

HTTP'yi (istemci tarafı) kendi kendinize deneyin

1. Bir Web server'a telnetle bağlan:

```
telnet 194.27.49.11 80
```

Port 80'e bir TCP bağlantısı açar (varsayılan HTTP sunucu portu) Yazılan herşey **194.27.49.11** 'da port 80'e gönderilir

2. GET HTTP isteğini girin:

```
GET /ee/bcavusoglu/ HTTP/1.1  
Host:194.27.49.11
```

Bu komutu yazarak (enter'a iki kere basın), bu minimum fakat tam GET isteğini HTTP sunucusuna gönderirsiniz

3. HTTP sunucusu tarafından gönderilen cevaba bakın!

Kullanıcı-sunucu durumu: çerezler

Çoğu Web sitesi çerezleri kullanır

4 içerik:

- 1) HTTP cevap mesajında çerez başlık satırı
- 2) HTTP istek mesajında çerez başlık satırı
- 3) Çerez dosyası kullanıcının hostunda saklanır ve kullanıcının internet tarayıcısı tarafından yönetilir
- 4) Sunucuda çerez veri tabanında saklanır

Örnek:

- Ahmet Internet'e sürekli aynı bilgisayardan erişiyor
- Bir e-ticaret sitesini ilk defa ziyaret ediyor
- İlk HTTP isteği siteye ulaştığında site yeni bir kimlik numarası üretir ve veri bankasına atar

Çerezler: "durum" takibi

istemci

sunucu

Çerez dosyası
ebay: 8734

http istek mesajı
http cevabı +
Set-cookie: 1678

sunucu
1678 kimliğini
kullanıcı için
oluşturur

Veri bankasına



Çerez dosyası
amazon: 1678
ebay: 8734

http istek mesajı
cookie: 1678
http cevap mesajı

Çerez-merkezli
aktivite

erişim

Bir hafta sonra:

Cookie file
amazon: 1678
ebay: 8734

http istek mesajı
cookie: 1678
http cevap mesajı

Çerez-merkezli
aktivite

erişim

Çerezler

Çerezler ne getirir:

- ❑ yetki
- ❑ Alışveriş çantaları
- ❑ tavsiyeler
- ❑ Kullanıcı oturum durumu (Web e-posta)

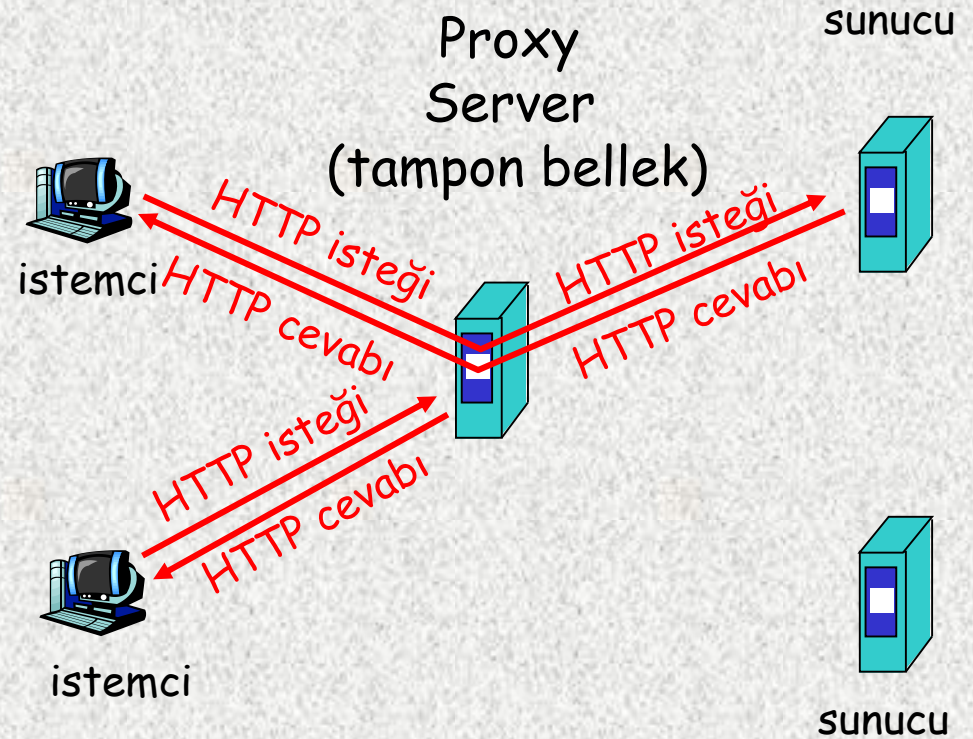
Çerezler ve gizlilik:

- ❑ Çerezler sitelerin sizin hakkınızda bilgi edinmesine izin verir
- ❑ Bu sitelere isim ve e-posta adresi verebilirsiniz

Web tampon belleđi (proxy server)

Amaç: istemcinin isteđini sunucuyu karıştırmadan yerine getirmek

- Kullanıcı internet tarayıcısını ayarlar: Web'e tampon bellek vasıtasıyla ulaşıyorum
- Tarayıcı bütün HTTP isteklerini tampon belleđe gönderir
 - Nesne bellekte: bellek nesneyi gönderir
 - deđilse bellek nesneye sunucudan ister ve sonra istemciye gönderir



Web tampon belleđi

- ❑ Bellek hem istemci hem sunucu olarak hareket eder
- ❑ Genelde bellek ISP tarafından oluşturulur (üniversite, şirket, yerel ISP)

Neden Web tamponu?

- ❑ İstemci isteđi için cevap süresini azaltır.
- ❑ Kurumun hatlarındaki trafiđi azaltır.

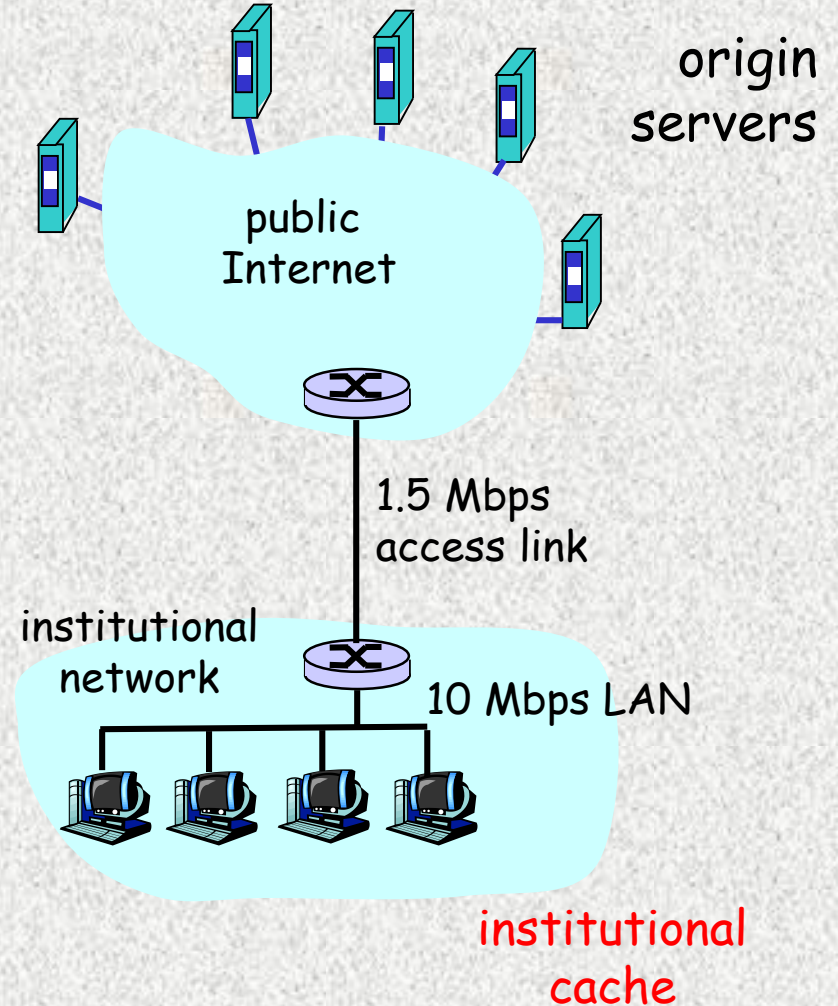
Tampon Bellek Örneği

Varsayımlar

- ❑ ortalama nesne büyüklüğü = 100,000 bit
- ❑ Kurumdan sunucuya avg. İstek sayısı = 20/sn
- ❑ Kurumdan sunucuya ve geri yönde gecikme = 2 sn

Sonuçlar

- ❑ LAN kullanımı = 20%
- ❑ Erişim linki kullanımı = 100%
- ❑ Toplam gecikme = İnternet gecikmesi + erişim gecikmesi + LAN gecikmesi
= 2 sn + dakikalar + milisaniyeler



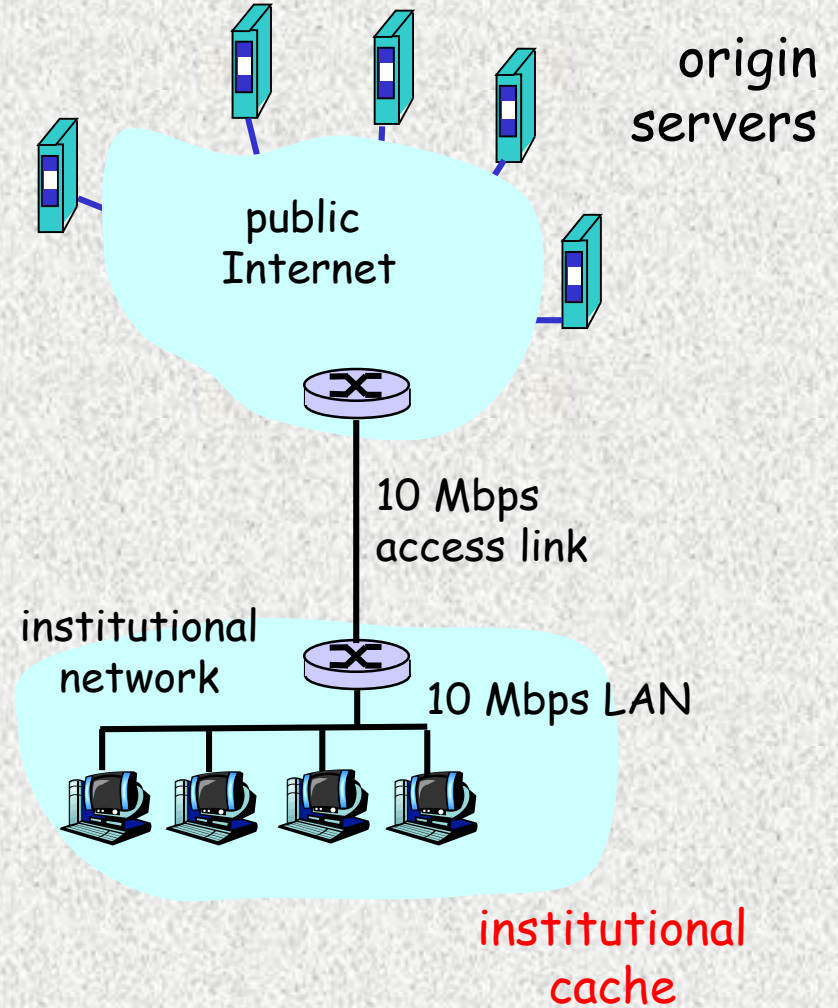
Tampon Bellek Örneği

Olası çözüm

- ❑ Bant genişliğini artırmak, 10 Mbps

Sonuçlar

- ❑ LAN kullanımı= 20%
- ❑ erişim linki kullanımı = 20%
- ❑ Toplam gecikme = Internet gecikmesi + erişim gecikmesi + LAN gecikmesi
= 2 sec + msec + msec
- ❑ Pahalı bir çözüm



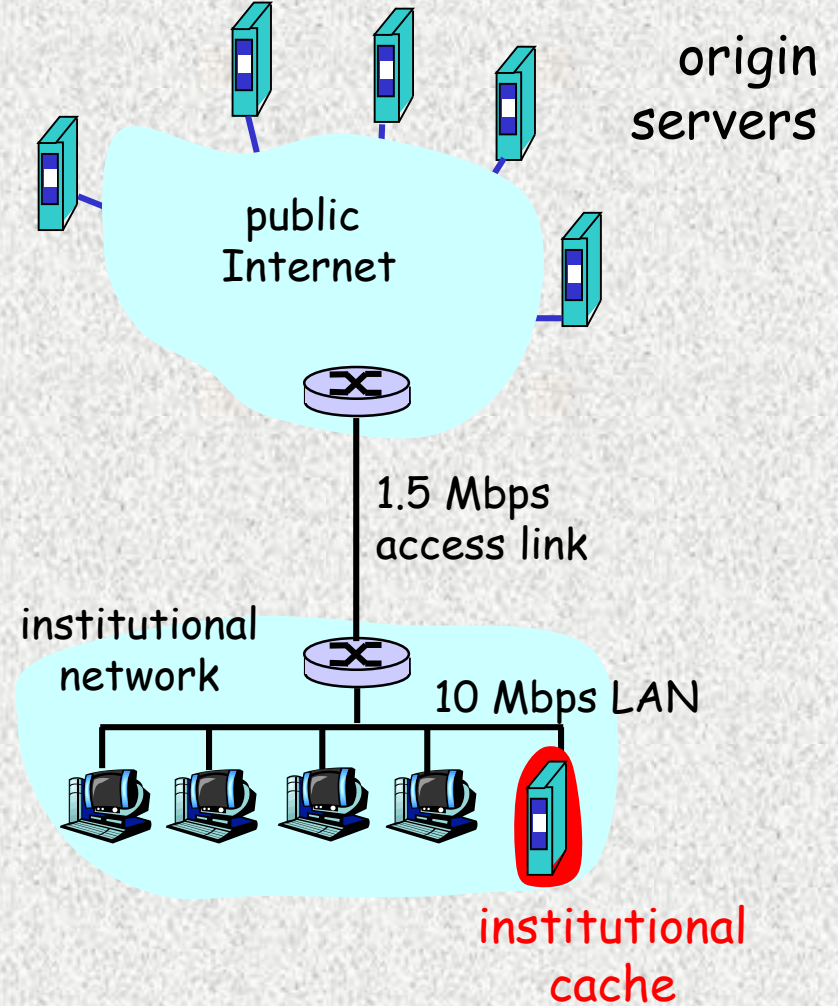
Tampon Bellek Örneği

Bellek oluştur

- Vurma oranı .4 olsun

Sonuç

- İsteklerin 40%'ı neredeyse anında karşılanacak
- İsteklerin 60% orjinal sunucu tarafından karşılanacak
- Erişim linkinin kullanımı 60%'a düşer, buda ihmal edilebilecek gecikmeleri mümkün kılar (~10 msn)
- toplam averaj gecikme =
Internet gecikmesi + erişim gecikmesi + LAN gecikmesi =
.6*(2.01) sn + millisaniyeler



Şartlı GET

- **Amaç:** Eğer tampon bellek nesnenin güncel versiyonuna sahipse nesneyi gönderme
- Tampon bellek: tampondaki nesnenin kopya tarihini HTTP isteğinde belirt
If-modified-since:
<tarih>
- sunucu: eğer tampondaki nesne güncelse sunucu nesneyi göndermez:
HTTP/1.0 304 Not Modified

Tampon bellek

SUNUCU

