

۷ کارنیل، بزرگترین شبکه موفقیت ایرانیان می باشد، که افرادی زیادی توانسته اند با آن به موفقیت برسند، فاطمه رتبه ۱۱ کنکور کارشناسی، محمد حسین رتبه ۶۸ کنکور کارشناسی، سپیده رتبه ۳ کنکور ارشد، مریم و همسرش راه اندازی تولیدی مانتو، امیر راه اندازی فروشگاه اینترنتی، کیوان پیوستن به تیم تراکتور سازی تبریز، میلاد پیوستن به تیم صبا، مهسا تحصیل در ایتالیا، و.... این موارد گوشه از افرادی بودند که با کارنیل به موفقیت رسیده اند، شما هم می توانید موفقیت خود را با کارنیل شروع کنید.

برای پیوستن به تیم کارنیلی های موفق روی لینک زیر کلیک کنید.

[www.karnil.com](http://www.karnil.com)

همچنین برای ورود به کانال تلگرام کارنیل روی لینک زیر کلیک کنید.

<https://telegram.me/karnil>



# اصول طراحی و ساخت نرم افزار میزبانی وب

اصول طراحی و ساخت نرم افزار میزبانی وب  
ISBN: 905-93555-6-4: 15000 ریال  
فهرست نویسی بر اساس اطلاعات فیبا  
عنوان اصلی:  
اصول طراحی و ساخت نرم افزار میزبانی وب - 104 ص  
نویسنده : محمد حسین دالوند  
1.مرجع تخصصی کامپیوتر  
007/43 QA8686 / الف 99 س /  
کتابخانه ملی ایران 12488 - 77 ت

انتشارات مرکز پژوهش و توسعه فن آوریهای نوین انتقال داده

نویسنده : محمد حسین دالوند

شابک : 905-93444-6-4

نوبت چاپ : اول 1386

تیراژ : 3000 جلد

قیمت : 1500 تومان

- استفاده از مطالب مندرجه با ذکر منبع بلامانع است

**بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ**

1. مقدمه
2. سخني با خوانندگان
3. ایده ي طراحي و ساخت وب سرور
4. شبکه چیست ؟
5. اینترنت چیست ؟
6. پروتکل چیست ؟
7. استاندارد در علوم کامپیوتر به چه معناست؟
8. وب چیست ؟ ( آشنایي مختصر با اصول آن )
9. DNS چیست ؟
10. معرفي انواع زبانهاي برنامه نویسي وب سایت
11. منظور از سیستم وب هاستینگ چیست ؟
12. معرفي انواع نرم افزارهاي وب هاستینگ
13. سخت افزار وب هاستینگ چیست و مطلوب آن کدام است
14. استانداردهاي یک وب سرور در RFC
15. ویژگیهاي الگوریتم یک نرم افزار وب هاستینگ
16. معرفي Socket programming یا برنامه نویسي پریز
17. انواع مفهوم پردازش برنامه
18. انواع مفهوم دسترسی به محتویات فایل
19. برنامه نویسي به روش انتقال مفهوم ( استفاده از روش پردازش در دسترسی به فایل )
20. Pre-trusted-connection

تقدیم به درگاه خداوند بلند مرتبه که علم را عامل پیشرفت  
بشر  
و تقرب به درگاه خویش معرفی نمود

و تقدیم به پدر گرامیم که استواری در برابر مشکلات را به من  
آموخت  
و تقدیم به مادر عزیزم که مهربانی و عشق را به من  
آموخت

و تقدیم به خواهران مهربانم

امروزه با پیشرفت علوم طبیعی و در کنار آن گسترش روز افزون اطلاعات و در دسترس بودن ابزاری به نام کامپیوتر و به طبع آن ابداع روزمره ی سیستمهای نوین ارتباطی مانده شبکه های مخابراتی و کامپیوتری این امکان به وجود آمده است که بتوان از طریق یک شبکه ی گسترده و بدون مرز یا بین المللی اطلاعات حاصله از گسترش علوم طبیعی و یا کسب و کار را در محیطی مجازی در اختیار همگان و یا گروهی خاص قرارداد.

در اختیار قرار دادن این اطلاعات در یک کلام انتشار آن در این محیط مجازی می باشد - اما روشهای زیادی برای انتشار اطلاعات در محیط مجازی وجود دارد. برای مثال قرار دادن اطلاعات بر روی میزبانهای بارگزاری فایل ( FTP SERVERS = File transfer protocol servers ) و یا انتشار اسناد از طریق پست الکترونیکی به فهرستی از اشخاص ( E-mail ) و یا طراحی یک وب سایت و انتشار اسناد بر روی آن ( Web designing ) که روش آخر جزو بهترین راهها می باشد- چرا که :

- دیگران بدون محدودیت در مکان و زمان می توانند به اطلاعات شما دسترسی داشته باشند
- افراد می توانند از طریق جستجو در اینترنت به اطلاعات شما دسترسی پیدا کنند
- امکان به روز رسانی اطلاعات ( Updating ) به سادگی
- امکان دریافت نظر سنجی و Feedback ( بازخورد )
- امکان ارتباطی ساده با بازدید کنندگان وب سایت
- و ...

از طرف دیگر تنها طراحی وب سایت باعث انتشار اطلاعات بر روی شبکه نمی شود و نیاز است که اطلاعات بر روی یک فضای مجازی به نام WEB SITE منسوب به فضای واقعی به نام WEB HOSTING که به صورت فیزیکی بخشی از دیسک سخت یک دستگاه کامپیوتر دارای نشان شبکه ای متصل به شبکه می باشد قرار گیرند.

سیستم عامل ( مانده UNIX و MS-Windows ) و سپس نرم افزار Web hosting می باشد - که مبحث اصلی این کتاب نرم افزار WEB HOSTING ویا همان نرم افزار میزبان وب می باشد.

امید است که خوانندگان محترم پس از مطالعه ی این کتاب به اهداف ذیل نایل آیند:

- آشنایی کلی با سیستم اینترنت
- آشنایی با مفاهیم پایه ای برنامه نویسی تحت شبکه
- آشنایی با برنامه نویسی پرز یا Socket
- آشنایی با مفهوم Web server
- آشنایی با الگوریتمهای متداول Web server
- آشنایی با نرم افزار MHD WEB SERVER
- توانایی در تشخیص بهترین WEB SERVER برای یک WEB SITE
- توانایی در طراحی و برنامه نویسی یک نرم افزار میزبان وب



در جامعه ی ما امروزه این اصطلاح متداول شده است که چرخ را نباید از ابتدا ساخت. اما به نظر این بنده ی حقیر کسی که بداند چرخ را چگونه می سازند و تولید آنرا حتی یکبار تجربه کرده باشد می تواند بهترین چرخ را تولید نماید.

اگر یک مهندس طراحی خودرو نداند که مکانیسم تولید چرخ چه است و تصویری از آن نداشته باشد - مطمئناً خودرویی تولیدی وی با مشکلاتی در حرکت رو به رو خواهد شد. مهندسی که بداند چرخ چگونه تولید می شود و درک درستی از آن داشته باشد - در صورت نیاز می تواند به راحتی بیان نماید که چه نوع چرخ مناسب خودرویی وی است و این کار را با کمترین هزینه و بالاترین کیفیت انجام خواهد داد.

برای مثال - اگر چرخ را مجموعه ای از رینگ و لاستیک در نظر بگیریم - و مهندسی بداند که رینگ را فلان کارخانه از جنس فلز A می سازد و مراحل ساخت آن سه مرحله دارد و فلان کارخانه از جنس فلز B می سازد و مراحل ساخت آن یک مرحله است و بنا بر اصول طراحی رینگ بداند که برای مثال هنگام ساخت رینگ اگر مراحل آن کمتر باشد بهتر است و جنس فلز A بهتر است به راحتی می تواند سفارش ساخت رینگ مشتری یا Customized ring را به کارخانه ای بدهد که رینگ را از جنس فلز A می سازد - و از آنها بخواهد که رینگ را در یک یا دو مرحله بجای سه مرحله بسازند.

دقیقا" همین اتفاق در دنیای کامپیوتر رخ می دهد. اگر برنامه نویسی که می خواهد یک برنامه ی حسابداری برای یک شرکت بزرگ بسازد و در این میان می داند که به بانک اطلاعاتی برای نرم افزارش نیاز دارد - در صورتی بهترین انتخاب را می کند - اگر بداند که یک نرم افزار بانک اطلاعاتی را چگونه می سازند و با اصول جستجوی داده و ... و نحوه ی دسترسی به فایل و داده در محیط یک سیستم عامل آشنا باشد.

اطلاعاتي به راحتی تصور کند.

مثالي ديگر. خودرويي دچار نقص فني در سيستم روشنائي مي شود. يك فرد مکانیک که به صورت تجربي با خودروهاي بسياري سر و کار داشته - همواره اساس کار سيستم روشنائي همه ي خودروها را به صورت ساده مي داند. اما يك فرد که صرفاً "راهنماي يك خودرو را مطالعه نموده است - به اندازه ي فرد مکانیک نمي تواند عيب خودرو را به راحتی شناسايي و حتي رفع نمايد.

براي مثال - اگر سيستم چراغ دنده ي عقب يك خودرو خراب شود و مدام روشن بماند - افراد عادي مي دانند که جايي خراب شده است - فردي که کمي اطلاعات دارد - مي داند که مشکل از کلید جعبه دنده است - فرد نماينده ي رسمي کارخانه مي داند که بايد آن کلید تعويض شود - اما هيچک نمي دانند قبل از رسيدن فرد به محل تعمير چگونه اين مشکل بر طرف شود؟!

چون کل سيستم روشنائي خودروي مفروض ما داراي يك فيوض است - و اگر آن فيوض خارج شود کل سيستم روشنائي از کار مي افتد - از طرفي نور چراغ دنده عقب شب هنگام در جاده خطر ناک است. پس چه بايد کرد؟

فردي که با سيستم طراحي خودرو به طور ساده آشنا باشد - مي داند که يك فيش ( Port ) در پشت لامپهاي دنده عقب خودرو است و به راحتی با جدا کردن آن مي توان آنرا خاموش نمود.

برنامه نويسي که بخواهد از امکانات آماده استفاده نمايد بدون آنکه نحوه ي کار آنها با بداند - مانند راننده اي است که شب هنگام محتاج مکانیک و تعمير کار خودرو است. اگر مشکلي در برنامه پيش آيد - و از نظر او کدها درست باشند - نمي داند که مشکل از سيستم عامل است و يا سخت افزار کامپيوتر!

طراحی کرد - زیرا - ممکن است روزی ما نیز مجبور شویم چرخ را بسازیم.

دانستن پایه ای و اساسی ما را یاری می کند در :

- برنامه ریزی صحیح جهت تولید
- عیب یابی دقیق
- پیشگیری از مشکلات
- سرعت بیشتر در طراحی و ساخت
- پیش بینی وضعیتهای بحرانی
- پیش بینی میزان اعتماد به نرم افزار
- شبیه سازی کارایی نرم افزار
- امکان بازنویسی برنامه به زبانهای مختلف جهت سازگاری با سیستم عاملهای مختلف
- امکان پیاده سازی ایده های جدید

انشاءالله که مطالب ذکر شده در این کتاب بتوانند به شما در جهت دانستن اصول طراحی چرخ در حیطه ی وب و شبکه کمک کنند و شما را یاری نمایند که تولید آنرا در این حیطه تجربه نمایید.

## : Web server

همانطور که در مقدمه این کتاب بدان اشاره شد - امروزه نشر اطلاعات به عنوان یکی از مهمترین ارکان پیشرفت و توسعه می باشد. امروزه اشخاص می توانند از طریق وب به انتشار افکار خود در محیطهایی مانند Web log بپردازند و شرکتها و سازمانها جهت تسهیل در ارتباط با مشتریان و اربابان رجوع وب سایت تاسیس می کنند و در آن اطلاعات مفیدی شامل آدرس شرکت و یا سازمان - خدمات و ... قرار می دهند.

دانشجویان و محققان از طریق وب اقدام به جستجوی مقالات می کنند و بابت این مسئله از پایگاههایی به نام Search engine استفاده می نمایند. این پایگاهها به آنها این امکان را می دهد که از طریق جستجوی کلید واژه ی خاص نتایجی را دریافت نمایند که آنها را به پایگاه اطلاع رسانی یا وب سایتی مرتبط می سازد و می توانند از طریق آنها و معمولاً" به صورت رایگاه اقدام به تحقیق و بررسی نمایند.

در این میان فروشگاههای اینترنتی وجود دارند که به کاربران این امکان را می دهند که بدون صرف هزینه ی رفت و آمد و فقط با استفاده از چند پیوند ( URL LINK ) و با استفاده از کارت اعتباری پرداخت ( Credit card payment ) اقدام به خرید و سفارش کالای مورد نظر خود نمایند.

برخی از پایگاههای اطلاع رسانی ( Web sites ) این امکان را به افراد می دهد که به صورت روی خط ( Online ) اقدام به مطالعه ی اخبار روزانه نمایند و یا به دانشجویان این امکان را می دهد که به صورت روی خط نمرات خود را بر روی وب سایت دانشگاه مشاهده نمایند.

چند وقتی است که در ایران نیز این سیستم راه اندازی شده است که به دانشجویان این امکان را می دهد که از طریق وب سایت دانشگاه اقدام به انتخاب واحد نمایند و حتی شهریه ی خود را از طریق وب سایت پرداخت نمایند و مطمئناً" همانطور که می دانیم که

شبکه ی اطلاع رسانی جهانی و یا Internet است.

در این راستا – این واقعیت احساس می شود که راه اندازی و استقبال استفاده از چنین پایگاههایی به طور روز افزون در حال افزایش است زیرا:

- هزینه ی دسترسی به اینترنت روز به روز در حال کاهش است
- شرکتهایی که خدمات خود را تحت وب ارائه می نمایند در حال افزایش هستند
- در هر ساعتی از شبانه روز می توان به وب سایت یک یا چندین شرکت دسترسی داشت
- دسترسی به یک وب سایت از هر کجای دنیا امکان پذیر است
- اطلاعاتی را که می توان از یک وب سایت بدست آورد معمولا" کاملتر از اطلاعاتی است که به طور حضوری می توان دریافت کرد
- مطالب یک وب سایت قابل ذخیره نمودن و انتشار به غیر است

و ...

در چنین وضعیتی و به دلیل جدید بودن این فن آوری یک رقابت مجازی فی ما بین شرکتها و سازمانها در جهت تجهیز شدن به این فن آوری به وجود می آید. که اهداف ذیل را مورد توجه قرار می دهند :

- قرار گرفتن URL وب سایت بر روی سربرگها و کارت ویزیتهای شرکت و سازمان
- داشتن آدرس پست الکترونیکی برای کارمندان و مدیران شرکت و سازمان
- اهداف تبلیغاتی

و ...

که در جهت مجهز شدن به این فن آوری معمولا" مطالب ذیل در عموم اهداف دخیل است :

- زیبایی در طراحی وب سایت
- چند زیانه بودن وب سایت
- On بودن و یا دسترس پذیر بودن وب سایت در طول 24 ساعت و 7 روز هفته
- زمان کم در طراحی وب سایت

و ...

و از آنجا که اطلاعات دقیقی در مورد مسایل امنیتی یک وب سایت وجود ندارد و عمده ی کارفرمایان فقط می دانند که این مسایل مهم هستند - از اینرو به دانستن و داشتن اطلاعاتی کافی در مورد این مسایل زیاد اهمیت نمی دهند و از طرف دیگر به دلیل آنکه تا به امروز ( تاریخ انتشار این کتاب ) تجربه ای در خصوص راه اندازی خدمات میزبانی وب ( Web hosting ) در کشور ما ایران به صورت رسمی وجود نداشته است و عمده ی پیشرفتهای در خصوص اجاره ی سرور در فلان کشور و در فلان مرکز نگهداری اطلاعات ( DATA CENTER ) بوده است . از اینرو کارفرمایان بی خبرند اگر به مسایل امنیتی توجه نداشته باشند :

- ممکن است سهم زیادی در بازار را از دست بدهند
- ممکن است یک یا چند جوان با نرم افزارهای ساده و اطلاعاتی جزئی بتوانند ساعتها وقت آنها را بگیرند
- ممکن است از وب سایت آنها سو استفاده شود

و ...

متأسفانه از آنجا که مفهوم تجارت الکترونیکی هنوز در کشور ما نوپا و جوان است - هنوز این خطر برای کارفرمایان محسوس نمی باشد که از دست دادن اعتبار در یک محیط مجازی به چه قیمتی برای وی تمام خواهد شد و اگر بداند صرفاً" به دلایل ذیل مجبور است به مجری طراح وب سایت اعتماد نماید:

- هزینه ی طراحی
- گفته های مجری
- تبلیغات هزینه های رقابتي Hosting ویا همان میزبانی وب
- گفته های دیگران
- ادعاهای تبلیغاتی

اما متأسفانه در کشور ما حتی برخی موسسات دولتي نیز قربانی حملات امنیتی شده اند و اعتبار اجرایی خود را مدتها به طول انجامیده است تا بازایند.

برای مثال - اگر کارفرمایی بخواهد با هزینه ی پنجاه هزار تومان وب سایت داشته باشد - به گفته ی یک مجری می داند که بخشی از آن صرف اجاره ی سالیانه نام وب سایت و یا همان Domain name می شود و بخش دیگر آن صرف اجاره ی فضا و یا همان Web site می شود. تنها بخشی که نیاز به تمدید سالیانه ندارد بخش طراحی وب سایت است مگر آنکه قرارداد دیگری به نام پشتیبانی منعقد شده باشد.

در بازار میزبانی وب - با مبلغ بیست هزار تومان حتی در برخی مواقع می توان یک خدمات میزبانی وب در کشور ایالات متحده آمریکا پیدا کرد که تا سقف پانصد مگابایت فضا در اختیار کاربر قرار می دهد. اما برای یک شرکت معتبر چقدر تضمین برای موارد ذیل وجود دارد :

- آن شرکت میزبانی روزی به دلایل سیاسی وب سایت او را مسدود ننماید
- آن شرکت میزبانی در سالهای آینده نرخ خود را ثابت نگهدارد
- تمامی مراحل مورد نیاز کارفرما بر روی میزبانهای آن شرکت میزبانی قابل اجرا باشند ( حال و در آینده - برای مثال خدمات بانک اطلاعاتی - )
- این فضا به راحتی قابل افزایش باشد

- مجاز جهت انتشار مطالب دیگر و بعضاً " غیر اخلافي استفاده ننماید ( متاسفانه نمونه هايي از این موارد در خود ایالات متحده رخ داده شده است )
- درصد On بودن میزبان

و ...

برخي از مسولان داخلي کشور جمهوری اسلامی ایران با توجه به مسایل مطروحه ي بالا - و در جهت درستي سعي بر آن داشته اند و دارند که خدمات میزباني وب را در محلهايي به نام مرکز ملي نگهداري داده هاي اینترنتی و يا National Internet Data Center نگهداري نمایند. ضمن آروزي موفقیت براي ایشان - باز نیز چندین نکته وجود دارد :

- استانداردهايي که در مورد نرم افزارهاي میزباني وب استفاده مي شوند عموماً " ساده هستند و عام پسند و شرکتهای خاص نمی توانند از آنها بهره کامل ببرند
- برخي از شرکتهای مایل هستند حتي اطلاعات تجاري خود را بر روي میزبانها قرار دهند و ترس از امنیت اطلاعات آنها را به سمت میزبانهاي خارجي سوق مي دهد
- و ...

با توجه به تمامی این مسایل مطرح شده اینطور به نظر مي رسد که باید راهکاري وجود داشته باشد که برخي شرکتهای بر حسب نیاز خود بتوانند سرویس میزباني خود را در محل شرکت خود راه اندازي نمایند - و این امکانات باید حداقل شرایط ذیل را داشته باشد :

- حد اقل نیاز سخت افزاري
- ارزان قیمت ( قابلیت جبران هزینه هاي استهلاک )
- سریع
- کاربري آسان
- امنیت بالا



- هزینه ی ارزان نگهداری
- پایداری بالا

اما تا به امروز مشکلات اساسی در اجرای این طرح وجود داشته است :

- سرعت پایین اینترنت به دلیل صرفه جویی در هزینه ها
- عدم وجود نرم افزار ساده در زمینه ی میزبانی وب
- نیاز به دانش فنی بالا در زمینه ی میزبانی وب
- عدم آشنایی مسولان انفورماتیک شرکتها با مسایل امنیتی میزبانی وب
- مشکلات پشتیبانی در زمینه ی میزبانی وب

و ...

برای مثال - یک شرکت تجاری قادر است در ماه هزینه ای نزدیک به یکصد تا پانصد دلار را بابت مصرف اینترنت داخلی خود بپردازد و معمولا" در این زمینه از سرویسهای ADSL و یا Point to point و یا ماهواره ای استفاده می شود.

اگر همچین شرکتی بداند که می تواند با استفاده از چنین پهنای باندي سرویس میزبانی وب خود را راه اندازی نماید مطمئنا" برایش بسیار مقرون به صرفه خواهد بود با توجه به مسایل مطرح شده در چند صفحه ی گذشته.

اما متاسفانه با نرم افزارها ی موجود در بازار این امر امکان پذیر نمی باشد. از طرف دیگر کار با نرم افزارهای موجود در بازار نیاز به دانش فنی - نرم افزاری خاصی دارد - که شاید آموزش برخی از آنها نیاز به ساعتها برنامه ریزی و کسب تجربه دارد.

از اینرو اینجانب مطالعاتی را انجام دادم تا بتوانم یک نرم افزار ویژه در این زمینه سازم - اما - در این حین اطلاعاتی را در این زمینه به

انها مي تواند به ديكران در زمينه ي توليد نرم افزارهاي مشابه و حتي بهتر از آن كمك نمايد.

فرض را بر این بگیریم که شما در منزل و یا محل کار خود دو دستگاه رایانه در این اتاق دارید. می خواهید یک فایل حاوی یک متن را به کامپیوتر دیگر انتقال دهید. چند راه وجود دارد ؟

- عیناً تایپ همان متن در رایانه ی دیگر
- استفاده از فلاپی دیسک در انتقال اطلاعات
- نوشتن بر روی یک لوح فشرده نوری ( Write a CD )
- استفاده از یک کابل Parallel متصل به دو کامپیوتر جهت انتقال اطلاعات
- استفاده از کارت شبکه و سیستم LAN
- اتصال دستگاه اول به شبکه ای مانند اینترنت و ارسال پست الکترونیکی به آدرس خودمان که همان فالیل بدان پیوست شده باشد و سپس اتصال با دستگاه دوم به همان شبکه و بارگذاری یا دانلود همان فایل

و ...

تعجب نکنیم - تمامی راههای فوق در اصل به یک منظور صورت می پذیرد و آن انتقال اطلاعات است. شاید صورت انجام برخی با برخی دیگر متفاوت باشد اما در اصل همه یک کار را انجام می دهند و از یک اصل پیروی می کنند و آن همان اصل انتقال اطلاعات از نقطه ای به نقطه ی دیگر است.

بر اساس معیارهای ذیل می توان بین روشهای بالا تفاوت قایل شد:

- هزینه ی جابه جایی اطلاعات
- سرعت جا به جایی اطلاعات
- زمان حصول اطمینان از صحت اطلاعات

شاید دو مورد اول در اکثر مقالات و متون علمی - کامپیوتری به وفور اشاره شده باشند - اما مورد آخر - یعنی زمان اطمینان از صحت

ام انرا به مباحث انتقال اطلاعات اضافه نمایم ، و این بخش به صورت واضح در فصل مربوط به Pre-trustet-connection توضیح داده می شود.

منظور از هزینه ی جا به جایی اطلاعات مقدار هزینه ای است که برای جابه جایی هر بیت از اطلاعات خود می پردازید. برای مثال در مورد اول - یعنی عیناً" تایپ همان مطلب در رایانه ای دیگر شما نیاز به صرف وقت زیاد و شاید استخدام تایپیست دارید. اگر یک تایپیست برای تایپ هر صفحه 200 تومان از شما دریافت نماید و هزینه ی ایاب ذهاب این تایپیست نیز بر عهده ی شما باشد و این هزینه در حدود 3000 تومان باشد - و متن شما در کل دارای 1500 کلمه باشد - پس شما 3200 تومان بابت انتقال این متن پرداخت نموده اید.

در مورد دوم یعنی استفاده از فلاپی دیسک در انتقال اطلاعات - به فرض اینکه دستگاه دوم در اتاق در کنار دستگاه اول باشد تنها هزینه ی خرید فلاپی مد نظر است که قیمتی بین 150 تا 300 تومان دارد.

در مورد سوم - یعنی نوشتن اطلاعات بر روی CD یا لوح فشرده - مانند استفاده از فلاپی تنها نیاز به خریداری یک لوح فشرده و نوشتن اطلاعات بر روی آن است ( البته اگر از هزینه ی خرید Cd-Rom و Cd-Writer صرف نظر شود ). قیمت یک لوح فشرده هم اکنون در حدود 250 تومان است.

در تمامی موارد بالا یک نقص کلی در مورد هزینه وجود دارد. فلاپی دیسک امکان دارد خراب شود و نیاز به خریداری فلاپی جدید باشد - و اگر استفاده مفید از یک فلاپی را 5 بار در نظر بگیریم - پس از هر 5 بار انتقال اطلاعات باید به طور میانگین مبلغی در حدود 200 تومان هزینه نماید.

در مورد CD قضیه کمی سخت تر است. زیرا CD بجز انواع خاصی برگشت نا پذیر است و تنها یک بار و در صورت بازگذاشتن انتهای آن - تا تکمیل شدن ظرفیت آن می توانید از آن استفاده نماید. به طور

تومان هزینه می نمایید.

در مورد تایپ مطلب نیز همینطور است. یعنی این هزینه فقط یکبار و برای هر صفحه باید صورت پذیرد.

استفاده از کابل پارالل مورد دیگری است که هزینه ی آن برابر با 1500 تومان است. این هزینه فقط یکبار انجام می پذیرد زیرا خود کامپیوتر شما مجهز به درگاه LPT می باشد و دو سر کابل را به دو دستگاه متصل می نمایید و سپس با استفاده از امکاناتی مانند Direct Connection در Ms-Windows 2000 و ... می توانید دو کامپیوتر را به یکدیگر متصل نمایید و اقدام به انتقال اطلاعات نمایید.

هزینه ی صرف شده ی شما به صورت جدول نزولی ذیل صرف می شود :

ردیف هزینه	نیاز به خرید کابل جدید	مبلغ صرف شده در خرید	هزینه ی انتقال هر صفحه از متن با توجه به هزینه ی اولیه	درصد صرفه جویی
1	بله	1500 تومان	1500 تومان	0%
2	خیر	0	0	100%
3	خیر	0	0	100%
...	...	...	...	...
...	...	...	...	...

همانطور که مشاهده نمودید تنها هزینه ای که شما صرف می کنید - هزینه ی خرید کابل است و پس از آن دیگر اطلاعات شما به صورت رایگان انتقال می یابند. می توان گفت یکی از روشهای واقعا " کار آمد در انتقال اطلاعات در محیطهای کوچک می باشد - که شرکتهایی

عاملهاي خود کنجانه اند.

روش ديگر استفاده از کابل LAN و سيستم شبکه ي LAN است ( LAN = Local Area Network ) - سيستم شبکه ي محلي يا LAN به شما اين امکان را مي دهد که از طريق یک کابل شبکه بجاي اينکه مانند کابل پارالل تنها بتوانيد دو دستگاه کامپيوتر را به هم متصل نماييد - بتوانيد چندین دستگاه ( حد اقل 2 الي 8 دستگاه و به طور میانگين 255 دستگاه و حد اکثر نامعلومي ) از دستگاهها را به هم متصل نماييد.

هزينه ي انتقال اطلاعات در اين سيستم نیز دقيقاً" به مانند سيستم پارالل است و تنها هزينه ي اوليه ي آن کمی بيشتر است. براي مثال - شما بايد ابتدا دو دستگاه کامپيوتر خود را به کارت شبکه ي LAN مجهز نماييد، سپس مقداري کابل شبکه با قيمتي در حدود متري 80 تومان خريداري نماييد و سپس دو کامپيوتر را به هم متصل نماييد.

در اين سيستم شما اگر بيشتر از دو کامپيوتر را بخواهيد به هم متصل نماييد - تنها کافي است یک دستگاه HUB SWITCH خريداري نماييد که در اندازه هاي مختلفي مي باشند. اين دستگاه به مانند جعبه ي تقسيم برق عمل مي نمايد. استفاده از اين دستگاه نیز اصولي دارد که از حوصله ي اين کتاب خارج است. اما پيشنهاد مي کنم براي کسب اطلاعات بيشتر به کتاب FCC ( Fundamental concepts of computing از انتشارات NCC International ) و يا Networking Essentials از انتشارات Microsoft مراجعه نماييد.

### **به مفاهيم LAN در ادامه ي مطالب نیاز دارید**

سيستم بعدي اتصال دستگاه اول به شبکه اي مانند اينترنت است و سپس ارسال ايميل. اين سيستم همانطور که در بخشهاي بعدي توضيح داده خواهد شد در واقع همان سيستم شبکه ي LAN است که به طور خاصي گسترده شده است و آنرا در مقیاسي WAN و در مقیاسي ديگر MAN مي نامند.

مقابل LAN به معنای Local Area Network (محیط شبکه ی محلی )  
 و در راستای MAN به معنای Main Area Network (محیط شبکه ی  
 اصلی ) .

در واقع WAN تشکیل شده از چندین LAN و MAN تشکیل شده از  
 چندین WAN است.

اما این سیستم برای شما هزینه بردارد است - یعنی اتصال به شبکه  
 و ارسال ایمیل به چند دلیل که قبل از توضیح آن می بایستی اصول  
 کار این سیستم توضیح داده شود :

برای اتصال به شبکه ای مانند اینترنت باید مراحل ذیل طی شود :

- اتصال به شبکه مستلزم استفاده از خط تلفن است و هر دقیقه استفاده از این خط برای شما هزینه دارد
- اتصال به شبکه نیازمند آن است که فرد یا شرکتی این مسوولیت را قبول کند که یک دستگاه کامپیوتر دارای مودم متصل به خط تلفن در 24 ساعت شبانه روز برای برقراری تماس شما و دیگران روشن نگهدارد.
- ارسال ایمیل مستلزم آن است که دستگاه کامپیوتری دیگر در آنسوی شبکه به صورت 24 ساعته روشن باشد و وظیفه ی دریافت و ارسال ایمیل شما را بر عهده گیرد که آن نیز به مانند شرکتی که وظیفه ی روشن نگهداشتن کامپیوتر همراه با دستگاه مودم برای اتصال شما به شبکه نیز این وظیفه را دارد.
- معمولاً شرکت و یا فرد دیگری وظیفه ی اتصال بین کامپیوتر اول که شما از طریق خطوط تلفن با آن رابطه برقرار می کنید و میزبان ایمیل یا کامپیوتری که وظیفه ی ارسال و دریافت ایمیل شما را بر عهده دارد - را بر عهده دارد.

هر یک از مراحل بالا در صورت وجود - کم و زیاد - می توانند دارای هزینه باشند و یا نباشند - و به صورت میانگین در حال حاضر هر ساعت استفاده ی شما از شبکه اینترنت که در بخش بعدی

برای شما در بر خواهد داشت علاوه بر هزینه ی تماس تلفنی که اخیراً سیستمهای زیادی جایگزین ارتباط Dial-up شده اند مانند ADSL و WLAN و WiFi و ... .

استفاده از مورد آخر – یعنی ارسال و دریافت ایمیل و روشهای تحت شبکه ای مانند روشهای اینترنتی مانند WEB و FTP و E-mail و ... در موارد زیادی کاربرد دارند و در برخی موارد با عنایت به روشهای اقتصادی بسیار مقرون به صرفه تر هستند. برای مثال – اگر شما درک صحیحی از شبکه ی اینترنت و ساختمان آن داشته باشید به راحتی می توانید حدس بزنید که ارسال همان یک صفحه فایل تایپ شده از شهر تکزاس ایالات متحد به شهر مسکو چقدر ارزان تر از حتی ارتباط تلفنی ساده بین دوشهر می باشد.

بر همین اساس است که امروزه سیستمهای تلفن – اینترنتی نیز راه اندازی شده اند که توضیح اصول آن از حوصله ی این کتاب خارج است.

پس از بررسی هزینه ی جا به جایی اطلاعات نوبت به سرعت جا به جایی اطلاعات می رسد. در مورد اول یعنی تایپ اطلاعات – مطمئناً شما هم با من هم عقیده هستید که زمان زیادی می برد. این زمان را 5 دقیقه در نظر بگیریم ( توجه داشته باشید که فعلاً دو دستگاه در کنار یکدیگر هستند ).

حال اگر بخواهیم اطلاعات را از طریق یک فلاپی دیسک 1.44 مگابایتی انتقال دهیم می بایستی حد اقل 30 ثانیه زمان برای نوشتن اطلاعات بر روی فلاپی صرف کنیم و سپس 10 ثانیه انتقال آن به دستگاه دیگر و سپس در نهایت مجدداً " حد اقل 30 ثانیه جهت بارگذاری اطلاعات. در نهایت می توان این نتیجه را گرفت که انتقال 1.44 مگابایت اطلاعات از یک کامپیوتر به کامپیوتر دیگر از طریق فلاپی در حدود 2 دقیقه زمان نیاز دارد. در مقایسه با تایپ اطلاعات رقم خیلی خوبی است.



دقیقه به طول انجامد و زمان انتقال آن نیز در نهایت 1 دقیقه به انضمام خواندن آن - بازهم زمان قابل توجهی است. چون بر روی CD بر خلاف فلاپی تا 650 مگابایت اطلاعات یعنی تقریباً 450 برابر فلاپی دیسک را می توان در نهایت 5 دقیقه انتقال داد.

در صورت استفاده از کابل پارالل می توان 30 کیلوبایت را در هر ثانیه انتقال داد - در واقع 9 مگابایت در 5 دقیقه. به نسبت فلاپی رقم خوبی است - اما به نسبت CD واقعا پایین است - اما هزینه اش صفر است!

در صورت استفاده از LAN تا رقمی در حدود 1 مگابایت در ثانیه خواهیم داشت. یعنی 300 مگابایت در 5 دقیقه. البته نوعی از این سیستم نیز در بازار موجود است که سرعت آن تا 10 مگابایت در ثانیه نیز می رسد. یعنی 3 گیگابایت در 5 دقیقه. به نظر می رسد رقیب جدی برای CD محسوب می شود با توجه به اینکه هزینه ی آن نیز صفر است.

در مورد استفاده از ارتباط شبکه ای - سرعت انتقال اطلاعات در آن تقریباً برابر 4 کیلوبایت در ثانیه. یعنی در 5 دقیقه 1.2 مگابایت. تقریباً برابر با فلاپی؟ کمی کمتر! اما در مسافتهای زیاد و هزینه ی کم آن جبران می شود.

به طور کلی می توان رابطه ی هزینه و سرعت انتقال اطلاعات را به این صورت بیان کرد که هرچه اطلاعات بیشتری را خواهیم انتقال دهیم هزینه این انتقال بیشتر شده و مدت زمان نیز به طبع آن افزایش می یابد و هرچه مقدار کمتری از اطلاعات را خواهیم انتقال دهیم هزینه آن انتقال کاهش یافته و مدت زمان آن نیز کاهش.

برای مثال - اگر خواهیم مقدار 600 مگابایت را از طریق کابل پارالل انتقال دهید در حدود 5 ساعت زمان نیاز داریم و این در حالی است که این انتقال از طریق یک خط Dial-up اینترنتی در حدود 41 ساعت زمان نیاز دارد.

حال - همین انتقال 600 مگابایتی از طریق یک خط LAN پیشرفته به حدود 10 دقیقه زمان نیاز دارد و اگر یک تایپیست بخواهد آنرا تایپ نماید اگر فرض کنیم هر 10.000 کلمه برابر با یک مگابایت است و 6 میلیارد کلمه برابر است با 600 مگابایت و در هر صفحه 1000 کلمه به طور میانگین داشته باشیم در حدود 6 میلیون صفحه باید تایپ شود و اگر یک کتاب به طور میانگین 250 صفحه داشته باشد - حدود 24 هزار کتاب را باید تایپ نماید. محاسبه ی زمان و هزینه با شما.

در خصوص انتقال 600 مگابایت اطلاعات از طریق فلاپی - باید به این نکته توجه داشت که از یک فلاپی به طور میانگین می توان 5 تا 10 بار استفاده کرد و شما برای انتقال اطلاعات 600 مگابایتی باید آن اطلاعات را به قطعات 1.44 مگابایتی تقسیم نمایید و از طریق فلاپی انتقال دهید و سپس اطلاعات را در دستگاه مبدا به یکدیگر پیوند دهید !

مورد بعدی زمان حصول اطمینان از صحت اطلاعات است. منظور از زمان حصول اطمینان از صحت اطلاعات مدت زمانی است که به طول می انجامد تا شما اطمینان حاصل نمایید که اطلاعاتی که از روی دستگاه اول به دستگاه دوم منتقل شده است با اطلاعات موجود بر روی دستگاه اول یکی هستند.

در مورد اول - در هنگام تایپ کردن - مطمئناً زمان زیادی لازم است تا اطلاعات تایپ شده بررسی شوند - که تقریباً زمانی در حدود زمان تایپ نیاز دارد تا اطلاعات بررسی شوند.

در خصوص مورد دوم - یعنی انتقال اطلاعات از طریق فلاپی - پس از هر انتقال بررسی حجم فایل انتقال یافته در نهایت کفایت می کند که زمان کمی می برد.

در خصوص مورد سوم ( انتقال از طریق CD ) نیز به مانند فلاپی تنها بررسی حجم فایل کفایت می کند.

یافته در هر مرحله می تواند ما را از نتیجه مطمئن سازد و این حجم بستگی مستقیم به حجم انتقال داده ی ما دارد. برای مثال برای یک فلاپی از 1 بایت تا 1.44 مگابایت و برای یک CD از 1 بایت تا 650 مگابایت و برای تایپ از یک حرف تا یک کلمه !

در خصوص مورد کابل پارالل نیز به همین صورت است. در هر ثانیه می توان 30 کیلوبایت را انتقال داد - پس حجم داده در هر ثانیه می تواند بین 1 بایت تا 30 کیلوبایت باشد. ( حد اقل را بایت در نظر گرفته ایم تا بلوک داده مان کامل باشد ) و همین نسبت در خصوص دیگر وسایل ارتباطی صدق می کند.

توجه کنید که در اینجا ما یک پروتکل را بررسی نمی کنیم. بلکه مواردی را بررسی می کنیم که در محاسبه ی سرعت انتقال اطلاعات ما را یاری می نمایند.

با توجه به موارد مطرح شده - در می یابیم که زمان حصول اطمینان از صحت انتقال اطلاعات نیز بخش مهمی است که باید به زمان کل انتقال اطلاعات اضافه شود. این بخش در فصل مربوط به Pre-trusted-connection به طور مفصل بررسی خواهد شد.

اینترنت جز یک شبکه ی غیر یکپارچه ی بزرگ شامل چندین شبکه ی کوچک و بزرگ نمی باشد. در واقع اینترنت یک شبکه ی جهانی است. شاید بتوان آنرا به International Network معنی کرد.

فرض کنید شما در منزلتان دارای دو دستگاه کامپیوتر هستید. به هر کامپیوتر شما یک شناسه می دهید مانند 1 و 2 و 3 و یا A و B و C و یا تلفیقی از اینها که من شناسه گذاری عددی را ترجیح می دهیم. در دفتر کار خود نیز 5 دستگاه کامپیوتر دارید که یکی از آنها به دستگاه چاپگر مجهز است. هم در منزل و هم در محل کار خود مجهز به خط تلفن هستید و یکی از کامپیوتر های موجود در دفتر کار شما مجهز به دستگاه مودم و یکی از کامپیوترهای منزل شما نیز مجهز به مودم هستند. در محل دفتر کار خود - کامپیوتر ها را به وسیله ی سیستم LAN به یکدیگر متصل نموده اید. این بدان معنی است که هرگاه بخواهید مطلبی را برای مثال چاپ نمایید - کافی است روی هر کدام از کامپیوترها که آن مطلب را دارید از طریق شبکه ی LAN به کامپیوتری که چاپگر بدان متصل است ارسال نمایید و سپس از آن طریق آنرا چاپ نمایید.

حال فرض را بر این بگذارید که مطلبی را در خانه تایپ نموده اید و می خواهید چاپ نمایید - کافی است از طریق خط تلفن به کامپیوتر دفتر خود متصل شوید و سپس فایل را به آن کامپیوتر موجود در دفتر خود از طریق شبکه ی LAN انتقال دهید و سپس دستور چاپ دهید.

فرض دیگر آن است که شما در دفتر کار خود نیاز به فایلی دارید که بر روی کامپیوتر دوم در منزلتان است که مودم نیز ندارد. کافی از یکی از کامپیوتر های دفتر خود را از طریق مودم و خط تلفن به آن کامپیوتر که در منزلتان دارای مودم است متصل نمایید و می دانید که آن کامپیوتر نیز از طریق کابل پارالل به کامپیوتر دوم متصل است که فایل مورد نظر شما بر روی آن است اما مودم ندارد. سپس با وارد نمودن شماره ی آن کامپیوتر از هر کدام از کامپیوترهای دفترتان می توانید به آن

دفتر کارتان انتقال دهید.

حال فرض کنید که شما دارای 5 دفتر بزرگ در تهران هستید. هر دفتر شما دارای حد اقل 10 کامپیوتر است و کارهای حسابداری هر دفتر بر روی یکی از 10 کامپیوتر موجود در هر دفتر انجام می شود. شما در هر دفتر هر 10 کامپیوتر را از طریق سیستم LAN به هم متصل نموده اید. یکی از کامپیوترهای هر دفتر را نیز از طریق خط تلفن به یکی از کامپیوترهای دیگر در دفتر دیگر متصل نموده اید. در اینصورت شما یک شبکه ی بزرگ دارید که شامل 50 کامپیوتر متصل به هم است.

در یک لحظه شما می توانید به هر کدام از کامپیوترها که مد نظرتان است - منجمله کامپیوترهای حسابداری دسترسی داشته باشید.

حال شما 5 دفتر در تهران دارید که هر دفتر 10 کامپیوتر دارد و یک دفتر در شیراز دارید که 10 کامپیوتر نیز آنجا دارید و یک دفتر نیز در مشهد که 20 کامپیوتر نیز در آنجا دارید. هم دفتر شیراز و هم دفتر مشهد مجهز به سیستم LAN هستند. شما 5 دفتر تهران خود را از طریق خط تلفن قبلا" به هم متصل نموده اید و یک شبکه ی 50 کامپیوتری تشکیل داده اید و حالا اگر یکی از این 50 کامپیوتر را از طریق خط تلفنی دیگر به یکی از کامپیوترهای دفتر شیراز متصل نمایید 10 کامپیوتر دفتر شیراز نیز به شبکه ی 50 کامپیوتری شما متصل می شود و اگر فقط یکی از کامپیوترهای دفاتر تهران یا یکی از کامپیوترهای دفتر شیراز را به یکی از 20 کامپیوتر دفتر مشهد متصل نمایید - شما 20 کامپیوتر دیگر به شبکه ی تهران و شیراز اضافه نموده اید و مجموعا" یک شبکه دارید به وسعت تقریبا" کشور ایران که شامل 80 کامپیوتر است. حال اگر فقط مدیران این دفاتر شبها از منزل خود بخواهند از طریق خط تلفنی دیگر به دفاتر خود متصل شوند - به ازای هر تماس 1 دستگاه کامپیوتر دیگر به شبکه ی شما اضافه خواهد شد.

اگر شما دفتر دیگری در لندن داشته باشید و آن دفتر 5 یا 10 یا حتی 1 یا 3 یا 100 کامپیوتر داشته باشد فقط با برقراری یک تماس تلفنی

شبکه ي خود اضافه نموده ايد.

در نظر داشته باشيد که به جاي ارتباط تلفني مي توان از ارتباطهايي مانند XDSL و يا ماهواره اي و يا حتي فيبر نوري نيز استفاده نمود.

اينترنت نيز همين است. در هر شهري يک ISP يا Internet Service Provider وجود دارد که وظيفه ي اتصال کامپيوتر و يا کامپيوتر هاي شما را به شبکه ي اينترنت دارد. در برخي از کشورها محيطهاي با نام IDC يا مرکز نگهداري اطلاعات اينترنتي وجود دارد. و يا شرکت و يا شخصي که از طريق ISP به اين شبکه متصل شده است و يا يک خط اينترنتي به صورت مستقيم وظيفه ي اتصال وي را به شبکه بر عهده دارد.

به محض اينکه شما از طريق ISP به اين شبکه متصل مي شود - دستگاه کامپيوتر شما به صورت خودکار يک شناسه ي مجازي يا حقيقي (Valid and Invalid IP address) را به خود اختصاص مي دهد. از اين به بعد شما نيز جزئي از شبکه ي اينترنت هستيد و شبکه به اندازه ي يک دستگاه کامپيوتر بزرگتر شده است.

لطفاً توجه فرماييد که بحثهاي مربوط به Topology شبکه و پيکربندي آن و Gateway و ... از حوصله ي اين کتاب خارج است.

از آنجا که در اينترنت شما مي توانيد از هر کامپيوتر به کامپيوتر ديگر راه يابيد و اين مسئله براي ديگران نيز صدق مي کند و معمولاً شرکتهاي تجاري و داشنگاهها بجاي اينکه شبکه ي دفاتر مختلف خود را از طريق خط تلفن به هم متصل نمايند ( به دليل هزينه ي زياد ) از طريق شبکه ي اينترنت به هم متصل مي کنند - و معمولاً اطلاعات مالي و تجاري و علمي خود را از طريق شبکه هاي خود جا به جا مي نمايند - از اينرو مسئله اي به نام امنيت شبکه و سطح دسترسي مطرح شده است.

LAN و یا پارالل به هم متصل شده اند پیاده سازی نمایید - انرا نیز می توانید در اینترنت پیاده سازی نمایید.

معنای فارسی پروتکل همان عهدنامه است. ما برای چه عهدنامه منعقد می کنیم؟ برای آنکه نوع روابط و ارتباطات را قانون مند نماییم. این قانون برای چه است؟

- جلوگیری از هرج و مرج
- سرعت بیشتر
- کنترل وضعیت
- افزایش کارایی

و ...

در ارتباطی که بین دو کامپیوتر برقرار شده است می بایستی عهدنامه ای منعقد گردد تا :

- سرعت ارتباط مشخص گردد
- صحت انتقال اطلاعات بررسی شود
- هزینه ی ارتباط کاهش یابد

و ...

پروتکل فقط مختص ارتباطات کامپیوتری نمی شود. ما انسانها نیز جهت برقراری ارتباط با یکدیگر از مفهوم عهدنامه ی ارتباطی استفاده می کنیم. برای مثال - برای برقراری ارتباط با شخصی معمولاً از واژه ی "سلام" استفاده می کنیم. پس ما بلافاصله با شنیدن واژه ی سلام می توانیم به شخص جواب دهیم "علیک سلام" و این یعنی یک عهدنامه ی اجتماعی - انسانی جهت برقراری تماس.

در سیستمهای کامپیوتری نیز وضع به همین صورت است. اگر دو کامپیوتر بخواهند با یکدیگر تماس برقرار نمایند کافی است یکی از دو کامپیوتر که قصد دارد با دیگری تماس برقرار نماید متن یا علامت یا حتی بیتی را به عنوان "سلام" ارسال نماید. برای مثال کامپیوتر الف



آنکه به کامپیوتر الف اطمینان بخشد که پیام او را دریافت کرده است و ارتباط برقرار شده است وازه ي WELCOME را براي او مي فرستد. بدین ترتیب کامپیوتر الف مطمئن می شود که ارتباطش با کامپیوتر ب برقرار شده است. برای پایان تماس نیز کافی است کامپیوتر الف وازه ي GOOD BYE را ارسال نماید و کامپیوتر ب متوجه می شود که ارتباط از جانب کامپیوتر الف قطع شده است. حال بنا به شرایط می تواند پیامی مبنی بر تأیید قطع ارتباط ارسال نماید و یا صرفاً "ارتباط قطع شده است".

آنچه توضیح داده شد - اساس ساده ي یک پروتکل ارتباطی است. انواع مختلفی از پروتکلهاي ارتباطی وجود دارند. برای مثال - پروتکلهاي امن - سریع - قابل اعتماد و غیر قابل اعتماد وجود دارند. برای آنکه با این مفاهیم بیشتر آشنا شویم ابتدا باید اصول جا به جایی فایل را از طریق یک پروتکل بررسی نماییم.

فرض نمایید که می خواهید یک فایل TXT با محتوی WELCOME TO MHDSOFT را از طریق یک پروتکل انتقال دهید.

راهبرد اول : انتقال از طریق یک پروتکل سریع:

وظایفی که بر عهده ي پروتکل انتقال سریع می باشد شامل :

- انتقال سریع اطلاعات از طریق یک بستر ارتباطی مانند کابل پارالل یا خط تلفن
- حصول اطمینان از انتقال اطلاعات

بر اساس وظایفی که بر عهده ي یک پروتکل انتقال سریع می باشد - می توان یک الگوریتم ساده به صورت ذیل آماده نمود - اما قبل از آن

است :

- وضعیت ارسال
- وضعیت دریافت

و برای یک پروتکل می تواند شرایط ذیل به وجود آید :

- در حال ارسال
- در حال دریافت
- سکوت
- خطا
- ارسال مجدد
- دریافت مجدد
- قطع ارتباط
- وصل ارتباط
- انتظار

در پروتکل انتقال سریع ما باید هر دو وضعیت عمومی پروتکلها را بسازیم ( یعنی ارسال و دریافت ) و از آنجا که باید وظایف خاص آنها در نظر بگیریم بنا بر این پروتکل ما می تواند در شرایط ارسال و دریافت و قطع ارتباط و وصل ارتباط و انتظار قرار گیرد.

شما می کلی پروتکل ارسال سریع بدین صورت است که کامپیوتر الف باید پورت ارتباطی را باز نماید ( برای مثال - کابل پارالل ) سپس خود را به عنوان فرستنده معرفی نماید. در کامپیوتر ب سیستم در حالت انتظار است و به محض دریافت کد برقراری تماس خود را در وضعیت دریافت پیام قرار می دهد - چون کامپیوتری که ارتباط را برقرار نموده است فرستنده می باشد.

سپس کامپیوتر الف - می گوید که می خواهد فایل با نام xxxxxx.txt را ارسال نماید. کامپیوتر ب فقط باید تایید کند که این نام را

الف ارسال نماید ( چون پروتکل سریع است نه قابل اعتماد ).

سپس کامپیوتر الف بر اساس حجم تعریف شده بر اساس پهناي باند ارتباطي آن مقدار اطلاعات را ارسال مي نماید. براي مثال - حجم ارتباطي پهناي باند Dial-up در حدود 4 کیلوبایت معرفي مي شود و براي پارالل 30 کیلوبایت در ثانیه. پس کامپیوتر الف باید آزمایش کند که آیا حجم داده ي xxxxxx.txt بیشتر از 4 یا 30 کیلوبایت است و یا کمتر.

براي مثال - حجم تعریف شده 1 بایت در ثانیه است. کمترین سرعت ممکن در ارتباطات کامپیوتری. جمله ي مورد نظر ما 18 بایت است ( WELCOME TO MHDSOFT ) بنا بر این حجم داده ي ما بیشتر از ظرفیت انتقال اطلاعات است - پس برنامه ي موجود در کامپیوتر الف باید یک کل اطلاعات را به قطعات یک بایتی تقسیم نماید. در اینصورت اطلاعات موجود در فایل به 18 قطعه ي یک بایتی تقسیم مي شود.

سپس برنامه ي کامپیوتر الف باید اعلام نماید که 18 بایت اطلاعات را مي خواهد انتقال دهد و پس از دریافت پیام تایید از کامپیوتر ب باید اطلاعات را به صورت بایت به بایت ارسال نماید و پس از اتمام ارسال اطلاعات باید اعلام نماید که اطلاعات را تماما" ارسال کرده است.

کامپیوتر ب نیز باید مجموع اطلاعات را حساب کند و اگر برابر با 18 بایت باشد باید پیغام تایید دهد و اگر کمتر یا بیشتر باید پیغام خطا دهد. در صورت دریافت پیغام خطا توسط کامپیوتر الف - انتقال مجددا" از ابتدا شروع مي شود در غیر اینصورت ارتباط پایان مي یابد.

اشکالات این نوع پروتکل عبارتند از :

- در صورتی که به خاطر وجود پارازیت یا اختلال در خط حرف A به صورت B دریافت شده باشد نمی توان آنرا تشخیص داد
- در صورتیکه در میان انتقال اطلاعات - ارتباط قطع شود - در صورت تمایل به ارسال مجدد باید اطلاعات از ابتدا دریافت شوند

- شوند
- این نوع ارتباط تنها جهت ارسال اطلاعات ASCII استعمال می شوند
- امکان شنود اطلاعات وجود دارد
- تنها یک ارتباط پشتیبانی می شود

مزایای این روش عبارتند از :

- سرعت بسیار بالا در انتقال اطلاعات
- مناسب برای انتقال اطلاعات متنی
- مناسب برای ارتباط میان دو دستگاه کامپیوتر نزدیک به هم می باشد

و اما شبه الگوریتم این روش :  
در کامپیوتر الف :

1. نام فایل را بگیر
2. فایل را باز کن - حجمش را بگیر
3. میزان پهنای باند را بگیر
4. آیا میزان پهنای باند بیشتر از حجم فایل است ؟ اگر بله به گام 13 برو
5. فایل را به قطعات متناسب با پهنای باند تقسیم کن
6. ارتباط را برقرار کن
7. آیا ارتباط برقرار است ؟ اگر خیر به گام 6 برو
8. نام فایل را ارسال کن
9. آیا تایید شد توسط کامپیوتر ب ؟ در صورت خیر به گام 8 برو
10. حجم فایل را ارسال کن
11. آیا تایید شد توسط کامپیوتر ب ؟ در صورت خیر به گام 10 برو
12. گام 13 را به تعداد بدست آمده در گام 5 اجرا کن
13. به اندازه ی پهنای باند - بایت از فایل ر ارسال کن
14. پایان ارسال را اعلام کن

- اگر بله به کام 16 برو و اگر خیر به کام 14 برو
16. آیا حجم ارسال شده توسط کامپیوتر ب با حجم فایل ارسال یکی است ؟ اگر خیر به گام 5 برو
17. پایان ارتباط - ارتباط را قطع کن

شبه الگوریتم موجود در کامپیوتر ب :

1. پورت را باز کن
2. آیا پیامی در پورت دریافت شده است ؟ اگر خیر به 1 برو
3. برقراری ارتباط را اعلام کن
4. آیا نام فایل را دریافت کرده ای ؟ اگر خیر به گام 3 برو
5. نام فایل را دریافت کن
6. آیا نام فایل را دریافت کرده ای ؟ اگر خیر به گام 5 برو
7. دریافت نام فایل را تایید کن
8. حجم فایل را دریافت کن
9. آیا حجم فایل دریافت شده است ؟ اگر خیر به گام 8 برو
10. دریافت حجم فایل را تایید کن
11. آیا اطلاعاتی ارسال شده است ؟ اگر بله آنرا در متغیر A بریز
12. آیا پایان انتقال اعلام شده است ؟ اگر خیر به گام 11 برو
13. میزان حجم اطلاعات دریافتی در متغیر A را بشمار و به کامپیوتر الف ارسال کن
14. آیا پایان ارتباط اعلام شده است ؟ اگر خیر به گام 1 برو
15. محتویات را در فایلی که نام آن قبلاً اعلام شده بریز
16. به گام 1 برو

همانطور که مشاهده می کنید اساس کار یک پروتکل سریع خیلی ساده است، و می توان این پروتکل را چرخ ابتدایی پروتکل های پیشرفته دانست. در واقع یک سکوی ساده برای دیگر پروتکلها.

برای مثال - برای اینکه بتوانید پروتکل سریع را به پروتکل قابل اعتماد تبدیل کنید فقط کافی است هر بار هر بایتهی که دریافت می کنید آنرا عیناً برای کامپیوتر ارسال کننده ارسال نمایید تا کامپیوتر ارسال کننده

شما رسیده است.

اما یک رابطه ی جالب بین سرعت و پروتکل وجود دارد. هرچه پروتکل قابل اعتماد تر باشد سرعت پایین تر می رود و برای افزایش سرعت با پروتکل قابل اعتماد نیاز به صرف هزینه ی بیشتر است. برای مثال استفاده از فن آوری پیشرفته.

برای تبدیل یک پروتکل سریع به یک پروتکل امن می توان اطلاعات را هنگام ارسال رمز کرد. برای مثال بجای ارسال حروف و کلمات ASCII آنها را به کد ASCII تبدیل کرد و ارسال نمود. برای مثال کد 13 برای مشخص کردن Enter یا EOL. و ایده ی بهتر آن است که این عدد را در عددی هنگام ارسال ضرب نمایید و هنگام دریافت تقسیم نمایید. بدین صورت دیگر اطلاعات شما قابل شنود نمی باشد. و اگر شیوه ی امن را نیز بدان اضافه نمایید واقعا "پروتکل امنی می شود.

همانطور که در چند پاراگراف گذشته نیز اشاره شد - اصول کار پروتکل سریع در واقع پایه و اساس پروتکل های دیگر است.

یک پروتکل را شما به گونه ای می توانید سر و سامان دهید که بتواند چندین ارتباط را به صورت همزمان پشتیبانی نماید. این بدان معنی است که اگر بجای یک کامپیوتر الف - ما دو کامپیوتر الف داشته باشیم این پروتکل می تواند به گونه ای برنامه ریزی شود که از هر دو کامپیوتر فایل را دریافت نماید. حتی اگر دو کامپیوتر الف به یکصد و یا هزار کامپیوتر الف تبدیل شود.

ایده ی این طرح بسیار ساده ولی کار آمد است و اساس کار اکثر پروتکل های اینترنتی نیز بدین صورت است. برای مثال از یک وب سایت ممکن است بیش از یک نفر در حال بازدید باشد و این را در نظر بگیرید که وقتی شما در حال مشاهده ی یک وب سایت هستید در واقع در حال دانلود یک فایل از آن وب سایت هستید و این مرور گر اینترنت شما است ( منظور از مرور گر - همان نرم افزاری است که به شما این امکان را می دهد که صفحات وب را مشاهده نمایید مانند نرم افزار

صفحه را به شما نشان می دهد.

البته همانطور که متوجه شدید - در وب پروتکل ارسال اطلاعات مانند پروتکل کامپیوتر الف است که اصطلاحاً "میزبان نام دارد و کامپیوترهایی که فایل را دریافت می کنند مانند کامپیوتر ب هستند که اصطلاحاً به آنها لفظ مشتري اطلاق می شود.

برای مثال - اگر قرار باشد پروتکل ب بتواند تعداد نا محدودی فایل را از تعداد نا محدودی کامپیوتر مانند الف دریافت نماید کافی است الگوریتم آن به شکل ذیل تغییر یابد:

ابتدا کامپیوتر الف به کامپیوتر ب یک پیام ارسال می کند مبنی بر برقراری اتصال و از آن می خواهد که یک کد شناسایی در اختیار وی قرار دهد. کامپیوتر ب دارای یک متغیر به نام ID است - به ID یک واحد اضافه می کند و آنرا برای کامپیوتر الف می فرستد. پس اگر مقدار ID برابر با 1 باشد - از این پس نام کامپیوتر الف ، 1 است و نه الف. بلافاصله در کامپیوتر ب یک خانه آرایه به نام TEMP(1) که شامل تعدادی بلوک است ایجاد می شود و این خانه آرایه مخصوص اطلاعاتی می شود که کامپیوتر 1 می خواهد ارسال کند.

همزمان با کامپیوتر الف - که الان نام مجازی آن 1 است - کامپیوتر ج اقدام به برقراری ارتباط می نماید. در این زمان کامپیوتر ب مجدداً یک واحد به ID می افزاید و مقدار ID از 1 به 2 افزایش پیدا می کند. این کد را برای کامپیوتر ج ارسال می نماید و کامپیوتر ج از الان می داند که نامش 2 است و نه ج. مانند قبل یک خانه آرایه با نام TEMP اما با اندیس 2 به نام TEMP(2) ساخته می شود که شامل تعدادی بلوک و مخصوص اطلاعاتی است که کامپیوتر 2 می خواهد ارسال نماید.

در این هنگام کامپیوتر 1 اطلاعات نام و حجم فایل را در قالبی مانند ذیل ارسال می نماید :

1-18-test.txt

1 یا همان الف ارسال شده است. عدد 18 به این معنی است که حجم اطلاعاتی که می‌خواهم بفرستم برابر با 18 بایت است و در آخر هم نام فایل. این اطلاعات در آرایه ی (1)TEMP درج می‌شوند و سپس وقتی که کامپیوتر الف یا همان 1 بخواهد اطلاعات بایت بایت را ارسال نماید کافی است در ابتدای اطلاعات شماره ی خود را درج نماید. برای مثال متن فایل ما که WELCOME TO MHDSOFT بود به صورت ذیل ارسال خواهد شد :

1)W

1)E

1)L

1)C

1)O

1)M

1)E

1)

1)T

1)O

1)

...

در این هنگام کامپیوتر 2 نیز می‌تواند به صورت همزمان اطلاعات خود را ارسال نماید. تنها تفاوتی که دارد این است ه در ابتدای اطلاعات بجای عدد 1 عدد 2 قرار می‌گیرد. حتی اگر آنچه ارسال می‌شود یکی باشد. برای مثال اگر کامپیوتر 2 قرار باشد که متن MHDSOFT را ارسال نماید کافی است با شناسه ی 2 آنهارا ارسال نماید و اگر این ارسالها همزمان شوند به صورت ذیل در خواهند آمد :

1)W

2)M

1)E

2)H



- 2)D
- 1)C
- 2)S
- 1)O
- 2)O
- 1)M
- 2)F
- 1)E
- 2)T
- 1)
- 2)
- 1)T
- 2)W
- 1)O
- 2)E
- 1)
- 2)B
- 1)M
- 2)
- 1)H
- 2)S
- 1)D
- 2)E
- 1)S
- 2)R
- 1)O
- 2)V
- 1)F
- 2)E
- 1)T
- 2)R

البته آنچه در اینجا بررسی نشد ظرفیت پهنای باند است. این بدان معنی است که از یک سیم در آن واحد چه مقدار اطلاعات می تواند عبور کند و آیا واقعا" دو کامپیوتر به صورت همزمان می توانند اطلاعات خود را ارسال نمایند؟

جواب این سوال بله است - دو کامپیوتر می توانند به صورت همزمان اطلاعات ارسال نمایند اما مشکلی به نام برخورد اطلاعات به وجود می آید و احتمالا" اطلاعات نا معلومی به میزبان خواهد رسید. از اینرو در شبکه بندی کامپیوتری از نرم افزار مدیریتی یا پروتکل مرجع استفاده می شود که وظیفه ی آن نوبت دهی به کامپیوتر ها است.

ایده ی اولیه این سیستم به این شکل بوده است که در یک شبکه که تعداد زیادی کامپیوتر به یکدیگر متصل هستند پروتکل مرجع و یا خود پروتکل ب بدانند که چه تعداد کامپیوتر در شبکه فعال هستند. به ترتیب برای هر یک یک پیغام ارسال کند که آیا پیامی برای من داری ؟ اگر جواب آری باشد پیام به پروتکل ب می رسد و اگر خیر پروتکل ب از دستگاه بعدی سوال می کند.

این کار باعث می شود :

- سرعت انتقال اطلاعات کاهش یابد
- از برخورد اطلاعات جلوگیری شود
- کارایی افزایش یابد
- تعداد زیادی کامپیوتر بتوانند اطلاعات خود را به میزبان ارسال کنند

اگر زمان ارسال سوال و دریافت پاسخ در یک شرایط برابر با یک ثانیه باشد و تعداد شصت کامپیوتر در یک شبکه حضور داشته باشند و پهنای باند 1 بایت باشد در اینصورت فقط انتقال یک بایت اطلاعات از هر یک از کامپیوترها برابر خواهد بود با 1 دقیقه. و همان 18 بایتی ما

وجود نیاید !

راه حل‌های متعددی برای افزایش سرعت و در نتیجه کاهش زمان در چنین شبکه ای وجود دارد - یک راه افزایش پهنای باند است. یعنی برای مثال استفاده از پارالل یا LAN و ...

اما بازهم در شرایطی با افزایش تعداد کامپیوترها در شبکه - سرعت شبکه کاهش خواهد یافت. راه حل دیگر بهینه سازی پروتکل است و تقسیم بندی هوشمند. یعنی سیستم پروتکل شما سوال خود را به کامپیوترهایی بفرستند که بیشترین درخواست را اخیراً داشته اند و با الگوریتم خاصی سوال خود را به کامپیوترهای خاموش بفرستند.

راه حل دیگر استفاده از سیستم چند کاناله است. در این سیستم دو زوج سیستم از هر کامپیوتر به میزبان ب متصل شده اند. یک زوج پهنایی باندهی در حدود 1 بایت برقرار می کنند ( برای مثال ) و زوج دیگر 30 کیلوبایت در ثانیه. سیستم میزبان ب ( پروتکل ب ) سوالهای خود را از طریق زوج با پهنای باند کم می فرستد ( کانال 1 ) و اگر سیستمی درخواست ارسال و یا دریافت فایل داشته باشد در اینصورت یک نوبت از میزبان ب می گیرد و به محض آنکه نوبت وی شد - اطلاعاتش را از طریق زوج با پهنای باند زیاد ارسال می کند.

اما به هر حال چه این ایده و چه ایده های دیگر بازهم با افزایش تعداد کامپیوترها در شبکه با مشکل کندهی سرعت مواجه خواهند شد.

در این صورت شما می توانید از سیستم پروتکل ایستگاهی استفاده نمایید.

همانطور که از نام این پروتکل بر می آید - شما در این روش دارای ایستگاههایی هستید - به این مضمون که هر برای مثال 5 کامپیوتر شما دارای یک میزبان هستند - برای مثال کامپیوتر 1 تا 4 همه به میزبان 5 متصل هستند و به این مجموعه یک ایستگاه می گوئیم. حال شما در شرکت خود دارای 4 ایستگاه هستید و هر ایستگاه شامل 5 کامپیوتر است. در این صورت 20 کامپیوتر در شرکت خود

نمایید در واقع یک شبکه با 20 کامپیوتر خواهید داشت که بین اولین کامپیوتر و آخرین کامپیوتر 20 ثانیه اختلاف زمانی است و انتقال برای مثال 18 بایت در پهنای باند 1 بایت برابر خواهد بود  $20 \times 18 = 360$  ثانیه برابر با 6 دقیقه! اصلاً عدد جالبی نمی باشد.

حال در نظر بگیرید که هر 5 کامپیوتر شما به یک ایستگاه تبدیل شده است. در این صورت هر 5 کامپیوتر شما که یک ایستگاه است در واقع یک کامپیوتر محسوب می شود. پس در شبکه ی اصلی خود شما تنها 4 کامپیوتر دارید و اختلاف زمانی اولین آنها تا آخرین آنها در حدود 4 ثانیه است و انتقال 18 بایت برابر خواهد بود با  $4 \times 18$  برابر با 72 ثانیه تقریباً برابر با 1.2 دقیقه. رقم قابل توجهی است.

در سیستم ایستگاهی - یک کامپیوتر به عنوان فرمانده ی ایستگاه عمل می کند و کامپیوترهای دیگر به عنوان فرمان بردار. کامپیوتر 1 در ایستگاه 2 درخواست فایل A را به فرمانده می فرستند - همزمان نیز کامپیوتر 3 در همین ایستگاه درخواست فایل B را به فرمانده می فرستند. فرمانده مجموع این سفارشات را دریافت نموده و منتظر پیام بر روی شبکه ی بین ایستگاهی می شود و مجموع درخواستهای خود را برای میزبان اصلی ارسال می کند.

معمولاً در چنین معماریهایی پهنای باند بین بین کامپیوترهای موجود در یک ایستگاه با پهنای باند کامپیوترهای بین ایستگاهی متفاوت است. به هر حال مجموعه ای از این نوع راه حلها و معماریها هستند که باعث می شوند سرعت یک شبکه افزایش یابد.

شرکتیهایی مانند Microsoft و دیگران راه حلهایی را در محیطهای شبکه ی گسترده مانند اینترنت ارائه کرده اند که باعث شده است سرعت عمومی این نوع از شبکه ها افزایش یابد. برای مثال - فشرده سازی اطلاعات هنگام ارسال آنها. برای مثال - ارسال یک متن 18 بایتی در قالب یک دسته کد مجموعاً 6 بایتی می تواند سرعت ارسال را 3 برابر کند!

است که پایه و اساس تمامی پروتکلها همان پروتکل سریع می باشد  
که الگوریتم آن در این فصل بررسی شد.

در سالهای ما بین 1985 تا 1995 کشمکش شرکتها ی مختلف جهت ارایه ی پروتکلهاي مناسب بسیار زیاد بود. شرکتي مانند Novell یک نوع پروتکل را معرفی مي کرد و در کنار آن Microsoft نوعي دیگر پروتکل و شرکتهایي دیگر نیز انواع مختلفی از پروتکل را معرفی مي کردند.

در این راستا یک شرکت اگر تجهیزات خود را به شبکه ی شرکت X مجهز نموده باشد و سپس بخواهد شبکه ی خود را به شبکه ی شرکتي دیگر که از تجهیزات شبکه ی Y استفاده مي کند متصل نماید – شاید این کار مستلزم صرف هزینه ی زیادی باشد چون عاقلانه ترین کار :

- ایجاد یک سیستم مترجم است

سیستم مترجم سیستمي است که اطلاعات یک پروتکل را به اطلاعات قابل درک برای پروتکل دیگر ترجمه مي کند. ذکر این نکته نیز بر اهمیت است که منظور از تجهیزات شبکه ، مجموعه ی سخت افزار و نرم افزار شبکه است.

اصول کار سیستم مترجم به این صورت است، که ابتدا برنامه نویسی یا برنامه نویسان بررسی مي کنند که در شبکه ی X شروع ارتباط به چه صورت است ؟ سپس در شبکه ی Y به چه صورت است. در این صورت در صورتیکه گاهمهای بر قراري ارتباط در دو شبکه یکی باشد – فقط کافی است که مترجم مثلاً " کد HELLO را در شبکه ی X به کد EHLO در شبکه ی Y تبدیل کند و آنرا ارسال نماید.

اما متأسفانه کلیه امور تنها به اینجا ختم نمی شود. همانطور که مي دانیم ممکن است اصول همه ی پروتکلها یکی باشد – اما مطمئناً به دلیل تلاشی که شرکتها در خصوص بهینه سازی ارتباط انجام مي دهند نوع گامهای پروتکلها نیز متفاوت شده اند.

استفاده می شود. یعنی به این صورت که میزبان یا SERVER مرکزی به ترتیب از هر کامپیوتر سوال می کند که آیا مطلبي براي ارسال داري؟ و در شبکه ي Y سیستم کاملاً" به صورتی متفاوت عمل می نماید - یعنی هر سیستم ابتدا باید از کانالی دیگر دریافت نوبت کند و سپس اجازه ي انتقال اطلاعات دارد.

در اینصورت برنامه نویس پروتکل مترجم مجبور است بخشی از عملیات مربوط به یکی از شبکه ها که معمولاً" در شرایط ذکر شده منظور عملیات شبکه ي Y است را شبیه سازی نماید.

در اینصورت با مشکلات ذیل می توانیم مواجه شویم :

- کاهش سرعت
- افزایش خطا
- اختلال در هر دو شبکه

و ...

تا قبل از سال 2000 بسیاری از شبکه ها با استفاده از پروتکل مترجم با یکدیگر در ارتباط بوده اند و متأسفانه ما با مشکل عدم یکپارچگی در دنیای شبکه های کامپیوتری مواجه بودیم. اما به هر حال مشکل عمده بر سر پیدا کردن استانداردی برای شبکه بوده است که اگر شرکتی در این راستا می توانست موفق شود - مطمئناً" سود هنگفتی نصیب خود می کرده است.

در این راستا پروتکلهاي خوب و بد زیادی معرفی شدند. در نتیجه شرکتهاي مختلف به این نتیجه رسیدند که ایجاد یک پروتکل کلی و کاملاً" استاندارد که بتواند سالها بدون جایگزین معرفی شود بسیار کار سخت و مشکلی است و عملاً" نشدنی.

بدین ترتیب - تصمیم بر آن شد که یک پروتکل مرجع انتخاب شود که بر اساس آن دیگر پروتکلها بتوانند با یکدیگر در ارتباط باشند. برای مثال

می توانید پروتکل دلخواه خود را طراحی کنید و سپس آنرا به TCP/IP ترجمه نمایید و در واقع TCP/IP اطلاعات را به زبان خود به کامپیوتر دیگر انتقال می دهد و در کامپیوتر دیگر باز به پروتکل مورد نظر شما.

TCP/IP استاندارد است که وظیفه انتقال اطلاعات شما را بر عهده دارد. این استاندارد امکان برقراری تماس کامپیوتر شما را با دیگر کامپیوترهای موجود در شبکه فراهم می نماید. همچنین این استاندارد نظم می بخشد ارتباط شما را با دیگر کامپیوترهای موجود در شبکه. این پروتکل استاندارد خاص برای آدرس دهی کامپیوترهای سرتاسر شبکه فراهم می سازد با نام آدرس IP و یا IP ADDRESS.

بنا بر این در حال حاضر شما این امکان را دارید که بتوانید پروتکل دلخواه خود را تحت شبکه اینترنت و یا LAN برنامه نویسی نمایید و سپس از طریق یک رابط به نام پریز ( Socket ) آنرا به هر دستگاه کامپیوتر دیگری در شبکه ارسال کنید.

در اینجا شما با مفهوم استاندارد در دنیای پروتکل آشنا شدید. در واقع استاندارد در علوم کامپیوتر یعنی انتخاب یک زبان و یا یک مفهوم مرجع به صورتی که همگانی باشد و اکثر برنامه نویسان بتوانند با آن برنامه های خود را بسازند که عمدتاً این استانداردها - استانداردهای ارتباطی می باشند.

در برخی مواقع یک استاندارد توسط یک سازمان ارایه می شود و در برخی مواقع تعداد استفاده کنندگان از یک مفهوم و یا ابزار آنقدر زیاد می شود که عملاً به صورت خود به خود به یک استاندارد تبدیل می شود.

برای کسب اطلاعات بیشتر پیشنهاد می کند کتاب شبکه های کامپیوتری - نویسنده : آقای تنباوم را مطالعه فرمایید.



شاید یکی از پر طرفدار ترین جنبه های اینترنت وب در آن باشد. WEB . بخشی از اینترنت است که اکثر ما کاملاً با آن آشنا هستیم . سایتهایی مانند Google و Yahoo و Microsoft و MHDsoft .

در یک کلام وب چیزی نیست جز دانلود کردن یک فایل خاص از یک میزبان و مشاهده ی محتویات آن توسط یک مرور گر یا نشان دهنده.

اما آن چیز که در وب آن را هیجان انگیز کرده است - مختصات خاص آن است. یعنی شما در یک صفحه می توانید همزمان - تصویر - صدا - متن و ارتباط با صفحات دیگر داشته باشید ( Link ). ساخت صفحات وب بسیار ساده است و معمولاً نیاز به آموزش زیادی ندارد و نرم افزارهای زیادی مانند Microsoft frontpage به شما در طراحی وب سایتان کمک می کنند.

فایلهای ساخته شده معمولاً با پسوندهایی مانند HTML و HTM و ASP و PHP و ... در میزبان قرار می گیرند. نقطه ی قوت آنها در این است که متن این فایلها همه TEXT هستند مانند متن ذیل :

```
<Title> MHDsoft Introduction page </title>
<Center> Welcome to MHDsoft </Center>
<p> MHDsoft international is a software company. </p>
```

همانطور که مشاهده می کنید متن یک فایل HTML تماماً TEXT است و شما می توانید با هر نوع Editor یا نرم افزار ویرایشگر متن آنرا بازبینی نمایید و حتی تغییراتی دهید. ( جهت آموزش طراحی وب - پیشنهاد می کنم به کتب منتشره در این زمینه مراجعه فرمایید - مانند مرجع طراحی وب - انتشارات نص و ... )

همانطور که در نمونه ی کد بالا مشاهده می نمایید - در ابتدا و انتهای متون بخشهایی درون <> مشخص شده اند که اصطلاحاً به آنها Tag می گویند. وقتی شما یک فایل HTML را دانلو می نمایید ( با

Internet Explorer ( نرم افزار مرور کر شما شروع به چیدمان متن شما بر حسب دستورات درون TAG می نماید.

برای مثال در متن بالا – تگ با عنوان TITLE یعنی جمله ی بعد از این دستور عنوان این صفحه است و تگی که با / شروع شده است مشخص کننده ی پایان دستور است. یعنی آنچه که بین <TITLE> و </TITLE> قرار گرفته است – عنوان صفحه است.

تگ بعدی <CENTER> و </CENTER> است. این بدان معنی است که هرچه بین این دو تگ قرار گرفته است باید در میان صفحه نشان داده شود. بنای بر این متن Welcome to MHDsoft در میان صفحه نشان داده خواهد شد.

تگ <p> و </p> مشخص کننده ی یک پاراگراف است و به نرم افزار می گوید متنی که بین این دو تگ است باید از سمت چپ صفحه شروع شود چون یک پاراگراف را مشخص می نماید.

به همین ترتیب تگها ( دستورات ) دیگری وجود دارند که با استفاده از آن شما می توانید صفحه ی خود را بخش بندی نمایید و یا تصویر و حتی یک صدا در صفحه قرار دهید. آنچه توضیح داده شد – اصول کار وب است . همانطور که متوجه شده اید به دلیل ماهیت وب و اینکه شما می توانید متن خود را نشانه گذاری نمایید – وب قابلیت پیشرفت بسیار زیادی دارد.

یکی از قابلیت های مهم وب آن است که شما می توانید با استفاده از همین تگها و دستورات صفحه ای را بسازید که بازدید کننده ی وب سایت شما را قادر می سازد که اطلاعاتی را برای شما ارسال نماید. در این صورت اطلاعات ارسال توسط بازدید کننده ی وب سایت شما نیاز به پردازش و ذخیره سازی دارد.

PreI در میزبان وب خود فرار دهید که وظیفه ی انها پردازش متون ارسالی است.

اطلاعات از کامپیوتر اشخاص به روشهای متداولی می تواند به وب سایت شما انتقال یابد. دو روش متداول شامل GET و POST است. روش GET روشی ساده و کارآ است. در روش GET شما می توانید اطلاعات مختصری را به وب سایت خود ارسال نمایید. برای مثال در نهایت در حدود 255 کاراکتر و در برخی سیستمها تا 600 کاراکتر اطلاعات.

اطلاعات GET به راحتی توسط کاربر قابل مشاهده هستند و بنا بر این در سیستمهای امنیتی از POST و مشابه آن استفاده می شود.

در سیستمهایی که کاربر می خواهد متن ارسال نماید و یا فایلی آپلود نماید معمولاً از سیستم POST استفاده می شود.

کاربرد دیگر استفاده از سیستم GET در موتورهای جستجو است. در این نوع از پایگاههای اطلاع رسانی کاربر می تواند با ارسال یک کلمه یا مجموعه ای از کلمات به میزبان آنرا در پایگاه داده جستجو نماید. برای مثال اگر شما فایلی با نام Search.Asp در وب سایت خود داشته باشید - کافی است با استفاده از متد GET اطلاعات خود را به صورت [Http://Website.com/search.asp?item=test](http://Website.com/search.asp?item=test) ارسال نمایید. این بدان معنی است که شما می خواهید وجود کلمه ی TEST را در پایگاه داده جستجو کنید. با استفاده از متد GET شما متغیر ITEM را مقدار گذاری نموده اید.

فایلهایی مانند ASP دارای تفاوتهای اساسی با فایلهای متنی HTM می باشند علی رغم اینکه محتوی هر دوی آنها متن است. در فایلهای ASP شما هم می توانید از تگهای HTM استفاده نمایید و هم می توانید اقدام به برنامه نویسی به زبان ASP نمایید. برای مثال - کدهای ذیل می توانند درون فایل Search.Asp باشند :

```
<title>Search result</title>
```

```
item=request.querystring("item")
```

```
rem search item
```

```
response.write "<p>The answer is: " & item & "</p>"
response.write "<font color=red> <Center>Thanks</Center>"
%>
<p>Search is done </p>
```

هنگامیکه برنامه میزبان وب می خواهد فایل ASP و مشابه آنرا به مشتری ارسال نماید ابتدا کدهای برنامه ی آنرا که مجموعه دستورات پردازشی هستند و با تگهایی مانند %< و %> شروع و پایان می یابند اجرا می نمایند. برای مثال در تکه کد بالا دستور Response.write به برنامه ی میزبان وب ( WEB SERVER ) دستور می دهد که این متن را به عنوان محتوی فایل برای کاربر ارسال کن. وقتی فایل های ASP و مشابه آن فراخوانده می شوند کدهای دستورات آنها به هیچ عنوان برای کاربر ارسال نمی شود و صرفاً نتیجه ی دستور العملها ارسال می شود.

این نوع از فایلها به صاحبان وب سایتها این امکان را می دهد که در مدت زمان کم و به سادگی نیازهای پردازشی خود را بر طرف نمایند. شما به راحتی می توانید بر روی وب سایت خود امکانات جستجو در وب سایت خودتان قرار دهید و یا کاربران شما بتوانند اقدام به عضویت در سایت شما نمایند و صدها کاربرد دیگر.

اما در اینجا شما باید مواظب باشید که زیاد به این نوع از قابلیتها اعتماد نکنید - مخصوصاً برای پروژه های بسیار بزرگ به دلایل امنیتی و ...

DNS یا Domain Name Server شاید یکی از جنجالی ترین و در عین حال ساده ترین ایده در وب باشد که توانسته است در سالهای اخیر وب را به یک قدرت تجاری تبدیل نماید.

شما وقتی کامپیوتر خود را به اینترنت متصل می نمایید بر اساس ساختمان پروتکل TCP/IP به کامپیوتر شما یک آدرس IP اطلاق می گردد - چیزی مانند 127.0.0.1 که هر بخش آن می تواند بین 0 تا 255 عدد گذاری شود. حال در صورتیکه این آدرس Valid باشد - هر شخصی از هر کجای دنیا که به شبکه ی اینترنت متصل باشد می تواند با وارد کردن این آدرس در پروتکل TCP/IP و یا برنامه های جانبی آن ( در بخش برنامه نویسی پرز به طور کامل بررسی می شود ) به کامپیوتر شما دسترسی داشته باشد.

برای مثال اگر آدرس IP کامپیوتر شما 10.9.1.0 باشد - و شما در تهران باشید - شخص دیگری که آدرس کامپیوتر آن 69.210.213.127 است و در ایالت پنسیلوانیا آمریکا است و هر دوی شما سیستم عامل Microsoft Windows 2000 برای مثال دارید - می تواند به راحتی منوی CMD را در کامپیوتر خود باز کند و تایپ نماید :

```
Net send 10.9.1.0 Hello
```

و چند لحظه بعد شما پیغامی بر روی صفحه ی مانیتور خود دریافت نمایید که نوشته شده است Hello !

بنا بر این اگر شما بخواهید فایلی را از روی یک میزبان نیز دانلود نمایید کافی است آدرس آی پی آنرا بدانید ! و آنرا وارد نمایید تا بتوانید به آن میزبان دسترسی داشته باشید. اما اگر تنها یک یا چند میزبان انگشت شمار وجود داشت این موضوع خیلی ساده بود. اما در حال حاضر در دنیای وب در اینترنت ما فقط حد اقل 15 میلیون وب سایت داریم - یعنی حد اقل 15 میلیون آدرس آی پی. این را در نظر بگیرید که یک شرکت بازرگانی در تبلیغات خود ذکر می کرد برای اطلاعات بیشتر به

خاطر سپاري اين ادرس كار ساده اي مي نمود ؟

و کدام ساده تر است ؟ MHDsoft.Com يا 69.126.178.11 و يا [www.Google.Com](http://www.Google.Com) و يا 68.11.134.56 ؟

مطمئنا" شما هم با من موافق هستيد که همين قابليت استفاده از نام بجاي آدرس آی پي بوده است که دنياي وب را متحول ساخته است.

DNS دقيقا" همين كار را انجام مي دهد - يعني ترجمه ي نام به آدرس آی پي. براي مثال اينبار که از نرم افزار Microsoft Internet Explorer استفاده مي نماييد دقت نماييد که در منوي اعلان سمت چپ پايين اين نرم افزار چه اتفاقي مي افتد - وقتي وارد مي نماييد [www.Yahoo.Com](http://www.Yahoo.Com) - پس از چند لحظه مشاهده مي نماييد که یک آدرس آی پي در آنجا شکل مي گيرد.

سيستم DNS به اينصورت است که وقتي شما وارد مي نماييد [www.domain.Com](http://www.domain.Com) ابتدا آنچه شما وارد نموديد به سه بخش تقسيم مي شود. یک بخش با عنوان Subdomain يا زیر شاخه ي Domain که در مثال بالا www است. بخش ديگر خود Domain که در مثال همان Domain است و بخش ديگر دامنه ي بين الملبي آن که در مثال بالا .Com است.

در دنيا چند ميزبان وجود دارند که بخش عمده ي آنها در ايالات متحده است. يکي از اين ميزبانها يا SERVER ها مسئول نگهداري دامنه هاي .Com است. براي مثال ميزبان ديگري در جمهوري اسلامي ايران است که مسئول نگهداري دامنه هاي .IR است. پس چون در مثال [www.Domain.Com](http://www.Domain.Com) دامنه ي بين المللي .Com است ابتدا ISP محل شما درخواستي به ميزبان DNS بين المللي مي فرستند و از آن آدرس IP مربوط به DNS مربوط به نام Domain مي شود. آن ميزبان براي مثال جواب مي فرستد که آدرس IP ميزبان DNS مربوط به دامنه ي Domain برابر است با 127.0.0.1 سپس ISP شما به آدرس آی

دامنه ي WWW چه است و ان ميزبان ( يعني 127.0.0.1 ) ادرسي را بر مي گرداند مانند 127.0.0.2 . درواقع Subdomain آخرين مرحله اي است که در بازخواني نام دامنه وجود دارد.

مثالي ديگر : [www.MHDsoft.Com](http://www.MHDsoft.Com)

ابتدا ISP شما به آدرس ميزبان بين المللي دامنه ها ي Com . که آدرس آي پي آن قبلا" در DNS SERVER مستقر در ISP شما قرار دارد پيامي مي فرستند که آدرس آي پي مربوط به دامنه ي MHDsoft چه است ؟  
آن سيستم جواب مي دهد ( براي مثال ) 66.12.13.15 و سپس ISP شما به آدرس آي پي 66.12.13.15 سوالي مي فرستند ميني بر اينکه آدرس IP مربوط به زير دامنه ي WWW چه است و سپس ميزبان جواب مي دهد ( براي مثال ) 99.123.23.10 .

مثالي ديگر: [Download.mhdsoft.com](http://Download.mhdsoft.com)

ابتدا به DNS بين المللي Com . ارسال مي شود که آدرس آي پي دامنه ي MHDsoft چه است ؟ آن ميزبان جواب مي دهد که 66.12.13.15 است. سپس ISP شما به ميزبان 66.12.13.15 سوالي ارسال مي کند که آدرس آي پي زير دامنه ي Download چه است ؟ و آن بر مي گرداند 88.20.17.18 .

اما درخواستهايي که به ميزبانها ارسال مي شود فقط محدود به وب نمي باشند. نمونه اي از جواب يک DNS SERVER به درخواست به صورت ذيل است :

Query for [www.MHDsoft.Com](http://www.MHDsoft.Com)

DNS SERVER: 127.0.0.1

A record 88.12.13.46

MX record 88.12.12.1

...

برای مثال اگر شما بخواهید یک نامه از طریق پست الکترونیک به [Info@MHDsoft.Com](mailto:Info@MHDsoft.Com) بفرستید میزبان ایمیل شما یا SMTP OUTGOING از طریق DNS آدرس MX را می خواند - یعنی میزبانی که بر روی پورت 25 جهت ارسال و دریافت پست الکترونیکی فعال است.

اما برای دسترسی به وب سایت فقط به A record بجای MX record نیاز است. یعنی میزبانی که بر روی پورت 80 جهت دسترسی به وب است.

( شماره ی پورت و مفاهیم آن در بخش برنامه نویسی پریش شرح داده خواهد شد )

محل میزبان DNS دوم ( یعنی بعد از بین المللی ) می تواند در محل میزبان وب و ایمیل باشد و هر سه اینها می توانند یکی باشند.

وقتی شما یک دامنه اجاره می کنید - برای مثال [www.test.net](http://www.test.net) هنگام ثبت این دامنه از شما خواسته می شود حداقل آدرس یک DNS به شرکت یا فرد موجر اعلام کنید. در هر دامنه شما می توانید به صورت استاندارد حد اقل یک آدرس DNS و حد اکثر 10 آدرس NS وارد نمایید. منظور از آدرس NS چیزی مانند [Ns1.MHDsoft.Com](http://Ns1.MHDsoft.Com) است - یعنی آدرس میزبان DNS برای دامنه ی [www.test](http://www.test) که زیر شاخه ی Net. و یا Com. است.

شاید این سوال برای شما پیش آید که چرا چند آدرس DNS و چرا نه فقط یکی ؟

فرض کنید که شما یک وب سایت بسیار مهم دارید. یک میزبان وب سایت شما در محل شرکت شما در تهران است. هر لحظه امکان دارد به دلایلی خط ارتباطی اینترنت میزبان شما قطع شود - حتی برای چند لحظه. اما شما می دانید که حتی یک لحظه قطع شدن وب سایتان برای شما دردسر ساز است. بنا بر این شما میزبان دیگری در



DNS server نیز فرار می دهد ( نرم افزار ) به صورت Ns1.Mydomain.Com برای میزبان تهران و Ns2.Mydomain.Com برای میزبان لندن. در صورتیکه به هر دلیلی میزبان تهران از کار بیفتد ISP هایی که می خواهند به میزبان شما دسترسی داشته باشند از طریق Ns2.Mydomain.Com اقدام می کنند بجای Ns1.Mydomain.Com و هر گاه که مشکل میزبان شما در تهران بر طرف شود دوباره همه ی کاربران به میزبان تهران شما متصل می شوند.

برای مثال میزبان سایت Google.Com دارای بیش از 20 میزبان DNS است.

برای مثال سایت Yahoo که خدمات پست الکترونیک رایگان می دهد - در صورت شلوغی بیش از حد میزبانهای پست الکترونیک به صورت خودکار بر اساس اطلاعات DNS تغییر می کنند تا ترافیک به صورت عادلانه پخش شود.

همانطور که متوجه شدید - دریافت آدرس آی پی آخرین یک میزبان وب و یا یک میزبان پست الکترونیک کمی زمان بر است. بر همین اساس - اکثر ISP ها جهت افزایش سرعت ارتباطی کاربرانشان آدرسهای DNS و IP های منسوب به آنها را Cash می نمایند و یا در حافظه ای موقت نگهداری می کنند.

در برخی مواقع پیش می آید که شما به میزبان وب خود بیشتر از یک خط اینترنت متصل می نمایید و به همان نسبت چند نرم افزار میزبان DNS بر روی همان کامپیوتر میزبان وب اجرا می نمایید. برای مثال فرض کنید شما یک خط فیبر نوری و سپس یک خط ADSL برای میزبان وب خود فراهم آورده اید که اگر چنانچه فیبر نوری شما قطع شد بتوانید از ADSL به عنوان پشتیبان استفاده نمایید. در چنین شرایطی آدرس IP خط فیبر نوری شما با آدرس IP خط ADSL متفاوت خواهد بود - اما هر دوی آنها به یک میزبان وب منتهی می شوند ( یعنی از نظر فیزیکی یک دستگاه کامپیوتر ) در اینصورت شما نیز می توانید آدرس DNS اول خود را همان آدرس IP خط فیبر نوری معرفی کنید که بر روی

شما نیز فعال است و ادرس DNS دوم خود را بر روی IP خط ADSL که بازهم مشابه قبل است. در اینصورت اگر به هر دلیلی خط فیبر نوری از کار بیفتد - خط ADSL بلافاصله جایگزین آن می شود.

ارتباط بین میزبانهای DNS بر اساس پروتکل UDP بجای TCP/IP می باشد و در واقع اینرا می توان پاکتهای اطلاعاتی غیر همزمان نام برد. شماره ی پورت آن 53 می باشد. البته برخی از DNS ها از پورت 53 بر روی TCP/IP نیز پشتیبانی می کنند.

- Html
- Xml
- ASP
- PHP
- JSP
- Cgi
- Perl
- Java script
- VB script
- Bat

و زبانهای دیگری که ممکن است بر اساس نوع برنامه ریزی صاحبان وب سایتها تولید شوند.

همانطور که در بحث مربوط به پروتکل بدان اشاره شده - منظور از وب هاستینگ و یا به عبارتی WEB HOSTING همان سیستم میزبان وب است. اما بهتر است نگاهی دقیقتر به آن بیندازیم. وب هاستینگ از نظر سخت افزاری یک دستگاه کامپیوتر به انضمام مقدار زیادی فضای دیسک سخت است که به وسیله ی یک یا چند خط اینترنت ( بنا بر تنظیمات DNS ) به شبکه ی جهانی اینترنت متصل شده است.

بر روی کامپیوتر مربوط به وب هاستینگ یک سیستم عامل نصب شده است که از آن جمله می توان به Microsoft Windows NT 4.x و یا Microsoft Windows 98 SE و یا Microsoft Windows 2000 all editions اشاره نمود و برخی از مشتریان ترجیح می دهند از سیستم عاملهایی مانند UNIX و یا Linux استفاده نمایند.

بر روی این سیستمهای عامل نرم افزارهایی مستقل و یا وابسته به سیستم عامل ( بستگی به سیستم عامل دارد ) جهت انجام خدماتی مانند میزبانی وب - میزبانی پست الکترونیکی و میزبانی DNS و FTP اجرا شده اند.

در کنار آندسته از نرم افزارها - نرم افزارهای دیگری مانند Firewall و AntiVirus نیز اجرا شده اند که کامپیوتر وب هاستینگ را از گزند حملات نفوذگران و یا هکرها و همچنین دریافت و ارسال فایلها و ویروسی از طریق FTP و یا E-mail جلوگیری نمایند.

بر روی هر کامپیوتر وب هاستینگ بسته به پهنای باند شبکه که دارد تعداد یک و یا بیشتر از یک وب سایت در حال میزبانی است و اگر هدف از راه اندازی این وب هاستینگ اجاره ی فضا بوده است - بنا بر این یک نرم افزار مدیریت کاربران و یا Accounting که به صورت مستقیم با نرم افزار میزبانی وب و پست الکترونیکی و FTP در ارتباط است نیز بر روی این کامپیوتر اجرا شده است.

نکته مهم در این نوع از سیستمها - نرم افزاري Backup مي باشد. این نرم افزار وظیفه دارد در یک دوره ي زمانی خاص - براي مثال - هر یک هفته و یا شاید هر 24 ساعت یکبار از کل اطلاعات سیستم رونوشت تهیه کند که اگر بنا به هر دلیلي اطلاعات سیستم از بین رفت - صاحب سیستم بتواند سرریعا آنها به سیستم بازگرداند.

مستاجران فضاي وب سایتها از طریق نرم افزار FTP ( File Transfer Protocol ) به آدرس Domain خود ( بر اساس آنچه در DNS شرح داده شده ) و یا به آدرس IP خاصي متصل مي شوند و از طریق پورت 20 یا 23 ( در برنامه نویسي پریرز شرح داده خواهد شد ) اقدام به جا به جايي اطلاعات مي نمایند. منظور از جا به جايي اطلاعات Upload و Download صفحات وب طراحی شده است. به نمودار ساده ي ذیل توجه فرمایید :



همانطور که در نمودار بالا مشاهده نمودید کاربران ممکن است بخواهند دسترسی هاي متفاوتي به میزبان وب و یا به سیستم وب هاستینگ داشته باشند - اما از نظر فیزیکی همه ي آنها به یک دیسک سخت و یا هارد دیسک دسترسی دارند.

براي مثال سایتهاي ذیل همگی بر روی یک میزبان وب قرار دارند :

[www.Parsian-Invest.Com](http://www.Parsian-Invest.Com)

[www.Mepekco.Com](http://www.Mepekco.Com)

و وقتي ادرس IP مربوط به DNS انها در Record مربوط به A را بررسی می کنید همه ی آنها ادرس ( برای مثال ) 127.0.0.1 را نشان می دهند. اما وقتي شما در نرم افزار مرور گر وب می خواهید Parsian- Invest.Com را ببینید صفحه ای متفاوت با MHDsoft.Com را به شما نشان می دهد.

اگر بخواهیم سیستم وب هاستینگ را از این دریچه بررسی کنیم به این نتیجه می رسیم که بر روی سیستم وب هاستینگ پوشه های مختلفی ( Folder ) وجود دارد که هر یک مربوط به یک وب سایت می شود.

در مثالی ساده تر می توان اینطور بیان کرد که وقتي شما تایپ می نمایید parsian-invest.com شما درواقع به C:\Parsian-invest.com دسترسی دارید و وقتي تایپ می نمایید [www.Parsian-invest.Com](http://www.Parsian-invest.Com) به C:\parsian-invest.com\www\ دسترسی دارید و وقتي تایپ می نمایید Download.MHDsoft.Com شما به C:\Mhdsoft.Com\Download\ دسترسی دارید.

به زبانی دیگر - وقتي ادرس IP میزبان از DNS پرسیده می شود ( مرحله ی آخر ) سپس نرم افزار مرورگر اینترنت شما ( مانند Microsoft Internet Explorer و یا FireFox ) به ادرس IP میزبان اعلام می کند که من می خواهم به [www.Parsian-invest.Com](http://www.Parsian-invest.Com) و یا [www.MHDsoft.Com](http://www.MHDsoft.Com) در محیط وب شما دسترسی داشته باشم.

اگر شما ادرس IP یک میزبان وب را بدست آورید که چندین وب سایت بر روی آن فعال است و فقط ادرس آی پی را در Address bar مرور گر خود وارد نمایید مانند این است که بگویید می خواهم به C:\ یا اصطلاحاً "به Root دسترسی داشته باشم".

مسئله ی بعدی در سیستم وب هاستینگ آن است که مدیریت آنکه کدام پوشه ( Folder ) به کدام وب سایت تعلق دارد با نرم افزار میزبان وب ( Web server ) است. در این نرم افزار وقتي اپراتور نرم افزار وارد

از او نام پوشه ای که فایل‌های وی بر روی آن است را جویا می‌شود و مثلاً "اپراتور می‌تواند C:\ یا D:\ یا C:\MHD یا C:\MHDsoft.Com را مشخص نماید.

سپس اپراتور باید Root folder مربوط به Subdomain را مشخص کند. منظور از این قسمت آن است که اگر بازدید کننده ای وارد نمود [Http://www.MHDsoft.Com](http://www.MHDsoft.Com) به فایل‌های کدام پوشه دسترسی داشته باشد؟ در این صورت اگر اپراتور وارد نماید C:\MHDsoft.Com\ درواقع به این معنی است که اگر بازدید کننده ای وارد کرد [www.MHDsoft.Com](http://www.MHDsoft.Com) و یا MHDsoft.Com هر دو به C:\MHDsoft.Com\ هدایت شوند ولی اگر اپراتور وارد نماید C:\MHDsoft.Com\www\ به این معنی است که اگر بازدید کننده ای وارد نمود MHDsoft.Com به C:\MHDsoft\ دسترسی داشته باشد و اگر وارد نمود [www.MHDsoft.Com](http://www.MHDsoft.Com) به C:\MHDsoft.Com\www\ دسترسی داشته باشد.

در مرحله ی بعد - اپراتور باید نام فایل مرجع را وارد نماید. این بدان معنی است که اگر کاربری وارد نمود <http://www.mhdsoft.com> در عمل به C:\mhdsoft.com\www\ دسترسی خواهد داشت اما چه چیز قرار است مشاهده نماید؟ در پوشه ی C:\mhdsoft.com\www\ ممکن است یک و یا شاید صدها فایل وجود داشته باشد.

در اینجا ممکن است جواب‌های متعددی اپراتور در این زمینه داشته باشد. برای مثال ممکن است بگوید در این شرایط فهرست فایل‌های موجود در این پوشه را به کاربر نشان بده که هر کدام را خواست بتواند به سلیقه ی خودش انتخاب نماید که اصطلاحاً "به این امکان Directory listing می‌گویند. جواب دیگر آن است که فایلی به نام فایل مرجع انتخاب شود. مثلاً فایلی با نام Welcome.htm معرفی شود که اگر کاربری خواست به این پوشه دسترسی داشته باشد این فایل برای وی ارسال شود. برخی از میزبان‌های وب فایلی با نام Index.htm را پیش فرض خود قرار داده‌اند.

( [Http://www.Parsian-invest.com/hello.htm](http://www.Parsian-invest.com/hello.htm) ) به این معنی است که می خواهید به فایل Hello.htm در پوشه ی C:\parsian-invest.com\www\ دسترسی داشته باشد و آنرا دانلود نماید.

و یا منظور از [Http://Download.mhdsoft.com/web.zip](http://Download.mhdsoft.com/web.zip) این است که می خواهیم فایل web.zip را به کامپیوتر خود از پوشه ی C:\mhdsoft.com\download\ انتقال دهیم.

همانطور که در بخشهای قبلی نیز توضیح داده شد - وقتی شما نام فایل HTML و یا مشابه اینرا وارد می نمایید نرم افزار میزبان وب آنرا برای شما ارسال می کند و برای نرم افزار میزبان فرقی نمی کند که شما فایل HTML درخواست نموده اید و یا فایل JPG و یا فایل EXE و یا ZIP . در هر صورت شما دارید یک فایل را دانلود می نمایید - اما برای برخی از فایلها مانند ASP که دارای کدهای پردازشی هستند - این بدان معنا است که قبل از ارسال فایل پردازشی در سرور انجام شود و نتیجه ی پردازش به عنوان محتویات فایل به کاربر ارسال شود.

وقتی که شما یک فایل HTML را دانلو می نمایید ممکن است شامل عکس - صدا و ... باشد. در اینصورت این نرم افزار مرور گر وب شما است ( مانند FireFox و یا Netscape و یا Ms-IE ) که به ترتیب آنها را می خواند و از میزبان وب دانلو می نماید.

برای مثال به تکه کد ذیل توجه فرمایید :

```
<title> MHDsoft test page </title>
<Center> <img src=welcome.jpg ></Center>
<Center> Welcome to MHDsoft International </Center>
<Center> <A href=contact.htm >Contact us </a></center>
```

ما تکه کد بالا را در فایلی با نام test.htm قرار می دهیم و به همراه یک فایل تصویری که شکل ذیل در آن است و نام آن Welcome.jpg

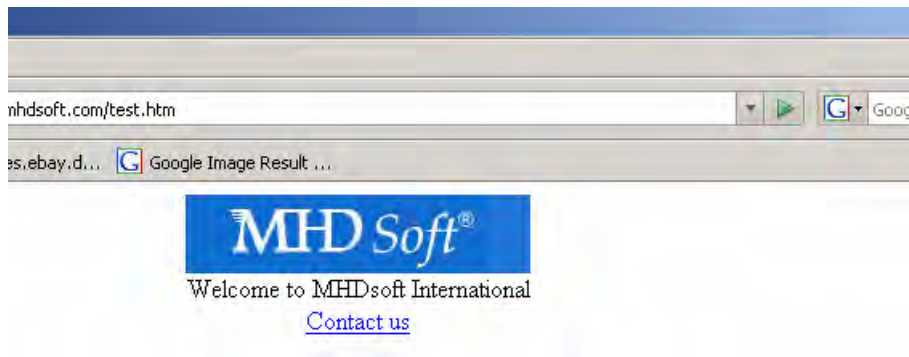


دهيم:

: محتوي فايل Welcome.jpg



سپس از طريق نرم افزار مرور گري مانند FireFox آدرس [Http://www.MHDsoft.Com/test.htm](http://www.MHDsoft.Com/test.htm) را وارد مي نماييم و صفحه ي ذيل را مشاهده مي نماييم :



در بالاي صفحه ي مرور گر ( سمت چپ در منوي آي رنگ ) آنچه را بين دو تگ <title> و </title> قرار داده ايم - قرار گرفته است . در بالاي صفحه تصوير فايل Welcome.Jpg . در اينجا مرور گر FireFox وقتي به تگ <img src > برخورد کرده است متوجه شده است که در اين قسمت بايد یک تصوير قرار دهد. نام تصوير Welcome.Jpg است - پس به [Http://www.mhdsoft.com/](http://www.mhdsoft.com/) ارسال کرده است که من فايل Welcome.Jpg را مي خواهم. در پايين صفحه یک ارتباط يا لينک ( Link ) وجود دارد که در متن HTML ما آمده است هر کس بر روي اين لينک

نماید.)

در کنار این موضوع وقتی که صاحب وب سایت MHDsoft و یا Parsian- invest بخواهند فایل‌های طراحی وب سایت خود را درون پوشه‌های خود قرار دهند معمولاً با استفاده از نرم افزار FTP مانند Cute FTP و یا Leap FTP استفاده می‌نمایند. در اینصورت پس از وارد نمودن Username و Password خود به ROOT پوشه‌ی خود که در مثال MHDsoft همان : C:\MHDsoft.com است دسترسی دارند. در این پوشه ممکن است پوشه‌های دیگری نیز وجود داشته باشند مانند : WWW و ... که صاحب وب سایت می‌داند این پوشه از طریق [www.mhdsoft.com](http://www.mhdsoft.com) قابل دسترسی است و یا Download که می‌داند از طریق [Download.mhdsoft.com](http://Download.mhdsoft.com) قابل دسترسی است. و یا ممکن است پوشه‌هایی دیگر داشته باشد که در صورت تعریف نشدن آن در Subdomain مانند [www](http://www) و [Download](http://Download) می‌تواند به صورت پس آدرس به آن دسترسی برقرار کند. برای مثال - اگر بخواهد به [www](http://www) بدون subdomain دسترسی داشته باشد می‌تواند از آدرس ذیل استفاده نماید : [Http://mhdsoft.com/www](http://mhdsoft.com/www) به شرط آنکه اپراتور میزبان وب پوشه‌ی C:\MHDsoft.Com را به عنوان پوشه‌ی اصلی قرار داده باشد.

بر همین اساس صاحب MHDsoft ممکن است فایل‌های تصویری خود را در یک پوشه قرار دهد که آن پوشه ارتباطی با Subdomain نداشته باشد و یا داشته باشد.

به طور کلی اساس تمامی نرم افزارهای میزبان وب به این صورت است که گذشت که مجموعه‌ی اینها را وب هاستینگ می‌نامند.

به طور كلي با تجاري شدن وب رقابت بر سر ارايه ي بهترين نرم افزار وب و اضافه نمودن امكانات ويژه بدان شدت گرفته است. در نمودار ذيل نام نرم افزارهاي ميزبان وب مطرح دنيا به انضمام MHD WEB SERVER كه برنامه نويسي آن توسط اينجانب ارايه شده است و توسط شركت سرمايه گذاري پارسيان خريداري و در حال حاضر استفاده مي شود آورده شده است.

Key Features	Zeus Web Server v4	MHD web server	Sun Java Web Server 6.1
<b>Core Technology</b>			
HTTP/1.1 support (incl. KeepAlives and Pipelining)	✓	✓	✓
Typical Hosting Capability (# sites)	1 million+	5 million+	< 1,000
Dynamic reconfiguration without restart	✓	✓	✗
Templated web site deployment	✓	✓	✗
On-demand web site	✓	✗	✗

High scalability process model	✓	✓	✗	✗
Optimization for Solaris Network Cache Accelerator (NCA)	✓	✓	✓	✗
Linear SMP scalability	✓	✓	✗	✗
Configurable log format	✓	✓	✓	✓
Customisable error pages	✓	✓	✓	✓
Dynamic log rotation	✓	✓	✓	—
Bandwidth throttling	✓	✓	✗	—
Basic Server-Parsed HTML (SSI) support	✓	✓	✓	✓
Dynamically-generated SSI support	✓	✓	✗	✗ ✓
Automated self-diagnostics	✓	✓	✗	✗
Security				
Anti-Denial-of-Service protection	✓	✓	✗	✗
HTTP request filtering	✓	✓	✗	✗
Malformed	✓	✓	✗	✗

protection				
Integrated attack monitoring and logging	✓	✓	✗	✗
Integrated SSL v2 and v3	✓	✓	✓	✗
Hardware crypto support	✓	✓	✓	✗
SSL certificate management	✓		✓	✗
Clustered SSL certificate deployment	✓		✗	✗
Management				
Web-based user interface	✓	✓	✓	✗
Secure, web-based remote management	✓	✓	✗	✗
Clustered web site management	✓	✓	✓	✗
Group-based web site configuration	✓	✓	✗	✗
Integrated licence management	✓	✓	✗	✗
Configuration wizards	✓	✓	✗	✗
Scriptable (command-line) configuration tools	✓	✓	✓	✗

web-based statistics	✓	✓	✓	✗
SNMP monitoring	✓	✓	✓	✗
Delegated user management (htaccess)	✓	✓	✓	✓
Scalable subserver support	✓	✓	✗	✗
<b>Web Application Support</b>				
NSAPI	✓	✓	✓	✗
ISAPI filters	✓	✓	✗	✗
ISAPI extensions	✓	✓	✗	✗
FastCGI	✓	✓	✗	✗
CGI	✓	✓	✓	✓
Generic FastCGI/CGI handler support	✓	✓	✗	✗
Secure CGI sandboxing	✓	✓	✗	✗
Limit CGIs by CPU or memory use	✓	✓	✗	✗
LDAP	✓		✗	✗
IP address authentication	✓	✓	✓	✓
DNS name authentication	✓	✓	✓	✓
User-based authentication	✓	✓	✓	✓

base authentication	✓	✓	✗	✗
Perl support	✓	✓	✓	✓
Basic PHP support	✓	✓	✓	✓
Secure setuid PHP support	✓	✓	✗	✗
XML support	✗	✓	✗	✗
Active Server Pages (ASP) support	✗	✓	✗	✗
Request rewriting	✓		✗	✗
Access Control APIs		ISAPI	NSAPI NSACL LDAP	Fast mod l
<b>Content Publication Support</b>				
Basic Microsoft® FrontPage™ support	✓	✓	✗	✗
Secure sandboxed Microsoft® FrontPage™ support	✓	✓	✗	✗
Subserver Microsoft® FrontPage™ support	✓	✓	✗	✗
Netscape™ Composer	✓	✓	✓	✗

Feature	Product A	Product B	Product C
<b>support (POT)</b>			
WebDAV support	☐	✓	✗
Web Folders (WebDAV)	☐	✓	✗
PICS ratings	☐	✓	✗
Static and dynamic content compression	✓	✓	✗
Search engine	✓	✓	☐
<b>Java</b>			
Java Servlets	☐		✓
JavaServer Pages (JSP)	☐		✓
Native JVM support	☐		☐
<b>System Requirements</b>			
<b>Minimum RAM</b>	32MB	16 MB	64MB per CPU
<b>Recommended disk space</b>	10MB	5 MB	110MB
<b>Platforms Supported</b>	AIX Compaq Tru64 FreeBSD HP-UX HP-UX IA64 IRIX Linux IA32 Linux IA64	Microsoft Windows 9x	HP-UX 11.0 HP-UX 11.0 IA64 (through binary compatibility) Linux IA32 Solaris SPARC Windows NT/2000



Linux Alpha  
Linux PPC  
MacOSX  
Solaris x86\_64  
Solaris SPARC  
Solaris IA32

پس از بررسی این جدول شما می توانید بهترین نرم افزار میزبان وب را جهت انجام پروژه ی خود انتخاب نمایید و یا بر اساس این جدول بتوانید امکانات نرم افزاری که در دست طراحی دارید را مشخص نمایید ( به عبارتی بهترین امکانات را در نرم افزار میزبان وب خود قرار دهید.

## است

همانطور که در بخشهای قبلی توضیح داده شد سخت افزار میزبان وب شامل یک کامپیوتر است که به یک خط یا چند خط اینترنت متصل شده است.

بنا بر کاربرد میزبانی وب شما سخت افزار آنهم تغییر می کند. برای مثال اگر شما بخواهید این سخت افزار را در اختیار مشتریان خود قرار دهید و به آنها فضا اجاره دهید باید موارد انتظار آنها را در نظر بگیرید که معمولاً شامل موارد ذیل است :

- قابل مشاهده بودن وب سایت در تمامی ساعات شبانه روز در طول مدت اجاره
- امکانات پست الکترونیک
- امکان تعریف Subdomain
- امکان دسترسی به FTP
- پشتیبانی از وب سایت

برای این منظور - شما باید حد اقل 2 خط اینترنت قابل اطمینان ( یعنی مطمئن باشید که این خط قطع نمی شود ) و یک باتری پشتیبان برق حد اقل 4 ساعته ( در صورت قطعی برق شهر - میزبان شما خاموش نشود ) و یک سیستم با پردازنده ی خوب و ایده آل داشته باشید که در مدت روشن بودن میزبان با مشکل مواجه نشود.

برخی از شرکتهای معتبر دنیا از آنجا که سرمایه گذاری عمده ای در زمینه ی WEB SERVER و WEB HOSTING نموده اند - از میزبانهای پیشرفته ای استفاده می کنند که معمولاً چند پردازنده ای هستند که این امر به آنها این امکان را می دهد که علاوه بر دارا بودن سرعت زیاد در ارتباط - در صورتیکه هر یک از پردازنده ها با مشکل مواجه شود دیگری بلافاصله جایگزین شود.

باطری ها و ژنراتورهای تولید برق استفاده می کنند که این امر به مشتریان این اطمینان را می دهد که میزان آنها صد در صد روشن است. برخی از شرکتها میزانهای خود را در ساختمانهای مخصوصی که دارای Air condition های بسیار خنک و سازه ای ضد زلزله و ضد حملات موشکلی است قرار می دهند ( IDC ) . ( که در رابطه با این موضوع در بخش IDC توضیح کامل داده خواهد شد ).

بنا بر این بر اساس نیاز مشتری و نیاز خود شما است که می توانید میزان وب خود را انتخاب نمایید. در این رابطه نباید چشم و همچشمی کرد. به یاد دارم اولین میزان وبی که خودم شخصا راه اندازی نمودم یک دستگاه کامپیوتر پنتیوم 120 مگاهرتزی با 16 مگابایت رم بود. که به راحتی توانست 5 وب سایت را میزبانی نماید.

اما در حال حاضر شرکت سرمایه گذاری پارسیان یک میزان وب پنوم 3 گیگاهرتزی با 500 مگابایت فضای دیگر و یک گیگابایت رم برای میزبانی 3 وب سایت دایر کرده است و نوعی آینده نگری منطقی را در این رابطه در نظر گرفته اند.

RFC ها به طور کلی منابع و مراجعی هستند که برنامه نویسان می توانند جهت ایجاد برنامه های کاربردی به آنها رجوع نمایند. برای مثال در یک RFC مربوط به میزبان وب مطالبی در مورد استاندارد ارتباطی و Header ها ( که در بخشهای بعدی به آنها اشاره می شود ) اشاره شده است.

RFC ها شامل استانداردهای مورد نیاز یک برنامه هستند که از طریق آنها بتوانند با دیگر برنامه های موجود ارتباط برقرار کنند. به طور کلی شرکت های برنامه سازی برای اینکه بتوانند برنامه های استانداردی داشته باشند باید از RFC ها که استانداردهایی عمومی هستند پیروی کنند.

در مراجع وب ( وبسایتهای موجود در وب ) RFC مربوط به وب با نام HTTP و یا Hyper Text Transfer Protocol معرفی شده است.

روایتهای مختلف این RFC که از سال 1990 منتشر شده است شامل Http/0.5 و Http/1 و Http/1.1 می باشد.

شما می توانید به سادگی با جستجوی RFC HTTP در موتورهای جستجویی مانند Google و یا Yahoo! به این سری از RFC ها دسترسی داشته باشید.

در رابطه با ویژگیهای الگوریتم یک نرم افزار وب هاستینگ و یا به طور کلی نرم افزار میزبانی وب - باید ابتدا انتظارات خود را در رابطه با این نرم افزار در نظر بگیرید و سپس استانداردهای وب را که معمولاً در RFC ها در نظر گرفته شده اند و سپس محدودیتهای خود را.

مورد آخر که در مورد محدودیتهای موجود می باشد می تواند نقطه ی کلیدی در طراحی الگوریتم نرم افزار میزبانی وب شما باشد.

این بدان معنی است که انتظارات شما از نرم افزار میزبان وب باید با در نظر گرفتن محدودیتهای موجود صورت پذیرد. نمونه ای از این محدودیتهای به شرح ذیل می باشند:

- پهنای باند شبکه ی شما
- تعداد بازدید کنندگان همزمان وب سایت شما
- شرایط On بودن میزبان ( Server )

بر اساس این محدودیتهای می توان انتظارات ذیل را در نظر گرفت :

- حد اکثر 2 بازدید کننده ی همزمان
- میزبان به صورت کاملاً "ON" در کلیه ی ساعات شبانه روز در مدت 1 سال
- حد اکثر حجم دانلود اطلاعات برابر با 100 کیلوبایت برای هر بازدیدکننده

در اینصورت شرایط ذیل می تواند برای نرم افزار میزبان وب شما پیش آید :

- در نهایت دو کاربر بتوانند همزمان از وب سایت بازدید به عمل آورند - این بدان معنی است که اگر دو کاربر مشغول دانلود اطلاعات وب سایت شما باشند - یعنی همزمان به وب سایت

بازدید به عمل آورد.

- هر کاربر می تواند در نهایت 100 کیلوبایت اطلاعات را در هر بار درخواست دانلود نماید و در صورتیکه سعی کند اطلاعات بیشتری را دانلود نماید - فایل مورد تقاضای وی ناقص دانلود می شود

با توجه به مواردی که شرح آن گذشت اینطور به نظر می رسد که شما در تهیه ی یک الگوریتم مناسب باید در دو سطح الگوریتم را تبدیل به یک سطح الگوریتم نمایید. دقیقاً" به مانند آنچه در مورد مسایل پروتکل بیان شد. اگر به یاد داشته باشید ما در بخش مربوط به پروتکل یک سطح پروتکل سریع و ساده را مطرح نمودیم و سپس این مسئله مطرح شد که دیگر پروتکلها نیز بر اساس آن پروتکل ساده و سریع طراحی می شوند.

این مسئله دقیقاً" در مورد الگوریتم وب هاستینگ و یا میزبانی وب نیز صدق می کند. زیرا اگر از یک سطح به این الگوریتم نگاهی بیاندازیم درواقع ما در صدد تهیه ی یک الگوریتم برای پروتکل وب هستیم و از سطح و منظر دیگر محدودیتها و نیازهای ما باید بر روی یک پروتکل وب ساده پایه ریزی شوند.

برای اینکه بتوانیم یک الگوریتم پایه ای مناسب برای میزبانی وب طراحی نماییم باید اول با اصول کارکرد پروتکل HTTP در وب آشنا شویم. در بخشهای قبلی اشاراتی به نحوه ی کارکرد این پروتکل شد اما در این بخش ناگزیر باید به صورت کامل بخشهای مهم و اساسی و کلیدی ارتباطات بین میزبان وب و نرم افزارهای مرور گر بیان شود.

این استانداردها بر اساس RFC مربوط به HTTP/1.1 می باشد.

به طور کلی پس از برقراری اتصال شما با شبکه ای مانند اینترنت کامپیوتر شما از شبکه ی میزبان یک شناسه دریافت می کند که این شناسه ممکن است ثابت باشد و یا متغیر.

بتوانید از خدمات وب در آن استفاده نمایید. به این منظور شما ناکزیر به استفاده از نرم افزارهایی با عنوان مرور گر وب مانند MS-IE و یا M-FireFox هستید.

تاریخچه ی این نوع از نرم افزارها به سالهای 1993 و 1995 بر می گردد که نرم افزاری به نام MOSAIC ابداع شده بود که بعدها توسط تیم Netscape گسترش یافت.

هنگامیکه شما در مرور گر اینترنت خود آدرس یک وب سایت را وارد می نمایید - همانطور که در بخش DNS گذشت آدرس IP مربوط به آن پایگاه وب و یا وب سایت به مرور گر اینترنت شما از جانب میزبان شبکه ی شما اطلاع داده می شود ( نحوه ی این نوع از ارتباطات در بخش برنامه نویسی پریر شرح داده خواهد شد )

سپس این نرم افزار مرور گر وب شما است که از طریق پروتکل وب می بایستی درخواست خود را به آن آدرس IP ارسال نماید و این شروع یک ارتباط محاوره ای در پروتکلها است.

در این هنگام مرورگر وب باید اطلاعاتی به شرح ذیل را به آن آدرس آی پی ارسال نماید :

- متد ارسال اطلاعات ( GET و یا POST )
- روایت پروتکل وب که پشتیبانی می شود
- نام نرم افزار مرورگر وب
- سایتی که درخواست دسترسی به آنرا دارد
- قالبهایی که مورد پذیرش هستند در مورد فایلهایی که این مرور گر می تواند دریافت و بازگشایی نماید
- نوع ارتباط

GET / HTTP/1.1  
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows NT  
4.0) Opera 5.01 [en]  
Host: www.roshd.ir  
Accept: text/html, image/png, image/jpeg, image/gif, image/x-  
xbitmap, \*/\*  
Accept-Language: en  
Accept-Encoding: deflate, gzip, x-gzip, identity, \*;q=0  
Connection: Keep-Alive

در سطر اول مشخص شده است که مرورگر وب از روش ارتباطی GET می خواهد استفاده نماید ( در بخشهای قبلی توضیح داده شده است که ما دو روش عمده ی ارسال اطلاعات به نامهای GET و POST داریم). پس از مشخص شده است که از استانداردهای موجود در HTTP /1.1 که در RFC نیز آمده است پیروی می شود.

در سطر بعدی مشخص شده است که نرم افزار مرور گر وبی که این درخواست را ارسال نموده است سازگار با استانداردهای Mozilla/4 می باشد و سیستم عاملی که این نرم افزار هم اکنون بر روی آن اجرا شده است Windows NT.4 است و نام نرم افزار Opera 5.01 وایت en و یا انگلیسی می باشد و همانند MSIE 5 و یا Microsoft Internet Explorer 5 می باشد و میزبان وب می تواند Opera را مانند MS-IE5 در نظر بگیرد.

معمولا" نرم افزار Microsoft Internet Explorer 4 به بالا در این سطر اطلاعاتی نظیر وضوح صفحه ی نمایشگر کاربر را نیز ارسال می نمایند.



ان را مرور کر وب شما دارد مشخص شده است.

همانطور که در بخش DNS بدان اشاره شد - بر روی یک آدرس آی پی ممکن است بیش از یک وب سایت حضور داشته باشد بنا بر این طبق استانداردهای HTTP /1.1 باید نام میزبان در این بخش معرفی شود. در اینجا میزبان [www.Roshd.ir](http://www.Roshd.ir) معرفی شده است.

در قسمت Accept قالبهای مورد پذیرش مرورگر وب معرفی شده اند. برای مثال منظور از text/html, image/png آن است که اطلاعات فایل HTML را تنها به عنوان TEXT و یا متنی دریافت می کند و اطلاعاتی فایل PNG را به عنوان تصویر در یک صفحه ی وب قبول و دریافت می کند.

این بدان معنی است که اگر کاربر درخواست فایل test.png را داشته باشد مرورگر منتظر دریافت متن مربوط به یک فایل تصویری است

در قسمت بعد زبان مورد پذیرش مرور گر وب شما توصیف شده است که برای مرورگرهایی که بر روی سیستم عاملهای فارسی نصب شده اند معمولاً به صورت ذیل است :

Accept-Language: fa

و یا

Accept-Language: en/fa

و در نهایت نوع ارتباط "Connection" مشخص شده است که در بخش برنامه نویسی پرریز توضیح داده خواهد شد.

پس از ارسال این Header و یا متن از طریق مرورگر شما به آن آدرس IP باید اطلاعات ارسالی شما توسط نرم افزار میزبان وب تفکیک

GET و اطلاعات مربوط به Host می باشند.

برای روش شدن این مهم توجه شما را به این مثال جلب می نمایم.  
اگر شما در مرورگر وب خود وارد نمایید : [www.MHDsoft.Com](http://www.MHDsoft.Com)  
اطلاعات Header ارسالی به شکل ذیل خواهد بود :

```
GET / HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows NT
4.0) Opera 5.01 [en]
Host: Www.MHDsoft.Com
Accept: text/html, image/png, image/jpeg, image/gif,
image/x-xbitmap, */*
Accept-Language: en
Accept-Encoding: deflate, gzip, x-gzip, identity, *;q=0
Connection: Keep-Alive
```

و اگر در مرورگر وب خود وارد نمایید :

[www.MHDsoft.Com/test.htm](http://www.MHDsoft.Com/test.htm)  
در اینصورت Header ارسالی به شکل ذیل خواهد بود :

```
GET /test.htm HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows NT
[4.0) Opera 5.01 [en]
Host: Www.MHDsoft.Com
Accept: text/html, image/png, image/jpeg, image/gif, image/x-
*/* ,xbitmap
Accept-Language: en
Accept-Encoding: deflate, gzip, x-gzip, identity, *;q=0
Connection: Keep-Alive
```

و این توصیف عیناً در هدر دوم تکرار شده است.  
[www.MHDsoft.Com](http://www.MHDsoft.Com)

اما آنچه در این میان متفاوت است عبارت پس از GET/ است. در هدر اول بین GET/ و HTTP/1.1 یک فضای یک کاراکتری موجود است و در هدر دوم بجای آن فضا نام فایل test.htm آورده شده است.

بنا بر این میزبان وب پس از دریافت هدر اول باید متوجه شود که درخواست اول شامل دسترسی به پوشه ی MHDsoft است و دانلود نمودن فهرست پوشه و یا فایل مرجع ( که در بخش وب توضیح داده شد ).

اما با دریافت هدر دوم میزبان وب باید متوجه شود که درخواست دوم شامل دسترسی به پوشه ی MHDsoft است و دانلود نمودن فایلی بانام test.htm .

یکی از علت‌هایی که در موتورهای جستجو مانند Google از روش GET جهت جستجوی اطلاعات استفاده می شود همین سادگی سیستم GET و سرعت بالا در درک آن است.

برای مثال اگر کاربری بخواهد در سایت Google به دنبال کلمه ی Hello در بیش از 10 میلیون وب سایت بگردد کافی است وارد نماید `Google.Com/search=hello` در اینصورت میزبان گوگل هدر ذیل را دریافت خواهد کرد :

```
GET /Search=hello HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows NT
4.0) Opera 5.01 [en]
Host: www.google.com
Accept: text/html, image/png, image/jpeg, image/gif, image/x-
xbitmap, */*
Accept-Language: en
```

دسترسی به شاخه ی Google و سپس فایل Search است.

در اینطور مواقع از آنجا که نرم افزار میزبانی وب به صورت انحصاری نوشته شده است نرم افزار میزبان می داند که باید مقدار متغیر Search را در خودش برابر Hello قرار دهد و عملیات جستجو را انجام دهد و از آنجا که تنها یک وب سایت بر روی این میزبان قرار داد معمولاً" به مقدار Host در هدر توجه نمی شود.

برخی از وب سایتها با استفاده از همین اطلاعات قید شده در هدر می توانند به راحتی آمارگیری نمایند. برای مثال می توانند متوجه شوند که چند درصد از بازدید کنندگان وب سایتشان از سیستم عامل Microsoft windows xp استفاده می نمایند و یا چند درصد از بازدید کنندگان وب سایتشان از نرم افزار مرورگر Firefox استفاده می نمایند.

فکر می کنم در حال حاضر شما نیز تا حدودی با الگوریتم مناسب جهت یک میزبان وب آشنا شده اید. در این الگوریتم شما باید شبهه گاهمهای ذیل را طری نمایید :

1. منتظر برقراری ارتباط بمان
2. آیا ارتباطی برقرار شده است ؟ اگر خیر به گام 1 برو
3. منتظر دریافت هدر بمان
4. آیا هدر دریافت شده است ؟ اگر خیر به 3 برو
5. مقدار GET را از هدر بخوان
6. مقدار HOST را از هدر بخوان
7. در اطلاعات جستجو کن مقدار HOST مربوط به کدام شاخه در دیسک سخت می شود
8. آیا شاخه ای برای HOST وجود دارد ؟ اگر بله به 11 برو
9. متن "Could not find the web page" را ارسال کن
10. ارتباط را قطع کن - به گام 1 برو
11. آیا GET دارای مقدار است ؟ اگر بله به گام 13 برو
12. نام فایل را Index.htm قرار بده و به گام 14 برو

14. فایل را باز کن و محتویات انرا ارسال کن

15. ارتباط را قطع کن

16. به گام 1 برو

الگوریتمی که در بالا شرح داده شد یک الگوریتم بسیار ساده است که می تواند مبنای الگوریتم تمامی میزبانهای وب قرار گیرد. اگر خوب به این الگوریتم دقت نمایید متوجه می شوید که شباهتهای زیادی با الگوریتم سریع که در بخش پروتکل شرح داده شد دارد.

دقیقا" تمامی مسایلی که در ارتباط با بسط و گسترش پروتکل بررسی کردیم در مورد این الگوریتم میزبان وب ساده نیز صدق می کند.

تنها تفاوتی که میان این میزبان وب و پروتکل سریع وجود دارد این است که میزبان وب نیازی به ایجاد وقفه جهت حصول اطمینان در رسیدن اطلاعات به مقصد نمی باشد. این وظیفه بر عهده ی پروتکل TCP/IP می باشد که شرح آن در بخشهای گذشته به اجمال گذشت.

مسایلی که شما در مورد این الگوریتم می توانید با توجه به مبحث پروتکلها بررسی نمایید و بسط دهید شامل موارد ذیل هستند :

- تنظیم سرعت ارسال اطلاعات
- جوابگویی به تعداد زیادی از کاربران به صورت همزمان
- کد نمودن اطلاعات ارسالی

و ...

برای مثال در مورد جوابگویی به تعداد زیادی از کاربران به صورت همزمان می توان به روش ایجاد Counter و یا همان انتساب ID که در بحث پروتکلها از آن استفاده نمودیم - نیز در اینجا استفاده نماییم.

## برنامه نویسی پرز

در یک کلام به استفاده از پروتکلهاي مياني مانند TCP/IP در پروتکلهاي سطح بالا مانند WEB ( HTTP ) برنامه نویسی پرز یا Socket programming مي گوييم.

درواقع برنامه نویسی پرز را مي توان گونه اي از برنامه نویسی شي گرا نیز معرفي کرد ( خواننده محترم ؛ جهت ادامه ي مطالب به مفاهيم برنامه نویسی شي گرا نیاز دارید - مي توانید به منابعي مانند زبان برنامه نویسی جاوا و ... رجوع نماييد ).

در برنامه نویسی پرز شما با حالتهاي ارتباطي ذيل سر و کار داريد:

- انتظار
- برقراري ارتباط
- قطع ارتباط
- اطلاعات با موفقیت ارسال شد

که همانطور که مشاهده نموديد و در بخش پروتکل نیز بدان اشاره شد - نیازهاي اوليه ي ارتباطي یک پروتکل را فراهم مي نمايد.

یک ميزبان وب که مبحث اصلي اين کتاب را تشکیل مي دهد همیشه در حالت انتظار قرار دارد. وقتي شما برنامه نویسی پرز را انجام مي دهيد مي توانید در دو حالت ذيل برنامه ي خود را قرار دهيد :

- انتظار
- ارتباط

- میزبان ( Server )
- مشتری ( Client )

در صورتیکه شما پریز خود را در حالت میزبان قرار دهید - برنامه ی شما به حالت انتظار می رود تا برنامه ای دیگر که حالت مشتری دارد با آن ارتباط برقرار نماید.

علی رغم اینکه مبحث این کتاب برنامه نویسی میزبان وب است و تنها نیاز ما در برنامه نویسی پریز حالت مشتری آن است - اما صلاح دانسته ام که هر دو حالت برقراری ارتباط در پریز را در این بخش بررسی نمایم.

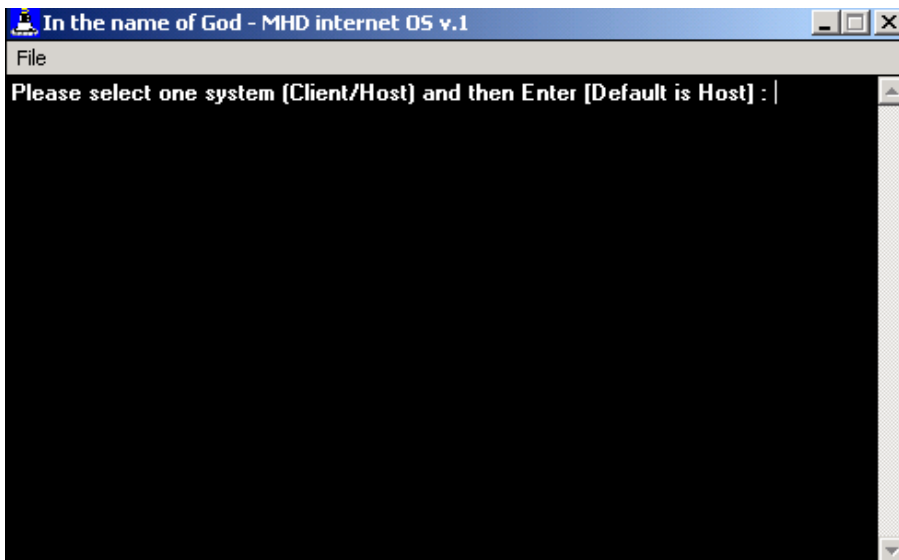
برای آنکه آموزش این بخش بتواند صورت عملی تری داشته باشد به خواننده ی محترم پیشنهاد می کند برنامه ی ذیل را از سایت MHDsoft دانلود نماید :

[Http://www.MHDsoft.Com/IOS.Zip](http://www.MHDsoft.Com/IOS.Zip)

پس از دانلود این برنامه که شامل فایل های ذیل می باشد:



فایل MHD\_OS.EXE را اجرا نمایید. این برنامه را سالها پیش در دوران تحصیلم نوشتم تا با استفاده از آن بتوانم نیازهای ارتباطی خود را بر روی پروتکل TCP/IP آزمایش نمایم.



در اینجا برنامه از شما می پرسد که می خواهید پریز خود را در کدام وضعیت قرار دهید ؟ در وضعیت میزبان ( Host ) و یا مشتری ( Client ) برای انتخاب یکی از این دو وضعیت کافی است نام وضعیت را تایپ نمایید.

در اینصورت اگر شما تایپ نمایید HOST وضعیت پریز شما به حالت میزبان خواهد بود و اگر تایپ نمایید Client وضعیت میزبان شما به حالت مشتری خواهد بود و اگر در ای « مرحله تنها Enter را بفشارید وضعیت میزبان شما به صورت پیش فرض به حالت HOST می رود.

در این مرحله شما تایپ نمایید CLIENT تا به وضعیت مشتری وارد شویم.

در این هنگام - این پنجره را Minimize نمایید و سپس فایل MHD\_OS.EXE را مجدداً اجرا نمایید. پس از اجرای مجدد آن شما پنجره ای مشابه پنجره ی اول مشاهده خواهید کرد. در این مرحله



شود.

آنچه اتفاق افتاده است - شبیه سازی دو کامپیوتر بر روی شبکه در داخل کامپیوتر شما می باشد که یکی مشتری است و دیگری میزبان.

حال بر روی پنجره ی اول که حالت Client بر روی آن مشخص است وارد نمایید Info و سپس کلید Enter را بفشارید :

```

In the name of God - MHD internet OS v.1
File
info
Client TCP/IP system settings:
Host address: 0.0.0.0
Host name:
In buffer size: 2048
Local host name: mhd
Local port: 0
Out buffer size: 2048
Port: 0
Win socket info: 16-bit Windows Sockets
  
```

یک کامپیوتر مشتری جهت اتصال به یک کامپیوتر میزبان باید اطلاعات بالا را به درستی داشته باشد. منظور از Host address همان آدرس IP میزبان است و منظور از Host name همان نام میزبان است که اگر در DNS ثبت شده باشد به آدرس IP تبدیل می شود ( در بخش DNS توضیح داده شده است ).

منظور از in buffer size مقدار حجم اطلاعات ورودی به بایت در هر بار دریافت است و منظور از Local host name نام کامپیوتر شما (

پیش فرض بر روی کامپیوتر مشتری است و منظور از Out buffer size مقدار حجم اطلاعات ارسالی به بایت در هر بار ارسال است و منظور از Port کانال ارتباطی است که می توانید با آن به میزبان متصل شوید.

در واقع اساس برنامه نویسی پرز آن است که بتواند ارتباطهای متفاوتی را بر روی کانالهای متفاوت ایجاد نماید. در واقع منظور از کانال ارتباطی چیزی جز این نمی باشد که اگر شما - شماره ی Port را برابر 1 قرار دهید و به کامپیوتر ی در حالت میزبان با شماره ی آی پی 127.0.0.1 ارسال نمایید کلمه ی Hello را - در اینصورت TCP/IP موجود بر روی آن کامپیوتر اطلاعات را به برنامه ای انتقال می دهد که بر روی پورت شماره ی 1 در حالت انتظار باشد. اگر برنامه ای در حالت انتظار بر روی پورت 2 باشد در اینصورت کلمه ی Hello را دریافت نخواهد کرد.

نکته ی مهم در زمینه ی برنامه نویسی پرز در حالت میزبان آن است که حداکثر یک برنامه می تواند در حالت انتظار بر روی یک پورت باشد. برای مثال در زمینه ی میزبانی وب - که پورت استاندارد آن 80 معرفی شده است حد اکثر یک برنامه میزبان بر روی یک کامپیوتر میزبان می تواند بر روی پورت 80 در حالت انتظار باشد.

اما نقطه ی به عکس آن این است که در حالت مشتری پیش از یک برنامه می تواند اقدام به برقراری تماس با میزبان نماید.

در برنامه ی IOS که آنرا دانلود و اجرا نموده اید ( در یکی از دو پنجره ی Host و یا Client ) وارد نمایید port list و سپس Enter را بفشارید. در اینصورت برنامه فهرست شماره ی پروتھای استاندارد را به شما نشان خواهد داد.

در این بخش آنها را نیز آورده ام :

```
7 ' echo
9 ' discard
```

13 ' daytime  
15 ' netstat  
17 ' qotd  
19 ' chargen  
20 ' ftp-data  
**21 ' ftp**  
23 ' telnet  
25 ' smtp  
37 ' time  
42 ' name  
43 ' whois  
**53 ' domain**  
57 ' mtp  
77 ' rje  
79 ' finger  
**80 ' web**  
87 ' link  
95 ' supdup  
100' mhd  
101' hostname  
102' iso-tsap  
103' dictatunary  
104' x400-snd  
105' csnet-ns  
109' pop  
110' pop3  
111' portmap  
113' auth  
115' sftp  
117' path  
119' nntp  
139' nbssession

158' tcprepo  
170' print-srv  
175' vmnet  
400' vmnet0  
512' exec  
513' login  
514' shell  
515' printer  
520' efs  
526' tempo  
530' courier  
531' conference

برای مثال پورت شماره ی 7 ( Echo ) به این معنی است که اگر شما هر نوع اطلاعاتی به میزبانی که بر روی پورت 7 بنا شده است ارسال کنید - طبق توافق بین المللی آن میزبان باید همان چیزی که شما ارسال نمودید را به شما برگرداند.

و یا استاندارد پورت HTTP و یا WEB برابر 80 است. این بدان معنی است که کانال ارتباطی و یا پورت میزبانهای وب دنیا باید به صورت استاندارد 80 باشد و نرم افزارهای مرورگر اینترنت مانند Firefox و یا Opera و یا NetScape و یا Internet Explorer به صورت پیش فرض می بایستی از طریق این پورت و یا کانال ارتباط برقرار نمایند.

در نمایی ساده در مورد پورت می توان گفت که پروتکل TCP/IP در ابتدای اطلاعات شماره ی پورت را ارسال می نماید (به مفاهیم مطرح شده در بخش پروتکل رجوع نمایید).

محدودیتی در شماره ی پورت وجود ندارد. نرم افزارهای بازبینی Spyware و یا Trojan با استفاده از یک برنامه نویسی ساده ی پریر اقدام به بررسی شماره های پروت می نمایند - از 1 تا 1 میلیون. و

می نمایند که می دانند Trojan های معروف بر روی آنها فعالیت می کنند.

Trojan نوعی برنامه ی مخفی است که بر روی کامپیوتر میزبان و یا مشتری نصب می شود ( به صورت خودکار و یا توسط پستهای الکترونیکی آلوده ) و سپس یک شماره ی پورت را مانند 888 در حالت میزبان قرار می دهد و کامپیوتر آلوده را به یک میزبان ساده تبدیل می کند و سپس کامپیوترهای دیگر از خارج از شبکه می توانند در حالت مشتری با این شماره ی پورت ارتباط برقرار کرده و دستوراتی از پیش تعیین شده را برای آن ارسال نمایند.

برای مثال - اگر در برنامه ی Trojan تعریف شده باشد - هر گاه کد 123 دریافت شده - کامپیوتر را خاموش کن - در اینصورت شما می توانید از طریق یک برنامه مانند IOS که آنرا از MHDsoft.Com دانلود نمودید به پورت شماره ی برای مثال 78908 که تروجان بر روی آن فعال است ارتباط برقرار نموده و کد 123 را ارسال نمایید.

در این راستا برنامه های Firewall که طرفداران زیادی نیز پیدا کرده اند بر اساس این اصل ساده که بر روی یک شماره ی پورت تنها یک برنامه می تواند در حالت میزبانی قرار گیرد - خود هنگام ورود شما به سیستم عامل همه ی پورتها را از شماره ی 1 تا چند میلیون میزبانی می کند و اجازه نمی دهد که برنامه ی دیگری میزبانی پورت خاصی را بدست گیرد. اما به هر حال - به دلیل عدم وجود محدودیت در شماره ی پورت این امکان وجود دارد که شماره های بالاتر مورد استفاده قرار گیرند.

در کنار این موضوع شرکتها ی ISP برای تامین امنیت بیشتر کاربرانشان فقط اجازه ی تبادل اطلاعات بر روی چند پورت از پیش تعیین شده مانند 80 برای وب - 25 و 110 برای پست الکترونیک و 23 برای FTP را می دهند.

میان این روشها باید کار آمد ترین ان برای میزبانهایمانند میزبان وب انتخاب شوند.

در برنامه نویسی پریز جهت برقراری تماس با میزبان در حالت مشتری باید به ترتیب مراحل ذیل انجام شود :

- شماره ی پورت ارتباطی مشخص شود
- آدرس آی پی و یا نام میزبان مشخص شود
- دستور برقراری تماس داده شود

و همچنین جهت ایجاد میزبان نیز باید مراحل ذیل انجام شود :

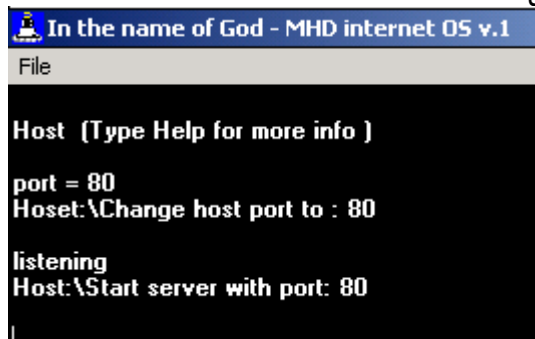
- شماره ی پورت مشخص شود
- دستور انتظار داده شود

برای آزمایش عملی این موضوع به برنامه ی IOS برگردید و پنجره ی HOST را Maximize نمایید و در آن وارد نمایید :

Port = 80

Listening

مانند تصویر ذیل :



مي توانستيد بجاي پورت 80 هر شماره ي ديکري را نيز انتخاب نماييد ولي از آنجا که در ادامه مثالهايي با یک مرورگر مي خواهيم داشته باشيم بنا بر اين از پورت 80 استفاده نموديم.

حال در پنجره ي Client اطلاعات ذيل را جهت بر قراري تماس وارد نماييد :

Port = 80  
Ip = 127.0.0.1  
Connect

مانند تصوير ذيل:

```
port = 80
Setting port 80 Complete, please type info to see

ip = 127.0.0.1
Set host address (IP) to : 127.0.0.1 For client system

connect
Please wait until connect to server and get ready message

Connect complete ( Client system )

Client : your text has been ready send to server
```

در انتهاي صفحه با مشاهده ي your text has been... متوجه مي شويم که ارتباط بر قرار شده است. پنجره ي Host را باز نماييد - مشاهده مي کنيد که نوشته شده است :

```
Host:\User connected by No. 1

Host:\Ready send your message to user No.1
```

پروتکل لایه ی بالاتر TCP/IP که همین نرم افزار IOS می باشد به آن کد 1 داده است و سپس اعلام شده است که کاربر شماره ی 1 آماده ی دریافت اطلاعات می باشد.

حال شما در پنجره ی Client تایپ نمایید : HELLO و سپس Enter را بفشارید. حال پنجره ی Host را باز نمایید – در آن نوشته شده است:

```
Host:\User No. 1 Send : HELLO
```

حال شما در HOST بنویسید WELCOME و سپس Enter را بفشارید و پنجره ی Client را باز نمایید – مشاهده خواهید کرد که در آن نوشته شده است :

```
Client:\WELCOME
```

در واقع شما یک ارتباط کامپیوتری را برقرار نمودید. آدرس آی پی میزبان در اینجا 127.0.0.1 بوده است که این آدرس به صورت پیشفرض در کامپیوترهایی که به شبکه متصل نمی باشند موجود است.

شما در این برنامه در محیط Client می توانید با تایپ عبارت ذیل آدرس آی پی یک میزبان مانند Yahoo را در صورت متصل بودن کامپیوترتان به اینترنت بدست آورید :

IP [www.yahoo.com](http://www.yahoo.com)

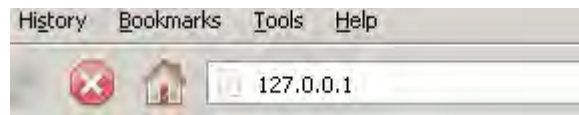
همچنین شما می توانید به جای بازکردن دو پنجره – تنها یک بار برنامه ی IOS را اجرا نمایید. برای حضور در محیط Client تنها تایپ نمایید Client و برای حضور در محیط Host تنها تایپ نمایید Host

حال برای انجام یک آزمایش مهیج آماده می شویم. پنجره ی Client را با تایپ دستور Exit ببندید و فقط پنجره ی Host باز باشد.



Address bar وارد نمایید : 127.0.0.1

مانند تصویر ذیل :



سپس به نرم افزار IOS نگاهی بیاندازید :

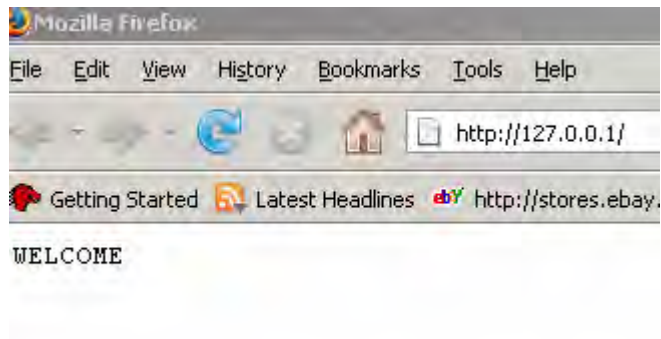


اطلاعات ارسال شده توسط Firefox همان Header است که در بخشهای قبلی بررسی شده اند.

حال شما در برنامه ی IOS تایپ نمایید Welcome و سپس Enter را بفشارید و سپس تایپ نمایید 1 Disconnect و سپس Enter را بفشارید مانند تصویر صفحه ی بعد :

```
disconnect 1
Host:\Send Disconnect message to Guest , wait for ready message....
```

حال به مرورگر خود نگاهی بیاندازید :



هیجان انگیز است – شما توانسته اید کلمه ی Welcome را به مرورگر خود ارسال نمایید.

به طورا کلی سیستم مرورگر وب به این صورت است که اطلاعات دریافتی را وقتی نشان بدهد که میزبان ارتباط را قطع کند که این مهم با دستور شما پس از ارسال اطلاعات به واسطه ی Disconnect 1 انجام شد که بدین معنی بود که ارتباط کاربر شماره ی 1 را قطع کن.

حال در مرور گر خود تایپ نمایید : [Http://127.0.0.1/welcome.htm](http://127.0.0.1/welcome.htm) و سپس Enter را بفشارید و نگاهی به سطر اول اطلاعات ارسالی به نرم افزار IOS نمایید :

```
Host:\User No. 1 Send : GET /welcome.htm HTTP/1.1
Host: 127.0.0.1
```

همانطور که در بخشهای گذشته نیز شرح داده شد به وضوح گزینه ی Host و GET مشخص است.

را از Host با نام 127.0.0.1 درخواست نموده اید. اگر نام کامپیوتر شما در تنظیماتش MHD باشد و در مرورگر خود تایپ نمایید :

[Http://mhd/welcome.htm](http://mhd/welcome.htm)

هدر ارسال به شکل ذیل خواهد بود :

```
GET /Welcome.htm HTTP/1.1
HOST: mhd
```

حال شما در IOS متن ذیل را تایپ نمایید :

```
WELCOME TO THE <font size=3> WEB <font size=5>
TECHNOLOGY
```

و سپس تایپ نمایید :

Disconnect 1

مانند شکل ذیل:

```
Host:\Ready send your message to user No.1
WELCOME TO THE <font size=3> WEB <font size=5> TECHNOLOGY
Host (Type Help for more info )

disconnect 1
Host:\Send Disconnect message to Guest , wait for ready message....
```

در اینصورت در مرورگر خود متن را به شکل ذیل خواهید دید :

WELCOME TO THE WEB TECHNOLOGY

با این کار درواقع شما محتویات یک فایل HTM را انتقال داده اید. این به مانند این بود که سیستم به صورت خودکار ( بجای عامل انسانی ) فایل را باز نماید و محتویاتش را ارسال نماید.

منظور از مفهوم پردازش برنامه طریقه ی اجرای برنامه در سیستم عامل است. علت اصلی مطرح شدن این بحث آن است که اولاً خواننده ی محترم با مفهوم پردازش آشنا شود و سپس بتواند توانایی های برنامه ی میزبان خود را افزایش دهد.

در بخش پردازش برنامه دانستن مطالب ذیل می تواند به شما در طراحی یک برنامه ی مقاوم و ایستا کمک کند.

در دوران اولیه ی کامپیوترها از نوعی پردازش به نام پردازش مرحله به مرحله و یا پردازش دسته ای استفاده می شده است. یعنی اول کار برنامه ی الف اجرا می شود و پس از اتمام آن کار برنامه ی ب.

پس از آن برای افزایش بهره وری پردازش سیستمهای کامپیوتری از بحث پردازش چند نخ استفاده شده است. نخ یک واحد اجرای برنامه است. اگر دو برنامه ی آماده ی اجرا داشته باشیم به نامهای الف و ب - و هر برنامه چند خط داشته باشد ابتدا خط اول برنامه ی الف و سپس خط اول برنامه ی ب و مجدداً خط دوم برنامه ی الف و سپس خط دوم برنامه ی ب اجرا می شود. با استفاده از این روش شما صرفه جویی زیادی در وقت نموده اید.

روش دیگر TDM و یا تقسیم زمانی است. یعنی منابع CPU و یا پردازشگر را در هر واحد زمانی به یکی از برنامه ها اختصاص می دهند.

برای مثال اگر CPU یک خط برنامه را در 0.1 ثانیه اجرا می کند - بنا بر این هر 0.1 ثانیه به اجرای بخشی از یک برنامه اختصاص می یابد و بدین ترتیب در یک ثانیه چندین برنامه اجرا می شوند.

از آنجا که در بحث میزبان وب کار اصلی که نرم افزار شما باید انجام دهد خواندن فایل و یا فایلها و ارسال محتویات آن به مشتری است بنا بر این دانستن این بخش خالی از لطف نمی باشد.

در بحث دسترسی به فایل انواع ذیل معرفی می شوند :

- خواندن
- نوشتن
- اشتراک

و انواع فایل :

- متنی
- باینری

در این روش شما می توانید مفاهیمی که در مورد پردازش برنامه و دسترسی به فایل مطرح شدند را عیناً در برنامه نویسی شبکه اجرا نمایید.

برای مثال - نرم افزار میزبان وب شما باید قادر باشد تعداد زیادی درخواست را به طور مستمر جواب دهد. شما به راحتی می توانید با استفاده از مفاهیم پردازش که در دو بخش پیش معرفی شدند این قسمت را برنامه نویسی کنید.

برای مثال - اگر سه درخواست به صورت همزمان سه فایل مجزا را درخواست نمایند شما می توانید برنامه را طوری تعیین کنید که ابتدا خط اول فایل الف را بخواند و ارسال کند و سپس خط اول فایل ب و سپس خط اول فایل پ و حتی می توانید از مفهوم پردازش همزمان نیز استفاده نمایید.

## Pre-trusted-Connection

یکی از مواردی که باعث کاهش سرعت میزبانهای وب می شود - نوع پروتکل آن است. در واقع سیستم اکثر میزبانهای وب به اینگونه است که پهنای باند را به تعداد کاربرانی که بدان متصل شده اند تقسیم می کنند. ( روش ارسال همزمان )

اکثر میزبانهای وب پس از ارسال اطلاعات از طریق پروتکل TCP/IP حجم داده ی ارسال شده را مورد بررسی قرار می دهند و اگر در این ارسال با خطا مواجه شوند این ارسال را از سر می گیرند.

تمامی این موارد باعث کاهش سرعت میزبان وب می شود. از اینرو می توان از یک روش ساده تر به نام Pre-Trusted-Connection استفاده نمود. برای شرح این روش که مجموعه ای از روشهای مختلف است باید ابتدا مسئله ای را توضیح دهیم.

علت اصلی تقسیم پهنای باند به تعداد کاربران در میزبانهای وبی مانند Microsoft Internet Information System و یا MS-IIS آن است که بتوانند بیشترین بهره وری را در استفاده از پهنای باند بدست آورند. اما استفاده از این نوع از نرم افزارهای هر چقدر هم که بخاطر امکاناتشان هیجان انگیز باشد اما با مشکلات ذیل مواجه خواهد شد :

- محدودیت در جوابگویی به درخواستهای همزمان
- ایجاد اختلال در میزبان
- ایجاد راه حلی برای استفاده توسط خرابکاران شبکه

شاید مورد آخر یکی از مهمترین موردهایی باشد که در کشورهایمانند ایران تا به امروز ( تاریخ انتشار کتاب ) کمتر به آن توجه شده باشد.

همانطور که در بخش برنامه نویسی پریش شرح داده شد - تعداد نرم افزارهایی که بر روی میزبان بر روی یک پورت خاص می توانند در حال

یک نرم افزار مي تواند بر روي ميزبان در حالت انتظار باشد. اما همانطور که در کنار این مطلب در بخش مربوط به برنامه نویسی پریز شرح داده شد - تعداد نا محدودی مشتری بر روی یک کامپیوتر می توانند وجود داشته باشند.

حال فرض کنیم یک نفر برنامه ای مانند IOS که شما در بخش برنامه نویسی پریز از سایت MHDsoft دانلود نمودید را طراحی نماید و بسازد که بتواند با یک دستور بیش از یک هزار ارتباط مشتری Client Connection بسازد و به یک میزبان خاص مثلاً با آدرس IP برابر با 127.0.0.1 ارتباط برقرار نماید.

از آنجا که پهنای باند میزبان وب مورد نظر مطمئناً روی عددی خاص محدود می شود و از آنجا که از روش تقسیم پهنای باند استفاده می کند از اینرو اصطلاحاً "میزبان به خواب فرو می رود و از این پس بازدید کنندگان وب سایت با پیغام :

The web server is too busy, please try later

مواجه خواهند شد.

برای مواجه شدن با پیغام بالا - نیاز به یک برنامه ی خاص نمی باشد. فرض کنیم یک میزبان در حدود 10 کیلوبایت در ثانیه پهنای باند دارد و از نرم افزار MS-IIS و یا Baby web server استفاده می کند.

این سایت قرار است نتیجه ی یک امتحان خاص را در اختیار یک میلیون نفر قرار دهد. اعلام نتایج از ساعت 7 صبح آغاز می گردد. طبق یک حدس ساده حد اقل 50% داوطلبین می خواهند نتایج امتحان خود را در همان ساعات اولیه مشاهده نمایند.

در اینصورت حد اقل 500 هزار نفر سعی می کنند در چند ساعت اولیه به میزبان دسترسی داشته باشند. هر کاربر باید حداقل 30 کیلوبایت اطلاعات را دانلود نماید.



کاربران با خطوط Dial-up به میزبان متصل می شوند که حد اکثر پهنای باند آنها برابر با 4 کیلوبایت در ثانیه است.

درواقع در بهترین شرایط 7.5 ثانیه زمان نیاز است که بتوان اطلاعات را دانلود نمود.

اگر تنها 2 کاربر به این میزبان متصل شوند هر یک با نهایت سرعت خطوط خود می توانند اطلاعات را دانلود نمایند. اگر این تعداد به 5 کاربر برسد - زمان دانلود اطلاعات هر یک از کاربران 15 ثانیه می شود و اگر این تعداد به 10 کاربر برسد - زمان دانلود اطلاعات به 30 ثانیه می رسد و اگر این عدد به 100 کاربر برسد زمان دانلود اطلاعات برای هر کاربر به 5 دقیقه خواهد رسید.

اکثر شرکتها بخاطر وجود چنین محدودیتهایی معمولا "پهنای باند خود را افزایش می دهند.

شرکت Microsoft در الگوریتم نرم افزاری جدید خود در رابطه با نرم افزار IIS خود - روشی جدید را معرفی نموده است و آن Compress نمودن اطلاعات می باشد.

در این روش انحصاری - اطلاعات میزبان به صورت فایل ZIP در آمده و سپس پس از دانلود توسط فقط نرم افزار Microsoft Internet Explorer این فایل فشرده شده بازگشایی می شود. به گفته ی شرکت مایکروسافت این روش 30% در کاهش هزینه های پهنای باند کمک خواهد کرد.

در کنار این موضوع - پهنای باند مورد نیاز یک میزبان وب با استفاده از نرم افزارهای مانند IIS باید از نوع Dedicated باشد که این نوع از پهنای باند معمولا "هزینه ی بسیار بالایی در بر دارد.

در این صورت می توان از راهکار Pre-trusted-connection استفاده نمود. خود من از این راهکار در طراحی و ساخت نرم افزار MHD WEB

را بر عهده دارد استفاده نمودم. [www.Parsian-invest.com](http://www.Parsian-invest.com)

به طور کلی این راهکار شامل روشی است که بتوان با استفاده از آن پهنای باند را مدیریت نمود و بیشترین سرعت را در انتقال داده بدست آورد.

در این روش بجای تقسیم پهنای باند به تعداد کاربران - با استفاده از روش TDM که در بخش پردازش برنامه توضیح داده شد - در هر واحد زمانی پهنای باند را به یک کاربر اختصاص می دهیم و سپس با استفاده از روش پردازش چند نخه ( Multi threading ) در ارسال محتویات فایل استفاده نمود.

در این روش - نرم افزار ابتدا به اندازی پهنای باند اطلاعات را به کاربر در یک واحد زمانی ارسال می نماید. میزان خطای ارسال - یعنی مقدار داده ای که ارسال نشده است منهای پهنای باند برابر می شود با پهنای باند کاربر. از این پس نرم افزار به این کاربر مقدار اطلاعات برابر با پهنای باند وی را در واحدهای زمانی ارسال می کند.

در کنار این موضوع - اگر کاربری بیش از 10 درخواست همزمان به میزبان ارسال کرد - درخواستهای دیگر او در صف FILO قرار خواهند گرفت. یعنی فقط به 10 درخواست یک کاربر ( شناسایی با آدرس IP او ) جواب داده می شود و به محض اینکه یکی از 10 درخواست کاربر به اتمام رسید به درخواست یازدهم ( در صورت وجود ) رسیدگی می شود.

با استفاده از این روش نیمه هوشمند علاوه بر اینکه میزبان خود را از حملات خرابکاران شبکه ( مانند حمله هایی که به MS-IIS می شود و عموماً "DoS نام دارند و شرح آن گذشت ) بیمه می نماید - بدون نیاز به یک خط Dedicated و با پهنای باند معمولی در حدود 3 کیلوبایت در ثانیه ( در حدود Dial-up ) می توانید تا 50 درخواست به صورت همزمان را با سرعتی برابر با حداکثر سرعت پهنای باند خود جوابگویی نمایید.

حاضر از این نرم افزار استفاده می نماید و اگر پهناي باندي در حدود 4 کیلوبایت داشته باشد - و 50 کاربر درخواست یک صفحه ي 4 کیلوبایتي نمایند - ( در یک زمان ) هر 50 کاربر به صورت همزمان اطلاعات را در بازه ي زمانی بین 1 الي 2 ثانیه دریافت خواهند کرد ( پهناي باند کاربران نیز برابر با میزبان است ) .

در میزبانهاي دیگر مانند MS-IIS بر روی چنین پهناي باندي شاید در نهایت 3 و یا 5 کاربر نتوانند به صورت همزمان اطلاعات دریافت نمایند.



در کانال تلگرام کارنیل هر روز انگیزه خود را شارژ کنید 😊

<https://telegram.me/karnil>

