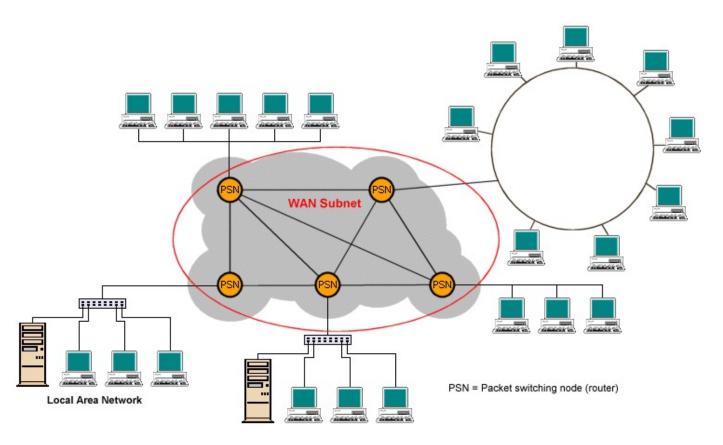
AĞ KATMANI

 Ağ Katmanı, farklı ağlardaki (farklı broadcast domain) iki bilgisayar arasındaki paket bazında haberleşme için, paketin göndericisinden alıcısına ulaşana kadar ağdaki en uygun düğümlerden geçirilip yönlendirilmesini sağlar. Karmaşık bir yapı içeren bir katmandır.



Özellikle iki nokta arasında farklı yollar mevcutsa AĞ Katmanı önem kazanır.

Ağ katmanı, veri paketlerinin alıcısına giderken, ağ koşullarını, bağlantı durumlarını, öncelikeri ve diğer birçok parametreleri değerlendirerek hangi yolun daha uygun olacağının belirlenmesi ve paketlerin bu yoldan aktarılmasının sağlanması için gerekenleri tarif eder.

Ağ katmanında tarif edilen diğer görevler;

**** Adresleme
**** Ağlar arası yönlendirme
**** Tıkanma kontrolü
**** Mantıksal adreslerden Fiziksel Adreslere dönüşüm
**** Yönlendirme cihazları arasındaki mesaj aktarımı
**** Kullanıcıların gönderdiği paketlerin yerine teslimi.

Tüm bu görevler nasıl yapılacaktır?

Ağ katmanında yapılması gereken işlevler sonucunda, paket sadece ilgili ağa gönderilir. Diğer ağlar meşgul edilmez

1-Adresleme: a)Mantıksal adresleme

b) Servis adresleri (Ağ +MAC+SAP)

2-Yol Belirleme (Route Discovery): a)Distance vector

b) Link-State

3-Yol Seçimi(Route Selection): a) Statik

b)Dinamik

4-Anahtarlama (Switching): a) Devre Anahtarlama

b) Mesaj Anahtarlama

c) Paket anahtarlama

5- Bağlantı Servisleri a) Ağ düzeyi akış kontrolu

b) Hata kontrolu

c) Paket sıralama kontrolu

Bu katmanda çalışan temel protokollar;

1-TCP/ IP'nin IP kısmı

2- Novel'in IPX protokolu

IP protokolu ve yapısı

- IP (Internet Protocol)
- IP protokolü, değişik ağ türlerini (örneğin, istemci tarafında FDDI bağlantı türü varken sunucu tarafında ethernet olması gibi) saydam olarak birbirine bağlayan yapıya sahiptir.
- Bu protokol, verinin karşı tarafa hangi yolla gittiği ile ilgilenir; son noktalar arasında güvenli veri akışını, akış kontrolünü, datagramların iletim sırasını belirleyemez.
- IP'nin Görevleri
 - Adresleme
 - Yönlendirme

IP Datagram Yapısı

 IP datagram, IP başlığı ve üst katmandan gelen bilgilerin birleşiminden oluşur.

Uyarlama		Başlık	Hizmet Türü	
Toplam Uzunluk				
Kimlik Saptaması				
Bayrak Bitleri		Fragment Kayıklığı		
Yaşam Süresi(TTL)			Protokol	
Başlık İçin Hata Sınama Bitleri				
Gönderici IP Adresi				
Alıcı IP Adresi				
Seçenekler				
TCP Segmenti				
(TCP Başlığı + Kullanıcı Verisi)				

IP datagramı oluşturan birimler:

- Version (Uyarlama 4bit): IP datagramın türünü tanımlar.
 IP version türleri
- 0→ayrılmış
- 1-3 \rightarrow tanımsız
- 4→IPv4
- 6→IPv6
- 15→Ayrılmış
- IHL(Internet Header Length-IP datagram başlık uzunluğu -4 bit):
 Datagram başlığının uzunluğunu tanımlamak için kullanılır. Bu uzunluk en az 20 oktet, en fazla 60 oktet olabilir.
- Servis(Hizmet) Türü(ToS-Type of Service 8 bit): Talep edilen hizmetler hakkında bilgi verir. Bu sayede datagram için öncelik, gecikme, güvenirlilik gibi tanımlamaların yapılmasını sağlar.
- Toplam Uzunluk (16 bit): IP başlığı ve datagram üzerinde taşınan verilerin oktet cinsinden uzunluğunu temsil eder. Datagramın alabileceği en büyük uzunluk 65535, en küçük uzunluk 576 oktettir.

- Datagram Tanım Alanı (Kimlik saptaması 16bit): İletilecek olan verinin tanınması için datagrama verilen değerdir. Bu değerler 0- 65535 arasında olabilirler.
- (Fragment 3 + 13 bit) Parçalama Bayrakları:
 Paketlerin parçalanması ve yeniden birleştirilmesi işleminde parçalama bilgilerinin tutulması ve karşı tarafa aktarım işinin yürütülmesi.
- Datagram Yaşam Süresi (Time to Live-TTL 8 bit):
 Datagramın ağ üzerinde ne kadar kalabileceğini belirtir.

 Sn cinsinden birimlendirilir.
- Protokol Alanı (8 bit): Üst katmanlara hangi protokolün kullanıldığı bilgisini verir. Örneğin, TCP için 6, UDP için 17 vb.
- Başlık Sınama Toplam Alanı (Header Checksum 16bit): TCP, IP gibi protokollerden alınan verilerin doğruluğunu denetlemek için kullanılan alandır.

- Kaynak ve Hedef IP Adresleri(32 + 32 bit):
 Paketin gönderilecek ve alınacak bilgisayarların
 IP adresini içinde barındıran alandır.
- Seçimlik (IP Options): Bazı durumlarda kullanılır. Örneğin çok ender rastlanan başlık bitlerini bilgi biti olarak kullanılmasını engellemek için kullanılabilir.
- Doldurma Biti: IP başlığının toplam uzunluğunu korumak için kısa başlıklarda, başlık sonuna ilave edilen sıfırları ifade eder.

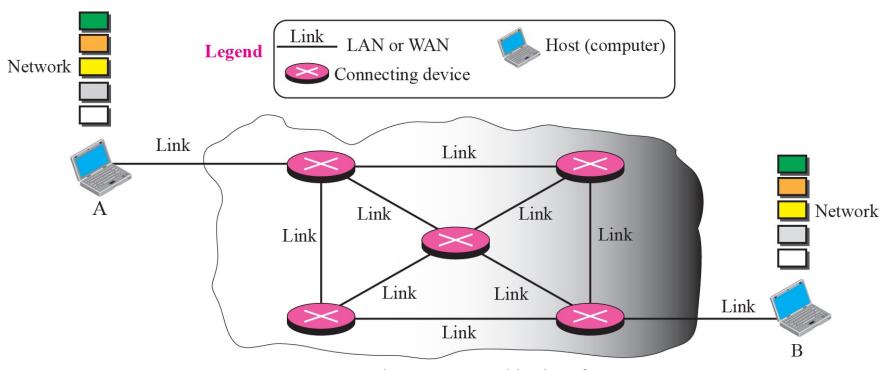
Anahtarlama Teknikleri

Haberleşmede amaç verinin bir noktadan, haberleşmek istenen diğer noktaya aktarılmasıdır.

Büyük ağlar arası ortamlarda iki nokta arasında 1'den fazla yol bulunabilir. Bu yollar üzerinde yönlendirici cihazlar vardır.

Dolayısıyla taşınacak veri de anahtar adı verilen bu yönlendirici cihazlar vasıtasıyla bir hattan diğerine aktarılarak uçtan uca iletilir.

Anahtarlar ilk telefon hatlarından günümüze kadar, haberleşme sistemlerinin vazgeçilmez elemanları olmuşlardır.

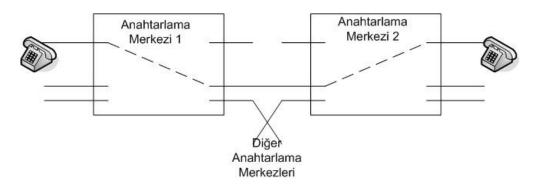


An internet as a combination of links and connecting devices

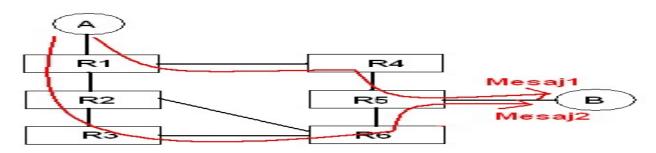
Devre Anahtarlama

Devre anahtarlamada haberleşecek iki uç düğüm arasında bir yol (fizikseldevre) kurulur. Bağlantı boyunca belirlenen yol kurulu kalır ve veri aktarımı bu yol üzerinden gerçeklenir.

- Devre anahtarlamalı veri aktarımında iki aşama vardır :
 - 1) Devrenin kurulması. Bu aşamada ilgili uç düğümler arasında özel sinyalleşme mesajları kullanılarak bir yol kurulur.
 - 2) Veri aktarımının başlatılması
- Bağlantı iki uç düğümden biri tarafından koparılabilir (bitirilebilir).
- +Telefon şebekesi, bu anahtarlama tipine örnektir. Zamana duyarlı gerçek zaman uygulamaları için devre anahtarlama uygun ortamdır.
- +Daha önceden kurulmuş devre üzerinden, trafik bekletilmeden aktarılır.
- +Sabit aktarım hızı için atanmış aktarım kanalı ve devrenin kurulduktan sonra, kanal erişim gecikmesinin olmaması avantajlarıdır.
- Uzun bağlantı kurma gecikmeleri, sadece tekbir iletişim için kanalın atanmış olması (veri miktarının zamana göre değişiminden dolayı kanal kapasitesinin etkin kullanılamaması), pahalı olması, dezavantajlarıdır.



Mesaj Anahtarlama



- Mesaj anahtarlamada aktarılması planlanan veri mesaj adını alır. Mesaj yaratıldıktan sonra iletilmek üzere hedef adresi eklenerek ağa gönderilir.
- Mesajin iletimi öncesinde bir yol kurulması söz konusu değildir.
- Mesaj bir düğümden bir sonrakine sakla-ve-gönder (store and forward) yöntemi ile iletilerek varış noktasına kadar yönlendirilir. Bu yöntemde bir mesajı kaynak düğümden varış düğümüne aktarırken geçilen tüm düğümlerde gelen mesaj öncelikle saklanır. Ulaşması gereken kaynak düğüm düşünülerek bir sonraki düğüm belirlenir ve o düğüme göndermek için işlemler başlatılır
- + Devre anahtarlamaya göre birden fazla birim paylaşımlı olarak aynı aktarım kanalını kullanabilir. Mesajların geçici olarak saklanması ağ trafik yoğunluğunu azaltabilir. Bir mesaj değişik yerlere gönderilebilir, mesajlara öncelik tanınabilir.
 - Gerçek zaman uygulamaları için uygun değildir. Anahtar bufferları büyük kapasiteli olmalıdır. Uzun erişim gecikmeleri olabilir.

11

Paket Anahtarlama

Paket Nedir?

Paketler bir bağlantı sırasında aktarılması planlanan verinin küçük parçalara bölünmüş halidir.

Donanım teknolojisine bağlı olarak paket yapısı da değişir. Belli bir ağ teknolojinde paket aktarımı için oluşturulan yapıya *çerçeve* (frame) denir. Şekilde de görüldüğü gibi verinin önünde ve arkasında çerçevenin başlangıcını ve sonunu gösteren küçük bloklar vardır. Bu bloklarda adres, hata kontrolü vb gibi veri aktarımı sırasında kullanılan ağ teknolojisine özel kontrol bilgisi bulunur.

Paketleri bölerek aktarım sırasında taşınan veri miktarı artar. Gerçekte taşınması istenen veriye ek olarak gönderilen fazlalıklara *ek yük* diyoruz.

çerçeve başı	veri bloğu	çerçeve sonu
--------------	------------	--------------

Paket anahtarlama; devre anahtarlama ve Mesaj anahtarlamanın pozitif yönlerinin kullanıldığı bir tekniktir.

Mesaj boyutunun değişken olması mesaj anahtarlamanın uygulamasında anahtarlarda bellek sorunu yaratır.

Ayrıca uzun mesajlarda mesajin başındaki veri, mesajın sonundaki veri gelene kadar bekletilir. Bu da, zamana duyarlı uygulamalarda sorunlara neden olur.

Paket anahtarlama tekniği iki farklı şekilde gerçekleştirilir.

- 1- Datagram Paket Anahtarlama
- 2- Sanal Devre paket Anahtarlama

Bu yöntemlerde mesajlar, küçük parçalara (paketlere) ayrılarak gönderilir. Herbir paket uygun biçimde kaynak,hedef, aradüğüm adresleri ile donatılır. Farklı düğümlerden geçerek alıcısına ulaşır.

Paketlerin maksimum uzunlukları belirlenmiş olduğundan, Erişim zamanı ve saklama donanımı minimumudur.

Küçük paketlere yerleştirilmiş veri daha hızlı hareket edebilir. Bir bağlantıya ait paketlerden biri herhangi bir düğüme varırken aynı bağlantının başka bir paketi o düğümden çıkabilir.

Datagram Paket Anahtarlama

Paket anahtarlama temelde mesaj anahtarlamanın sakla-ve-gönder yöntemini kullanır.

Paket anahtarlamada da mesaj anahtarlamada olduğu gibi bağlantı öncesinde yol kurulmaya gerek duyulmaz.

Paket anahtarlamada aynı bağlantıya ait bir paket bir öncekinden farklı bir yolu kullanarak varış noktasına ulaşabilir. Bu da paketlerin varış düğümünde sıralanmasını gerektirir.

Paketlerin varış düğümüne ulaştırılırken farklı yollar izlemesi her paketin düğümler üzerinde ayrı ayrı işlenmesi ve o an için en uygun yoldan varış düğümüne yönlendirilmesinin sonucudur. Bu nokta hataya duyarlılık söz konusu olduğunda bir avantaja dönüşür.

Kaynak-varış arasındaki bir düğüm bozulduğunda ya da bir bağlantı koptuğunda alternatif yolların kullanılması mümkündür. Devre anahtarlamalı yöntemde kurulan yolun değiştirilmesi mümkün değildir.

Paket anahtarlama yöntemi ağ üzerindeki kaynakları en iyi şekilde kullanmayı hedefler. Kaynakları önceden rezerve etmesi söz konusu değildir.

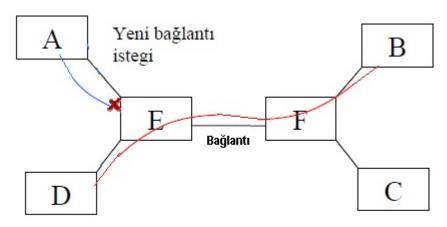
Ancak bu durum gerçek zamanlı ve belli bir kapasiteye ihtiyaç duyan uygulamalar için istenmeyen sonuçlar yaratabilir. Bu nedenle, gerçek zamanlı uygulamalarda tercih edilen bir yöntem değildir

Paket Anahtarlama Temelli Ağlarda Ağ Kaynakları Daha İyi Paylaşılır

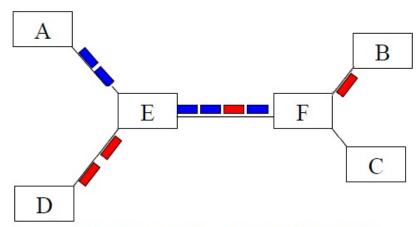
Bu konuyu aşağıdaki basit örnek üzerinde anlatmak daha kolay olabilir. Altı düğümlü (A,B, C, D, E ve F düğümleri) bir haberleşme ağımız olduğunu ve bu ağ üzerinde <u>devre</u> <u>anahtarlama</u> temelli bağlantıların kurduğunu düşünelim.

Bu durumda, B ve D düğümleri arasında kurulmuş bir bağlantı D-E, E-F ve F-B hatlarını kullanacaktır. Aynı ağ üzerinde, daha sonra, A ve C düğümleri arasında başka bir bağlantı kurmak istersek D ve B düğümleri arasındaki bağlantının bitmesini beklememiz gerekir. Çünkü E-F hattının kullanılıyor olması A ve C düğümleri arasında yeni bir bağlantı kurulmasını engeller.

Paket anahtarlama tekniğinde ise ikinci bağlantı birinci bağlantının sona ermesini beklemeden paket aktarımına başlayabilir. Bu da kaynakların verimli kullanıldığını gösterir.



Devre anahtarlamada aynı kaynakları kullanacak iki bağlantı isteğinin ardışıl gelmesi durumu



Paket anahtarlamada aynı kaynakları kullanacak iki bağlantı isteğinin ardısıl gelmesi durumu





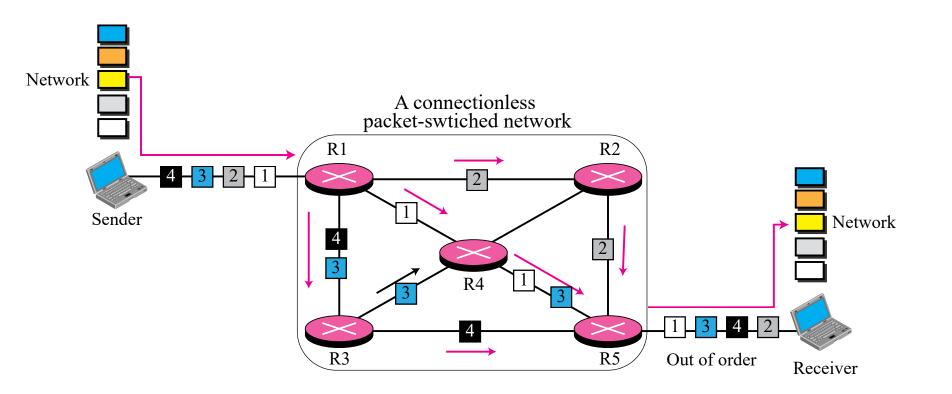
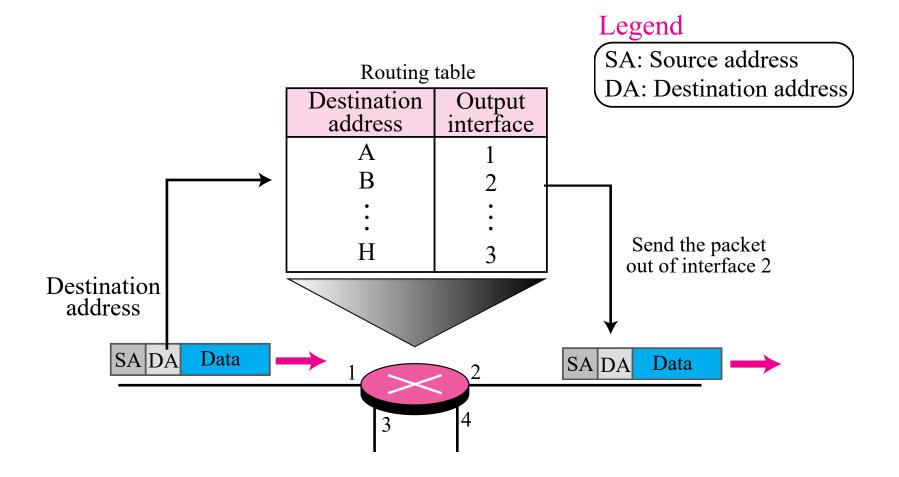


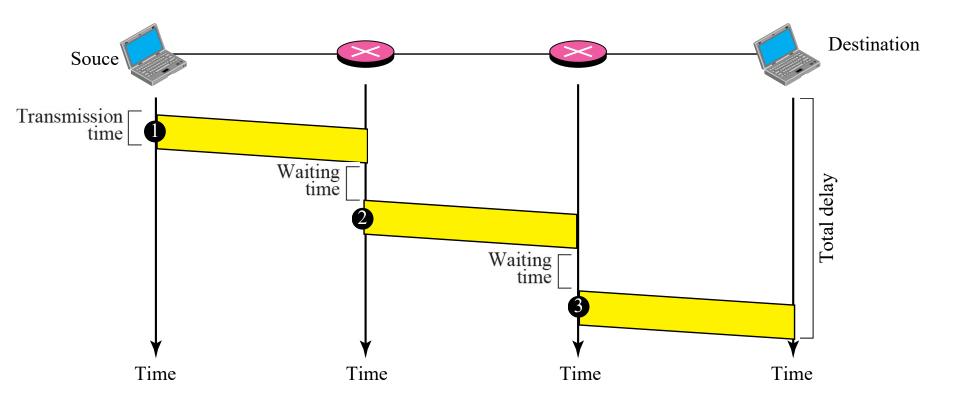
Figure 4.4 Forwarding process in a connectionless network





Bağlantısız paket-anahtarlamalı ağlar'da yönlendirme kararı paketteki hedef adrese göre verilir.

Figure 4.5 Delay in a connectionless network



Sanal Devre Paket Anahtarlama yöntemi

Devre ve datagram anahtarlama tekniklerinin avantajlarından yararlanmak için sanal devreler (virtual circuits) geliştirilmiştir.

Sanal devrelerde, devre anahtarlamada olduğu gibi, bağlantı isteği geldiğinde bağlantının gerçekleneceği yol belirlenir ve bu yol üzerindeki kapasitenin bir bölümü bu bağlantı için ayrılır. Ayrılan kapasite bağlantının ihtiyaçlarına cevap verecek düzeyde olmalıdır. Aksi halde bir devre kurulamaz.

Sanal devreler üzerinde veri paketler halinde taşınır. Ancak bu paketlerin başlık bölümleri paket anahtarlamada kullanılan paketlerin başlık bölümlerinden daha küçüktür. Çünkü kurulmuş bir yol üzerinde yönlendirme yapmak daha kolaydır.

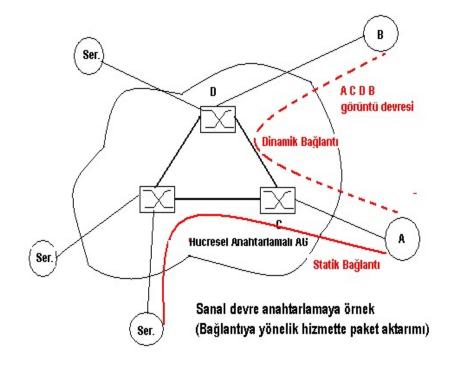
Bu nedenle paketlerdeki ek yük daha az olur. Paketlerin başlıklarındaki yönlendirme amaçlı bilgiler bağlantının kurulması sırasında belirlenir ve daha sonra yaratılan paketlerin başlığında bu bilgiler kullanılır.

Bir sanal devreye ait paketler her zaman aynı yolu takip ederek varış noktasına ulaşır. Paketlerin sırasının değişmesi mümkün değildir.

Bir hat, kapasitesi elverdiğince, pek çok sanal devre tarafından paylaşılır. ATM ve x.25 ağları bu tekniği kullanır.

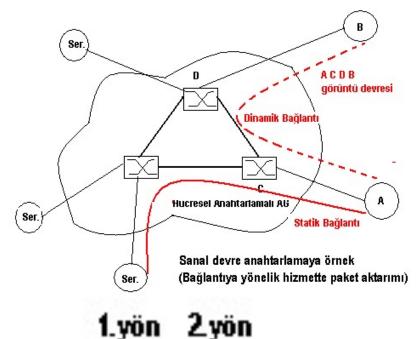
Bir sanal devre, kaynak veya varış düğümlerinden birinin isteği sonucu koparılabilir. Bu durumda, o sanal devre için ayrılmış kaynaklar (hatlar üzerindeki kapasite vd) diğer sanal devrelerin kurulumunda kullanmak için ağa iade edilir.

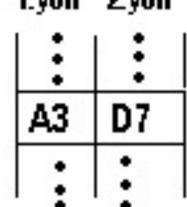
- Gönderici ve alıcı arasındaki sanal devre, göndericinin alıcıya "bağlantı isteği kurma hizmet" paketini gönderimi ile başlatılır.
- Bu mesaj paketinin ağ üzerinde izlediği yol (yörünge), geçtiği tüm aradüğümlerin veritabanlarına işlenir. Bu yörüngeye sanal (görüntü-virtual) devre denir.
- Alıcı bu isteğe olumlu veya olumsuz bir cevap gönderir. <u>Bu mesajlaşmada,</u> <u>max. mesaj boyutu, kullanılacak sanal yol,</u> <u>ağ gecikmesi v.b (QOS) anlaşmalar</u> mevcuttur.
- Bağlantı kurma hizmet paketinin içinde varış ve kaynak IP'leri yanı sıra görüntü devre numarası adı verilen bir alan Burada mevcuttur. oluşan değerler, görüntü devre kurulduktan sonra o devre üzerinden yollanan ilgili bağlantılı her veri paketinin icerisine yerleştirilecektir. Böylece ara düğümler, veri paketlerinin hangi görüntü devreye ait olduğunu, dolayısıyla hangi uçtan gönderilmesi gerektiğini belirlerler.



- Görüntü devre numaraları düğümler arasında farklı değerler alabilir. Örneğin şekildeki A,C,D,B görüntü devresine ilişkin olarak, görüntü numarası AC arasında 3, CD arasında 7, DB arasında 5 olabilir.
- A, ACDB görüntü devresine ilişkin bağlantı kurma istek paketinin içerisindeki görüntü devre no alanına 3 rakamını koymuştur (Çünkü o ana kadar ilk durağı A olan ve C'ye yönlenmiş 0,1,2 nolu başka görüntü devreleride kurulmuştur.

- Bu paket C'ye ulaştığında, C'nin D yönündeki 0,1,2,3,4,5,6 numaralı farklı görüntü devreleri (Bu görüntü devreleri C'den de başlıyabilir, A'dan başlayıp C, D üzerinden farklı yerlere kurulan başka görüntü devreleri de olabilir) önceden kurulmuş olduğundan, C bu yeni görüntü devreye, D yönüne giden başka görüntü devrelerle karışmaması için, kullanılmamış 7 rakamını verecektir.
- D düğümü için de aynı yorum yapılabilir.
- Her bir düğümde, o düğümden başlayan, biten yada geçen her bir görüntü devrenin hangi komşu düğümden geldiği, hangi komşu düğüme gideceğini belirten bir kayıt tablosu vardır.
- Ayrıca o düğümde başlayan veya biten görüntü devrelerde tabloda belirtilir.
- Kayıt tablosuna örnek olarak, bu son kurulan görüntü devreye ilişkin tablo kayıdı şekildeki gibidir.
- Görüntü devreler çift yönlüdür. Yani paketler her iki yöndede giderken aynı görüntü devre numaralarını alırlar.
- Devreyi koparmak isteyen düğüm bağlantı koparma paeketi yayınlar.Karşı düğüm bunu anladığını belirtmek için karşıya gönderdiği paket ile bağlantı yollarını siler.

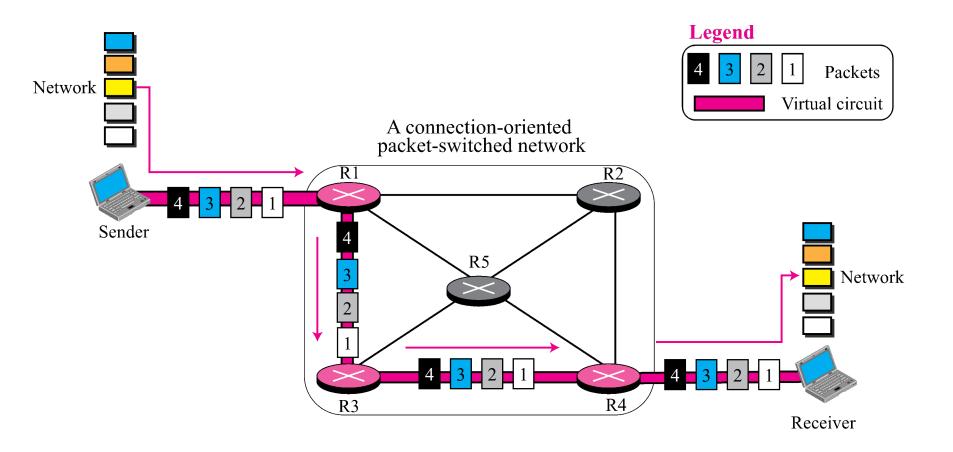




C düğümündeki Görüntü devresi tablosu







Note

Bağlantıya yönelik (Sanal devre) paket anahtarlamalı ağlarda paketi yönlendirme kararı, paketin başlığındaki Label'a göre yapılır.

Figure 4.7 Forwarding process in a connection-oriented network

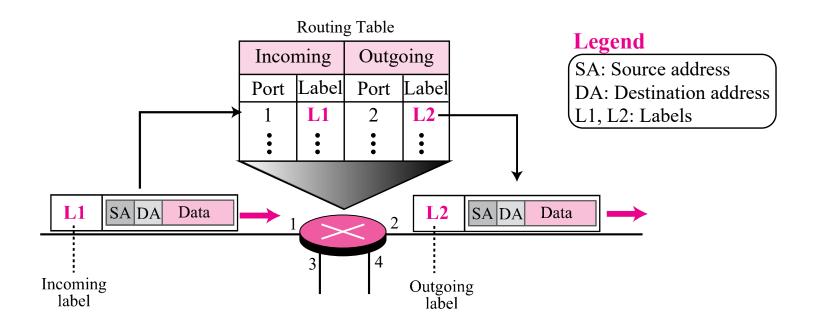
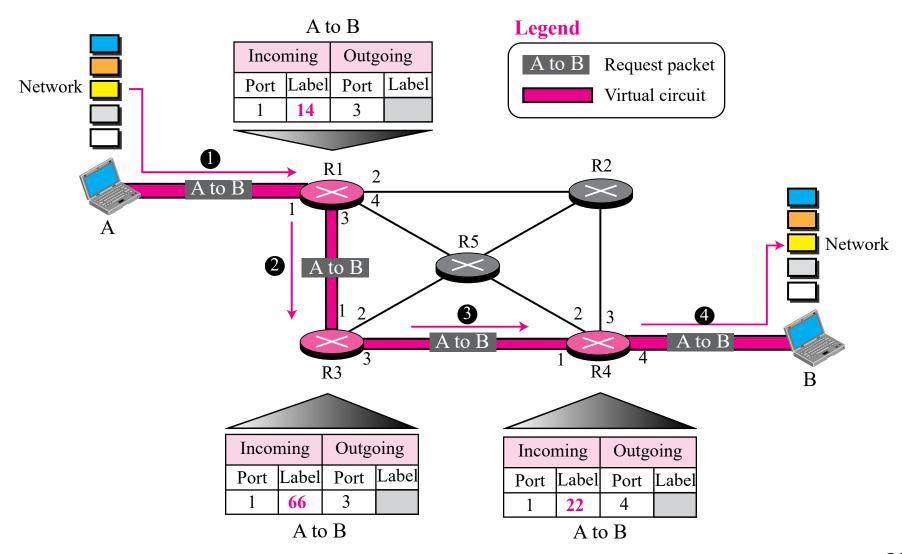
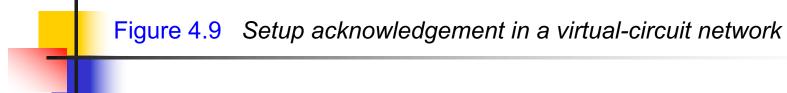




Figure 4.8 Sending request packet in a virtual-circuit network





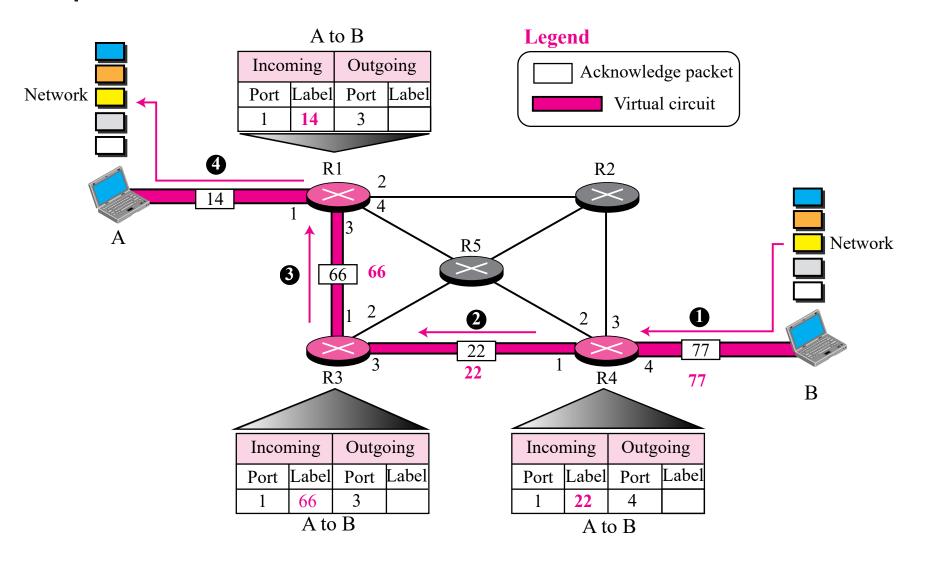


Figure 4.10 Flow of one packet in an established virtual circuit

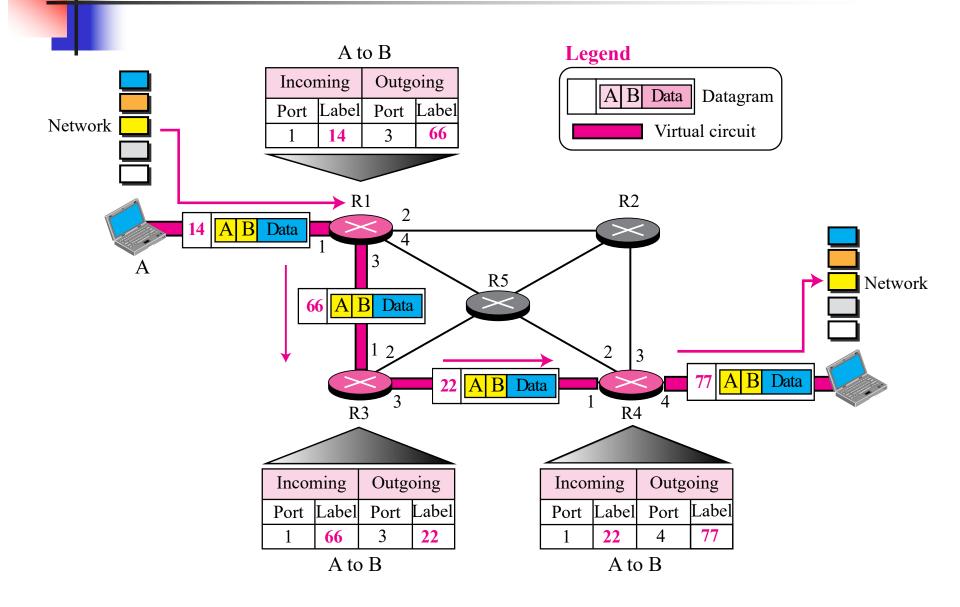
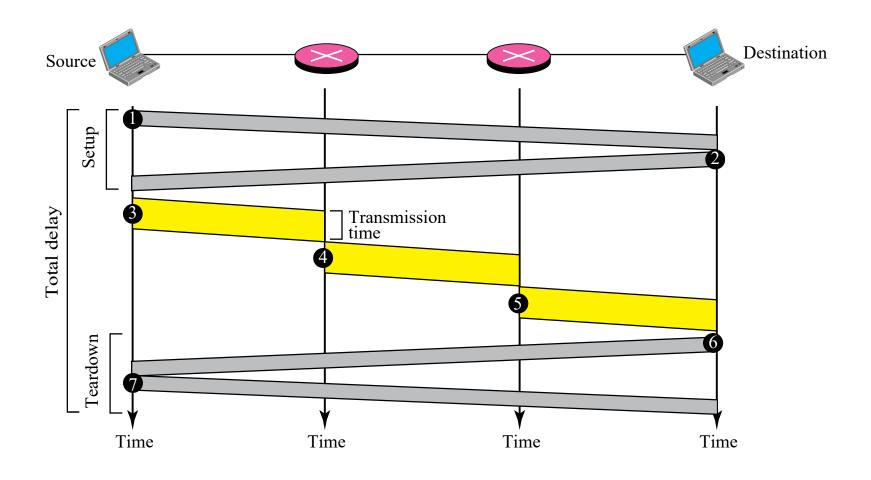


Figure 4.11 Delay in a connection-oriented network



Bağlantısız ve Bağlantıya yönelik hizmetlerin karşılaştırılması

Özellik	Bağlantısız	Bağlantılı
Alıcı ve gönderici adresleri	Her pakette var	Bağlantı kuma hizmet paketinde var
Paket teslim düzeni	Sırasız	Gönderildiği sırada
Routerlarda hata denetimi	Yok	Var
Yönlendiricilerde akış denetimi	Yok	Var
QOS pazarlığı	Yok	Var
Adresleme	Her pakette alıcı gönderici adresi	Paketlerde, görüntü devre no
Yönlendirme	Her paket o andaki ağın durumuna göre yönlendirilir	Paketler önceden kurulmuş gör.devre yörüngesini izlerler.
Düğümün arızalanması	Arızalı düğümdeki paketler yok olur.	Arızalı düğümün üzerinde olduğu tüm görüntü devreler kopar
Tıkanma	Denetim yok	Gör.devre kurulurken gerekli bant genişiliği ayrılır.
Ağ katmanı karmaşıklığı	Fazla değil	Fazla

Yönlendirme

- Yönlendirme (Routing); bir düğümün, alıcı adresi (Mantıksal Adres) belli olan paketi en uygun komşusuna geçirme mekanizmasıdır.
- Bağlantılı düzende çalışan bir ağda bu mekanizma, görüntü devre kurulurken en uygun komşu belirlenir.
- Bağlantısız düzende çalışan ağlarda ise bu mekanizma, her gelen pakette yeniden karar verir.
- Ağda yönlendirme işlemini yapabilmek için yönlendiricilerin (Routerların) bazı bilgilere ihtiyaç vardır.

Bunlar

- Yönlendirilecek paketin hedef adresi,
- Rotaların keşfedilmesi (komşulardan gelen bilgiler),
- En iyi rotanın seçilmesi,
- Yönlendirme tablolarının ve yönlendiriciler arasında devamlılığının sağlanmasıdır.

- Yukarıda belirtilen bilgilerin bir kısmı <u>yönlendirilen</u> <u>protokoller</u> (routed protocol), diğer bir kısmı ise <u>yönlendiren protokoller</u> (routing protocol) tarafından sağlanmaktadır.
- Bu iki kavram birbirinden farklı işlevlere sahiptir. Yönlendirilen protokoller, mantıksal adresleme yapan 3. katman protokollerdir. IP ve IPX bu protokollere örnek olarak verilebilir. Paketlerin hedef ve kaynak adresleri bu prokoller ile öğrenilmektedir.
- Yönlendiren protokoller ise IP ve IPX gibi yönlendirilen protokolleri kendilerine özgü algoritmalar kullanarak yönlendirirler. Yönlendiren protokoller dinamik olarak çalışmaktadır. Fakat "IP ve IPX paketleri" sabit olarak da yönlendirilebilmektedir

Yönlendirme Tabloları

Her yönlendirici üzerinde oluşturulan ve gelen paketi nereye rotalayacığının yazıldığı tablolardır. Bu tabloların doğru bir şekilde oluşturulması önemlidir.

Yönlendirme tabloları temelde *statik* ve *dinamik* olmak üzere iki yaklaşım ile oluşturulur.

Statik yönlendirme tabloları belli bir algoritmaya dayanarak önceden oluşturulur. Bu durumda, bir düğümden diğer düğümlere ulaşmak için kullanılacak yollar önceden bellidir ve ağdaki trafiğin değişiminden etkilenmez.

Dinamik yönlendirme tablolarında ise tabloların zaman içinde, ağ trafiğinde ya da bağlantılarda meydana gelen değişimlerle, güncellenmesi hedeflenir.

- Statik yönlendirme basittir. Güncelleme gerektirmez ancak zaman zaman belli noktalarda oluşan tıkanıklıklar, trafiği farklı yollara yönlendirme imkanı olmadığı için, başarımın düşmesine neden olur. Ya da bazı hatların kopması sonunda, önceden atanmış yolları değiştirmek mümkün olmadığı için bazı düğümler arasında bağlantı kurulamayabilir. Az sayıda ağ için uygundur.
- Dinamik yönlendirmede ise düğümler üzerindeki trafik yükünün artması, tıkanmalar ya da bağlantıların kopması sonucunda alternatif yollar oluşturulur. Ancak, dinamik yönlendirme algoritmaları statik algoritmalardan daha karmaşıktır.

Hataya dayanıklı olması beklenen ağlarda yönlendirme tablolarının dinamik teknikler kullanılarak oluşturulması gerekir.

Yol Belirleme algoritmaları

Yönlendirici; kaynak ile hedef arasındaki en uygun yolu bulabilmeli, en kısa yolu tercih etmelidir. Aslında en kısa yol (mesafe) metrik tanımlamalardan sadece biridir.

Metrik tanımlama olarak basitlik, iletişim maliyeti, sağlamlık ve güvenilirlik, bağlantı hızı, en az düğüm sayısı ve esneklik bilgileridir.

Yol belirleme işlemini gerçekleştiren mekanizmalar ise yönlendirme algoritmalarıdır.

Yönlendirme algoritmalarını iki gruba ayrılır

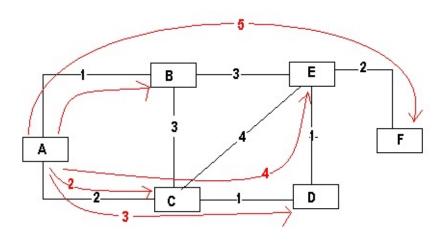
1- Sabit (Statik) Yönlendirme Algoritmaları

- a) En Kısa Yol Algoritması
 - o Dijkstra Algoritması
 - o Bellman-Ford Algoritması
- b) Taşma (Su Baskını) Algoritması
- c) Rastgele Yönlendirme Algoritması
- d) Akış Durumu Yönlendirme Algoritması

2-Uyarlanabilir (Dinamik) Yönlendirme Algoritmaları

- a) <u>Uzaklık Vektörü Algoritması</u>:(Distance Vector)
- b) Bağlantı Durumu Algoritması: (Link State)

Statik yönlendirmeye örnek

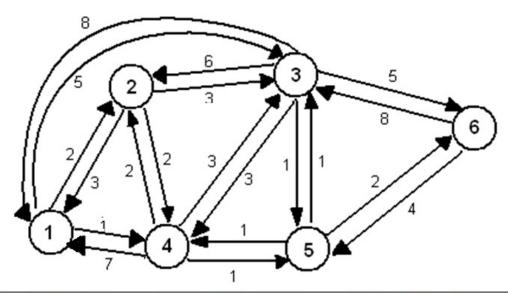


- Şekilde ağda düğümler arasında alternatif yollar mevcuttur. Her yolun iletim süreleri (veya maliyetleri) üzerlerinde verilmiştir. Buna göre düğümler arasındaki en hesaplı yol maliyti aşağıdaki gibi hesaplanır.
- A → B : 1
- A → C : 2
- $A \rightarrow D : (A \rightarrow C \rightarrow D) : 3$
- $A \rightarrow E : (A \rightarrow C \rightarrow D \rightarrow E) : 4$
- $A \rightarrow F : (A \rightarrow B \rightarrow E \rightarrow F) : 6$

A düğümünün yönlendirme tablosu

Alıcı düğüm	Komsu düğüm	Maliyet
В	В	1
C	С	2
D	С	3
Е	С	4
F	В	6

Dijkstra Algoritması (En kısa yolu bulma algoritması)



Düğüm Nol	ktası 1 Planı	Düğüm Nol	ctası 2 Planı	Düğüm Nol	ctası 3 Planı
Hedef Düğüm	Sonraki Düğüm	Hedef Düğüm	Sonraki Düğüm	Hedef Düğüm	Sonraki Düğüm
2	2	1	1	1	5
3	4	3	3	2	5
4	4	4	4	4	5
5	4	5	4	5	5
6	4	6	4	6	5
Düğüm Nol	ktası 4 Planı	Düğüm Nol	ctası 5 Planı	Düğüm Noktası 6 Planı	
Hedef Düğüm	Sonraki Düğüm	Hedef Düğüm	Sonraki Düğüm	Hedef Düğüm	Sonraki Düğüm
1	2	1	4	1	5
2	2	2	4	2	5
3	5	3	3	3	5
5	5	4	4	4	5
6	5	6	6	5	5

Taşkın (Flooding- Su basması) Algoritması

- Bu yöntemde bir düğüme ulaşan paketin kopyaları çıkarılır ve bu kopyalar paketin geldiği bağlantı (hat) dışındaki tüm bağlantılara gönderilir. Doğal olarak bu yöntem aynı paketin pek çok kopyasının yaratılmasına ve bu kopyaların ağdaki trafiği aşırı derecede yoğunlaştırmasına neden olacaktır.
- Oluşabilecek sınırsız artışın önlenmesi amacıyla paketlerin başına bir atlama sayacı (Hop Counter) yerleştirilir. Bu sayacın başlangıç değeri ağdaki maksimum hop sayısına ayarlanır. Bu sayaç her düğümden geçişinde "1" azaltılır ve sayaç değeri "sıfır" olduğunda paket yok edilir
- Bu dezavantaja karşın, taşkın yönteminde seçilecek hat için özel hesaplamalar yapılmasına gerek kalmaz. Paket, doğal olarak, ek kısa yol üzerinden varış noktasına erişir. Ancak bu sırada aynı paketin pek çok kopyası yaratılır. Hatta aynı kopyalar pek çok kez aynı düğümlere ulaşır.

Rastgele Yönlendirme Algoritması (Random Routing Algorithm)

- Bu algoritmada bir düğüme gelen paketin gönderileceği yol, geliş yolu dışındaki yollar arasından rastgele seçilir.
- Tüm yolların seçilme olasılıkları eşittir. Bu algoritmanın biraz gelişmiş modelinde, yol seçimi veri hızına bağlı olarak olasılıklandırılır.
- Rastgele yönlendirme algoritması da "flooding" algoritması gibi ağ bağlantı bilgilerini kullanmaz. Yol seçimi rastgele yapıldığından, seçilen yol en kısa olanı olmayabilir.

Akış Durumu Yönlendirme Algoritması (Flow-Based RoutingAlgorithm)

- Ağ bağlantı bilgilerinin ve trafik akışının birlikte kullanılarak yönlendirme yapıldığı bir sabit yönlendirme algoritmasıdır.
- Temel mantığı ortalama paket gecikmelerinin kuyruk teorisine göre hesaplanmasına dayanır.

Dinamik Yönlendirme Algoritmaları

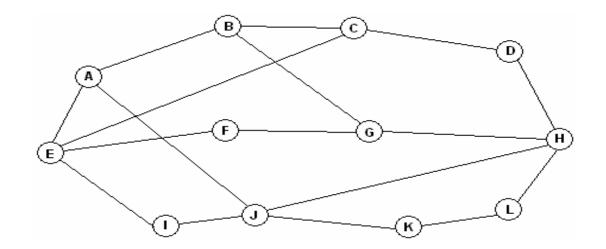
- Düğümler ağ topolojisindeki değişiklikleri ve trafik yoğunluğunu izlyerek o anki durum hakkında karar verirler. Bu yapıda düğümler periyodik olarak kendi durumlarını belirten "durum paketlerini" komşularına göndeririler, gelen bilgilerle de kendi yol tablolarının güncellerler.
 - 1- Uzaklık Vektörü Yönlendirmesi: (Distance Vector Routing)
 - 2- Bağlantı Durumu Algoritması: (Link-State Algorithm)

Uzaklık Vektörü Yönlendirmesi (Distance Vector Routing)

Bu teknikte her yönlendirici bir yönlendirme tablosu tutar bu tabloda ağdaki her yönlendirici için bir satır bulunur. Bu tabloya "vektör" adı verilir.

- Tablonun her bir satırında, herhangi bir yönlendiricinin, tablonun bulunduğu yönlendiriciye olan uzaklık bilgisi (en iyi yol bilgisi) ve ilgili yönlendiriciye, en iyi yol üzerindeki hangi komşu düğüm üzerinden ulaşılacağı bilgisi saklanır.
- Kullanılan ölçütün uzaklık olması gerekmez. Bu ölçüt *gecikme*, *sekme sayısı*, ... olabilir (Metriklerden herhangi birisi).
- Sistem, ilk çalışmaya başladığı anda tüm düğüm noktaları kendilerine komşu düğüm noktalarına özel bir "bilgi paketi" yollayarak komşularının uzaklık bilgilerini edinir. Tüm düğüm noktalarının elde ettiği bu uzaklık bilgileri sonuçta her düğüm noktasının diğer düğüm noktalarına olan uzaklıklarının listelendiği bir tablo hâlini alır.
- Her düğüm, kendisi ile komşu olan düğümler arasındaki uzaklığı bulmak için çeşitli ölçümler yapar. Bu ölçümde ölçüt atlama sayısı (number of hops) seçilmiş ise her düğüm ile komşuları arasındaki uzaklık "bir atlama noktası"nı ifade eder.

- Eğer ölçüt gecikme (delay) olarak seçilmişse her yönlendirici özel yankı (echo) paketleri göndererek gecikme ölçümlerini yapar. Yankı paketini alan düğüm, aldığı paket üzerine alma ve gönderme zamanlarını yazarak geri gönderir. Aynı şekilde komşulardan da benzer bilgiler alır.
- Uzaklık vektörü yönlendirmesinde, her yönlendirici (periyodik olarak) her T milisaniyede bir kendi tablosunda bulunan ölçüt değerlerini komşularına gönderir ve benzer bir tabloyu da komşusundan alır. Gelen tablolardaki verilere bakarak her yönlendirici doğrudan bağlı olmadığı yönlendiriciler ile arasındaki gecikme değerlerini ve o yönlendiricilere nasıl ulaşacağını bulabilir.
- Bir sonraki ağ konfigürasyonunu inceleyelim.

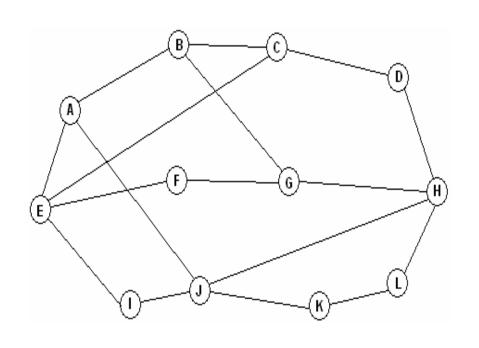


Şekildeki ağı göz önüne alalım. B yönlendiricisinin yaptığı ölçümler sonucunda, C ile arasındaki gecikme 13 ms olarak bulunmuş olsun ve komşusu C'den aldığı C-D yolu gecikmesinin de 15 ms olduğu varsayılsın.

B yönlendiricisi bu bilgileri kullanarak B'den D'ye olan gecikmeyi 13+15=28 ms olarak hesaplayacaktır.

Bu hesaplamalar tüm yönlendiriciler için yapılır ve en az gecikmeli yolun gecikme süresi ve bu yola hangi komşu yönlendirici üzerinden erişilebileceği bilgisi yönlendirme tablosuna kaydedilir. Yönlendirme tablosundaki eski değerler bu hesaplamalara katılmaz ve hesaplamalar sonucunda bu tablolar güncellenir.

Komşu yönlendiricilerden J yönlendiricisine gelen gecikme vektörleri (ölçüm sonuçları)



Varış	A	Н	I	K
J'den Gecikme (ms)	8	12	10	6

Tablo 2.3: J'nin kendi ölcüm sonucları

Varış	A'dan Gecikme (ms)	Varış	H'den Gecikme (ms)	Varuș	l'dan Gecikme (ms)	Varış	K'den Gecikme (ms)
A	0	A	20	A	24	A	21
В	12	В	31	В	36	В	28
С	25	С	19	С	18	С	36
D	40	D	8	D	27	D	24
E	14	E	30	E	7	E	22
F	23	F	19	F	20	F	40
G	18	G	6	G	31	G	31
H	17	Н	0	Н	20	H	19
I	21	I	14	I	0	I	22
J	9	J	7	J	11	J	10
K	24	K	22	K	22	K	0
L	29	L	9	L	33	L	9

Uzaklık vektörü algoritması, yukarıdaki tablo verilerini kullanarak tüm yönlendiricilere hangi yol üzerinden erişebileceğini ve en kısa gecikmenin hangi değerde olduğunu hesaplayarak yönlendirme tablosunu günceller.

Verilen bu örnekte J'nin yönlendirme tablosu şu şekilde oluşturulur:

J'den A'ya gitmek için en iyi yol seçimi ;

A üzerinden : JA = 8 ms gecikmeli yol ile (Minimum),

H üzerinden : JA = JH + HA = 12 + 20 = 32 ms gecikmeli yol ile,

I üzerinden : JA = JI + IA = 10 + 24 = 34 ms gecikmeli yol ile,

K üzerinden : JA = JK + KA = 6 + 21 = 27 ms gecikmeli yol ile,

Sonuçta en az gecikme A'ya doğrudan giden 8 ms gecikmeli yoldur.

J'den B'ye gitmek için en iyi yol seçimi ;

A üzerinden : JB = JB + BA = 8 + 12 = 20 ms gecikmeli yol ile (Minimum),

H üzerinden : JB = JH + HB = 12 + 31 = 43 ms gecikmeli yol ile,

I üzerinden : JB = JI + IB = 10 + 36 = 46 ms gecikmeli yol ile,

K üzerinden : JB = JK + KB = 6 + 28 = 34 ms gecikmeli yol ile,

Sonuçta J'den B'ye en az gecikme A üzerinden giden 20 ms gecikmeli yoldur.

J'den C'ye gitmek için en iyi yol seçimi ;

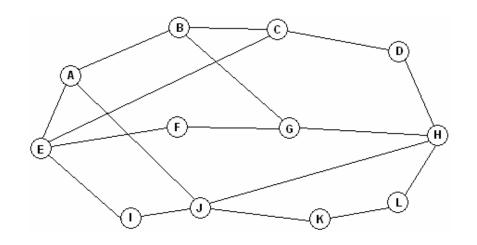
A üzerinden : JC = JA + AC = 8 + 25 = 33 ms gecikmeli yol ile,

H üzerinden : JC = JH + HC = 12 + 19 = 31 ms gecikmeli yol ile,

I üzerinden : JC = JI + IC = 10 + 18 = 28 ms gecikmeli yol ile (Minimum) ,

K üzerinden : JC = JK + KC = 6 + 36 = 42 ms gecikmeli yol ile,

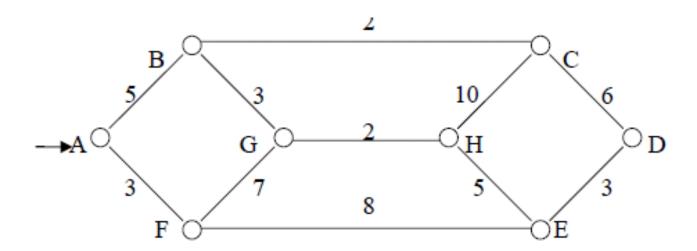
Sonuçta J'den C'ye en az gecikme I üzerinden giden 28 ms gecikmeli yoldur. Bu işlemler tüm yönlendiriciler için tekrarlanarak aşağıdaki yönlendirme tablosu oluşturulur.



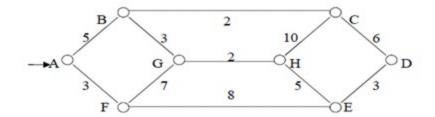
Varış	A	В	C	D	E	F	G	H	I	J	K	L
J'den Gecikme (ms)	8	20	28	20	17	30	18	12	10	0	6	15
En İyi Yol	A	A	I	Н	I	I	Н	Н	I	•	K	K

Tablo 2.4: J'nin oluşturduğu yeni yönlendirme tablosu

Örnek Uygulama



- Şekildeki topoloji üzerindeki değerlerin bir anlık gecikme değerleri olduğunu ve ağın çok çok kısa bir süre önce aktif hale geldiğini varsayalım.
- Bu durumda her yönlendirici sadece komşusuna olan gecikmeyi bilsin ve diğer yönlendiricilere nasıl ulaşılacağı konusunda bilgi sahibi olmasın.
- Yönlendiricilerin tabloları bir sonraki slayttaki gibi olacaktır.



	* *	***					٠	
A	Y	ön	lei	ıd	11	пc	1	S

	gecikme	Bir- sonraki
A	0	-
В	5	В
C	00	-
D	00	-
E	00	-
F	3	F
G	00	-
H	00	-

B Yönlendiricisi

	gecikme	Bir- sonraki
Α	5	A
В	0	-
С	2	С
D	00	-
E	00	-
F	00	-
G	3	G
Н	00	-

C Yönlendiricisi

	gecikme	Bir- sonraki düğüm
A	00	-
В	2	В
C	0	-
D	6	D
E	00	-
F	00	-
G	00	-
H	10	Н

D Yönlendiricisi

	gecikme	Bir- sonraki düğüm
A	00	-
В	00	-
C	6	C
D	0	-
E	3	E
F	00	-
G	00	-
H	00	-
G	00	-

E Yönlendiricisi

	gecikme	Bir- sonraki düğüm
A	00	-
В	00	-
C	00	-
D	3	D
E	0	-
F	8	F
G	00	-
Н	5	H

F Yönlendiricisi

	gecikme	Bir- sorraki düğüm
A	3	A
В	8	-
С	00	-
D	00	-
E	8	E
F	0	-
G	7	G
H	00	-

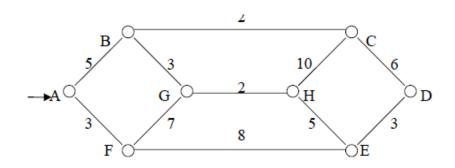
G Yönlendiricisi

	gecikme	Bir- sonraki dūgūm
A	00	-
В	3	В
C	00	-
D	00	-
E	00	-
F	7	F
G	0	-
Н	2	Н

H Yönlendiricisi

	gecikme	Bir- sonraki dūžūm
A	00	-
В	00	-
C	10	С
D	00	-
E	5	Е
F	00	-
G	2	G
H	0	-

- Bu aşamada B, F ve H yönlendiricilerinden gelen tablolara göre G yönlendiricisinin kendi yönlendirme tablosunu güncelleyişi aşağıdadır.
- •G yönlendiricisi B, F ve H'den kendisine ulaşan vektörlere göre yönlendirme tablosundaki değerleri günceller ve doğrudan bağlı olmadığı yönlendiricilere olan gecikmeyi hesaplar. Güncelleme sırasında amaç en kısa gecikmeye sahip çıkış hattını bulmaktır. Bir yönlendiriciye ulaşmak için alternatif yollar olabilir ancak bu yolların en kısası seçilmelidir. Örneğin, G yönlendiricisi, A'ya F (7+3) ya da B (3+5) üzerinden ulaşabilir. Ancak B üzerinden gidilen yol daha kısa olduğu için onu tercih eder.
- •Uzaklık vektörlerinin aktarılmasından sonra G'nin doğrudan bağlı olmadığı A, C ve E yönlendiricilerine olan gecikme değerlerine sahip olduğunu görmekteyiz.



G yönlendiricisindeki tablonun güncellenmesi

Yön	lendi	ricisi	F Yön	lendir	ricisi	H Yön	lendi	ricisi	G Yöi	ılendi	ricisi
	gecikme	Bir- sonraki düğüm		gecikme	Bir- sonraki düğüm		gecikme	Bir- sonraki düğüm		gecikme	Bir- sonraki
Α	5	A	A	3	A	A	oo	-	A	3+5	В
В	0	2	В	00	-	В	00	-	В	3	В
С	2	С	C	00	-	C	10	С	C	3+2	В
D	00	-	D	00	-	D	00	-	D	-	-
Е	00	-	Е	8	Е	Е	5	Е	E	2+5	Н
F	00	-	F	0	-	F	00	-	F	7	F
G	3	G	G	7	G	G	2	G	G	0	1
Н	00	-	Н	00		Н	0		Н	2	Н

Link State Algoritması

- •Uzaklık vektörü algoritmasının başlıca iki sorunu vardır:
- Birincisi yön seçimi yaparken sadece kuyruktaki gecikmeleri ölçüt olarak alması, hat kapasitesini hesaba katmamasıdır.
- •İkinci sorun ise uzaklık vektörü algoritmasının sonuca ulaşmasının (converge) uzun süre almasıdır.
- Bağlantı durumu algoritması bant genişliğini de dikkate alarak daha etkin bir yönlendirme tablosu elde edilmesini sağlar.
- Ayrıca bağlantı durumu algoritmasında yönlendirme tabloları uzaklık vektörü algoritmasından farklı olarak sadece komşu yönlendiricilere değil tüm yönlendiricilere yollanır.

LİNK STATE algoritması

Bağlantı durumu yönlendirmesi de dinamik bir tekniktir. Amacı, topolojideki değişimlere kolayca adapte olmak ve trafikteki değişimlere göre gerektiğinde alternatif yollar bulmaktır.

Bağlantı durumu yönlendirmesini, en kısa yol bulma (Dijkstra) algoritmasının uygulaması olarakda düşünülebilir. Burada, yönlendiriciler bağlı oldukları hatlar(komşuları ile aralarındaki) üzerindeki gecikmeleri gösteren verileri ağdaki tüm yönlendiricilere ulaşması için gönderirler.

Bu verileri alan yönlendiriciler ağ topolojisini oluşturur ve diğer yönlendiricilere en kısa yoldan ulaşmak için bu topoloji üzerinde en kısa yol algoritmasını çalıştırırlar. Uygun yolları belirlerler.

Hat durumunu gösteren paketler, en kolay taşkın yöntemi ile yayılabilir. Bu durumda paketlerin gereksiz yere ağ içinde dolaşmasını önlemek için hat durumu paketlerinin üzerine yaş (sekme sayacı) alanı koymak anlamındadır.

Bağlantı durumu algoritmasının çalışması aşağıda anlatılmıştır:

Her yönlendirici ilk açıldığı zaman kendisine komşu olan düğüm noktalarını ve adreslerini öğrenir, bu işlemi bağlı olduğu tüm düğümlere "selamlama (HELLO)" paketi göndererek ve gelen yanıtları inceleyerek yapar.

Komşu düğüm noktalarına olan uzaklıkları, gecikme zamanlarını ve maliyetlerini hesaplar, bu işlemi bağlı olduğu tüm düğümlere "yankı (ECHO)" paketi göndererek yapar.

Bu bilgiler ışığında bir yönlendirme tablosu hazırlar. Bağlantı durum tablosu adı verilen bu yönlendirme tablolarının hazırlanması kolaydır. Fakat bu tabloların ne zaman hazırlanacağına karar vermek gerekir.

Bu tablolar, bazen belirli aralıklarla bazen de ağa düğüm eklenip çıkarıldığı zamanlarda yeniden hazırlanır. Bu hazırlanan tablolar bir paket hâlinde, "Su Baskını (Flooding) Algoritması" kullanılarak tüm düğüm noktalarına yollanır.

Dijkstra algoritması uygulanarak tüm düğümlere olan en kısa yollar hesaplanır.

Gönderilen bu paketler sonunda her düğüm noktası kendisi için en uygun yönlendirme tablosunu hazırlar.

Uzaklık Vektörü Algoritması	Bağlantı Durumu algoritması			
Ağın topolojisini komşularının açısından görebilir.	Tüm ağın topolojisini elde edebilir.			
Uzaklık bilgileri yönlendiriciden	Diğer yönlendiricilere olan uzaklıklar her			
yönlendiriciye toplanır.	yönlendiricide hesaplanır.			
Yönlendirme tabloları periyodik olarak	Yönlendirme tabloları topoloji			
güncellenir.	değişikliklerinde güncellenir.			
Yönlendirme tabloları sadece komşulara	Yönlendirme tabloları tüm yönlendiricilere			
gönderilir.	gönderilir.			

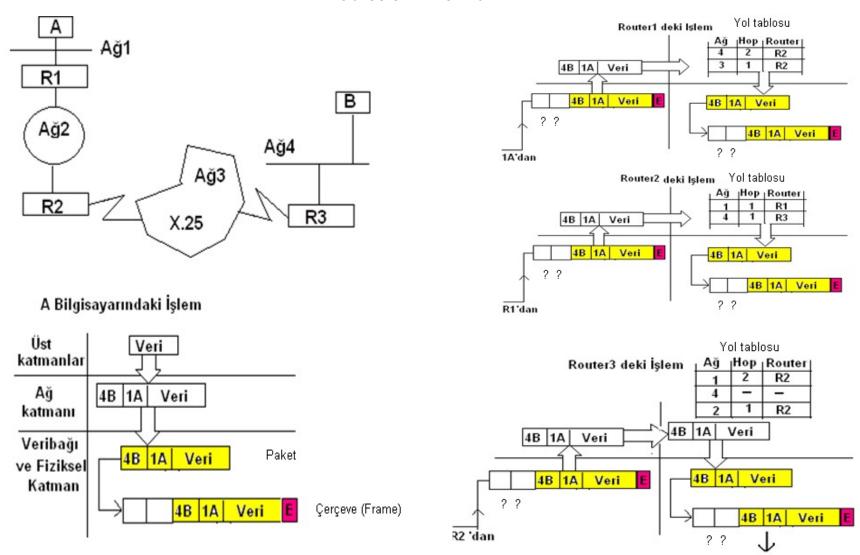
Yönlendirme protokolları

- •Yönlendirme protokolları, yönlendirme bilgilerinin ağ üzerindeki alış verişini kolaylaştırdıkları gibi, yukarıda bahsedilen algoritmalara göre tablo oluşturmak için kullanılırlar.
- •WAN sistemlerinde önem kazan bu protokollardan IGP,RIP,IGRP,EIGRP,OSPF,EGP,BGP v.b'ler ilerleyen haftalarda incelenecektir.

- Paketlerin yönlendirilmesi (geçilecek düğümlerin belirlenmesi) iki şekilde gerçeklenir :
- 1- Kaynakta yönlendirme(source routing)
- 2-Sekerek yönlendirme (hop by hop routing).
- Kaynakta yönlendirmede kaynak düğüm, paketin sırası ile geçeceği düğümleri (yönlendiricileri) belirler ve bu bilgi pakete eklenir. Yönlendiriciler bu bilginin bulunduğu özel alana bakarak paketi sırası ile geçmesi gereken düğümlere aktarırlar. Bu yöntemde, yönlendirme işlemi kaynak düğümde yapılır. Diğer düğümler paketin belirlenen yol üzerinden geçmesini sağlar.

- Sekerek (hop bay hop)yönlendirmede ise paketin üzerindeki varış düğümü adresine bakarak paketin gönderileceği bir sonraki düğümün adresi belirlenir ve paket o düğüme aktarılır.
- Yol üzerindeki her sekmede (düğümde) bu işlem yapılır ve paket varış düğümüne kadar ulaştırılır.
- Bu işlemler yönlendirme tabloları ile gerçekleştirilir.

Ağ1'deki A Bilgisayarı, **Ağ4**'deki B bilgisayarıyla iletişim kuracaktır. Bu süreçteki olacakları şematik olarak gerçekleştiriniz. Routerların MAC adresleri sırasıyla **R11,R12**, **R22**, **R23,R33,R34**' dür. Bilgisayarların MAC adresleri A ve B'dir.



59