

PART -1

HTTP:THE WEB's FOUNDATION

Bu bolimde HTTP protokolune giris yapılacaktır.Onumuzdeki dort bolimde HTTP teknolojisinin ana kisimlari ve Webin bulunmasi gibi kisimler islenecektir.

Bolum 1,Http'ye genel bakis.

Bolum 2,URL'ler ve Kaynaklar- Url formatlarinin detaylari ve cesitleri,ayni zamanda URN'nin evrimi.

Bolum 3,Http Mesajlari

Bolum 4,Baglanti Yonetimi- Genellikle yanlis anlasilan ve az bilinen kurallar tartisilacak ve TCP baglantilariyla HTTP'nin davranislari incelencektir.

BOLUM 1

HTTP'ye genel bakis

Dunyadaki web tarayicilari,serverlar,web uygulamalari hepsi birbiriyle HTTP yoluyla konusur.Hypertext Transfer Protocol.HTTP modern internetin ana dilidir.

Bu bolumde kısa ve oz olarak HTTP anlatilacaktir.Web uygulamalarinin HTTP yi nasil kullandigini anlayacak,HTTP'nin az cok nasil calistigi ile ilgili fikre sahip olacaksınız.Ozellikle su konular konusulacaktir.

1-Serverlar ve istemciler nasil haberlesir.

2-Kaynaklar nereden ulastirilir.

3-Web islemleri nasil calisir.

4-HTTP yi kullanan mesaj formatlari

5-Belli basil TCP baglantilarinin cozumlenmesi

6-HTTP'nin cesitli varyasyonlari

7-Bir suru HTTP tabanlı mimariler internette gezinmektedir.

HTTP hakkında ogrenilecek bir suru sey var o halde HTTP uzerine tura baslayalim.

HTTP:Internetin Multimedya Kargosu

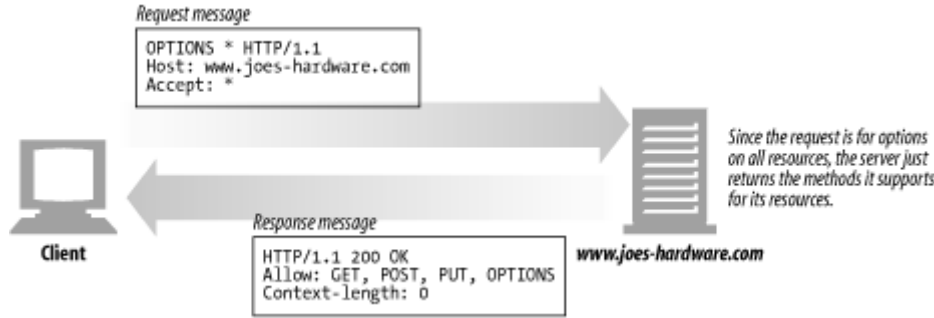
Milyarlarca jpeg resimler,html sayfaları,metin dosyaları,mpeg videoları,wav ses dosyaları,java uygulamaları ve daha fazlası her gun internette bulunmaya basliyor.HTTP toplu bilgileri hizlica,kolayca ve guvenli bir sekilde web serverlarından dunyadaki kullanicilarin hepsinin browserlarına tasir.

Cunku HTTP guvenilir veri iletim protokolleri kullanir.HTTP ulasim sirasinda verinin zarar gormeyeceginin veya kurcalanmayacaginin garantisini verir.Baska bir gezegenden gelicek olsa bile:D.Bu kullanicilar icin iyi bir sey.Cunku kullanicilar bilgiye butunluktan ksku etmeden ulasabilir.Guvenli tasima ayrica Internet developerlari icinde iyi bir sey.Cunku HTTP iletisimi hakkında veya bilginin kurcalanmasi,bozulmasi ile ilgili endiselenmenize gerek yok.Sadece uygulamanizin belirli detaylarına konsantre olmanız yetecektir.Internetin kusurlari ve zaafllari icin endiselenilmesine gerek yoktur.

Simdi HTTP'nin Web trafiginin nasil tasidigina daha yakindan bakalim.

Web Istemicileri ve Serverlari

Web icerikleri web serverlarında yasar. Web server HTTP protokolu ile konusur. Yani onlar icin HTTP serverlari diyebiliriz. Bu HTTP serverlari Internet datalarını depolar ve HTTP ile istek gelindiginde bu depoladigi datayı istemciye saglar. Istemci HTTP istegini serverlara yollar ve server istenen datayı HTTP cevabini dondurur. Altteki Sekilde Gordugunuz gibi



Bu haberlesme beraber World Wide Web'in yani Internetin basic tabanini olusturur diyebiliriz. Siz muhtemelen (Yani kesin cok buyuk cogunlukla) HTTP'yi her gun kullaniyorsunuz. En muhtemelen kullanım yeri web tarayicilaridir (browserlar googleChrome, Mozilla, Safari gibi). Web tarayicilari HTTP istegi yaparak objeleri kendi ekranizda gormenizi saglar. Bir sayfayi arattiginiz zaman mesela `"http://www.egitimsistemindekoyayim.com/index.html"` web tarayiciniz `www.egitimsistemindekoyayim.com` 'a HTTP istegi yollar. Yukardaki resimde oldugu gibi. Server istenen objeyi yani `index.html`'i bulmaya calisir ve basarili olursa. Objeyi Istemciye HTTP Response mesajıyla beraber yollar. Objenin tipi, boyutu ve diger bilgilerle birlikte.

Kaynaklar

Web serverlari web kaynaklarinin sahibidir. Web kaynaklari ise web iceriklerine sahiptir. En basit tabiriyle web kaynaklari web serverlarindaki statik dosyalardir. Bu dosyalar herhangi bir seyi kapsayabilir. Metin dosyasi olabilir. Html dosyasi olabilir, Microsoft Word dosyasi olabilir, Adobe Acrobat dosyasi olabilir, Jpeg resim dosyasi olabilir, Avi video dosyasi olabilir, yada dusunebildiginiz herhangi bir format.

Ancak, kaynaklar statik dosya olmak zorunda degildir. Kaynaklar ayni zamanda talep edilen icerikleri ureten bir yazilim programi olabilir. Bu dinamik icerik kaynaklari kimliginize bagli olarak yada istediginiz bilgiye bagli olarak yada gunun saatine gore degisebilir. Size kameradan canli bir resim gosterebilir veya bir ilan yada bir arazi arayabilir veya internetten hediye almanizi saglayabilir

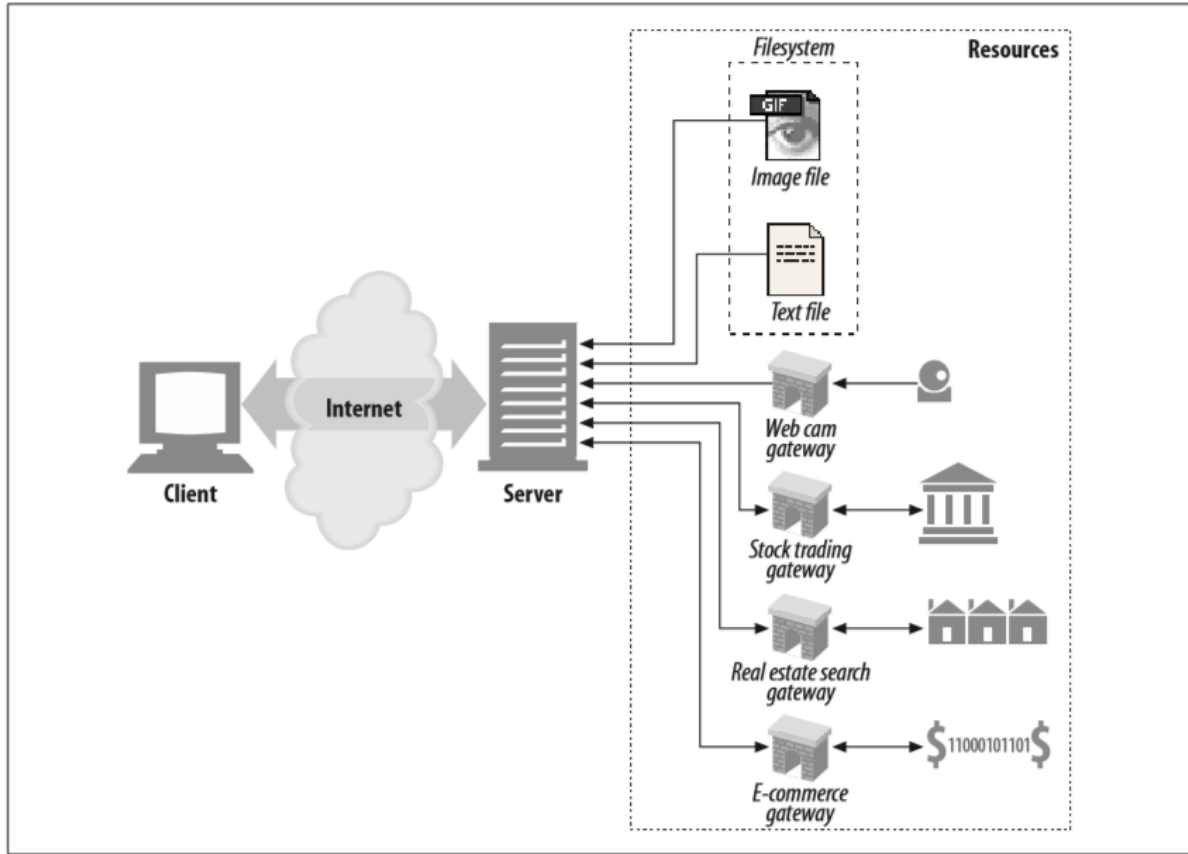


Figure 1-2. A web resource is anything that provides web content

Ozetle,kaynak herhangi bir iceriktir.Bir dosya sirketin tahmin raporunu kapsayabilir bu bir kaynaktır.Bir halka acik kutuphanenin raflarini tarayan web gateway bir kaynaktır.Bir Internet arama moturu kaynaktır.

Medya Tipleri

Internet binbir cesit data tipine evsahipligi yapıyor.HTTP dikkatlice transportasyonda kullanılacak olan yani karsi tarafa iletilecek olan objeyi etiketler.Buna MIME tipi denir.MIME(Multipurpose Internet Mail Extensions) orjinalde elektronik mail sistemleri arasindaki mesajlasmalarda karsilasilan sorunlari cozmek icin dizayn edilmistir.MIME email icin cok iyi calisir.HTTP'de bunu multimedia iceriklerini kolayca tarif etmek ve etiketlemek icin kendine MIME'I adapte etmistir.

Web serverlari tum HTTP objelerini MIME tipi ile iliskilemistir.Alttaki resimde gordugunuz gibi.Web tarayicisi serverdan bir objeyi serverdan geri getirdigi zaman.Eger browser objenin ustesinden gelebilirse.MIME tipi olarak gelir.Cogu browser(tarayici) yuzlerce populer obje tiplerini yonetebilir.Resmi gosterme,yapistirmek(kopyala ve yapistirin yapistiri baska yerlere cekmeyin)HTML dosyalarını bicimlendirme,bilgisayar spekarından ses dosyasi calma,veya özel formatları kontrol etmek için yuklenen plug-in yazılımını calistirma.

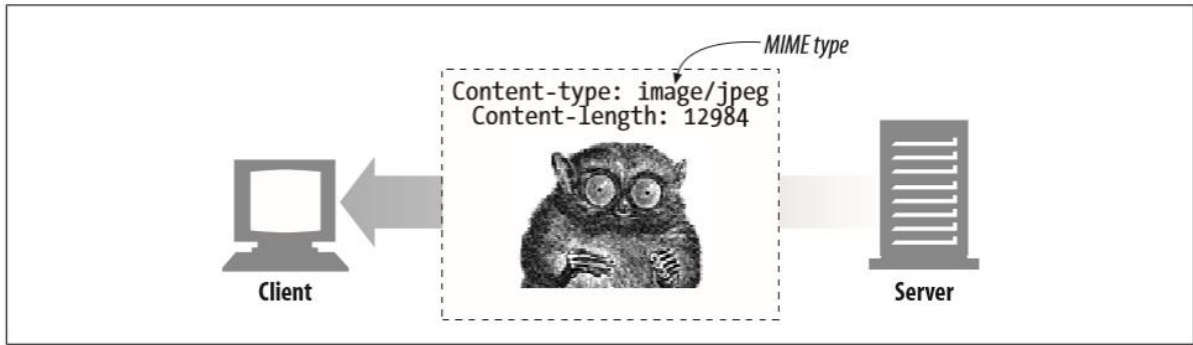


Figure 1-3. MIME types are sent back with the data content

MIME tipi metinsel bir etikettir, ana obje tiplerini ve belirli alt alanlari temsil eder, slash(/)isareti ile ayrilir.Ornegin:

1-Bir HTML formatindaki metini text/html olarak etiketlenicek.

2-ASCII formatindaki metini text/plain olarak etiketlenicek.

3-JPEG versiyonundaki bir image image/jpeg

4-Gif formati image/gif

5-Apple QuickTime video/quicktime.

6-Microsoft PowerPoint sunumlari application/vnd.ms-powerpoint.

Yuzlerce populer MIME tipi var ve cogu deneysel veya limitli tipler.Bir suru MIME tipini kitap sonunda bulabileceksiniz Umarim MIME tiplerini anlamissinizdir.

URI'ler

Her web server kaynagi bir isme sahip.Isime sahip olduklari istemciler hangi kaynaklarla ilgilendiklerini belirtebiliyorlar.Server kaynak isimlerine biz URI diyoruz(uniform resource identifier).URI Internetin posta kodu gibi bir sey.Dunyanin her tarafindaki bilgi kaynakarini,Essiz olarak belirler ve konumlandirir.

URI icin bir ornek.

<http://www.joes-hardware.com/specials/saw-blade.gif>

Asagidaki resimde URI nin saw-blade gif kaynagini Joe'nun servarinda nasil belirledigini goruceksiniz.URI sayesinde HTTP objeyi belirleyip getirdi.URI iki cesit olarak bilinir ve soylenir.URL ve URN.Simdi iki kaynak belirliyeciyede goz gezdirecegiz.

URL'ler

URL(uniform resource locator)en cok kullanimli kaynak belirleyicidir.Herkes bunu bilir zaten.Kaynagin belirli yerlerini serverlarda tarif eder.URL tam olarak kaynagi nasil getirilcegini soyler ve kesin ve fix sekilde getirir.Asagidaki resimde URL kaynagin nerde oldugunu ve nasil ulasilacagini guzel bir sekilde soyler.

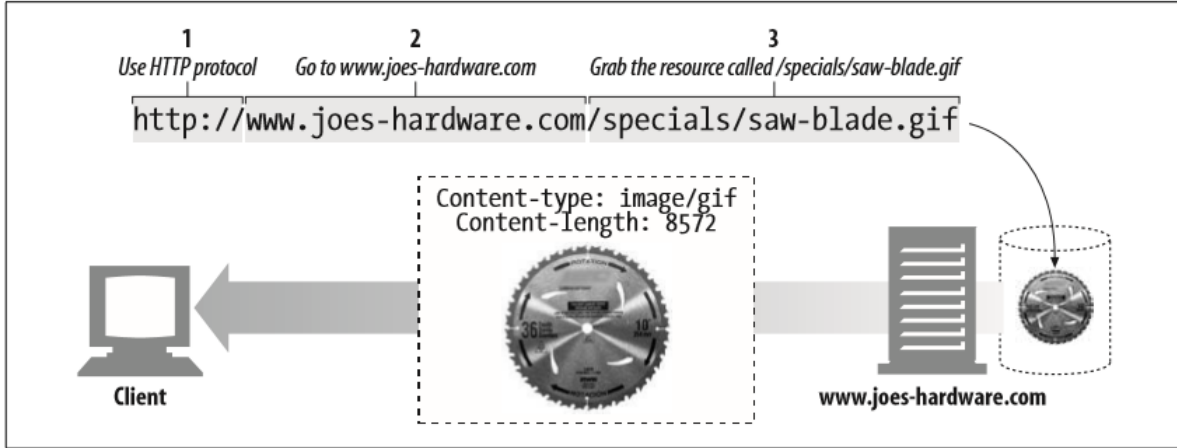


Figure 1-4. URLs specify protocol, server, and local resource

Table 1-1. Example URLs

URL	Description
<code>http://www.oreilly.com/index.html</code>	The home URL for O'Reilly & Associates, Inc.
<code>http://www.yahoo.com/images/logo.gif</code>	The URL for the Yahoo! web site's logo
<code>http://www.joes-hardware.com/inventory-check.cgi?item=12731</code>	The URL for a program that checks if inventory item #12731 is in stock
<code>ftp://joe:tools4u@ftp.joes-hardware.com/locking-pliers.gif</code>	The URL for the <i>locking-pliers.gif</i> image file, using password-protected FTP as the access protocol

Most URLs follow a standardized format of three main parts:

- The first part of the URL is called the *scheme*, and it describes the protocol used to access the resource. This is usually the HTTP protocol (`http://`).
- The second part gives the server Internet address (e.g., `www.joes-hardware.com`).
- The rest names a resource on the web server (e.g., `/specials/saw-blade.gif`).

Today, almost every URI is a URL.

Yukarda dedigi gibi bugun nerdeyse her URI bir URL'dir.

URN'ler

Ikinci olarak URN(uniform resource name).URN icerigin ozel yerleri icin essiz isimler saglar.Kaynagin bulunduđu yerden bagimsizdir.Bu konum-bagimsizlik olayi kaynaklara bir yerden bir yere tasinmasina izin verir.URN ayni zamanda ayni isimle birden cok ag erisim protokolunun kaynaklara erismesine izin verir.

Ornegin,simdiki URN "RFC 2141" nerede bulunduđunu iplemeden isimlendirecektir.

`urn:ietf:rfc:2141`

Urn hala deneysel ve adapte edilmedi.Efektif calisabilir ama uzerine hala calisilmasi gerek.Gelecek icin umut vaad edici.Herki bolumlerde URN'den gene bahsedecegiz ama bizim bu kitapta isimiz URL ile.

CALISILMASI

HTTP'nin kullanıcı tarafından nasıl kullanıldığına ve nasıl çalıştığına yakından bakalım.

HTTP çalışması HTTP istegi ve HTTP cevabini içerir.Bu iletisim bicimlendirilmis block seklindeki datalar ile gercekleşir.Asagida ornek resim var.

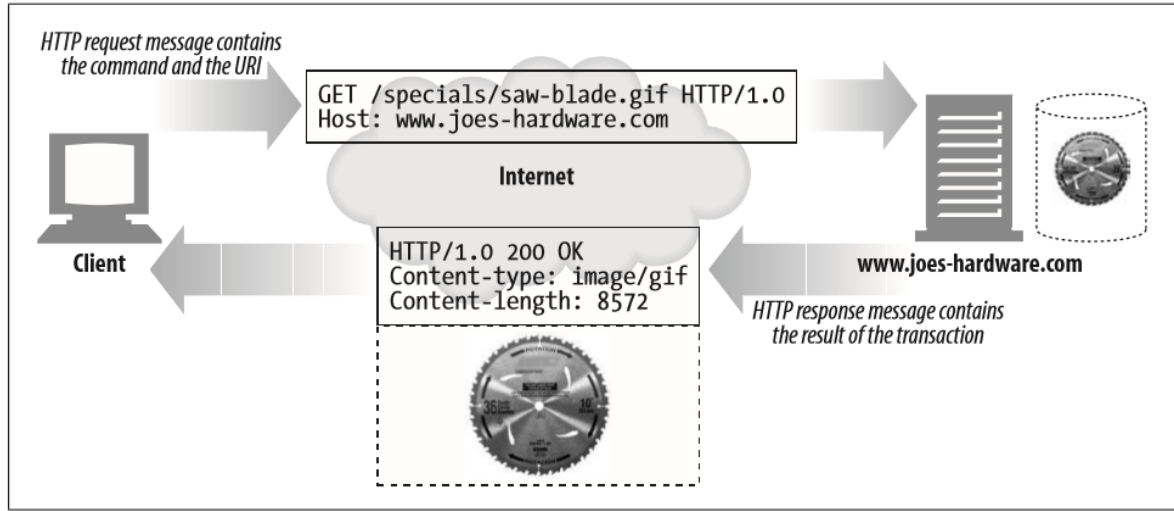


Figure 1-5. HTTP transactions consist of request and response messages

Methodlar

HTTP birkaç istek komutlarını destekler.Bunlara HTTP methodlari denir.Her HTTP istegi bir methoda sahiptir.Bu method servera hangi actionda bulunacağını söyler(sayfa getir,dosya sil gibi).En çok kullanılan 5 HTTP methodu.3. bölümde methodlara değineceğiz.

Table 1-2. Some common HTTP methods

HTTP method	Description
GET	Send named resource from the server to the client.
PUT	Store data from client into a named server resource.

Table 1-2. Some common HTTP methods (continued)

HTTP method	Description
DELETE	Delete the named resource from a server.
POST	Send client data into a server gateway application.
HEAD	Send just the HTTP headers from the response for the named resource.

STATUS KODLARI

Her HTTP cevabi bir status kod ile geri doner. Status kodlari 3-rakamli kodlardir. Eger istek basarili veya baska bir action olursa kullaniciya soyler. Asagida birkac bilinen status kodu var

Table 1-2. Some common HTTP methods (continued)

HTTP method	Description
DELETE	Delete the named resource from a server.
POST	Send client data into a server gateway application.
HEAD	Send just the HTTP headers from the response for the named resource.

We'll discuss HTTP methods in detail in Chapter 3.

HTTP ayni zamanda metin olarak kodun nedenide kodun yaninda yollar. Ama her zaman yollamak zorunda degildir. Asagida ornekleri var. Status kodlari bolum 3de sonra incelencektir

200 OK
200 Document attached
200 Success
200 All's cool, dude

HTTP status codes are explained in detail in Chapter 3.

Web Sayfaları Birden fazla Obje icerebilir.

Bir Uygulama siklikla birden fazla HTTP islemlerinde gorev yerine getirilirken sorun cikarabilir. Ornegin, bir web tarayicisi asamali olarak bir HTTP isleminde yuksek grafikli bir web sayfasini getirip ve gosterirken sikinti cikabilir. Tarayici bir islemde HTML iskelet sayfasini getirir. Ondan sonra HTTP islemlerinde fazladan sikinti cikaran seyleri teker teker getirir. Bu seyler farkli serverlardan gelebilir. Asagidaki resimde goreceksiniz. Boylece, web sayfasi dedigimiz sey genellikle birden fazla kaynagini koleksiyonudur. Yalnizca bir kaynak degildir.

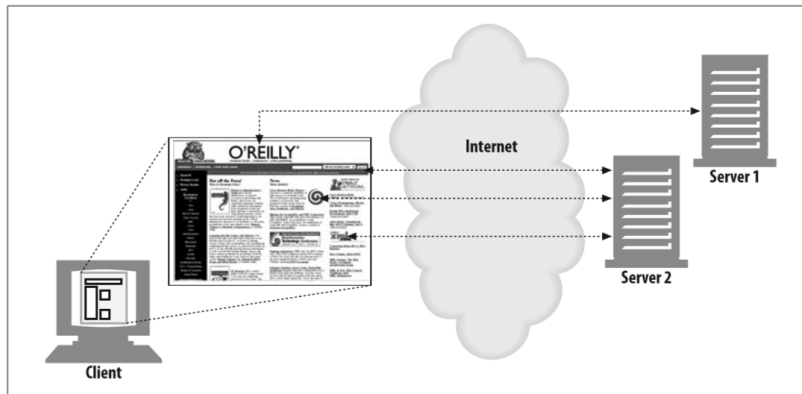


Figure 1-6. Composite web pages require separate HTTP transactions for each embedded resource

MESAJLAR

Simdi HTTP'nin istek ve cevap yapısına hizlica bakalim.HTTP mesajlarina 3.bolumde detaylariyla deginecegiz demekki 3.bolum onemli.

HTTP mesajlari basittir.Sirali olarak satirlardan olusan karakterlerdir.Cunku plain texttir yani acik metindir,binary falan degildir.Insanlarin okuyabilmesi ve yazabilmesi icin kolaydir.Asagidaki resim HTTP mesaj islemlerinin basit bir hali.

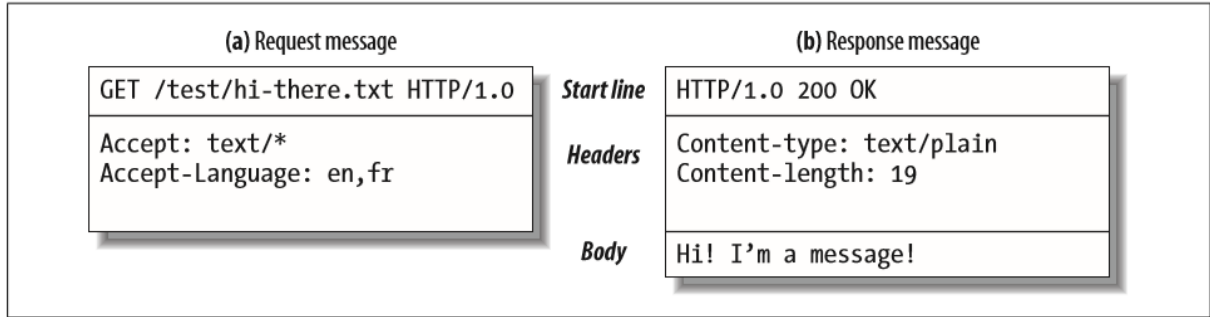


Figure 1-7. HTTP messages have a simple, line-oriented text structure

HTTP mesajlari istemciden server istek mesajlari ile yollanir.Serverdan istemciye giden mesajlarda Cevap Mesajlari olarak yollanir.Baska turlu bir HTTP mesaji yoktur.HTTP'nin istek ve cevap mesajlari birbirine benzerdir.

HTTP mesajlari 3 bolum icerir:

1-Baslangic Satiri(Start line)

Mesajin ilk satiri baslangic satiridir.Istek icin ne olacagini ve cevapta ne olacagini kapsar.

2-Baslik bolumu(Headers)

0 veya birden cok header baslangic satirindan sonra devam eder.Her header isim ve deger barindirir.Birbirinden ayristirilmasi icin : ile ayrilir.Headerler bosluk satiri ile biter.Header eklemek satir eklemek kadar kolaydir.

3-Ana bolum(Body)

Bosluk satirindan sonra opsiyonel olarak her turlu data tasiyabilir.Istek mesajindaki body kismi servera data tasir.Cevap mesajindaki body kismi istemciye geri doner.Baslangic satiri ve headlerin aksine body kismi text ve yapili olmasiyla beraber binary datasida tasiyabilir(resim,video,ses kayitlari,yazilim uygulamari).Tabi ki,body ayni zamanda text de icerir.

Basit Mesaj Ornegi

Bir kac bolumden sonra asagida bir resim olacak.HTTP mesajlarinin basit haliyle anlatimi oluacak.Browser yani tarayici kaynaklari istedi <http://www.joes-hardware.com/tools.html>.

Resimde tarayici HTTP istek mesajlari yollar.Istek mesajlari baslangic satirinda GET methoduna sahiptir,ve local kaynak /tools.html. dir.Istek mesajlari HTTP protokolunun versiyonun barindir.Istek mesajlari body kismi yoktur,cunku serverdan basit bir documenti GET etmek icin istek bodysine gerek yoktur.

Server geri HTTP cevabi gonderir.Cevap HTTP'nin versiyonunu,status kodunu,sebebi kismini ve cevap kisminda blok olarak header kismi,ve hepsinden sonra cevap mesajlari body kismi istek mesajlari istedigini getirir.Cevap mesajlari body uzunluk Content-length headerinda ve documentin MIME tipi is Content-Type headerinda bulunur.

Baglantilar

Simdi HTTP mesajlari nasil gorunur,bir mesaj bir yerden bir yere nasil tasinir bunun hakkında konusalim.TCP ile (Transmission Control Protocol)

TCP/IP

HTTP bir uygulama katmanli protokoludur.HTTP isin asil meselesi olan ag iletisimleri ile ilgilenmez,endiselenmez onun yerine detaylari ve isleri TCP/IP ye birakir.Guvenilir ve populer olan Internetin Altyapisi dedigimiz bir protocol.

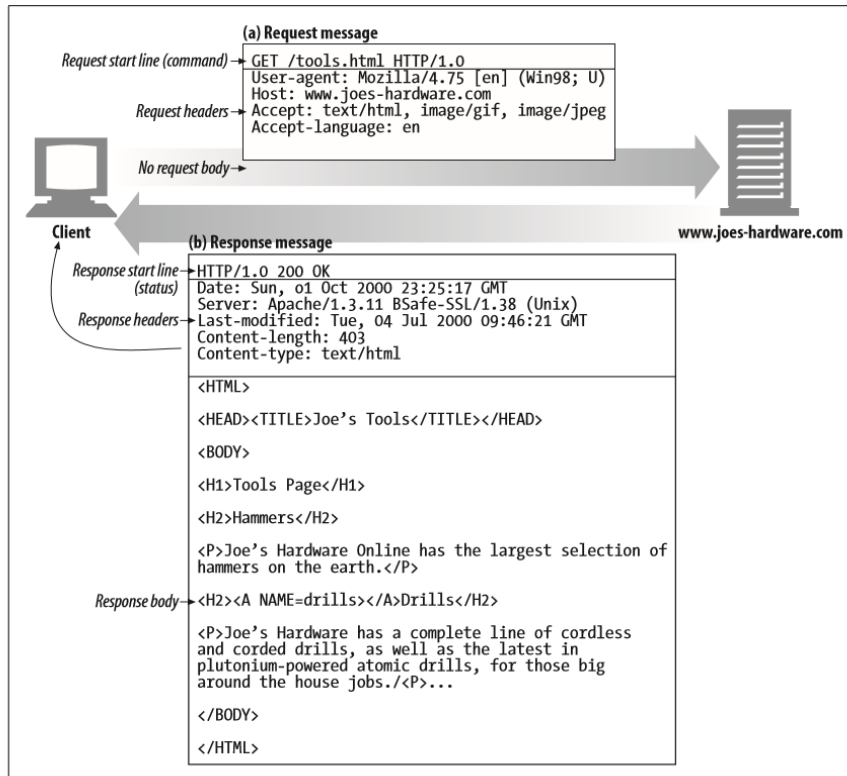


Figure 1-8. Example GET transaction for <http://www.joes-hardware.com/tools.html>

Tep sunlari saglar:

1-Hata kontrolu

2-Kesinlikle teslim

3-Ayrilmamis bozulmamis veri

Internetin kendisi TCP/IP tabanlidir,populer,katmali ve paket anahtarlama ag protokoludur.Bilgisayarlardan tutun icinde ag karti olan tum cihazlari birbiriyle haberlesitirir.TCP/IP kisisel aglari ve donanimlari acayipliklerini ve zayifliklari gizler.Her cihaziz birbiriyle guvenli bir sekilde haberlesmesine izin verir.

TCP baglantisi bir kere saglandimi,istemci ve sunucu arasinda gidip gelen mesajlar asla bozulmaz,zarar gormez veya emirlerin disina cikmaz.

Ag terimlerinde,HTTP prokolu TCP'nin uzerinde bir katmandir.HTTP datalari tasimak icin TCP'yi kullanir.Ayni Sekilde TCP de IP uzerinde bir katmandir.Resimde goruldugu gibi.

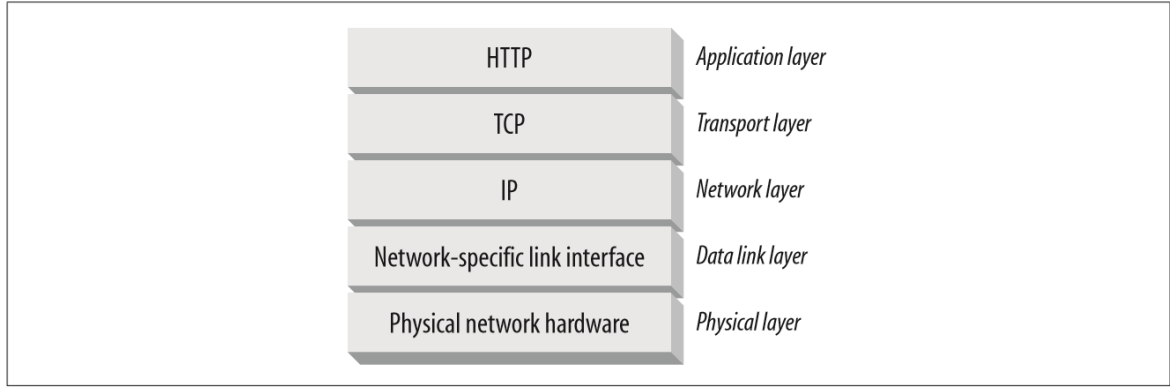


Figure 1-9. HTTP network protocol stack

Baglantilar,IP Adresleri ve Portlar

HTTP istemcisi server mesaj yollamadan once TCP/IP ile servera baglanmis olmasi lazim bu baglantida IP adresi ve port sayilariyla olur.

TCP baglantisi ayarlamak sirket ofisindeki birini aramak gibidir.Ilk olarak sirket numarasini cevrilir bu sizi dogru organizasyona getirir.Sonra Ozellikle ulasmak istediginiz kisiyi ararsiniz.

TCP de server bilgisayarinin IP adresine ve serverdaki calisan yazilim programiyla haberlesmek icin TCP port numarasina ihtiyac vardir.

Bunlari hepsi iyi guzel ama IP adresini ve port numarasini ilk basta nasil elde ediyoruz.Nasilmi bahsetmistik,URL ile!!Daha oncesinde URL icin kaynaklari adresi demistik.Dogal olarak Ip adresi ile kaynaga sahip olan server makinasina ulasmamizi saglamasi yeterlidir.Bir kac ornege goz gezdirelim.

http://207.200.83.20:80/index.html

http://www.netscape.com:80/index.html

http://www.netscape.com/index.html

İlk Url makinanın IP adresi “207.200.83.29” ve port numarası “80” e sahip.

İkinci URL numerik olarak IP adresine sahip değil. Metinsel bir domain ismine sahip, veya hostname diyebiliriz (“www.netscape.com”). Hostnameler IP adreslerin daha insancıl, canayakın ortakları diyebiliriz. Hostnameler kolayca DNS dedigimiz (Domain Name Service) ile IP adresine çevrilebilir. Şimdi tamamız DNS ve URL ile ilgili Bölüm 2 de konuşcaz.

Son URL de port numarası yok. URL de port numarası olmadığı zaman default olan 80 portu kullanılır.

IP adresi ve port numarasıyla istemci kolayca haberleşmede bulunabilir. (TCP/IP yani)

Aşağıdaki örnek HTTP uzak serverlarda bulunan basit HTML kaynaklarını nasıl kullandığını gösteriyor.

Adımlar:

- Tarayıcı URL’den server hostname’ini çıkarır.
- Tarayıcı server hostname’i IP adresine çevirir.
- Tarayıcı eğer port numarası varsa URL’den onu çıkarır.
- Tarayıcı server ile TCP bağlantısı kurar.
- Tarayıcı HTTP istek mesajı yollar.
- Server HTTP cevap mesajı yollar.
- Bağlantı kapanır ve tarayıcı documenti gösterir.

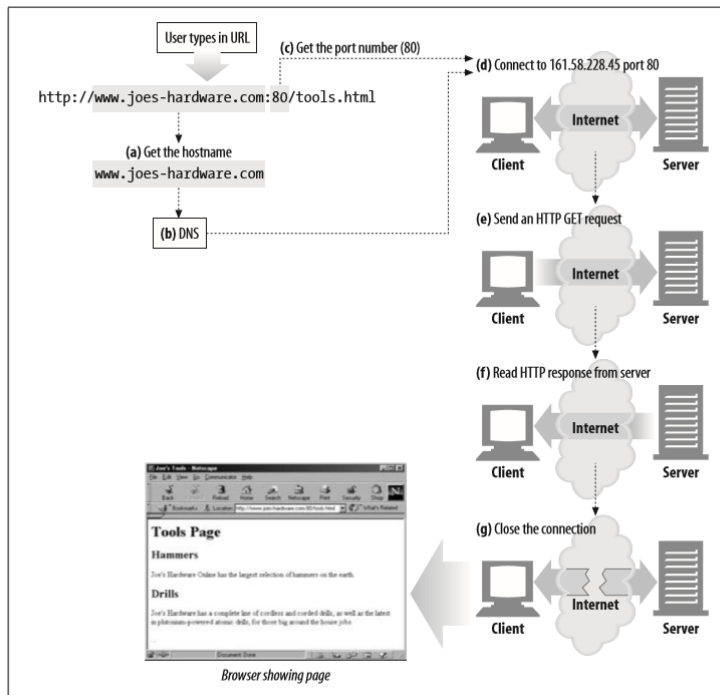


Figure 1-10. Basic browser connection process

TELNET İLE GERCEK ORNEK

HTTP TCP/IP kullandigindan dolayi ve metinsel tabanlı olduğu için binary formatının aksine web server ile konuşmak çok kolaydır.

Telnet'in olayı klavyenizi hedefin TCP portuna hedefin TCP port çıkışını sizin ekranıza bağlar.Telnet genelde uzaktan terminal oturumunda kullanılırdi,ama genelde herhangi bir TCP serverina bağlanabilir,HTTP serverlarına bile.

Telneti direk olarak web serverlarıyla konuşmakta kullanabilirsiniz.Telnet makinanın portunda TCP bağlantısı yapmanıza izin vererek porta direct olarak karakter girmenizi sağlar.Web server istemci olarak davranır ve TCP bağlantısında dönen dataların hepsi ekranınızda görünür.

Simdi Telnet ile gerçek bir web serverıyla etkileşimde bulunalım.Telneti documente bağlanmak için kullanacağız.URL <http://www.joes-hardware.com:80/tools.html>

Özetle neler olacak söyleyelim:

İlk olarak www.joes-hardware.com 'un Ip adresine bakmamız ve TCP bağlantısıyla port 80'e bağlanmamız gerek.Telnet bu ayak işlerini bizim için yapar.

Bir kere TCP bağlantısı açıldı,HTTP istegini yazmamız gerekecek.

İstek tamamlandığı zaman(Host ile başlayan kalın yazıdan sonra boşluk ile istegin tamamlandığı belirtilmiş) HTTP cevabının içinde istenen içerik gönderilerek bağlantı kapanacaktır.

<http://www.joes-hardware.com:80/tools.html> ile örnek gösterilecek.

Example 1-1. An HTTP transaction using telnet

```
% telnet www.joes-hardware.com 80
Trying 161.58.228.45...
Connected to joes-hardware.com.
Escape character is '^]'.
GET /tools.html HTTP/1.1
Host: www.joes-hardware.com

HTTP/1.1 200 OK
Date: Sun, 01 Oct 2000 23:25:17 GMT
Server: Apache/1.3.11 BSafe-SSL/1.38 (Unix) FrontPage/4.0.4.3
Last-Modified: Tue, 04 Jul 2000 09:46:21 GMT
ETag: "373979-193-3961b26d"
Accept-Ranges: bytes
Content-Length: 403
Connection: close
Content-Type: text/html
```

Example 1-1. An HTTP transaction using telnet (continued)

```
<HTML>
<HEAD><TITLE>Joe's Tools</TITLE></HEAD>
<BODY>
<H1>Tools Page</H1>
<H2>Hammers</H2>
<P>Joe's Hardware Online has the largest selection of hammers on the earth.</P>
<H2><A NAME=drills></A>Drills</H2>
<P>Joe's Hardware has a complete line of cordless and corded drills, as well as the latest
in plutonium-powered atomic drills, for those big around the house jobs.</P> ...
</BODY>
</HTML>
Connection closed by foreign host.
```

Telnet hostname'e bakar ve www.joes-hardware.com web serveri ile baglantiyi saglar,port 80 dinleyerek tabi ki.Komuttan 3 satir sonra telnet bize baglantinin saglandigini soyluyor.

Basit istek komutunu yazdiktan sonra "GET /tools.html HTTP/1.1" ve Host headerini yollamak bize orjinal hostname'i sagladi,sonra bosluk satiri,serverin bize "/tools.html" kaynagi getirmesi icin GET methodu ile www.joes-hardware.com 'dan kaynagi istedik.Sonra,server cevap satiriyla cevabi yolladi,birkac cevap headeri,bosluk satiri ve body kısmi yani HTML documenti

Telnet HTTP istemcisi olmayi iyi taklit eder ama bu server olarak o kadar ise yaramaz Bundan kacinin.Ve otomatik telnet scriptleride hos degil.Daha esnek tool istiyorsaniz,netcate bakmak isteyebilirsiniz netcat(nc).netcat toolu kolayca UDP,TCP trafiklerini HTTP yi yonetmenizi saglar.

Protokol Versiyonlari

Birkac cesit HTTP protokolu var bugunlerde kullanılan.HTTP uygulamalarinin saglam olarak siki calismasi icin HTTP'nin farkli varyasyonlu versiyonlarına ihtiyac duyulmus.

Versiyonlar:

HTTP\0.9

1991 de ortaya cikan HTTP'nin prototip versiyonu HTTP/0.9.Bu protocol cok ciddi sikintilar barindiriyor ve sadece birlikte calisan istemcilerde kullanilmasi oneriliyor.Sadece GET method destekleniyor.MIME tipleri,HTTP headerlari ve versiyon numaralari yok.HTTP/0.9 orjinalde HTML dosyalarini cekmek icin tanimlanmistir.Kisa surede yerini HTTP/1.0 ile degistirmistir.

HTTP/1.0

Her yonuyle gelistirilen ilk HTTP versiyonudur.HTTP/1.0 da versiyon numaralari,HTTP headerlari,fazladan methodlar ve multimedia obje yonetimi eklenmistir.HTTP/1.0 pratik olarak grafik cazipli sayfalar ve interaktif formlar ortaya cikmistir.Genis olarak World Wide Web'e adapte olunmasina yardimci olmustur HTTP/1.0.Bu ozellik asla iyi olarak belirlenmemistir.Zamanina gore protocol koleksiyonlarından en pratigi,hizlisi ve evrileni olarak gosterilir.

HTTP/1.0+

Birçok web istemcisi ve serverları 90'ların ortasında hızlıca HTTP'yi kullanmaya başlamış ve World Wide Web'e integre etmeye başlamıştır. Çoğu grup kendi extension modellerini çıkararak ve HTTP'ye katkıda bulunarak kendilerine göre HTTP'yi yığmışlardır. Örnek olarak Uzun süren "keep-alive" bağlantıları, sanal host desteği, proxy bağlantı desteği gibi. Bu eklentiler HTTP/1.0+ 'yi meydana getirmiştir. HTTP/1.0'in extended versiyonu diyebiliriz.

HTTP/1.1

HTTP/1.1 HTTP'deki mimari dizayn sıkıntılara odaklanmıştır. Anlam kazandırma, özellikle performans optimizasyonu ve eksik veya yanlış şeyler düzeltilmiştir. HTTP/1.1 90'ların sonuna doğru aynı zamanda daha komplike web uygulamalarını desteklemeye başlamıştır. HTTP/1.1 HTTP'nin şu anki versiyonudur.

HTTP-NG(HTTP/2.0)

HTTP/1.1'in performans optimizasyonuna ve daha güçlü çerçeve çalışmasına odaklanarak daha iyisini ortaya çıkarmak için meydana gelen bir prototiptir. HTTP/2.0 araştırmaları 98'de bitirilmiştir. HTTP/1.1'in yerini alması içinde bir çalışma veya plan yoktur.

Web Mimari Ogeleri

Bu bölümde özetle, iki web uygulamasına yönelicez. Web serverları ve tarayıcılar. Bu uygulamaları işlemlerini nasıl gerçekleştiriyor. Buna bakıcaz.

İnternetle etkileşime geçmiş bir sürü web uygulamaları var. Şimdi onlara bir bakalım.

Proxiler

Server ve istemci arasında durarak onlara aracı olur.

Cacheler(Onbellek)

HTTP depoları kullanıcıya yakın olan popüler web sayfalarının kopyasını tutar.

Gatewayler

Diğer uygulamalara bağlanan özel web serverları

Tuneller

HTTP iletişimine kafa atarak iletişime giren özel proxiler

Ajanlar

Otomatik HTTP istegi yapan yarı-zeki web istemcileri

Proxiler

HTTP proxy serverlarına göz gezdirerek başlayalım. Web güvenliği, uygulama integrasyonu ve performans optimizasyonu için önemli yapılardır.

Aşağıdaki resimde gördüğünüz gibi proxiler istemci ile server arasında durur. Tüm İstemci isteklerini alır ve geçisi Servera yapar (İstegi düzenleyebilme şansınız var). Bu uygulamalar kullanıcı için proxy olarak davranır. Kullanıcı adına servera bağlanır.

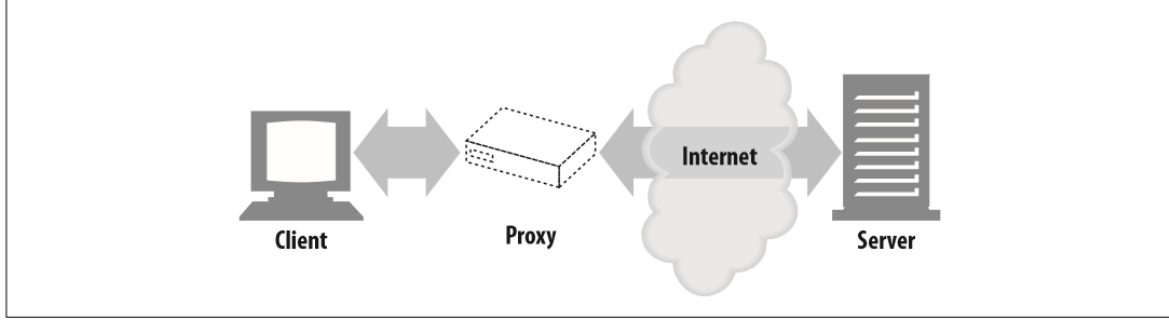


Figure 1-11. Proxies relay traffic between client and server

Proxiler sıklıkla güvenlik için kullanılır. Güvenli bir aracı olarak davranır üzerinden tüm web trafiği akışı sağlanır. Proxiler aynı zamanda istek ve cevapları filtreleyebilir. Örneğin; şirketi indirmelerinde bir virüsü tespit etmek için veya yetiskin içeriklere filter koyarak orta-okul çocuklarını korumak gibi. Proxilerle ilgili Bölüm 6 da konuşacağız.

Onbellek(Cacheler)

Web onbellegi veya proxy onbellekleme HTTP proxy serverin popüler kopya documentlerini proxyden geçirir. Bir dahaki istemci aynı documenti istediği zaman cache'nin personel kopyası sunulur.

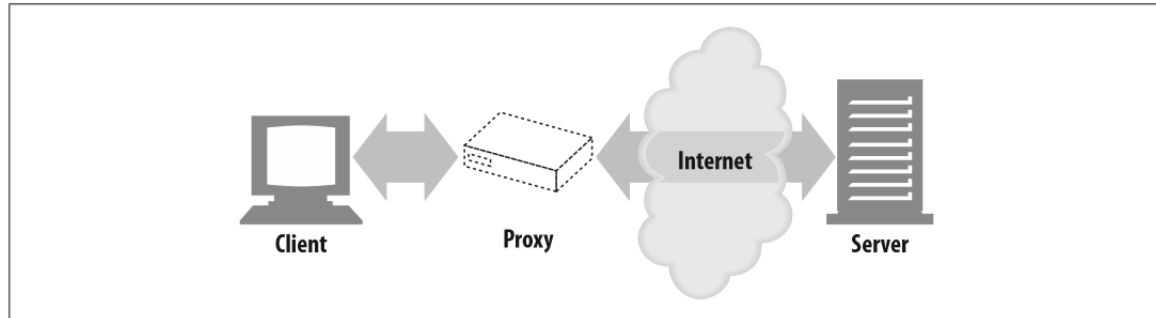


Figure 1-11. Proxies relay traffic between client and server

Bu sayede belkide kullanıcı documenti proxy cache sayesinde daha hızlı indirebilecek. HTTP daha efektif onbellekeme ve özel onbellek içeriklerini düzenlenmesi için bir sürü imkanlara sahiptir. Bölüm 7'de cache teknolojilerine değineceğiz.

Gatewayler

Gatewayler serverlar için aracı olarak davranan özel bir serverdır. Genellikle HTTP trafğini başka bir protokole dönüştürmek için kullanılır. Eğer asıl server kaynaksa gateway her zaman istekleri alır. Kullanıcı iletişimde gateway kullanıldığının farkında olmayabilir.

Örneğin, bir HTTP/FTP gatewayi FTP isteklerini alır ve HTTP istegi olarak devam eder ama documenti FTP protokolu ile çeker. Dokuman sonuc olarak HTTP mesajı olarak kullanıcıda sonuclanır. Bölüm 8 de Gatewayleri tartışacağız.

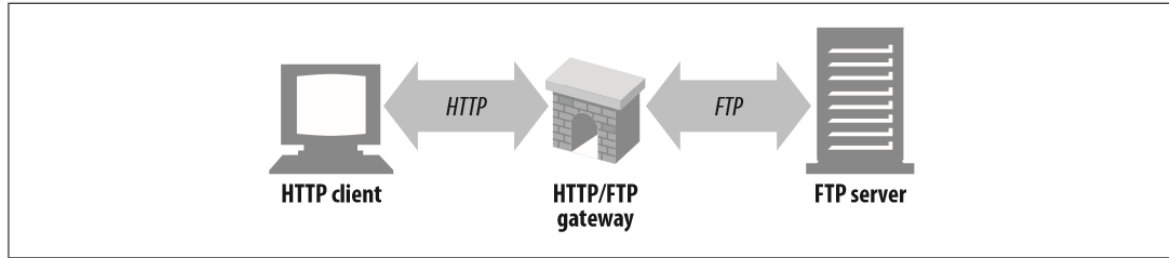


Figure 1-13. HTTP/FTP gateway

Tuneller (Havalandırma - yemezler o bir kere olur)

Tuneller HTTP uygulamalarıdır, yüklendikten sonra, iki bağlantı arasındaki raw datayı hiç içine bakmadan geçirir. HTTP tunnelleri sıklıkla non-HTTP datayı birçok kez HTTP bağlantısıyla taşımakta kullanılır, verinin içeriğine bakılmaksızın hemde.

Popüler bir tunnel hepimiz biliyorsunuz SSL (Secure Sockets Layer). Enkript edilerek HTTP bağlantı trafiği sağlanır. Şirketlerin güvenlik duvarları sadece SSL'li web trafiğine izin verir. Aşağıdaki resimde gördüğünüz gibi bir HTTP/SSL tunneli HTTP istegini alır hedefle adres ve portu kullanıp bağlantı kurar, sonra SSL trafiği üzerinden ilerleyerek HTTP kanalında enkript bir şekilde hedef servera ulaşır.

Ajanlar

Ajanlar kullanıcı adına HTTP istegi yapan programlardır. Web istegi yapabildiğimiz herhangi bir program HTTP ajanıdır. Şimdiye kadar, sadece bir tane HTTP ajanından bahsettik, web tarayıcıları (browserları). Ama bir sürü çeşit ajan var tabiki.

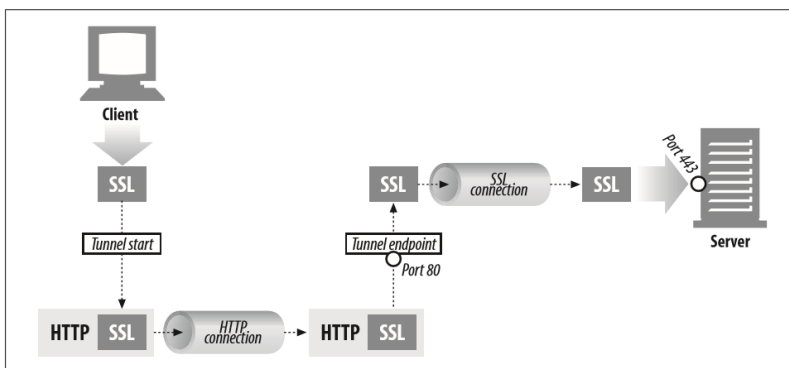


Figure 1-14. Tunnels forward data across non-HTTP networks (HTTP/SSL tunnel shown)

Ornegin,Webi otonomlu bir otomatik makine ajani dolasiyor.HTTP islemleri ve icerik cekmek gibi seyler yapiyor,insan vizyonu olmadan.Biz bu otonomlari siklikla renkli isimler takariz,spiders(orumcek) veya web robotlari gibi(Asagidaki resimde goruceksiniz).Orumcekler web iceriklerinin kullanisi arshivlerini dolasirlar,arama motoru veya urun kataloglarini karsilastiran bir robot gibi.Bolum 9 da daha fazla bilgi olacaktir.

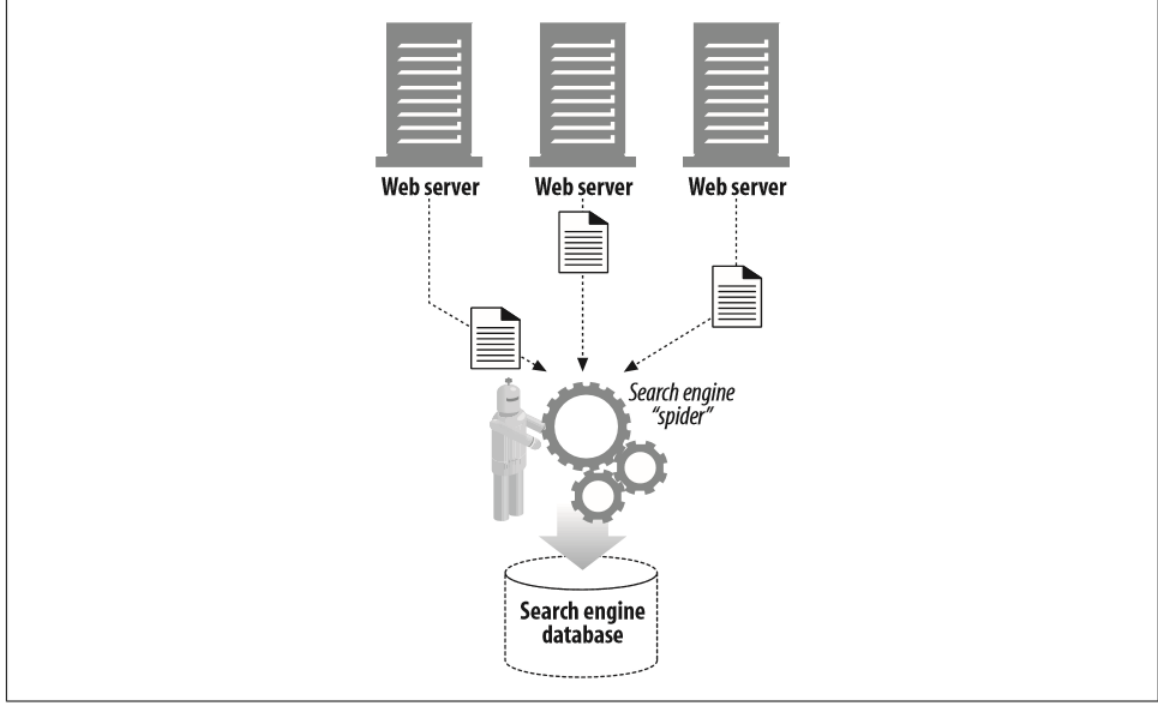


Figure 1-15. Automated search engine “spiders” are agents, fetching web pages around the world

Baslangicin Bitisi

Burdaki bilgiler HTTP hakkında giris bilgileri idi.HTTP’yi multimedia tasima protokolu olarak role burdurduk.HTTP URI’leri nasil kullanir,HTTP istek ve cevap mesajlari nasil yonetildigini gorselledik,ve sonda HTTP’in kullandigi az cok web uygulamarina baktik.

Geriye Kalan bolumler HTTP protokolu,uygulamalar ve kaynaklar hakkında daha teknik bilgiler icerecektir.

Daha Fazla Bilgi

Kitabin ilerleyen bolumlerinde HTTP hakkında daha fazla detaya girecegiz.Ilerleyen zamanlarda daha ozel spesifik bilgilere girecegiz.Kendinizde arastirmalar yapabilirsiniz HTTP hakkında internette wikipediden tutun forumlara kadar bir suru kaynak var ama benim bu cevirisinde bulundugum kitap HTTP’in baba kitabi olarak bilinir ve internet hakkında internet teknolojileri hakkında tum bilgileri icerir.Ben bu kitapta bulamayacaginiz baska yerde bulucaginiz bir bilgi bulabileceginizi sanmiyorum,zaten bu yuzden bu kitabi cevirmeye calisiyorum ve ozellikle kitabın konularından ve iceriginden disari cikmayarak elimden geldigince dili sade tutmaya calisiyorum.Umarim size yardimi dokunur.