

UYGULAMA KATMANI

Ağ çekirdeğindeki araçlar için yazılım yazılmaz:

- ☐ Ağ çekirdeği araçları uygulama katmanında çalışmazlar
- ☐ Bu tasarım hızlı uygulama geliştirmeye izin verir.

İSTEMCİ-SUNUCU MİMARİSİ:

Sunucu:

- Daima açık bir ana sistem
- Sabit IP adresli
- Ana sistem kümesi –sunucu çiftliği (server farms)

İstemci:

- sunucu ile iletişim kurar
- İstemci her zaman açık olmak zorunda değildir
- Dinamik (değişebilir) IPadresine sahip olabilir
- Birbirleriyle direk olarak iletişime geçmezler

P2P MİMARİSİ:

- Her zaman açık olan sunuculara ihtiyaç yoktur
- Gelişigüzel uç sistemler birbirleriyle direk olarak iletişime geçerler
- Uç sistemler birbirleriyle ara sıra bağlanırlar ve IP adreslerini değiştirirler
- Örn: Gnutella

Büyük oranda ölçeklenebilir(scalable)

Fakat yönetilmeleri zordur

İstemci-Sunucu ve P2P Mimarisini Birleştiren Melez Mimari:

Napster:

- Dosya transferi P2P olarak gerçekleştirilir.
- Dosya arama merkezidir.
 - Eşler, içeriği merkezi sunucuya kaydeder
 - Eşler, içeriği bulmak için aynı merkezi sunucuyu sorgular

Anında Mesajlaşma:

İki kullanıcının mesajlaşması P2P dir

Varlık tespiti/bulması (presence detection/location) merkezidir :

- Kullanıcılar IP adreslerini çevrimiçi olunca merkezi sunucuya kayıtlı ederler.
- Kullanıcılar diğerlerinin IP adreslerini bulmak için merkezi sunucu ile iletişime geçerler.

Süreç(proses) Bir uç sistemde çalışan program.

- Aynı uç sistem üzerinde,iki süreç iç-süreç iletişimi (işletim sistemi tarafından tanımlanan) kullanarak birbirleriyle iletişime geçebilirler.
- Farklı sistemler üzerinde çalışan süreçler mesaj alış veriş ile haberleşirler

İstemci süreçleri:

İletişimi başlatan süreç.

Server süreci:

İletişime geçilmeyi bekleyen süreç.

Not: P2P mimarideki uygulamalarda hem istemci süreçleri hem de sunucu süreçleri yer alır.

Proses ve Bilgisayarlar arasındaki arabirimler:(Soketler)

Bir proses, mesajlara ağa soket adındaki yazılımlar aracılığıyla gönderir ve alır.

Soket kapı benzeşmesi:

- Gönderici proses mesajı kapıya(sokete) doğru iter
- Gönderici proses, kapının diğer tarafında mesajı hedef sürecin kapısına taşıyacak olan bir taşıma altyapısının olduğunu varsayar.

Soketlere API (Uygulama Arabirimi) de denir.

Uygulama geliştiricileri soketin:

- uygulama tarafında her şey üzerinde kontrolleri vardır
- Taşıma tarafında ise sadece (1) taşıma protokolünün seçimi; (2) bazı parametreleri ayarlayabilme(maksimum tampon ve minimum segment boyutları gibi) üzerinde kontrolleri

Prosesleri işaret etmek:

Bir sürecin mesaj alabilmesi için bir tanımlayıcı olması gerekir.(Identity)

Bir tanımlayıcı sistem 32 bitlik bir IP adresine sahiptir.

Süreci tanımlayabilmek için sürecin çalıştığı ana sistemin sadece IP adresini bilmesi yeterli midir?

Cevap: Hayır, aynı ana sistem üzerinde pek çok süreç çalışıyor olabilir

❓ Tanımlayıcı hem IP adresini hem de süreçle ilgili hedef bağlantı noktası numarasına (port numbers) sahiptir

❓ Örnek port numaraları

❓ HTTP sunucusu: port 80

❓ Posta sunucusu: port 25

UYGULUMA KATMANI PROTOKOLLERİ NELERİ TANIMLAR:

Karşılıklı iletilen mesajlar, örn. istek&cevap mesajları

❑ Mesajların sentaksı: mesajlardaki alanlar ve bu alanların nasıl tanımlandığı

❑ Alanların anlamları, alanlardaki bilgilerin anlamları

❑ Süreçlerin ne zaman mesajları alıp göndereceklerine dair kurallar.

Genel etki alanı Protokolleri:

RFC lerde tanımlanmıştır

❑ Birlikte işlerliğe (interoperability) olanak sağlar

❑ Örn., HTTP, SMTP

Özel Protokoller:

Örn: KaZaA

Uygulama Tarafından İhtiyaç Duyulan Taşıma Servisleri:

Veri kaybı(data loss):

Bazı uygulamalar bir miktar veri kaybını tolere edebilir (örn., ses/video)

❑ Diğerleri 100% güvenli veri transferine ihtiyaç duyarlar (örn., dosya transferi, telnet)

Zamanlama Timing:

Bazı uygulamalar “etkili” olabilmek için az gecikme gerektirirler (örn., Internet telefonu, etkileşimli oyunlar)

Bant Genişliği(Bandwidth):

Bazı uygulamalar “etkili” olabilmek için belli bir miktarda bant genişliğine ihtiyaç duyarlar (örn., multimedia)

❑ Diğerleri bant genişliğini ne kadarını kullanabilirlerse onunla idare edebilirler (“elastik uygulamalar)

Internet Tarafından Taşıma Servisleri:

TCP servisleri:

❑ Bağlantı-yönelimli (connectionoriented): istemci ve sunucu süreçler arasında bağlantı kurulumu gerekir

❑ Gönderici ve alıcı süreçler arasında güvenilir taşıma (reliable transport)

❑ Akış kontrolü (flow control): gönderici alıcıyı sıkıştırmaz

❑ Tıkanıklık kontrolü (congestion control): ağ çok yüklendiğinde göndericinin gönderdiği veri miktarını kısar

❑ Zamanlama, minimum bant genişliği garantisi sağlamaz.

UDP Servisleri:

Gönderici ve alıcı süreçler arasında güvenilir olmayan veri iletimi

❑ Bağlantı kurulumu (connection setup), güvenilirlik (reliability), akış kontrolü (flow control), tıkanıklık kontrolü (congestion control), zamanlama (timing), veya bant genişliği garantisi (bandwidth guarantee) SAĞLAMAZ

HTTP : hypertext transefer protokol

Web'in uygulama katmanı protokolüdür.

❑ İstemci/sunucu modeli

❑ istemci: Web nesnelerini isteyen, alan ve gösteren tarayıcı

❑ sunucu: isteklere karşılık olarak nesneleri gönderen web sunucusu

❑ HTTP 1.0: RFC 1945

❑ HTTP 1.1: RFC 2068

Tcp kullanır:

İstemci sunucuyla port 80'den TCP bağlantısı (socket oluşturur) başlatır

❑ Sunucu istemcinin TCP bağlantı isteğini kabul eder.

❑ HTTP mesajları (uygulama katmanı protokolü mesajları) tarayıcı (HTTP istemcisi) ve web sunucusu (http sunucusu) arasında karşılıklı gönderilir

❑ TCP bağlantısı kapatılır.

Kalıcı Olmayan http: (nonpersistent)

Genellikle bir TCP bağlantısı ile bir nesne gönderilir.

❑ HTTP/1.0 kalıcı olmayan (nonpersistent) HTTP'yi kullanır

Kalıcı Olan http: (persistent)

İstemci ile sunucu arasında tek bir TCP bağlantısı ile birden fazla nesne gönderilebilir.

❑ HTTP/1.1 varsayılan modunda kalıcı (persistent) bağlantı kullanır

Cevap Zamanı(response time):

RTT: İstemciden sunucuya giden bir paketin gidiş dönüş zamanı

Cevap zamanı (responsetime):

- ☐ TCP bağlantısını başlatmak için bir RTT
- ☐ HTTP isteği ve http cevabının ilk birkaç bitinin dönüşü için bir RTT
- ☐ Dosya iletim zamanı toplam= 2RTT+iletim zamanı

Kullanıcı Tarafli Etkileşim(Cerezler) Cookie

Pek çok web sitesi cookie kullanır

Dört bileşen:

- 1) HTTP cevap mesajındaki bir cookie başlık satırı
- 2) HTTP istek mesajındaki bir cookie başlık satırı
- 3) Kullanıcın uç sisteminde tutulan ve tarayıcı tarafından yönetilen bir cookie dosyası
- 4) Web sitesindeki bir arka uç veritabanı

Cookie lerin sağladıkları:

- ☐ Yetkilendirme(authorization)
- ☐ Alışverişsepeti(shopping carts)
- ☐ Tavsiyeler
- ☐ Kullanıcı oturum durumu (user sessionstate) (Web e-mail)

Web tampon belleği (caches) (vekil sunucusu -proxy server)

Amaç: HTTP isteklerini köken Web sunucusu yerine karşılayan bir ağ varlığıdır.

Kullanıcı tarayıcısını, kullanıcın tüm http isteklerini önce bu Web tampon belleğine yönlendireceği şekilde yapılandırabilir

- ☐ Tarayıcı tüm http isteklerini bu tampon belleğe yönlendirir.
- ☐ Nesne tampon bellekte ise: tampon bellek nesneyi istemciye döndürür.
- ☐ Nesne tampon bellekte değilse: tampon bellek köken sunucudan nesneyi ister ve sonrasında nesneyi istemciye döndürür.

Tampon bellek aynı anda hem sunucu hem istemci gibi davranır

- ☐ Genel olarak tampon bellekler bir ISP(Internet servis sağlayıcı) tarafından kurulurlar (üniversite, şirket, yerleşim yeri ISP'si)

Neden web tampon belleği?

☐ Bir istemci isteği için cevapzamanını azaltır

☐ Kurumun erişim hattındaki trafiği azaltır.

☐ Tampon belleklerin yoğun olduğu bir Internet “zayıf” içerik sağlayıcıların içeriği daha iyi sunmalarını sağlar (fakat aynısını P2P dosya paylaşımı da yapabilir)

Tampon bellek kurarsak

☐ Hit alma oranı .4 varsayılırsa

Sonuçları

☐ İsteklerin 40% ‘ı neredeyse hemen karşılanır

☐ İsteklerin 60% ‘ı köken sunucular tarafından karşılanır

☐ Erişim hattının kullanılması 60 % ‘a azaltıldı, önemsiz gecikmeler olabilir (10 msn kadar)

☐ Toplam ort. gecikme = Internet gecikmesi + erişim gecikmesi + LAN gecikmesi

Amaç: eğer tampon bellek güncel sürüme sahipse nesneyi gönderme

☐ Tampon bellek: HTTP isteği içerisinde belleğe alınmış kopyanın tarihini belirler

If-modified-since:

<date>

☐ sunucu: eğer tampondaki kopya güncelse cevapta nesne bulunmaz:

HTTP/1.0 304 Not Modified

Dosya transfer protokolü :FTP (file transfer protocol)

Uzak ana sistem e/den dosya transferi

☐ İstemci sunucu modeli

☐ *istemci*: transferi başlatan taraf

☐ *sunucu*: uzaktaki ana sistem

☐ ftp: RFC 959

☐ ftp sunucus: port 21

FTP: ayrı kontrol ve veri bağlantıları

FTP istemcisi FTP sunucusu ile TCP’yi taşıma protokolü olarak kullanarak port 21 üzerinden iletişim kurar

☐ İstemci kontrol bağlantısı üzerinden yetkilendirme alır.

☐ İstemci uzaktaki dizinleri kontrol bağlantısı üzerinden gönderdiği komutlar aracılığıyla tarar.

☐ Sunucu dosya transferi ile ilgili bir komut aldığında, sunucu istemciye TCP veri bağlantısı açar.

☐ Dosyayı transfer ettikten sonra sunucu bağlantıyı kapatır.

Sunucu başka bir dosya transfer edilmek istenirse yeni bir TCP veri bağlantısı açar.

☐ Kontrol bağlantısı: “bant dışı (out of band)”

☐ FTP sunucusu “durum” korumalıdır: halihazırdaki dizin, önceki yetkilendirmeler

Sayfa 46 Ftp komutları cevapları

ELEKTRONİK POSTA:

Üç ana bileşen

- ☐ Kullanıcı temsilcileri (user agents)
- ☐ Posta sunucuları (mail servers)
- ☐ Basit posta transfer protokolü (simple mail transfer protocol: SMTP)

User Agent (Kullanıcı aracı)

- ☐ a.k.a. “posta okuyucuları”
- ☐ Mesaj okuma, cevap verme, iletme, kaydetme ve oluşturma
- ☐ e.g., Eudora, Outlook, elm, Netscape Messenger
- ☐ Sunucuda depolanan gelen ve giden mesajlar

ELEKTRONİK POSTA: POSTA SUNUCUSU:

Posta Sunucuları

- ☐ Posta kutusu kullanıcıya gelen mesajları içerir.
- ☐ Mesaj kuyruğu giden (gönderilecek) mesajları içerir
- ☐ SMTP protokolü posta sunucuları arasında posta göndermek için kullanılır
- ☐ istemci: gönderen posta sunucusu
- ☐ “sunucu”: alıcı posta sunucusu

Elektronik Posta: SMTP [RFC 2821]

- ☐ İstemciden sunucuya eposta mesajlarını güvenilir olarak transfer etmek için port 25 üzerinden TCP kullanır
- ☐ Direk transfer: gönderen sunucudan alıcı sunucuya
- ☐ Üç fazlı transfer
 - ☐ El sıkışma (greeting)
 - ☐ Mesajın iletimi
 - ☐ Kapatma (closure)
- ☐ Komut /cevap etkileşimi (command/response interaction)
 - ☐ komutlar: ASCII metin
 - ☐ cevaplar: durum kodları ve ilgili deyimler
- ☐ Mesajlar 7 bit’lik ASCII metinleridir.

Senaryo: Alice Bob'a mesaj gönderir

- 1) Alice kullanıcı temsilcisi kullanarak mesaj gönderir ve "to" satırına bob@someschool.edu yazar
- 2) Alice'in kullanıcı temsilcisi mesajı kendi posta sunucusuna gönderir; mesaj mesaj kuyruğuna konur
- 3) SMTP'nin istemci tarafı Bob'un posta sunucusuna TCP bağlantısı açar
- 4) SMTP istemcisi Alice'in mesajını TCP bağlantısı üzerinden gönderir.
- 5) Bob'un posta sunucusu mesajı Bob'un posta kutusuna koyar
- 6) Bob mesajı okumak için kendi kullanıcı temsilcisini çalıştırır.

SMTP SON NOTLAR:

SMTP kalıcı (persistent) bağlantı kullanır

☐ SMTP mesajları 7 bit'lik ASCII metinden oluşur (header & body)

☐ SMTP sunucusu msaj sonlarını belirtmek için CRLF.CRLF kullanır

CR : carriage return

LF : line feed

SMTP HTML ARASINDAKİ FARKLAR:

HTTP: çekme (pull)

☐ SMTP: itme (push)

☐ Her ikisi de etkileşim için ASCII komut ve cevaplarını, durum kodlarını kullanırlar

☐ HTTP: her nesne kendi cevap mesajı içerisine sarmalanır

☐ SMTP: tüm nesneler bir mesaj içerisine yerleştirilir.

☐ HTTP bant içi SMTP bant dışı kontrolü kullanır

SMTP:

eposta mesajlarının gönderimi için bir protokoldür

RFC 822: metin mesaj biçimini belirleyen standarttır:

- ☐ Başlık satırları, e.g.,
- ☐ To:
- ☐ From:
- ☐ Subject:
- SMTP komutlarından farklıdır !

☐ gövde

- ☐ Sadece ASCII karakterlerden oluşan "mesaj"

Mesaj biçimi: multimedia için

MIME: çok amaçlı Internet posta uzantıları (multi purpose mail extension), RFC 2045, 2056

☐ MIME içerik tipini tanımlamak için mesaj başlığına ek başlıklar eklenmelidir

Posta erişim protokolleri

☐ SMTP: alıcı sunucuya gönderme/depolama ile ilgilidir

☐ Posta erişim protokolleri: sunucudan alma ile ilgilidir

- ☐ POP: Post Office Protocol [RFC 1939]
 - yetkilendirme (agent <-->server) ve download
- ☐ IMAP: Internet Mail Access Protocol [RFC 1730]
 - Daha fazla özellik (daha karmaşık)
 - Sunucuda depolanan mesajların idaresi
- ☐ HTTP: Hotmail , Yahoo! Mail, etc.

DNS: Internet'in Dizin Servisi

İnsanlar: pek çok tanımlayıcı:

- SSN, isim, pasaport #

Internet ana sistemleri, yönlendiriciler:

- IP adreslerini (32 bit) - datagram ların adreslenmesi için kullanılırlar
- İnsanlar "isim", e.g., ww.yahoo.com - kullanırlar

Q: IP adresleri ile isimler arasında eşleşme nasıl sağlanır?

Etki Alanı Ad Sistemi

(Domain Name System):

☐ Bir *DNS sunucu* hiyerarşisi içerisinde uygulanan *dağıtık bir veritabanı (distributed database)*

☐ Ana sistemlerin dağıtık veritabanı sorgulamasını sağlayan bir *uygulama katmanı protokolüdür.*

- not: ana Internet fonksiyonu,uygulama katmanı protokolü olarak uygulanır

- complexity at network's "edge"

DNS servisleri

- ☐ Ana sistem isimleri-IP adres çevirisi
- ☐ Ana sistem lakapları(aliasing)
 - ☐ Kurallı (canonical) ve lakap (alias) adları
- ☐ Posta sunucusu lakapları
- ☐ Yük dağıtımı
 - ☐ Çoğaltılan web sunucuları için bir IP adresi seti, bir kurallı isimle ilişkilendirilir

Neden DNS merkezileştirilmiyor?

- ☐ Bir tek başarısızlık noktası
- ☐ Trafik hacmi
- ☐ Uzakta merkezileştirilmiş veritabanı
- ☐ bakımölçeklenemez!

İstemci www.amazon.com için IP istiyor; 1inci tahmin:

- ❓ İstemci com DNS sunucusunu bulmak için kök dizin sorgusu yapar
- ❓ İstemci com DNS sunucusunu amazon.com DNS sunucusunu bulmak için sorgular
- ❓ İstemci amazon.com DNS sunucusunu www.amazon.com un IP adresini almak için sorgular

Üst-seviye etki alanı (TLD) sunucuları: com, org, net,

edu, gibi, ve tüm üst düzey ülke etki alanlarından uk, fr, ca, jp, gibi sorumludurlar

- ❓ Network solutions com TLD sunucularını işletmektedir
- ❓ Educause edu TLD sunucularını işletmektedir

Yetkili DNS sunucuları: kurumların DNS sunucuları,diğer kurum sunucularına yetkili alan adı ve IP eşleşmesi sunarlar (e.g., Web ve posta).

- ❓ Kurum tarafından ya da servis sağlayıcı tarafından işletilebilirler

YEREL DNS SUNUCLARI:

- ❓ Katı bir şekilde hiyerarşiye ait değildir
 - ❓ Her ISP'ye ait bir tane vardır (yerel ISP, şirket, üniversite).
 - ❓ “default name server”
- ❓ Bir ana sistem DNS sorgusu yaptığıında, sorgu kendi yerel DNS sunucuna gönderilir
 - ❓ Vekil (proxy) gibi davranır, sorguyu hiyerarşiye gönderir.

Tekrarlanan sorgu(iterated query)

- ❓ Başvurulan sunucu başvurulacak sunucu adıyla cevap verir
- ❓ “Ben bilmiyorum ama şu sunucuya sorabilirsin”

Yinelenen Sorgu(reqursive query)

- ❓ İsim eşleştirmenin zorluğunu başvurulanan sunucuya yükler
- ❓ ağır yük?

Dns Tampon Bellek ve Kayıtları Güncelleme:

Bir sunucu eşleştirmeyi öğrendiğinde, eşleştirmeyi tampon belleğe alır

- ☐ Tampon bellek girdileri belli bir süre sonrakaybolurlar (timeout)(TTL)
- ☐ TLD sunucuları tipik olarak yerel DNSsunucularının tampon belleğinde yer alırlar
 - Bu nedenle kök DNS sunucularına sıklıkla başvurulmaz

☐ Güncelleme/uyarı mekanizması IETF'nintasarımındadır

- ☐ RFC 2136
- ☐ <http://www.ietf.org/html.charters/dnsind-charter.html>

DNS KAYITLARI:

DNS: kaynak kayıtlarını (resource records (RR)) depolayan dağıtık veritabanı

Type=A

- ☐ name sistem adıdır
- ☐ value IP adresineeşleme sunar

Type=NS

- ☐ name etki alanıdır (e.g. foo.com)
- ☐ value bu etki alanına ait yetkili DNS 'in IP sine eşleme sunar

Type=CNAME

- ☐ name lakab ana sistem adıdır www.ibm.com
aslında servereast.backup2.ibm.com
- ☐ value kurallı ana sistemadıdır

Type=MX

- ☐ Value, name ana sistemi lakabına sahip olan bir posta sunucusunun kurallı adıdır.

DNS Mesajları:

DNS protokolü : sorgu ve cevap mesajları, aynı mesaj biçimindedir.

msg başlığı

☐ Tanımlama (identification): sorgu için 16 bit lik alan

kullanılır ,

☐ Bayraklar (flags):

- ☐ Sorgu veya cevap
- ☐ İstenen yineleme
- ☐ Mevcut yineleme
- ☐ Cevap yetkilidir

DNS veritabanına kayıt girmek

Örnek: "Network Utopia" yı yeni oluşturduk

☐ networkutopia.com adını **kaydedici (registrar)** 'ye kayıt ettiririz. (örn, Network Solutions)

- ○ Kaydediciye yekiliDNS sunucumuzun adı ve IP adresini de vermek gerekir (birincil ve ikincil)
- ○ Kaydedici TLD sunucusuna iki RR ekler:

(networkutopia.com, dns1.networkutopia.com, NS)

(dns1.networkutopia.com, 212.212.212.1, A)

☐ Yetkili sunucuya www.networkutopia.com için Type A kaydı ve mail.networkutopia.com için Type MX kaydı konulur

☐ **İnsanlar sizin sitenizin IP adresine nasıl erişecekler?**

SOKET PROGRAMLAMA:

Amaç: soketleri kullanarak iletişim kuran bir istemci/sunucu uygulamasının nasıl yapılacağını öğrenmek.

Soket API'si

- ☐ 1981'de BSD4.1 UNIX ile tanıtıldı
- ☐ Uygulamalar tarafından yaratıldı, kullanıldı ve sunuldu
- ☐ İstemci/sunucu paradigması
- ☐ Soket API'si ile iki taşıma hizmeti:
 - Güvenilir olmayan datagram
 - güvenilir, byte stream yönelimli

TCP İLE SOKET PROGRAMLAMA:

Soket: uygulama süreci ve uçtan uca taşıma protokolü (UDP veya TCP) arasında bir kapı

TCP servisi: bir süreçten diğerine **bitlerin** güvenli iletimi

İstemci sunucu ile iletişime geçmeli

- ☐ Sunucu süreci öncelikle çalışır olmalı
- ☐ Sunucu istemcinin iletişimini karşılayacak soket (kapı) oluşturmalı

İstemci sunucuyla:

- ☐ İstemciye özel TCP soketi oluşturarak
- ☐ Sunucu sürecin IP adresini ve port numarasını belirleyerek iletişim kurar.
- ☐ **İstemci soket oluşturduğunda:**

İstemci TCP sunucu TCP si ile bağlantı kurmuş olur

İstemci tarafından iletişime geçen **sunucu TCP si** sunucu sürecinin istemciyle haberleşmesi için **yeni bir soket oluşturur. server TCP creates new socket**

- ○ Sunucunun birden fazla istemci ile konuşmasını sağlar
- ○ Kaynak port numaraları istemcilerin ayırt edilmesini sağlar

AKIM (STREAM) JARGON:

- ❑ **Akım (stream)** sürecin içine ya da içinden akan karakterler sırasıdır.
- ❑ **Girdi akımı (input stream)** süreç için bir girdi kaynağına bağlıdır, örn, klavye veya soket.
- ❑ **Çıktı akımı (output stream)** bir çıktı kaynağına bağlıdır, eg, monitor veya soket.

Örnek istemci/sunucu uygulaması

- 1) İstemci standart girdi den satır okur (inFromUser stream) , soket aracılığıyla sunucuya gönderir (outToServer stream)
- 2) Sunucu soketten satırı okur
- 3) Sunucu satırı büyük harfe çevirir, istemciye geri gönderir
- 4) İstemci okur, soketteki değişen satırı çıktı alır (inFromServer stream)

UDP İLE SOKET PROGRAMLAMA:

UDP: istemci ile sunucu arasında “bağlantı” yoktur

- ❑ El sıkışma yoktur
 - ❑ Gönderici her pakete IP adresi ve hedef port numarasını ekler
 - ❑ Sunucu gelen paketdeki IP adresi ve portu açmak zorundadır.
- UDP: gönderilen veri sırasız bir şekilde alınabilir ya da kaybolabilir**

ÖZET:

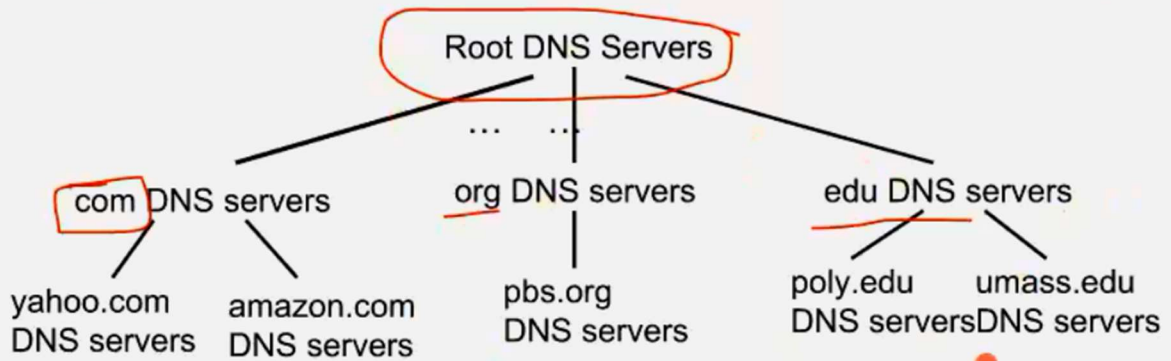
- ❑ Tipik istek/cevap mesaj değişimi:
 - ○ İstemci bilgi ya da servis ister
 - ○ Sunucu veri ya da durum kodu ile cevap verir
- ❑ Mesaj biçimleri:
 - ○ başlıklar: veril ile ilgili bilgi veren alanlar
 - ○ veri: iletilecek bilgi
- ❑ kontrol vs. veri mesajları
 - ○ hat içi, hat dışı
- ❑ merkezi vs. merkezi olmayan
- ❑ durumsuz vs. durumlu
- ❑ güvenilir vs. güvenilir olmayan mesaj transferi
- ❑ “ağ sınırında karmaşıklık”

5.HAFTA

DNS:

- Dns:uygulama katmanında çalışır.
- Alan adları(**DNS**) dağıtık bir veri yapısı kullanır.

DNS: Dağıtık hiyerarşik veritabanı



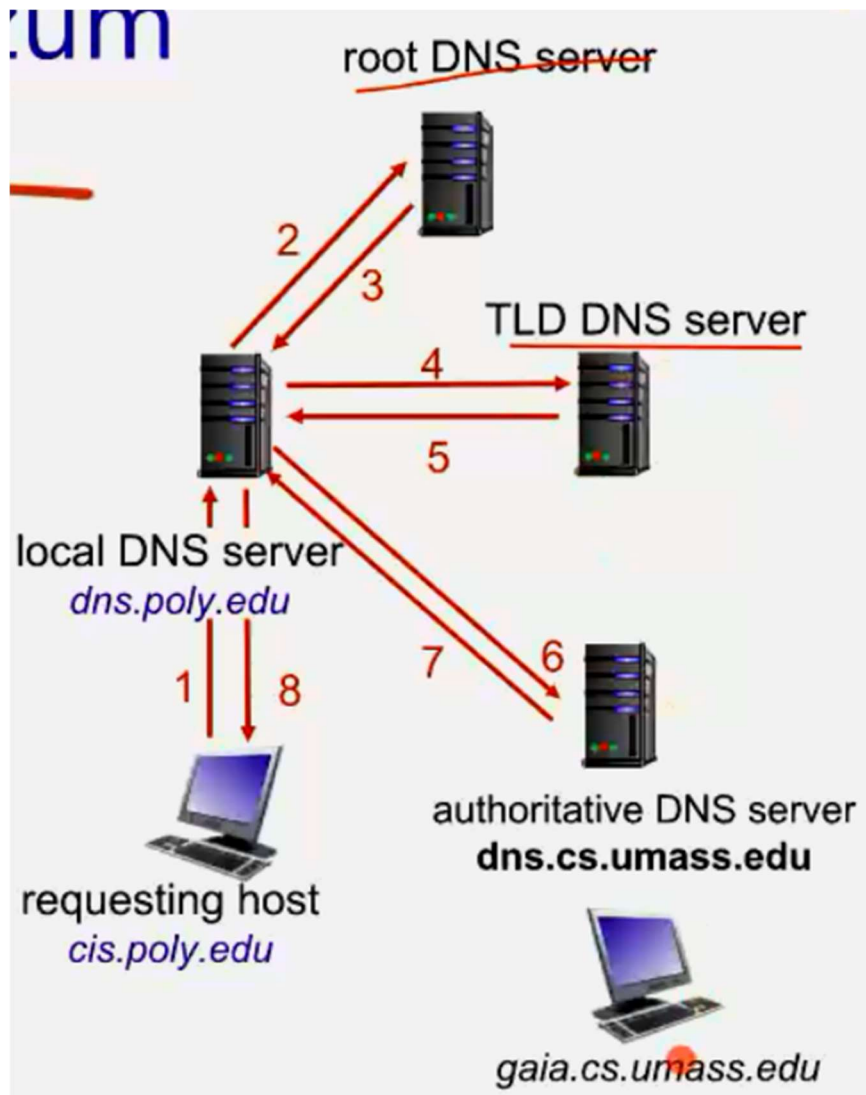
Müşteri www.amazon.com; a bağlanmak istiyor diyelim:

- ❖ Müşteri önce root sunucuya com DNS sunucu adresini sorar
- ❖ Müşteri com DNS sunucusundan amazon.com DNS sunucusunun adresini sorgular
- ❖ Müşteri DNS sunucudan www.amazon.com IP adresini öğrenir.

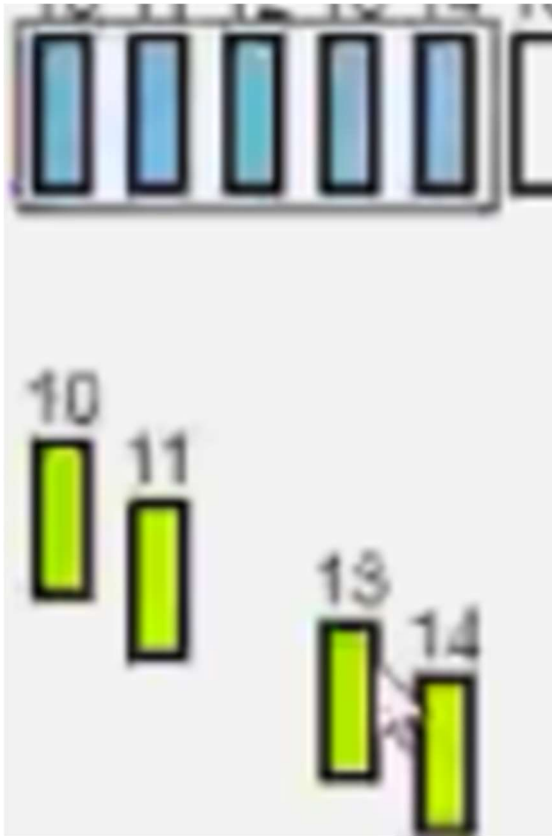
Application Layer 2-3

- 13 adet dns sunucusu = top level domains

- İTERATİF SORGU:



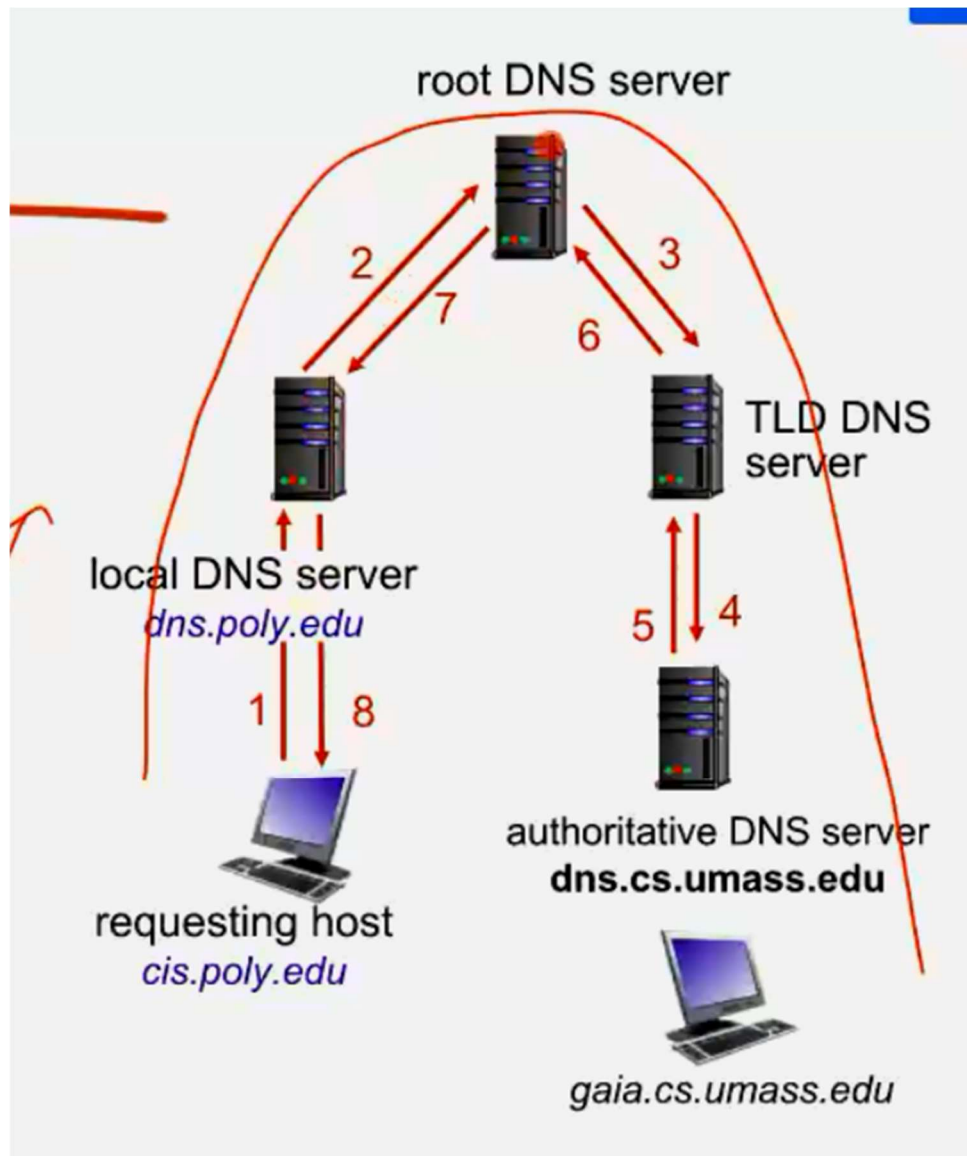
Go back Nde dönerken paket kaybolursa da mavi yanar



Selective repeatte geriye dönerken kaybolursa paket patlar



- Recursive sorgu



○ Dns Kayıtları

DNS kayıtları

DNS: kaynak kayıtlarını depolayan dağıtılmış veritabanı (RR)

RR formatı: (ad, değer, tür, ttl)

tip=A

- **isim** ana bilgisayar adı
- **değer** IP adresi

tip=NS

- **isim** etki alanıdır (örneğin, foo.com)
- **değer** bu etki alanı için yetkili ad sunucusunun ana bilgisayar adıdır

tür=CNAME

- **isim** dır-dir bazıları için takma ad “kanonik” (gerçek) isim
- **www.ibm.com** gerçek **servereast.backup2.ibm.com**
- **değer** kanonik addır

tür=MX

- **değer** ile ilişkili posta sunucusunun adıdır **isim**

P2p application

dosyalar eşler arasında paylaştırılır. Birden fazla kaynaktan bir kullanıcı, istediği dosyaları alabilir. Eşler fazlalaşırsa download etme süresi kısalır.

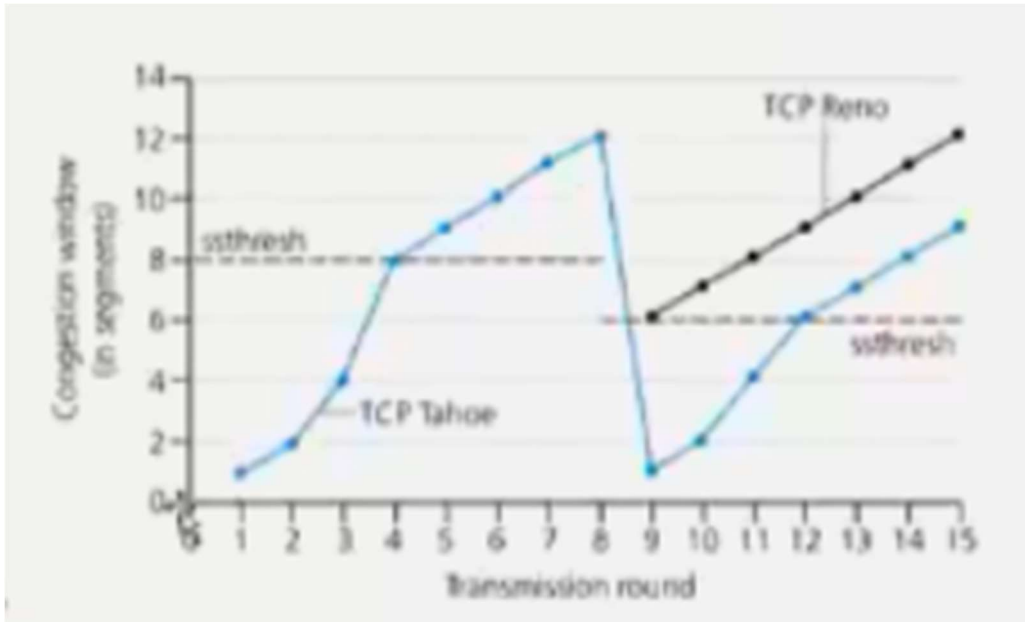
Key-value-original key şeklinde hash tablede distributed tutulur dosyanın verileri. Sebebi güvenlik.

Socket programlama

Thread denilen iplikler iler zamanda bölüşerek çalışır.

Taşıma katmanında çalışır

Tcp ve Udp de taşıma katmanına çalışır.



- 4-8 arası congestion avoidance
- 1-9 arası slow start

NETWORK KATMANI – CHAPTeR 4

Broadcast : tüm uçlara yayın

Multicast : birden çok uçlara yayın

NETWORK KATMANI: Paket hakkında karar verir. uç birimler arası routerler üzerinden iletilmesine yardımcı olur. Kendi ağı içerisinde kayıtlı değilse ilgili routerin ilgili birimine gönderir.

NAT:Adreslerin çözümlemesi

Dataplane VS Routeplane

Yavaş Hızlı

--

Routing processor bir işlem yapar switching fabrics üzerinde olur

Forward tableya aktarılarak bilgiler aktarılır

Forwarding: bilgilerin iletilmesi

--

İnternet best effort çalışır.

Atm de internet altyapısını kullanır.

Dataplane

Router plane

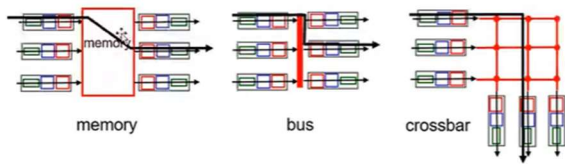
15. dakika

Meet:

Hafıza üstünden sonuçta hafızanın hızı düşük olacağından yavaş bir yöntemdir memory

Switching fabrics

- transfer packet from input buffer to appropriate output buffer
- switching rate: rate at which packets can be transfer from inputs to outputs
 - often measured as multiple of input/output line rate
 - N inputs: switching rate N times line rate desirable
- three types of switching fabrics



İkinci(bus) busta yavaş yeterli değil. Cr

Crossbar yani 3. : daha hızlı. Buffer gerekiyor.

SELF ROUTING ÖDEV

- recent recommendation: with N flows, buffering equal to

$$\frac{RTT \cdot C}{\sqrt{N}}$$

Buffer pahalı olduğu için

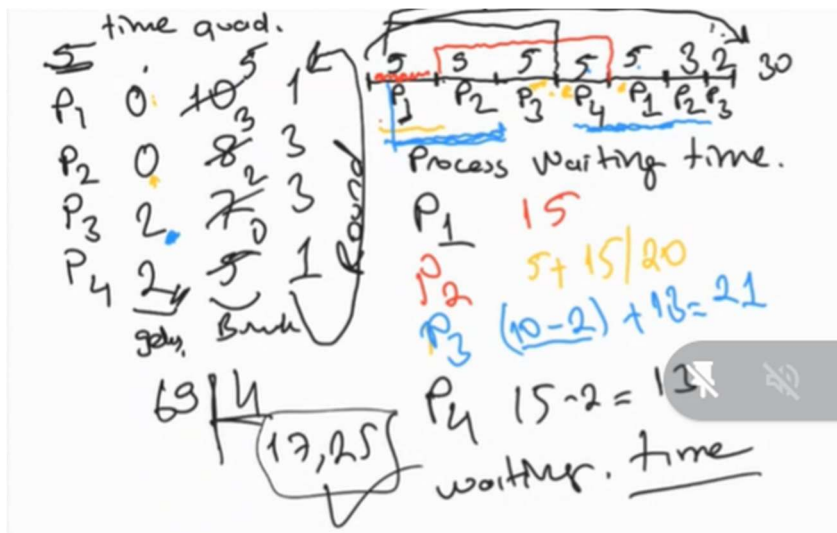
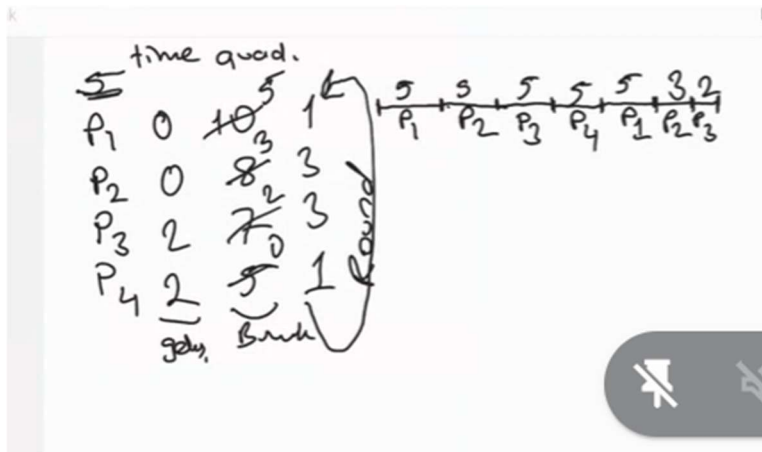
Yöntemi kullanıldı

Fifo sıkıntı olduğundan priority verilir

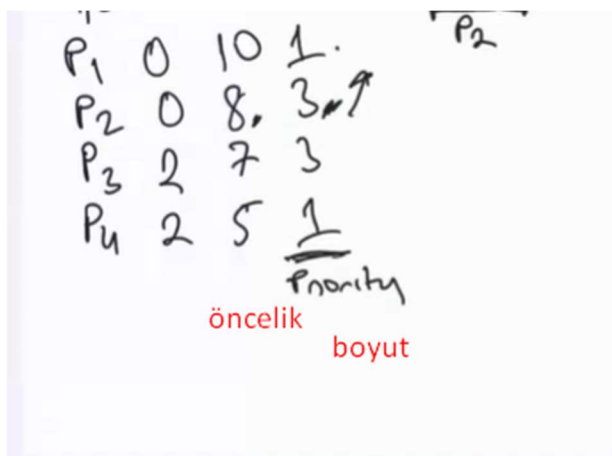
Slayt 27

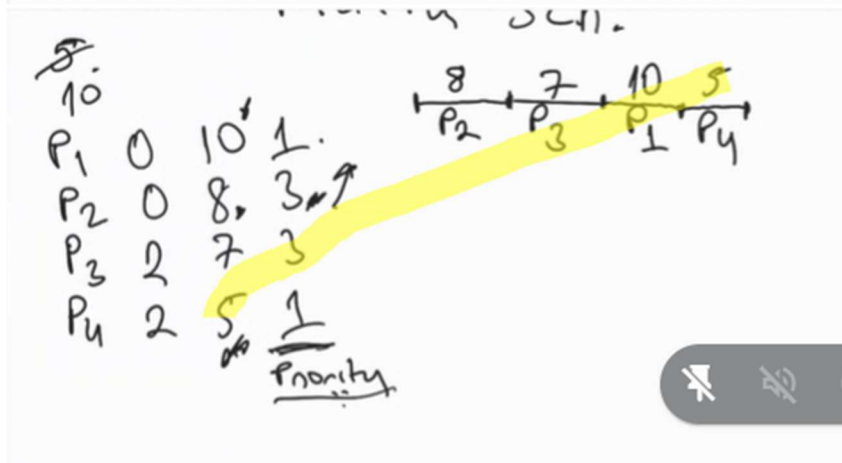
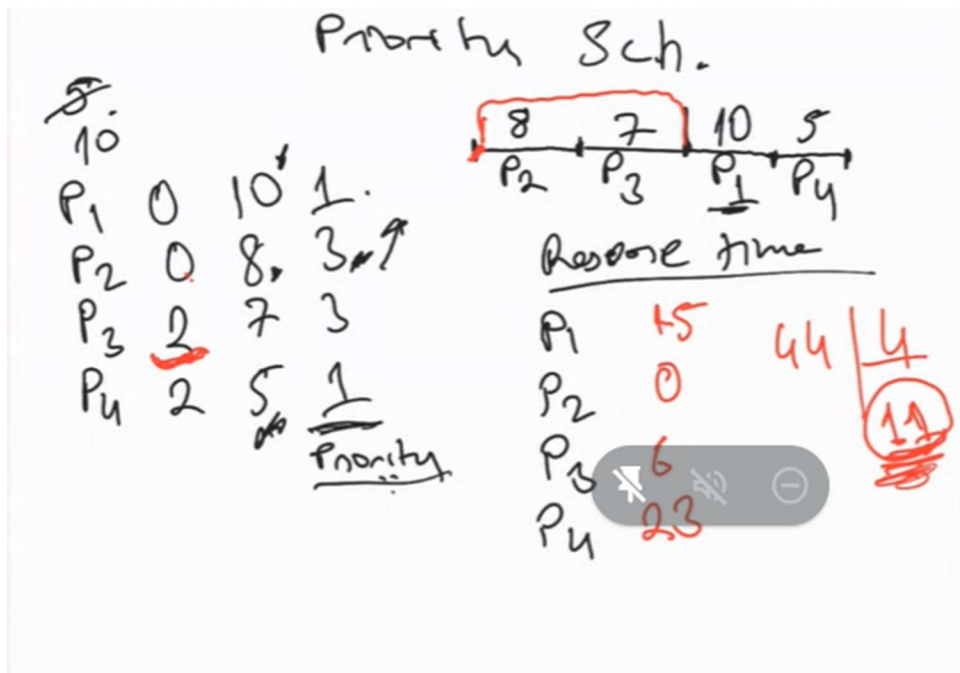
Router içinde çalışan iki konu

Rount robin:



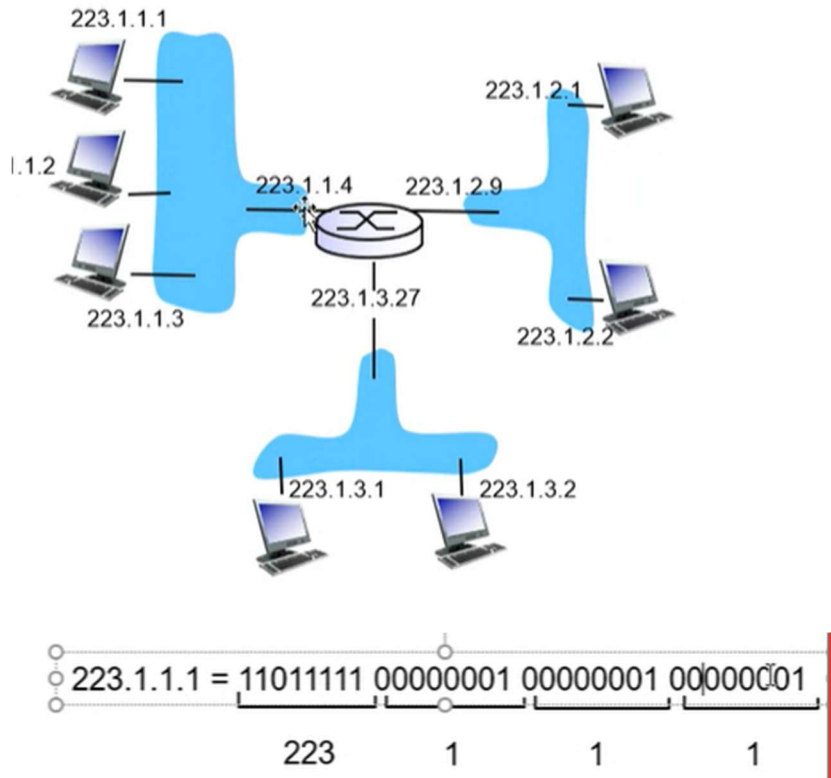
Priority scheduling





Ip V4 adressing:

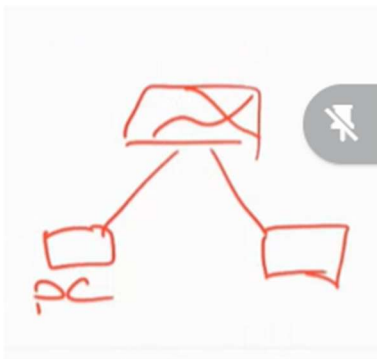
32 bitlik bir tanımlayıcıdır. 222.1.1.1



SUBNET MASK:

255.255.255.0

Ne işe yarıyor anlatacağım dendi



Bu iki bilgisayar aynı ağda mı ?

$$\begin{array}{r} 192.168.1.50/24 \\ \underline{255.255.255.0} \\ 8 \end{array} \quad \begin{array}{r} 192.168.1.109/24 \\ \underline{255.255.255.0} \\ 8 \end{array}$$

DHCP:

Dinamik host control protokol

İp dağıtır

--

Subnet hesaplama

UYGULAMA KATMANI

Ağ çekirdeğindeki araçlar için yazılım yazılmaz:

☐ Ağ çekirdeği araçları uygulama katmanında çalışmazlar

☐ Bu tasarım hızlı uygulama geliştirmeye izin verir.

İSTEMCİ-SUNUCU MİMARİSİ:

Sunucu:

- Daima açık bir ana sistem
- Sabit IP adresli
- Ana sistem kümesi –sunucu çiftliği (server farms)

İstemci:

- sunucu ile iletişim kurar
- İstemci her zaman açık olmak zorunda değildir
- Dinamik (değişebilir) IPadresine sahip olabilir
- Birbirleriyle direk olarak iletişime geçmezler

P2P MİMARİSİ:

- Her zaman açık olan sunuculara ihtiyaç yoktur

- Gelişigüzel uç sistemler birbirleriyle direk olarak iletişime geçerler
- Uç sistemler birbirleriyle ara sıra bağlanırlar ve IP adreslerini değiştirirler
- Örn: Gnutella

Büyük oranda ölçeklenebilir(scalable)

Fakat yönetilmeleri zordur

İstemci-Sunucu ve P2P Mimarisini Birleştiren Melez Mimari:

Napster:

- Dosya transferi P2P olarak gerçekleştirilir.
- Dosya arama merkezidir.
 - Eşler, içeriği merkezi sunucuya kaydeder
 - Eşler, içeriği bulmak için aynı merkezi sunucuyu sorgular

Anında Mesajlaşma:

İki kullanıcının mesajlaşması P2P dir

Varlık tespiti/bulması (presence detection/location) merkezidir :

- Kullanıcılar IP adreslerini çevrimiçi olunca merkezi sunucuya kayıt ederler.
- Kullanıcılar diğerlerinin IP adreslerini bulmak için merkezi sunucu ile iletişime geçerler.

Süreç(proses) Bir uç sistemde çalışan program.

- Aynı uç sistem üzerinde,iki süreç iç-süreç iletişimi (işletim sistemi tarafından tanımlanan) kullanarak birbirleriyle iletişime geçebilirler.
- Farklı sistemler üzerinde çalışan süreçler mesaj alış veriş ile haberleşirler

İstemci süreçleri:

İletişimi başlatan süreç.

Server süreci:

İletişime geçilmeyi bekleyen süreç.

Not: P2P mimarideki uygulamalarda hem istemci süreçleri hem de sunucu süreçleri yer alır.

Proses ve Bilgisayarlar arasındaki arabirimler:(Soketler)

Bir proses, mesajlara ağa soket adındaki yazılımlar aracılığıyla gönderir ve alır.

Soket kapı benzeşmesi:

- Gönderici proses mesajı kapıya(sokete) doğru iter
- Gönderici proses, kapının diğer tarafında mesajı hedef sürecin kapısına taşıyacak olan bir taşıma altyapısının olduğunu varsayar.

Soketlere API (Uygulama Arabirimi) de denir.

Uygulama geliřtiricileri soketin:

- uygulama tarafında her řey üzerinde kontrolleri vardır
- Tařıma tarafında ise sadece (1) tařıma protokolünün seçimi; (2) bazı parametreleri ayarlayabilme(maksimum tampon ve minimum segment boyutları gibi) üzerinde kontrolleri

Prosesleri iřaret etmek:

Bir sürecin mesaj alabilmesi için bir tanımlayıcı olması gerekir.(Identity)

Bir tanımlayıcı sistem 32 bitlik bir IP adresine sahiptir.

Süreci tanımlayabilmek için sürecin çalıştığı ana sistemin sadece IP adresini bilmesi yeterli midir?

Cevap: Hayır, aynı ana sistem üzerinde pek çok süreç çalışıyor olabilir

☐ Tanımlayıcı hem IP adresini hem de süreçle ilgili hedef bağlantı noktası numarasına (port numbers) sahiptir

☐ Örnek port numaraları

☐ HTTP sunucusu: port 80

☐ Posta sunucusu: port 25

UYGULUMA KATMANI PROTOKOLLERİ NELERİ TANIMLAR:

Karşılıklı iletilen mesajlar, örn. istek&cevap mesajları

☐ Mesajların sentaksı: mesajlardaki alanlar ve bu alanların nasıl tanımlandığı

☐ Alanların anlamları, alanlardaki bilgilerin anlamları

☐ Süreçlerin ne zaman mesajları alıp göndereceklerine dair kurallar.

Genel etki alanı Protokolleri:

RFC lerde tanımlanmıştır

☐ Birlikte işlerliğe (interoperability) olanak sağlar

☐ Örn., HTTP, SMTP

Özel Protokoller:

Örn: KaZaA

Uygulama Tarafından İhtiyaç Duyulan Tařıma Servisleri:

Veri kaybı(data loss):

Bazı uygulamalar bir miktar veri kaybını tolere edebilir (örn., ses/video)

☐ Diğerleri 100% güvenli veri transferine ihtiyaç duyarlar (örn., dosya transferi, telnet)

Zamanlama Tıming:

Bazı uygulamalar “etkili” olabilmek için az gecikme gerektirirler (örn., Internet telefonu, etkileşimli oyunlar)

Bant Genişliği(Banwidth):

Bazı uygulamalar “etkili” olabilmek için belli bir miktarda bant genişliğine ihtiyaç duyarlar (örn., multimedia)

❑ Diğerleri bant genişliğini ne kadarını kullanabilirlerse onunla idare edebilirler (“elastik uygulamalar)

Internet Tarafından Taşıma Servisleri:

TCP servisleri:

❑ Bağlantı-yönelimli (connectionoriented): istemci ve sunucu süreçler arasında bağlantı kurulumu gerekir

❑ Gönderici ve alıcı süreçler arasında güvenilir taşıma (reliable transport)

❑ Akış kontrolü (flow control): gönderici alıcıyı sıkıştırmaz

❑ Tıkanıklık kontrolü (congestion control): ağ çok yüklendiğinde göndericinin gönderdiği veri miktarını kısar

❑ Zamanlama, minimum bant genişliği garantisi sağlamaz.

UDP Servisleri:

Gönderici ve alıcı süreçler arasında güvenilir olmayan veri iletimi

❑ Bağlantı kurulumu (connection setup), güvenilirlik (reliability), akış kontrolü (flow control), tıkanıklık kontrolü (congestion control), zamanlama (timing), veya bant genişliği garantisi (bandwidth guarantee) SAĞLAMAZ

HTTP : hypertext transefer protokol

Web’in uygulama katmanı protokolüdür.

❑ İstemci/sunucu modeli

❑ istemci: Web nesnelerini isteyen, alan ve gösteren tarayıcı

❑ sunucu: isteklere karşılık olarak nesneleri gönderen web sunucusu

❑ HTTP 1.0: RFC 1945

❑ HTTP 1.1: RFC 2068

Tcp kullanır:

İstemci sunucuyla port 80’den TCP bağlantısı (socket oluşturur) başlatır

❑ Sunucu istemcinin TCP bağlantı isteğini kabul eder.

☐ HTTP mesajları (uygulama katmanı protokolü mesajları) tarayıcı (HTTP istemcisi) ve web sunucusu (http sunucusu) arasında karşılıklı gönderilir

☐ TCP bağlantısı kapatılır.

Kalıcı Olmayan http: (nonpersistent)

Genellikle bir TCP bağlantısı ile bir nesne gönderilir.

☐ HTTP/1.0 kalıcı olmayan (nonpersistent) HTTP'yi kullanır

Kalıcı Olan http: (persistent)

İstemci ile sunucu arasında tek bir TCP bağlantısı ile birden fazla nesne gönderilebilir.

☐ HTTP/1.1 varsayılan modunda kalıcı (persistent) bağlantı kullanır

Cevap Zamanı(response time):

RTT: İstemciden sunucuya giden bir paketin gidiş dönüş zamanı

Cevap zamanı (responsetime):

☐ TCP bağlantısını başlatmak için bir RTT

☐ HTTP isteği ve http cevabının ilk birkaç bitinin dönüşü için bir RTT

☐ Dosya iletim zamanı toplam= 2RTT+iletim zamanı

Kullanıcı Tarafli Etkileşim(Cerezler) Cookie

Pek çok web sitesi cookie kullanır

Dört bileşen:

1) HTTP cevap mesajındaki bir cookie başlık satırı

2) HTTP istek mesajındaki bir cookie başlık satırı

3) Kullanıcın uç sisteminde tutulan ve tarayıcı tarafından yönetilen bir cookie dosyası

4) Web sitesindeki bir arka uç veritabanı

Cookie lerin sağladıkları:

☐ Yetkilendirme(authorization)

☐ Aışverişsepeti(shopping carts)

☐ Tavsiyeler

☐ Kullanıcı oturum durumu (user sessionstate) (Web e-mail)

Web tampon belleęi (caches) (vekil sunucusu -proxy server)

Amaç: HTTP isteklerini köken Web sunucusu yerine karşılayan bir aę varlığıdır.

Kullanıcı tarayıcısını, kullanıcın tüm http isteklerini önce bu Web tampon belleęine yönlendireceęi şekilde yapılandırabilir

☐ Tarayıcı tüm http isteklerini bu tampon belleęe yönlendirir.

☐ Nesne tampon bellekte ise: tampon bellek nesneyi istemciye döndürür.

☐ Nesne tampon bellekte deęilse: tampon bellek köken sunucudan nesneyi ister ve sonrasında nesneyi istemciye döndürür.

Tampon bellek aynı anda hem sunucu hem istemci gibi davranır

☐ Genel olarak tampon bellekler bir ISP(Internet servis sağlayıcı) tarafından kurulurlar (üniversite, şirket, yerleşim yeri ISP'si)

Neden web tampon belleęi?

☐ Bir istemci isteęi için cevapzamanını azaltır

☐ Kurumun erişim hattındaki trafięi azaltır.

☐ Tampon belleklerin yoğun olduęu bir Internet "zayıf" içerik sağlayıcıların içerięi daha iyi sunmalarını sağlar (fakat aynısını P2P dosya paylaşımı da yapabilir)

Tampon bellek kurarsak

☐ Hit alma oranı .4 varsayılrısa

Sonuçları

☐ İsteklerin 40% 'ı neredeyse hemen karşılanır

☐ İsteklerin 60% 'ı köken sunucular tarafından karşılanır

☐ Erişim hattının kullanılması 60 % 'a azaltıldı, önemsiz gecikmeler olabilir (10 msn kadar)

☐ Toplam ort. gecikme = Internet gecikmesi + erişim gecikmesi + LAN gecikmesi

Amaç: eęer tampon bellek güncel sürüme sahipse nesneyi gönderme

☐ Tampon bellek: HTTP isteęi içerisinde belleęe alınmış kopyanın tarihini belirler

If-modified-since:

<date>

☐ sunucu: eęer tampondaki kopya güncelse cevapta nesne bulunmaz:

HTTP/1.0 304 Not Modified

Dosya transfer protokolü :FTP (file transfer protocol)

Uzak ana sistem e/den dosya transferi

- ☐ İstemci sunucu modeli
- ☐ *istemci*: transferi başlatan taraf
- ☐ *sunucu*: uzaktaki ana sistem
- ☐ ftp: RFC 959
- ☐ ftp sunucus: port 21

FTP: ayrı kontrol ve veri bağlantıları

FTP istemcisi FTP sunucusu ile TCP'yi taşıma protokolü olarak kullanarak port 21 üzerinden iletişim kurar

- ☐ İstemci kontrol bağlantısı üzerinden yetkilendirme alır.
- ☐ İstemci uzaktaki dizinleri kontrol bağlantısı üzerinden gönderdiği komutlar aracılığıyla tarar.
- ☐ Sunucu dosya transferi ile ilgili bir komut aldığı anda, sunucu istemciye TCP veri bağlantısı açar.
- ☐ Dosyayı transfer ettikten sonra sunucu bağlantıyı kapatır.

Sunucu başka bir dosya transfer edilmek istenirse yeni bir TCP veri bağlantısı açar.

- ☐ Kontrol bağlantısı: “bant dışı (out of band)”
- ☐ FTP sunucusu “durum” korumalıdır: halihazırdaki dizin, önceki yetkilendirmeler

Sayfa 46 Ftp komutları cevapları

ELEKTRONİK POSTA:

Üç ana bileşen

- ☐ Kullanıcı temsilcileri (user agents)
- ☐ Posta sunucuları (mail servers)
- ☐ Basit posta transfer protokolü(simple mail transfer protocol:SMTP)

User Agent(Kullanıcı aracı)

- ☐ a.k.a. “posta okuyucuları”
- ☐ Mesaj okuma, cevap verme, iletme, kaydetme ve oluşturma
- ☐ e.g., Eudora, Outlook, elm, Netscape Messenger
- ☐ Sunucuda depolanan gelen ve giden mesajlar

ELEKTRONİK POSTA:POSTA SUNUCUSU:

Posta Sunucuları

- ☐ Posta kutusu kullanıcıya gelen mesajları içerir.
- ☐ Mesaj kuyruğu giden (gönderilecek) mesajları içerir
- ☐ SMTP protokolü posta sunucuları arasında posta göndermek için kullanılır

- istemci: gönderen posta sunucusu
- “sunucu”: alıcı posta sunucusu

Elektronik Posta: SMTP [RFC 2821]

☐ İstemciden sunucuya eposta mesajlarını güvenilir olarak transfer etmek için port 25 üzerinden TCP kullanır

☐ Direk transfer: gönderen sunucudan alıcı sunucuya

☐ Üç fazlı transfer

- ☐ El sıkışma (greeting)
- ☐ Mesajın iletimi
- ☐ Kapatma (closure)

☐ Komut /cevap etkileşimi (command/response interaction)

- ☐ komutlar: ASCII metin
- ☐ cevaplar: durum kodları ve ilgili deyimler

☐ Mesajlar 7 bit’lik ASCII metinleridir.

Senaryo: Alice Bob’a mesaj gönderir

- 1) Alice kullanıcı temsilcisi kullanarak mesaj gönderir ve “to” satırına bob@someschool.edu yazar
- 2) Alice’in kullanıcı temsilcisi mesajı kendi posta sunucusuna gönderir; mesaj mesaj kuyruğuna konur
- 3) SMTP’nin istemci tarafı Bob’un posta sunucusuna TCP bağlantısı açar
- 4) SMTP istemcisi Alice’in mesajını TCP bağlantısı üzerinden gönderir.
- 5) Bob’un posta sunucusu mesajı Bob’un posta kutusuna koyar
- 6) Bob mesajı okumak için kendi kullanıcı temsilcisini çalıştırır.

SMTP SON NOTLAR:

SMTP kalıcı (persistent) bağlantı kullanır

☐ SMTP mesajları 7 bit’lik ASCII metinden oluşur (header & body)

☐ SMTP sunucusu msaj sonlarını belirtmek için CRLF.CRLF kullanır

CR : carriage return

LF : line feed

SMTP HTML ARASINDAKİ FARKLAR:

HTTP: çekme (pull)

☐ SMTP: itme (push)

☐ Her ikisi de etkileşim için ASCII komut ve cevaplarını, durum kodlarını kullanırlar

☐ HTTP: her nesne kendi cevap mesajı içerisine sarmalanır

☐ SMTP: tüm nesneler bir mesaj içerisine yerleştirilir.

☐ HTTP bant içi SMTP bant dışı kontrolü kullanır

SMTP:

eposta mesajlarının gönderimi için bir protokoldür

RFC 822: metin mesaj biçimini belirleyen standarttır:

- ☐ Başlık satırları, e.g.,
- ☐ To:
- ☐ From:
- ☐ Subject:
- SMTP komutlarından farklıdır !

☐ gövde

- ☐ Sadece ASCII karakterlerden oluşan “mesaj”

Mesaj biçimi: multimedia için

MIME: çok amaçlı Internet posta uzantıları (multi purpose mail extension), RFC 2045, 2056

☐ MIME içerik tipini tanımlamak için mesaj başlığına ek başlıklar eklenmelidir

Posta erişim protokolleri

☐ SMTP: alıcı sunucuya gönderme/depolama ile ilgilidir

☐ Posta erişim protokolleri: sunucudan alma ile ilgilidir

- ☐ POP: Post Office Protocol [RFC 1939]
 - yetkilendirme (agent <-->server) ve download
- ☐ IMAP: Internet Mail Access Protocol [RFC 1730]
 - Daha fazla özellik (daha karmaşık)
 - Sunucuda depolanan mesajların idaresi
- ☐ HTTP: Hotmail , Yahoo! Mail, etc.

DNS: Internet'in Dizin Servisi

İnsanlar: pek çok tanımlayıcı:

- SSN, isim, pasaport #

Internet ana sistemleri, yönlendiriciler:

- IP adreslerini (32 bit) - datagram ların adreslenmesi için kullanılırlar
- İnsanlar "isim", e.g., ww.yahoo.com - kullanırlar

Q: IP adresleri ile isimler arasında eşleşme nasıl sağlanır?

Etki Alanı Ad Sistemi

(Domain Name System):

- Bir *DNS sunucu* hiyerarşisi içerisinde uygulanan *dağıtık bir veritabanı (distributed database)*
- Ana sistemlerin dağıtık veritabanı sorgulamasını sağlayan bir *uygulama katmanı protokolüdür.*
- not: ana Internet fonksiyonu, uygulama katmanı protokolü olarak uygulanır
- complexity at network's "edge"

DNS servisleri

- Ana sistem isimleri-IP adres çevirisi
- Ana sistem lakapları(aliasing)
 - □ Kurallı (canonical) ve lakap (alias) adları
- Posta sunucusu lakapları
- Yük dağıtımı
 - □ Çoğaltılan web sunucuları için bir IP adresi seti, bir kurallı isimle ilişkilendirilir

Neden DNS merkezileştirilmiyor?

- Bir tek başarısızlık noktası
- Trafik hacmi
- Uzakta merkezileştirilmiş veritabanı
- bakımölçeklenemez!

İstemci www.amazon.com için IP istiyor; 1inci tahmin:

- ❓ İstemci com DNS sunucusunu bulmak için kök dizin sorgusu yapar
- ❓ İstemci com DNS sunucusunu amazon.com DNS sunucusunu bulmak için sorgular
- ❓ İstemci amazon.com DNS sunucusunu www.amazon.com un IP adresini almak için sorgular

Üst-seviye etki alanı (TLD) sunucuları: com, org, net,

edu, gibi, ve tüm üst düzey ülke etki alanlarından uk, fr, ca, jp, gibi sorumludurlar

❓ Network solutions com TLD sunucularını işletmektedir

❓ Educause edu TLD sunucularını işletmektedir

Yetkili DNS sunucuları: kurumların DNS sunucuları,diğer kurum sunucularına yetkili alan adı ve IP eşleşmesi sunarlar (e.g., Web ve posta).

❓ Kurum tarafından ya da servis sağlayıcı tarafından işletilebilirler

YEREL DNS SUNUCLARI:

❓ Katı bir şekilde hiyerarşiye ait değildir

- ❓ Her ISP'ye ait bir tane vardır (yerel ISP, şirket, üniversite).
- ❓ “default name server”

❓ Bir ana sistem DNS sorgusu yaptığında, sorgu kendi yerel DNS sunucuna gönderilir

- ❓ Vekil (proxy) gibi davranır, sorguyu hiyerarşiye gönderir.

Tekrarlanan sorgu(iterated query)

❓ Başvurulan sunucu başvurulacak sunucu adıyla cevap verir

❓ “Ben bilmiyorum ama şu sunucuya sorabilirsin”

Yinelenen Sorgu(reursive query)

❓ İsim eşleştirmenin zorluğunu başvuru sunucuya yükler

❓ ağır yük?

Dns Tampon Bellek ve Kayıtları Güncelleme:

Bir sunucu eşleştirmeyi öğrendiğinde,eşleştirmeyi tampon belleğe alır

- ❓ Tampon bellek girdileri belli bir süre sonrakaybolurlar (timeout)(TTL)
- ❓ TLD sunucuları tipik olarak yerel DNSsunucularının tampon belleğinde yer alırlar
 - Bu nedenle kök DNS sunucularına sıklıkla başvurulmaz

❓ Güncelleme/uyarı mekanizması IETF'nintasarımındadır

- ❓ RFC 2136
- ❓ <http://www.ietf.org/html.charters/dnsind-charter.html>

DNS KAYITLARI:

DNS: kaynak kayıtlarını (resource records (RR)) depolayan dağıtık veritabanı

Type=A

- name sistem adıdır
- value IP adresineeşleme sunar

Type=NS

- name etki alanıdır (e.g. foo.com)

- value bu etki alanına ait yetkili DNS 'in IP sine eşleme sunar

Type=CNAME

- name lakab ana sistem adıdır www.ibm.com
aslında servereast.backup2.ibm.com
- value kurallı ana sistem adıdır

Type=MX

- Value, name ana sistemi lakabına sahip olan bir posta sunucusunun kurallı adıdır.

DNS Mesajları:

DNS protokolü : sorgu ve cevap mesajları, aynı mesaj biçimindedir.

msg başlığı

☐ Tanımlama (identification): sorgu için 16 bit lik alan

kullanılır ,

☐ Bayraklar (flags):

- ☐ Sorgu veya cevap
- ☐ İstenen yineleme
- ☐ Mevcut yineleme
- ☐ Cevap yetkilidir

DNS veritabanına kayıt girmek

Örnek: "Network Utopia" yı yeni oluşturduk

☐ networkutopia.com adını **kaydedici (registrar)** 'ye kayıt ettiririz. (örn, Network Solutions)

- ○ Kaydediciye yetkili DNS sunucumuzun adı ve IP adresini de vermek gerekir (birincil ve ikincil)
- ○ Kaydedici TLD sunucusuna iki RR ekler:

(networkutopia.com, dns1.networkutopia.com, NS)

(dns1.networkutopia.com, 212.212.212.1, A)

☐ Yetkili sunucuya www.networkutopia.com için Type A kaydı ve mail.networkutopia.com için Type MX kaydı konulur

☐ **İnsanlar sizin sitenizin IP adresine nasıl erişecekler?**

SOKET PROGRAMLAMA:

Amaç: socketleri kullanarak iletişim kuran bir istemci/sunucu uygulamasının nasıl yapılacağını öğrenmek.

Soket API'si

- ❑ 1981'de BSD4.1 UNIX ile tanıtıldı
- ❑ Uygulamalar tarafından yaratıldı, kullanıldı ve sunuldu
- ❑ İstemci/sunucu paradigması
- ❑ Soket API'si ile iki taşıma hizmeti:
 - Güvenilir olmayan datagram
 - güvenilir, byte stream yönelimli

TCP İLE SOKET PROGRAMLAMA:

Soket: uygulama süreci ve uçtan uca taşıma protokolü (UDP veya TCP) arasında bir kapı

TCP servisi: bir süreçten diğerine **bitlerin** güvenli iletimi

İstemci sunucu ile iletişime geçmeli

- ❑ Sunucu süreci öncelikle çalışır olmalı
- ❑ Sunucu istemcinin iletişimini karşılayacak soket (kapı) oluşturmali

İstemci sunucuyla:

- ❑ İstemciye özel TCP soketi oluşturarak
- ❑ Sunucu sürecin IP adresini ve port numarasını belirleyerek iletişim kurar.
- ❑ **İstemci soket oluşturduğunda:**

İstemci TCP sunucu TCP si ile bağlantı kurmuş olur

İstemci tarafından iletişime geçen **sunucu TCP si** sunucu sürecinin istemciyle haberleşmesi için **yeni bir soket oluşturur. server TCP creates new socket**

- ○ Sunucunun birden fazla istemci ile konuşmasını sağlar
- ○ Kaynak port numaraları istemcilerin ayırt edilmesini sağlar

AKIM (STREAM) JARGON:

- ❑ **Akım (stream)** sürecin içine ya da içinden akan karakterler sırasıdır.
- ❑ **Girdi akımı (input stream)** süreç için bir girdi kaynağına bağlıdır, örn, klavye veya soket.
- ❑ **Çıktı akımı (output stream)** bir çıktı kaynağına bağlıdır, eg, monitor veya soket.

Örnek istemci/sunucu uygulaması

- 1) İstemci standart girdi den satır okur (inFromUser stream) , soket aracılığıyla sunucuya gönderir (outToServer stream)
- 2) Sunucu soketten satırı okur
- 3) Sunucu satırı büyük harfe çevirir, istemciye geri gönderir
- 4) İstemci okur, soketteki değişen satırı çıktı alır (inFromServer stream)

UDP İLE SOKET PROGRAMLAMA:

UDP: istemci ile sunucu arasında “bağlantı” yoktur

- ❑ El sıkışma yoktur
- ❑ Gönderici her pakete IP adresi ve hedef port numarasını ekler
- ❑ Sunucu gelen paketdeki IP adresi ve portu açmak zorundadır.

UDP: gönderilen veri sırasız bir şekilde alınabilir ya da kaybolabilir

ÖZET:

- ❑ Tipik istek/cevap mesaj değişimi:
 - ○ İstemci bilgi ya da servis ister
 - ○ Sunucu veri ya da durum kodu ile cevap verir
- ❑ Mesaj biçimleri:
 - ○ başlıklar: veril ile ilgili bilgi veren alanlar
 - ○ veri: iletilecek bilgi
- ❑ kontrol vs. veri mesajları
 - ○ hat içi, hat dışı
- ❑ merkezi vs. merkezi olmayan
- ❑ durumsuz vs. durumlu
- ❑ güvenilir vs. güvenilir olmayan mesaj transferi
- ❑ "ağ sınırında karmaşıklık"

IANA, ip adreslerini ve otonom sistem numaralarını dağıtan bir kuruluştur. Bu kurulu, 1998 tarihinden itibaren, ticari tescili, ICANN (Internet Corporation for Assigned Names and Numbers) şirketinde yürütmektedir. Alan Adları, ağaç yapısı kurallarına göre dallanan bir yapıda kullanılmaktadır. Amerika haricinde, internete bağlı olan tüm ülkelerdeki adresler o ülkenin ISO3166 ülke kodu ile bitmektedir (TLD, top level domain). Türkiye için bu kod TUR`dur. ICANN tarafından yetkilendirilen alan adı tescil yöneticileri, bugün geçerli olan kuralları gözeterek, internet yararına, tescil işlemlerini yürütmekle sorumludurlar.

Soket numaralarıyla iletim sağlanır: iletim katmanı bu iş görür.

Udp best effort çalışıyor. Paketler kaybolursa bir şekilde telafi etmeye çalışıyor

Tcp daha güvenli

Udp kullanan protokoller: snmp dns

Güvenilir veri transferi ağ ile ilgili en önemli 10 konudan biridir.

Rdt1.0 da güvenilir kanal kullanılır. **Unreliable_chanel()**e uğramaz

Rdt2.0: paket bozulduysa NAK gönderir paket bozulmadıysa iletir ve karşı tarafa acknowledgement(ack) iletir.Sürekli beklemeler olduğu için performansı düşük

Rdt2.1 ile bu sorunun çözümü sıra numarası eklenir. Ack bozulursa ek tedbir alınmış olur. 1, 2, 3, 4 şeklinde daireler vardı. Ack gelir sorun çıkmazsa 1. Paketi gönderir.

2.1 de karmaşıklık vardı. 4 ayrı durum içinde kod yazmamız gerektiğinden karmaşık bir işlem oluşur.

2.2 : karmaşıklık bir nebze aşılmıştır. 2.1e göre Nak kullanılmamıştır. Zaman burada kötü kullanılmıştır. Timer yok

Rdt 3.0 ile Bekleme süresi hesaba katılmıştır. Timer eklenmiştir. Verimsizdir. RTT aralığında bir tane paket gönderiliyor. Verimsizdir. Çözüm önerileri geliştirilmiştir. Pipelining birden fazla dosyayı

aynanda gönderilmek fikri ortaya atılmış, zamandan tasarruf edilmiştir. Pakaet kaybolunca gidince, dönünce durumlarını incelediğimiz olay..Selective repeat vb

Piplined protocols: Tek bir hat üzerine birden çok verinin gönderilmesi.

TCP fast retransmint: triplle ack gelirse (aynı paekte ayıt), o paketi hemen geri gönderiyordu

Relaibledata transfer karşı tarafa sıkıntısız bir şekilde veriyi iletilmeini sağlar. Sonlu durum daireleri ile çözüyorduk (udt_send(),extract(packet,data)).

Acknowledgements ile sağlanır -> ben bu gönderilen paketi aldım bilgisini bize verir.

İki yönlü ack handshaking sıkıntılı. Üç yönlü anlatıldı.

Sıkışıklık kontrolündeki temel sorun: bir ağda çok fazla kaynak var. Hepsini göndermeye çalışınca bir sıkışıklık ortaya çıkar. **Bu ağ ile ilgili en önemli 10 problemten biridir.**

Paketleri zamanından iletemezse routerlerdeki bufferlar dolar, sistem yavaşlar. Buffer sonsuz olsa sorun olmaz.

Buffer olayları:

TCP congestion control(sıkışıklık kontrolü):

Tcp nin sıkışıklığı azalmak için yöntemi: toplamsal artırım çarpımsal azaltım. **Kaşıkla verdim kepçeyle geri aldım ☹️. Paket kaybı sinyali gelene kadar y yi arttırır. Geldiğinde ise yarıya düşürür.**

TCP de bu olaylar nasıl gerçekleşiyor?

Tcp slowstart: Clientler anlaştıktan sonra paket gider

Cwnd(cong. Window) = 1mss

32 den 16 ya düşürdüysek tcpyi, triple acknowledgment mı timeout mu sebep olur?

Triple ack olur.(

32 den 1 e düşseydi time out olurdu(thoma)

TCP:detecting, reacting to loss

Eğer tcpde kayıp varsa-> geribildirim yok, timeout

Timeout olursa segment daha az olması için pencere boyutunu küçültür.(geçen haftaki animasyon)

Sstresh(eşik seviyesi) sınırından sonra cwnd birer birer artar.

Grafikte 12 yi geçtiğinde triple ack gelirse 6 y düşürür. Tcp Reno gelirse triple veya timeout olması önemli olmadan 1e gönderir

Multimedia izleyen kullanıcı, Bantı genişliğini sömürür.

Oturum 2:

NETWORK LAYER:

Uç birim iletim yapmak isterse gerçekleştirir. Routerler arasında dolaşır, iletilmesi gereken uç yöne bağlanmaya çalışır. Eğer gittiği router gitmek istediği routerde değilse, router diğer routerlere ileterek, hedef bilgisayar ve routeri bulur ve iletir.

Network layerin 2 önemli görevi yani fonksyonu vardır.

Routerlerdeki ilgili yönlendirme tablosuna bakarak diğer routere iletim sağlanır. Kontrol tablosunu doldurma işi control plane katmanı ile gerçekleştirilir.

Tcp taho(tcp ssthresh) : üçlü ack ta gelse timeoutta gelse hemen 1 e düşürür.

Forward tableya aktarılarak bilgiler aktarılır

--

İnternet best effort çalışır.

Atm de internet altyapısını kullanır.

Dataplane

Router plane

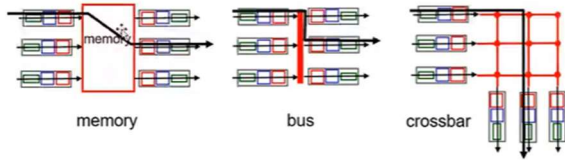
15. dakika

Meet:

Hafıza üstünden sonuçta hafızanın hızı düşük olacağından yavaş bir yöntemdir memory

Switching fabrics

- transfer packet from input buffer to appropriate output buffer
- switching rate: rate at which packets can be transfer from inputs to outputs
 - often measured as multiple of input/output line rate
 - N inputs: switching rate N times line rate desirable
- three types of switching fabrics



İkinci(bus) busta yavaş yeterli değil. Cr

Crossbar yani 3. : daha hızlı. Buffer gerekiyor.

SELF ROUTİNG ÖDEV

- recent recommendation: with N flows, buffering equal to

$$\frac{RTT \cdot C}{\sqrt{N}}$$

Buffer pahalı olduğu için

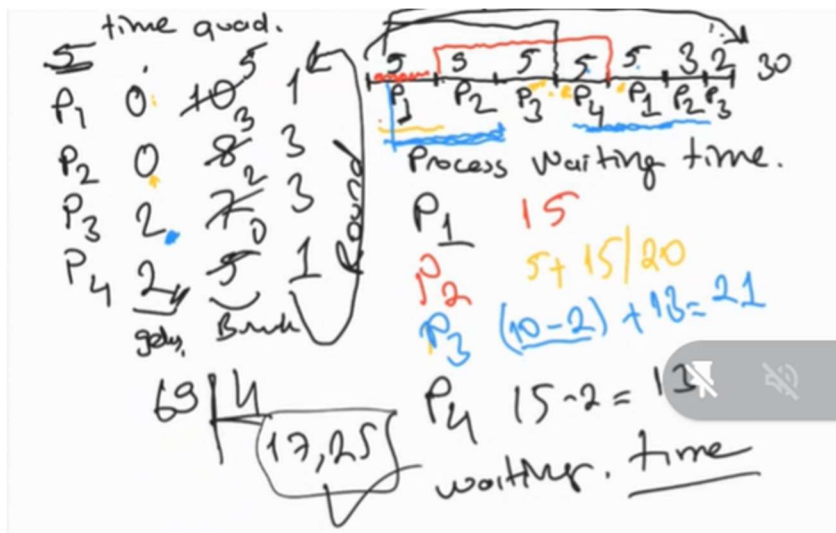
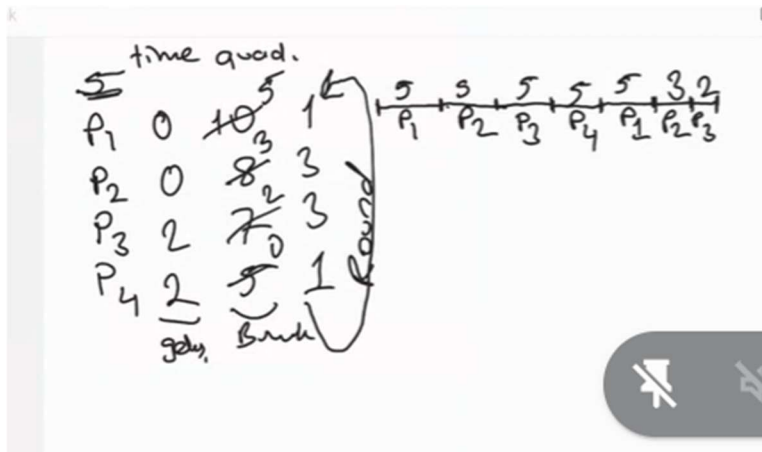
Yöntemi kullanıldı

Fifo sıkıntı olduğundan priority verilir

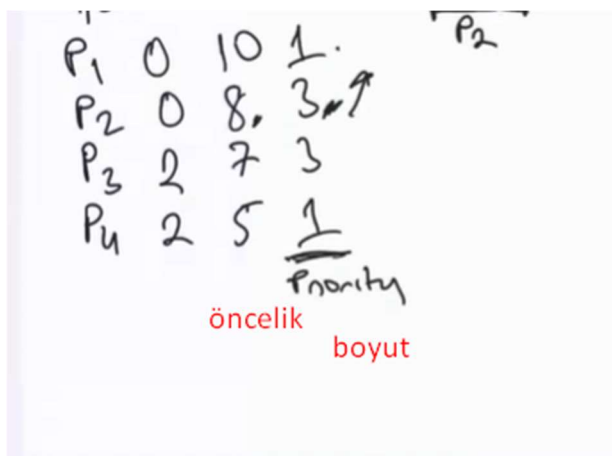
Slayt 27

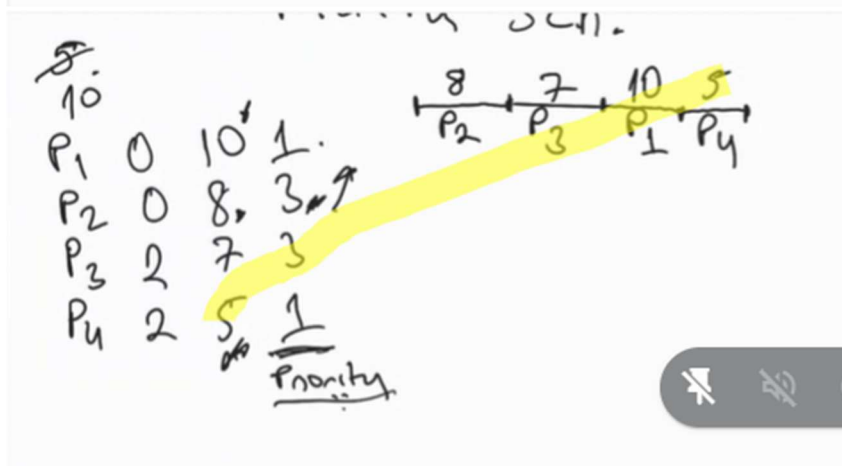
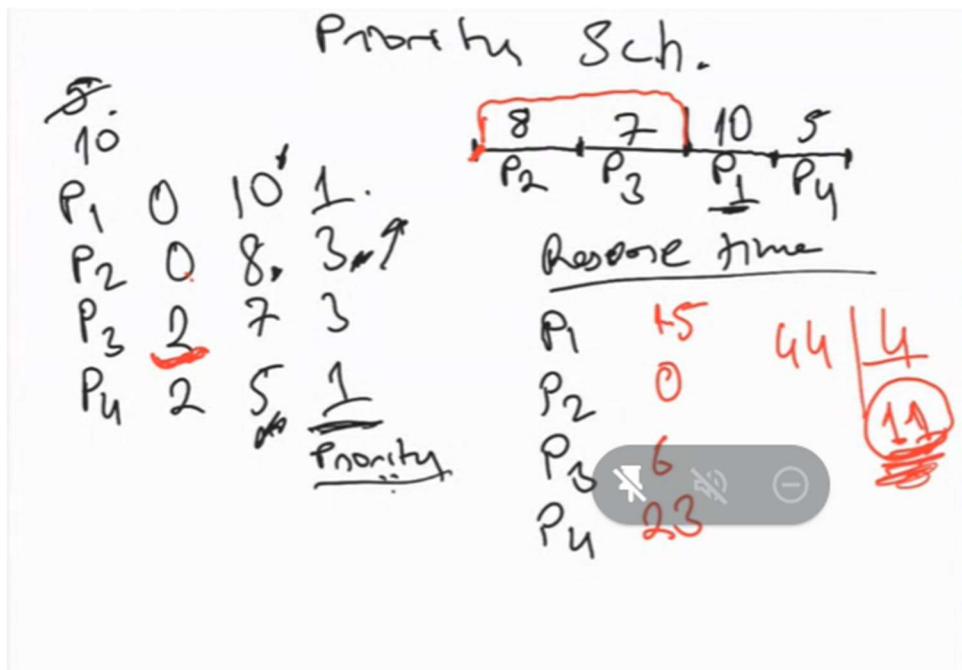
Router içinde çalışan iki konu

Rount robin:



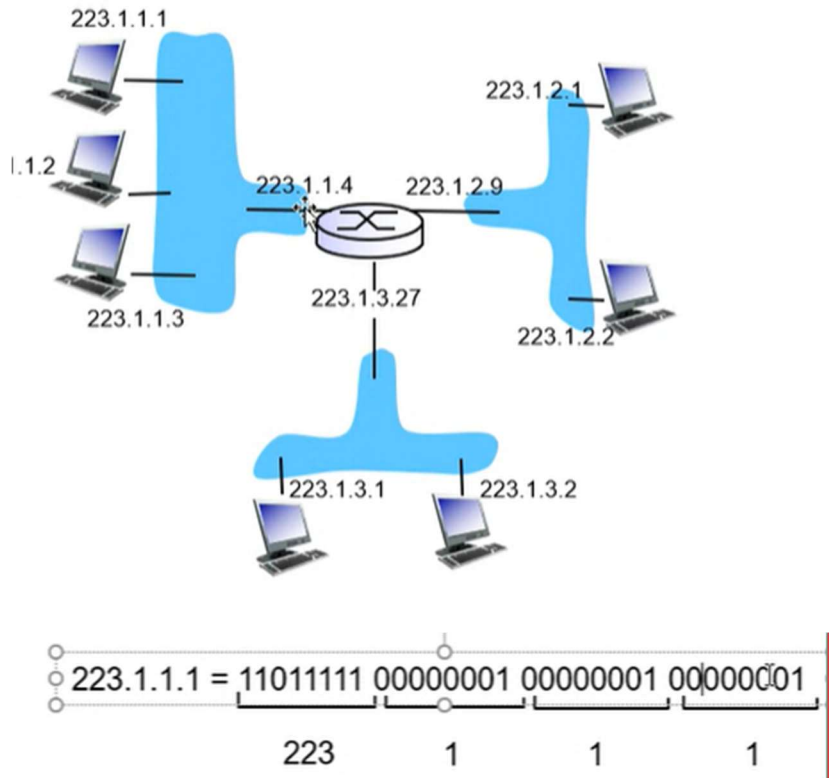
Priority scheduling





Ip V4 adressing:

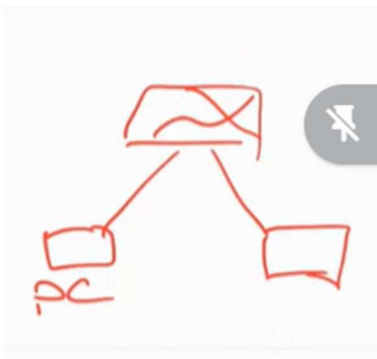
32 bitlik bir tanımlayıcıdır. 222.1.1.1



SUBNET MASK:

255.255.255.0

Ne işe yarıyor anlatacağım dendi



Bu iki bilgisayar aynı ağda mı ?

$$\begin{array}{r} 192.168.1.50/24 \\ \underline{255.255.255.0} \\ 8 \end{array} \quad \begin{array}{r} 192.168.1.109/24 \\ \underline{255.255.255.0} \\ 8 \end{array}$$

DHCP:

Dinamik host control protokol

İp dağıtır

--

Subnet hesaplama

Forward tableya aktarılarak bilgiler aktarılır

--

İnternet best effort çalışır.

Atm de internet altyapısını kullanır.

Dataplane

Router plane

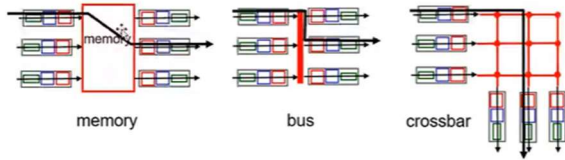
15. dakika

Meet:

Hafıza üstünden sonuçta hafızanın hızı düşük olacağından yavaş bir yöntemdir memory

Switching fabrics

- transfer packet from input buffer to appropriate output buffer
- switching rate: rate at which packets can be transfer from inputs to outputs
 - often measured as multiple of input/output line rate
 - N inputs: switching rate N times line rate desirable
- three types of switching fabrics



İkinci(bus) busta yavaş yeterli değil. Cr

Crossbar yani 3. : daha hızlı. Buffer gerekiyor.

SELF ROUTİNG ÖDEV

- recent recommendation: with N flows, buffering equal to

$$\frac{RTT \cdot C}{\sqrt{N}}$$

Buffer pahalı olduğu için

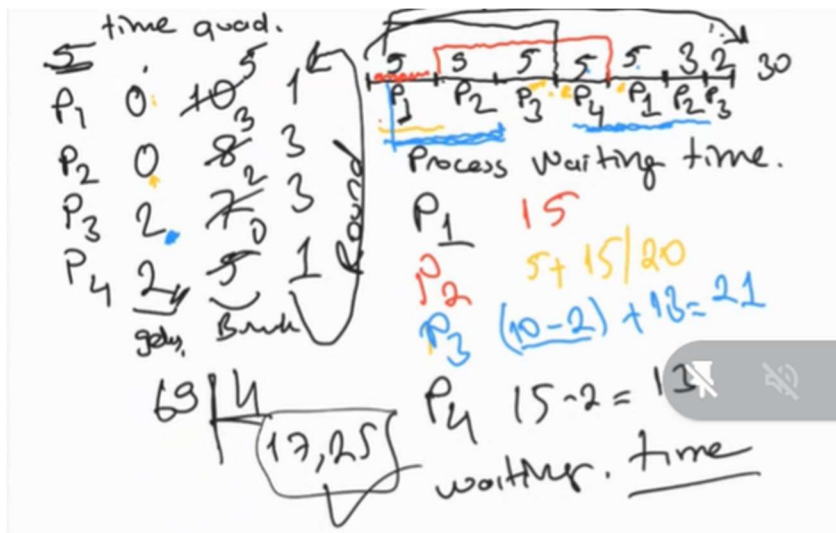
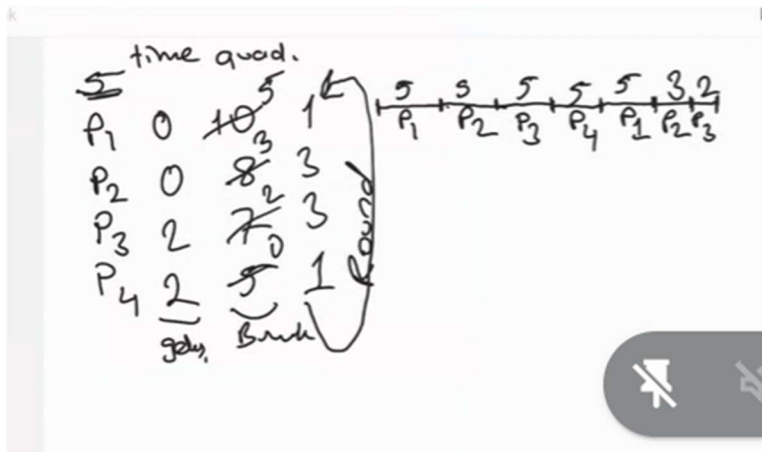
Yöntemi kullanıldı

Fifo sıkıntı olduğundan priority verilir

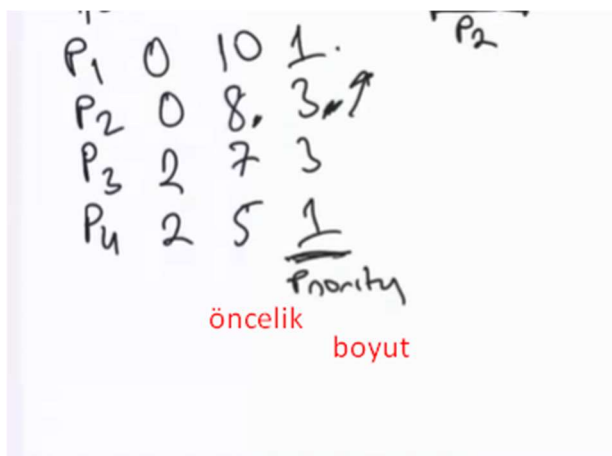
Slayt 27

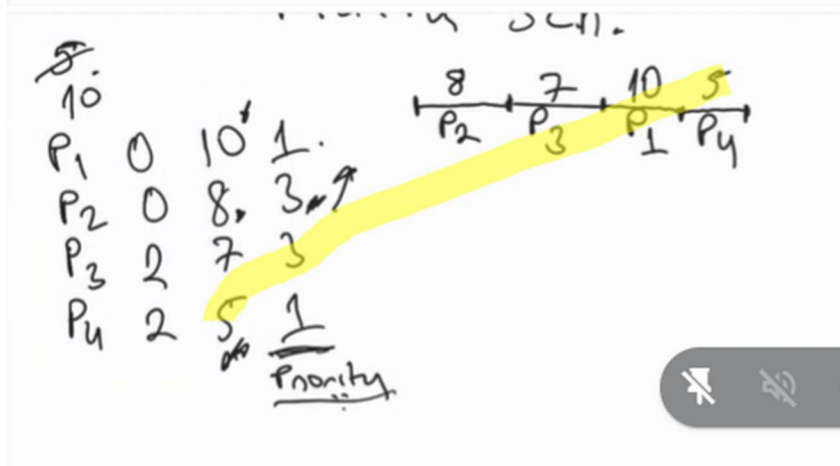
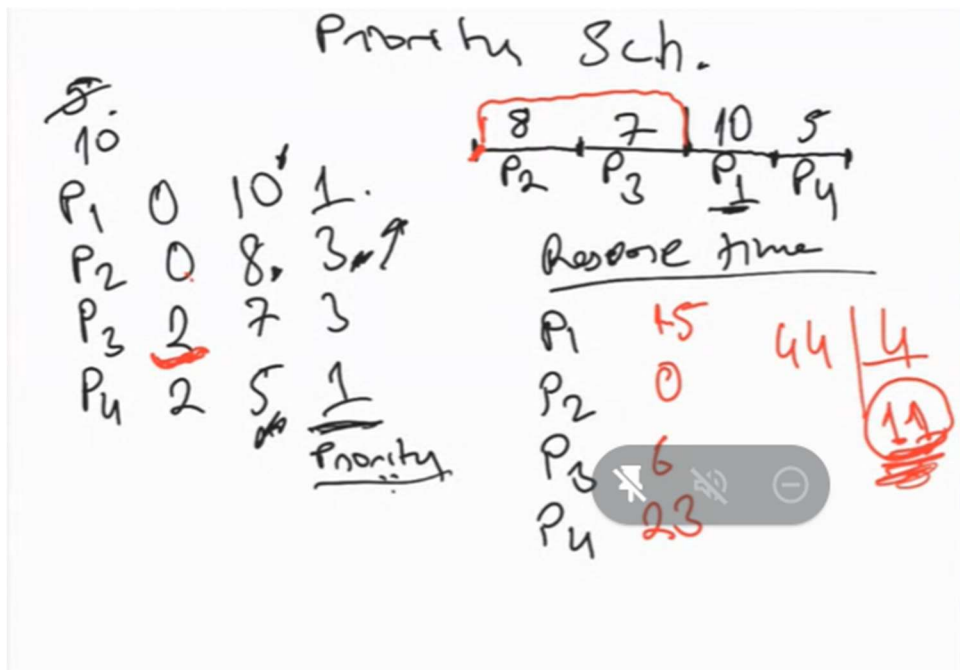
Router içinde çalışan iki konu

Rount robin:



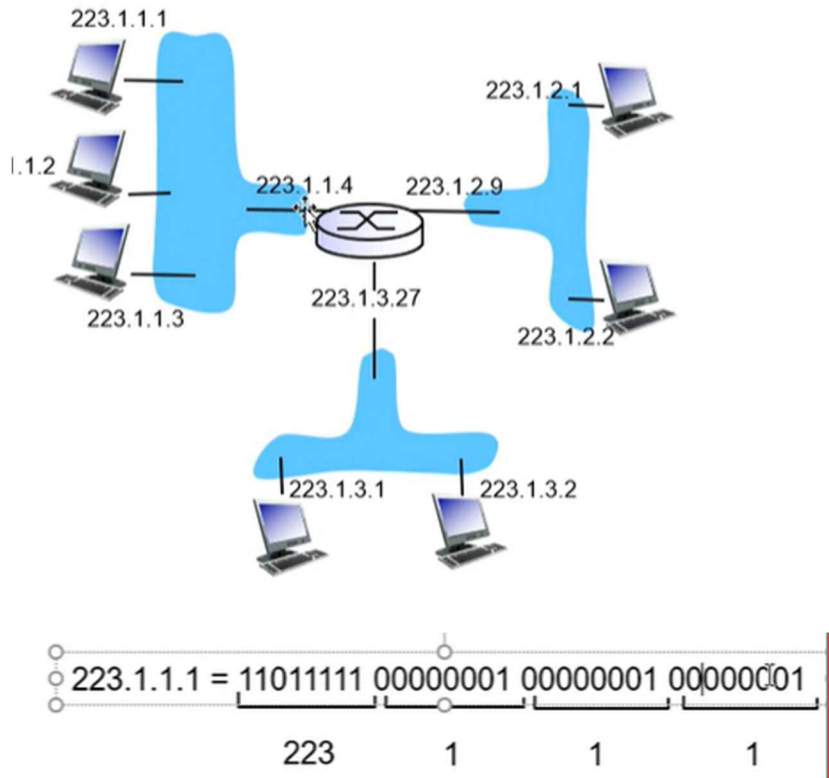
Priority scheduling





Ip V4 adressing:

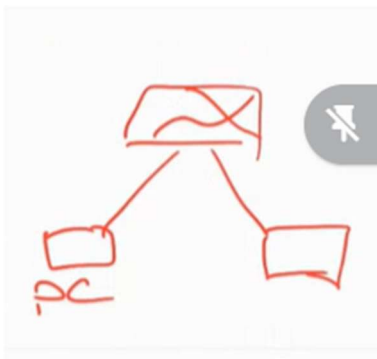
32 bitlik bir tanımlayıcıdır. 222.1.1.1



SUBNET MASK:

255.255.255.0

Ne işe yarıyor anlatacağım dendi



Bu iki bilgisayar aynı ağda mı ?

$$\begin{array}{l} 192.168.1.50/24 \\ \underline{255.255.255.0} \\ 8 \end{array} \quad \begin{array}{l} 192.168.1.109/24 \\ \underline{255.255.255.0} \\ 8 \end{array}$$

DHCP:

Dinamik host control protokol

İp dağıtır

--

Subnet hesaplama

İlk yedi maddede ip adreslerini ve konfigrasyon ayarlarıyla ilgili olarak kullanıcının wireshark programına hazır etmektedir.

örneğin

1.Cmd`yi açıp ipconfig /release komutunu girdim. Böylece mevcut ip bırakıldı

2.wireshark açıldı

1. Are DHCP messages sent over UDP or TCP?

UDP ile çalışır

2. Draw a timing datagram illustrating the sequence of the first four-packet

Discover/Offer/Request/ACK DHCP exchange between the client and server. For

each packet, indicated the source and destination port numbers. Are the port

numbers the same as in the example given in this lab assignment?

anlamadım

3. What is the link-layer (e.g., Ethernet) address of your host?

00:08:74:4f:36:23

4. What values in the DHCP discover message differentiate this message from the DHCP request message?

Dhcp katmanında opinion içerisinde gördüğüm kadarıyla 53tür

4. What is the value of the Transaction-ID in each of the first four (Discover/Offer/Request/ACK) DHCP messages?
0x3e5e0ce3
0x3e5e0ce3
0x3e5e0ce3
0x3e5e0ce3

What are the values of the

Transaction-ID in the second set (Request/ACK) set of DHCP messages?

What is

the purpose of the Transaction-ID field?

6. A host uses DHCP to obtain an IP address, among other things. But a host's IP address is not confirmed until the end of the four-message exchange! If the IP address is not set until the end of the four-message exchange, then what values are used in the IP datagrams in the four-message exchange? For each of the four DHCP messages (Discover/Offer/Request/ACK DHCP), indicate the source and destination IP addresses that are carried in the encapsulating IP datagram.

Dhcp discover , offer , request ve ack adımları gerçekleşir

7. What is the IP address of your DHCP server?

192.168.1.1

8. What IP address is the DHCP server offering to your host in the DHCP Offer message? Indicate which DHCP message contains the offered DHCP address.

192.168.1.101

Dhcp protokolü içinde opinion içerisinde teklifi görebiliriz.

9. In the example screenshot in this assignment, there is no relay agent between the host and the DHCP server. What values in the trace indicate the absence of a relay agent? Is there a relay agent in your experiment? If so what is the IP address of the agent?

10. Explain the purpose of the router and subnet mask lines in the DHCP offer message.

255.255.255.128

11. In the DHCP trace file noted in footnote 2, the DHCP server offers a specific IP address to the client (see also question 8. above). In the client's response to the 1 gün olarak ayarlanmış kill timesi. Buna da Lease süresi denmektedir.

3 What do we mean by "annotate"? If you hand in a paper copy, please highlight where in the printout

you've found the answer and add some text (preferably with a colored pen) noting what you found in what

you've highlight. If you hand in an electronic copy, it would be great if you could also highlight and annotate.

first server OFFER message, does the client accept this IP address? Where in the client's RESPONSE is the client's requested address?

12. Explain the purpose of the lease time. How long is the lease time in your experiment?

Lease time ağ içindeki ipleri dağıltmaya yarayan servist olan Dhcp serverin sona erme süresidir. Süre bitmeden kullanıcı tekrar bağlanırsa, bu değer yenilenir

13. What is the purpose of the DHCP release message? Does the DHCP server issue

Lease süresi bittiğinde kullanıcıya ip adresini bırakması gerektiğinin mesajıdır. Manuel olarak ipconfig /release yazarakta yapılabilir

an acknowledgment of receipt of the client's DHCP request? What would happen if the client's DHCP release message is lost?

14. Clear the bootp filter from your Wireshark window. Were any ARP packets sent or received during the DHCP packet-exchange period? If so, explain the purpose of those ARP packets.

Bağlantıdaki düm düğümler bilinir.

Belirli bir kaynak düğümünden diğer düğümlere en kısa yol hesaplanır

$C(x,y)$ x ile y düğümü arasındaki bağlantının maliyetidir. Direkt olarak birbirlerine bağlı değilse bu bağlantı sonsuzdur

$D(v)$ kaynaktan hedefe giderkenki maliyettir

-Aynı otonom sistem içerisindeki routerlar Aynı yönlendirme protokolünü uygulamalıdır

-gateway router: bir uç routeri diğerine bağlar

2.kayıt 16. Dakka sınavla ilgili

Link-State algoritması(yönlendirmesinde) tüm routerlar birbirinden haberdar oluyordu.

Tüm dünyadaki routerlar birbiriyle haberleşir mi ?

Hayır.routing tablosuna tüm yönlendirmeleri yazamayız.

Link-State algoritması(yönlendirmesinde) bilgisayar iletişimi için paket anahtarlama ağlarında kullanılan iki ana yönlendirme protokolü sınıfından biridir, diğeri uzaklık vektörü yönlendirme protokolleridir.

İnternet = network of networks

Network içindeki bilgiler diğer networkler ile paylaşılmak istenmez

Peki tüm routingler birbirine bağlı ve haberdar değilse ne olacak?

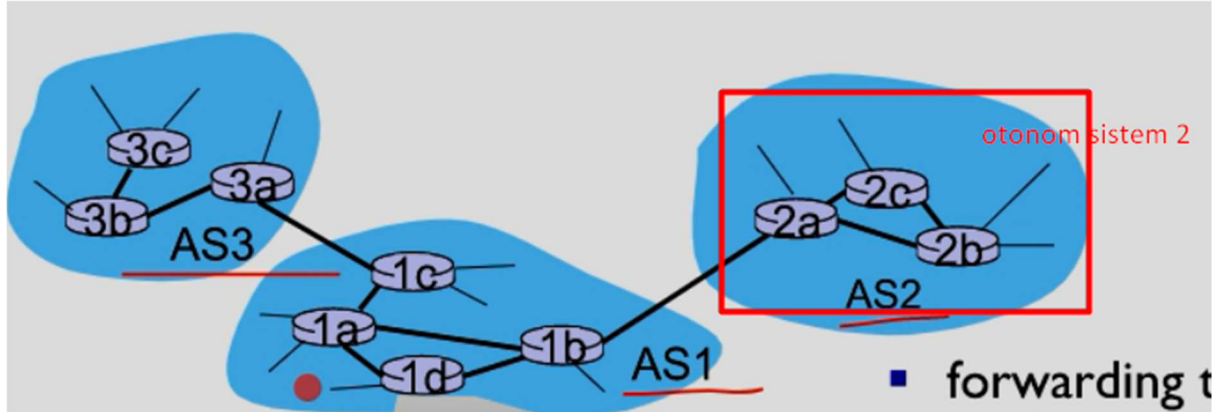
Hepsi kendi içinde ağlanacak yani scale edilecek.

Bölgelere ayrılır. Bu bölgeler **otonom sistem veya domain** olarak adlandırılır

İntra(intrior)-AS routing(otonom içi yönlendirme)

Tüm hostlar routerlar aynı network içinde olmalı

Gateway router: bir otonom sistemi diğer bir otonom sisteme bağlayan, uçta yer alan routerlardır.



Intra as Routing protokolleri: biz ospf üzerinde duracağız

Intra-AS Routing

- also known as *interior gateway protocols (IGP)*
- most common intra-AS routing protocols:
 - RIP: Routing Information Protocol
 - OSPF: Open Shortest Path First (IS-IS protocol essentially same as OSPF)
 - IGRP: Interior Gateway Routing Protocol (Cisco proprietary for decades, until 2016)

Ospf ip protokolü ile bir routerdeki durum bilgisi diğer routerlara iletilir. İp ile iletilmesine UDP ve TCP de dahildir.

IS-IS protokolü de OSPF gibi çalışır.

RIP:

RIP (Router Information Protocol - Yönlendirme Bilgisi Protokolü), uzaklık vektör algoritmasıyla çalışan ve yönlendirmeleri hesaplamak için Bellman-Ford algoritmasını kullanan bir protokoldür. RIP, yönlendirici cihazların tablosunda **Administrative Distance (Yönetim Mesafesi)** 120 olarak yer alır. RIP yönlendiriciler, en iyi yol seçimini yaparken sadece **geçtiği cihaz (hop)** sayısına bakar. RIP en fazla 15 hop' u kabul eder. Bu sayı aşıldığı zaman (yani 16. hopa gelince) **destination unreachable (kaynak bulunamadı)** hatasını verir.

RIP mesajları kapsüle edilmiş şekilde **UDP (User Datagram Protocol – Kullanıcı Datagramı Protokolü)** segmentinde 520'nci porttan yollar. RIP kullanan yönlendiriciler, 30 saniyelik döngüler halinde komşu yönlendiricilere tüm **routing (yönlendirme) tablosunu** gönderir.

RIP avantajları:

- Küçük ağlarda çok kullanışlıdır.
- Kullanımı ve uygulaması kolaydır.
- Tüm topolojiyi bilmediğinden yönlendiricide az bellek tüketimini ve az işlemci yükünü sağlar.

RIP dezavantajları:

- RIP, büyük ve çok büyük ağlarda ölçekleme konusunda yetersiz kalır.
- RIP, en fazla 15 hop gidebilir. Ağ 15 cihazdan büyükse protokol ulaşılamaz hatası verir.
- Büyük bir ağ içinde her yönlendirici RIP anonslarını yapması demek internette büyük bir trafiğin oluşması ve bant genişliğinin azalması anlamına gelmektedir.

- RIP'in **kurtarma (recovery)** süresi uzundur, bu da değişen topolojinin tekrardan düzenlenebilmesini geciktirir ve ağda istenmeyen döngülere neden olur. Bu döngüler yüzünden de veriler ulaşamaz, kullanıcıya teslim edilemez.

OSPF(ripte olmayan ospf de olan özellikler)

Security: uçtan uca güvenlik sağlanır. Doğrulanarak karşıya iletir.

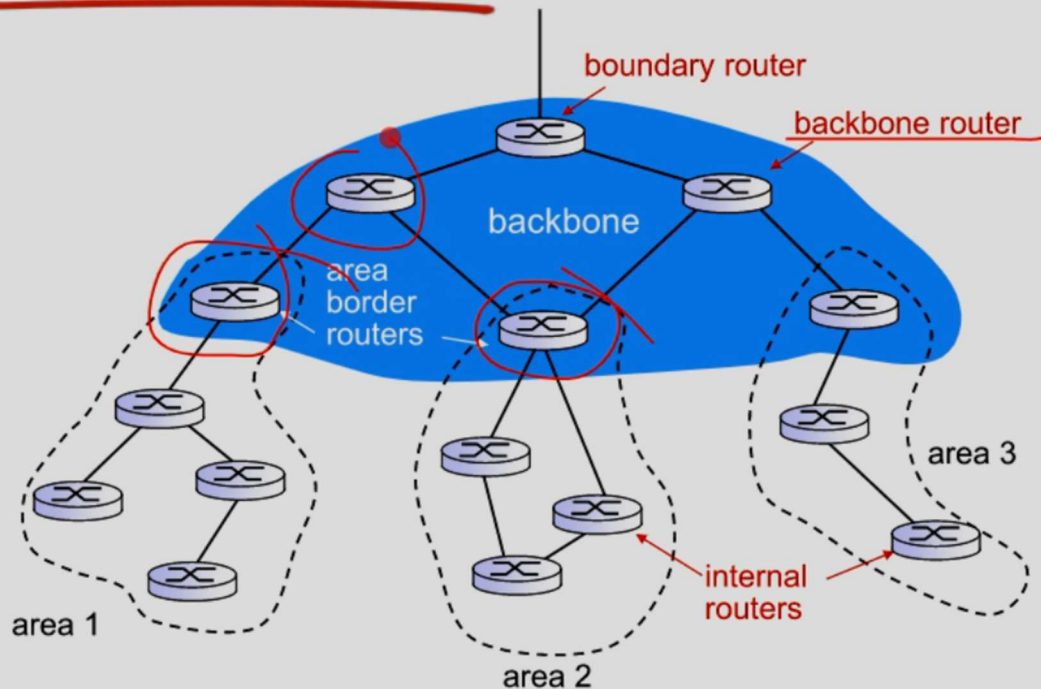
Multiple: aynı maliyetli yollara ospf izin verir

Maliyet ve performans arasında denge sağlama özelliği

Area border routers: hiyerarşideki diğer border routerlerden haberdar olur. Kendi içindeki ağları diğer routerlara haber verir

Hierarchical OSPF

LONGEST PREFIX MATCHING



Hierarchical OSPF

- **two-level hierarchy:** local area, backbone.
 - link-state advertisements only in area
 - each nodes has detailed area topology; only know direction (shortest path) to nets in other areas.
- **area border routers:** “summarize” distances to nets in own area, advertise to other Area Border routers.
- **backbone routers:** run OSPF routing limited to backbone.
- **boundary routers:** connect to other AS' es.

Borader routerlerin kullandığı protokol:

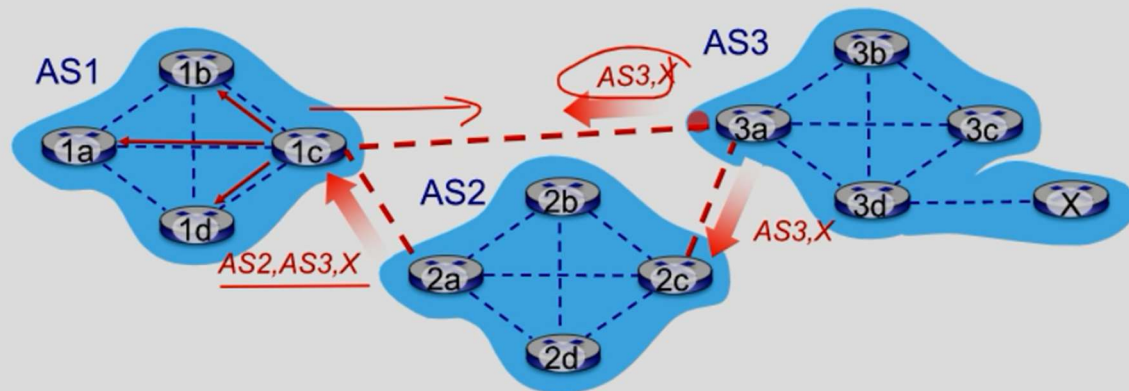
BGP(boarder gateway protokol): internetleri birbirine yapııştırıyor.

ExteriorBGP: dış bgp

interiorBGP: iBGP: politikalar tarafından iyi olan rotaları belirler. Diğer uç birimlere gidecek verilerden önce iBGP ye uğrar.

OTONOM SİSTEMLER HABERLEŞİRKEN HANGİ YÖNÜ TAKİP EDER?

BGP path advertisement



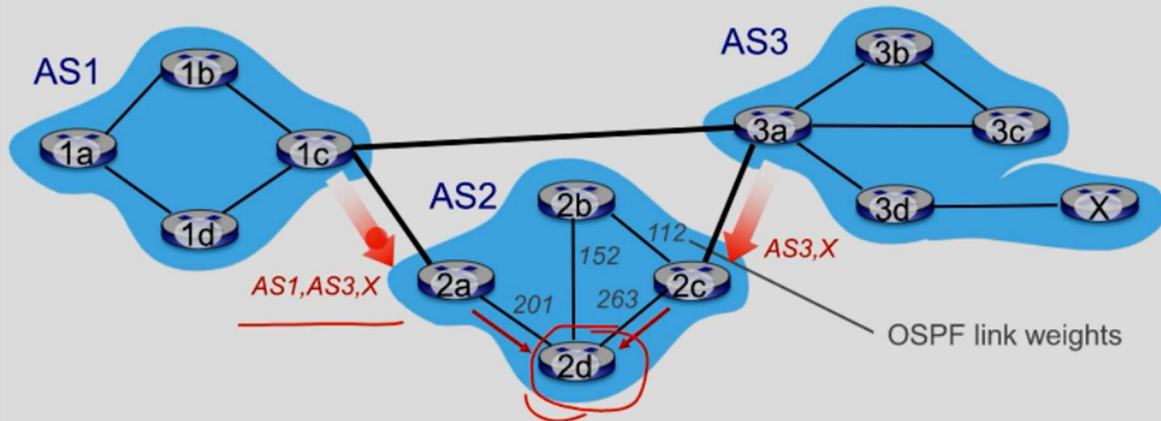
Bgp mesajları

BGP messages

- BGP messages exchanged between peers over TCP connection
- BGP messages:
 - **OPEN:** opens TCP connection to remote BGP peer and authenticates sending BGP peer
 - **UPDATE:** advertises new path (or withdraws old)
 - **KEEPALIVE:** keeps connection alive in absence of UPDATES; also ACKs OPEN request
 - **NOTIFICATION:** reports errors in previous msg; also used to close connection

Hot potato routing: yol maliyetini hesaplamadan en kısa yoldan paketi gönderir

Hot Potato Routing



- 2d learns (via iBGP) it can route to X via 2a or 2c
- *hot potato routing*: choose local gateway that has least intra-domain cost (e.g., 2d chooses 2a, even though more AS hops to X): don't worry about inter-domain cost!