارای معایبی نیز می باشد :

- نقطه تنهای خرابی- اگر سرویس دهندها کار بیفتد، کل سیستم از کار خواهد افتاد. حتی اگر یک مزرعه سرور با سرورهای افزونه مورد استفاده قرار گیرد، ارتباطات اینترنت به مزرعه سرورها دچار مشکل خواهد شد و باعث از کار افتادن برنامه خواهد شد.

- گلوگاه کارایی و هزینه زیر ساخت در یک سیستم P2P بزرگ با صدها هزار کاربر متصل به آن، یک سرور مرکزی بایستی یک اندیس بسیار عظیمی را نگهدری نموده و بایستی در هر ثانیه به هزار درخواست پاسخ بدهد. در واقع در سال ۲۰۰۰، زمانیکه NAPSTER محبوب ترین کابرد اشتراک گذاری فایل PZZ بود، به همین مشکل ترافیک دچار شد.

- نقض کپی رایت. اگر چه این موضوع خارج از حیطه این کتاب می باشد. ما به طور خلاصه اشاره می کنیم که صنعت ضبط ادعا کرده است که سیستم اشتراک گذاری فایل به صورت PZP به کاربران اجازه می دهد که به راحتی بتوانند مطالب کپی رایت را به طور مجانی بدست آورند. (برای دیدن یک بحث جالب در مورد قوانین کپی رایت در مورد PZP به (VON LOHMANN 2003 رجوع کنید) زمانیکه یک شرکت اشتراک گذاری فایل به صورت PZP یک سرور متمرکز داشته باشد، اقدامات قانونی ممکن است باعث خاموش کردن سرور اندیس بشود. اما خاموش کردن معماری غیر متمرکز به این سادگی ها نخواهد بود.

طوفان پرس و جوها

درست در انتهای طیف مخالف فهرست های متمرکز، روش کاملاً غیرمتمرکز طوفان پرس و جوها می باشد. طوفان پرس و جوها برای اولین بار توسط پروتکل کنوتلا (CNUTTELA) بکار برده شد در اینروش اندیس در طول اجتماع جفت ها توزیع شده است. هر جفت فقط فایل هایی را که خود به اشتراک گذاشته اندیس می کند و نه بیشتر.

در طوفان جستجوها، جفت ها یک شبکه انتزاعی و منطقی به نام شبکه پوشش ها را ایجاد می نمایند، که در اصطلاح تئوری گرافها بصورت زیر تعریف می شود.اگر جفت x یک ارتباط TCP را با جفت y داشته باشد، آنگاه می گوییم یک یال بین X و Y وجود دارد. گراف شامل تمامی جفت های فعال و یال های ارتباطی شبکه پوششی را تعریف می نماید. توجه داشته باشید که یک

wWw.it98.ir

www.it98.ir

یال، یک لینک ارتباطی فیزیکی نیست، بلکه یک یال، یک لینک انتزاعی بوده که ممکن است شامل چندین لینک فیزیکی باشد. یک یال در شبکه پوششی به طور مثال ممکن است نشان دهنده ارتباط TCP بین یک جفت در لیتوانی و جفت دیگر در برزیل باشد.

اگر چه چنین شبکه پوششی ممکن است شامل صدها هزار جفت شرکت کننده باشد اما یک جفت معین به طور معمول به تعداد بسیار کمینود دیگر در شبکه پوشی (معمولا کمتر از ۱۰ عدد) متصل می باشد. همانند شکل ۲/۲۸. در ادامه خواهیم دید که چنین شبکه پوششی چگونه با ورود و خروج جفت ها تشکیل شده و بر جا خواهد بود. هم اکنون اجازه دهید فرض کنیم که این شبکه پوششی موجود بوده و ما فقط بر روی اینکه یک جفت چگونه محتوایی را بازیابی نماید، تمرکز می نماییم.

در این طرح، جفت ها پیام را از طریق شبکه پوششی از روی ارتباطات TCP از قبل موجود، به همسایگان خود می فرستند. زمانی که آلیس می خواهد محتوایی در مورد "عشق شبکه" را پیدا کند، برنامه سمت مشتری امر یک پیام پرس و جو که شامل کلمات کلیدی " عشق شبکه" می باشد را به تمام همسایگان خود هدایت می نمایند و همین عمل نیز ادامه خواهد یافت.

چنین طوفان پرس و جویی در شکل ۲/۲۸ نشان داده شده است. زمانی که جفتی یک پرس و جو را دریافت می دارد، بررسی می نماید که آیا مطابقتی با فایل هایی که خودش برای اشتراک گذاری کذاشته دارد یا خیر. اگر مطابقتی پیدا شود، آن جفت یک پیام برخورد پرس و جو را به آلیس بر می گرداند که این پیام شامل نام فایل و اندازه فایل می باشد. پیام برخورد پرس وجو مسیر معکوس را پیمایش کرد نما به آلیس برسد. در این حالت آلیس متوجه می شود که جفتی یک کپی از فایل مورد نظر او را دارا می باشد. اگر چه چنین طرح غیر متمرکزی ساده و زیبا به نظر می رسد، اما انتقادهایی هم برای غیرمقیاس پذیر بودن دارا می باشد. به ویژه زمانی که یک طوفان پرس و جویی انجام می شود، درخواست به هر جفتی که در شبکه پوششی وجود دارد ارسال شد و باعث می شود که یک حجم ترافیک بسیار زیادی بین جفت ها در شبکه (به طور مثال اینترنت) ایجاد شود. طراحان گنوتلا با استفاده از طوفان پرس و جوهای محدود به این مشکل جواب دادند.

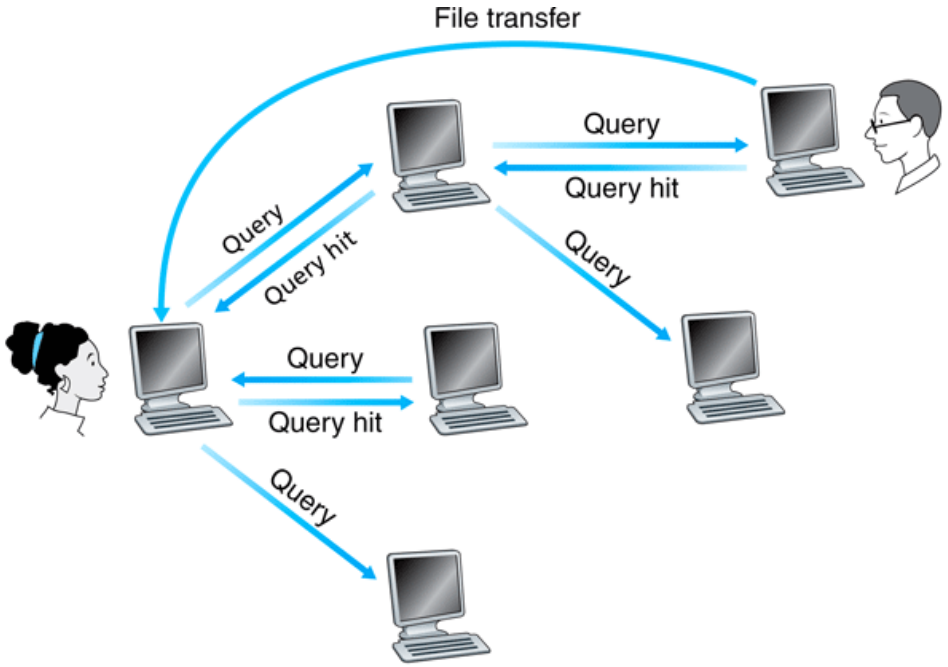


Figure 2.28 ♦ Query flooding

کاربردهای نظیر به نظیر

www.it98.ir

www.it98.ir

همانطوری که در بخش ۲/۱/۱ گفتیم یک برنامه کاربردی یا به صورت معماری مشتری سرویس دهنده و یا معماری نظیر به نظیر طراحی می شود. کاربردهایی که تا کنون در این فصل توضیح داده شدند، شامل وب ایمیل، و DNS ، همگی از معماری مشتری سرویس دهنده استفاده کرد و اتکای زیادی به زیرساخت های سروری می باشد. بجای آن جفت ها بطور مستقیم با هم ارتباط برقرار می نمایند. جفت ها توسط فراهم آورندگان سرویس نگهداری نمی شوند، بلکه به جای آن ماشین های رومیزی و لب تابهایی هستند که توسط کاربران کنترل می شوند.

در این بخش ما سه کاربرد مختلف که به طور مناسبی برای شروع تحقیق در مورد P2P که به طور شفافی قابلیت خود مقیاس پذیری معماری P2P را نمایش می دهد.

بعنوان یک مثال خاص، پروتکل معروف Bittorrent را بررسی خواهیم نمود.

کاربرد دوم در مورد سازماندهی و جستجوی اطلاعات در یک اجتماعی از جفت ها می باشد. برای این کاربرد ما تعدادی طرح مختلف را بررسی خواهیم نمود که هر کدام از آنها در سیستم های اشتراک گذاری فایل به صورت pzp در مقیاس بزرگ مورد استفاده قرار گرفته اند. و برای کاربرد سوم، ما برنامه skyp را که یک برنامه موفق تلفن اینترنتی می باشد مورد بررسی قرار خواهیم داد.

۲/۶/۱ توزیع فایل بصورت P2P

بررسی خود از مورد p2p را ..... با یک کاربرد بسیار طبیعی بنام توزیع یک فایل بزرگ از یک سرور تنها به تعداد زیادی میزبان (جفت) شروع خواهیم نمود.

فایل ممکن است نسخه جدیدی از سیستم عامل لینوکس، وصله های یک نرم افزار برای سیستم عامل و با برنامه کاربردی یک فایل mps موزیک و یا ویدئوی MPEG باشد.

در توزیع فایل به صورت مشتری سرویس دهنده، سرویس دهنده بایستی یک کپی از فایل را به هر جفت ارسال نماید، که این عمل باعث بار زیادی بر روی سرویس دهنده خواهد شد و نیاز به مقدار زیادی پهنای باند برای آن می باشد.

اما در توزیع فایل P2P هر جفت می تواند هر بخشی از فایل را که تا کنون دریافت کرد. توزیع مجدد نماید و به این ترتیب سرویس دهنده را در فرایند توزیع فایل، بیت تورنت می باشد. طبق برخی برآوردها، حدود ۳۰٪ ترافیک ستون فقرات اینترنت توسط آن استفاده می شود. این برنامه در ابتدا توسط برام کوهن توسعه داده شد (بخش مصاحبه با او در انتهای فصل را ببینید) اما برنامه مشتری بیت تورنت مستقل مختلفی را که مطابق وب که همگی مطابق با پروتکل HTTP کار می کنند، موجود می باشد. در این بخش، ما ابتدا مورد مقیاس پذیری معماری P2P در زمینه توزیع فایل را مورد بررسی قرار خواهیم داد و سپس بین تورنت را با برخی جزئیات و ویژگی ها و خصوصیات برجسته آن بررسی خواهیم داد.

مقیاس پذیری معماری P2P

برای مقایسه معماری مشتری سرویس دهنده با معماری نظیر به نظیر و نمایش ذاتی بودن خود مقیاس پذیری در P2P، هم اکنون ما یک مدل ساده "مقداری" برای توزیع یک فایل به یک مجموعه تابشی از جفت ها برای هر دو معماری توجه خواهیم نمود.

همانطوری که در شکل ۲/۲۴ نشان داده شده است، سرویس دهنده و جفت ها از طریق لینک های دسترسی به اینترنت منتقل شده اند. نرخ آپلود لینک دسترسی سرویس دهنده را با US، نرخ آپلودنا امین جفت با UI و نرخ دانلود ناامین جفت را با DI نمایش می دهد. همچنین اندازه فایل توزیعی (برچسب بیت) F و تعداد جفت ها که خواهان دریافت یک کپی از فایل هستند، N می باشد.

"زمان توزیع" ، مقدار زمانی که فول می کشد تا یک کپی از فایل به تمام جفت ها برسد. در تحلیل زمان توزیع که در ادامه آمده است، برای هر دو معماری ، ما فرضیات ساده سازیی کرده ایم که هسته اینترنت دامای پهنای باند کافی بوده، به این معنی که تمام

wWw.it98.ir

www.it98.ir

گلوگاهها فقط در بخش دسترسی شبکه است (و نه هسته شبکه). همچنین فرض کرده ایم که هم سرویس دهنده و هم مشتری ها در هیچ کاربرد شبکه ای دیگری در کار نبوده و تمام پهنای باند آپلود و دنلود فتوا برای توزیع این فایل اختصاص دارد.

اجازه دهید ابتدا زمان توزیع را برای معماری مشتری به سرویس دهنده بدست اوریم که آن را با Dcs نشان می دهد. در این معماری هیچ کدام از جفت ها در توزیع فایل کمکی نمی کند و موارد زیر قابل مشاهده هستند :

- سرویس دهنده بایستی یک کپی از فایل را به هر کدام از N جفت ارسال بدارد. بنابراین سرویس دهنده بایستی NF بیت را ارسال نماید. از آنجایی که نرخ آپلود آن US است زمان لازم برای توزیع فایل حداقل NF/ US می باشد.

- فرض کینم D min نرخ دانلود جفتی با کمترین نرخ دانلود باشد، یعنی

D min = min $\{d//.-/dn\}$ جفتی با این نرخ دانلود نمی تواند تمام F بیت را در کمتر از F/Dmin دانلود نماید.

بنابراین کمترین زمان توزیع، حداقل F/Dmin می باشد.

با ترکیب در مورد بالا، خواهیم دانست :

$$Dcs \geq Max\left\{\frac{NF}{US} / \frac{F}{D\min}\right\}$$

این رابطه محدوده پایین را برای کمترین زمان توزیع نشان می دهد.

در تمرینات شما نشان خواهید داد که سرویس دهنده می تواند با زمان بندی ارسال خود، به این حد پایین دست یابد. حال اجازه دهید این حد پایین را به عنوان زمان توزیع واقعی در نظر بگیریم یعنی :

$$Dsc = Max\left\{\frac{NF}{US} / \frac{F}{D\min}\right\} (2/1)$$

از معادله ۲/۱ برای N های به اندازه کافی بزرگ کرد، زمان توزیع توسط رابطه $\frac{NF}{US}$ به دست خواهد آمد. بنابراین زمان و توزیع بطور خطی با تعداد جفت ها اضافه می شود. برای مثال اگر تعداد جفت ها از هفته ای به هفته دیگر هزار برابر بشود (از هزار به میلیون برسد)، زمان توزیع فایل به تمام جفت با ضریب ۱۰۰۰ اضافه خواهد شد.

حال اجازه دهید آنالیزه مشابه ای را برای معماری PZP داشته باشیم. در این معماری هر جفت می تواند سرویس دهنده را در توزیع فایل یاری نماید، به ویژه زمانی که یک جفت بخشی از فایل را دریافت می نماید می تواند از ظرفیت آپلود خود برای توزیع مجدد فایل به جفت های دیگر استفاده نماید. در این حالت به هر حال محاسبه زمان توزیع به نوعی نسبت به حالت معماری مشتری سرویس دهنده پیچیده تر می باشد زیرا در حالت PZP، زمان توزیع به اینکه هر جفت چگونه در توزیع بخشی از فایل را که در اختیار دارد کمک کند، بستگی دارد. با این وجود می توان یک عبارت ساده برای کمترین زمان توزیع بدست آورد.

برای پایان دادن به این موضوع، ما موارد زیر را خواهیم دید :

- در شروع توزیع، فتوا سرویس دهنده را دارد. برای بدست آوردن فایل توسط اجتماع جفت ها، سرویس دهنده بایستی هر بیت فایل را حداقل یکبار به لینک دسترسی خود آپلود نماید. بنابراین کمترین زمان توزیع حداقل $\frac{F}{US}$ است. (بر خلاف طرح مشتری

سرویس دهنده در PZP : یک بیت ارسال شده توسط سرویس دهنده، دیگر نیازی به ارسال مجدد ندارد، زیرا توزیع مجدد ندارد، زیرا توزیع مجدد بیت جفت ها نمی توان توسط خود جفت ها انجام شود)

- همانند معماری مشتری سرویس دهنده. ۱ جفتی جفتی با پایین ترین نرخ دانلود نمی تواند تمام F بیت را در کمتر از F/Dmin ثانیه بدست آورد. بنابراین کمترین زمان توزیع حداقل F/Dmin می باشد.

www.it98.ir

www.it98.ir

- و در نهایت مشاهده می کنیم که ظرفیت کل آپلود سیستم برابر با مجموع نرخ دانلود تمام جفت ها به علاوه سرویس دهنده می باشد. U+ta=Us+U1+ … Un سیستم بایستی f بیت را به تمام N جفت تحویل دهد. بنابراین بایستی NF بیت تحویل داده شود. این عمل نمی تواند با نرخی سریع تر از U+tal انجام شود.

بنابراین کمترین زمان توزیع حداقل $\frac{NF}{Us+U1+...+Un}$ می باشد.

با ترکیب موارد فوق کمترین زمان توزیع در pzp نشان می دهد، خواهیم دانست :

$$(2/2)Dpzp \geq Max\left\{\frac{F}{US}/\frac{F}{D\min}/\frac{NF}{Us+\sum_{i=1}^{n}ui}\right\}$$

رابطه ۲/۲ حد پایین را برای کمترین زمان توزیع ارائه می دهد.

اگر تصو کنیم که هر وقت می تواند به محض دریافت هر بین آن را توزیع مجدد نماید. آنگاه یک طرح توزیعی وجود دارد که واقعا به این حد پایین خواهیم رسید. در واقعیت بجای اینکه جفت ها هر بیت دریافتی را توزیع مجدد کنند، یک قطعه را بعد از دریافت، توزیع می کنند.

با این احتساب، حد پایین را بطه ۲/۲ را به صورت زمان توزیع واقعی در در نظر می گیریم :

$$Dpzo = Max\left\{\frac{F}{US},\frac{F}{D\min},\frac{NF}{Us+\sum_{i=1}^{n}ui}\right\}(2/3)$$

شکل ۲/۲۵ کمترین زمان پاسخ برای معیارهای مشتری سرویس دهنده و PZP ا با هم مقایسه می نماید. در این شکل فرض شده است که تمام جفت ها دارای نرخ آپلود U می باشند. در این شکل ما ساعت ۱= $\frac{F}{U}$ و Dmin≥ Us /Us=IOU در نظر گرفته ایم. بنابراین یک جفت می تواند تمام فایل را در یک ساعت منتقل کند و سرور ۱۰ بار سریع تر این کار را انجام دهد و برای سادگی فرض شده است که جفت ها دارای نرخ دانلود بالایی هستند که تاثیرگذار روی سیستم نیستند.

از شکل ۲/۲۵ مشخص می شود که برای معماری مشتری- سرویس دهنده از زمان توزیع بطور خطی و بدون هیچ محدودیتی با تعدا جفت ها افزایش می یابد، اما برای معماری PZP کمترین زمان پاسخ نه تنها همواره پایین تر است (نسبت به مدل مشتری- سرویس دهنده) بلکه همواره هر تعداد جفت (N) نیز کمتر از یک ساعت می باشد.

بنابراین کاربردهای با معماری PZP می توانند خود مقیاس پذیرتر باشند.

این خود مقیاس پذیری نتیجه مستقیم این موضوع است که هر جفت در توزیع مجدد بیت ها نیز سهم دارد.

Bit Torrent

بیت تورنت یک پروتکل معروف در زمینه توزیع فایل می باشد. در ادبیات بیت تورنت، مجموعه تمام جفت های شرکت کننده در توزیع یک فایل خاص، "یک تورنت" نامیده می شود.

جفت ها در یک تورنت قطعات فایل هم اندازه ای را از همدیگر دانلود می نمایند.

اندازه نوعی قطعات 256KB می باشد. زمانیکه یک جفتی به تورنت می پیوندد، حاوی هیچ قطعه ای(از فایل نیست). با گذشت زمان قطعات بیشتر و بیشتری را جمع می کند. در زمانیکه مشغول دانلود قطعه جدیدی می باشد، می تواند قطعاتی را به جفت ها یکدیگر آپلود بنماید. زمانیکه جفتی تمام فایل را به طور کامل دریافت نمود، ممکن است (به طور خودخواهانه ای) تورنت را ترک

wWw.it98.ir

www.it98.ir

کرده و یا (به طور نوع دوستانه ای) در تورنت باقی مانده و به آپلود قطعات به دیگر جفت ها ادامه دهد. همچنین هر جفتی ممکن است در هر زمان تورنت را تنها با داشتن فقط بخشی از قطعات ترک کرده و سپس بعداً دوباره به تورنت ملحق شود.

حال اجازه دهید نگاهی دقیق‌تر به نحوه کار بیت تورنت داشته باشیم. از آنجائی‌که پروتکل بیت تورنت بسیار پیچیده‌تر می‌باشد، ما در اینجا فقط مکانیزم‌های مهم آن را بررسی خواهیم نمود. و جزئیات ریز آن را در زیر فرش خواهیم گذاشت. اینکار باعث خواهد شد که بتوانیم جنگل را از میان درخت‌ها ببینیم. هر تورنت دارای یک زیر ساختی است به نام تراکر (دنبال کننده). زمانی‌که یک جفتی به تورنت می‌پیوندد، خود را با تراک ثبت کرده و به طور دوره‌ای به تراک اطلاع می‌دهد که هنوز در تورنت حضور دارد. به این ترتیب تراکر دنباله‌ای ردّ هر جفت را که در تورنت شرکت دارند را نگهداری می‌نماید. یک تورنت ممکن است حاوی صدها و یا حتی هر جفت شرکت‌کننده در آن هر لحظه‌ای از زمان باشد.

همانطوری که در شکل ۲/۲۶ نشان داده شده است، زمانیکه یک جفت جدید آلیس، به تورنت ملحق می شود، تراک زیرمجموعه ای از جفت های تورنت را (برای مثال ۵۰ عدد) را به طور تصادفی انتخاب کرده و آدرس IP آنها را به آلیس ارسال می دارد.

آلیس با پردازش این لیست، سعی در برقراری ارتباط TCP همزمان با تمام جفت های درون لیست می نماید. اجازه بدهید تمام جفت هایی را که آلیس با آنها ارتباط TCP موفق برقرار کرده "جفت های همسایه" بنامیم. (در شکل ۲/۲۶ آلیس با فقط سه همسایه نشان داده شده است در حالت طبیعی او تعداد بیشتری همسایه دارد) . با گذشت زمان برخی از این جفت ها ممکن است تورنت را ترک کرده و جفت های دیگر (خارج از لیست اولیه ۵۰ تایی) سعی در برقراری ارتباط TCP با آلیس بنماید. بنابراین جفت های همسایگی یک جفت ممکن است در طول زمان دارای نوسان باشد.

در هر لحظه ای از مان هر جفت می تواند حاوی زیر مجموعه ای از قطعات یک فایل باشد و جفت های مختلف زیرمجموعه ای مختلفی به طور دوره ای، آلیس از هر جفت همسایه خود (از طریق ارتباز TCP) برای لیست های قطعاتی که آن جفت دارد و سوال می نماید.

اگر آلیس دارای همسایه مختلف باشد، آنگاه او مالیست از قطعات را بدست خواهد آورد. با داشتن این اطلاعات آلیس درخواست هایی را (دوباره از طریق ارتباطات TCP) برای قطعاتی ارسال می دارد.

بنابراین در هر لحظه از زمان آلیس زیر مجموعه ای از قطعات را دارد و می داند کدام قطعات را همسایگانش دارند. با داشتن این اطلاعات، آلیس بایستی دو تصمیم مهم را بگیرد. اول، کدام قطعات را او ابتدا از همسایگانش طلب نماید ؟ و دوم، به کدام یک از همشایگانش که درخواست قطعاتی را از او دارند ارسال نماید ؟

در تصمیم گیری برای اینکه کدام قطعه را درخواست نماید او از تکنیکی بنام آول کمیاب ترین استفاده می کند. این ایده تعیین می کند از بین قطعاتی که او ندارد. قطعه ای که کمیاب ترین در بین همسایگان می باشد (این قطعه دارای کمترین کپی در بین همسایگان آلیس می باشد) را در ابتدا سفارش و درخواست بدهد.

به این ترتیب کمیاب ترین قطعات به سرعت توزیع مجدد شده و باعث می شوند (به طور تقریبی) تعداد کپی های قطعات در تورنت یکسان باشد.

برای تعیین اینکه به کدام درخواست او جواب دهد، بیت تورنت از یک الگوریتم تجارتی هوشمندی استفاده می نماید. ایده اصل آن است که آلیس به همسایگانی که داده های او را هم اکنون پشتیبانی می کنند، بر حسب بالاترین نرخی را که او از آنها بیت دریافت می کند را اندازه گیری کرده و چهار تا از بالاترین آنها را تعیین می کند. او سپس با ارسال قطعات به این (۴ فت با بالاترین نرخ) معامله می نماید. (قطعات را به انها ارسال می دارد). هر ۱۰ ثانیه او این نرخ ها را مجدداً محاسبه کرد و این مجموعه چهارتایی را اصلاح می نماید.

www.it98.ir

www.it98.ir

نکته مهم اینکه، او از هر ۳۰ ثانیه یکبار یک همسایه اضافی جدید را به طور تصادفی انتخاب کرد و قطعات را به او ارسال می دارد. اجازه دهید این جفت انتخاب شده را باب بنامم. از آنجایی که آلیس به باب داده ارسال می دارد، ممکن است جزو یکی از ۴ آپلود کننده باب بشود که در این حالت باب هم شروع به ارسال داده به الیس خواهد نمود. اگر نرخی که باب به آلیس می فرستد بالا باشد، ممکن است بابا تبدیل به یکی از چهار آپلود کننده برتر آلیس بشود. به عبارت دیگر هر ۳۰ ثانیه، آلیس به طور تصادفی یک شریک تجاری جدید را برخواهد گزید و شروع به معامله با او خواهد نمود و اگر هر دو آنها از این معامله راضی باشند، آنها هر کدام را در لیست چهارتایی دیگری قرار خواهند داد و به تجارب و معامله با یکدیگر ادامه خواهند داد تا اینکه جفت شریک بهتری را پیدا نمایند. تاثیر این روش آن است که جفت هایی که قابلیت نرخ آپلود سازگاری با هم دارند، همدیگر را پیدا خواهد نمود. انتخاب همسایه تصادفی هم کمک می کند جفت های جدید هم قطعات را دریافت بدارند به طوری که بتوانند گاهی هم تجارت بنمایند.

تمام جفت های همسایه دیگر به غیر از این ۵ تا (۴ تا بالاترین و یکی هم تصادفی) شوک شده هستند، به این معنی که هیچ قطعه ای را از آلیس دریافت نخواهند نمود.

در اشتراک گذاری فایل به صورت P2P یک مشکل عمومی گسواری مجانی" می‌باشد، که در آن یک جفت فایلی را از سیستم اشتراک گذاری دانلود می کند، بدون اینکه هیچ فایل را آپلود نماید. الگوریتم تجارت بیت تورنت تقریباً این مشکل سواری مجانی را رفع می نماید. به همین منظور آلیس برای دانلود بیت ها از باب با نرخ قابل قبول در یک دوره زمانی طولانی، او بایستی با همان نرخ آپلود نماید که این کار سواری مجانی را از بین خواهد برد.

بیت تورنت‌ها دارای تعداد زیاد دیگری مکانیزم‌های جالب می‌باشد که در اینجا مورد بحث قرار نخواهد گرفت از جمله تکه‌ها (قطعات کوچک)، خط لوله، انتخاب تصادفی اولین، حالت آخرین بازی و آنتی‌استابینگ می‌باشد (cohen 2003)

۲-۷: برنامه نویسی سوکت با TCP

حال که ما تعدادی برنامه‌های کاربردی شبکه مهمی را دیده‌ایم، اجازه دهید بررسی نماییم که این برنامه‌های کاربردی واقعاً چگونه نوشته می‌شوند. در این بخش ما برنامه‌های کاربردی خواهیم نوشت که از TCP استفاده می‌نمایند. در بخش بعد برنامه‌هایی خواهیم نوشت که از UDP استفاده می‌نمایند.

از بخش ۲/ ۱ به یاد داریم که بیشتر برنامه‌های شبکه شامل یک زوج برنامه می‌باشد -یک برنامه مشتری و یک برنامه سرویس‌دهنده- که بر روی دو سیستم پایانی مختلف قرار می‌گیرند. زمانی‌که این دو برنامه اجرا شوند، یک فرآیند مشتری و یک فرآیند سرویس‌دهنده ایجاد شده و این فرآیندها از طریق خواندن و نوشتن در سوکت با همدیگر ارتباط برقرار می‌نمایند. در هنگامی‌که یک برنامه شبکه نوشته می‌شود، کار اصلی توسعه‌دهنده برنامه، نوشتن کدهایی برای هر دو برنامه مشتری و سرویس‌دهنده می‌باشد.

دو نوع برنامه کاربردی شبکه موجود می‌باشد. یک نوع آن پیاده‌سازی یک پروتکل استاندارد می‌باشد که بطور مثال در یک RFC تعریف شده است. برای چنین پیاده‌سازی‌هایی سمت مشتری پروتکل FTP باشد، که در بخش ۲/۳ شرح داده شد و بطور دقیقی در RFC959 تعریف شده است. مشابهاً برنامه سرویس‌دهنده می‌تواند یک پیاده‌سازی پروتکل سرویس دهندهٔ FTP باشد که در RFC959 بطور کامل شرح داده شده است. اگر یک توسعه‌دهنده کدهای برنامه مشتری را بنویسد و بطور مستقل از او توسعه‌دهنده دیگری کدهای برنامه سرویس‌دهنده را بنویسد و هر دو قوانین ارائه شده در RFC را رعایت نمایند، آنگاه این دو برنامه قادر به همکاری و کار کردن با یکدیگر خواهند بود. در واقع بسیاری از برنامه‌های کاربردی شبکه امروزی که درگیر ارتباطات بین برنامه‌های سرویس‌دهنده و مشتری می‌باشند، توسط توسعه‌دهندگان مستقل نوشته می‌شوند. به‌عنوان مثال، مرورگر فایرفاکس که با سرویس‌دهنده وب آپاچی ارتباط برقرار می‌نمایند و یا مشتری FTP روی یک PC فایلی را به یک سرویس‌دهنده FTP لینوکس آپلود می‌نماید. زمانی‌که یک برنامه مشتری و یا سرویس‌دهنده یک پروتکل تعریف شده توسط RFC را پیاده‌سازی می‌نماید،

wWw.it98.ir

www.it98.ir

بایستی از شماره پورت اختصاص داده شد. به آن پروتکل استفاده نماید (شماره پورت‌ها بطور خلاصه در بخش ۱/ ۲ بحث شدند و بطور کامل با جزئیات بیشتر در فصل ۳ پوشانده خواهد شد).

نوع دیگر برنامه‌های کاربردی شبکه، برنامه‌های شبکه اختصاصی می‌باشد. در این حالت پروتکل لایه کاربرد استفاده شده توسط برنامه‌های مشتری/سرویس‌دهنده، نیازی به مطابقت داشتن با هر RFC موجود نیست. یک توسعه‌دهنده تنها (و یا تیم توسعه‌دهنده) هر دو برنامه مشتری و سرویس دهنده را ایجاد می‌نمایند و توسعه‌دهنده کنترل کاملی بر روی آنچه که در کد می-گذرد دارد. اما از آنجایی‌که یک پیاده سازی پروتکل دامنهٔ عمومی را ندارد توسعه‌دهندگان مستقل دیگر قادر به توسعه کد برای تعامل با برنامه نیستند. در هنگام توسعه یک کاربرد اختصاصی، توسعه‌دهنده نبایستی از شماره پورت‌های شناخته شده از RFC آمده‌اند، استفاده نماید.

در این بخش و بخش بعد ما نکات کلیدی در توسعه کاربردهای مشتری – سرویس‌دهنده را مورد بررسی قرار خواهیم داد. در حین فاز توسعه یکی از تصمیماتی که توسعه‌دهنده بایستی بگیرد این است که آیا برنامه بر روی TCP و یا UDP اجرا خواهد شد. مجدداً یادآوری می نمائیم که TCP اتصالگرا بود و یک کانال قابل اعتمادی از رشته بایت‌ها را که از طریق آن دو سیستم پایانی جریان داد را منتقل می‌کنند، فراهم می‌آورد. UDP غیر اتصالگرا بود و بسته‌های غیر وابسته‌ای از داده‌ها از یک سیستم به سیستم دیگری بدون تضمین تحویل را ارسال می‌کند.

در این بخش ما یک برنامه کاربردی مشتری ساده که بر روی TCP اجرا شد را توسعه خواهیم داد و در بخش بعد برنامه کاربردی مشتری که بر روی UDP اجرا شود را توسعه خواهیم داد. ما این برنامه‌های سادهٔ TCP و UDP را به زبان جاوا ارائه خواهیم داد. ما می‌توانستیم این کدها را به زبان C یا C++ بنوییسم اما، ما جاوا را به دلیل اینکه برنامه‌ها آراسته‌تر و واضح‌تر می‌باشند را انتخاب کرده‌ایم. با جاوا تعداد کمتری خط کد خواهیم داشت و هر خط کد می‌تواند بدون سختی زیاد برای یک برنامه‌نویس مبتدی توضیح داده شود. دوماً برنامه‌نویسی مشتری/سرویس‌دهنده در جاوا به سرعت در حال گسترش است و امکان دارد حتی برنامه و قانونی اصلی در سال‌های آینده شود. اما اگر شما با جاوا آشنا نیستید، جای نگرانی نیست. اگر شما تجربه برنامه نویسی با زبان‌های دیگر را دارید، می‌توانید به راحتی کدها را دنبال نمایید.

برای خوانندگانی که علاقه‌مند به برنامه‌نویسی مشتری/سرویس‌دهنده به C می‌باشند، مراجع بسیار خوبی در دسترس می‌باشد. [Donahoo 2000; Steven 1997; Frost 1994; Kurose 1996]

۲,۷,۱برنامه‌نویسی سوکت با TCP

از بخش ۱/ ۲ بیاد می‌آوریم که فرآیندهایی که روی ماشین‌های مختلف اجرا می‌شوند از طریق ارسال پیام به سوکت‌ها با همدیگر ارتباط برقرار می‌نمایند. ما گفتیم که هر فرآیند مشابه یک خانه است و سوکت فرآیند مشابه درب می‌باشد. همانطوری‌که در **شکل ۲/۲۶** نشان داده شده است، سوکت دری است بین فرآیند کاربردی و TCP. توسعه‌دهنده برنامه کاربردی، کنترل کاملی بر روی سمت لایه کاربرد سوکت دارد. اما کنترل بسیار ناچیزی بر روی سمت لایه انتقال می‌تواند داشته باشد (توسعه‌دهنده حداکثر قابلیت تنظیم چند پارامتر TCP نظیر حداکثر مقدار بافر و حداکثر اندازه قطعه را دارا می‌باشد).

www.it98.ir

www.it98.ir

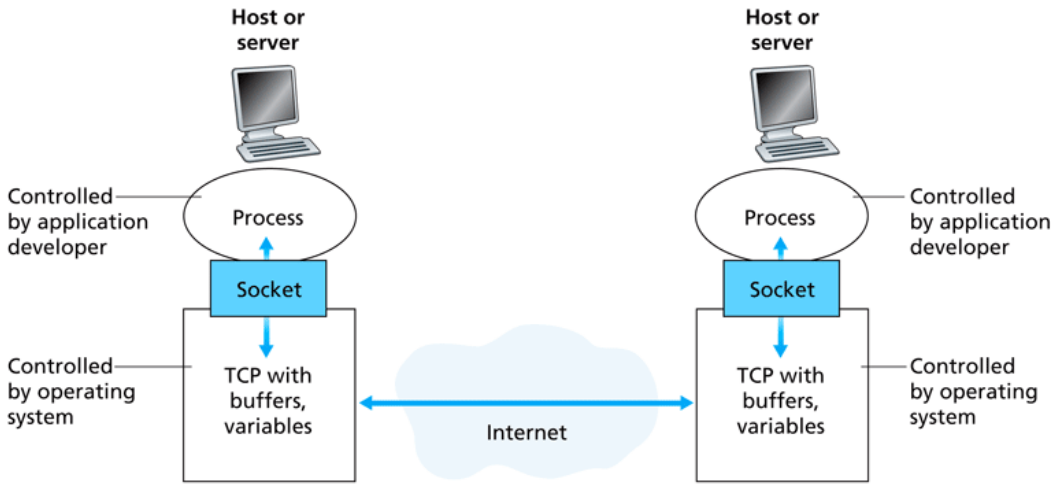


Figure 2.3 ♦ Application processes, sockets, and underlying transport protocol

حال اجازه دهید نگاهی دقیق‌تر به تعامل بین برنامه‌های مشتری و سرویس‌دهنده داشته باشیم. مشتری وظیفه شروع ارتباط با سرویس‌دهنده را دارد. برای اینکه سرویس‌دهنده قادر به انجام عکس‌العمل به تماس شروع مشتری داشته باشد، سرویس‌دهنده می‌بایستی آماده باشد. این موضوع شامل دو نکته می‌باشد، اول اینکه سرور نمی‌تواند غیر فعال و خوابیده باشد، یعنی قبل از اینکه یک مشتری سعی در شروع به برقراری ارتباط با آن بنماید بایستی به‌عنوان یک فرآیند اجرا شده باشد. دوم اینکه برنامه سرویس‌دهنده بایستی دارای نوع خاصی از درب -دقیق تر یک سوکت- باشد که به تماس‌های برقرار شده از طرف فرآیند مشتری که روی هر میزبان دلخواهی هست، خوش‌آمد گوید. با استفاده از قیاس خانه/درب برای فرآیند/سوکت، ما گاهی اوقات تماس شروع مشتری را به‌عنوان "ضربه زدن به درب خوش‌آمد" تعبیر می‌نماییم.

اگر فرآیند سرویس‌دهنده در حال اجرا باشد، فرآیند مشتری می‌تواند یک ارتباط TCP را با سرویس‌دهنده شروع نماید. این عمل در برنامه مشتری با ایجاد یک سوکت انجام می‌پذیرد. زمانی‌که مشتری سوکت خودش را ایجاد می‌کند، آدرس پروسس سرور، با نام آدرس IP میزبان سرویس‌دهنده و شماره پورت فرآیند سرویس‌دهنده را تعیین می‌نماید. به محض ایجاد شدن سوکت در برنامه مشتری، TCP در مشتری یک دست‌تکانی سه مرحله‌ای را شروع کرده و یک ارتباط TCP را با سرویس‌دهنده برقرار می‌نماید، که این عمل در لایه انتقال انجام شده و برای مشتری-سرویس دهنده کاملاً ناپیدا می‌باشد. در حین عمل دست‌تکانی سه مرحله‌ای، فرآیند مشتری بر درب خوش‌آمد سرویس‌دهنده ضربه می‌زند. زمانی‌که سرویس‌دهنده این ضربه زدن را می‌شنود، یک درب جدید – به‌عبارت دقیق‌تریک سوکت- که به این مشتری خاص اختصاص دارد را باز می‌نماید. در ادامه ما در مثال، درب خوش‌آمد یک شیء Server Socket است که ما آن‌را WelcomeSocket می‌نماییم. زمانی‌که یک مشتری بر این درب می‌کوبد، برنامه متد Accept() از WelcomeSocket را فراخوانی می‌کند که یک درب جدید برای مشتری ایجاد می‌نماید. در پایان فاز دست‌تکانی، یک ارتباط TCP بین سوکت مشتری و سوکت سرور موجود خواهد بود. از این رو، ما به این سوکت اختصاص جدید سرویس‌دهنده، "سوکت ارتباطی سرویس دهنده" می‌گوییم.

از نظر برنامه کاربردی، ارتباط TCP یک لوله مجازی مستقیم بین سوکت مشتری و سوکت ارتباطی سرویس‌دهنده می‌باشد. فرآیند مشتری می‌تواند بایت‌ها را به دلخواه خود به درون سوکت خودش ارسال داشته و TCP تضمین می‌نماید که فرآیند سرویس‌دهنده هر بایت را به همان ترتیب که ارسال شده، دریافت خواهد داشت (از طریق سوکت ارتباطی). بنابراین TCP یک سرویس "جریان بالینی قابل اطمینان" بین فرآیندهای مشتری و سرویس‌دهنده بوجود می‌آورد. علاوه‌بر آن، همان‌طور که مردم می‌توانند از یک درب هم وارد و هم خارج شوند، فرآیند مشتری هم نه تنها بایت‌ها را ارسال بلکه می‌تواند آنها را از سوکت دریافت هم

wWw.it98.ir

www.it98.ir

بدارد. سرویس‌دهنده هم همچنین نه تنها دریافت، بلکه می‌تواند بایت‌ها را به سوکت ارتباطی خود ارسال هم بدارد. این موضوع در شکل ۲/۲۷ نشان داده شده است. شکل ۲/۲۷- سوکت خوش‌آمد و سوکت ارتباطی را نشان می‌دهد. از آنجایی‌که سوکت‌ها نقش محدودی در کاربردهای مشتری/سرویس‌دهنده دارند، به همین رو به توسعه کاربردهای مشتری/سرویس‌دهنده برنامه‌نویسی سوکت نیز گفته می‌شود.

قبل از ارائه مثال‌مان درباره کاربرد مشتری/سرویس‌دهنده، خوب است در مورد اصلاح جریان (استریم) صحبتی داشته باشیم. یک استریم، دنباله‌ای از کاراکترها است که به درون و یا خارج از فرآیند جریان دارند. هر استریم برای یک فرآیند یا "استریم ورودی" است یا "استریم خروجی". اگر استریم ورودی باشد آنها برای فرآیند به برخی منابع ورودی (نظیر صفحه کلید) و یا یک سوکت که داده ها از اینترنت به درون آن جریان می یابد، متصل شده است. اگر سوکت خروجی باشد،

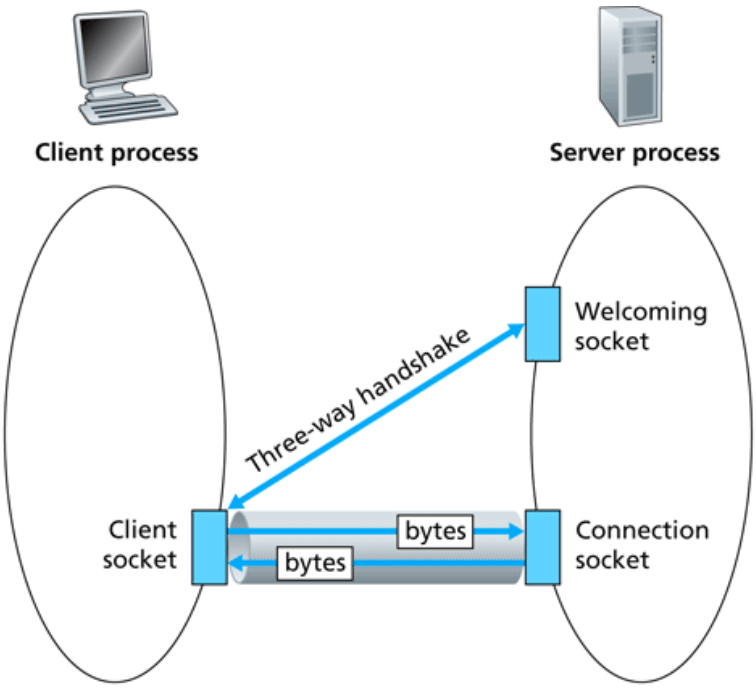


Figure 2.31 ♦ Client-socket, welcoming socket, and connection socket

آنگاه برای فرآیند به برخی منابع خروجی متصل است. نظیر خروجی استاندارد (مانیتور) و یا یک سوکت خروجی که داده‌ها را به درون اینترنت ارسال می‌دارد.

۲,۷,۲ مثالی از کاربرد مشتری-سرویس‌دهنده در جاوا:

ما از برنامه کاربردی مشتری-سرویس‌دهنده ساده -زیرا برای نمایش برنامه نویسی سوکت برای TCP و UDP- استفاده خواهیم نمود.

۱.مشتری یک خط از ورودی استاندارد (صفحه کلید) خوانده و آنرا به درون سوکت خودش به سمت سرور ارسال می‌دارد.

۲.سرویس‌دهنده یک خط را از سوکت ارتباطی خود می‌خواند.

۳.سرویس‌دهنده خط را به حالت بزرگ (حروف کوچک به بزرگ) تبدیل می‌نماید (پردازش).

۴.سرویس‌دهنده خط پردازش شده را به سمت سوکت ارتباطی خود برای مشتری ارسال می‌دارد.

۵.مشتری این خط پردازش شده را از سوکت خود خوانده و خط را در خروجی استاندارد خود (مانیتور) چاپ می‌نماید.

www.it98.ir

www.it98.ir

شکل ۲,۲۸ فعالیت‌های اصلی مرتبط با سوکت را برای مشتری و سرویس‌دهنده نشان می‌دهد.

سپس ما جفت برنامه برای مشتری-سرویس‌دهنده با پیاده‌سازی TCP ارائه خواهیم داد. و جزئیات آنرا خط به خط تحلیل خواهیم نمود. برنامه مشتری تحت نام TCPClient.java و برنامه سرویس‌دهنده تحت نام TCPServer.java می‌باشد.

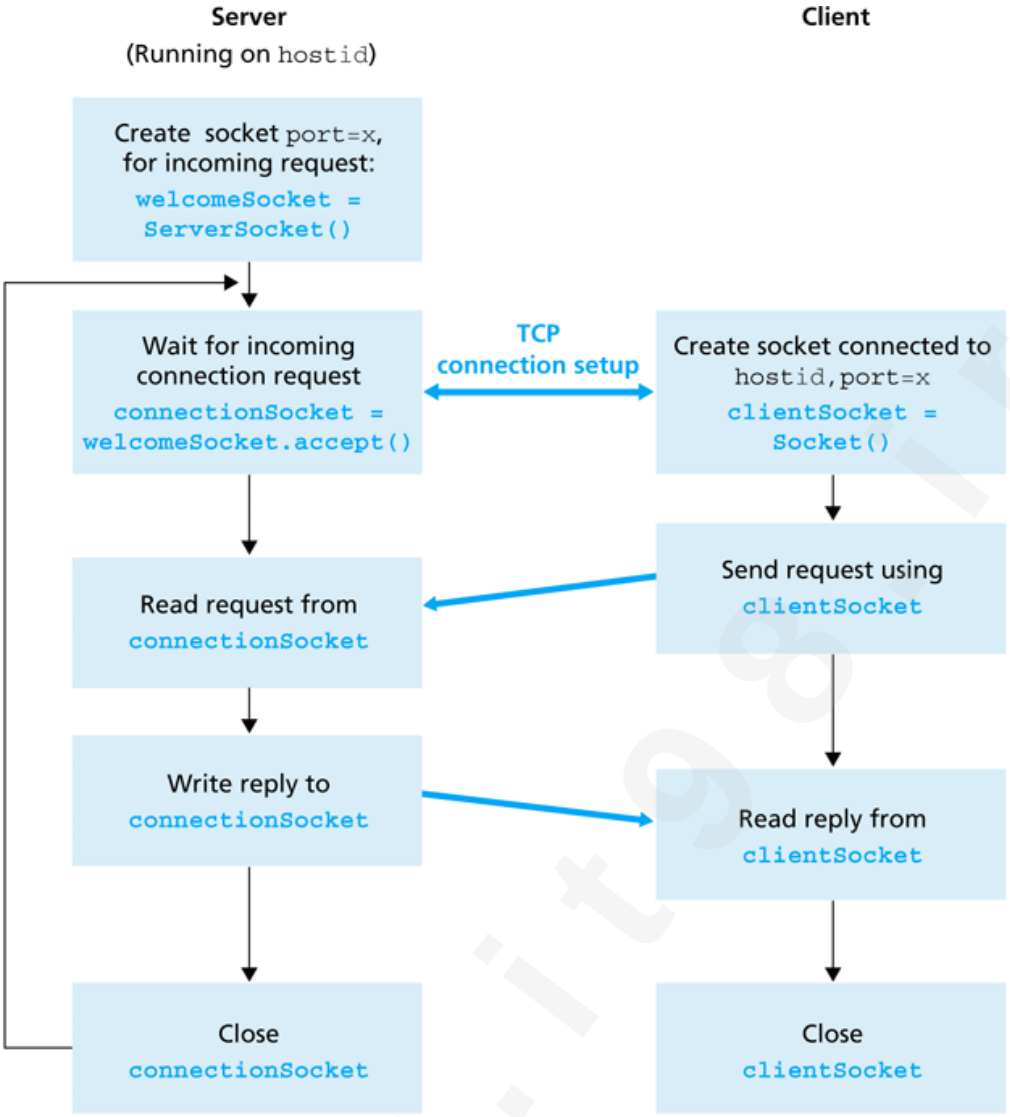


Figure 2.32 ♦ The client-server application, using connection-oriented transport services

به منظور تأکید روی جنبه‌های کلیدی، ما عمداً کدی ارائه داده‌ایم که به آنها اشاره داشته باشد. اما ممکن است برنامه جامعی نباشد. البته یک کد خوب بایستی یکسری خط‌های کمکی بیشتری داشته باشد.

بعد از آنکه هر دو برنامه (مشتری و سرویس‌دهنده) بر روی میزبان‌های خود کامپایل شدند، ابتدا برنامه سرویس‌دهنده بر روی میزبان سرویس‌دهنده اجرا شده و باعث ایجاد یک فرآیند سروری در آن خواهد شد. همانطور که در بالا اشاره کردیم فرآیند سرویس‌دهنده منتظر تماس گرفته شدن توسط یا فرآیند مشتری می‌باشد. در این برنامه مورد مثال ما، زمانیکه برنامه مشتری اجرا شود، فرآیندی در مشتری ایجاد شده و این فرآیند بلافاصله با سرویس‌دهنده تماس گرفته و یک ارتباط TCP را با آن برقرار می-نماید. سپس کاربر در مشتری می‌تواند با استفاده از این برنامه یک خط ارسال داشته و نسخه حالت بزرگ حروف را (از سمت سرویس دهنده) دریافت نماید.

TCPClient.java

wWw.it98.ir

www.it98.ir

در اینجا یک کد برای سمت مشتری کاربرد قرار دارد.

```
Import java.io.*;
Import java.net.*;
Class TCPClient {
Public static void main (string argv[])throws Exception
{
        String sentence;
String modified sentence;
Bufferedreader inFromUser=new Bufferedreader(
        New InputStreamReader(system.in));
Socket ClientSocket=new Socket (“hostname”,6789);
DataOutputStream outToServer = new outputStream(
        clientSocket.getInputStream()));
Bufferedreader inFromUser=
        new Bufferedreader(new InputStreamReader(
clientSocket.getInputStream()));
sentence = inFromUser.readLine();
outToServer.WriteBytes (sentence+ ‘\n’);
modifiedsentence= inFromServer.readLine();
system.out.println (“From SERVER: “+
modifiedsentence);
        ClientSocket.Close();
}
}
```

برنامه مشتری TCPClient سه استریم و یک سوکت را همانطور که در شکل ۲/۲۹ نشان داده شده است ا ایجاد می‌نماید. نام سوکت ClientSocket می‌باشد. استریم inFromUser یک استریم ورودی برنامه بوده که به ورودی استاندارد (که صفحه کلید است) متصل می‌شود. زمانی‌که کاربر توسط صفحه‌کلید کاراکترهایی را تایپ می‌نماید، کاراکترها به درون این استریم جریان می‌یابند. استریم inFromServer نیز استریم ورودی دیگری است که به سوکت متصل شده است. کاراکترهایی که از طرف شبکه به داخل جریان می‌یابند، وارد این استریم می‌شوند. در نهایت OutToServer یک استریم خروجی است که همچنین به سوکت متصل شده است. کاراکترهایی که مشتری به سمت شبکه ارسال می‌دارد به درون این استریم جریان می‌یابند.

حال اجازه بدهید نگاهی به خط‌های مختلف در کد داشته باشیم.

```
Import java.io.*;
Import java.net.*;
```

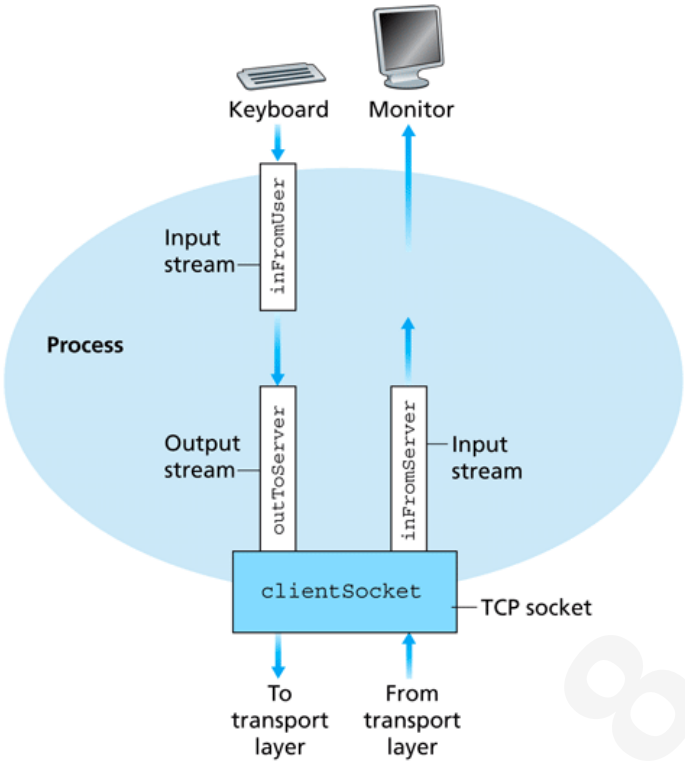
Java.io و java.net، پکیج‌های جاوا می‌باشند. Java.io شامل کلاس‌هایی برای استریم‌های ورودی و خروجی می‌باشد. بطور خاص بستهٔ java.io شامل کلاس‌های Buffered Reader و Data Output Stream می‌باشد، کلاس‌هایی که برنامه از آنها برای ایجاد سه استریمی که قبلاً صحبت کردیم، استفاده می‌نماید. بستهٔ java.net شامل کلاس‌هایی برای پشتیبانی از شبکه می-باشد. بویژه شامل Socket و ServerSocket می‌باشد. شیء ClientSocket دز این برنامه از کلاس Socket مشتق شده است.

```
Class TCPClient {
```


www.it98.ir

www.it98.ir

Public Static Void Main (String argv[]) Throws Exeption
{.....}



}**Figure 2.33** ♦ TCPclient has three streams through which characters flow

کدهای بالا کدهای استانداردی است که در شروع هر برنامه جاوا دیده می‌شود. خط سوم، شروع بلوک تعریف یک کلاس می‌باشد. کلمه کلیدی کلاس Class، تعریف کلاس را برای کلاس با نام TCPClient شروع می‌نماید. یک کلاس شامل متغیرها و متدها می‌باشد. متغیرها و متدهای کلاس توسط براکت‌های شروع و پایان تعریف کلاس محصور می‌شوند. کلاس TCPClient حاوی هیچ متغیر کلاسی نبود و دقیقاً شامل یک متد – متد main() – می‌باشد. متدها مشابه توابع و پروسیجرها در زبان‌های دیگر نظیر C می‌باشند، متد main() در زبان جاوا مشابه تابع main() در C و C++ می‌باشد.زمانی‌که مفسر جاوا یک برنامه را اجرا می‌نماید، با فراخوانی متد main() کلاس شروع می‌شود. آنگاه متد main() تمام متدهای دیگر مورد نیاز برای اجرای برنامه را فراخوانی می-نماید.برای این برنامه مقدماتی برنامه‌نویسی سوکت در جاوا، می‌توانید از کلمات کلیدی Public، Static، Void و Throws Exception عبور کنید (اگرچه کد بایستی شامل آنها باشد).

String Sentecne;
String modified Sentece;

این دو خط بالا، اشیایی از نوع String را اعلان می‌دارند. شیء Sentence نوع رشته‌ای است که توسط کاربر و برای ارسال به سمت سرویس‌دهنده کاربر دارد. شیء modified Sentence، رشته‌ای است که از طرف سرویس‌دهنده دریافت شده و به سمت خروجی استاندارد کاربر ارسال می‌شود.

Buffered Reader inFromUser=new Buffered Reader (new InputStreamReader(System.in));

این خط کد، شیء استریم inFromUser را با نوع Buffered Reader ایجاد می‌نماید و استریم ورودی توسط System.in به ورودی استاندارد متصل می‌شود.

wWw.it98.ir

www.it98.ir

```
Socket ClientSocket=new Socket("hostname",6789);
```

این خط کد شیء ClientSocket از نوع Socket را ایجاد می‌نماید. همچنین این دستور یک ارتباط TCP بین مشتری و سرویس‌دهنده را شروع می‌نماید. رشته "hostname" بایستی با نام میزبان سرویس‌دهنده (برای مثال "apple.poly.edu") جایگزین شود. قبل از اینکه ارتباط TCP واقعاً برقرار شود، ابتدا مشتری بایستی یک عمل DNS برای بدست آوردن IP میزبان انجام بدهد. شماره ۶۷۸۹، شماره پورت می‌باشد. شما می‌توانید از شماره پورت دیگری استفاده نمایید. همانطور که قبلاً هم بحث شد، آدرس IP میزبان همراه با شماره پورت برنامه کاربردی، مشخص کننده فرآیند سرویس‌دهنده می‌باشد.

www.it98.ir

www.it98.ir

```
Data Output Stream OuttoServer =
        New Data OutStream(ClientSocket.getOutputStream());
Buffered Reader inFromServer=
        New Buffered Reader(new
 inputStreamReader(ClientSocket.getInputStream());
```

دو خط بالا، اشیاء استریم برای اتصال به سوکت را ایجاد می‌کنند.رشتهٔ Out_ToServer فرآیند خروجی برای سوکت را فراهم می‌کند.استریم inFromServer برای فرآیند ورودی از سوکت را فراهم می‌آورد (شکل ۲/۲۹ را ببینید).

```
Sentence=inFromUser.ReadLine();
```

این خط دستور، یک خط تایپ شده توسط کاربر را درون رشته Sentence قرار می‌دهد. رشتهٔ Sentence کاراکترها را تا رسیدن به کاراکتر CR (Cariage Return) دریافت می‌دارد. این خط از طریق استریم inFromUser به رشتهٔ Sentence وارد می‌شود.

```
OutToServer.writeBytes(Sentence+'\n');
```

این خط رشتهٔ Sentence که با CR اضافه شده است را به استریم OutToServer ارسال می‌دارد. Sentence تکمیل شده با CR از طریق سوکت مشتری عبور کرد و به لوله TCP وارد می‌شود. مشتری سپس منتظر دریافت کاراکترها از طریق سرویس‌دهنده می‌ماند.

```
Modified Sentence=inFromServer.ReadLine();
```

وقتی‌که کاراکترها از سرویس‌دهنده می‌رسند، آنها از طریق استریم inFromServer جریان یافته و درون رشتهٔ modified Sentente قرار می‌گیرد. کاراکتر تا رسیدن به پایان خط با کاراکتر CR درون modified Sentence جمع می‌شوند.

```
System.out.println("From SERVER"+modifiedSentence);
```

خط بالا رشتهٔ modifiedSentence برگردانده شده از طرف سرویس‌دهنده را بر روی مانیتور چاپ می‌نماید.

```
ClientSocket.close();
```

آخرین خط کد بوده و سوکت را می‌بندد و بنابراین ارتباط TCP بین مشتری و سرویس دهنده را پایان می‌دهد. این دستور باعث می‌شود که مشتری یک پیام TCP به TCP سرویس‌دهنده ارسال نماید

TCPServer.java

حالا بیایید نگاهی به کد سرویس‌دهنده بیندازیم،

wWw.it98.ir

www.it98.ir

```
import java.io.*;
import java.net.*;
class TCPServer {
   public static void main(String argv[]) throws Exception
      {
         String clientSentence;
         String capitalizedSentence;
         ServerSocket welcomeSocket = new ServerSocket
            (6789);
         while(true) {
            Socket connectionSocket = welcomeSocket.
               accept();
            BufferedReader inFromClient =
               new BufferedReader(new InputStreamReader(
                  connectionSocket.getInputStream()));
            DataOutputStream outToClient =
               new DataOutputStream(
                  connectionSocket.getOutputStream());
            clientSentence = inFromClient.readLine();
            capitalizedSentence =
                  clientSentence.toUpperCase() + '\n';
            outToClient.writeBytes(capitalizedSentence);
         }
      }
}
```

TCPServer شباهت‌های بسیاری به TCPClient دارد. اجازه دهید نگاهی به خط‌های درون TCPServer.java بیندازیم. خطوطی را که همان و یا مشابه با دستورات TCPClient.java باشند را توضیح نخواهیم داد.

اولین خط در TCPServer اساساً با آنچه که در TCPClient دیدیم متفاوت می‌باشد.

ServerSocket welcomeSocket=new ServerSocket(6789);

این خط باعث ایجاد شیء WelcomeSocket می‌شود، که از نوع ServerSocket می‌باشد.

WelcomeSocket نوعی درب است که منتظر شنیدن ضربه‌ای از طرف مشتری می‌باشد. این welcomeSocket به پورت شماره ۶۷۸۹ گوش فرا می‌دهد. خط بعد به‌صورت زیر است:

Socket connectionSocket=welcomeSocket.accept();

این خط زمانی‌که یک مشتری به welcomeSocket ضربه می‌زند، یک سوکت به نام connectionSocket ایجاد می‌نماید (در فصل ۳ بیان خواهیم کرد که چرا هر دو سوکت بایستی دارای شماره یکسان باشند). سپس TCP یک ارتباط خط لوله‌ای مجازی بین ClientSocket در سمت مشتری با connectionSocket در سمت سرور برقرار می‌نماید. سپس مشتری و سرویس‌دهنده قادر خواهند بود از طریق این لوله بایت ها را ارسال و دریافت نمایند و تمام بایت در طرف دیگر به همان ترتیب که ارسال شده بودند، دریافت می‌شوند. با connectionSocket برقرار شده سرویس‌دهنده می‌تواند به دریافت درخواست‌های دیگر از طرف مشتریان با اسافاده از welcomeSocket ادامه دهد (البته این نسخه از برنامه ارائه شده واقعاً به بیشتر از یک درخواست در آن واحد گوش فرا می‌دهد، اما برای انجام این کار می‌تواند با نخ‌ها اصلاح شود).سپس برنامه تعدادی شیء استریم، مشابه با اشیاء استریم در ClientSocket می‌سازد. حال به دستور زیر توجه نمایید:

CapitalizedSentence=clientSentence.toUpperCase()+'\n';

این دستور قلب برنامه کاربردی است. یک خط ارسال شده توسط مشتری را دریافت کرده، آنرا به حروف بزرگ تبدیل می‌نماید و یک کاراکتر CR به انتهای آن اضافه می‌نماید. این دستور از متد toUpperCase() استفاده می‌نماید. تمام دستورات دیگر در این برنامه جانبی و فرعی بودند و فقط برای برقراری ارتباط با مشتری مورد استفاده قرار می‌گیرند.

www.it98.ir

www.it98.ir

برای آزمایش این جفت برنامه، بایستی TCPClient.java را در یک میزبان و TCPServer.java را در میزبان دیگری نصب و کامپایل نمود. مطمئن شوید که نام صحیح میزبان "hostname" را در سرویس دهنده در TCPServer.java وارد کنید. سپس کلاس TCPServer.class، برنامه سرویس دهنده کامپایل شده را در سرویس دهنده اجرا نمایید. این عمل باعث ایجاد فرآیند سرویس دهنده شده و تا زمان تماس گرفتن توسط مشتریها بیکار باقی خواهد ماند. سپس برنامه کامپایل شده در مشتری، TCPClient.class، را اجرا نمایید. این عمل یک فرآیند در مشتری ایجاد کرده و باعث برقراری یک ارتباط TCP بین مشتری و سرویس دهنده خواهد شد. سپس با استفاده از برنامه (مشتری) جمله‌ای را تایپ نموده و در انتها CR را بزنید.

برای توسعه برنامه مشتری-سرویس دهندهٔ خودتان، می‌توانیر برنامه را با کمی تغییر اصلاح نمایید. برای مثال، به جای تبدیل تمام حروف به حالت بزرگ، سرویس دهنده می‌تواند تعداد دفعات تکرار کاراکتر s را شمارش کرده و آنرا برگرداند.

۲,۸برنامه‌نویسی با سوکت UDP:

در بخش قبل یاد گرفتیم، هنگامی‌که دو فرآیند بوسیلهٔ TCP با یکدیگر ارتباط برقرار می‌کنند، مقل این است که یک راه ارتباطی بین دو فرآیند وجود دارد. این راه ارتباطی در آن مکان تا زمانی‌که یکی از دو فرآیند بسته شود، باقی می‌ماند. هنگامی‌که یکی از فرآیندها بخواهدمقداری بایت به فرآیند دیگر ارسال کند، به سادگی بایت‌ها را وارد این راه ارتباطی می‌کند. لازم نیست که فرآیند ارسالی آدرس مقصد را ضمیمه داشته باشد زیرا راه ارتباطی به‌طور منطقی با مقصد ارتباط دارد. بعلاوه، راه ارتباطی یک کانال جریان بایت قابل اعتماد را فراهم می‌کند. سلسله بایت‌ها که بوسیلهٔ فرآیند گیرنده دریافت می‌شوند ، دقیقاً سلسله بایت‌هایی هستند که فرستنده آنها را در راه ارتباطی قرار داده است.

UDP همچنین اجازه می‌دهد دو (یا بیشتر) فرآیند بر روی میزبان‌های مختلف برای ارتباط اجرا شوند. هر چند UDP با TCP از نظر بسیاری راه‌های اساسی با هم فرق می‌کنند. اولاً، UDP یک سرویس بدون ارتباط است . فاز اولیه دست تکانی در طول مدتی که یک راه ارتباطی بین دو فرآیند برقرار شده، وجود ندارد. بخاطر اینکه UDP را ارتباطی ندارد، هنگامی‌که یک فرآیند می‌خواهد یک دسته از بایت‌ها را به فرآیند دیگری ارسال کند، فرآیند فرستادن باید آدرس فرآیند مقصد را به دستهٔ بایت‌ها ضمیمه کند و این کار باید برای هر دسته از بایت‌های فرآیند فرستادن ارسال شده، انجام شود. به عنوان یک شباهت، یک گروه ۲۰ نفری را در نظر بگیرید که ۵ تاکسی برای یک مقصد مشترک می‌گیرند؛ هنگامی‌که اشخاص سوار تاکسی می‌شوند، هر راننده تاکسی باید بطور جداگانه از مقصد مطلع شود. بدین ترتیب، UDP شبیه سرویس تاکسی است. آدرس مقصد یک محمولهٔ چندتایی از IP آدرس‌های میزبان مقصد و شماره پورت فرآیند مقصد است. ما به دسته بایت‌های اطلاعاتی همراه با IP آدرس‌های مقصد و شماره پورت، بسته (Packet) می‌گوییم. UDP یک مدل سرویس پیام جهت‌دار و غیر قابل اعتماد را فراهم می‌کند، در آن حالت بهترین تلاش را برای تحویا دادن دسته بایت‌ها به محل انجام می‌دهد اما هیچ تضمینی نمی‌کند که بسته بایت‌ها قطعاً تح.یا داده می‌شوند. سرویس UDP بطور اساسی (از چندین جنبه) با مدل سرویس جریان بایت قابل اعتماد TCP تفاوت دارد. بعد از ساخته شدن یک بسته (Packet) فرآیند فرستادن بسته را به درون شبکه از میان یک سوکت می‌فرستد. در ادامه فرضیهٔ تاکسی، از طرف دیگر سوکت فرستادن، یک تاکسی منتظر بسته است. سپس تاکسی پاکت را به آدرس مقصد می‌برد- هر چند تاکسی تضمینی نمی‌کند که نهایتاً بسته را به مقصد نهایی‌اش برساند- تاکسی ممکن است خراب شود یا حوادث غیر قابل پیش‌بینی برایش رخ دهد. به بیان دیگر، UDP یک سرویس انتقال غیر قابل اعتماد به فرآیندهای ارتباطی‌اش فراهم می‌کند.

در این بخش برنامه‌نویسی سوکت بوسیلهٔ گسترش مجدد برنامه مشابه بخش‌های قبلی‌اش را به شما نشان می‌دهیم، اما این دفعه بوسیلهٔ UDP. خواهیم دید که کد برای UDP از کد TCP در بسیاری از روش‌ها و راه‌های اصلی متفاوت است. بطور نمونه،

۱.هیچ دست‌تکانی اولیه بین دو فرآیند وجود نداردو در نتیجه نیازی به سوکت خوش‌آمد گویی نیست.

wWw.it98.ir

www.it98.ir

۲.هیچ استریمی به سوکت‌ها ضمیمه نخواهد شد.

۳.میزبان فرستادن بسته‌ها را بوسیلهٔ ضمیمه کردن IP آدرس مقصد و شماره پورت برای هر دسته بایت‌هایی که می‌فرستد، می-سازد.

۴.فرآیند گیرنده باید هر بستهٔ دریافتی را برای دستیابی به بایت‌های اطلاعاتی بسته، بشکافد.

یکبار دیگر برنامه را بطور ساده می‌گوئیم:

۱.یک مشتری خط را از روردی استاندارد خود (صفحه کلید) می‌خواند و قسمت‌های علامت‌گذاری شده خود را به سرویس‌دهنده از طریق سوکت خود ارسال می‌کند.

۲.سرویس‌دهنده یک خط را از سوکت خود می‌خواند.

۳.سرویس‌دهنده خط را به حروف بزرگ تبدیل می‌کند.

۴.سرویس‌دهنده قسمت‌های علامت‌گذاری شده تغییر یافته سوکت خود را به مشتری ارسال می‌کند.

۵. مشتری خط تغییر یافته را از سوکت خود می‌خواند و خط را بر روی خروجی استاندارد خود (مانیتور) نمایش می‌دهد.

شکل ۲,۳۰ فعالیت سوکت وابسته مشتری و سرویس‌دهنده را که ارتباط را بوسیلهٔ یک سرویس انتقال بدون ارتباط (UDP) را برقرار می‌کند را بطور خلاصه نشان می‌دهد.

UDPClient.java

در اینجا کد برای طرف مشتری برنامه قرار دارد.

```
import java.io.*;
import java.net.*;
class UDPClient {
   public static void main(String args[]) throws Exception
   {
      BufferedReader inFromUser =
         new BufferedReader(new InputStreamReader
                  (System.in));
      DatagramSocket clientSocket = new DatagramSocket();
      InetAddress IPAddress =
                  InetAddress.getByName("hostname");
      byte[] sendData = new byte[1024];
      byte[] receiveData = new byte[1024];
      String sentence = inFromUser.readLine();
      sendData = sentence.getBytes();
      DatagramPacket sendPacket =
         new DatagramPacket(sendData, sendData.length,
                  IPAddress, 9876);
      clientSocket.send(sendPacket);
      DatagramPacket receivePacket =
         new DatagramPacket(receiveData,
                  receiveData.length);
      clientSocket.receive(receivePacket);
      String modifiedSentence =
         new String(receivePacket.getData());
      System.out.println("FROM SERVER:" +
                  modifiedSentence);
      clientSocket.close();
   }
}
```

www.it98.ir

www.it98.ir

برنامهٔ UDP.Client.java یک استریم و یک سوکت را همانطور که در شکل ۲,۳۱ نشان داده شده است، بنا می‌کند.سوکت Socket Client نامیده می‌شود و مدلی از Datagram Socket است. گفتنی است که UDP یک نوع سوکت متفاوت از TCP در مشتری استفاده می‌کند. بطور نمونه، با UDP مشتری‌هایمان یک DatagramSocket اشتفاده می‌کنند، در حالیکه با TCP مشتریان یک Socket استفاده می‌کنند. استریم inFromUser یک استریم ورودی برای برنامه است.

که به ورودی استاندارد ضمیمه شده است، یعنی به صفحه کلید. ما یک استریم برابر در نسخه TCP برنامه‌مان داشتیم. هنگامی- که کاربر وهژه‌ها را بر روی صفحه کلید تایپ می‌کند، واژه‌ها بدون استریم inFromUser جریان می‌یابند. اما در مقایسه با TCP هیچ استریمی (ورودی یا خروجی) به سوکت ضمیمه نشده است. بجای تغذیه بایت‌ها برای استریم ضمیمه شده به یک Socket هدف، UDP بسته‌های شخصی را از میان هدف DatagramSocket ارسال خواهد کرد.

اکنون نگاهی به خط‌های در کد که بطور اساسی با TCPClient.java تفاوت دارند می‌اندازیم:

DatagramSocket Client Socket=new DatagramSocket();

این خط یک هدف ClientSocket نوع Datagra,Socket می‌سازد. در مقایسه با TCPClient.java یک اتصال TCP اولیه برقرا نمی‌کند. میزبان مشتری بر اساس این خط ارتباطی با میزبان سرویس دهنده برقرار نمی‌کند.

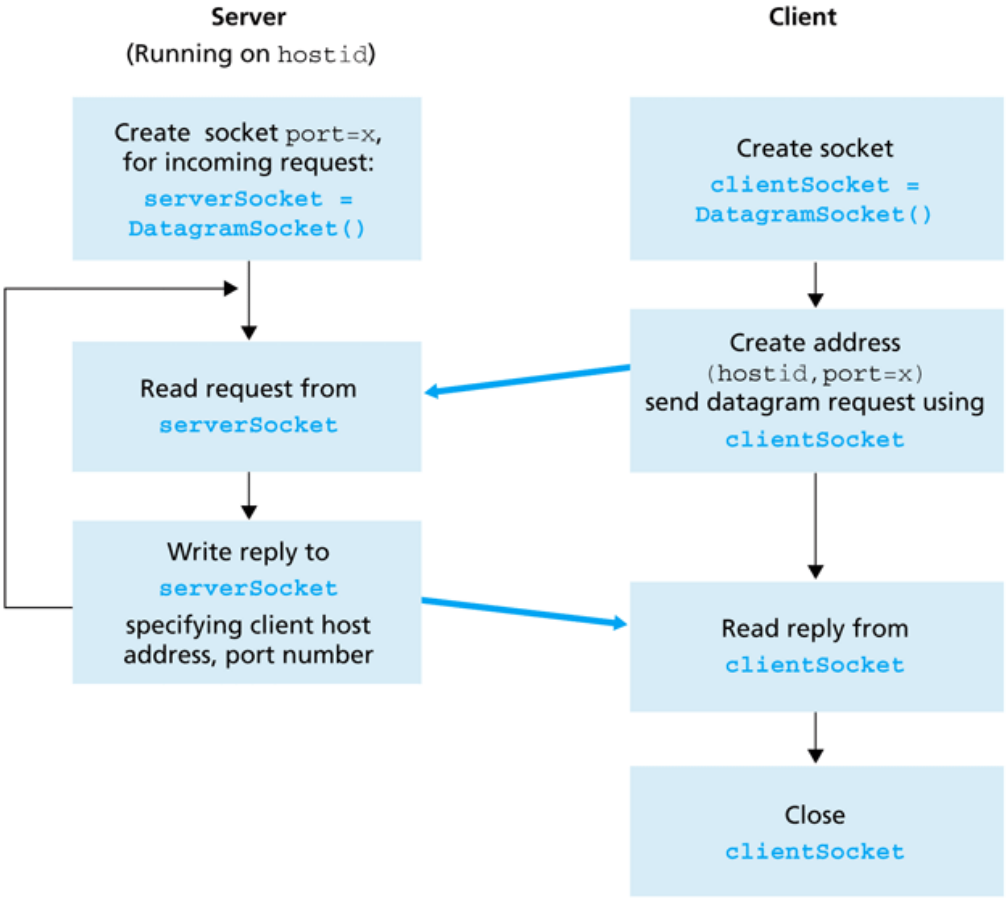


Figure 2.34 ♦ The client-server application, using connectionless transport services

به همین دلیل، همانطور که گفتیم سازنده DatagramSocket() نام میزبان سرویس‌دهنده یا شماره پورت را نمی‌برد. با استفاده از تشابه در راه ارتباطی، اجرای خط بالا یک درب برای فرآیند مشتری می‌سازد اما یک راه ارتباطی بین دو فرآیند نمی‌سازد.

InetAddress=Inet Address.get ByName(“hostname”);

www.it98.ir

برای اینکه بایت‌ها به فرآیند مقصد ارسال شوند، ما به آدرس فرآیند احتیاج داریم. قسمتی از این آدرس، IP آدرس میزبان مقصد است. خط بالا یک جستجوگر را احضار می‌کند که نام میزبان را (در این مثال، در کد بوسیلهٔ توسعه‌دهنده ذخیره شده است) به یک IP آدرس تغییر می‌دهد. DNS همچنین بوسیلهٔ نسخه TCP مشتری احضار شده است.

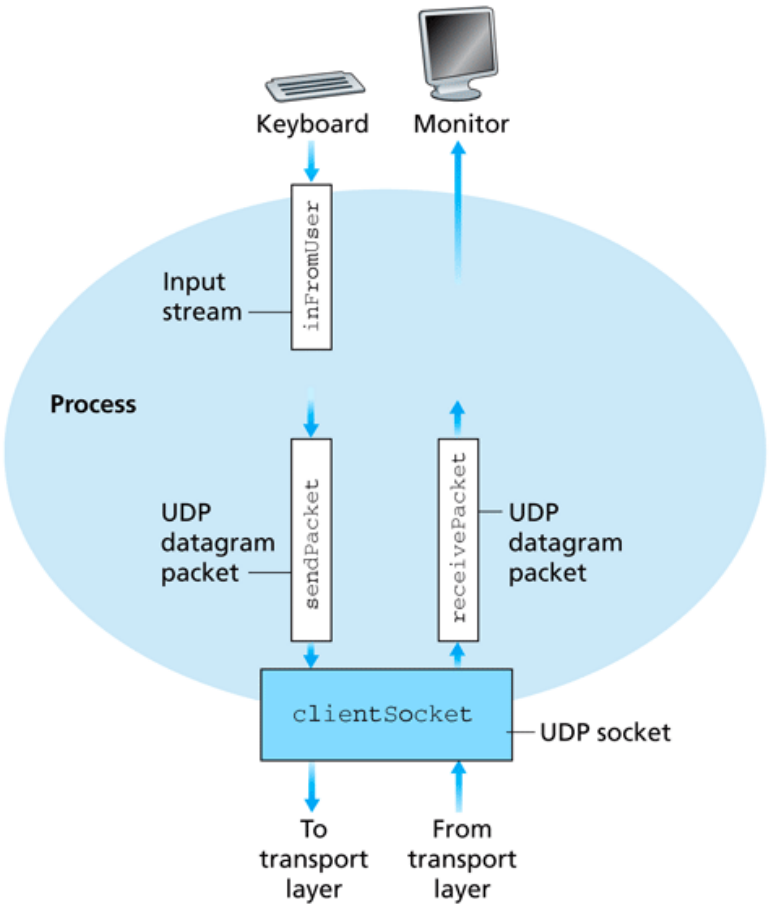


Figure 2.35 ♦ `UDPClient` has one stream; the socket accepts packets from the process and delivers packets to the process.

```
String ModifiedSentence =
        New String(Recievepacket.getData());
```

این دو خط بالا داده ها را از RecievePacket استخراج می‌کند و بایت های أرایه را اجرا و درون رشته modifiedSentence انتقال می‌دهد

```
System.out.println("From Server:" + modifiedSentence);
```

این خط –که همچنین در TCPClient ارایه شد- رشته موجود در modifiedsentence را بر روی صفحه نمایش مشتری نمایش می‌دهد

```
ClientSocket.Close();
```

این خط آخر سوکت را می‌بندد.زیرا UDP بدون ارتباط است.این خط باعث نمی‌شود که مشتری پیام Transparent-Layer را به سرویس دهنده ارسال کند(در مقایسه با TCPClient).

UDPServer.java

www.it98.ir

www.it98.ir

```
import java.io.*;
import java.net.*;
class UDPServer {
   public static void main(String args[]) throws Exception
      {
      DatagramSocket serverSocket = new
                              DatagramSocket(9876);
      byte[] receiveData = new byte[1024];
      byte[] sendData = new byte[1024];
      while(true)
         {
         DatagramPacket receivePacket =
               new DatagramPacket(receiveData,
                        receiveData.length);
         serverSocket.receive(receivePacket);
         String sentence = new String(
                     receivePacket.getData());
         InetAddress IPAddress =
                     receivePacket.getAddress();
         int port = receivePacket.getPort();
         String capitalizedSentence =
                     sentence.toUpperCase();
```

```
         sendData = capitalizedSentence.getBytes();
         DatagramPacket sendPacket =
         new DatagramPacket(sendData,
                  sendData.length, IPAddress, port);
         serverSocket.send(sendPacket);
      }
   }
}
```

برنامه UDPServer.java همانطور که در شکل ۲,۳۲ می‌بینیدیک سوکت ایجاد می‌کند. سوکت ServerSocket نامیده می-شود.یک شی از نوع DatagramSocket است.همانطور که سوکت در طرف لایه کاربرد مشتری بود.یکبار دیگر هیچ استریمی به سوکت متصل نیست.

بیایید نگاهی به خط های کد که متفاوت از TCPServer.java هستند بیاندازیم.

DatagramSocket serversocket =New datagramsocket (9876);

wWw.it98.ir

www.it98.ir

خط بالا datagramsocket serversocket را در بخش ۹۸۷۶ ایجاد می‌کند.تمامی داده‌هایی که ارسال و دریافت شده‌اند، باید از طریق این سوکت عبور کنند.چون UDP بدون ارتباط است، ما مجبور نیستیم یک سوکت جدید بسازیم و به گوش دادن به درخواست های جدید ادامه می دهیم.

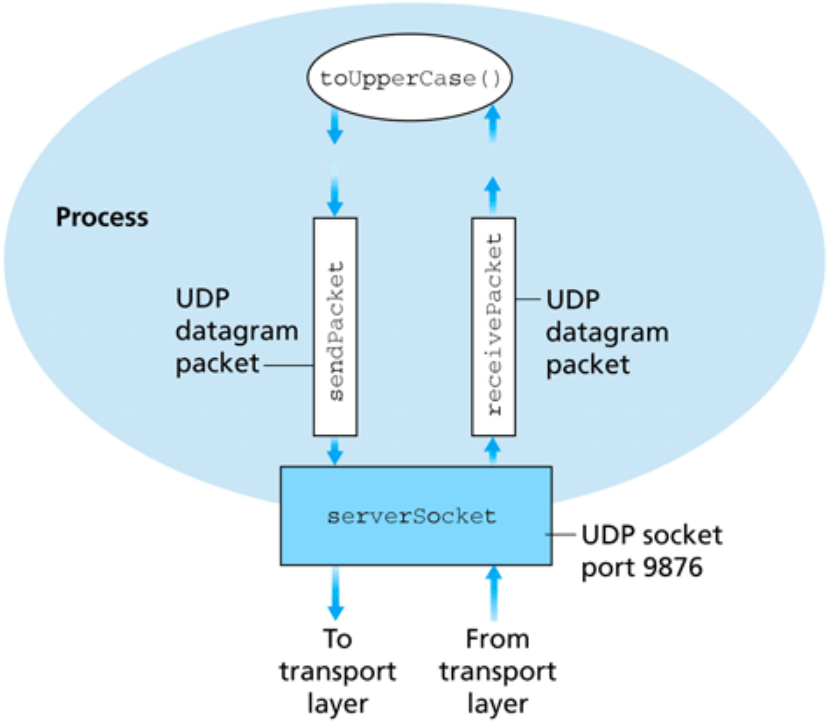


Figure 2.36 ♦ UDPServer has no streams; the socket accepts packets from the process and delivers packets to the process.

```
String sentence = new String(receivePacket.getData();
InetAddress IPAddress  =  receivePacket.getAdress();
int port  = receivePacket.getPort();
```

سه خط فوق، بسته ای را که از سوی یک مشتری رسیده است ، باز می کند. اولین آن ، اطلاعات را از بسته خارج کرده و (داده ها) را در جمله زنجیره ای می گذارد ؛ وخطی مشابه در.UDP Client

دارد. خط دوم آدرس IPرا استخراج کرد، خط چهارم شماره درگاه (ورودی)مشتری را گرفته که از سوی خود مشتری انتخاب شده ومتفاوت از شماره ورودی سرور 9876 می باشد (ماشماره های ورودی مشتری را به طوری مفصل در بخش های بعدی توضیح خواهیم داد .)برای سرور لازم است که آدر سهای (IP وشماره ورودی) مشتری را بگیرد تا بتوان جمله با حروف درشت را به مشتری برگرداند .

این ، تحلیل مااز جفت برنامه ی UDP را کامل می کند .برای بررسی کاربردی آن ، شما .VDP CLIENT.JAVA را در یک گروه (میزبانUDP SERVER.JAVA را در گروه دیگر نصب وهمگردانی (تألیف) کنید. (مطمئن بشوید که بهترین نام میزبانی سرور را در UDP CLIENT لحاظ کرده باشید). سپس دو برنامه را درگروه های خود اجرا کنید . برخلاف .TCP ، شما ابتدا می توانید بخش مشتری وسپس بخش سرور را اجرا کنید . این بدان دلیل است که (پردازش) فرایند مشتری نمی کوشد تا اتصالی با سرور آغاز کند مخصوصاً وقتی که شما برنامه مشتری را اجرا می کنید .یکبار که شما برنامه های سرور ومشتری را اجرا کرده اید

www.it98.ir

www.it98.ir

ممکن است کاربرد آن را با تایپ کردن یک خط در قسمت مشتری مورد استفاده قرار داده باشید. دو نوع ساخت یک وب سرور ساده: اکنون کهHTTP را با جزئیات کمی مورد مطالعه قرار دهیم وآموخته ا یم که چگونه کارهای مشتری سرور را در JAVAبنویسیم ، می خواهیم این دانش جدید را با هم ترکیب کرده ویک وب سرور ساده در HTTP بسازیم. خواهیم دید که این کار به طور قبل مورد ملاحظه ای ساده است.

۱ـ۹ـ۲ نقش ها (کارهای) وب سرور

هدف ما ، ساختن یک سرور است که کارهای زیر را انجام دهد .

- فقط یک در خواستHTTP را انجام دهد.
- در خواست HTTPرا پذیرفته وتجزیه نماید.
- فایل درخواستی را از سیستم فایل سرور بگیرد.
- یک پیام پاسخی HTTPرا بسازد که شامل فایل درخواستی باشد که از قبل توسط خطوط سه اندازه داده شده باشد.
- پاسخ را مستقیماً برای مشتری بفرستد .

بگذارید ساختن یک کد را که تا حد امکان ساده باشد ،امتحان پا کنیم، تا موارد شبکه سازی کاری انجام داده باشیم .رمزی که ما ارائه میکنیم (تحویل می دهیم) بسیار دور از ایمنی (مثل بقیه کد های اصلی) خواهد بود .مثلاً نمی خواهیم نگران بروز استثنائات باشیم .همچنین بگذارید فرض کنیم که مشتری چیزی را خواسته که مطمئناً در سیستم فایل سرویس دهنده قرار دارد.

WebServer.java

www.it98.ir

wWw.it98.ir

www.it98.ir

```
import java.io.*;
import java.net.*;
import java.util.*;
class WebServer {
   public static void main(String argv[]) throws Exception {
      String requestMessageLine;
      String fileName;
      ServerSocket listenSocket = new ServerSocket(6789);
      Socket connectionSocket = listenSocket.accept();
      BufferedReader inFromClient =
         new BufferedReader(new InputStreamReader(
               connectionSocket.getInputStream()));
      DataOutputStream outToClient =
         new DataOutputStream(
               connectionSocket.getOutputStream());
      requestMessageLine = inFromClient.readLine();
      StringTokenizer tokenizedLine =
         new StringTokenizer(requestMessageLine);
      if (tokenizedLine.nextToken().equals("GET")){
         fileName = tokenizedLine.nextToken();
         if (fileName.startsWith("/") == true )
               fileName = fileName.substring(1);
         File file = new File(fileName);
         int numOfBytes = (int) file.length();
         FileInputStream inFile = new FileInputStream (
               fileName);
         byte[] fileInBytes = new byte[numOfBytes];
         inFile.read(fileInBytes);
         outToClient.writeBytes(
               "HTTP/1.0 200 Document Follows\r\n");
         if (fileName.endsWith(".jpg"))
            outToClient.writeBytes("Content-Type:
                  image/jpeg\r\n");
         if (fileName.endsWith(".gif"))
```

www.it98.ir

www.it98.ir

```
            outToClient.writeBytes("Content-Type:
                  image/gif\r\n");
        outToClient.writeBytes("Content-Length: " +
              numOfBytes + "\r\n");
        outToClient.writeBytes("\r\n");
        outToClient.write(fileInBytes, 0, numOfBytes);
        connectionSocket.close();
        }
      else System.out.println("Bad Request Message");
      }
}
```

حالا بگذارید تا نگاهی به کد بیاندازیم . نیمه اول برنامه تقریباً برابر باTCP server .javaاست . مثل TCP server .javaما بسته های java .i oوjava.netرا وارد می کنیم .علاوه براین دو بسته، ما Java .u t I lرا هم وارد می کنیم که شامل طبقه string to k e n i z e r می شود وبرای تجزیه ی پیام های درخواست HTTPاستفاده می شود .حالا نگاهی به خطوط داخل طبقه وب سرور انداخته و دو رشته هدف را تعریف میکنیم.

String request message line
String file Name

خط در خواست پیام هدف یک رشته ای است که شامل خط اول در پیام درخواست HTTPخواهد بود File name (اسم فایل) رشته ای است که شامل اسم فایل درخواستی می باشد .مجموعه بعدی فرمان ها برابر با مجموعه مرتبط فرمان ها در TCP server java. است .

```
ServerSocket listenSocket = new ServerSocket(6789);
Socket connectionSocket = listenSocket.accept();
BufferedReader inFromClient =
   new BufferedReader(new InputStreamReader
            (connectionSocket.getInputStream()));
   DataOutputStream outToClient =
      new DataOutputStream(connectionSocket.
            getOutputStream());
```

دو شی سوکیت مانند ساخته می شوند . اولین آنها listen socketنوع سرور سوکیت می باشد. شی برنامه سرور قبل از اینکه در خواستی را برای یک ارتباط (اتصالTCP سوی مشتری دریافت کند ؛ایجاد می شود. به ورودی ۶۷۸۹ گوش می دهد ومنتظر در خواست از سوی برخی از مشتریان می ماند تا اتصال .....را انجام دهد . هنگامی که در خواستی برای اتصال می رسد روش قبول () شی جدیدی را می سازد به نام سوکیت اتصال که باز هم از نوع سوکت است .

سپس دور شته یا ردیف بوجود می آید Buffer Reader in for client(خواندن حائل هشدار به مشتری) و Date out put stream out to client (رشته اطلاعات خروجی برای مشتری) پیام درخواست HTTPاز شبکه واز طریق سوکت اتصالی می آید

wWw.it98.ir

www.it98.ir

ووارد بخش اطلاع به مشتری می شود پیام پاسخHTTP.وارد Out to client می شود .و از طریق سوکیت اتصال گذشته ووارد شبکه می شوند . بخش باقی مانده که نسبت به TCPC server java طور قابل ملاحظه ای متفاوت است .

```
Request message line =in from client .read line()
```

دستور بالا خط پیام درخواست ..... را می خواند.فرض میشود که این خط به شکل زیر باشد

```
GET file _name HTTP/1.0
```

حال سرور ما باید خط را تجزیه کرده تا نام سیستم را اسخراج کند.

```
String Tokenize r  tokenize r =new string Tokenize r

If  (tokenized line . next Token( )  .equals
File Name= tokenize d line .next Token ( )
If (File Name .start s with ( "/") =true)
File Name = File Name .sub string (1)
```

دستورات بالا خط اول پیام در خواست را تجزیه می کنند تا اسم فایل در خواستی را بیابند.

To k e n I z e d lineرا می توان به عنوان خط درخواست اولیه با هریک از کلمات HTTP/1010وfile nameوgetدر یک جایگاه مخصوص وجداگانه بنام .....نشانه قرار می گیرد.سرور از HTTP RFC.می داند که اسم فایل برای آن فایل درخواستی در آن نشانه لحاظ شده که پس از آنIFمی آید .این اسم فایل در یک رشته بنام FILE NAMEدر کد بالا بداشتن اسلاش است که ممکن است پیش از اسم فایل بیابید.

```
File Input stream in file =new file Input stream
```

فرمان بالا یک رشته را وصل می کند به اسم فایل.

```
 Byte [ ] file In bytes =new byte [num of bytes];
In file .read ( file in bytes);
```

این فرمان ها ، اندازه ی فایل راتعیین می کند ورشته ای از بایت های آن سایز رامی سازد. نام آرایش شده FILE in By t esاست.آخرین فرمان از رشتهin fileمی خواند تا آرایش file In Bytesرا صورت بدهد .برنامه باید به بایت ها تغییر کند زیرا رشته ممکن است فقط با بایت تغذیه شود.

اکنون ما حاضریم تا پیام در خواست HTTP را بسازیم . برای این منظور ما اول باید پاسخ HTTP خطوط سرس را در Data out put stream out to clientبفرستیم.

www.it98.ir

www.it98.ir

```
outToClient.writeBytes("HTTP/1.0 200 Document
        Follows\r\n");
if (fileName.endsWith(".jpg"))
        outToClient.writeBytes("Content-Type:
        image/jpeg\r\n");
if (fileName.endsWith(".gif"))
        outToClient.writeBytes("Content-Type:
        image/gif\r\n");
outToClient.writeBytes("Content-Length: " + numOfBytes +
          "\r\n");
outToClient.writeBytes("\r\n")
```

مجموعه فرامین بالا بسیار جالب هستند. این فرمانه خطوط فوری را برای پیام پاسخ.HTTP آماده می کندوخطوط سری به حائل ارسالی TCP می فرستد.فرمان اول ،خط ومکان اجباری HTTP/1/0/200را می فرستد که به دنبال آن سر سطر رفته ویک خط تغذیه می شود .دو خط فرمان بعدی یک سرخط تکی از نوع محتوی را مهیا می کند. اگر سرور برای اتقال یک تصویر .باشد ،پس سرور سر خط را آماده می کند Content-type: image /gif از سوی دیگر ،اگر سرور برای انتقال یک تصویر JPEG.باشد ، پس سرور سر خط content-type:image/jpegرا آماده می کند (در این وبسرور ساده ، اگر هدف تصویرJPEGیا GIF نباشد ،خط مفهومی ارسال نمی شود). پس سرور یک سر خط محتوی را تهیه وارسال می کند ویک جاخالی اجباری برای آن قبلی می فرستد که حتماً باید ارسال می شد .حال باید فایل file name را.بهData out put بفرستیم .که اکنون می توانیم فایل در خواستی را بفرستیم .

Out To Client. Write(file In Bytes ,num of bytes).

در فرمان بالا فایل درخواستی را درخواست می کند ،file in bytes. .تاTCPحائل را ارسال کند .

TCP.فایل را مورد توجه قرار خواهد داد ،File in bytes. .تاخطوطی که قبلاً ایجاد شده اند ، تا اگر لازم است ، تسلسل را قطعه قطعه (جدا) نمایند وبخش های ..را برای مشتری بفرستند .پس از بکار بردن یک در خواست برای یک بکارگیری یک درخواست برای فایل ،سرور چندین جا خالی را با بستن سوکیت connection socketانجام دهد.

Connection socket .close ( );

برای بررسی این وب سرور ، آن را روی یک (میزبان) نصب کنید. همچنین چند فایل را در گروه بگذارید .پس از یک تست که روی هر ماشین بصورت جسته وگریخته می خواند .

تا یک فایل را از سرور در خواست نمایند . وقتی شما یک فایل را درخواست می کنید ، نیاز به عدد (مقدار)شماره ورودی خواهید داشت که در کد سرور لحاظ خواهید کرد. (مثلاً ۶۷۸۹). اگر سرور شما در some – lost.وe d uکهsomeواقع شده باشد ، فایل some file .htm1وعددورودی ۶۷۸۹است ، پس آن که جسته وگریخته میخواند باید به شرح زیر درخواست شود:

http ://some host .some where.edu:6789/some file.htm1

۱۰ـ۲ خلاصه

در این بخش جنبه های مفهومی (محتوایی) وبکار گیری شبکه را مطالعی کرده ایم .ما در موردی طراحی سرور مشتری که همیشه حاضر باشد آموخته ایم که بار کارهای اینترنتی گرفته شده واستفاده ی آ« را در پروتکل هایHTTP,FTP,SMTP,POP3.وDNS دیده ایم. ما این پروتکل های باسطح کاربری مهم رابررسی کرده ایم .کارهای مرتبط با آنها (وب،انتقال فایل ، ایمیل وDNS ،) بصورت جزئی هم بوده اند .ما همچنین در مورد طراحی P2Pمعمول که کاربردبسیاری

wWw.it98.ir

www.it98.ir

خم دارد چیزهایی یاد گرفتیم . واستفاده های آن را در تقسیم فایل P2P دیده ایم. ما بررسی کردیم که چگونه سوکت ..می تواند برای ساختن کارهای شبکه‌ای استفاده شود .ما سری به استفاده ازTCPیا اتصال مداری وVDPیا خدمات اتقال پایانه ای بدون اتصال زدیم وساختن یک وب سرور را با استفاده از سوکت تمرین کردیم اولین مرحله از سفر ما بسوی طراحی شبکه های لایه‌ای، کامل شده است.

در ابتدای این کتاب در بخش ۱ـ۱ ما تعریفی مبهم وناخوانا از یک پروتکل داشتیم. فرمت وترتیب پیام هایی که بین دویا چند مقدار محاوره ای مثال کار های انجام شده بر دریافت ویا انتقال یک پیام ویا رخدادی دیگر تبادل می شوند . مبحث در این خش مخصوصاً در موردجزئیات بررسی که انجام داده ایم پروتوکول های ,HTTP,FTP,SMTP,POP3وDNSدر شبکه سازی هستند مطالعه ی پروتوکولها ی کاربردی این فرمت را به ما داده تا نظریه ذاتی خود را برای آنچه که پروتوکولها به ما معرفی می کنند بیشتر خواهیم کرد .

در بخش ۱ـ۲ ما مدل های خدماتی را توضیح دادیم که TCP وUDP از لحاظ کاربرد ارائه می کنند .مانگاهی بسیار دقـــیق تر به این مدل های خدماتی انداختیم مخصوصاً آنگاه که کاربردهای ساده ای را در مورد اینکهTCPوUDPاین مدل های خدماتی را چگونه ارائه می کنند مطالب تهی گفته ایم. مثلاً ما در مورد اینکه چطور TCP.یک سرویس انتقال اطلاعات معتبر را برای کارهای خود تهیه میکند کمی سخن گفته ایم .در بخش بعد مانگاه دقیق تری نه تنها به چیستب بلکه به چگونگی ودلیل پروتوکول های انتقالی خواهیم انداخت .

با داشتن اطلاعات در مورد ساختار اینتر نت وپروتوکولهای سطح کاربری حاضریم تا پروتوکول ها را در بخش ۳ بیشتر مورد مطالعه قرار دهیم.

۱ـ پنج کاربرد بدون الویت اینترنت وپروتوکول های لایه ی کاربردی را که آنها اسفاده می کنند نام ببرید؟

۲ـ تفاوت بین طراحی شبکه وطراحی کاربردها چیست؟

۳ـ از چه طریقی یک پیام لحظه ای ترکیبی از سرور مشتری وطراحی p2pاست؟

۴ـبرای یک بخش ارتباطی بین جفت پر وسه چه فرآیندی برای مشتری وکدام برای سرور می باشد؟

۵ـ در یک فایل p2pشما با این جمله موافق هستید یا نه :<<مفهومی از بخش های سرور ومشتری از یک بخش محاوره ایوجود ندارد؟چرا؟چرا نه؟

۶ـ از چه اطلاعاتی در یک گروه برای شناسایی یک فرآیند در گروه یا میزبان دیگر استفاده می شود ؟

۷ـ نام چندین عامل کاربر شبکه ای مختلف رالیست کنید به طور روزانه از آن استفاده می کنید؟

۸ـ با توجه به شکل ۴ـ۲ می بینیم که هیچ یک از کاربردهای لیست شده در شکل ۴ـ۲ نیازی به از دست دادن اطلاعات یا زمان ندارند آیا می توانید کاربردی را تصور کنید که به از دست دادن اطلاعات نیازی نداشته ولی تا حدود زیادی حسلس به زمان باشند؟

۹ـ پروتوکل دست دادن یعنی چه؟

۱۰ـ چرا HTTP,FTP ,SMTP,POP3,IMAPبه جای اینکه روی UDPکار کنند برروی TCPکا می کنند؟

۱۱ـ یک ساعت تجارت الکترونیک را در نظر بگیرید که می خواهد خرید هایش را برای هریک از مشتریانش به شکل ثبت شده نگه دارد .توضیح دهید که چنین چیزی چطور با COOKIES انجام میشود ؟

۱۲ـ تفاوت بین مقاوم با خط گذاری وبدون آن چیست؟ کدامیک از این دو با 101/HTTP استفاده می شود؟

www.it98.ir

www.it98.ir

۱۳ـ شرح دهید که چطور دسیتیلبی به وب میتواند تاخیر در دریافت یک مورد درخواستی را کم کند آیا رسیدن به وب تاخیر برای تمام موارد در خواست شده از سوی یک کاربر یا فقط برای برخی از آن موارد کم خواهد کرد ؟چرا؟

۱۴ـ تل نت در یک وب سرور گذاشته ویک درخواستی چند خطی ارسال کنید.در پیام درخواستی سرخط IF modified since رالحاظ کرده تا پیام را با کد شرایط304 Not modified

تقویت نماید .

۱۵ـ چرا گفته می شود که FTP اطلاعات کنترل را Out of band خارج از باند می فرستد؟

۱۶ ـ فرض منید که آلیس با یک حساب ایمیل وب مثل هات میل یک پیام برای باب می فرستد. چه کسی نامه اش را لز میل سرور با استفاده از pop3 دریافت می کند ؟ بحث کنید که چطور پیام ها از میزبان آلیس به میزبان باب میرسد مطمئن شوید که مجموعه ای از پروتوکول های لایه ی کاربردی را لیست کنید که برای ارسال پیام بین دو میزبان استفاده می شود ؟

۱۷ ـ از سر یک پیام ایمیل که قبلا دریافت کرده اید پرینت بگیرید . چه تعداد خطوط دریافتی در آنجا وجود دارد هر یک از آنها را در یک پیام تحلیل کنید .

۱۸ ـ از دیدگاه یک کاربرفرق بین حالت دانلود حذف ودر pop3 چیست ؟

۱۹ ـ آبا برای یک وب سرور یا میل سرور سازمانی داشتن نام مستعار مشابه برای یک اسم میزبان امکان پذیرا است (مثلاًfoo.com) نوع RP که شامل نام میزبان میل سرور میشود چه باید شد؟

۲۰ـ یک شبکه پوششی در یک سیستم فایل P2P چیست؟ آیا شامل مسیر صاف کن هم می شود؟ حاشیه های پوششی چیستند؟ چطور شبکه پوششی .....ایجادوحفظ می شود ؟

۲۱ـ سه شرکت را بیابید که خدمات تقسیم فایل P2P را ارائه می کند ؟ چه نوع مفاهیمی از طریق این شرکت ها منتشر میشود ؟چطور هریک از این طرح ها کاربر را قادر میسازد تا ؟آن محتوی را بیابد؟

۲۲ ـ شروع VDP توصیف شده در بخش ۸ـ۲ فقط به یک سوکیت نیاز داشت در حالیکه سرور TCP گفته میشود در بخش ۷ـ۲ به دو سوکت نیاز داشت چرا ؟ اگر سرور TCP برای پشتیبانی از N اتصال همزمان باشد که هر کدام از یک میزبان مشتری جداگانه سرور TCP

به چند سوکت نیاز خواهد داشت ؟

۲۳ ـ برای کاربرد مشتری /سرور TCP گفته می شود در بخش ۷ ـ چرا برنامه ی سرور باید قبل از برنامه مشتری اجرا شود ؟برای کاربرد مشتری /سرور در روی یک VDP که در بخش ۸ـ۲ گفته شد چرا ممکن است که برنامه مشتری قبل لز برنامه سرور اجرا شود ؟

مسئله

۱ـ صحیح یا غلط

a. وقتی یک کاربر درخواست یک صفحه وب می کند که شامل چند متن و۲ تصویر باشد برای این صفحه مشتری یک پیام در خواست خواهد فرستاد وسه پیام دریافت خواهد شد .

b. دو صفحه ی وب مشابه می تواند تز طریق اتصال مشا بهی ارسال شوند.

c. یک اتصال نا پایدار بین سروراصلی وآن که سریعاً عوض می شود برای یک قطعه TCP مجزاامکان دارد که دو پیام درخواست HTTP مشابه منتقل شود .

wWw.it98.ir

www.it98.ir

d. تاریخ : در سر پیام پاسخ HTTP مشاهده می شود کا موردی در پاسخ برای آخرین بار توصیف شده باشد

۲ ـ ۹۵۹RFC را برای FTP بخوانید . لقام دستورات مشتری را لیست کنید که ازسوی RFC پشتیبانی می شوند .

۳ ـ HTTP: //www.iani.org رادیدن کنید .ارقام ورودی معروف برای پروتوکول انتقال فایل ساده (s ftp) چیستند ؟برای شبکه های خبری پروتوکول NNTP را انتقال می دهد

۴ـ یک HTTP مشتری را در نظر بگیرید که می خواهد یک مستند وب را در URL داده شده باز یافت کند ، آدرس IP از سرور HTTP ابتدا نا شناخته است .مستند وب در URL یک تصویر GIF الحاقی دارد که در سند اولیه وهمان سرور باقی می ماند ، به چه پروتوکول های انتقالی به غیر از HTTP در این شرایط نیاز است .

۵ـ کاربرد خاص HTTP /۱۰۱ را به دست بیاورید وبه سؤالات زیر پاسخ دهید.

a. مکا منیسم بکار رفته برای سیگنال دهی بین مشتری ، وسرور را توضیح دهید تا نشان دهد که اتصال همیشگی بسته می شود آ؟یا مشتری ، سرور یا هر دو اتصال را نزدیک می کند یا نه؟

b. چه خدمات پنهانی از سوی HTTP تهیه می شود ؟

۶ ـ دریک وب متغیر فرض کنید که روی لینک کلیک میکنید تا به صفحه ی وب برسید آدر س IP برای URL مرتبط در میزبان محلی شما حاصل نمی شود پس یک DNC لازم است تا آدرس IP را به دست آورد .فرض کنیدکه ۸ سرور DNC قبل از دریافت آدرس IP میزبانتان از DNC دیده میشود .بازدید های موفقی از یک ,RTT ازRTT صورت می گیرد .باز هم فرض کنید که صفحه وب مرتبط با لینک دقیقاً شامل یک موضوع است که مقدار کمی متن HTML دارد می خواهیم .

RTT،RTT بین میزبان محلی و سرور دارنده آن هدف را نشان میدهد. فرض میگیریم که زمان انتقال آن هدف صفر باشدء چه مدت زمان صرف میشودء از وقتی که مشتری روی لینک کلیک کرده تا مشتری هدف را دریافت کند؟

۷.با مراجعه به مسئله ۶، فرض کنید فایل HTML سه هدف بسیار کوچک را روی یک سرور به کار می بندد صرف نظر از زمان انتقال، چه مدت صرف میشود برای

a) HTPP ناپایدار با اتصال TCP موازی؟

b) HTPP ناپایدار بدون اتصال TCP موازی؟

c) HTPP پایدار با خطوط؟

دو روش درخواست HTPP, GET و POST هستند.آیا روشهای دیگری هم در HTPP/1.0 وجود دارد؟ اگر بله چرا از آنها استفاده میشود؟ آیا روشهای دیگری هم در HTPP/1.1 وجود دارد؟

.شکل ۱۱-۲ را در نظر بگیرید،که در کدام شبکه ای متصل به اینترنت میشود؟ فرض کنید که سایز متوسط شیء مورد نظر ۹۰۰۰۰۰۰ بیت و میزان متوسط درخواست از بروزر معمولی برای سرور مبدأ ۱/۵ درخواست در هر ثانیه است. همچنین فرض کنیدکه مدت زمانی که طول میکشد تا لینک دستیابی بخش اینترنت به سمت درخواست HTPP برود تا پاسخ را دریافت کند به طور مینگین ۲ ثانیه است(بخش۶-۲-۲ را ببینید). مدل کل مدت زمان متوسط را به عنوان جمع تأخیر در دسترسی و متوسط تأخیر اینترنتی را در یک مدل بگذارید. برای تأخیر متوسط دسترسی، (دلتابتا-۱)/دلتا را استفاده کرده طوری که دلتا، متوسط زمان مورد نیاز برای ارسال یک هدف در لینک دسترسی و بتا، میزان دریافت آن هدف در لینک دسترسی باشد.

a. کل زمان متوسط پاسخ دهی را بیابید؟

www.it98.ir

www.it98.ir

b. حال فرض کنید که یک مخزن در LAN معمولی نسب کرده اید، فرض میگیریم که نرخ(آهنگ) ضربه ۰/۴ است کل مدت زمان پاسخ را حساب کنید؟

۱۰.یک برنامه TCP آسان برای یک سرور بنویسد که خطوط ورودی را از یک مشتری گرفته و این خطوط را بر خروجی استاندارد سرور پرینت میگیرد(شما میتوانید ابن کار را با اصلاح یا تغییر برنامه سرور javaTCP در متن انجام دهید). برنامه تان را همگردانی و اجرا کنید. روی هر ماشین دیگری که دارای وب بروزر باشد یک سرور نماینده را در

میزبان نصب کنیدتا برنامه سرور شما را اجرا کند همچنین شماره(عدد) ورودی را به تناسب دو بیتی پیکربندی کنید.حال، بروزر شما باید پیام های درخواست GET را به سرور شما ارسال کند و سرور شما باید پیامها را روی خروجی استاندارد آن نمایش دهد. از این فرصت استفاده کنیدتا تعیین کنید که آیا بروزر شما پیامهای GET شرطی برای اهدافی میدهد که بطور محلی قرار گرفته اند.

۱۱. POP3RFC,RFC1939 را بخوانید. منظور از فرمان UIDLPOP3 چیست؟

۱۲.دستیابی ایمیلتان را با POP3 در نظر بگیرید.

a. فرض کنید که شما POP میل مشتری خود را طوری پیکربندی کرده اید که در حالت دانلود و حذف کار کند(اجرایی شود). انتقالات زیر را کامل کنید

C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: blah blah …
S: …….blah
S: .
؟

؟

b. فرض کنید شما popmain مشتری را جهت عملیاتی کردن در دانلود و نگهداری آن تصویرسازی کرده اید. مبادله ی پایین را کامل کنید.

C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: blah blah …
S: ……. Blah
S: .
?
?

c. رونوشت خودتان را در قسمت b استفاده کنید. فرض کنید شما مش خواهید پیام ۱ و ۲، pop خروجی را بازیابی کنید، و سپس ۵ دقیقه بعد، شما برای بازیابی E_mail جدید، مجدداً به pop نیاز دارید. فرض کنید که در فاصله ی زمانی ۵ دقیقه، هیچ پیام جدیدی برای شما ارسال شده است. یک رونوشت از pop دوم تهیه کنید.

wWw.it98.ir

www.it98.ir

a.13. یک پایگاه داده whois چیست؟

b. پایگاه داده های مختلفی بر روی اینترنت را برای به دست آوردن نام هایی از دو سرور dns به کار ببرید. نشان دهید که شما از پایگاه داده whois چگونه استفاده می کنید.

۳. nslookup را در جایگاه میزبان خودتان جهت ارسال سئوالات dns به سه سرور dns به کار ببرید:

جایگاه سرور dns شما و دو سرور dns دیگر در بخش (b) وجود دارند. برای استفهام گزارشات نوع a ، ns ، mx و تلاش کنید. یافته های خود را خلاصه کنید.

d. nslookup را جهت یافتن سرور وب که آدرس های ip را چند شاخه می کند، استفاده کنید، آیا سرور وب از موسسه ی شما (مدیر یا شرکت) آدرس های ip شما را چند شاخه می کند؟

e. پایگاه داده arin whois برای تعیین آدرس های ip در مقیاس به کار برده شده دانشگاه خودتان، مورد استفاده قرار دهید.

f. توصیف کنید که چگونه یک مهاجم فرضی می تواند پایگاه داده whois و ابزار nslookup را برای عملیات شناسایی بررسی یک موسسه قبل از پرواز استفاده کند.

g. بحث کنید که چگونه پایگاه داده های whois، می تواند در دسترس عموم قرار گیرد.

۱۴. فرض کنید شما در حال دانلود mp3 هایی برای استفاده در برخی از سیستم p2p file shaving هستید، پورتال محدود در اینترنت در دسترس شما است که 128 kbps با لینک کامل دوگانه است، در حالی که شما در حال دانلود mp3 های خود هستید، ناگهان ۱۰ کاربرد دیگر دانلود mp3 را از کامپیوتر شما آغاز کرده اند. با فرض اینکه کامپیوتر شما خیلی توان بالایی داشته باشد و همه آنها را دانلود کند نمی توان هر نمونه ای را روی کامپیوتر شما گذاشت (cpu, disk I/O , ...) ،

موارد مشابه را upload خواهد کرد که همچنین از پورتال محدود شما در حال عبور و بارگذاری است. دانلودتان با سرعت پایین صورت می گیرد؟ چرا یا چرا نه؟

همچنین پاسخ سوال برای زمانی که شما جریانی فراتر از 128 kbps و کمتر از 512 kbps را از بخش اتصالی یک adsl خواهید داشت ، یکسان و همانند است.

۱۵. فرض کنید n جفت فعال در شبکه gnutella دارید، و هر کدام از جفت ها یک اتصال فعال tcp دارند. به علاوه فرض کنید که اتصالات tcp از خلال یک مسیر m می گذرد. چند تا گره و لبه در شبکه مشابه وجود دارد؟

۱۶. در پوشش gnutella در بخش ۲۰۶ ما برخی جزئیات را توضیح می دادیم که چگونه یک جفت جدید در شبکه gnutella در کنار هم قرار می گیرد. در این مساله ما می خواهیم بیابیم چه روی می دهد هنگامی که یک جفت شبکه gnutella را ترک می کند. به فرض اینکه هر گروه اتصالات tcp را حداقل برای همیشه با چهار جفت نگه می دارد. به فرض جفت x که پنج اتصال tcp با دیگر جفت ها دارد، قصد دارد ترک کند.

a. ابتدا ملاحظه ترک جالب جفت x که به طور واضح اجرای آن متوقف شده است، به موجب آن به طور مطبوعی اتصالات tcp خودش را مسدود کرده است.

چه اقداماتی هر چنج اتصال مذکور جفت ها روی می دهد؟

b. اکنون فرض کنید که x به صورت اختلال آمیزی بدون توه به پنج همسایه اش که اتصال tcp را متوقف کرده اند، از اینترنت قطع ارتباط کرده است، چه روی خواهد داد؟

www.it98.ir

www.it98.ir

۱۷. در این مساله ما به دنبال کشف مسیر وارونه (معکوس) معمولی پیام های quevry hit در gonutella هستیم. فرض کنید که موضوعات alice یک یام query ست. علاوه بر آن فرض کنید که bob پیام های query دریافت کرده است (که ممکن است از سوی چندین جفت واسطه فرشتاده شده باشد.) و یک فایل دارد که سوال را در کنار هم قرار می دهد.

a. بیاد آورید هنگامی که یک جفت یک فایل مشترک دارد آن یک پیام query hit را از طریق مسیرهای معکوس مشابه پیام query ارسال کند. یک طراحی جایگزین برای bob موجود بود تا یک اتصال مستقیم tcp با alice را برقرار کرده و از خلال این اتصال پیام query hit را ارسال کند. مزایا و مضرات این قیبل طراحی جایگزین چه چیزهایی هستند؟

b. در پروتکل gnutella زمانی که جفت alice یک پیام query ایجاد می کند، آن یک id بی نظیر را به پیام اضافه می کند. هنگامی که جفت bob یک همتا دارد، آن پیام query hit را برای id پیام یکسان به عنوان پیام query به کار می برد. توضیح دهید چگونه جفت ها می توانند رشته id پیام را به کار ببرند و جداول مسیرهای معمولی را برای همراهی با مسیر معکوس تعیین کنند.

۳. رویکرد جایگزین که مشخص کننده های پیام را استفاده نمی کند، در مبحث بعدی آمده است.

هنگامی که یک پیام query یک جفت می رسد با قبل از ینکه پیام بفرستند، جفت پیام را با آدرس ip تقویت می کند، توضیح دهید چگونه جفت ها می توانند این مکانیسم را برای کاربرد مسیر معکوس، استفاده کنند.

۱۸. مساله ۱۷ را مرور کنید، اما حاله به سوالات مرتبط با پیام های gnutella ping و pong پاسخ دهید (با توجه به پیام های query hit , `uery)

. در این مساله ما پی می بریم به مراد طراحی یک سیستم شبکه kazaa که معمولاً گروها، رهبران گروه و رهبران فرا گروه را شامل می شود.

a. فرض کنید رهبر فراگروه به سختی ۲۰۰ رهبر گروه است، و رهبر گروه به سختی پاسخگوی ۲۰۰ جفت معمولی است. چد تا از رهبر فراگروه برای یک شبکه چهار میلیون جفتی لازم است.

b. چه اطلاعاتی باید هر رهبر گروه ذخیره کند؟ چه اطلاعاتی باید هر رهبر فرا گروه ذخیره کند؟ چگونه باید تحقیق در این گونه طاحی سه گانه انجام شود؟

۲۰. به سوال جاری در فابل p2p توجه کنید، با در نظر گرفتن بحث انجام شده در ۲۰۶، فرض کنید که هر جفت به حداکثر n همسایه در سرتاسر شبکه متصل شود. همچنین فرض کنید که شماره گروه ابتدا به k در نظر گرفته می شود، فرض کنید alice یک سوال را مطرح کند. درمی یابید یک پیوند فراتر بر روی تعدادی از پیام های query وجود دارد که به درون حوزه تحت پوشش شبکه فرستاده می شود.

۲۱. برنامه ای tcpclient save و udpclient را بر روی یک میزبان و tcp server و udp server را روی میزبان دیگر، راه اندازی و گردآوری کنید.

a. فرض کنید شما tcpcilent را قبلاً از tcp server راه اندازی کنید، چه روی می دهد، چرا؟

b. فرض کنید شما udd server را قبل از udp server راه اندازی کنید، چه روی می دهد، چرا؟

۳. چه روی می دهد گر شما اعداد مختلف را برای مشتری و سرور به کار ببرید؟

۲۲. tcp server . java را آنقدر بازنویسی کنید که بتواند اتصالات چند جانبه را بپذیرد (hint: شما به آن نیاز خواهید داست.)

wWw.it98.ir

www.it98.ir

Datagrams socket client socket= new datagram socket ( ): با Datagrams socket client socket= new datagram socket (5432) آیا متغییر java. Udp server نیاز خواهد بود؟ برای سوکت های در udp client و udp server چه بخش اعدادی هستند؟ آنها قبل از ایجاد این تغییر چه بوده اند؟

سوالات مباحثه‌ای

۱. چرا شما تصور می کنید اعمال فایل p2p خیلی عمومی و عادی است؟ چرا آنها (به طور غیرقانونی) فیلم و موسیقی رایگان را ترویج می کنند؟

چرا تعداد زیادی از سرورها به طور موثر به تقاضای زیاد برای مگابایت ها پاسخ می دهند؟ باید به همه آنها پاسخ دهند؟

۲. آیا هر تلاشی جهت یک منبع باز سیستم p2p file- showing وجود دارد که دگرگون سازی این جفت ها را باعث شود (تکنولوژی مناسب kazaa)

۳. The darknet and the futwre of content distreluti را بخوانید که به وسیله ی biddle.

England و peinado و willman نوشته شود [biddle 2003] . آیا شما با همه دیدگاه های نویسندگان موافق هستید؟ چرا یا چرا نه؟

۴. سایت E- commerce و وب سایت های دیگر پایگاه داده ای پشتیبان دارند. چگونه سرورهای http با این پایگاه داده ای پشتیبان می شوند؟

۵. html پویا چیست؟ مثال هایی از وب سایت هایی بیاورید که html پویا را به کار می برند؟

۳. برنامه زبان نویسی برخی از سرورها چیست؟ هدف از این زبان نویسی چیست؟

۷. چگونه شما می توانید برنامه مشاهده سایت ها را برای موارد مخفی تغییردهید؟ چه انتخاب ها مخفی شما دارید؟

۸. چگونه می تواند برنامه مشاهده سایت را جهت باز کردن اتصالات همزمان چندگانه یک وب سایت تغییر دهید؟ چه مزایا و مضراتی اتصالات متعد و همزمان یک tcp دارد؟

۹. ما دیده ایم که سوکت های tcp اینترنت داده ها را به عنوان یک جریان بایتی ارسال می کند. گیری بایتی چه هستند که به طور واضح api را تشخیص می دهند و محدودیت های پیام را پیشگیری می کنند؟

۱۰. در بخش ۲۰۴۴ را ما متذکر شدیم که سرویس وب مبتنی بر e-mail اغلب با استفاده از یک سرور وب و یک سرور imap عملی می شود. چگونه سرور http با سرور imap ارتباط برقرار می کند؟

۱۱. فرض کنید که شما یک پایگاه داده مشتمل بر داده های نجوم و ستاره شناسی دارید و شما می خواهید داده های در دسترس را برای راه اندازی مشاهده گره های وب بر روی pcs و pdas و تلفن های معمولی ایجاد کنید (متن را برای سخنرانی نیاز دارید). شما چگونه آن را انجام می دهید؟

۱۲. ارتباط میان پیام فوری و پروتکل sip را مورد بحث قرار دهید؟

۱۳. چگونه لیست دوستانتان را در سرورهای پیام فوری به کار می برید؟

۱۴. سرور وب aduche چیست؟ چقدر هزینه دارد؟ اکنون چه کارآیی دارد؟

www.it98.ir

www.it98.ir

۱۵. فرض کنید که سازمان های استاندارد های وب تصمیم گرفته است نامگذاری معمول را آنقدر تغییر دهد که هر موضوعی دارای نام مرجع باشد به وسیله ی یک نام اختصاری که جایگاه مستقلی را به خود اختصاص می داد (urn نامیده می شد) بحث کنید که چگونه برخی از موضوعات اطراف ما این چنین تغییر می کنند؟

۱۶. توجه کنید به برنامه های غیرمدرن و تشویقی که اکنون به وسیله kazaa جهت ترغیب و تشویق کاربران استفاده می شود تا آپ لود فعال داشته باشند. چگونه ممکن است یک سیستم فایل p2p تشویق هایی را برای موضوعات خود فراهم کند؟

(وظایف) برنامه ای سوکت (وسیله برای انتقال ورودی و خروجی بر روی شبکه)

تفویض ۱: سرور وب theared چندگانه

زنجیره ای از پیام ها با موضوع واحد بر روی اینترنت = thear با تفویض این برنامه شما می تواند یک سرور thear چندگانه را گسترش دهید که قادر است چند سوال موازی را سرویس دهد. شما قصد دارید نسخه 1.0 از http را طبق تعریف rfc در سال ۱۹۴۵، اجرا کنید.

بیاد آورید که http/ 1.0 یک اتصال tcp جداگانه برای هرجفت سوال/ پاسخ ایجاد می کند. یک th جداگانه هر کدام از این اتصال را حفظ می کند. همچنین یک thear اصلی وجود خواهد داشت، جایی که سرور درخواست ارباب رجوع را برای اتصالات پایدار متوجه شود.

جهت ساده کردن وظیفه این برنامه، شما می توانید که را در دو مرحله بسط دهید. در مرحله ی اول شما یک سرور thear چندگانه را بنویسید که به طور ساده محتویات پیام های مورد تقاضای http را که دریافت شده است نمایش می دهد. بعد از این برنامه، به درستی راه اندازی کنید، شما که مورد نیاز برای ایجاد یک پاسخ مناسب اضافه خواهید کرد.

با توجه به اینکه شما که را بسط دهید، شما سرور خود را می توانید با یک برنامه نویسی وب امتحان کنید. اما بیاد آورید که شما نمی توانید در پورت ۸۰ سرویس دهی داشته باشید، زیرا شما به نام میزبان خود نیاز دارید. Host. Someschool سرور شما برای پورت ۶۷۸۹ برنامه ریزی شده است، و شما می خواهید فایل indox. Html را تکرار کنید، سپس شما به طور ویژه url ذیل را در browser وارد کنید.

http:// host.someschool.edu:6788/index.html

هنگامی که سرور شما با یک خطا روبرو شده، آن می بایست یک پیام پاسخ دهنده را با یک منبع مناسب html ارسال کند به طوری که اطلاعات خطا در پنجره browser نمایش داده شود. شما می توانید جزئیات کامل این ماموریت را بیابید، مانند کد java در وب سایت :

http://www.awl.com/ kurose-ross.

ماموریت (وظیفه) ۲: ایمیل در باب رجوع (مشتری)

در این ماموریت، شما می توانید در java یک ایمیل کاربرد را با ویژگی های زیر کامل کنید:

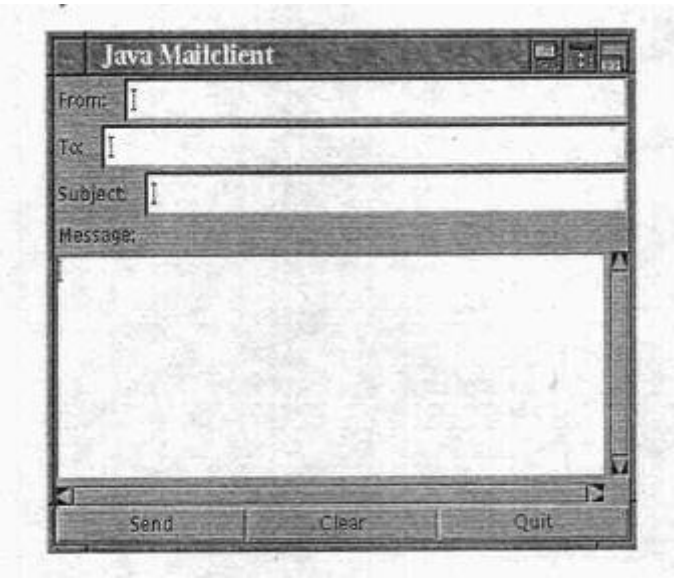
یک نمای گرافیکی را برای فرستنده فراهم کنید. در زمینه هایی برای سرور تعیین محل ایمیل، آدرس فرستنده ایمیل، آدرس گیرنده ایمیل، موضوع پیام و خود پیام.

یک اتصال tcp بین ایمیل مشتری و سرور محل ایمیل ایجاد کنید. دستورات smpt را به سرور محل ایمیل ارسال کنید. دریافت ها و روش های دستورات smpt را از سرور محل ایمیل به دست آورید.

در اینجا آنچه شبیه سیستم شما است، وجود دارد.

wWw.it98.ir

www.it98.ir



شما می توانید نماینده کاربردی را کامل کنید طوری که یک پیام الکترونیکی را به حداکثر دریافت کنندگان در یک زمان ارسال کند. علاوه بر این، نماینده کاربر گمان خواهد کرد که حوزه ی آدرس دریافت کنندگان نام قراردادی سرور smpt دریافت کنندگان است. (نماینده کاربر نمی تواند یک جستجوی dns را برای هر ثبت mx انجام دهد) شما می توانید به جزئیات کامل ماموریت پی ببرید مانند قسمت های مهم کد (رمز) java در وب سایتhttp:// www.awl.com /kurose-ross

ماموریت ۳: udp pinger lab

در این آزمایشگاه شما می توانید یک udp ساده مبتنی بر مشتری و سرور اجرا کنید. کارآیی قابل دسترس در این برنامه مشابه برنامه صدا است که در سیستم های عملیاتی نوین قابل دسترس است. استاندارد ping با (icmp) پروتکل پیام کنترل اینترنت ارسالی- پیام echo کار می کند ه به عنوان وسیله حذف ویژگی های برگشتنی به فرستنده است سپس فرستنده می تواند زمان چرخشی میان خود و کامپیوتر تولید صدا را تعیین کند.

Java نمی تواند هر کارآیی (کاربرد) را برای فرستادن و دریافت پیام های icmp فراهم کند که در این آزمایشگاه شما اجرا خواهید کرد، صدای در لایه ای اجرایی b به وسیله پیام ها و سوکت های udp . شما می توانید جزئیات کامل این ماموریت را از قسمت های مهم کد java در وب سایت http:// www.awl.com /kurose-ross بیابید.

ماموریت ۴: web proxy server وب سرور میان سرور واقعی و مشتری

در این آزمایشگاه شما یک سرور پروکسی وب ساده را کامل خواهید کرد، که همچنین قادر است صفحات وب را مخفی کند. این سرور یک پیام get را از browser خواهد پذیرفت، پیام get را به سرور وب مقصد بفرست، پیام پاسخ http را از سرور مقصد دریافت کن، و پیام پاسخ http را به browser بفرست. این یک سرور وب پروکسی بسیار ساده است. آن وقت تقاضاهای ساده get را می شناسد. بهرحال، همه انواع موضوعات را حفظ کند، نه فقط صفحات html ، بلکه تصاویر. شما می توانید جزئیات کامل ماموریت را از قسمت های مهم کد java بیابید در وب سایت :

http:// www.awl.com /kurose-ross
ethereal labs
http

پایتان را با ethereal packet smiffel که آزمایشگاه شما یک مرطوب کنید. شما حالا آمادگی دارید تا etereal را برای تحقیق پروتکل ها در عملیات استفاده کنید. در این آزمایشگاه، ما چندین تعریف از پروتکل http را بررسی خواهیم کرد. اساس

www.it98.ir

www.it98.ir

واکنش get/ پاسخ. فرمت های پیام http فایل های خروجی بزرگ html فایل های خروجی html با درج urls ، اتصالات پایدار و ناپایدار و پایداری و ایمنی (اینست) http. با توجه موارد بر وجود همه در آزمایشگاه های ethereal توضیح کامل این آزمایشگاه در وب سایت این کتاب قابل دسترس است:

http:// www.awl.com /kurose-ross

dns

در این آزمایشگاه، یک نگاه دقیق تر مشتری در dns وجود دارد، پروتکل که نام های میزبان اینترنت را به آدرس ip، ترجمه می کند. از بخش ۲,۵ بیاد آورید که نقش مشتری در dns نسبتاً معمولی است- یک مشتری یک سوال را به محل سرور dns می فرستند و یک پاسخ را دریافت می کند. بیشتر تحت پوشش ادامه پیدا می کند، برای مشتری dns قابل شماهده نیست، سرورهای dns منظم با دیگری ارتباط برقرار می کند که به صورت تکرار یا مستمر، سوال مشتری را مجدداً حل کند.

از نظر مشتری dns، بهرحال پروتکل کاملاً ساده است. یک سوال برای سرور dns شکل می گیرد و یک پاسخ از همان سرور دریافت می کند. ما عمل dns را در این آزمایشگاه مشاهده می کنیم. توصیف کامل این آزمایش در وب سایت این کتاب قابل دسترس است.

http:// www.awl.com /kurose-ross

مصاحبه با tim burners- lee

Tim مدیر کنسرسیوم www (w3c) و دانشمند علوم پایه در آزمایشگاه mits برای دانش کامپیوتر در سال ۱۹۵۹ است. در حالی که در آزمایشگاه اروپایی particle physic (cern) کار می کند. Tim یک مبنای پایه ای اینترنتی را برای شراکت اطلاعات جهانی اختراع کرد که به طور معمولا به عنوان word wide web آن را می شناسد. یک سال بعد او اولین وب مشتری و سرور را نوشت. Tim مدرک خود را از انگلیس در سال ۱۹۷۰ در رشته ی فیزیک از دانشگاه آکسفورد دریافت کرد.

- شما اساساً فیزیک مطالعه می کردید. شبکه چه شباهتی با فیزیک دارد؟ هنگامی که شما فیزیک مطالعه می کنید شما متوجه می شوید قوانین حاکم بر رفتار در مقیاس کوچک، امکان مشاهده آن در مقیاس بزرگ جهانی را به ما می دهد. هنگامی که شما یک سیستم جهانی مانند وب را طراحی می کنید، شما تلاش می کنید تا قوانین رفتاری صفحات وب و پیوندها و چیزهایی را اختراع کنید که آن را در مقیاس یزرگ جهانی ببینیم.

تجزیه و تحلیل

یکی آنالیز است و دیگر سنتز (ساخت و ساز) ولی آنها خیلی شبیه هستند.

- نفوذ شما در ویژه ساختن، شبکه چیست؟

بعد از اخذ مدرک فیزیک خودم، تحقیق ارتباطات از راه دور شرکت ها، بهنظر می رسید که در این زمینه علایق زیادی وجود دارد. این فرایندهای کوچک به وجود آمده اند و ارتباطات از راه دور خیلی سریع از /// قانونی به سیستم های مبتنی بر فرآیند های کوچک، آغاز شد. آن خیلی هیجان انگیز بود.

- اکثر چالش های کارش چه بود؟

هنگامی که دو گروه به شدت در یک مورد با هم مخالف هستند، اما می خواهند در پایان به یک هدف مشترک برسند، واقعاً آنها درمی یاند هرچیزی چه مفهومی دارد و عدم درک متقابل کجاست، خیلی مورد مطالبه می تواند باشد. جایگاه هر گروه کاری می داند که بهرحال، این چیزی است که پیشرفت را به سوی مقبولیت عام در یک مقیاس بزرگ را باعث می شود؟

- شما چه رویایی را برای آینده شبکه/ اینترنت دارید؟

wWw.it98.ir

www.it98.ir

با توجه به گفته ام در کتاب، من یک فکر (رویا) برای وب دارم ... و آن دو بخش است.

در بخش اول یک وسیله با توانمندی خیلی زیاد برای ملاطفت و انس بین مردم باشد. من همیشه تصور کرده ام که فضای اطلاعات به عنوان چیزی است که هر کسی دستیابی فوری و مستقیم درشتر باشد و نه فقط به browse به ایجاد ... . علاوه بر این رویای ارتباط افراد با افراد از خلال دانش مشترک باید برای گروه هایی از همه اقشار امکان پذیر باشد، واکنش اکترونیکی با روانی و همواری زیاد را در فرد انجام می‌دهند.

در بخش دوم از رویا، انس و ملاطفت به کامپیوترها کشانده می شود. دستگاه های قادرند همه داده ها بر روی وب را آنالیز کنند. محتوی، پیوندها و واکنش های میان افراد و کامپیوترها. یک "semantic web" که باید این امکان را ایجاد کند، تاکنون ابداع شده است، اما هنگامی آن انجام می شود، مکانیسم های روز به روز معاوضه، بروکراسی و زندگی روزانه ما، با ماشین ها قابل مدیریت است و علوم استقبال انسان از تهیه و تدویت القایی و مستقیماً ... این ماشین های همشمند وب از خلال، عملکرد یک سری از پیشرفت های تکنیکی و هنجارهای اجتماعی به دست خواهد آمد که اکنون در حال آغاز شدن است.

از آنجایی که رویا دو بخش قابل دستیابی است، وب یک فضایی خواهد بود که تصور موجودیت انسان و دلیل وجود همزمان ماشین در یک روند مورد دلخواه، ترکیب توانمندی را ایجاد کند.

– مردم چه چیزی به صورت حرفه ای به شما القاء و الهام کرده اند؟

والدینم، که در وزهای اول محاسبه درگیر بودند. یک قوت قلب همراه با اتهام موضوعات به من دادند. Mike sendall و piggie rimmer همان کسانی هستند که من در زمان های مختلف برای آنها در cern کار کردم، در میان مردم، مرا بیشتر از همه آموزش داند و تشویق کردند. من بعداً یاد گرفتم که دیگران را تحسین کنم مانند vanever , busl ... و ted nelson که رویاهای مشابه در زمان خود داشتند اما حضور و وجود مناسب در p3s نداشتند و اینترنت قادر بود آن را درک کند.

ماموریت کنسرسیوم www است به هداست وب به بیشترین توانایی بالقوه خود در دیدگاه شما پتانسیل کامل وب چیست؟ چه راه هایی منجر به دستیابی وب به پتانسیل کامل خود می شود؟

ما در حال تلاش برای پیشبرد آن در دو دستورالعمل هستیم که من متذکر شدم، پیش به سوی انس و ملاطفت و پیش به سوی وب semantic در هر زمانی اما در حال تلاش برای ارتقای جهانی وب به جایی که فقط یک وب دارد. آن همیشه قابل دسترس است و هر کسی می تواند بدون در نظر گرفتن وسیله سخت افزاری دست فروش نرم افزار، موقعیت جغرافیایی، ناتوانی، زبان و فرهنگ به آن دست یابد.

انجمن فناوری اطلاعات
INFORMATION TECHNOLOGY FORUM
www.it98.ir