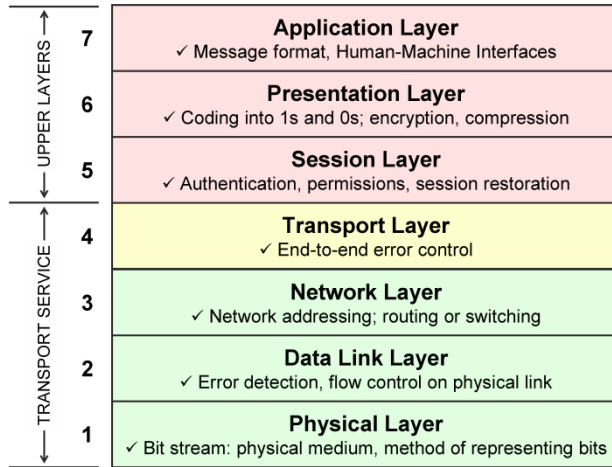


Farklı Makinaların Prosesleri Arasında Haberleşme

Farklı makinalar birbirlerine ağ içerisinde bağlanmış olabilir. Biz bir makinada çalışan bir programın ağa bağlı başka bir makinadaki prosese bilgi göndermesini ve almasını isteyebiliriz. Böyle bir haberleşmede artık işletim sisteminin dışında başka birtakım aktörler de devreye girecektir. Örneğin kablolu sistemde kullanılan hub'a kadar bazı donanım birimleri için içine karışmaktadır. Üstelik bu tür haberleşmelerde işletim sistemleri bile birbirlerinden farklı olabilmektedir. İşte böyle ortamlarda haberleşmenin sağlıklı yürütülmesi için önceden belirlenmiş birtakım kuralların bulunması gerekir. Örneğin kablo standartları ve konnektörler nelerdir? Network kartının özellikleri nasıl olacaktır? Bilgiler nasıl paketlere ayrılıp gönderilecektir? Makinalar nasıl birbirlerinden ayrılacaktır vs. gibi... İşte tüm bu belirlemelere protokol denilmektedir.

Tıpkı fonksiyonların birbirlerini çağırarak daha yüksek seviyeli işlemleri yapar hale gelmesi gibi protokoller de üst üste yığılarak ayrı ayrı oluşturulmaktadır. Her üst protokol aşağının zaten hazır olduğu fikriyle yalnızca kendi gereksinimlerini tanımlamaktadır. Böyle katmanlı tasarımın pek çok faydası vardır. Örneğin bu sayede üst seviye protokoller detay barındırmazlar ve aşağı düzeydeki protokollerin değişmesinden fazlaca etkilenmezler. İşte farklı makinaların haberleşmesi için bu biçimde oluşturulmuş pek çok protokol ailesi vardır. Örneğin AppleTalk, NETBIOS vs. gibi...

Network altında bilgisayar haberleşmesi için protokol katmanlarının nasıl oluşturulması gerektiğine yönelik IEEE, ismine OSI (Open System Interconnection) denilen bir belge yayınlamıştır. Buna OSI model denilmektedir. OSI model bir protokol ailesi değildir. Protokol ailesi oluşturacaklar için bir kılavuz niteliğindedir. OSI'nin toplam 7 katmanı vardır:



OSI'nin en aşağı katmanına "Fiziksel Katman (Physical Layer)" denilmektedir. Fiziksel katmanda iletişimin yapılacağı ortam tanımlanmaktadır. Örneğin kullanılacak kablolar, konnektörler, gerilim seviyeleri gibi. Bunun üzerinde "Veri Bağlantı Katmanı (Data Link Layer)" bulunmaktadır. Bu katmanda network kartlarına ilişkin belirlemeler, fiziksel adresleme belirlemeleri vs. bulunmaktadır. Örneğin Ethernet kartlarının protokolü olan Ethernet Protokolü bir Veri Bağlantı Katmanı Protokolüdür. Network katmanı (Network Layer) mantıksal adreslemenin tanımlandığı, bilginin nasıl paketlere ayrılıp gönderileceğinin tanımlandığı en önemli katmanlardan biridir. Örneğin IP protokol ailesinin IP Protokolü (Internet Protocol) OSI'ye göre Network katmanına ilişkindir. Network katmanında ayrıca "internetworking" için rotalama belirlemeleri de bulunmaktadır. Network üzerinde "İletim Katmanı (Transport Layer)"

bulunmaktadır. Burada paketlerin numaralandırılması, mantıksal port adreslerinin tanımlanması, hata durumunda bunun telafi edilmesi gibi belirlemeler bulundurulmaktadır. Örneğin IP protokol ailesindeki TCP ve UDP protokolleri İleti Katmanına ilişkin protokollerdir. "Oturum Katmanı (Session Layer)" pek çok ailede bulunmamaktadır. Burada haberleşme için gereken oturum açmaya yönelik belirlemeler bulunur. Örneğin izinler, kimlik doğrulama gibi. Bunun yukarısında da "Sunum Katmanı (Presentation Layer)" bulunur. Sunum katmanında gönderilip alınan bilgilerin sıkıştırılmasına, açılmasına, şifrelenmesine vs. yönelik belirlemeler bulunmaktadır. IP protokol ailesi Sunum Katmanına da sahip değildir. Nihayet en tepede "Uygulama Katmanı (Application Layer)" bulunmaktadır. Bu katman artık belli bir amacı gerçekleştirmek için oluşturulan yazılımların kullanacağı belirlemeleri içerir. Örneğin eposta için kullanılan POP3, dosya transferi için kullanılan FTP birer Uygulama Katmanı Protokolüdür.

İnternetin Kısa Tarihi

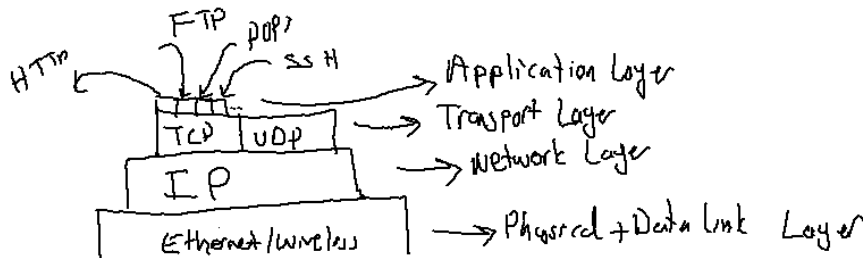
Bilgisayarları birbirlerine bağlamak ilk kez 60' yıllarda insanların aklına gelmiştir. Soğuk savaş yıllarında Amerika Savunma Bakanlığına bağlı olan DARPA (Defense Advanced Research Project Agency) kurumu birkaç üniversite ile 1969 yılında ARPANET isimli bir proje başlattı. ARPANET ilk kez 1969 yılında uzak mesafeden dört üniversitenin birbirlerine bağlanmasıyla hayata geçirilmiş oldu. ARPANET'te daha sonra bazı devlet kurumları ve üniversiteler katılmaya başlamıştır. 70'li yılların sonlarına doğru ARPANET Amerika'da gelişmeye başlamıştır. 1983 yılında ARPANET NCP (Network Control Protocol) protokolünü bırakarak IP ailesine ailesine geçmiştir. Ve artık ağ İnternet ismiyle yayılmaya devam etmiştir. İnternet 80'li yıllarda Avrupa'ya ve Türkiye'ye de geldi. Ancak tabi kişisel bilgisayarlar daha yeniydi ve İnternet'e ancak Üniversitelerden ve bazı devlet kurumlarından, özel sektörden bağlanılabiliyordu. 1990-91 yıllarında HTTP protokö tasarlandı ve ilk Web sayfaları oluşturulmaya başlandı. 90'lı yılların ortalarına doğru tüm dünyada kişisel bilgisayarlarla servis sağlayıcılar sayesinde İnternet'e girmek mümkün hale gelmiştir. Daha sonraları modern modem/router'larla yüksek hızlı evden erişimler sağlanmıştır.

İnternet ismi "internetworking" sözcüğünden gelmektedir. Internetworking "yerel ağların birbirlerine router isimli cihazlarla bağlanmalarıyla oluşturulmaktadır. Internetworking temel bir terimdir ve IP protokol ailesinin ismi buradan gelmektedir. Bugün İnternet denildiğinde herkesin bağlandığı ARPANET'ten evrimleşen dev ağ aklımıza gelir. (İnternet yazarken l'yı büyük yazarsak bu ağ anlaşılır.) Şüphesiz mevcut protokoller sayesinde herkes kendi internetini kurabilir. Örneğin biz de birkaç arkadaşımızla ayrı bir İnternet dünyası oluşturabiliriz. Hatta bazı ülkelerin bu biçimde kendilerine özgü İnternet'leri vardır.

IP Protokol Ailesi

IP açık bir protokol ailesidir. Burada açık demekle hiçbir şirketin malının olmadığı bağımsız konsorsiyumlar tarafından yönetildiği anlamına gelmektedir. Ayrıca dokümanlar herkes tarafından paylaşılmakta ve isteyen kişiler önerilerde bulunabilmektedir.

IP protokolü Vint Cerf ve Bob Kahn tarafından 1974 yılında önce TCP sonra IP biçiminde tasarlanmıştır. Sonra aileye diğer üyeler katılmıştır. İlk ciddi gerçekleştirmesi BSD sistemlerinde yapılmıştır. 1983 yılında ARPANET'in IP ailesine geçmesiyle popülaritesi çok artmıştır.



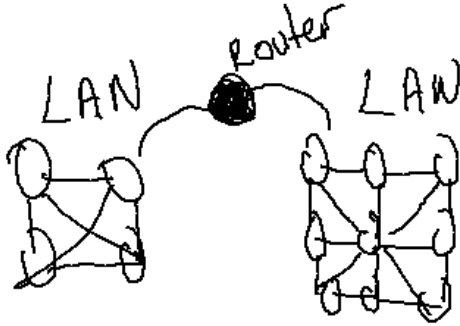
IP protokol ailesinin temel protokolleri dört katmandan oluşmaktadır.

IP protokol ailesi aslında geniş bir ailedir. Ailede pek çok yardımcı protokol vardır. Yukarıdaki şekil yalnızca kursumuzda söz konusu edilen konuları kapsayacak biçimde oluşturulmuştur.

Ailenin en önemli taban protokolü IP (Internetworking Protocol) protokolüdür. Zaten aileye ismini bu protokol vermiştir. IP protokolü paket anahtarlama (packet switching) bir protokoldür. Yani bilgiler paket denilen öbeklere ayrılarak gönderilip alınır. IP protokolünde adresleme artık fiziksel değil mantıksaldır. IP protokol ailesinde ağa bağlı her birime "host" denilmektedir. IP protokolünde her host'un ismine ilişkin IP adresi denilen mantıksal bir adresi vardır. Mantıksal adres bunun donanımsal olarak belirlenmediği yazılımsal olarak atandığı anlamına gelmektedir. Fakat örneğin Ethernet protokolünün kullandığı MAC adresi fiziksel bir adrestir. Fiziksel adres bunun donanımsal olarak kartın üzerine çakılı olduğu ya da donanımın kendisinin bunu tespit edip işlem yaptığı adres demektir. Dolayısıyla mantıksal adresler dinamiktir, fiziksel adresler statiktir. Mantıksal adresler biz ağa dahil olduğumuzda bize atanmaktadır. Tabi biz de istediğimiz adresin atanması konusunda ısrarcı olabiliriz.

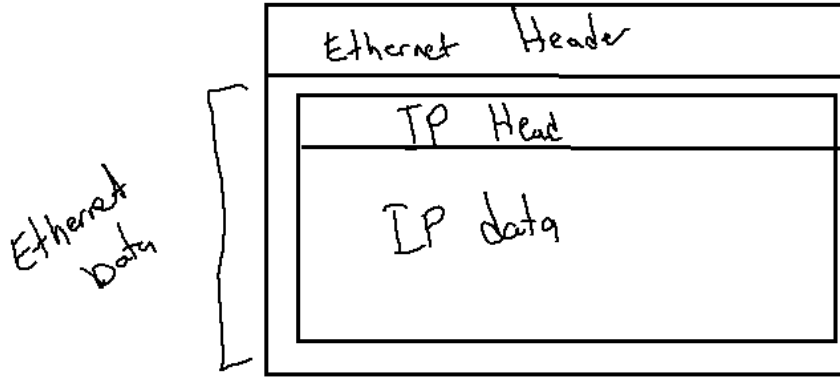
IP protokolünün de versiyonları vardır. Şu anda hala ağırlıklı kullanılan versiyon IPV4'tür. Ancak IPV6 yavaş yavaş daha yaygın kullanılır hale gelmiştir. IPV4te IP adresleri 4 byte uzunluktadır. Ancak IPV6'da IP adresleri 16 byte'tır. 4 byte'lık IP adresleri şu an için artık çok yetersiz kalmaktadır.

Bugün bilgisayarlarımızda fiziksel ve data link katmanı olarak Ethernet ve Wireless Protokolleri kullanılmaktadır. Ethernet protokolü ethernet kartına gereksinim duyar. Bu kart fiziksel olarak bilgileri bilgisayarımızdan dışarı gönderip almakta kullanılır. Ethernet protokolü de paket anahtarlama bir protokoldür. Yani bilgiler paket paket gönderilip alınır. Paket anahtarlama hattın etkin kullanımını sağlar. Biz Ethernet kartlarını bir hub'la birbirine bağlayarak yerel bir ağ (local area network) oluşturabiliriz. Bugün evlerimizdeki ağ da yerel bir ağdır. Yerel ağları birbirlerine bağlamak için "router" denilen aygıtlar kullanılır. Ethernet kartı (yani network kartı) aynı ağdaki bir bilgisayardan diğerine paket haberleşmesi için kullanılmaktadır. Ancak router farklı ağlar arasında paket haberleşmesi için kullanılır. Bugün evlerimizdeki ADSL modemler aynı zamanda birer router görevindedir.



Bizim evimizdeki yerel ağ Internet isimli dev ağa router aracılığıyla tek bir host gibi bağlanmaktadır. Dolayısıyla bizim Internet için dışarıdan kullanılacak tek bir IP adresimiz vardır (Tabi tek bir router ve hattımızın bulunduğunu varsayıyoruz). Bizim evimizdeki yerel ağ ayrı bir IP ağıdır. Yani ayrı bir dünyadır. Biz istersek hiç Internet'te çıkmadan kendi yerel ağımızda tüm Internet uygulamalarını (Yani IP protokol uygulamalarını) çalıştırabiliriz. Buna genellikle "Intranet" denilmektedir. O halde bizim evimizdeki bir bilgisayarın bir yerel IP adresi vardır bir de router'ımızın Internet'ten görülen bir IP adresi vardır. Router dış dünyadan gelen paketleri yerel ağda uygun bilgisayara dağıtmaktadır. Yerel ağdaki paketleri de dış dünyaya ilişkinse dış dünyaya yollamaktadır. Biz yerel ağımızdaki bir host'tan diğerine bilgi gönderirken router devreye girmez.

Ip protokolünde gönderilen bir paketin başında "IP header" isimli bir başlık kısmı vardır. Burada pakete ilişkin metadata bilgileri bulunur. Örneğin paket hangi IP adresine gönderilmektedir? Checksum bilgisi nedir? Hangi IP versiyonu kullanılmaktadır? vs. Aslında tabi (böyle olmak zorunda değil ama) bilgiler neticede ethernet kartı ile gönderilip alındığı için IP paketi aslında Ethernet protokolünün ethernet paketinin data bölümünde kodlanır. Ethernet protokolünün de ayrı bir header bölümü vardır. Örneğin:



Ethernet protokolü IEEE 802.3 numaralı standardıyla belirlenmiştir. Wireless protokolü de aynı ailedendir. O da IEEE 802.11 numaralı standarttır.

Ip protokü ile birden fazla paketten oluşan bilgi gönderilebilir mi? Evet fakat bunun için paketlere numara vererek bizim de adeta ayrı bir protokol oluşturmamız gerekir. Zaten TCP protokolü buna benzer bir protokoldür.

TCP protokolü güvenilir (reliable) bir protokoldür. Burada güvenilir demek alışverişin yolda bozulmasının telafi edilmesi ve paketlerin düzgün aktarılması anlamına gelir. Çünkü TCP'de bir akış kontrolü (flow control) vardır. Gönderen tarafla alan taraf karşılıklı konuşarak hatalı giden paketlerin telafisini sağlayabilmektedir. TCP stream tabanlı bir protokoldür. Stream tabanlı demekle byte byte okumaya kaldığı yerden devam edebilmek anlaşılır. TCP ile biz daha büyük bilgileri gönderip alabiliriz. TCP bu durumda bu bilgiyi IP paketlerine böler. Onlara numara verir ve onların karşı tarafa güvenli ulaşmasını denetler. Karşı taraf gelen bilgiyi sanki borudan okuma yapıyormuş gibi byte byte elde edebilir.

UDP (User Datagram Protocol) güvenilir olmayan paket tabanlı (datagram) bir haberleşme sunar. Yani UDP'de bilgiler IP'deki gibi bağımsız paketler halinde gönderilip alınır. UDP'de bir paket ya alınır ya alınmaz. Byte byte okuma mümkün değildir. Paketin alındığına dair bir geri bildirim yapılmaz. Tabi bu özelliğinden dolayı UDP daha hızlıdır. UDP özellikle periyodik veri gönderimlerinde, televizyon yayını gibi işlemlerde tercih edilmektedir.

TCP bağlantılı (connection oriented) bir protokoldür, UDP bağlantısızdır (connectionless). Bağlantılı protokol demek iki taraf haberleşmeden önce birbirlerine bağlanıp karşılıklı konuşma için birbirlerini tanımaları demektir. TCP tipik olarak client-server tarzda bir çalışmayı akla getirmektedir. Client-server haberleşmede bir taraf client bir taraf server olur. Client taraf server tarafa bağlanır, haberleşme bundan sonra yapılır.

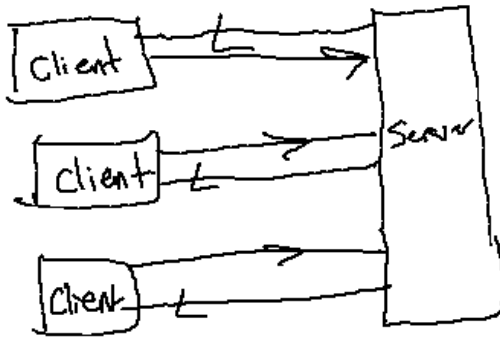
TCP	UDP
Bağlantılı	Bağlantısız
Stream Tabanlı	Datagram Tabanlı
Güvenilir	Güvenilir Değil
Yavaş	Hızlı

TCP ve UDP'de işin içine port numarası kavramı da girmektedir. Port numarası aynı host'taki uygulamaları birbirlerinden ayırmak için düşünülmüştür. Adeta şirketlerdeki içsel (internal) telefon numaralarına benzetilebilir. TCP ve UDP protokollerinde bilgi göndermek için yalnızca gönderilecek host'un IP'sinin bilinmesi yeterli değildir. Aynı zamanda oradaki uygulamanın hangi port ile ilgilendiğinin de bilinmesi gerekir. Genellikle gösterimde ip adresi ve port numarası aralarına ':' karakteri getirilerek "ip:port" biçiminde belirtilmektedir. IPV4'te toplam 65536 port numarası vardır (yani port numarası için iki byte yer ayrılır). IPV6'da ise port numaraları 4 byte uzunluğundadır. IPV4'te ilk 1024 port numarası Internet'in kendi uygulama protokolleri için ayrılmıştır. Bunlara "well known" portlar da denilmektedir. Örneğin FTP 21, SSH 22, Telnet 23, HTTP 80 numaralı portları kullanmaktadır. Biz kendi uygulamalarımız için port numarası belirleyeceksek ilk 1024 portu kullanmamalıyız.

Client-Server Çalışma Modeli

Yukarıda da belirtildiği gibi TCP tipik olarak client-server bir çalışmayı akla getirmektedir. Client-Server modelde ismine client ve server denilen iki ayrı program vardır. Asıl işi server program yapar. Client yalnızca istekte bulunur. Server işi yapar sonuçları client'a gönderir. Bir server birden fazla client'a hizmet verebilmektedir.

Anahtar Notlar: Server bir client' a hizmet verirken aynı anda başka bir client hizmet alamıyorsa bu tarz server'lara "single-client" veya "iterative" server denilmektedir. Tersine bir client hizmeti devam ederken başka bir client da hizmet alabiliyorsa bu tarz server'lara "multi-client" veya "concurrent" server denilmektedir.



Client-Server modelde önce client server'a bağlanır. Haberleşme ondan sonra başlar. Client-Server uygulamalar her ne kadar TCP'yi çağırıştırıyorsa da aslında bu bir haberleşme mimarisidir. Yani aslında client-server çalışma için IP ailesinin kullanılması gerekmez. Bu çalışma örneğinin aynı makinadaki prosesler arasında borularla mesaj kuyruklarıyla da sağlanabilir.

Client-Server çalışmanın şu avantajları vardır:

- 1) Server programın çalıştığı makine güçlü olabilir. Biz de onun gücünden yararlanmak istiyor olabiliriz. Örneğin uzun zaman alan bir işlemi el terminalinden yapmak yerine el terminalini client olarak kullanıp asıl işi server'a yaptırmak uygun olabilir.
- 2) Server program kaynak paylaşımı sağlayabilir. Örneğin yazıcı tek bir bilgisayara bağlıdır. Başka bilgisayardaki print programları client gibi çalışarak yazıcının bağlı makinadaki server programa isteği iletir. Server da print işlemini client için yapar. Ya da örneğin server'a bir veritabanı bağlıdır. Client ondan istekte bulunur. Örneğin banka ATM'lerinde veritabanı ATM makinasının içerisinde değildir. ATM'deki program client program gibi davranmaktadır.
- 3) Server program client'lar arasında işbirliği sağlayabilir. Onlar arasındaki iletişime aracılık edebilir. Örneğin bir chat programında client'lar birbirini tanımamaktadır. Herkes yalnızca server'ı tanır. Her client server'a bağlanır. Server client arasında haberleşmeye aracılık eder. Ağ üzerinde çalışan oyun programları bu biçimde bir server'ın işbirliği ile gerçekleştirilmektedir.
- 4) Client-Server çalışma dağıtık uygulamalarda da karşımıza çıkabilmektedir. Yani bir işin belirli parçalarını başka bilgisayarlarda yapıp sonra onu birleştirmek isteyebiliriz.

Soket (Socket) Arayüzü

Soket arayüzü ağ haberleşmesi için kullanılan bir kütüphanedir. Soket kütüphanesi ilk kez 1983 yılında BSD sistemlerinde gerçekleştirilmiştir. Daha sonra başka sistemlere uygulanmıştır. Microsoft'un soket arayüzü BSD soketlerinden alınmadır. Buna Winsock kütüphanesi denilmektedir. Windows'ta iki grup soket API'si vardır. Bunlardan birincisi tamamen BSD uyumlu API'lerdir. (Burada fonksiyon isimleri BSD'deki ile aynıdır.) İkinci olarak başı WSA ile başlayan Windows'a özgü soket API'leridir. Biz Windows'ta da BSD uyumlu soket fonksiyonlarını kullanırsak UNIX/Linux uyumunu da sağlamış oluruz.

Soket arayüzü yalnızca IP ailesi için düşünülmüş bir arayüz değildir. Diğer protokolleri de kapsayan genel bir arayüzdür. Bu nedenle fonksiyonların parametrik yapıları biraz daha karmaşık olma eğilimindedir.