

**2.** Routing algoritmalar iki gruba ayrılır. Bunlar dinamik yönlendirme ve statik yönlendirme gruplarıdır.

**Statik yönlendirme algoritmaları :** Bu tür algoritmalar için, yönlendirme kararı mevcut trafik ve topolojinin ölçümüne veya tahminlerine dayanmamaktadır. Ancak rotanın seçimi önceden yapılır.

**Dinamik yönlendirme algoritmaları:** Bu algoritmalar için, topoloji veya trafikte herhangi bir değişiklik olduğunda yönlendirme kararı değiştirilebilir.

**Statik algoritma örnekleri:** Shortest path routing, Flooding ve Flow Based Routing tir.

**Dinamik algoritma örnekleri:** Distance vector Routing Algorithm ve Link State Routing tir.

### **Shortest path routing**

Ağ topolojisi ve ağdaki her bağlantıya veri gönderme maliyetini açıklayan bir dizi ağırlık verildiğinde belirtilen bir kaynaktan ağdaki diğer tüm hedeflere giden en kısa yolu bulma. Kaynaktan gelen geçici mesafe, çalışma düğümüne bitişik tüm düğümlere hesaplanır. Bu, mevcut geçici mesafeden daha kısa ise hedefin geçici mesafesini değiştirilir ve orada çalışan düğüm kaydedilir.

### **Flooding**

Bu algoritmada, gelen her paket, geldiği hat hariç her giden hatta gönderilir. Flooding dezavantajı, çok sayıda kopya paket oluşturmaktır. Aslında, süreci bir şekilde terk etmedikçe sonsuz sayıda yinelenen paket üretir. Bu algoritmada her gelen paket her çıkış hattına gönderilmez. Bunun yerine, paket sadece yaklaşık olarak doğru yönde giden hatlara gönderilir.

### **Flow Based Routing**

Akışa dayalı yönlendirme, statik yolları belirlemek için ağ topolojisi, trafik matrisleri ve kapasite matrislerini kullanır.

### **Distance vector Routing Algorithm**

Bu algoritmada, her bir yönlendirici, vektör olarak adlandırılan bir tabloyu temsil eder, böyle bir tablo, her varış noktasına en iyi bilinen mesafeyi ve oraya ulaşmak için hangi hattın kullanılacağı hakkında bilgi verir. Bu durumda, her yönlendiricinin ağdaki diğer tüm yönlendiricilerin kimliğini bildiğini, ancak her yönlendiriciye en kısa yolun bilinmediğini varsayıyoruz. Distance vector Routing Algorithm Bellman-Ford yönlendirme algoritması olarak da bilinir.

### **Link State Routing**

Her yönlendirici komşularını bulmalı ve ağ adreslerini almalıdır. Sonra bu komşuların her birine gecikme ya da maliyeti ölçmelidir. Ağ adreslerini ve tüm komşuların gecikmelerini içeren bir paket oluşturmalıdır. Bu paket diğer tüm yönlendiricilere gönderilir. Diğer yönlendiricilere en kısa yol hesaplanır.

Bazen ağ o kadar büyük olur ki, yönlendirici tablasının boyutu çok büyük olur ve hemen hemen her yönlendiricinin diğer tüm yönlendiriciler için bir girişi olması imkansız hale gelir. Ardından telefon şebekelerinde kullanılan gibi hiyerarşik yönlendirme benimsenmelidir.

**3.a.** 192.1.0/22 and 192.1.4/24.

**b.** Longest prefix match kullanarak, 1.2.3.5 kullanılmalıdır.