بررسی سرویس های اینترنتی

پروتکل انتقال ابرمتن(HTTP)

پروتکل انتقال ابرمتن(Hyper Text Transfer Protocol) یک پروتکل لایه کاربرد برای سیستم‌های اطلاعاتی توزیع شده، مشارکتی و ابررسانه می‌باشد. این پروتکل اساس ارتباط داده ای برای وب جهان گستر است. *ابرمتن* متنی ساخت یافته است که پیوندهای منطقی(ابر پیوند-Hyperlink) میان گره های حاوی متن را به کار می برد. HTTP پروتکلی برای تبادل یا انتقال ابرمتن ها است. این پروتکل عمومی علاوه بر استفاده اصلی آن در ابرمتن‌ها در بسیاری از زمینه‌های دیگر کامپیوتری مانند سیستم نام دامنه (DNS) قابل استفاده است. از نسخه اولیه، این پروتکل در وب جهانی استفاده می‌شد و آخرین به‌روز رسانی آن در ماه جون ۲۰۱۴ تحت عنوان «HTTP/1.1» صورت گرفت.

توسعه ی استانداردهای این پروتکل بر عهده نیروی امور مهندسی اینترنت (IETF) و کنسرسیوم وب جهان گستر (W3C) می‌باشد. این امر در گروه کاری پروتکل انتقال ابرمتن (HTTP Working Group) صورت می‌گیرد.

تیم برنرز لی، طراح و پیشنهاد دهنده وب جهانی که اکنون تحت عنوان WWW شناخته می‌شود، برای اولین بار پروتکل انتقال ابر متن را به همراه ساختار اولیه زبان نشانه گذاری ابرمتن (HTML) در یک وب سرور ساده و یک مرورگر مبتنی بر متن ارائه داد. در این نسخه ی اولیه تنها روش درخواست (Request Method) موجود GET و تمامی پاسخ ها به زبان HTML بودند.

اولین نسخه ی مستند پروتکل انتقال ابرمتن نسخه ی ۰،۹ آن بود که در سال ۱۹۹۱ منتشر شد. دیو راگت، که در سال ۱۹۹۵ گروه کاری پروتکل انتقال ابرمتن (HTTP Working Group) را رهبری می‌کرد، خواستار گسترش این پروتکل شد و در نهایت نسخه ۱،۰ تحت عنوان «HTTP/1.0» در سال ۱۹۹۶ به صورت رسمی معرفی شد.

HTTP به شکل یک پروتکل درخواست-پاسخ در مدل محاسباتی سرویس دهنده-سرویس گیرنده عمل می کند. برای مثال یک مرورگر ممکن است سرویس گیرنده باشد و برنامه کاربردی روی کامپیوتری که میزبان یک وب سایت است می تواند سرویس دهنده باشد. سرویس گیرنده یک پیغام درخواست HTTP به سرویس دهنده ارسال می کند. سرویس دهنده که منابعی مثل فایل های HTML یا دیگر محتوا را آماده می کند یا فعالیت های دیگری از طرف سرویس گیرنده انجام می دهد، یک پیغام پاسخ به سرویس گیرنده بازمی گرداند. پاسخ شامل اطلاعات وضعیت اتمام پیرامون درخواست می شود و همچنین ممکن است شامل محتوای درخواستی در بدنه ی پیغام باشد.

مرورگر وب نمونه ای از یک عامل کاربری(User Agent) است. دیگر انواع این عامل های کاربری نرم افزارهای نمایه سازی به کار رفته توسط ارائه دهندگان جستجو(Web Crawler) مرورگر های صوتی، برنامه های تلفن همراه و نرم افزارهای دیگر که محتوای وب را مصرف می کنند، نمایش می دهند و یا به آن دسترسی دارند هستند.

HTTP یک پروتکل از لایه کاربرد است که در چارچوب مجموعه پروتکل های اینترنت (Internet Protocol suite) طراحی گردیده است. تعریف آن بر مبنای پروتکل لایه انتقال زیرین و قابل اطمینان است و معمولا پروتکل کنترل انتقال (TCP) به کار می رود. اگرچه HTTP می تواند پروتکل های غیرقابل اطمینان نظیر پروتکل دیتاگرام کاربر (UDP) را نیز به کار برد مثلا در پروتکل کشف خدمات ساده(SSDP).

منابع HTTP بر روی شبکه بوسیله شناسه های منبع یکنواخت (URI) یا به طور مشخص تر مکان یاب های منبع یکنواخت (URL) – که از طرح های شناسه های منبع یکنواخت http و https استفاده می کنند- شناسایی و موقعیت یابی می شوند.

جلسه(Session)

در پروتکل انتقال ابر متن به دنباله ای از درخواست ها و پاسخ ها جلسه(Session) گفته می‌شود. سرویس گیرنده با ایجاد یک TCP بر روی یک پورت(Port) از پیش تعیین شده بر روی سرویس دهنده (معمولا پورت شماره ۸۰)، جلسه را آغاز می‌کند. سرور وب همواره بر روی پورت، در انتظار درخواست های سرویس گیرنده ها می باشد. بعد از دریافت درخواست ارسال شده، سرویس دهنده با ارسال یک خط وضعیت(Status Line) و بدنه، پاسخ سرویس گیرنده را به او باز می گرداند. بدنه بسته ی پاسخ، معمولا حاوی منبع درخواست شده است؛ با این حال از آن برای ارسال خطا و اطلاعات دیگر نیز استفاده می شود.

یک نمونه از خط وضعیت در پاسخ به یک درخواست مجاز:

```
HTTP/1.1 200 OK
```

روش های درخواست(Request Methods)

پروتکل انتقال ابر متن روش هایی را برای درخواست تعریف کرده است (Request Method) که هر کدام از آنها باعث انجام عمل خاص در سمت سرویس دهنده می شوند. نسخه ی ۱.۰ روش های درخواست GET، POST و HEAD را دارا بود. در نسخه ی ۱.۱ پنج روش جدید افزوده شد: OPTIONS، PUT، DELETE، TRACE و CONNECT. از آنجایی که عملکرد این روش ها به طور کامل تعریف و شرح داده شده است، لذا تمامی مرورگر ها و سرور ها به راحتی می توانند این روش ها را پیاده سازی و استفاده نمایند. اگر روشی برای سرور تعریف نشده باشد، با آن به عنوان یک روش غیر امن برخورد خواهد کرد. در تعداد روش ها هیچ محدودیتی وجود ندارد. این نکته باعث می شود که گسترش احتمالی این پروتکل در آینده به زیر ساخت های فعلی آن آسیبی نرساند و آنها

را تغییر ندهد. برای مثال در حال حاضر پروتکل WebDAV هفت روش جدید درخواست را تعریف کرده است. این نکته نیز قابل تامل است که برخی از این درخواست ها(مثل HEAD، GET، OPTIONSو TRACE) به عنوان ایمن تعریف شده اند، به این معنا که فقط برای بازیابی اطلاعات قابل استفاده اند و نباید تغییر حالتی در سرویس دهنده ایجاد کنند. به عبارت دیگر نباید هیچ اثر جانبی، حتی اثرات نسبتا بی ضرر مانند ورود(Logging)، ذخیره سازی(Caching) و تبلیغات یا افزایش شمارنده ی وب، داشته باشند.

GET

درخواست نمایش منبع درخواست داده شده را می دهد. (این منبع معمولا یک فایل یا پرونده می باشد.) این روش فقط اطلاعات را از سرور دریافت می کند و نباید هیچ تاثیری بر روی منابع سرور بگذارد.

HEAD

این روش دقیقا مانند روش GET عمل می کند با این تفاوت که بدنه پاسخ را نمی خواهد. از این روش برای به دست آوردن فرا داده های موجود در سر آیند (Header) استفاده می شود. یکی از استفاده های رایج این نوع درخواست، بررسی تغییر یافتن یک منبع است.

POST

در این روش به همراه بسته ی درخواست اطلاعاتی نیز فرستاده می شود. سرور با توجه به نشانی وب (URL) درخواست شده و اطلاعات ارسال شده، منبع مورد نظر را در بسته ی پاسخ بر می گرداند. این اطلاعات ارسالی می تواند نام کاربری و کلمه ی عبور، یک نظر بر روی یک مطلب و یا اطلاعات هر فرم دیگری که توسط کاربر وارد شده است، باشد.

PUT

در این روش منبعی به همراه بسته ی درخواست ارسال شده و از سرور تقاضا می شود که این منبع را در آدرس موجود در بسته بار گذاری کند. اگر در محل درخواست شده قبلا منبع دیگری قرار داشته باشد، منبع جدید جایگزین خواهد شد.

DELETE

از سرور درخواست می کند که آدرس فرستاده شده را حذف نماید.

TRACE

در این روش سرور اطلاعات ارسال شده را عیناً به سرویس گیرنده باز می گرداند. (برای بررسی تغییراتی که واسط های شبکه بر روی بسته می گذارند، از این روش استفاده می شود.)

OPTIONS

از سرور تقاضا می کند تا روش های درخواست (Request Method) موجود برای نشانی فرستاده شده را اعلام نماید. برای گرفتن تمامی روش های درخواست قابل اجرا بر روی سرویس دهنده می توان از نشانی همه(یا *) استفاده کرد.

CONNECT

بسته ی پروتکل ابر متن را به یک تونل TCP/IP تبدیل می کند. این عمل معمولاً برای برقراری ارتباط امن (HTTPS) بر روی یک پراکسی سرور ناامن استفاده می شود.

PATCH

این روش که در سال ۲۰۱۰ به پروتکل افزوده شد، برای ایجاد تغییرات جزیی بر روی منابع استفاده می شود. سرورهای وب موظف هستند حداقل روش های GET و HEAD را پیاده سازی نمایند.

وضعیت جلسه

پروتکل انتقال ابر متن یک پروتکل بدون وضعیت(Stateless) می باشد. بدین معنی که سرویس دهنده در یک جلسه هیچ ردی از کاربر ذخیره نمی کند. به طور مثال، سرویس دهنده وب هیچگاه نمی تواند به یاد بیاورد که شما در این وب سایت ورود کرده اید یا نه! اما به دلیل نیاز شدید نرم افزار های تحت وب به ثبت وضعیت، با استفاده از تکنیک ها زیر این عمل انجام می گیرد:

- کوکی
- استفاده از متغیر های پنهان در فرم های وب
- استفاده از متغیر های موجود در رشته ی درخواست. مانند: index.php?session_id=some_unique_id

مثالی از یک جلسه

درخواست سرویس گیرنده

```
GET /index.html HTTP/1.1
Host: www.example.com
```

پاسخ سرویس دهنده

```
HTTP/1.1 200 OK
```

```
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
ETag: "3f80f-1b6-3e1cb03b"
Content-Type: text/html; charset=UTF-8
Content-Length: 131
Accept-Ranges: bytes
Connection: close

<html>
<head>
  <title>An Example Page</title>
</head>
<body>
  Hello World, this is a very simple HTML document.
</body>
</html>
```

پروتکل انتقال فایل(FTP)

پروتکل انتقال فایل(File Transfer Protocol) یک پروتکل شبکه استاندارد است که برای انتقال فایل های کامپیوتری از یک میزبان به میزبان دیگری بر روی شبکه های مبتنی بر TCP نظیر اینترنت به کار می رود.

FTP بر اساس معماری سرویس دهنده- سرویس گیرنده (Client-Server) ساخته شده و از ارتباط مجزای کنترلی و داده ای میان سرویس گیرنده و سرویس دهنده بهره می برد. برای احراز هویت کاربران FTP از پروتکل ورود شفاف معمولا به شکل یک شناسه و رمز عبور استفاده می شود اما در صورت تنظیم سرویس دهنده امکان اتصال ناشناس نیز وجود دارد. برای انتقال ایمن که شناسه و رمز عبور را محافظت می کند و محتوا را پنهان می کند، FTP غالبا با SSL/TLS (FTPS) ایمن می گردد. پروتکل انتقال فایل SSH (SFTP) نیز گاهی به جای آن استفاده می گردد اما از لحاظ فنی متفاوت است.

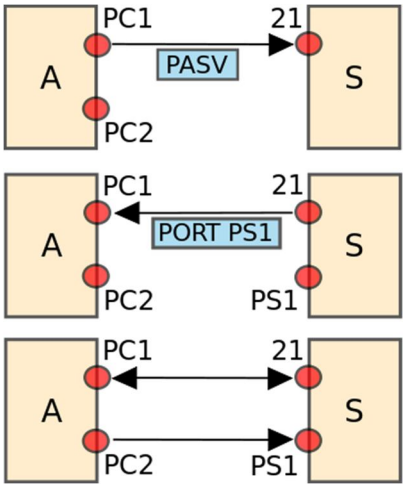
برنامه های کاربردی سرویس گیرنده FTP در ابتدا برنامه های مبتنی بر خط فرمان بودند که قبل از اینکه سیستم عامل ها رابط کاربری گرافیکی داشته باشند، توسعه یافتند و هنوز هم به همراه سیستم عامل های رایج ارائه می شوند. تا کنون سرویس گیرنده ها و ابزارهای خودکارسازی FTP متعددی برای کامپیوترهای رومیزی، سرورها، دستگاه های همراه و سخت افزارها ارائه شده اند و FTP در برنامه های کاربردی ویژه ای مثل ویرایش گر های صفحات وب گنجانده شده است.

خدمات پروتکل انتقال فایل(FTP)

- تهیه ی لیستی از فایل های موجود از سیستم فایل کامپیوتر راه دور

- حذف، تغییر نام و جابجا کردن فایل های کامپیوتر راه دور
- جستجو در شاخه های کامپیوتر راه دور
- ایجاد یا حذف شاخه روی کامپیوتر راه دور
- انتقال (بار گذاری) فایل از کامپیوتر راه دور به کامپیوتر میزبان(download)
- انتقال فایل و ذخیره ی آن از کامپیوتر میزبان به کامپیوتر راه دور(upload)

قابلیت هایی که پروتکل FTP عرضه می کند همانند Telnet می تواند برای سیستم سرویس دهنده بسیار خطرناک باشد زیرا به سادگی می توان فایل های یک کامپیوتر راه دور را آلوده یا نابود کرد. پس در این پروتکل کاربران باید قبل از تقاضای هر سرویس شناسه و کلمه ی عبور خود را وارد کنند و سرویس دهنده پس از احراز هویت کاربر، سطح دسترسی و عملیات مجاز برای کاربر را تعیین می کند و یک نشست FTP آغاز می شود.



شکل ۳ – نمایش آغاز یک اتصال منفعل با استفاده از پورت ۲۱

روش های برقراری یک نشست FTP

ایجاد نشست بین سرویس دهنده و مشتری FTP با دو روش معمولی(Normal Mode) و روش غیر فعال (Passive Mode) امکان پذیر است در روش معمولی برای برقراری یک نشست FTP مراحل زیر انجام می شود:

الف) در برنامه ی سمت مشتری ابتدا دو سوکت نوع TCP با شماره پورت تصادفی بالای ۱۰۲۴ ایجاد می شود.

ب) در مرحله ی دوم برنامه ی سمت مشتری سعی می کند با استفاده از دستور connect اتصال یکی از سوکت های ایجاد شده را با پورت شماره ۲۱ از سرویس دهنده برقرار نماید. اگر این اتصال برقرار شود در حقیقت کانال فرمان باز شده و پروسهPI آماده تفسیر فرامین صادره از سمت مشتری است.

ج) برنامه ی سمت مشتری با فرمان "PORT x" به برنامه ی سمت سرویس دهنده شماره ی پورت سوکت دوم را اعلام می نماید و منتظر می ماند. (در حقیقت برنامه مشتری روی سوکت دوم عمل listen انجام می دهد.)

د) در ادامه برنامه ی سرویس دهنده سعی می کند یک اتصال TCP با شماره پورت اعلام شده از مشتری برقرار نماید.

ه) برنامه سمت مشتری اتصال TCP شروع شده از سرویس دهنده را تصدیق کرده و یک نشست FTP آغاز می شود. ابتدا برنامه ی سمت مشتری دو سوکت مجزا باز کرده و شماره پورت های دلخواه و تصادفی (مثل ۵۱۵۰ و ۵۱۵۱) را به آن ها مقید می کند. سپس از طریق سوکت اول یک اتصال TCP با پورت ۲۱ از سرویس دهنده برقرار کرده و پس از برقراری اتصال، با ارسال فرمان "PORT 5151" شماره پورت سوکت دوم خود را اعلام می کند. برنامه ی سمت سرویس دهنده ضمن تصدیق پذیرش درخواست نشست، بلافاصله اقدام به برقراری یک اتصال TCP بین پورت ۲۰ خودش و پورت دوم (شماره ۵۱۵۱) از مشتری می نماید. با تصدیق این اتصال توسط مشتری نشست FTP آغاز می شود.

رای برقراری یک نشست FTP به روش غیر فعال نیز باید:

الف) در برنامه سمت مشتری ابتدا دو سوکت نوع TCP با شماره پورت تصادفی بالای ۱۰۲۴ ایجاد می شود.

ب) برنامه ی سمت مشتری سعی می کند اتصال TCP یکی از سوکت های ایجاد شده را با پورت شماره ی ۲۱ از سرویس دهنده برقرار نماید. با برقراری این ارتباط کانال فرمان باز شده و پروسه ی PI آماده ی تفسیر فرامین صادره از سمت مشتری خواهد بود.

ج) برنامه ی سمت مشتری با فرمان PASV به برنامه ی سمت سرویس دهنده اعلام می کند که خواستار یک نشست از نوع غیر فعال است.

د) برنامه ی سمت سرویس دهنده یک سوکت با شماره پورت تصادفی بالای ۱۰۲۴ ایجاد کرده و شماره ی آن را به برنامه ی مشتری اعلام می نماید.

ه) برنامه ی سمت مشتری اتصال سوکت دوم خود را با شماره پورت اعلام شده برقرار کرده پس از تصدیق اتصال، نشست FTP آغاز می شود.

نحو (Syntax) در FTP

نحو FTP URL طبق استاندارد به شکل زیر است:

```
ftp://[<user>[:<password>]@]<host>[:<port>]/<url-path>
```

که موارد داخل کروشه اختیاری هستند.

برای مثال :

```
ftp://public.ftp-servers.example.com/mydir/myfile.txt
```

فایلی با نام myfile.txt را از پوشه یmydir بر روی سرویس دهنده ی public.ftp-servers.example.com به عنوان منبع FTP نمایش می دهد.

یا در این مثال:

```
ftp://user001:secretpassword@private.ftp-servers.example.com/mydir/myfile.txt
```

شناسه و رمز عبور مورد نیاز برای دسترسی به همان فایل را ذکر کرده است.

پروتکل انتقال نامه ساده(SMTP)

پروتکل انتقال نامه ساده (Simple Mail Transfer Protocol) استاندارد اینترنتی برای انتقال پست الکترونیکی (E-Mail) است. اولین بار در سال ۱۹۸۲ تعریف شد. و در سال ۲۰۰۸ با عنوان Extended SMTP برای آخرین بار به روز رسانی شده که امروزه این پروتکل به طور گسترده استفاده می شود.

به طور پیش فرض SMTP پورت TCP شماره ۲۵ را به کار می برد. پروتکل تایید نامه نیز همان است اما پورت ۵۸۷ را استفاده می کند. اتصالSMTP ایمن شده با SSL که به نام SMTPS شناخته می شود روی پورت ۴۶۵ است.

در حالیکه سرورهای نامه و دیگر عامل های انتقال نامه از SMTP برای ارسال و دریافت پیام های نامه استفاده می کنند، برنامه های کاربردی نامه سرویس گیرنده های سطح کاربر عموما از SMTP تنها برای ارسال پیام ها به سرورهای نامه به منظور Relay کردن بهره می برند. برای دریافت پیام ها، برنامه های کاربردی سرویس گیرنده از POP3 و یا IMAP استفاده می کنند.

در حالی که سیستم های اختصاصی (مثل MS Exchange) و Webmail (مثل Hotmail، Gmail و Yahoo) پروتکل های غیر استاندارد خودشان را برای دسترسی به حساب های نامه ی بر روی سرویس دهنده های نامه شان به کار می برند، اما تمامی آنها برای ارسال و دریافت ایمیل از بیرون سیستم هایشان از SMTP استفاده می کنند.

مدل پردازش نامه

ایمیلی که توسط یک سرویس گیرنده ی نامه (MUA) به یک سرور نامه (MSA) ارسال می شود، از SMTP روی پورت TCP شماره ۵۸۷ استفاده می کند. بسیاری از خدمات دهنده های ایمیل هنوز اجازه می دهند ایمیل از طریق پورت ۲۵ ارسال شود. از آنجا MSA نامه را به عامل انتقال نامه(MTA) تحویل می دهد. اغلب این دو عامل دقیقا ماهیت های متفاوتی از یک نرم افزار اجرا شده با حالت های متفاوت بر روی یک ماشین هستند. پردازش

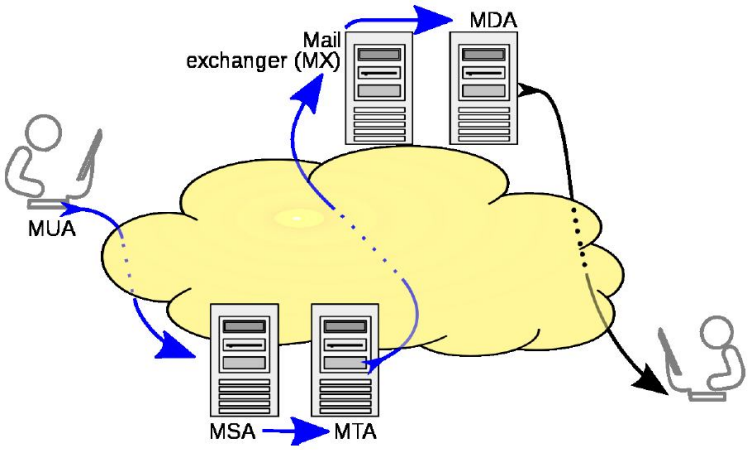
محلی هم می تواند روی یک ماشین و هم میان دستگاه های مختلف تقسیم شود. در حالت اول، فرآیندهای درگیر می توانند فایل ها را به اشتراک بگذارند. در حالت دوم. SMTP با هر میزبانی که برای استفاده از دستگاه بعدی به عنوان *میزبان* هوشمند تنظیم شده، برای انتقال داخلی پیام ها استفاده می شود. هر فرآیند یک MTA با قوانین خودش است. به طور کلی آن، یک سرور SMTP است.

MTA نهایی باید میزبان مقصد را مکان یابی کند. برای این کار از DNS استفاده می کند تا رکورد Exchanger نامه(MX Record) را برای دامنه ی گیرنده(بخشی از آدرس ایمیل که بعد از @ می آید) بررسی کند. رکورد MX بازگردانده شده شامل نام میزبان مقصد است. سپس MTA به سرور Exchange به عنوان یک سرویس گیرنده SMTP وصل می شود.

بعد از اینکه MX مقصد پیام ورودی را پذیرفت، آن را به دست یک عامل تحویل نامه(MDA) برای تحویل محلی نامه می دهد. MDA قادر است پیام ها را در قالب صندوق پستی مربوطه ذخیره کند. باز هم، دریافت پست الکترونیکی می تواند با استفاده از چندین کامپیوتر و یا فقط یکی انجام شود(تصویر دو صندوق نزدیک به هم را در هر دو حالت نمایش می دهد). MDA ممکن است پیام را به طور مستقیم به ذخیره سازی تحویل دهد یا آنها را بوسیله SMTP یا هر روش دیگری مثل LMTP یا نوع طراحی شده خاصی از SMTP به شبکه دیگری ببرد.

با تحویل به سرور نامه محلی، نامه برای بازیابی دسته ای بوسیله سرویس گیرنده های نامه تصدیق شده (MUAها) ذخیره می شود. نامه بازیابی شده توسط برنامه کاربردی کاربر نهایی، که سرویس گیرنده ایمیل نامیده می شود، از پروتکل دستیابی پیام اینترنتی(IMAP)، پروتکلی برای تسهیل هم دسترسی به نامه و هم مدیریت نامه های ذخیره شده، یا پروتکل دفتر پستی (POP) که از قالب فایل نامه mbox قدیمی استفاده کرده یا از سیستم اختصاصی مثل MS Exchange/Outlook، بهره می برد.

SMTP انتقال پیام را و نه محتوای آن را تعریف می کند. پس، آن پاکت نامه و پارامترهای آن را مثل فرستنده پاکت و نه عنوان(به جز اطلاعات ردیابی) و متن پیام را تعریف می کند.



شکل ۴ - خطوط جهت دار آبی رنگ می توانند با انواع مختلف SMTP پیاده سازی شوند

بررسی پروتکل SMTP

SMTP یک پروتکل مبتنی بر متن اتصال گرا است که در آن یک فرستنده نامه با گیرنده ی نامه به وسیله صدور رشته های دستور و ارائه داده های ضروری روی یک کانال جریان داده سفارشی قابل اعتماد، معمولا TCP، ارتباط برقرار می کند. یک جلسه SMTP شامل فرمان هایی است که از سرویس گیرنده ی SMTP (عامل آغاز کننده، فرستنده یا ارسال کننده) آغاز شده و از سرور SMTP مربوطه(عامل شنود کننده، یا دریافت کننده) پاسخ داده شده در نتیجه جلسه باز می شود و پارامترهای جلسه مبادله می شوند. یک جلسه ممکن است هیچ تراکنش SMTP را شامل نشود یا چندین تراکنش را شامل شود. یک تراکنش SMTP شامل سه دنباله ی فرمان/پاسخ است:

۱. دستور MAIL: برای برقراری آدرس بازگشت، یا مسیر بازگشت یا مسیر معکوس یا فرستنده پاکت و ... . این همان آدرسی است که پیام Bounce باید به آن ارسال شود.
۲. دستور RCPT: برای ایجاد یک دریافت کننده پیام. این فرمان می تواند چندین بار، هرکدام برای یک دریافت کننده، صادر شود. این آدرس ها نیز جزئی از پاکت هستند.
۳. DATA برای اعلام آغاز متن یا محتوای پیام. شامل یک عنوان پیام و یک بدنه پیام که با یک خط خالی از هم جدا شده اند. DATA در واقع مجموعه ای از فرمان هاست، و سرور دو بار پاسخ می دهد: یکبار دقیقا به فرمان DATA برای پاسخ به اینکه آماده دریافت متن است و بار دوم پس از دنباله end-of-data برای اینکه کل پیام را بپذیرد یا رد کند.

از طرف دیگر پاسخ های میانی به DATA، هر پاسخ سرور می تواند مثبت(کد پاسخ 2xx) یا منفی باشد. پاسخ های منفی می توانند دائمی(کدهای 5xx) یا گذرا (کدهای 4xx) باشند. پاسخ reject نقص دائمی به وسیله سرور

SMTP است؛ در این شرایط سرویس گیرنده SMTP باید یک پیام bounce بفرستد. پاسخ drop جوابی مثبت است که با پیام چشم پوشی به جای تحویل ادامه می یابد.

مقایسه POP و IMAP

- POP پروتکل بسیار ساده تری است که باعث پیاده سازی ساده تر آن می شود.
- ایمیل POP پیام ها را از سرور ایمیل به کامپیوتر محلی شما منتقل می کند، همچنین اغلب این انتخاب را دارید تا پیام ها روی سرور ایمیل نیز باقی بمانند. اما پیش فرض IMAP این است که پیام ها روی سرور ایمیل باقی مانده، یک نسخه محلی از آنها به سادگی بارگذاری می شود.
- POP با صندوق پستی به عنوان یک فضای ذخیره سازی کلی رفتار می کند، و درکی از پوشه ها ندارد.
- یک سرویس گیرنده IMAP پرس و جو های پیچیده ای مثل پرسش عنوان ها از سرور، یا بدنه پیام هایی خاص و یا جستجوی پیام ها با استفاده از ضوابط خاص را اجرا می کند. پیام ها در مخزن نامه با پرچم های وضعیت گوناگونی(مانند "حذف شده" یا "پاسخ داده شده") علامت گذاری می شوند و در مخزن باقی می مانند تا زمانی که کاربر به طور صریح آنها را حذف کند- که ممکن است تا جلسه های بعدی این اتفاق رخ ندهد. به اختصار: IMAP به گونه ای طراحی شده که اجازه می دهد با صندوق های پستی راه دور به عنوان محلی دستکاری شوند. با توجه به پیاده سازی سرویس گیرنده ی IMAP و معماری ایمیل مورد نظر مدیر سیستم، کاربر می تواند پیام ها را مستقیما روی ماشین سرویس گیرنده ذخیره کند، یا آنها را روی سرویس دهنده ذخیره نماید و یا این انتخاب را داشته باشد که هر یک از آنها را انجام دهد.
- در پروتکل POP سرویس گیرنده متصل فعلی باید تنها سرویس گیرنده متصل به صندوق پستی باشد. بر خلاف این، پروتکل IMAP به طور صریح اجازه دسترسی همزمان توسط چندین سرویس گیرنده را می دهد و مکانیزم هایی دارد تا سرویس گیرندگان از تغییرات انجام شده در صندوق پستی توسط دیگر سرویس گیرندگان با هم متصل شده، آگاه شوند.
- وقتی POP پیامی را بازیابی می کند، تمام بخش های آن را دریافت می کند، در حالی که پروتکل IMAP4 اجازه بازیابی هر یک از بخش های MIME منفرد – مثلا بازیابی متن ساده بدون بازیابی فایل های پیوست شده - را به طور مجزا می دهد.
- IMAP از پرچم ها روی سرور برای پیگیری وضعیت پیام پشتیبانی می کند: به طور مثال، آیا پیام پاسخ داده شده، خوانده و یا حذف شده است یا خیر.