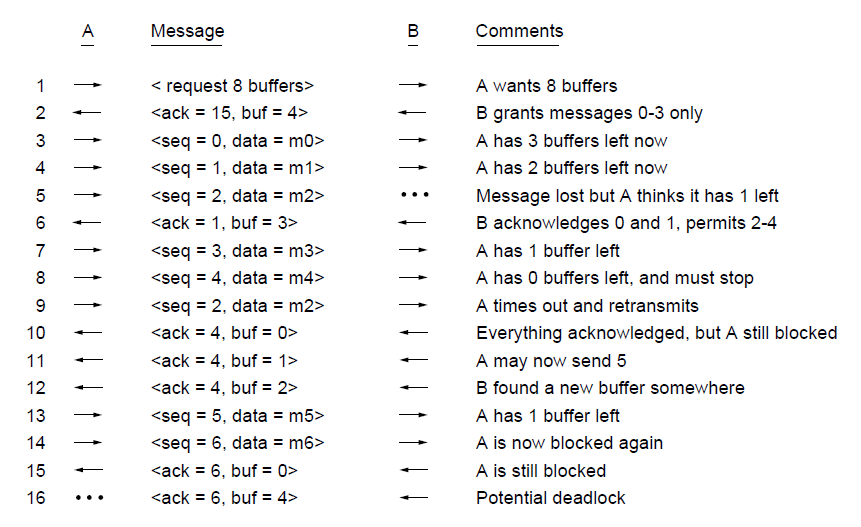
1. Aşağıdaki olay deadlock a sebep olurmu olursa nasıl bir önlem alınmalıki deadlock önlensin



Transpor layerların haberleşmesindeki buffer yönetimindeki akıştır. Buffer managment yöntemlerine göre dinamik buffer yönetminin sayısının allokate i

A b ye bilgi yollyacak a diyorki b ye 8 buffer a ihtiyacımız var bu b ye 8 buffer ihtiyacı olarak geliyor ama oanda b nin 8 buffer lık yeri yok

B diyorki ben sana en fazla 4 buffer verebilirim 0,1,2,3 ben senin banayolamış oldugunun 15 numaralı segmenteacknowledgeediyorum onu aldım ve senin buffer ihtyacın olarak 4 buffer veriyorum acknowledge içerisinde bir başka bilgi daha bu da piggybacking .

0 da a nın 3 tane daha yollama hakkı kaldı A çetere tutuyor. Bir taen daha yolluyor 2 bir tane daha yolluyor 1 ama ne yazıkki sequence 2 yolda kaybalıyor ama A nın bundan haberi yok . data 2 kabaluyor geriye 1 tane buffer kaldığını düşünüyor.

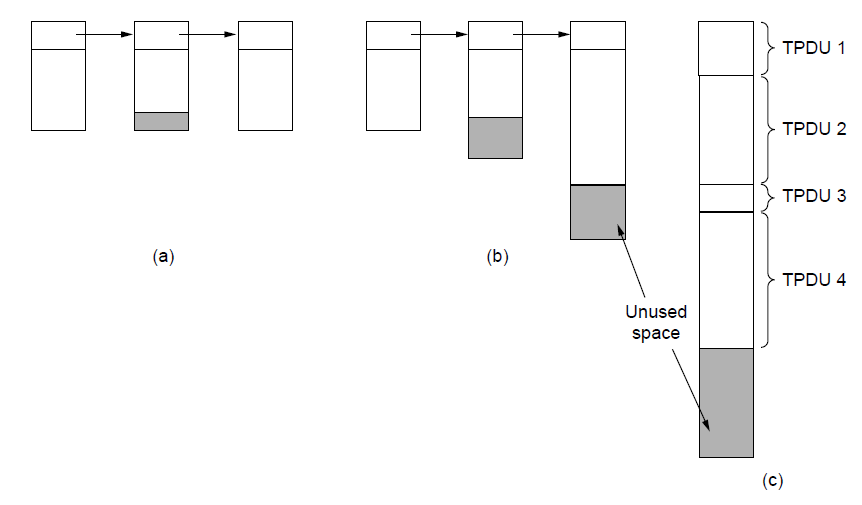
B 0 ve 1 i acknowlege ediyor 2 tanesini doğrulattı ack 1 (0ve 1 ii ack ettim)

3 ü yolladı 4 dü yolladı orda artık A b tarafında artık bufferolmadığını biliyor orda durmak zorunda.A nın B tarafında kullanabileceği buffer kalmadı B o buffer ları boşaltamamış durumda segmenler sıralı olamak zorunda 2 nolu segmenti tekrar yolluyamıyor çünkü B de artık buffer yok

B buffer boşa çıkartamıyor segmentler sıralı gelmiş durumda transportta.**(deadlock)**

**Bir transport layer her zaman için karşı tarafa announceettiğinden daha fazla bufferının olması gerekiyor. Yani A onu yoluyacak B alt tarafta onu görecek ona göre A yadevam edebilirsin dicek.**

1. Eror control flowcontrol



3 paket olsun 1 geldi 2 geldi 3ü bekliyoruz. Eksik olanın yollanması 2 numaralının gelmesini beklemek demek 1 ve 3 ü bufferlamak gerek bu yüzden buffer management a ihtiyacımızvar.

Gelen bir segmentin oldugu gibi o bufferiçerisine sığması gerekiyor mtu dan ufak bir segment geldiğinde buffer o kadar boşta kalır bu durumda Sabit olmalı (mTU) = sabit ise kayıp **memory kullanımı en aza inmiş olur. Flexible değil max transmission unit olmalı ve mümkünse bütün transmiison unit ler bu mtu ile aynı olamalı yani Transportlayer nin paket ortalama boyut ile maxımımum boyutun eşit olması durumunda düşünülmüş şey ( a -Chained fixed-size buffers)**

**Farklı boyutlarda buffer ayarlanır ortalama değer ile maxdeğer her zaman eşit olması mümkün değil o yüzden max degere denk düşecek tarza bütün bufferları açarsak ortalama değer kadar olursak 1. Yönteme gore b ufferların büyük bir kısmı kullanılmıyor olacak.**

**Interneal fragmentation I en aza indirmek için gelen paketi en uygun yere yerleştirmemiz gerekir.**

**Herbiri farklı boyutta değil 1000 byte 1500 byte var**

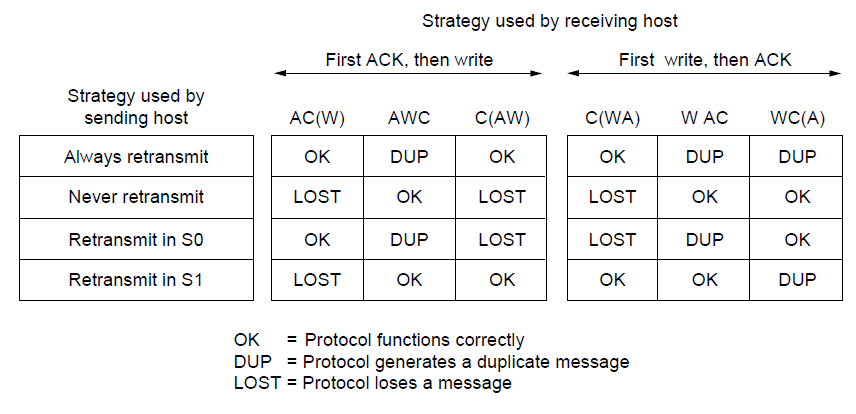
1. **A chained fixed size buffer boyutu sabitve MTU kadar olmalıdır.**
2. **Transport layer dinamik olarak memory allocate etmez zaman kaybı temizlenecek buffer sayısı kadar garbage collecteor sayısı lazımdır.**
3. **One large circular buffer per connection genelde datalink layer kullanılır.**
4. **Önce yukarya yazıp clienta yolladığımız da mesajın duplicate olmasına neden olur**

**önce client a acknowledge yollayıp yukarıya aktarsak dersek eksik bir mesaj olur**

**çözüm: bir üst katman daha olmalıki o rası neyin gelip gelmediğinin farkına varabilsin.**

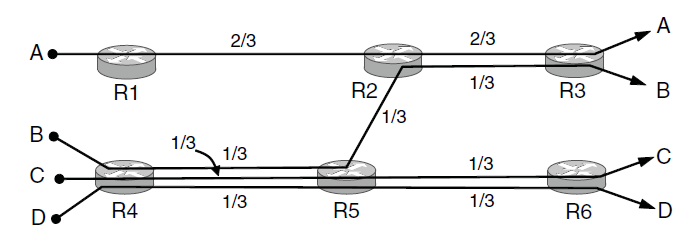
**Session layer transport layer ın görüşmesi durumunda bu olayı kurtarır.**

**Application göstüğünde uygulamanın kendi kendini toparlama şansı yok .**

****

**Recoverynin üstkatmana application layer a aktrılması.**

1. **Network layerload shading yaptığında transport layer kendini re transmissionlara ayarlamak zorunda .**
2. **Tp açısından 2 noktada congesiton htiyaç duyulan bant genişliğinin ayarlanabilmesi . Diğer taraftan network layer ın admission yada load shadeing yapması kontrol yapmasının sonucu transport layer a dayanır Tp de segment çıkarma kendisinden netwokr layer a bilgi gönderme hızını regule etmeli**
3. **Congestion ölçüm kriterlerlerinden biride gecikmedir asimtotik sonsuza gider.**
4. **TP layer için 2 trans port layer ın karşılıklı konuşması için END TO END(bis sürü atlama noktası) olmalı . TP layer Da congesitton kriteri olarak END TO END DELAY i kabul eder acknowledge bilgisinin gidip gelme süresi nedir . buna bakar. ACK uzadıkça network layerdan**
5. **VBR = 1/3 (B C D)**

****

**Amaç soldaki ağdan sağdak i a ya erişmek**

**Mevcut bandwith 3 te bir olarak paylaşmışlar R5 ile R6 arasında yine 3 te 1 oranında paylaşıyorlar fakat bir taner akış eksik çünkü B ye erişmek için önce R’ sonra R3 e gidilmesi gerekiyor. Waited fair queu (Desirablebandwith allocation) ağırlıklarına göre kuyruklama.**

1. **Transport layer da end to end communications . Datalink layer da linki iki ucunda yaptığımız flow kontrolü transport layer da end to end yapıyoruz.**
2. **Explicit : açık bir şekilde congesiton oluşacağını söylüyoruz**