

آموزش الکترونیک برای همه

Electro Volt.ir

FPGA

ARM

AVR

پروژه های الکترونیک

نرم افزارهای الکترونیک

کتاب های الکترونیک



Electrovolt_ir



Electrovolt.ir

آشنایی با لایه کاربرد در مدل TCP/IP و تشریح عملکرد پروتکل HTTP

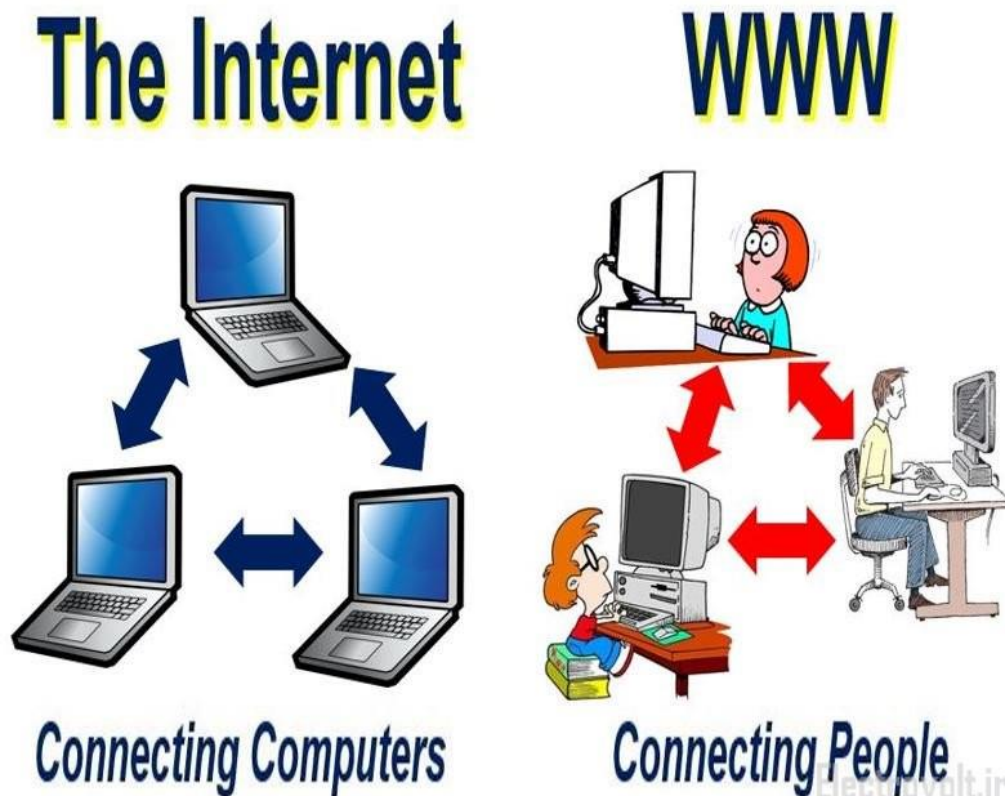
مقدمه

در مقاله قبلی با عنوان آشنایی با اصول و مبانی شبکه های کامپیوتری به تشریح انواع شبکه ها و مدل TCP/IP پرداختیم و گفتیم شبکه جهانی وب و اینترنت بر اساس این مدل کار می کند. بالاترین لایه در این مدل ، لایه اپلیکیشن (Application Layer) می باشد که به طور مستقیم با کاربر در ارتباط است. در این مقاله ابتدا به طور کلی به بررسی و تشریح این لایه خواهیم پرداخت و سپس پروتکل های آن از جمله پروتکل HTTP را بررسی خواهیم کرد.



تفاوت اینترنت و وب چیست؟

اکثر کاربران فکر می کنند که وب همان اینترنت است اما اینترنت در سال ۱۹۸۳ وقتی که آرپانت که همان پدر اینترنت نامیده می شود شروع به استفاده از پروتکل TCP/IP کرد شکل گرفت. "اینترنت" لایه ی فیزیکی یا شبکه ای متشکل از سویچ ها، روترها و تجهیزات دیگر شبکه ای است که وظیفه ای اصلی آن انتقال اطلاعات به صورت سریع، قابل اطمینان و امن از یک به نقطه به نقطه ی دیگر است. از طرف دیگر "وب" توسط تیم برنرزی انگلیسی در سال ۱۹۸۹ ساخته شده است. وب روی اینترنت کار می کند و نقش اصلی آن هم فراهم کردن واسطی است تا اطلاعاتی که به وسیله ی اینترنت منتقل می شود قابل استفاده و کاربردی شود.



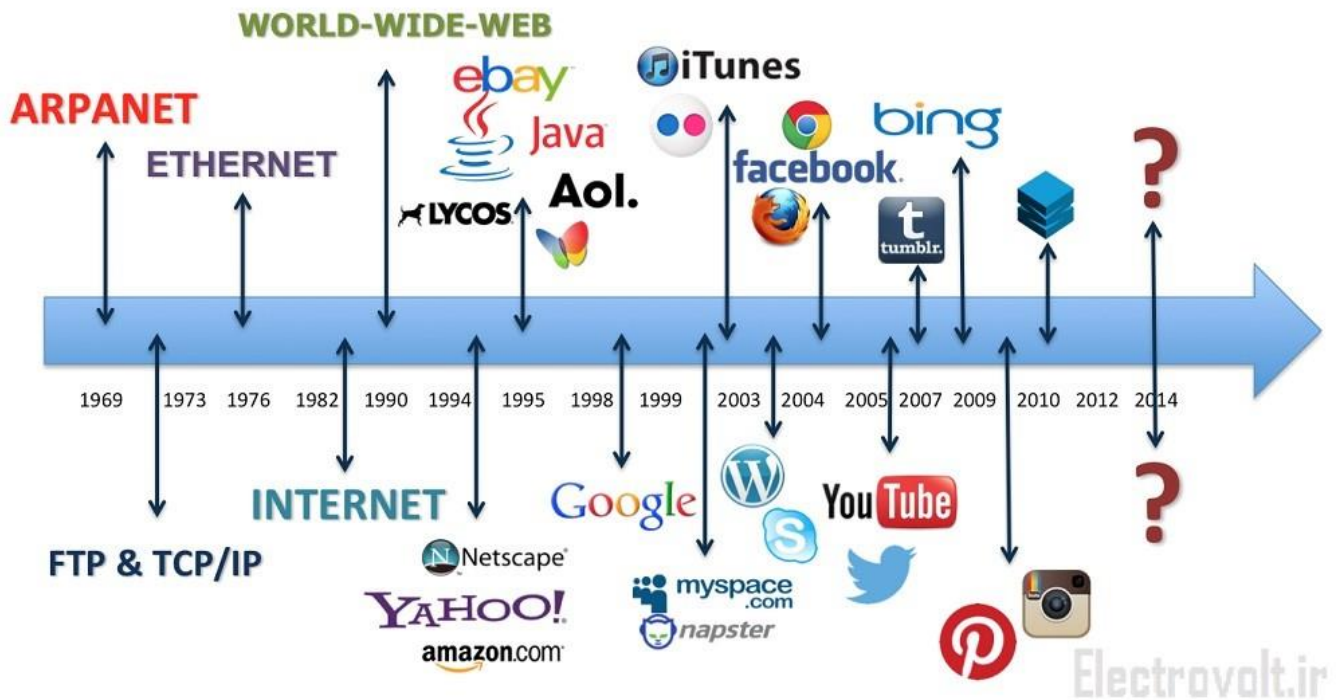
در واقع اینترنت مجموعه ای از فناوری است که موجودیت وب را رقم زده است. اگر اینترنت نبود وب هم نبود و اگر وب نبود اینترنت هم چنان در اثر هجر باقی می ماند.

وب در چهار مرحله دچار تحولی عظیم شد:

- 1- اولین مرحله فاز پژوهش بود، در این مرحله وب فقط در مراکز تحقیقاتی و آکادمیک استفاده می شد.
- 2- مرحله دوم به فاز بروشور گونه معروف است. در این مرحله بر روی نیاز کمپانی ها و سازمان ها برای به اشتراک گذاری اطلاعات روی اینترنت تمرکز شده بود تا مردم با محصولات و سرویس های مختلف آشنا شوند.

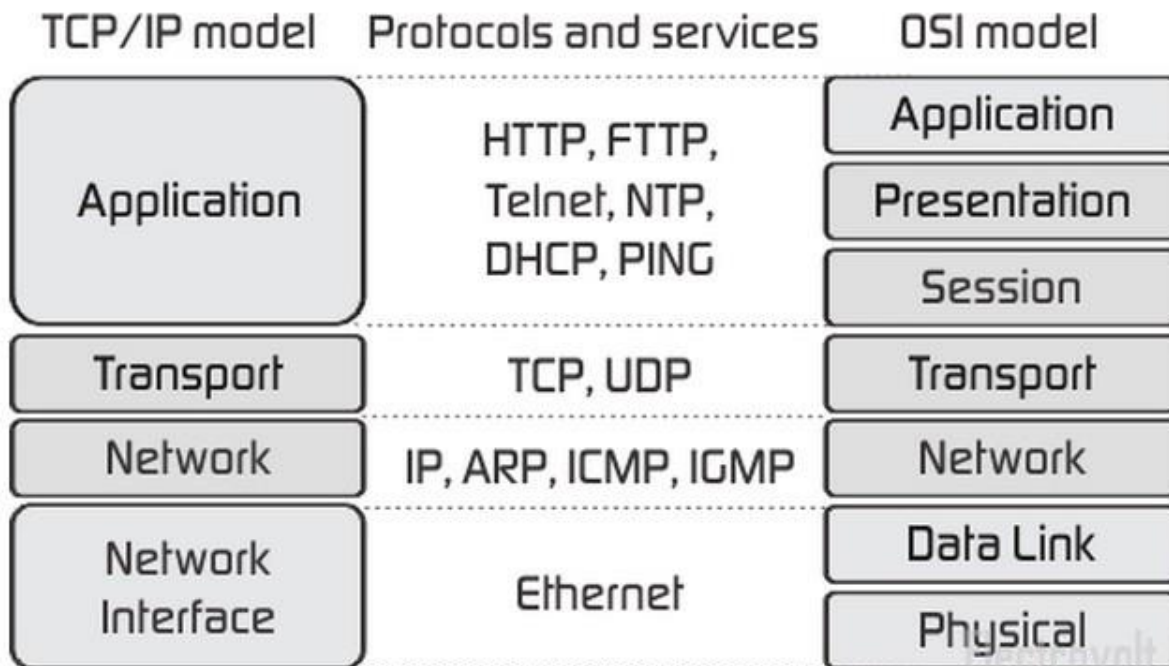
3- تحول سوم آنجایی بود که وب از یک محیط با داده های ایستا به یک محیط تعاملی و اطلاعات تراکنشی تبدیل شد و محصولات و خدمات گوناگون قابل خرید و فروش بود. شروع فعالیت سایت هایی مانند ebay و amazon در این مرحله قابل توجه است.

4- مرحله چهارم جایی است که ما در حال حاضر در آن قرار داریم ، یعنی "وب اجتماعی" جایی که علاوه بر امکانات قبلی سایت هایی مثل فیسبوک و توییتر وجود دارند و انسان ها از این طریق با هم ارتباط برقرار می کنند.



لایه کاربرد (Application Layer) چیست ؟

لایه کاربرد بالاترین لایه در مدل های OSI و TCP/IP می باشد. این لایه در هر دو مدل به دلیل اهمیت تعامل کاربر و برنامه کاربری وجود دارد. اما تفاوتی بین این دو مدل وجود دارد و آن هم این است که سه لایه بالایی در مدل OSI تبدیل به لایه کاربرد در مدل TCP/IP شده است. شکل زیر این موضوع را نشان می دهد.



در حقیقت در مدل TCP/IP، لایه کاربرد شامل تمام پروتکل‌ها و متدهایی می‌باشد که در حوزه پردازش عملیات‌های مملوس با کاربر، در سرتاسر شبکه‌های مبتنی بر پروتکل اینترنت (IP) قرار می‌گیرند. اما در مدل OSI، تعریف لایه کاربرد دقیق‌تر است، زیرا عملیات افزون‌تری نسبت به لایه کاربرد در TCP/IP دارد بطوری که دو لایه اضافی یعنی لایه نشست (Session Layer) و لایه نمایش (Presentation Layer) بین این لایه و لایه انتقال قرار می‌گیرد. مدل OSI ثابت کرده که بهترین ابزار برای توصیف شبکه‌های کامپیوتری است. اما پروتکل‌های TCP/IP در مقیاس وسیعی از جمله در وب و اینترنت مورد استفاده قرار می‌گیرد.

یک کاربر ممکن است بصورت مستقیم یا غیر مستقیم با برنامه‌های کاربردی تعامل داشته باشد. لایه کاربرد جایی است که ارتباط واقعی برقرار شده و منعکس می‌شود. به این دلیل این لایه در بالای تمام لایه‌ها قرار گرفته زیرا هیچ خدماتی به لایه‌های دیگر ارائه نمی‌کند. لایه کاربرد از لایه انتقال کمک می‌گیرد در حالی که لایه‌های دیگر ایجاد ارتباط و انتقال داده‌ها به میزبانهای دیگر را انجام می‌دهند. زمانی که یک پروتکل لایه کاربرد بخواهد با پروتکل نظیر خود در میزبان راه دور (Host) ارتباط برقرار کند اطلاعات یا داده‌ها را به لایه انتقال می‌فرستد. لایه انتقال مابقی کارها را با کمک سایر لایه‌ها انجام می‌دهد.

ابهامی در درک لایه کاربرد و پروتکل‌های آن وجود دارد. هر نرم‌افزار کاربر نمی‌تواند در لایه کاربرد قرار داده شود. به جز آن دسته از برنامه‌های کاربردی که با سیستم ارتباطی در تعامل هستند. به عنوان مثال، نرم‌افزار طراحی و یا ویرایشگر متن به عنوان برنامه‌های لایه کاربرد در نظر گرفته نمی‌شوند. از سوی دیگر، زمانی که ما از یک مرورگر وب استفاده می‌کنیم، در واقع توسط پروتکل انتقال ابر متن (HTTP) در حال تعامل با شبکه می‌باشیم HTTP. یکی از پروتکل‌های لایه کاربرد است پس در نتیجه مرورگر وب یک نرم‌افزار لایه کاربرد می‌باشد. مثال دیگر پروتکل انتقال فایل می‌باشد که برای انتقال فایل‌های مبتنی بر متن یا باینری در سراسر شبکه به کاربران کمک می‌کند. کاربر می‌تواند این پروتکل را در نرم‌افزارهای مبتنی بر رابط کاربری

گرافیکی مانند FileZilla استفاده کند. همچنین کاربر می تواند FTP را در حالت خط فرمان نیز استفاده کند. از این رو، صرف نظر از اینکه کدام نرم افزار را استفاده می کنید، این پروتکل در لایه کاربرد قرار دارد و توسط آن نرم افزار استفاده می شود.

پروتکل های لایه کاربرد

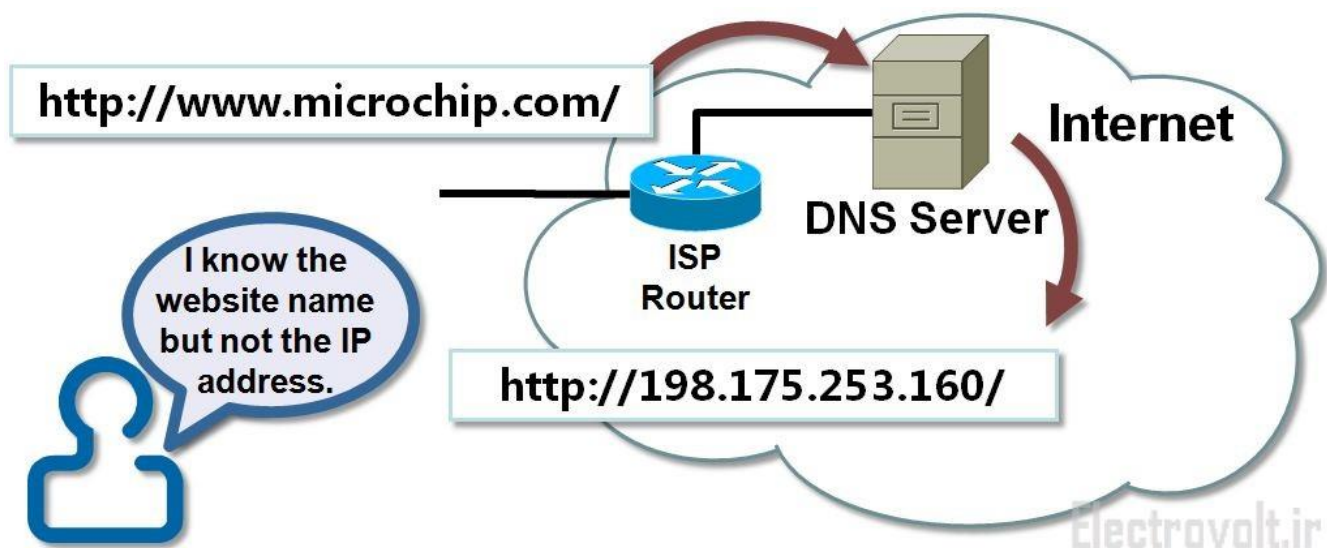
دستگاه های مبدأ و مقصد ، از پروتکل هایی که در لایه کاربرد زندگی می کنند ، برای ایجاد یک جلسه (Session) استفاده می کنند تا ارتباط به صورت صحیح انجام پذیرد. به طور کلی وظایف پروتکل های لایه کاربرد شامل موارد زیر است:

1. ایجاد قوانین سازگار برای انتقال اطلاعات بین سرویس ها و برنامه های کاربردی که بر روی دو کامپیوتر مبدأ و مقصد اجرا شده اند.
 2. مشخص کردن نوع و ساختار داده ها، پیام هایی که بین مبدأ و مقصد جابجا می شوند. این پیام ها می تواند درخواست سرویس، پیام تصدیق دریافت، داده ها، وضعیت پیام و خطاها باشند.
 3. انجام تشخیص خطا به منظور ایجاد اطمینان از اینکه داده به صورت صحیح به مقصد رسیده باشد.
- در شکل زیر برخی از پروتکل های مهم در لایه کاربرد را مشاهده می کنید.

Application	Description
DHCP	Dynamic Host Configuration Protocol assigns IP addresses
DNS	Domain Name System translates website names to IP addresses
HTTP	Hypertext Transfer Protocol used to transfer web pages
NBNS	NetBIOS Name Service translates local host names to IP addresses
SMTP	Simple Mail Transfer Protocol sends email messages
SNMP	Simple Network Management Protocol manages network devices
SNTP	Simple Network Time Protocol provides time of day
Telnet	Bi-directional text communication via a terminal application
TFTP	Trivial File Transfer Protocol used to transfer small amounts of data

سرویس نام گذاری دامنه (DNS)

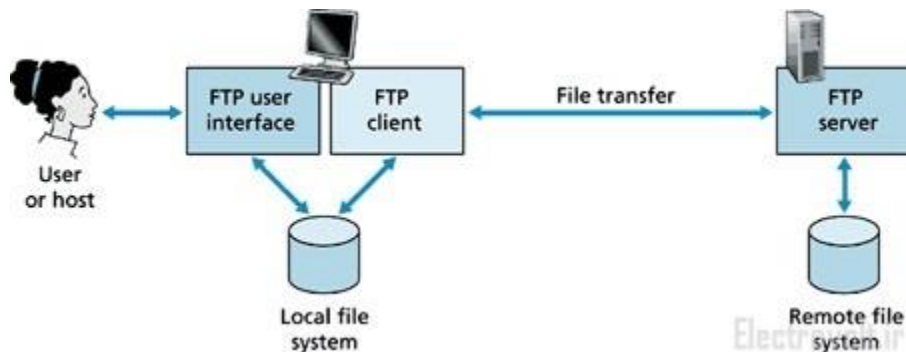
Domain Name Service یا DNS برای تبدیل نام به آدرس IP استفاده می‌شود. فرض کنید می‌خواهید به سروری متصل شوید که آدرس IP آن را ندارید، طبق مدل TCP/IP اتصال فقط از طریق IP صورت می‌گیرد. پس کامپیوتر شما قبل از اتصال به سرور اصلی، آدرس IP مربوط به سرور را از سرویس DNS درخواست می‌کند. آدرس IP سرور را به کامپیوتر داده و سپس اتصال به وسیله IP بین سرور و کلاینت انجام می‌پذیرد. در حقیقت سرویس DNS تمامی نام‌ها و آدرس‌های IP کلیه سرورهای دنیا را به صورت جدول بندی شده در درون خود دارد و به محض درخواست، اطلاعات آدرس IP مورد نظر را در اختیار قرار می‌دهد. DNS با استفاده از پورت UDP 53 کار می‌کند. برای مثال زمانی که شما electrovolt.ir را در مرورگر خود وارد می‌کنید، ابتدا سیستم شما به سرویس DNS مراجعه کرده و آدرس IP سرور الکترو ولت برگشت داده می‌شود.



سرویس ارسال فایل و پروتکل FTP

یکی از دیگر پروتکل‌های معروف لایه کاربرد پروتکل ارسال فایل یا File Transfer Protocol است. از این پروتکل برای ارسال فایل بین کلاینت و سرور استفاده می‌شود. برای اینکه FTP به درستی کار کند نیاز است تا نرم‌افزار FTP Client بر روی کلاینت و نرم‌افزار FTP Server بر روی سرور اجرا باشد. به صورت کلی در هنگام برقراری ارتباط FTP دو اتصال بین سرور و کلاینت ایجاد می‌شود که شامل یک اتصال برای تبادل اطلاعات کنترلی و اتصال دیگر برای تبادل داده‌ها است. در اتصال FTP کلاینت قادر خواهد بود تا فایل‌هایی را بر روی سرور Upload و یا Download کند. مشتری درخواست برای یک فایل را به سرور می‌دهد. هنگامی که سرور درخواست برای یک فایل دریافت می‌کند، ابتدا یک اتصال TCP برای مشتری باز و انتقال فایل انجام می‌شود.

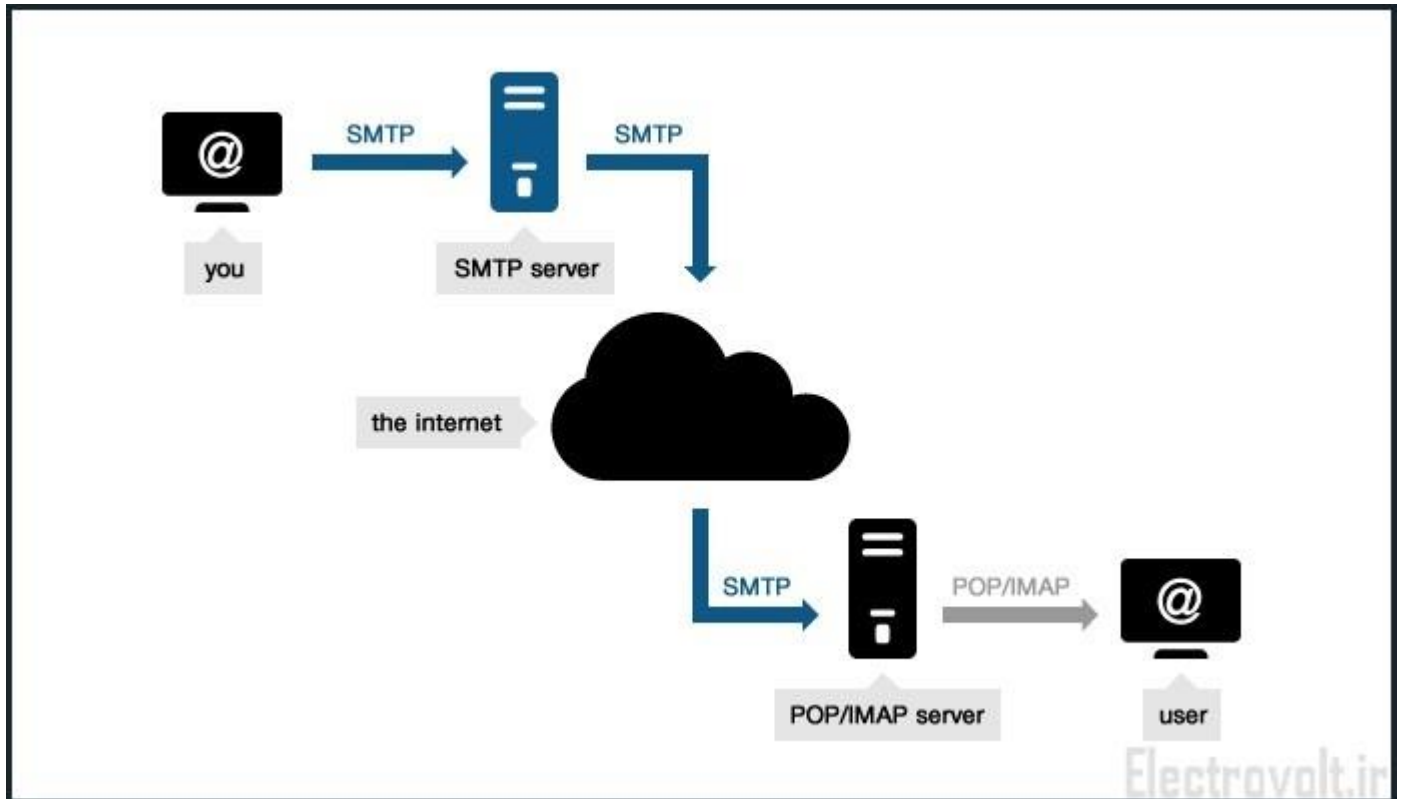
پس از اینکه انتقال کامل شد، سرور اتصال را می بندد. برای ارسال فایل دوم، مشتری درخواست دوباره می دهد و سرور یک اتصال TCP جدید بازگشایی می کند.



بسیار عجیب است که پس از گذشت سالیان سال، کاربران زیادی هنوز از این پروتکل برای ارسال و دریافت داده هایشان استفاده می نمایند. دلیل محبوبیت این پروتکل، شاید بتوان گفت در کاربر پسندی آن است. کاربران می توانند از طریق نرم افزارهای مخصوصی که برای این کار طراحی شده اند، از روی دسکتاپ کامپیوتر خود، فایل های مورد نیازشان را به سرور ارسال، یا دریافت دارند. درست مثل آنکه این عملیات دارد از طریق MyComputer انجام می شود. ولی در پروتکل HTTP اوضاع به این منوال نیست. کاربران بایستی از طریق سایت، ابتدا به سرور سایت متصل شوند و سپس مبادرت به دریافت و ارسال داده ها نمایند.

سرویس ایمیل و پروتکل های POP/IMAP/SMTP

یکی از محبوب ترین سرویس های شبکه، سرویس ایمیل است که کاربران با استفاده از این سرویس می توانند ایمیل های خود را چک کنند. سه پروتکل معروف لایه کاربرد در این رابطه POP، IMAP و SMTP است. به طوری که ابتدا سرویس گیرنده در مبدا، ایمیل مورد نظر خود را با پروتکل SMTP به سرور ایمیل مبدا تحویل می دهد. سپس این ایمیل از طریق اینترنت به سرور ایمیل مقصد تحویل داده می شود. در نهایت سرویس گیرنده در مقصد، از پروتکل های دسترسی به پیام اینترنت IMAP و پروتکل POP برای دریافت و مدیریت ایمیل بهره می برد. شکل زیر روند ارسال و دریافت ایمیل را نشان می دهد.



سرویس پیکربندی پویای هاست و پروتکل DHCP

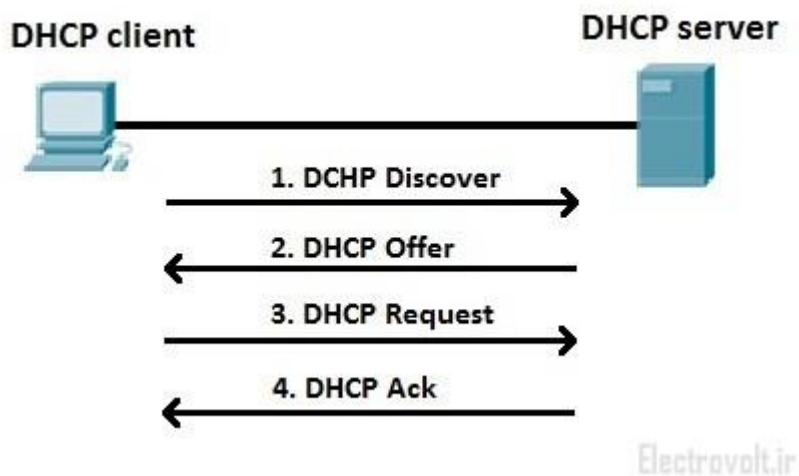
در یک شبکه ممکن است تعداد زیادی کلاینت وجود داشته باشد. از آن جایی که هر کلاینت باید آی پی منحصر به فرد خود را داشته باشد، DHCP (dynamic host configuration protocol مخفف) وظیفه دارد تا آدرس های IP را به صورت متمرکز به کلاینت ها اختصاص دهد. برای اینکه با نحوه عملکرد DHCP آشنا شوید بهتر است تا کلمه DORA را در ذهن خود نگاه دارید. منظور از DORA چهار کلمه: Discover، Offer، Request و Acknowledgment است که مکانیزم های اصلی مربوط به DHCP است.

Discover: هنگامی که یک کلاینت روشن می شود و تنظیمات IP به روی حالت اتوماتیک باشد، در ابتدا یک پیام با نام DHCP Discover را در کل شبکه ارسال می کند.

Offer: هر سرور DHCP که پیام DHCP Discover را دریافت کند یک پیام که محتوای آن آدرس IP و سایر تنظیماتی که برای کلاینت است را در قالب پیام DHCP Offer به سمت کلاینت ارسال می کند.

Request: در صورتی که بیشتر از یک سرور DHCP در شبکه وجود داشته باشد، کلاینت چندین پیام DHCP Offer را دریافت می کند. کلاینت یکی از این پیشنهادها را قبول کرده و پیامی مبتنی بر قبول تنظیمات به سرور ارسال می کند که به این پیام DHCP Request گفته می شود.

Acknowledge: هنگامی که سرور DHCP پیام DHCP Request را دریافت می کند، آدرس IP اختصاص داده شده را به عنوان آدرس در حال استفاده علامت گذاری می کند و یک پیام به نام DHCP Ack به کلاینت ارسال می کند. در صورتی که به هر دلیلی سرور نتواند آدرس IP را به کلاینت اختصاص دهد پیام DHCP Nack را ارسال کرده و کلاینت نمی تواند آدرس IP را به خود اختصاص دهد و دوباره این عملیات چهارگانه را تکرار می کند.



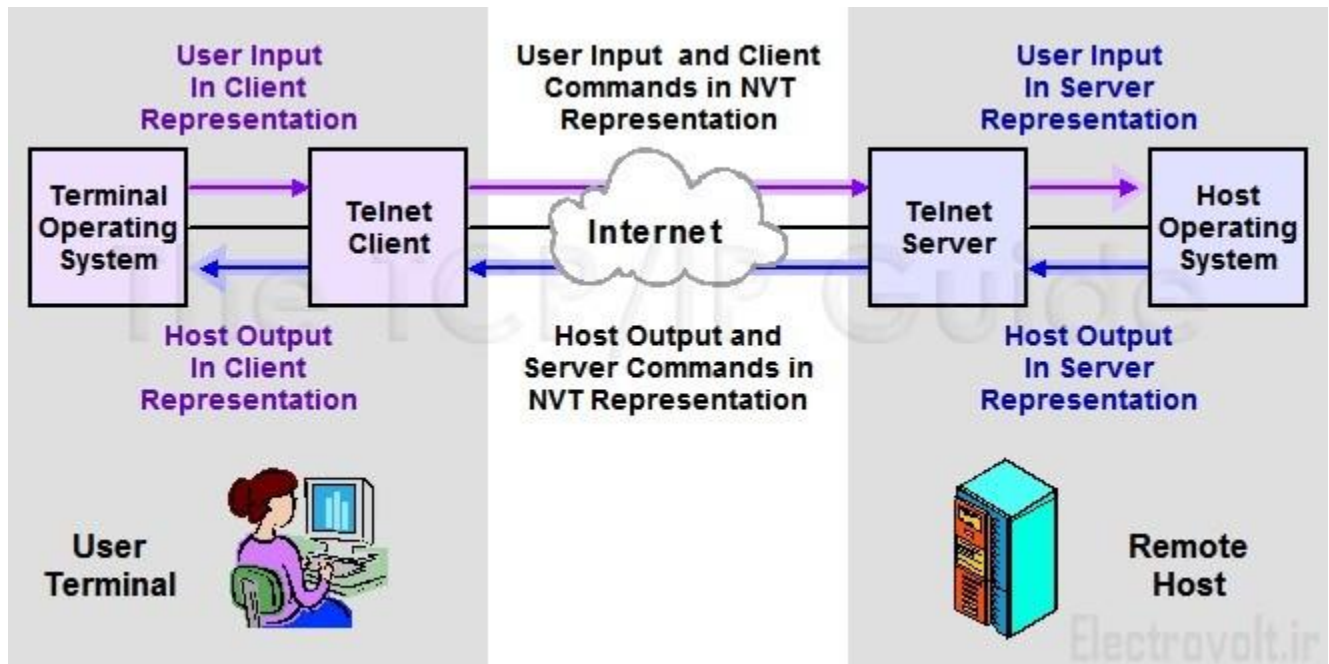
سرویس و پروتکل Telnet

قبل از پیشرفت سیستم عامل ها به صورت گرافیکی، ارتباط بین کاربران و سیستم عامل از طریق محیط های متنی انجام می گرفت. امروزه نیز متخصصان علم کامپیوتر محیط های متنی را به گرافیکی ترجیح می دهند چراکه این محیط ها به دلایل زیر نسبت به محیط های گرافیکی بهتر عمل می کنند:

- سیستم عامل سبک تر خواهد بود و در هنگام کنترل از راه دور نیاز به ارسال داده های کمتری است.
- امنیت در محیط های متنی بیشتر است چراکه اکثر نفوذها از طریق باگ های موجود در محیط گرافیکی صورت می پذیرد.
- پایداری بیشتر و خرابی کمتر اتفاق می افتد چراکه بیشتر خطاها بدلیل وجود محیط گرافیکی می باشد.
- خطاهای انسانی کاهش می یابد و سرعت عملکرد افزایش پیدا می کند.

امروزه نیاز است تا کاربران بتوانند از طریق محیط متنی و از راه دور دستگاه‌ها و سیستم‌ها را مدیریت و کنترل کنند که برای این منظور پروتکل Telnet توسعه داده شده است.

به اتصالی که Telnet از آن استفاده می‌کند (Virtual Terminal) VTی گفته می‌شود. در واقع اتصال Telnet چندین موضوع از جمله چگونگی ایجاد و خاتمه اتصال و نیز دستورات کنترلی مورد نیاز را مشخص می‌کند.



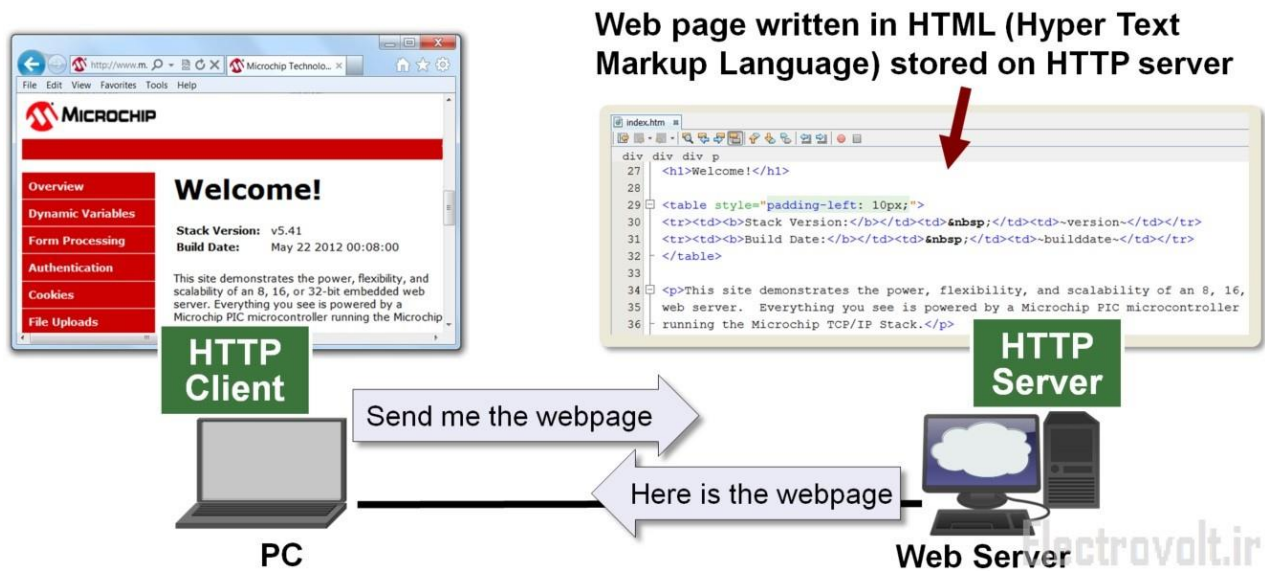
اگرچه Telnet قبل از برقراری اتصال بین کلاینت و سرور عملیات احراز هویت کاربران را انجام می‌دهد اما داده‌های تبادل شده رمزگذاری نشده و به صورت متن ساده ارسال می‌شود که این موضوع یک مشکل امنیتی تلقی می‌شود. یا SSH یا Secure Shell پروتکلی است که کاملاً شبیه به Telnet است اما با این تفاوت که اطلاعات ارسال شده بین کلاینت و سرور رمزگذاری می‌شود و مشکل Telnet را برطرف می‌سازد.

سرویس www و پروتکل HTTP

پروتکل HTTP (مخفف Hypertext Transfer Protocol) یک استاندارد تعریف شده برای شبکه‌های کامپیوتری برای توزیع و نمایش محتوا است که در واقع بعد از ایجاد اولین سایت‌های اینترنتی پایه ریزی شد و به مرور زمان نسخه‌های مختلف آن توسعه پیدا کرد. کنسرسیوم جهانی وب (W3C) مسئولیت توسعه و نظارت بر این پروتکل را بر عهده دارد. مرورگرهای وب و سرورهای اینترنتی برای برقراری ارتباط از این پروتکل استفاده می‌کنند. ارتباط بین کلاینت و سرور در پروتکل HTTP از طریق

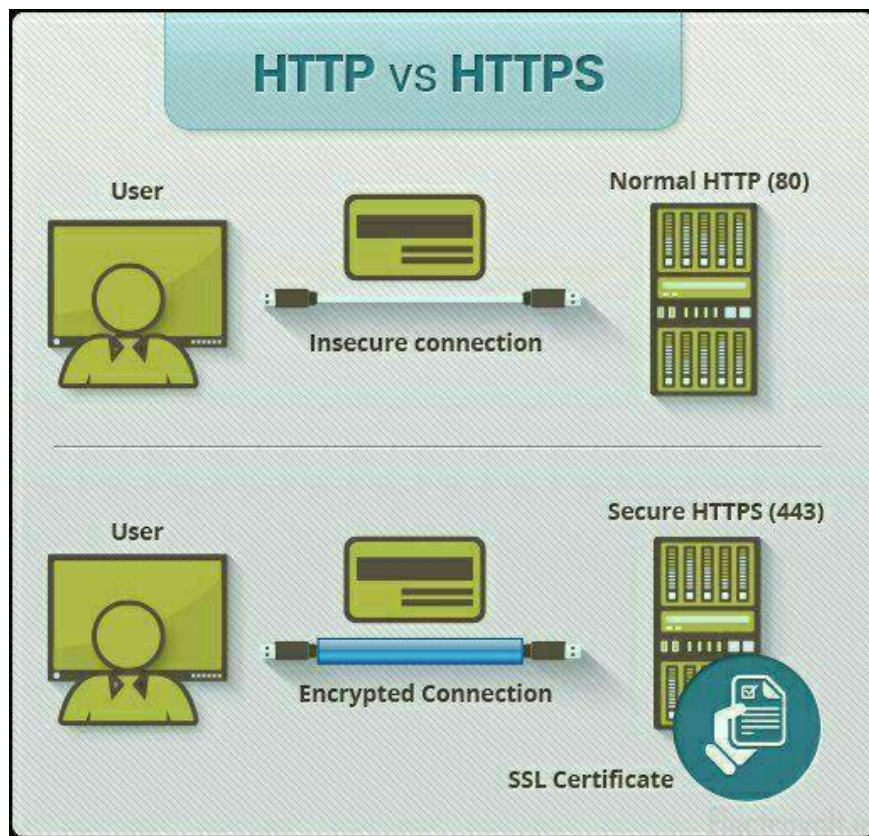
پورت 80 انجام میشود و به مجموعه اطلاعات رد و بدل شده بین دو سیستم نشست یا session گفته میشود. شناسایی این پروتکل آسان است چرا که هنگام بازدید از یک وبسایت، درست در داخل URL وبسایت نوشته شده است. برای مثال آدرس سایت الکترو ولت <https://electrovolt.ir> است. حتی اگر شما <https://> را در ابتدای آن وارد نکنید، مرورگر شما اتوماتیک آن را لحاظ می کند.

پروتکل HTTP بعد از بوجود آمدن زبان برنامه نویسی HTML یا Hyper Text Markup Language در سال 1989 توسط آقای تیم برنرز لی پایه گذاری شد و شروعی برای ایجاد صفحات تحت وب و اولین سایت های اینترنتی بود. این پروتکل ارتباط بین سرور و کلاینت را در یک شبکه کامپیوتری برقرار می کند. در واقع مرورگر اینترنت نصب شده در کامپیوتر کاربر نقش کلاینت را ایفا میکند و با نرم افزاری که در سیستم سرور نصب شده ارتباط برقرار کرده و طبق استانداردهای تعریف شده محتوی را رد و بدل میکند و در این میان درخواست های مکرری از کلاینت به سرور فرستاده شده و پاسخ آن از سمت سرور به کاربر فرستاده میشود. شکل زیر این موضوع را نشان می دهد.



پروتکل HTTPS

پروتکل HTTP دادهها را به صورت متن ساده ارسال و دریافت می کند که این موضوع در هنگام ارسال و دریافت اطلاعاتی که حاوی رمز عبور است مناسب نبوده و ممکن است تا اطلاعات شخصی افراد به مخاطره بیافتد. برای مقابله با این چنین خطراتی یک ارتباطات امن در محیط وب بنام Secure HTTP یا HTTPS به وجود آمد که در این پروتکل دادهها در مبدأ رمزگذاری و در مقصد رمزگشایی می شوند؛ بنابراین دادهها در میان راه از دسترس هکرها در امان هستند. شکل زیر تفاوت بین HTTP و HTTPS را نشان می دهد.



ویژگی های عمومی پروتکل HTTP

سه ویژگی که باعث شده HTTP در عین سادگی قدرتمند باشد عبارت است از:

بدون اتصال بودن (Connectionless)

مستقل بودن (Independent)

بدون وضعیت بودن (Stateless)

بدون اتصال بودن (Connectionless)

Client به وسیله ی مرورگر یک سرویس HTTP درخواست می کند ، Client به server وصل نیست، پس، منتظر پاسخ می ماند Server. درخواست Client را می بیند و پردازش می کند و برای ارسال پاسخ، دوباره یک ارتباط جدید با Client برقرار می کند.

مستقل بودن (Independent)

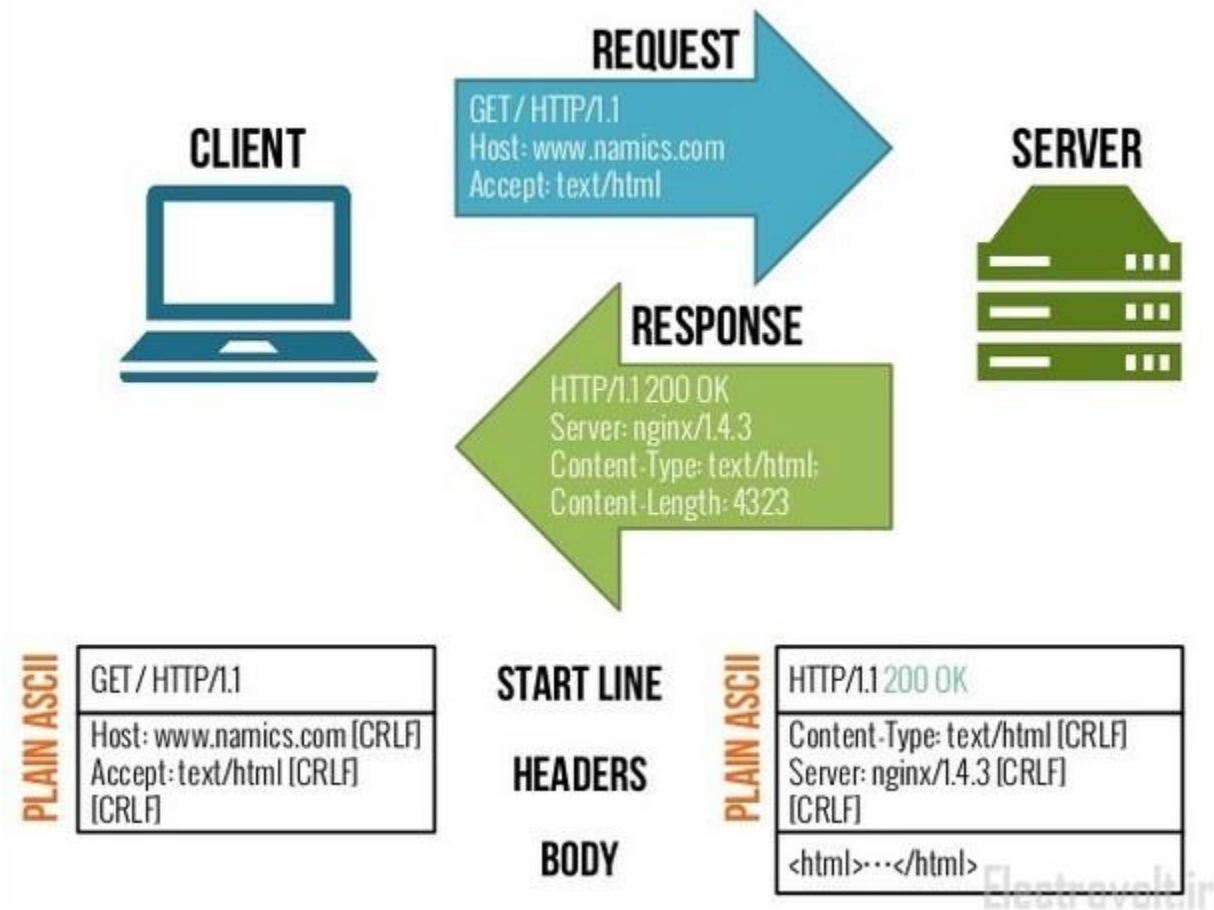
HTTP یک رسانه ی مستقل است، HTTP بر اساس نوع داده ای محتویات ارسالی توسط Client و همچنین نوع داده ای محتویاتی که Server به Client پاسخ می دهد ، یک MIME مناسب درخواست می دهد، این بدان معناست که HTTP می تواند هر نوع داده ای که توسط Client و Server درخواست می شود، بدون هیچ مشکلی فراهم کند و محدود به داده های خاصی نمی باشد، پس کاملا مستقل است.

بدون وضعیت بودن (Stateless)

همان طور ک گفتیم HTTP بدون اتصال می باشد Server. و کاربر Client هر کدام تنها زمانی که آن دو با هم در ارتباط و تعامل هستند از یکدیگر با خبرند. بعد از آن هر دو آن ها، هر چیزی که بینشان بوده فراموش می کنند. با توجه به ماهیت این نوع پروتکل، نه Client و نه مرورگر نمی توانند اطلاعات بین درخواست های یک صفحه ی وب را نگه داری کنند. به خاطر همین موضوع است که داشتن وب سایتی که بصورت هوشمند به ورودی کاربر واکنش نشان دهند (Responsive) سخت است. این کمبود HTTP با کمک تکنولوژی های تازه نظیر Java ، Javascript ، و cookies برطرف شده است.

تشریح عملکرد پروتکل HTTP

در ادامه به تشریح مهمترین پروتکل لایه کاربرد یعنی HTTP خواهیم پرداخت که پایه و اساس شبکه جهانی وب است. پروتکل HTTP در مدل کلاینت/سرور کار می کند. هنگامی که یک کاربر می خواهد به هر صفحه HTTP بر روی اینترنت دسترسی داشته باشد، ابتدا ماشین کلاینت در سمت کاربر یک ارتباط TCP به سرور روی پورت 80 آغاز می کند که به آن HTTP Request گفته می شود. سپس سرور درخواست کلاینت را بررسی کرده و به آن در خواست پاسخ می دهد که به آن HTTP Response گفته می شود. برای مثال یک درخواست وب نمونه و پاسخ آن را در شکل زیر مشاهده می کنید.



نسخه های مختلف پروتکل HTTP

- HTTP 0.9: تنها از دریافت اسناد ابر متن ساده پشتیبانی می کند.
- HTTP 1.0: از HTTP غیر مداوم استفاده می کند. حداکثر یک شی می تواند از طریق یک اتصال TCP ارسال شود.
- HTTP 1.1: از HTTP مداوم استفاده می کند. در این نسخه، اشیاء متعدد می توانند از طریق یک اتصال TCP ارسال شوند.
- HTTP-NG: متوقف شد و به عنوان استاندارد ثبت نشد.
- HTTP 2.0: در این نسخه ضمن سازگاری با نسخه های قبلی، بسیاری از عملکردها بهینه سازی شده است. در شکل زیر کلیه نسخه های HTTP را مشاهده می کنید.

1945 – THE MEMEX SYSTEM	MICROFILM MANAGEMENT
1965 – HYPERTEXT	PROJECT XANADU
1991 – HTTP 0.9	GET ONLY
1996 – HTTP 1.0	HEADERS, MIME, ...
1997 – HTTP 1.1	FLAW CORRECTION
1998 – HTTP-NG	STOPPED
2015 – HTTP/2	RFC PUBLISH DATE

روش های ارسال درخواست HTTP

سه نوع روش (Method) اصلی برای ارسال پیام HTTP وجود دارد که شامل GET ، POST و HEAD می باشند.

- پیام های HTTP GET که به سرور ارسال می شوند تنها حاوی URL می باشند. ممکن است پارامترهای داده اختیاری نیز به انتهای این URL اضافه شود که در این صورت در انتهای URL هر یک از این پارامترها با یک علامت سوال (?) اضافه می گردد. سرور بخش پارامتر داده اختیاری را پردازش می کند، اگر وجود داشته باشد، سپس نتیجه را با در نظر گرفتن آن پارامتر به مرورگر بر می گرداند.
 - در پیام های HTTP POST هر گونه پارامتر داده اختیاری به جای اینکه در انتهای URL و با علامت سوال نوشته شود ، در بدنه پیام و با علامت & گذاشته می شود.
 - درخواست HTTP HEAD همانند درخواست های GET عمل می کند. با این تفاوت که به جای پاسخ دادن با محتوای کامل URL ، سرور تنها اطلاعات هدر پاسخ را به مرورگر بر می گرداند.
- در شکل زیر کلیه متد های ارسال درخواست HTTP را مشاهده می کنید.

SAFE METHODS NO ACTION ON SERVER	}	GET	HTTP/1.1 MUST IMPLEMENT THIS METHOD
		HEAD	INSPECT RESOURCE HEADERS
MESSAGE WITH BODY SEND DATA TO SERVER	}	PUT	DEPOSIT DATA ON SERVER — INVERSE OF GET
		POST	SEND INPUT DATA FOR PROCESSING
		PATCH	PARTIALLY MODIFY A RESOURCE
		TRACE	ECHO BACK RECEIVED MESSAGE
		OPTIONS	SERVER CAPABILITIES
		DELETE	DELETE A RESOURCE — NOT GUARANTEED

هدر درخواست HTTP

هدر در خواست HTTP (یعنی HTTP Request header) به اطلاعاتی گفته می شود که به شکل متنی (Text) هستند و در هنگام درخواست HTTP (یعنی HTTP Request) توسط مرورگر به وب سرور ارسال می شود. این اطلاعات شامل جزئیاتی از درخواست مرورگر می باشد که توسط وب سرور در نظر گرفته می شود. برای مثال اطلاعاتی نظیر نام مرورگر (IE, Chrome, FireFox) نسخه ویندوز (XP, 7, 8, 10) و ... که به سرور کمک می کند تا پاسخ مناسب تری بدهد. سپس سرور از اطلاعات هدر درخواست، برای ساخت و ارسال هدر پاسخ استفاده می کند. همچنین مرورگر موظف است بعد از ارسال هدر درخواست، داده های مورد نیاز (در صورت وجود) را برای سرور ارسال کند. در شکل زیر مثالی از هدر درخواست را مشاهده می کنید.

REQUEST HEADERS

REQUEST HEADER FROM CHROME

GET / HTTP/1.1

Host: namics.com

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_5

Connection: keep-alive

Accept: text/html,application/xhtml+xml,application/xml;

q=0.9,image/webp,*/*:q=0.8

Accept-Encoding: gzip,deflate,sdch

Accept-Language: en-US,en;q=0.8,de;q=0.6

Cookie: my_namics_SSL=1; LtpaToken=AAECAzUOMOI3RTIBNTQzQkZEMkFDTj1

} **ACCEPT HEADERS**
Capabilities + Preferences

Q: QUALITY FACTOR - RELATIVE DEGREE OF PREFERENCE

هدر پاسخ HTTP

هدر پاسخ HTTP (یعنی HTTP Response Header) به اطلاعاتی گفته می شود که به شکل متنی (Text) هستند و در هنگام پاسخ به دریافت یک درخواست HTTP توسط وب سرور به مرورگر کاربر ارسال می شود. هدر پاسخ شامل تاریخ، حجم، و نوع فایلی که به کاربر ارسال می شود نیز می باشد. همچنین اطلاعاتی درباره خود سرور نیز به همراه دارد. سرور موظف است بعد از ارسال هدر پاسخ آن فایل مورد نظر را نیز ارسال نماید. در شکل زیر مثالی از هدر پاسخ را مشاهده می کنید.

RESPONSE HEADERS

RESPONSE HEADER FROM WWW.NAMICS.COM

```
HTTP/1.1 200 OK
Date: Mon, 13 Oct 2014 16:18:08 GMT
Transfer-Encoding: chunked
Connection: keep-alive
Server: nginx/1.4.3
Content-Type: text/html; charset=utf-8
Vary: Accept-Encoding
Age: 1110
X-Mod-Pagespeed: 1.6.29.7-3343
X-Powered-By: Secret Namics rocket technology
X-Varnish: 1491545138 1491540973
Via: 1.1 varnish
```



کدهای وضعیت HTTP

خطاها در اینترنت میتوانند کاملاً خسته کننده باشند، خصوصاً اگر شما تفاوت بین خطای 404 و خطای 502 را ندانید. این پیام های خطا، که کدهای وضعیت HTTP نیز گفته میشوند کدهای پاسخی هستند که توسط وب سرورها داده میشوند و کمک به شناسایی علت مشکل میکند.

برای مثال "404 File Not Found" یک کد وضعیت رایج HTTP است. این ارور بدین معنا است که وب سرور درخواست شما را نمیبیند. فایل -صفحه وب یا دیگر سندی که میخواهید با مرورگر خود آنها را لود کنید- جابجا یا پاک شده و یا اینکه شما آدرس URL اشتباه وارد کرده اید.

دانستن معنای کد وضعیت HTTP به شما کمک میکند تا بفهمید اشکال کار در کجاست. برای مثال در ارور 404 شما آدرس URL را چک میکنید که اگر غلط املایی داشته باشید آنرا اصلاح کنید و دوباره امتحان کنید. اگر درست نشد با پاک کردن اطلاعات بین هر بک اسلش آنقدر به عقب بر میگردید تا به صفحه ای برسید که ارور 404 نداشته باشد. از آنجا احتمال دارد صفحه ای را که دنبال آن هستید پیدا کنید.

در شکل زیر کلیه پیغام های سرور را مشاهده می کنید.

1XX	100-101	INFORMATIONAL
2XX	200-206	SUCCESSFUL
3XX	300-305	REDIRECTION
4XX	400-415	CLIENT ERROR
5XX	500-505	SERVER ERROR

Electrovolt.ir

آموزش الکترونیک برای همه

Electro Volt.ir

FPGA

ARM

AVR

پروژه های الکترونیک

نرم افزارهای الکترونیک

کتاب های الکترونیک



Electrovolt_ir



Electrovolt.ir