



Bölüm 12: Çoklu İşlemcili Sistemler

İşletim Sistemleri



Çoklu İşlemcili Sistemler

- Bellek ve G/Ç kaynaklarını paylaşan çok işlemcili bilgisayar sistemi.
- **Simetrik Çoklu İşlemci (SMP):**
 - Her işlemci, paylaşılan kaynaklara eşit düzeyde erişime sahiptir.
- **Asimetrik Çoklu İşlemci (AMP):**
 - Bir işlemci ana işlemci, diğerleri yardımcı işlemci görevi görür.
- **Küme Sistemleri:**
 - Tek bir sistem olarak birlikte çalışan,
 - Birbirine bağlantılı (*connected*) bağımsız sistemler kümesi.



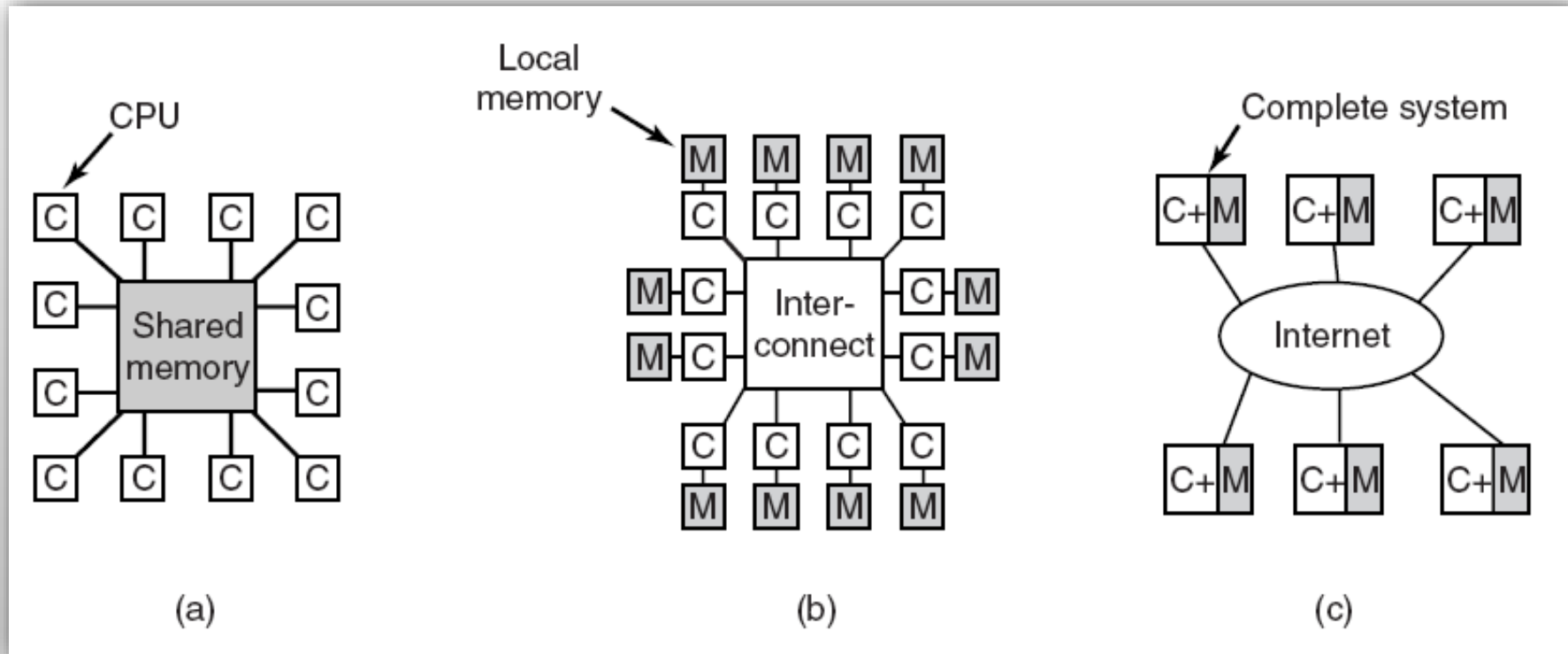
Çoklu İşlemcili Sistemler

- İşlemci ve kaynakları bağlamak için *ağ ve veri yolu* kullanılır.
- İşletim sistemi,
 - İşlemci tahsisi, senkronizasyon ve süreçler arası iletişimi yönetir.
- Daha fazla işlemci eklenerek, performans artırılabilir.
- Paylaşılan veriler güncel ve tutarlı olmalı.
- Görevler işlemciler arasında dağıtılır.
- Aynı anda birden çok süreç yürütebilmeli.



Çoklu İşlemcili Sistemler

- (a) Paylaşımlı bellek kullanımı. (b) Mesaj iletimi ile haberleşme.
- (c) Geniş alana dağıtılmış bir sistem.





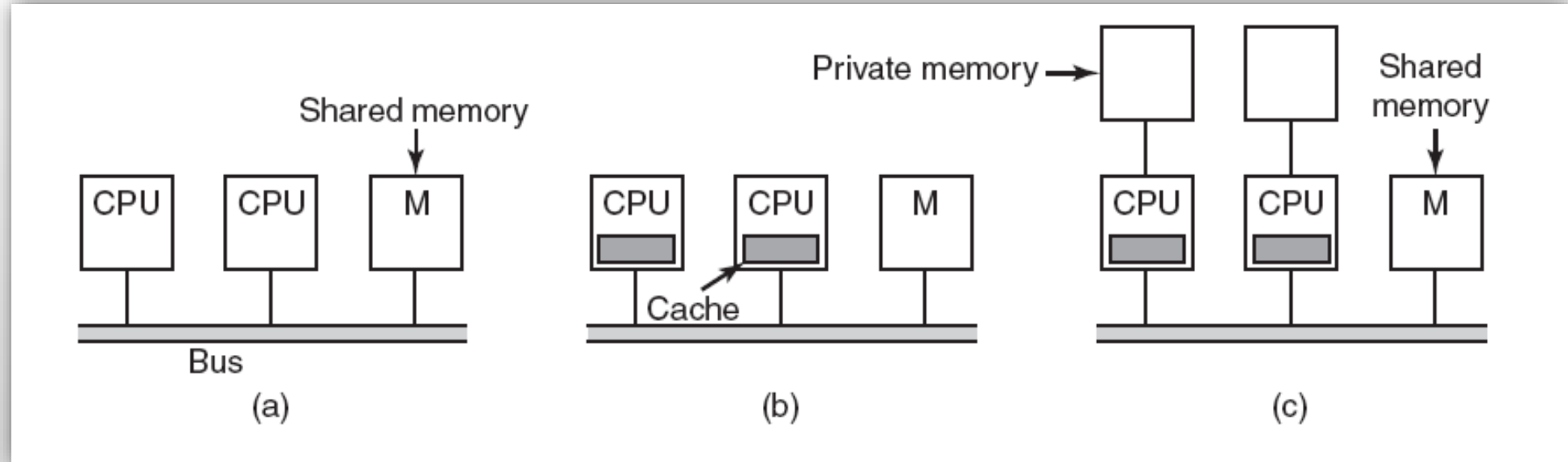
Veri Yolu Tabanlı Mimariler

- **Tekdüzen Bellek Erişimi:** (*UMA Unified Memory Access*)
 - İşlemciler paylaşımlı belleğe eşit erişim süresine sahiptir.
 - Paralel algoritmaların programlanmasını basitleştirir.
- **Önbellek:** (*cache memory*)
 - İşlemciye yakın yerleştirilmiş küçük ve hızlı bellek.
- **Paylaşımlı Bellek:** (*shared memory*)
 - İşlemcilerin aynı belleğe erişimi vardır.
- **Özel Bellek:** (*local memory*)
 - İşlemcinin kendisine ait bellek.
 - Yerel değişken ve verilerin saklanması için kullanılır.



Veri Yolu Tabanlı Mimariler

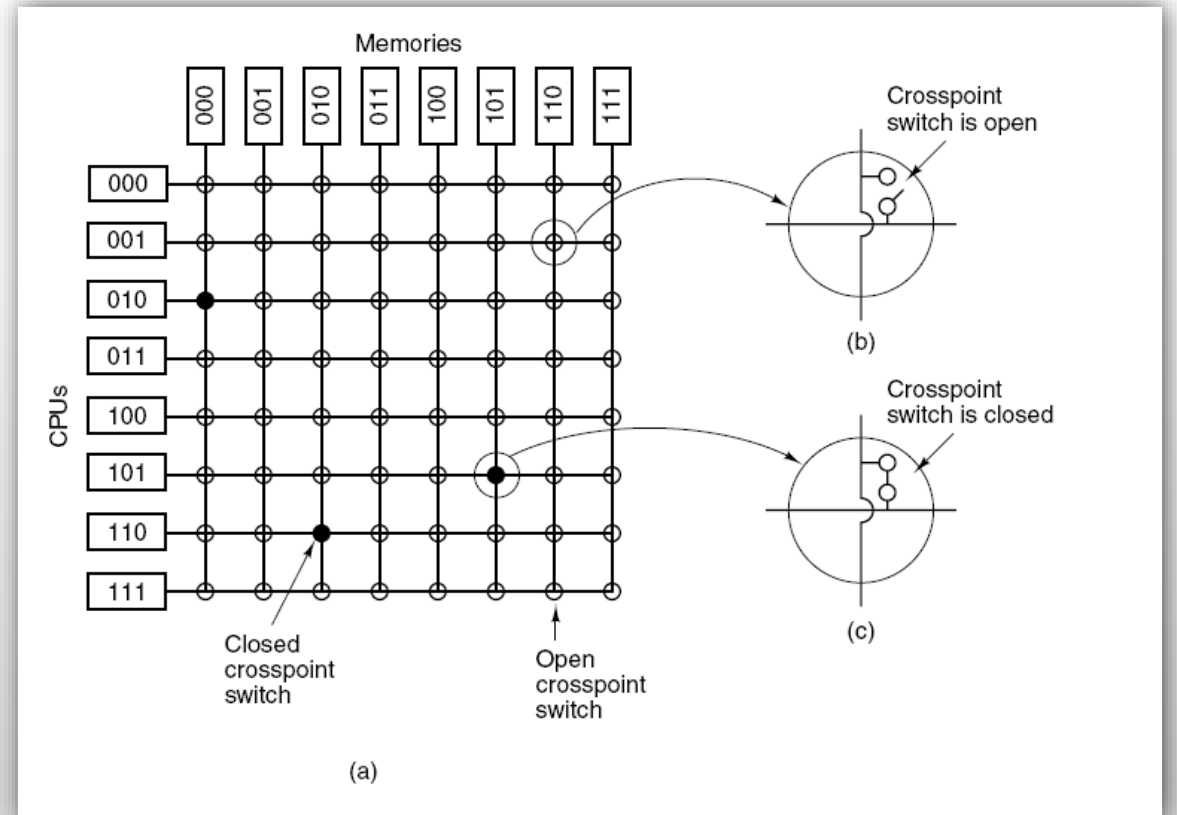
- (a) paylaşımlı bellek. (b) önbellek. (c) özel bellek.





Çapraz Çubuk (Cross Bar) Anahtarları

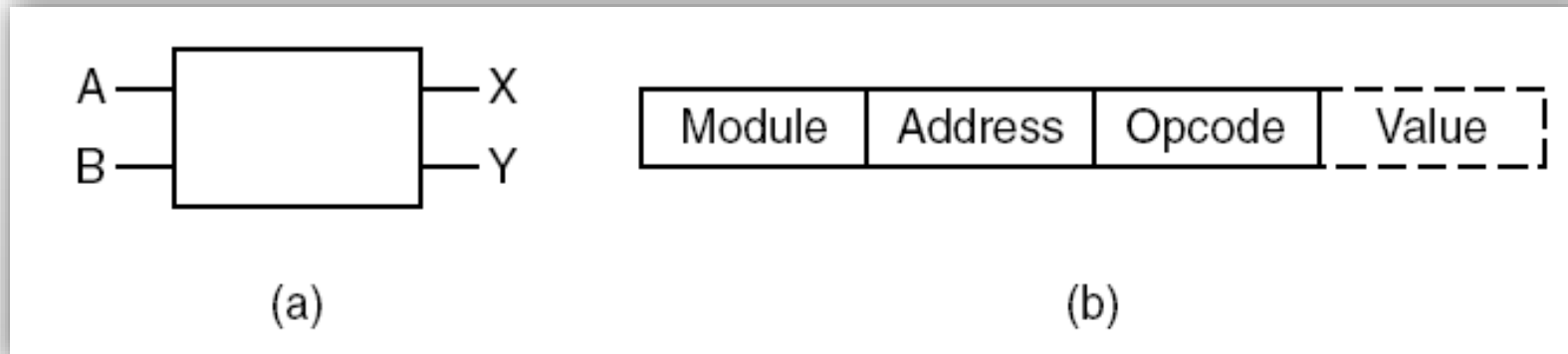
- (a) 8 × 8 çapraz çubuk anahtarı.
(*crosspoint switch*)
- (b) anahtar açık.
- (c) anahtar kapalı.





Çok Aşamalı Anahtarlama

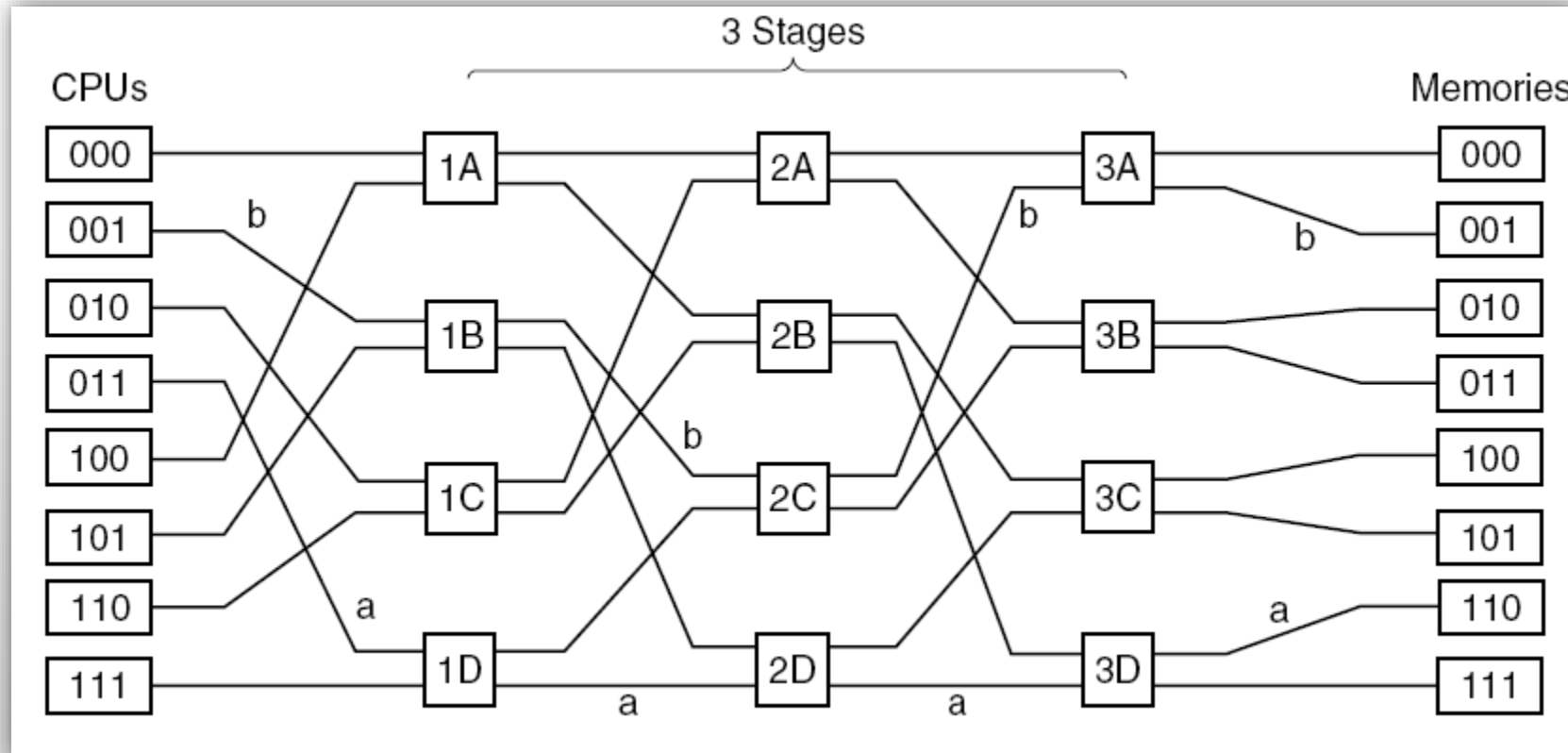
- (a) İki giriş (A ve B) ve iki çıkış (X ve Y) hattına sahip 2×2 anahtar.
- (b) Mesaj formatı.





Omega Anahtarlama Ağı

- 3 aşamalı anahtarlama ağı.





Omega Anahtarlama Ađı

- İşlemciler ve bellek arasında bağlantı sağlayan, bilgisayar ađ mimarisi.
- Anahtar düğümler ve düğümler arasındaki bağlantılar ađa benzer.
- İşlemciler arasında eşzamanlı (*senkron*) iletişime izin verir.
- Tasarımı karmaşık ve uygulaması zor.
- Çok sayıda anahtar düğümü nedeniyle güç tüketimi yüksek.



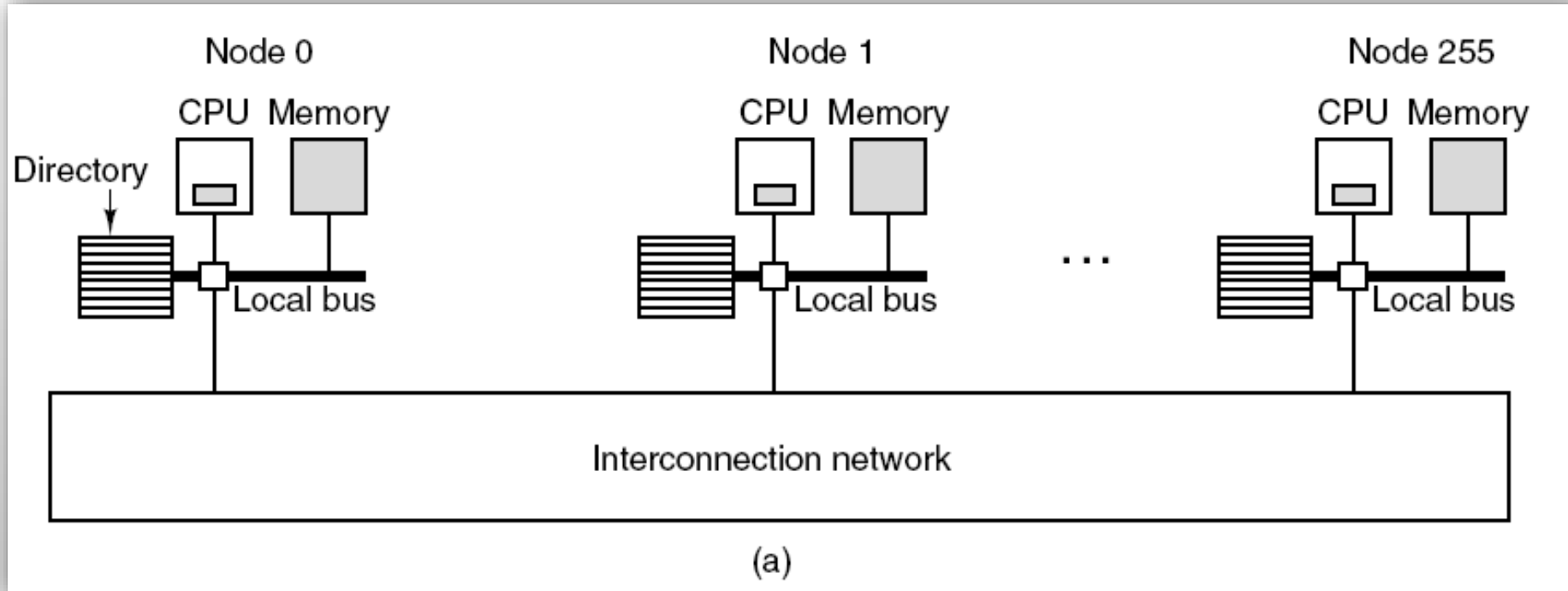
NUMA Çoklu İşlemcileri

- Non-unified memory access:
 - İşlemciler tarafından görülebilen tek bir adres alanı var.
 - Uzak belleğe erişim *LOAD* ve *STORE* komutları ile yapılır.
 - Uzak belleğe erişim, yerel belleğe erişimden daha yavaş.



NUMA Çoklu İşlemcileri

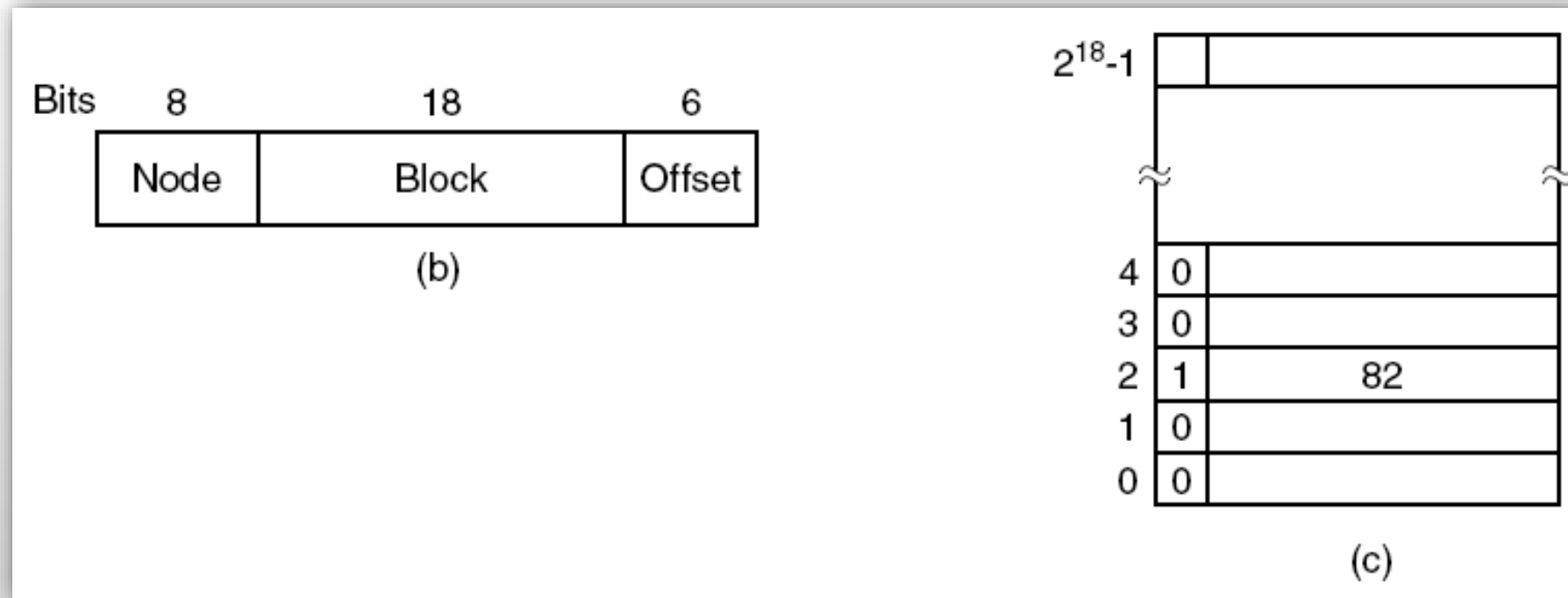
- (a) 256 düğümlü dizin tabanlı çok işlemci.





NUMA Çoklu İşlemcileri

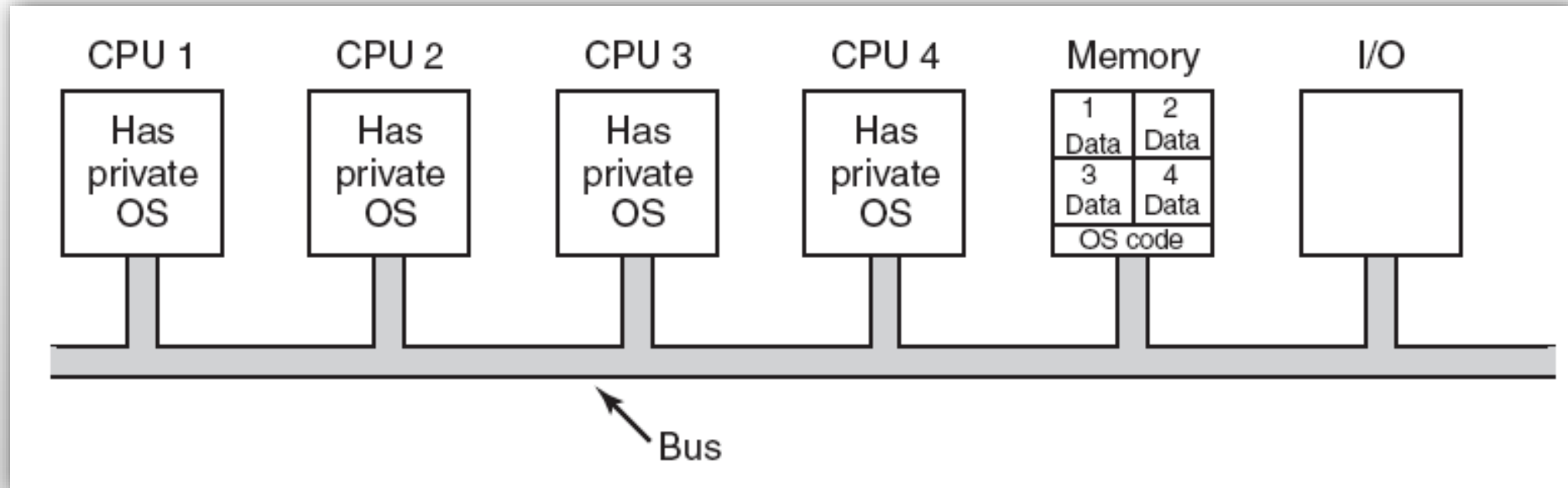
- (b) 32 bitlik bellek adresinin alanları. (c) 36. düğümdeki dizin.





Her CPU'nun Kendi İşletim Sistemi Vardır

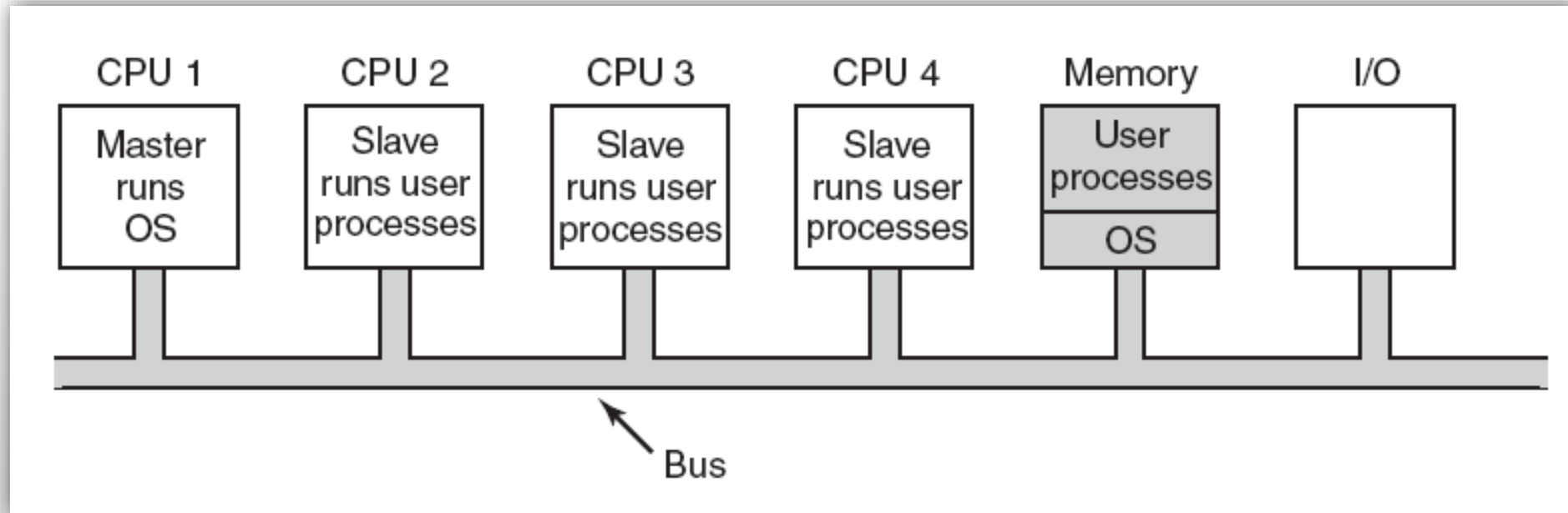
- Her işlemcinin kendine özel ayrılmış belleği vardır.
- İşletim sistemi kodu paylaşılır.





Usta Köle (Master Slave) Çoklu İşlemcili Model

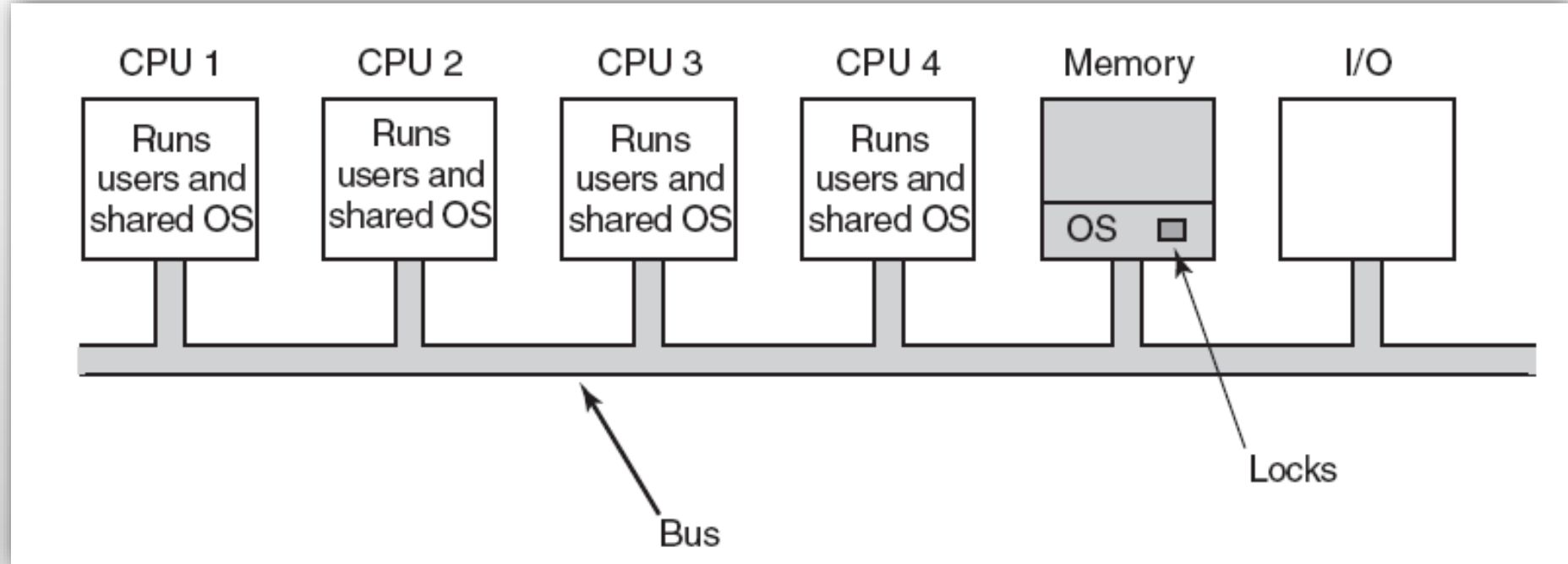
- Usta işlemci, işletim sistemini koşar,
- Köle işlemciler ortak belleği paylaşarak süreçleri yürütürler.





Simetrik Çoklu İşlemciler

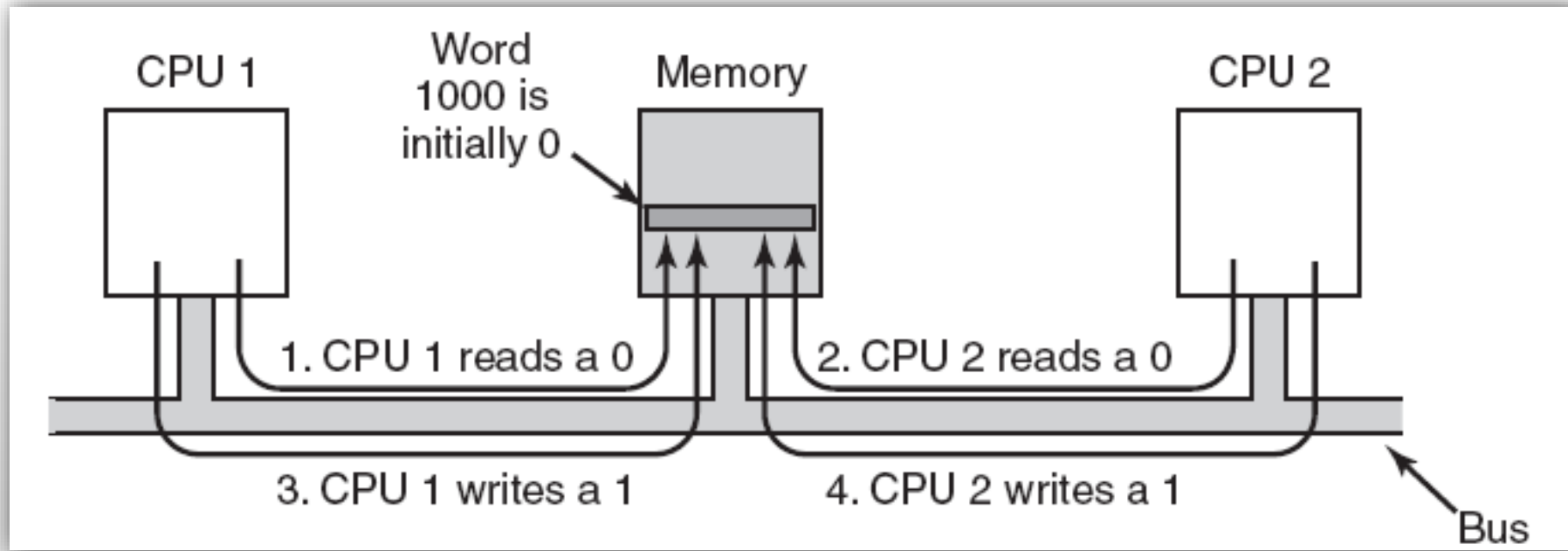
- SMP çok işlemcili model.





Çoklu İşlemcili Senkronizasyon

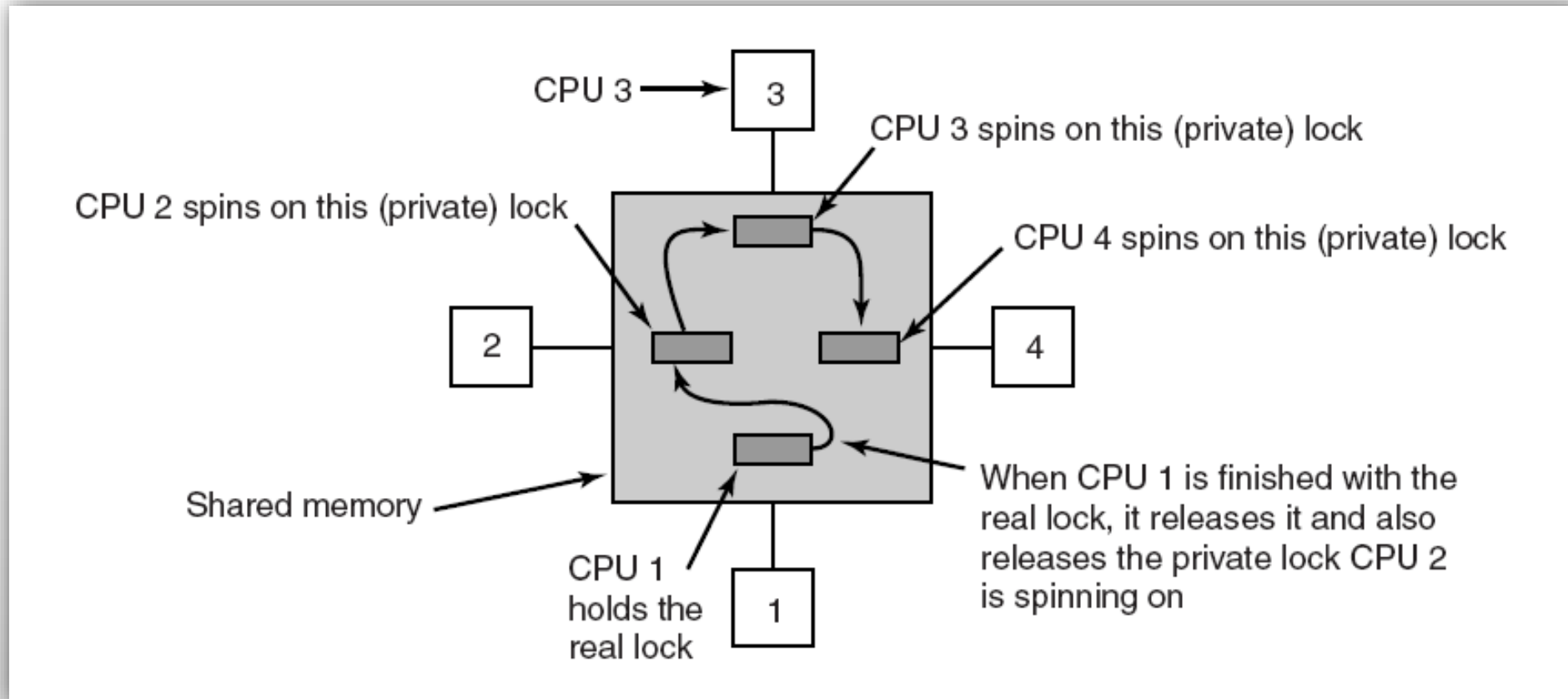
- Veri yolu kilitlenemezse *TSL* komutu başarısız olabilir.





Çoklu İşlemcili Senkronizasyon

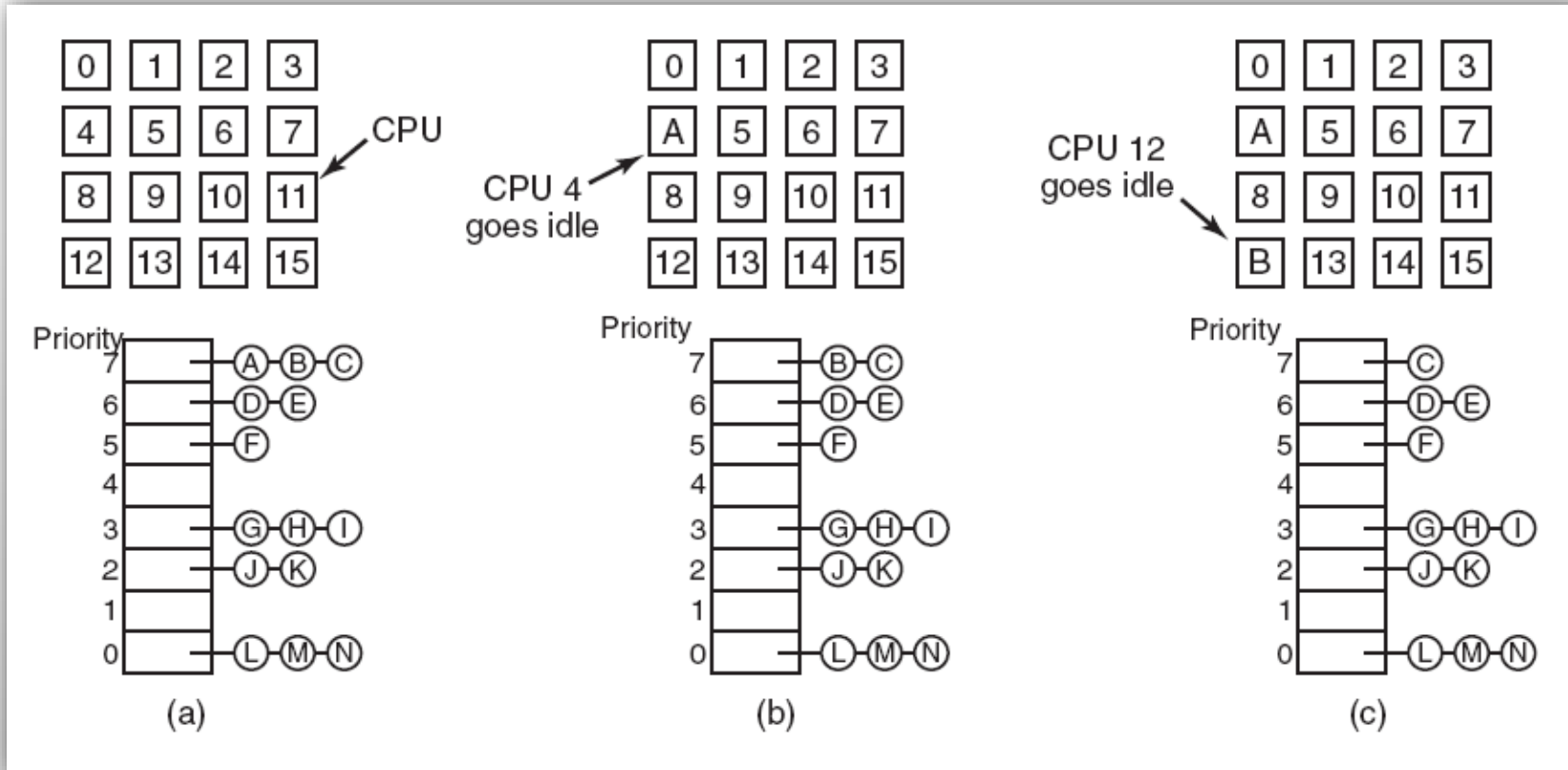
- Önbellekte veri tutarsızlığını önlemek için çoklu kilit kullanımı.





Zaman Paylaşımı

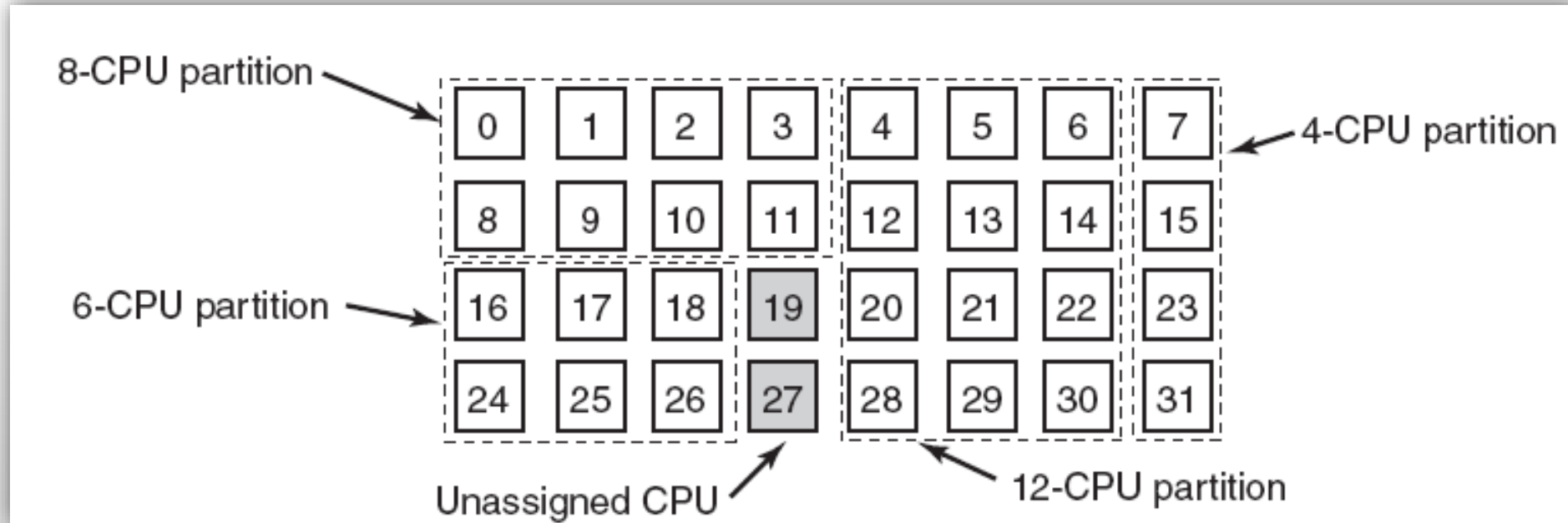
- Çoklu işlemcili çizelgeleme için veri yapısı kullanımı.





Alan Paylaşımı

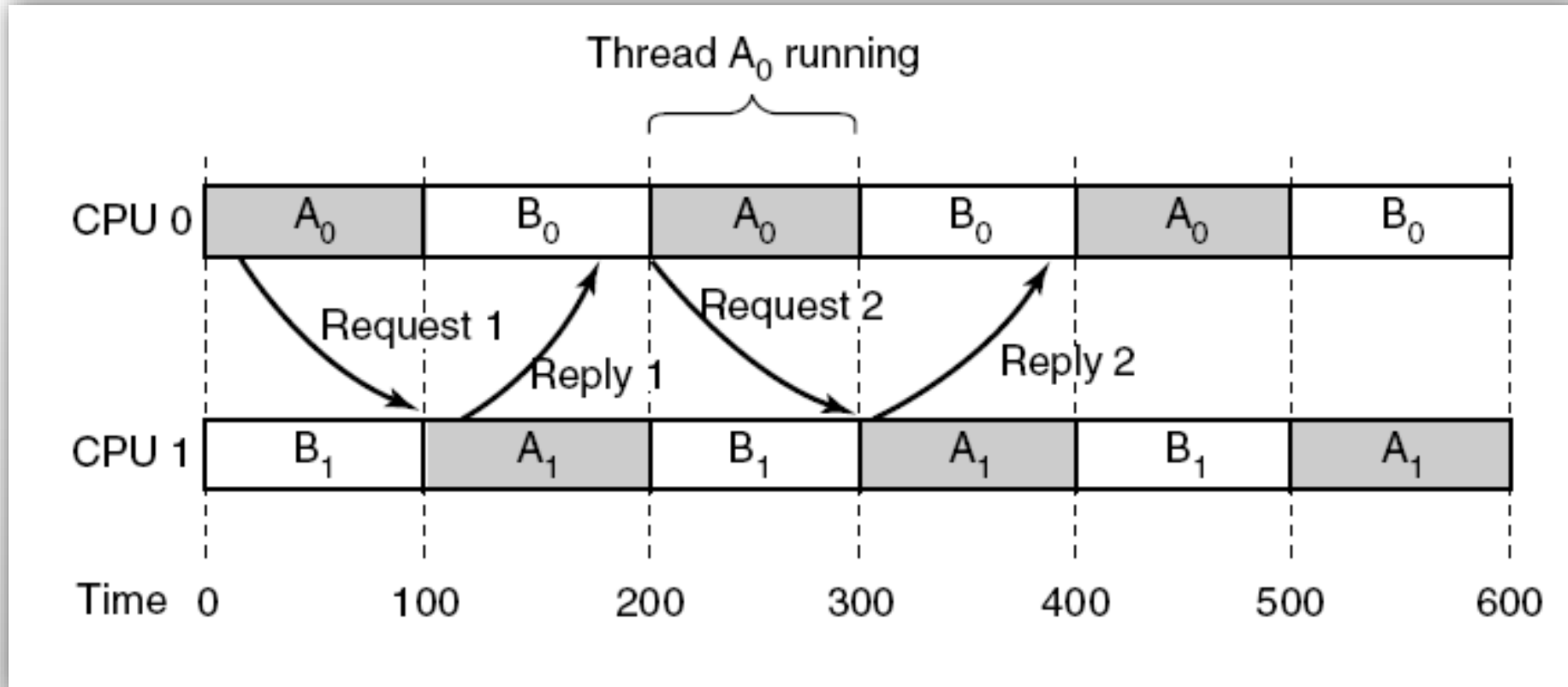
- 32 işlemciden oluşan bir küme. 2 işlemci boşa.
- 4, 6, 8 ve 12 işlemciden oluşan dört gruba ayrılmış.





Çete (Gang) Çizelgeleme

- Farklı işlemciye atanan A iş parçacıklarının arasındaki iletişim.





Çete (Gang) Çizelgeleme

- İlgili iş parçacığı grupları, çete olarak toplu çizelgelenir.
- Her bir iş parçacığı,
 - farklı işlemcilerde aynı anda çalışır.
 - zaman dilimlerini birlikte başlatırlar.



Çete (Gang) Çizelgeleme

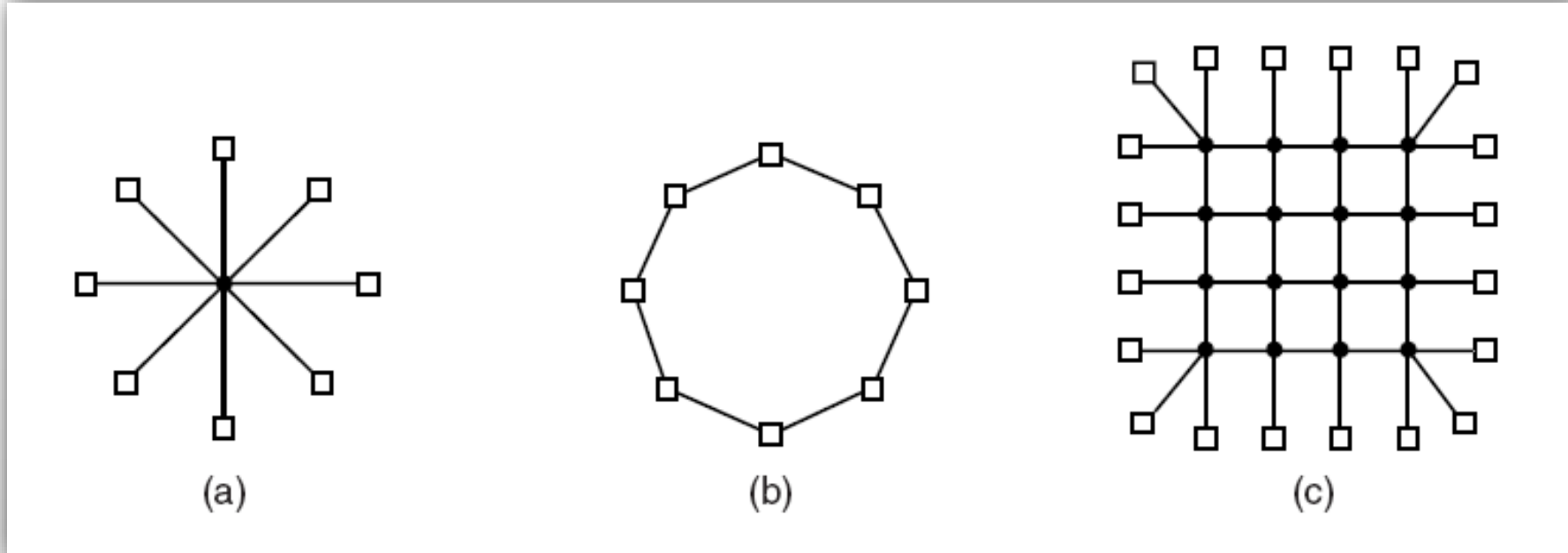
■ .

		CPU					
		0	1	2	3	4	5
Time slot	0	A ₀	A ₁	A ₂	A ₃	A ₄	A ₅
	1	B ₀	B ₁	B ₂	C ₀	C ₁	C ₂
	2	D ₀	D ₁	D ₂	D ₃	D ₄	E ₀
	3	E ₁	E ₂	E ₃	E ₄	E ₅	E ₆
	4	A ₀	A ₁	A ₂	A ₃	A ₄	A ₅
	5	B ₀	B ₁	B ₂	C ₀	C ₁	C ₂
	6	D ₀	D ₁	D ₂	D ₃	D ₄	E ₀
	7	E ₁	E ₂	E ₃	E ₄	E ₅	E ₆



Ara bağlantı (Interconnection) Topolojileri

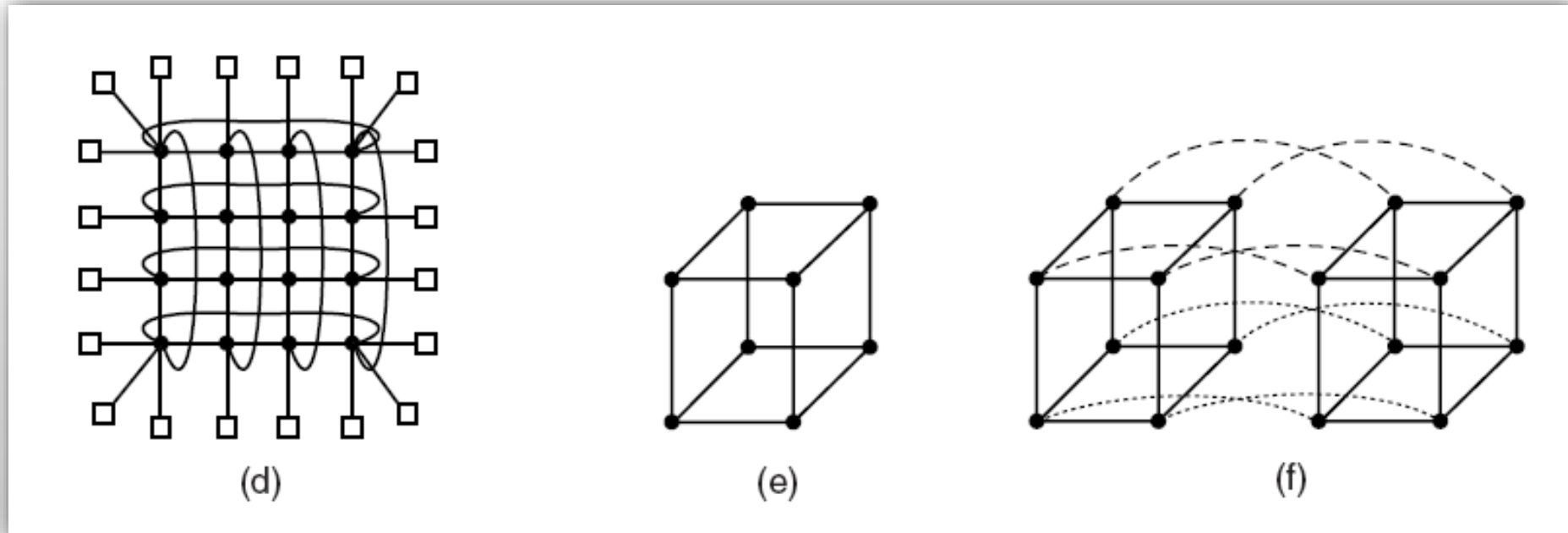
- (a) Tek anahtar (*switch*). (b) yüzük (*ring*). (c) ızgara (*grid*).





Ara bağlantı (Interconnection) Topolojileri

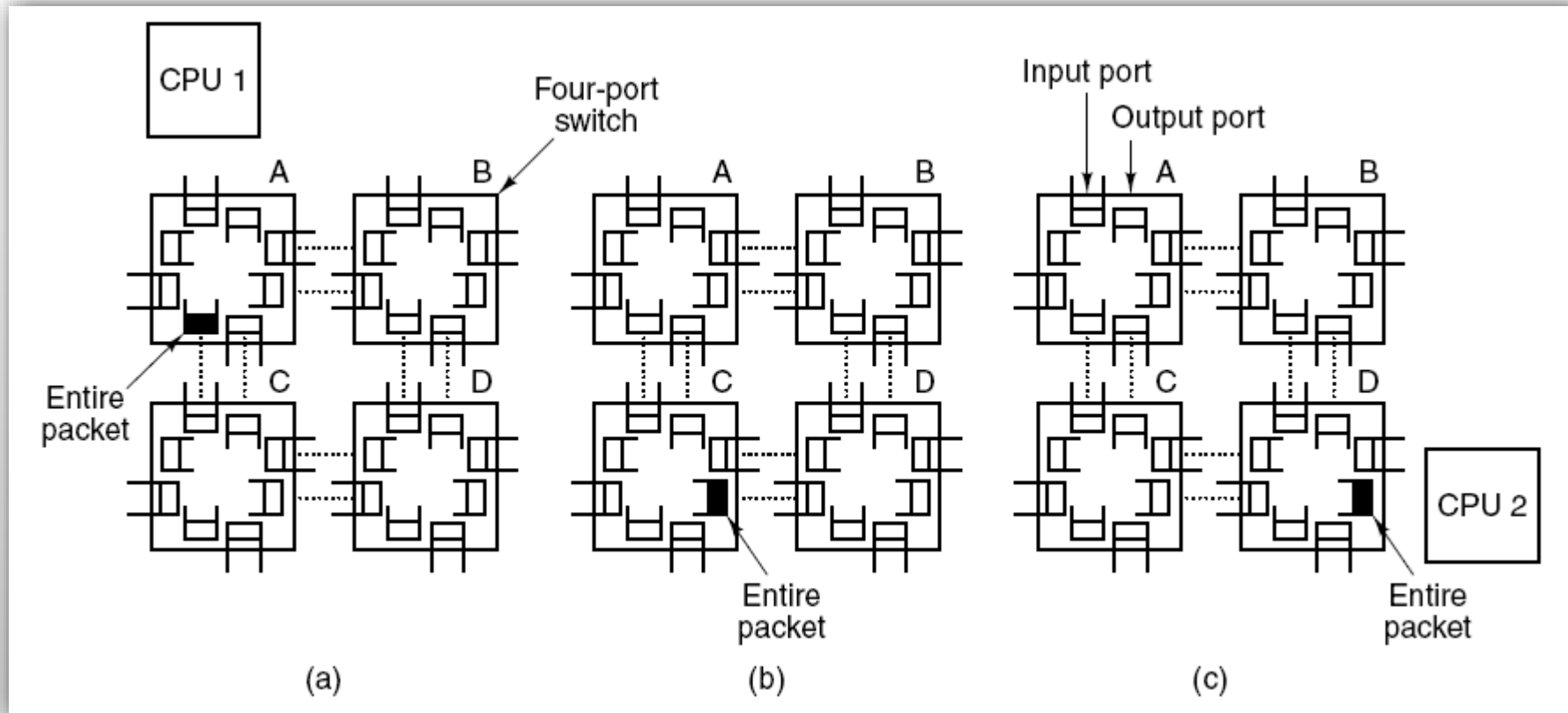
- (d) Çift simit (*torus*). (e) küp (*cube*). (f) 4D hiperküp (*hypercube*).





Ara bağlantı (Interconnection) Topolojileri

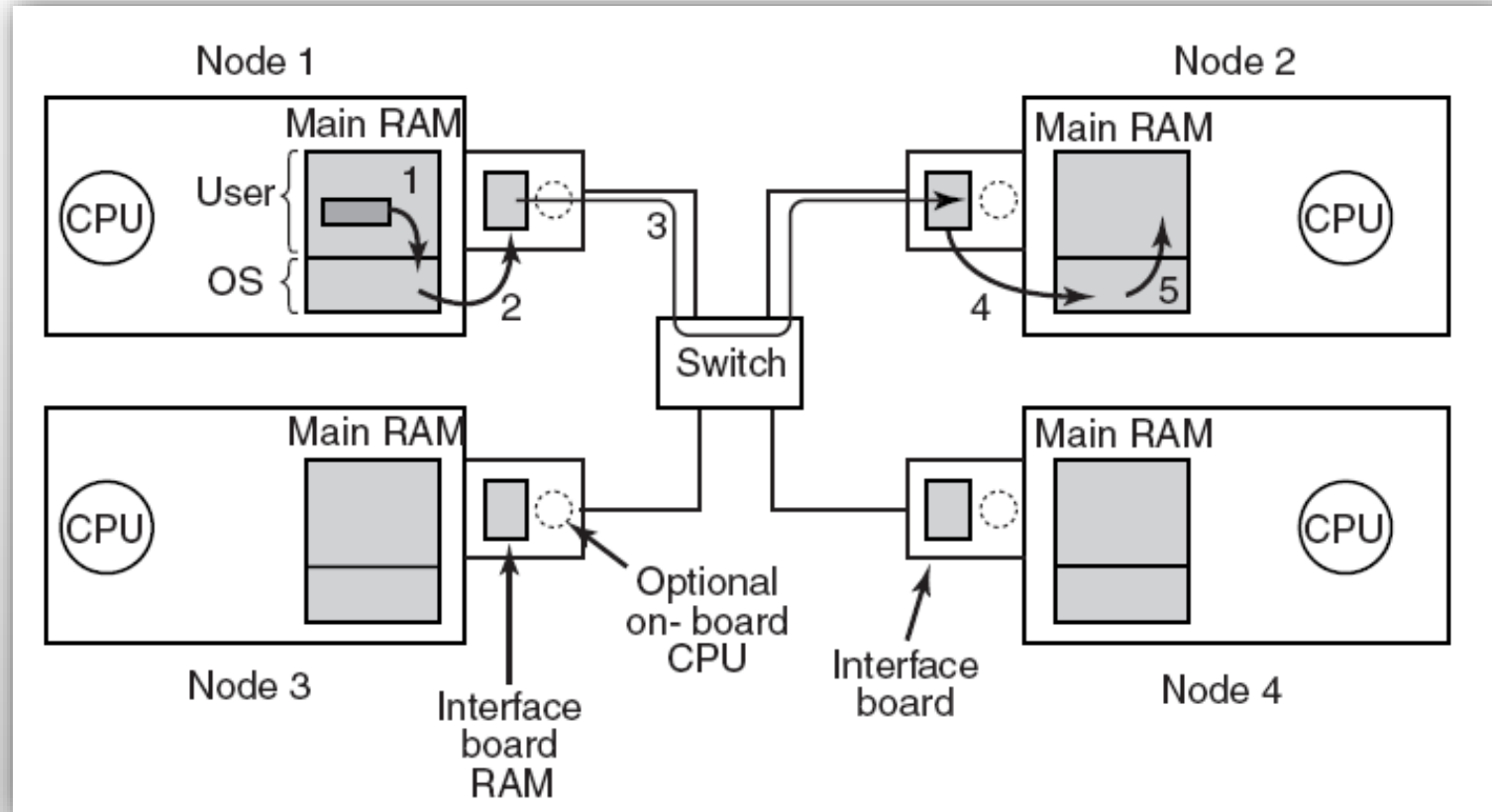
- Sakla ve ilet (*store and forward*) paket anahtarlama.





Ağ Arayüzleri

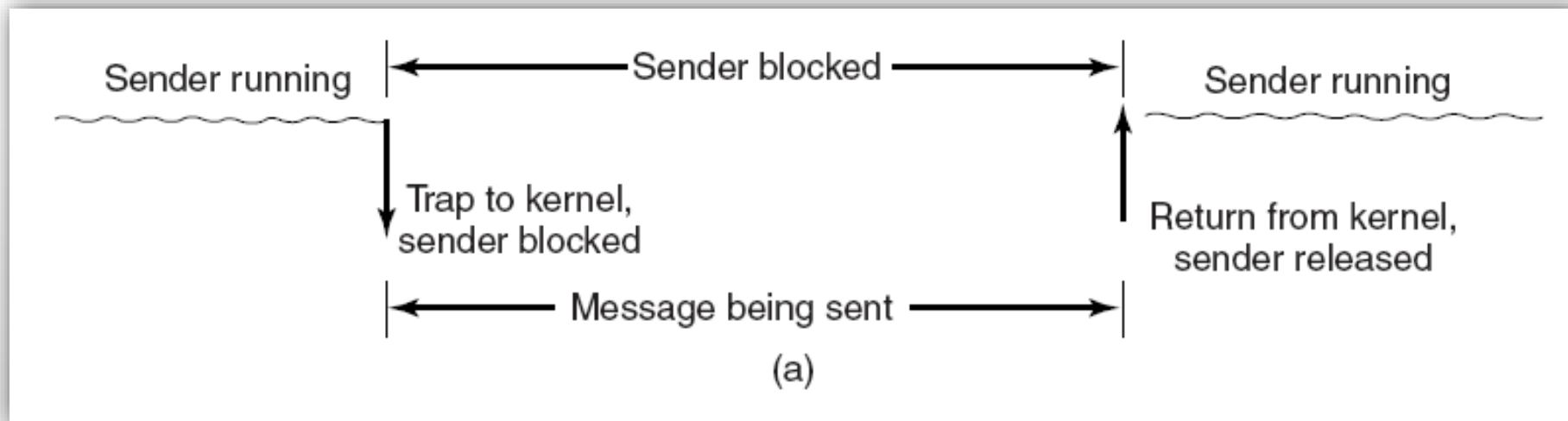
- Çoklu işlemcili sistemde ağ arayüz (*network interface*) kartlarının konumu.





Bloke Eden ve Etmeyen Çağrılar

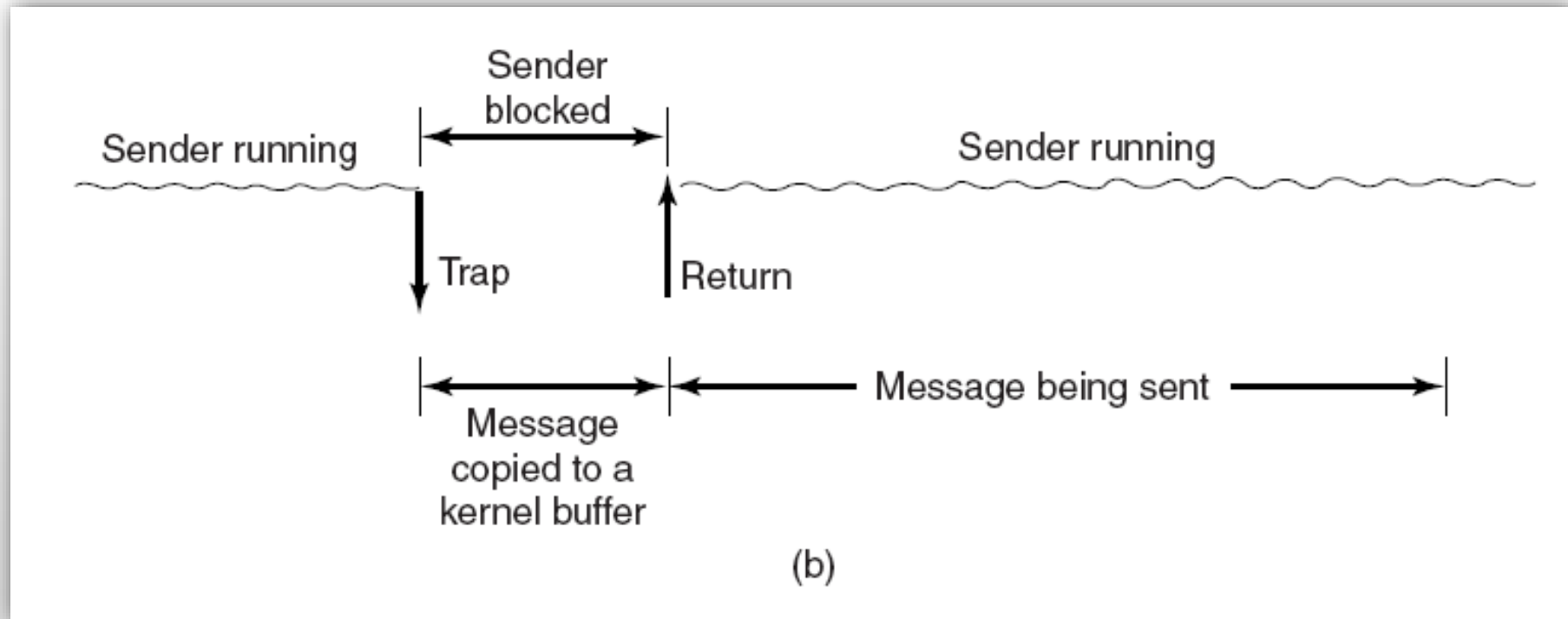
- (a) Bloke eden (*blocking*) gönderme çağrısı.





Bloke Eden ve Etmeyen Çağrılar

- (b) Bloke etmeyen (*non-blocking*) gönderme çağrısı.





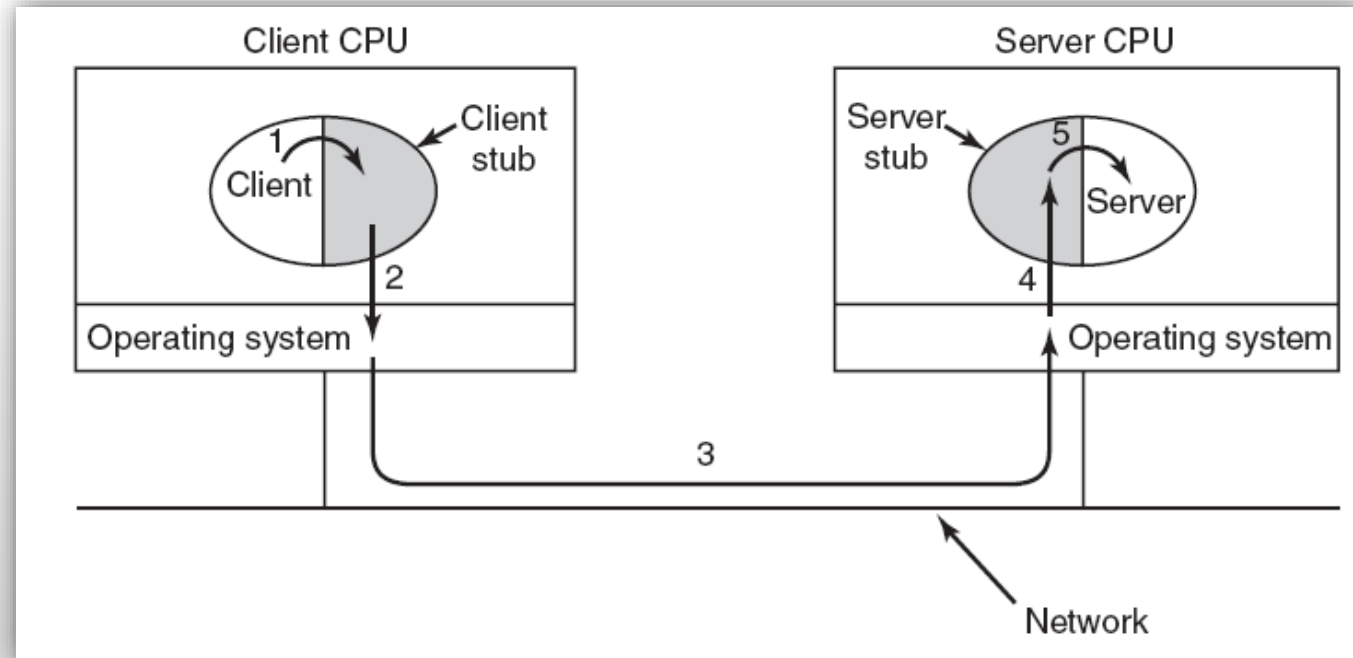
Bloke Eden ve Etmeyen Çağrılar

- Bloke eden (*blocking*) gönderim,
 - mesaj iletimi sırasında CPU boşta bekler.
- Bloke etmeyen (*non-blocking*) gönderim,
 - fazladan kopyalama işlemi için CPU zamanı boşa harcanır.
- Kesme ile bloke etmeyen gönderim,
 - programlamayı zorlaştırır.
- Yazarken kopyalama, (*copy on write*)
 - muhtemelen fazladan kopya gerekir.



Uzaktan Prosedür Çağrısı

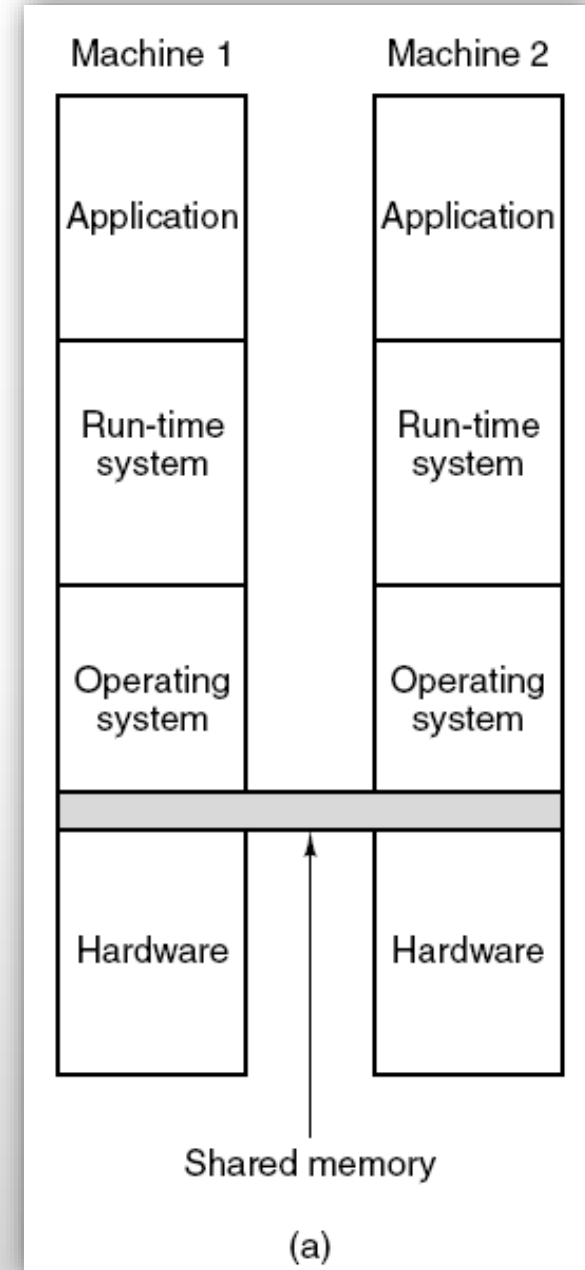
- *remote procedure call.*
- **Stub**: Sonra aslıyla değiştirilmek üzere geçici olarak yürütülebilir program.





Dağıtık Paylaşımlı Bellek

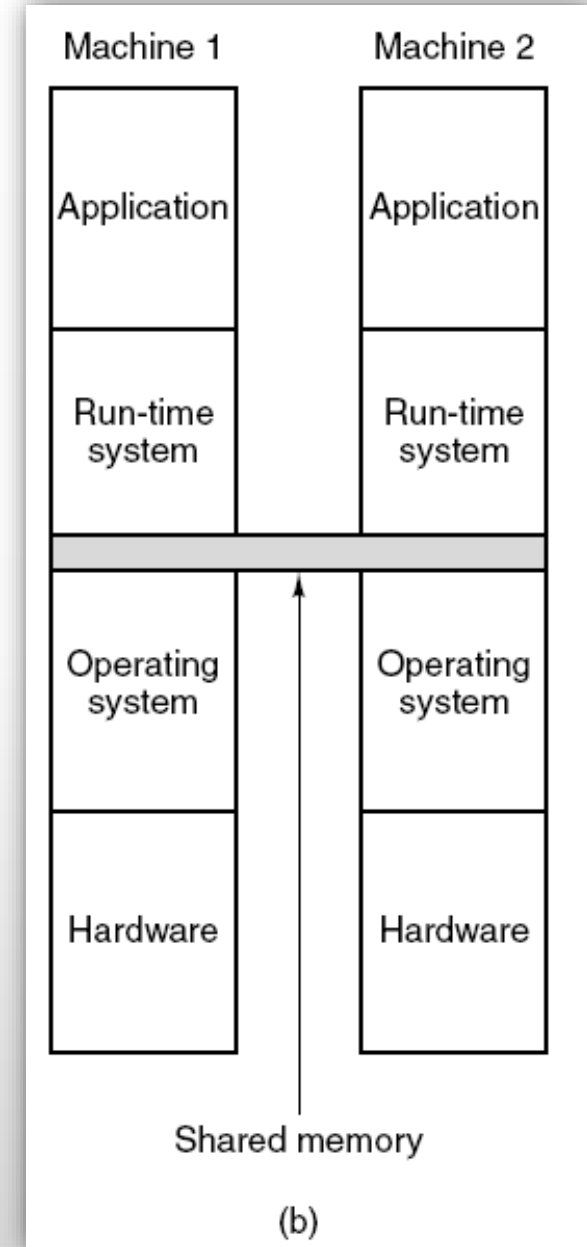
- *Distributed shared memory.*
- Paylaşımlı belleğin uygulanabileceği katmanlar. (a) Donanım.





Dağıtık Paylaşımlı Bellek

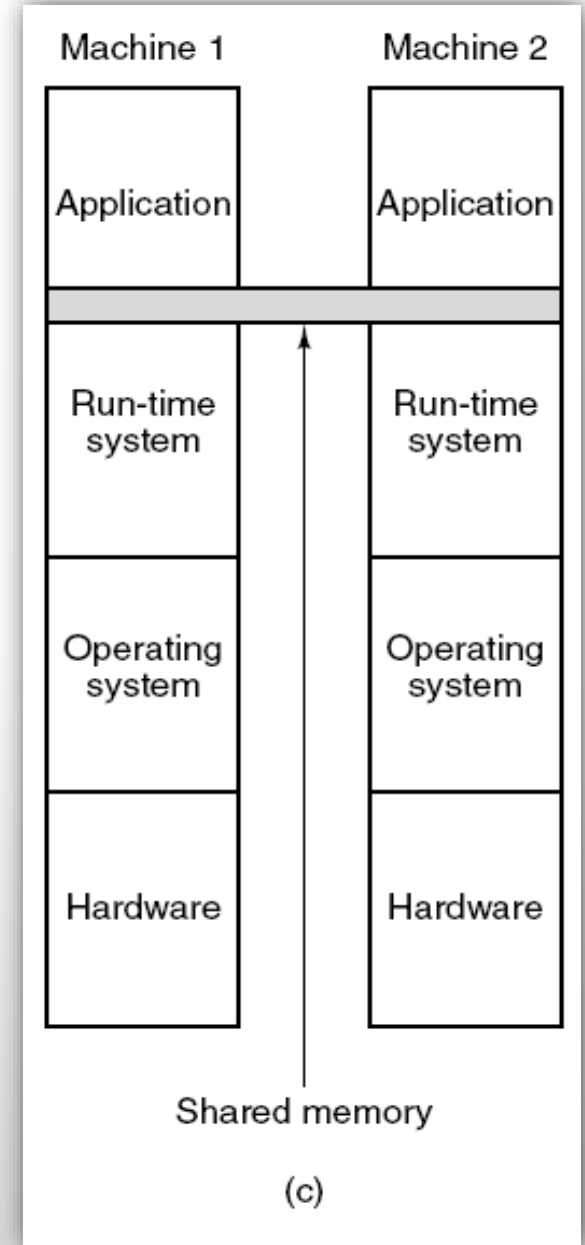
- *Distributed shared memory.*
- Paylaşımlı belleğin uygulanabileceği katmanlar. (b) İşletim sistemi.





Dağıtık Paylaşımlı Bellek

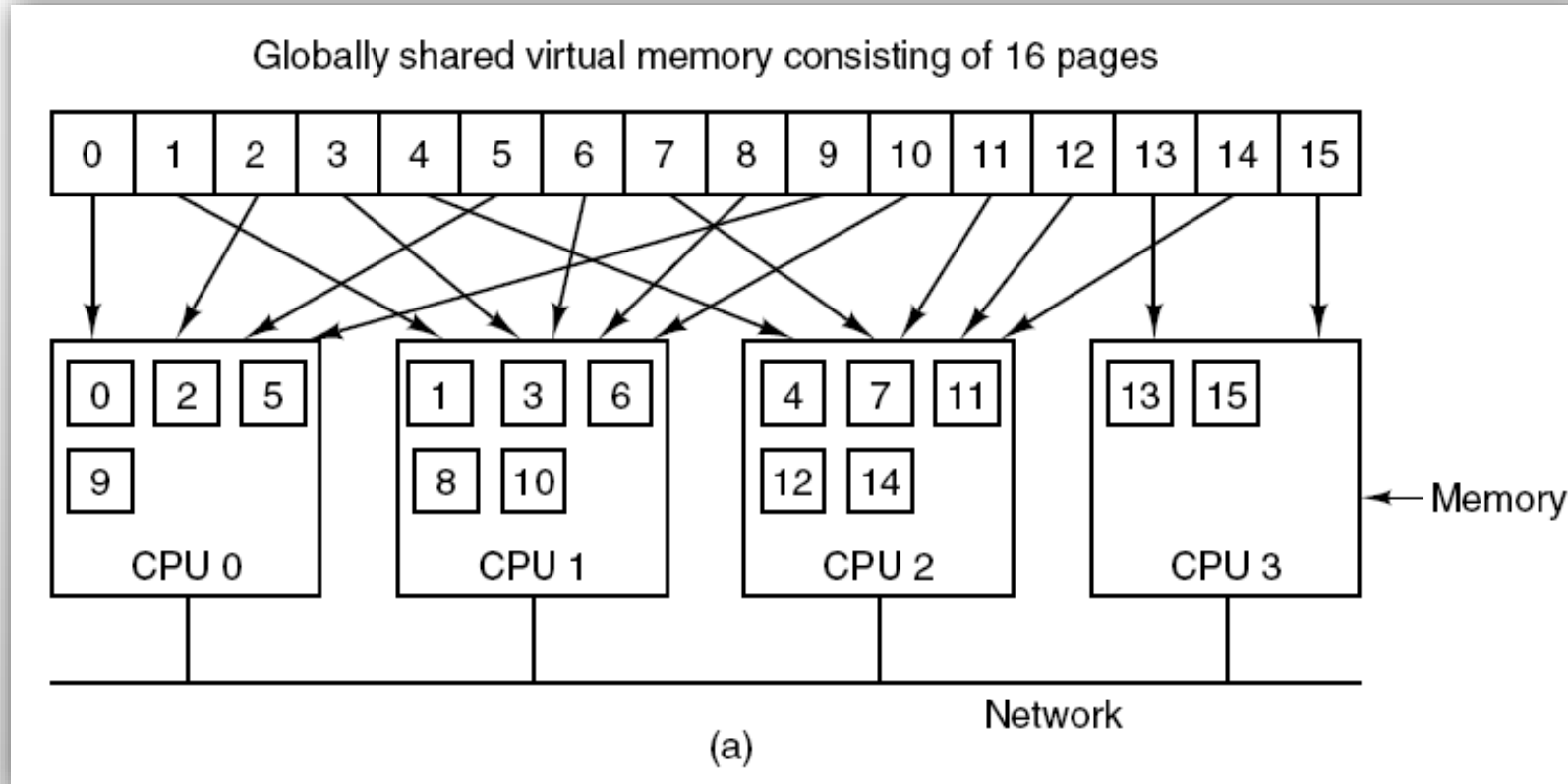
- *Distributed shared memory.*
- Paylaşımlı belleğin uygulanabileceği katmanlar. (c) Kullanıcı yazılımı.





Dağıtık Paylaşımlı Bellek

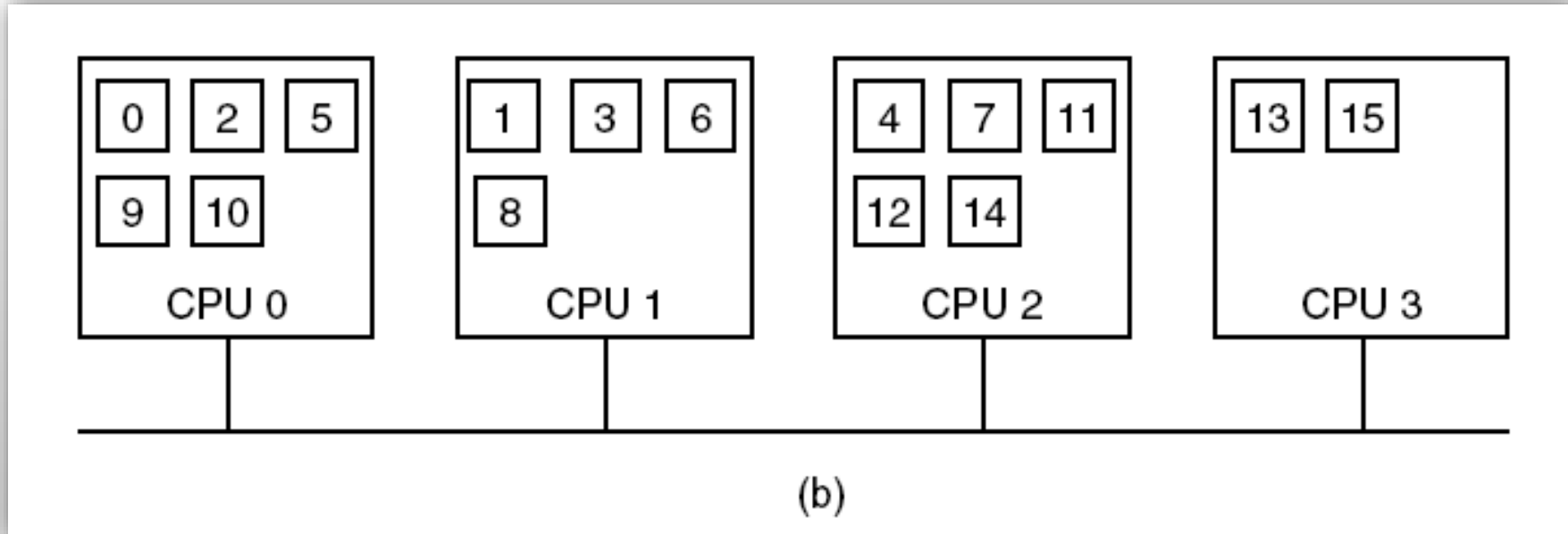
- (a) Dört işlemci arasında dağıtılan adres alanının sayfaları.





Dağıtık Paylaşımlı Bellek

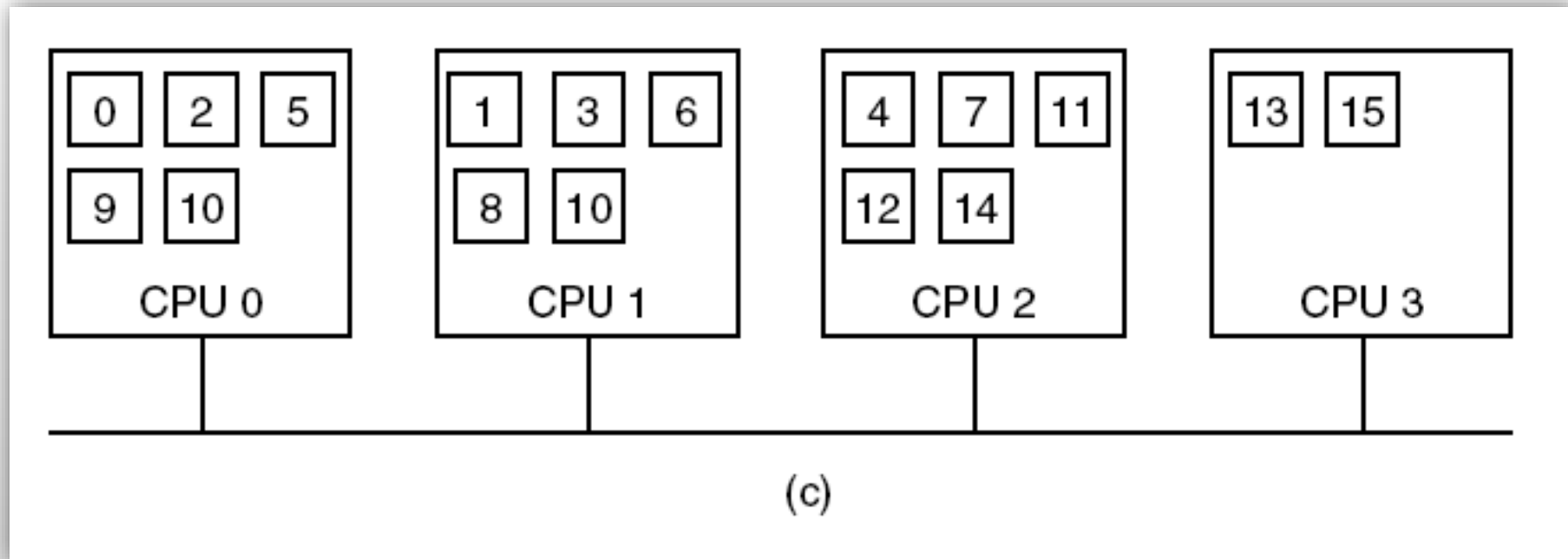
- (b) CPU 1 sayfa 10'a referans verir, sayfa taşındıktan sonraki durum.





Dağıtık Paylaşımlı Bellek

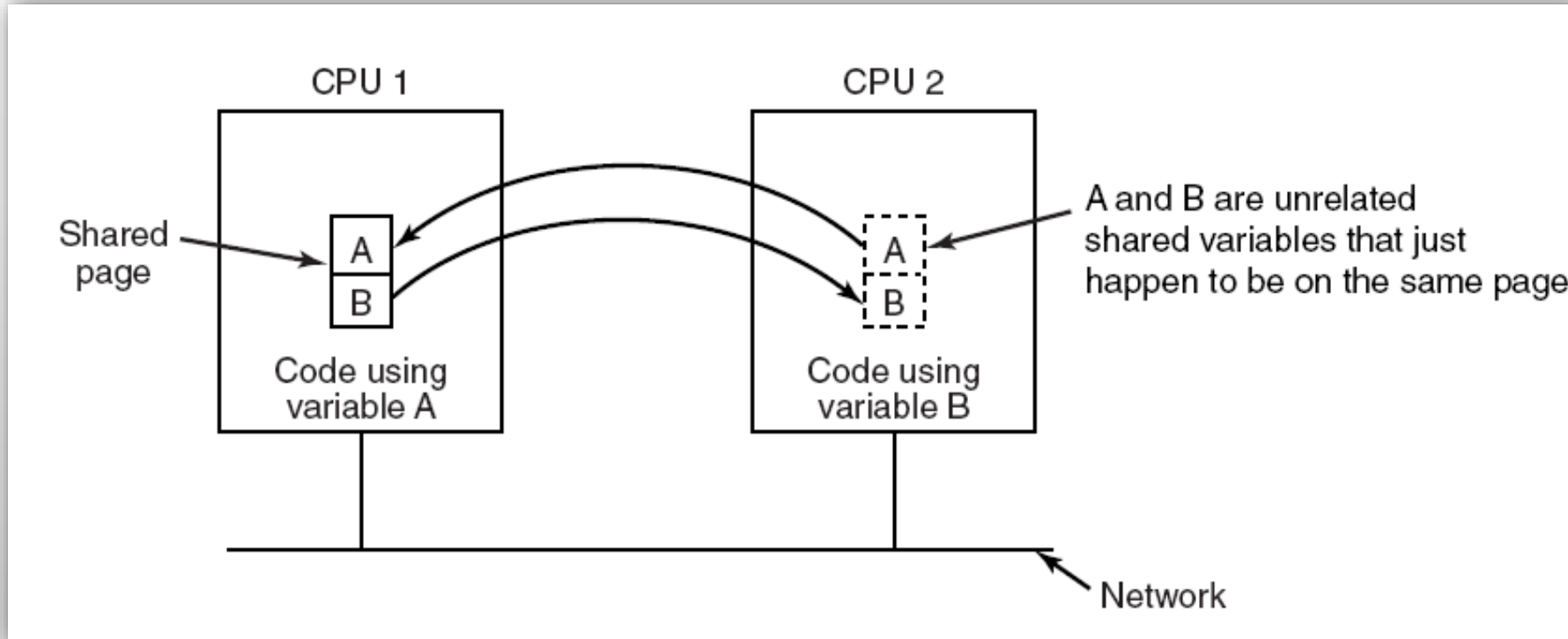
- (c) Sayfa 10 salt okunursa, kopya ile çoğaltılır. (*replication*)





Yanlış Paylaşım (False Sharing)

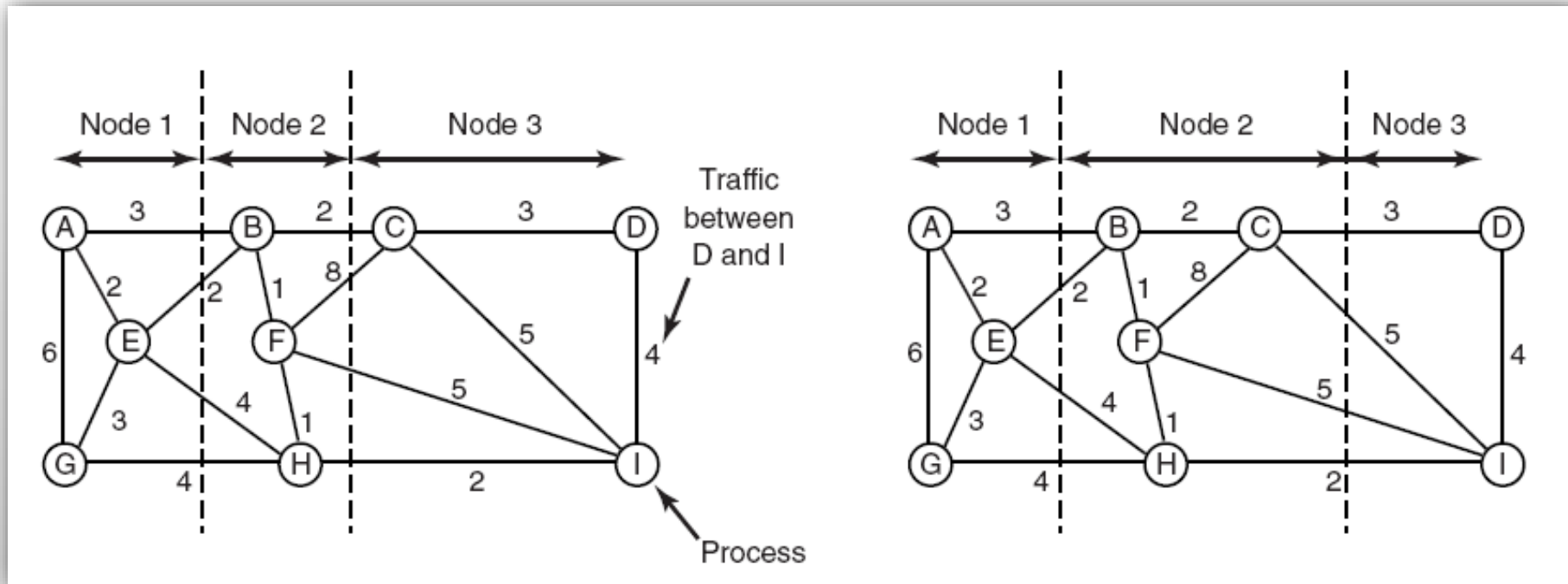
- İki birbirinden bağımsız değişken içeren bir sayfanın *yanlış paylaşımı*.





Bir Çizge-Teorik Deterministik Algoritma

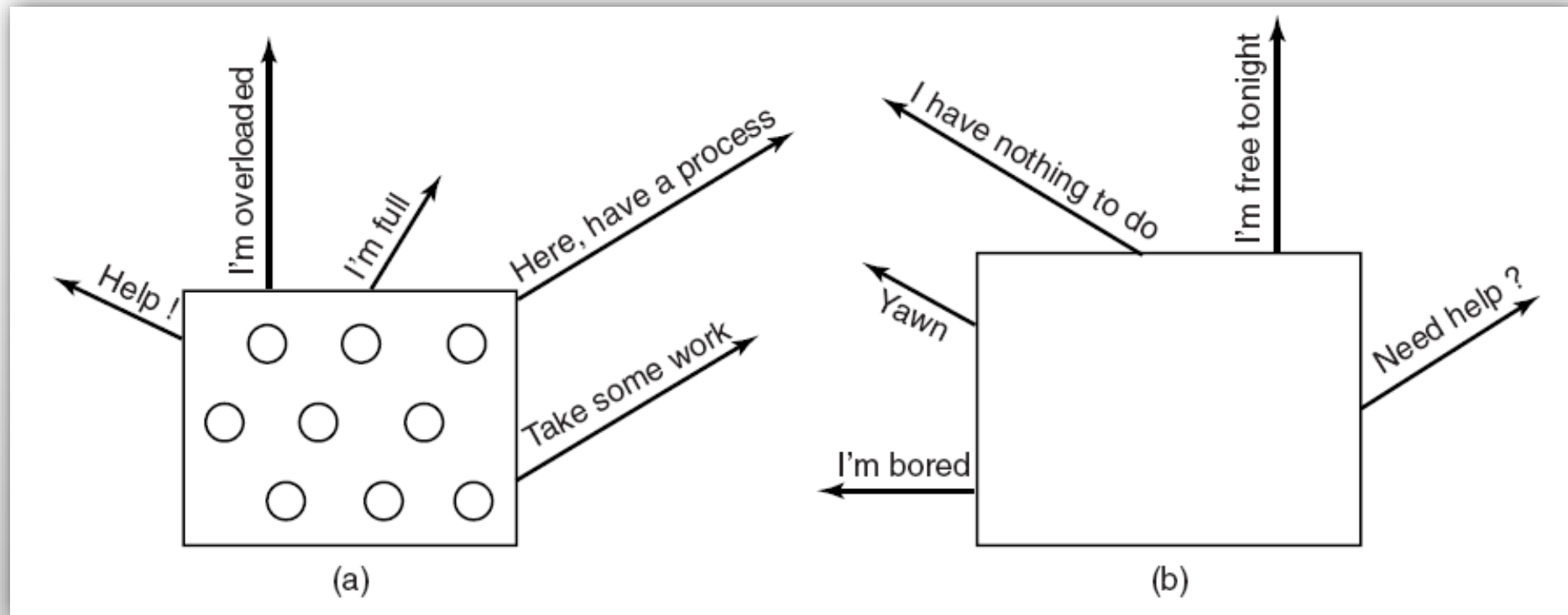
- 9 süreci 3 düğüme atama.





Dağıtık Sezgisel Algoritma

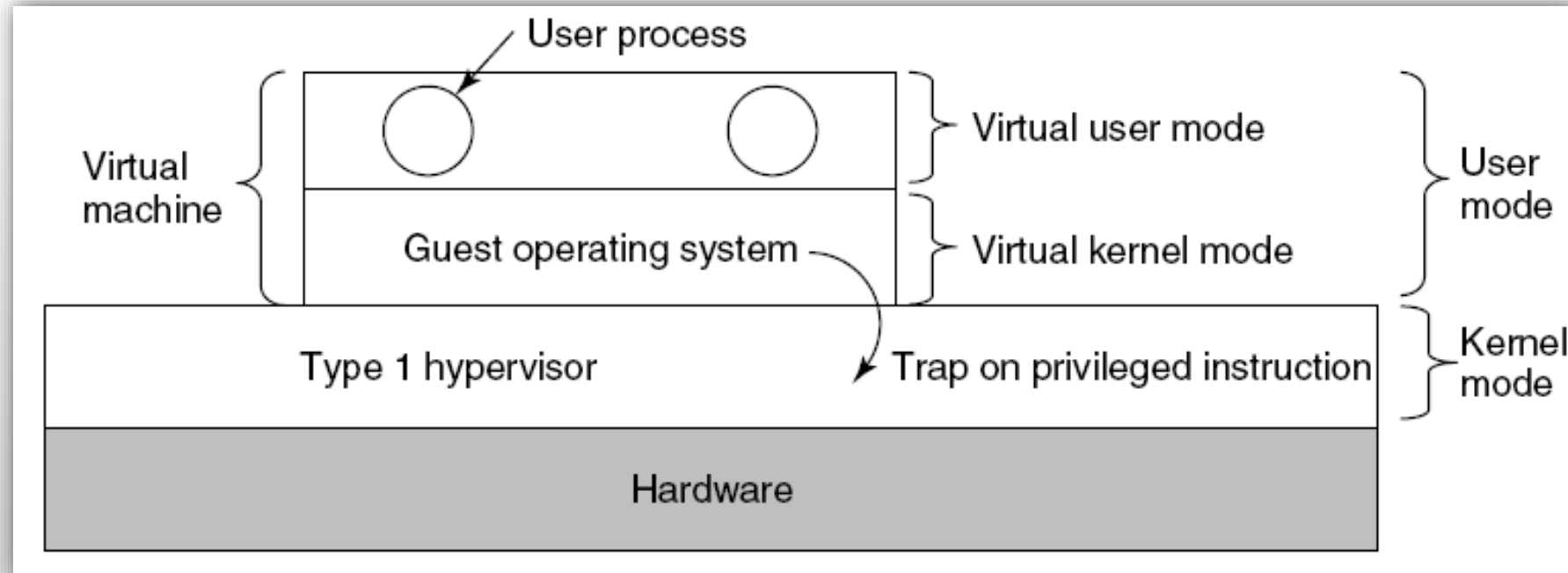
- (a) Aşırı yüklenmiş bir düğüm. (b) İş arayan boş bir düğüm.





Tip 1 Hipervizörler

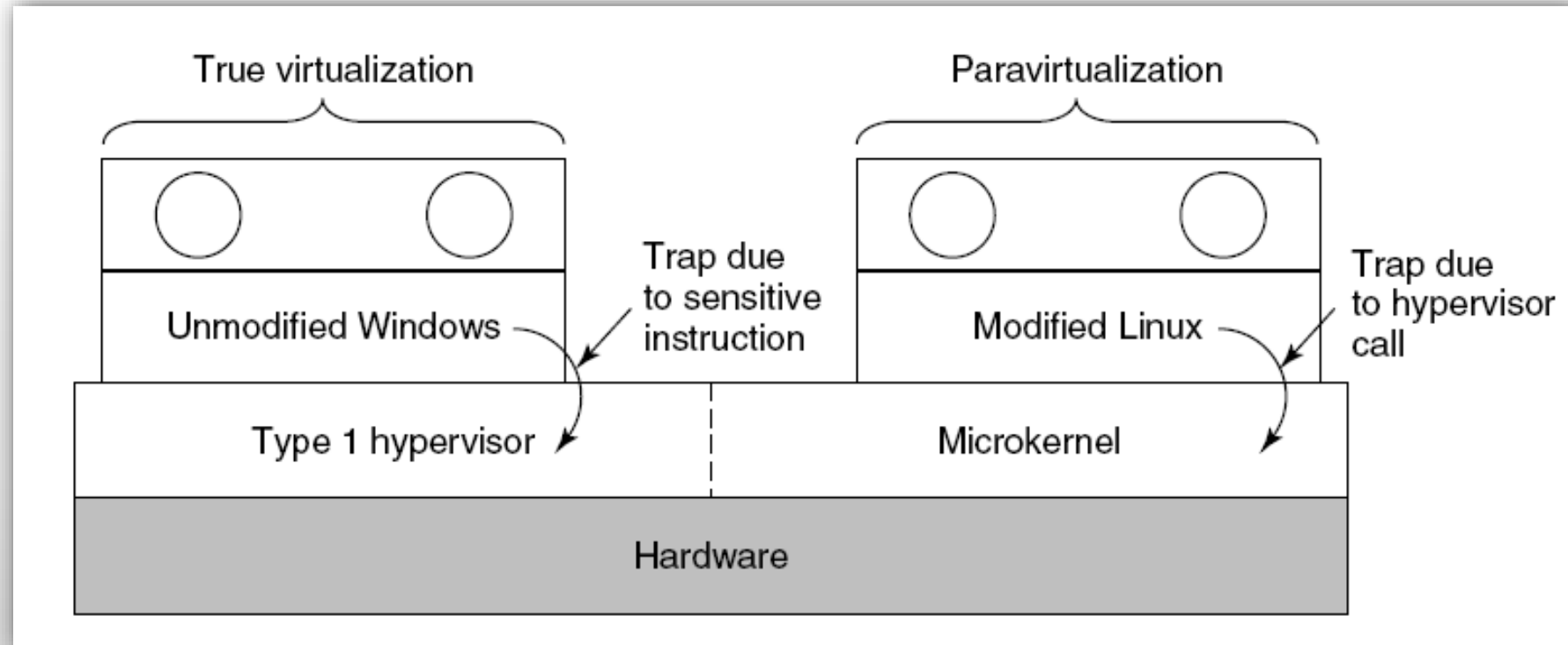
- Sanal makinedeki işletim sistemi, donanıma erişemez, komutu hipervizöre (hypervisor) iletir (*trap*).





Yarı Sanallaştırma (Paravirtualization)

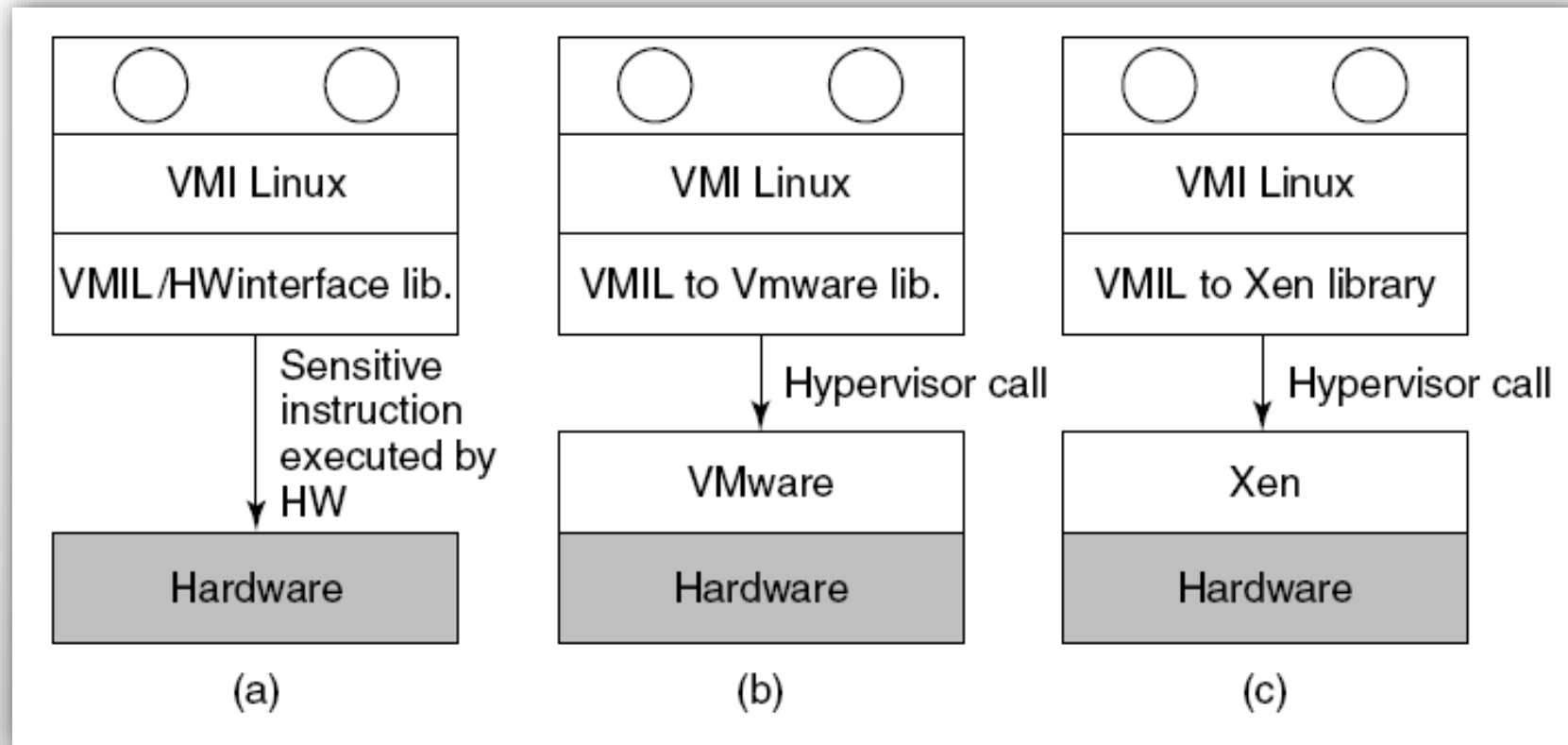
- Sanal makinedeki işletim sistemi modifiye edilir.





Yarı Sanallaştırma (Paravirtualization)

- (a) Çıplak (*bare*) donanım (b) VMware (c) Xen üzerinde çalışan VMI Linux.





Çoklu Bilgisayar Sistemi

- Bir ağ ile birbirine bağlantılı bağımsız bilgisayardan oluşur.
- Her bilgisayar kendi işletim sistemine sahip.
- Bilgisayarlar arasında iletişim sınırlı.
- Büyük ölçekli simülasyonlar ve paralel bilgi işlem görevleri için uygun.



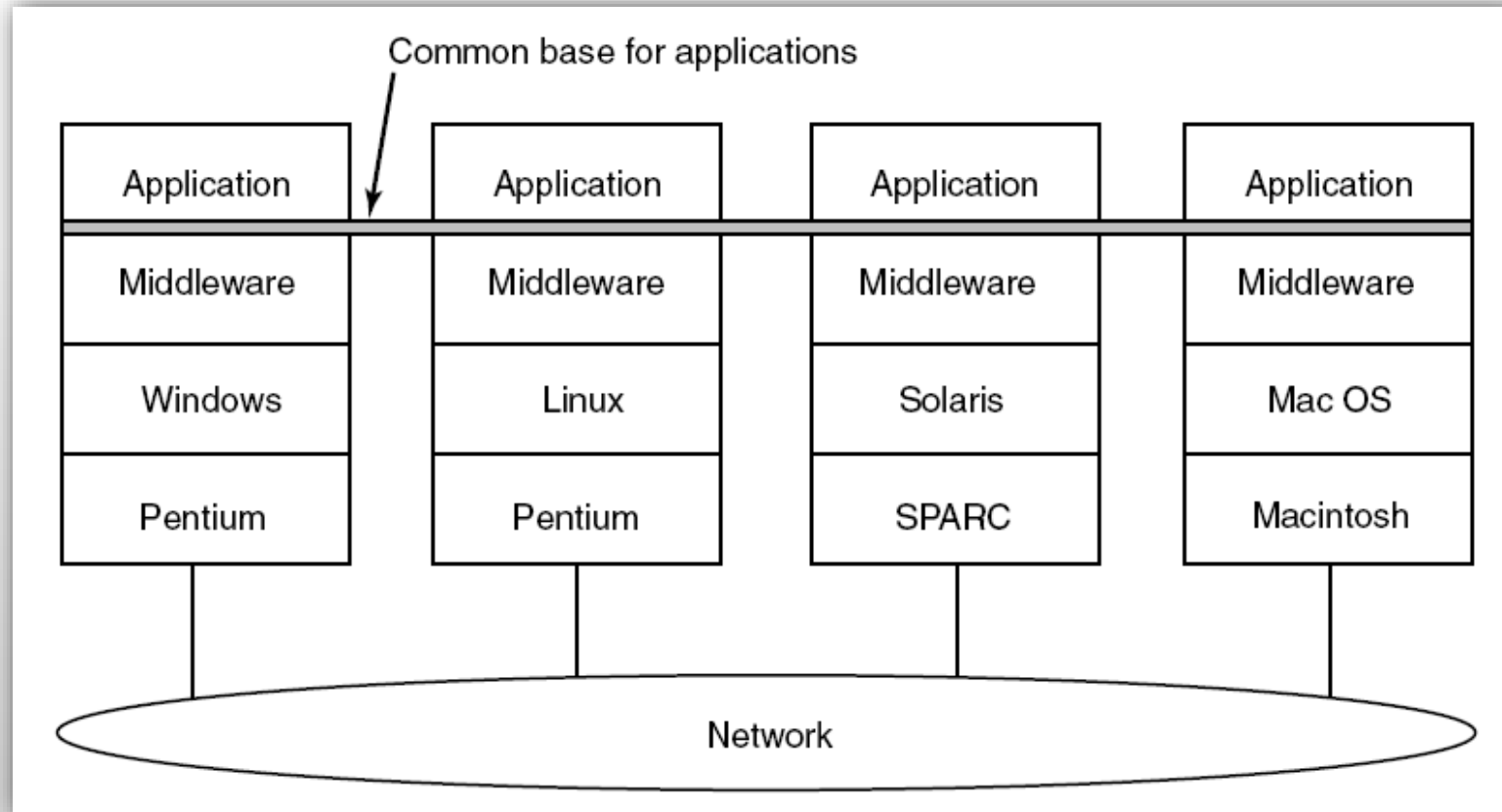
Dağıtık Sistem

- Bir ağ aracılığıyla iletişim kuran otonom bilgisayar topluluğu.
- Her bilgisayar,
 - kendi işletim sistemini çalıştırır.
 - kendi belleğine ve kaynaklarına sahiptir.
- Kaynak tahsisi ve görev dağıtımı,
 - dağıtık işletim sistemi tarafından yönetilir.
- Esneklik ve ölçeklenebilirlik gerektiren uygulamalar için uygun.



Dağıtık Sistem

- Dağıtık bir sistemde ara yazılımın konumu.





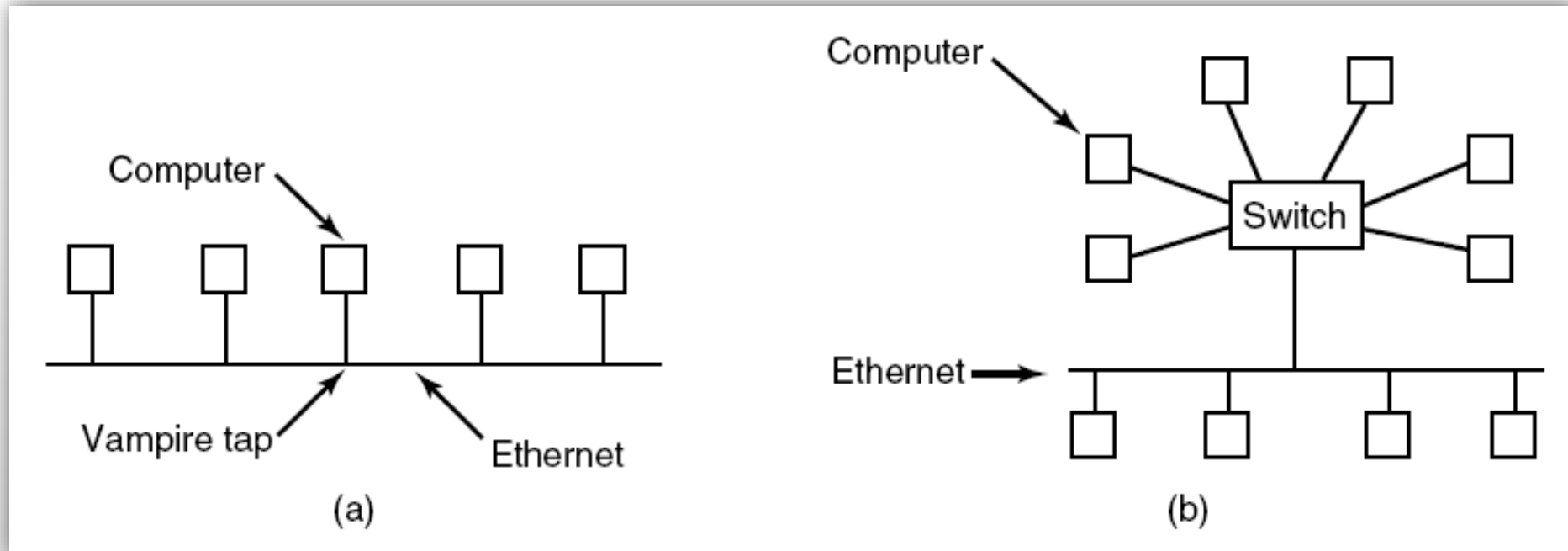
Karşılaştırma

	Çoklu İşlemci	Çoklu Bilgisayar	Dağıtık Sistem
Düğüm	CPU	CPU, RAM, ağ birimi	Tüm Bilgisayar
Çevre birimleri	Tümü paylaşılır	Disk paylaşılır	Tümü paylaşılır
Konum	Aynı raf	Aynı oda	Dünya çapında
İletişim	Paylaşılan RAM	Özel ara bağlantı	Geleneksel ağ
İşletim sistemleri	Bir, paylaşılan	Çoklu, aynı	Hepsi farklı
Dosya sistemleri	Bir, paylaşılan	Bir, paylaşılan	Kendine ait
Yönetim	Tek	Tek	Çoklu



Ethernet

- (a) Klasik. (b) Anahtarlmalı.





Ağ Protokolleri

- **İletim Kontrol Protokolü (TCP):** (*transmission control protocol*)
 - Hata algılama ve düzeltme yeteneği.
 - Güvenilir, bağlantı yönelimli (*connection oriented*) veri aktarımı sağlar.
- **Kullanıcı Datagram Protokolü (UDP):** (*user datagram protocol*)
 - Hata algılama veya düzeltme yok.
 - Güvenilir olmayan, bağlantısız (*connectionless*) veri aktarımı sağlar.
- **İnternet Protokolü (IP):** (*internet protocol*)
 - Ağlar arasında veri iletimi için yönlendirme ve adresleme işlevi sağlar.
- **Basit Ağ Yönetimi Protokolü (SNMP):** (*simple network management*)
 - Ağ cihazlarını yönetmek ve izlemek için kullanılır.

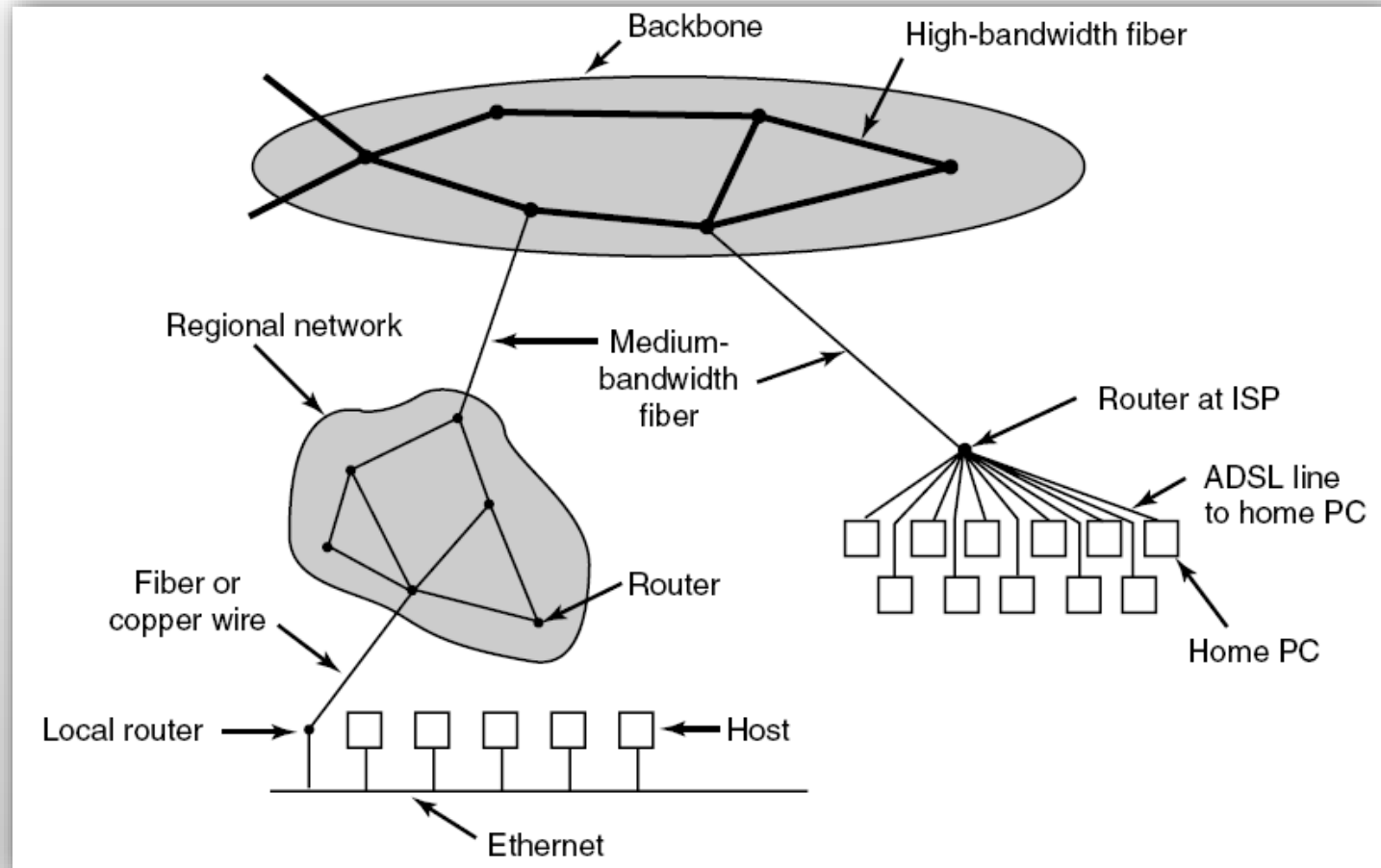


Ağ Protokolleri

- **Alan Adı Sistemi (DNS):** (*domain name system*)
 - Alan adını IP adresine çözümler. Hiyerarşik, dağıtık adlandırma sistemi.
- **Dosya Aktarım Protokolü (FTP):** (*file transfer protocol*)
 - İnternet üzerinden dosya aktarımı için kullanılır.
- **Köprü Metni Aktarım Protokolü (HTTP):** (*hypertext transfer protocol*)
 - Web (*World Wide Web*) sayfalarının görüntülenmesini sağlar.
- **Basit Posta Aktarım Protokolü (SMTP):** (*simple mail transfer protocol*)
 - E-posta mesajlarının aktarımı için kullanılır.



İnternet Omurgası (Backbone)





Güvenilir ve Olmayan Ağ Servisleri

- Güvenilir (*reliable*):
 - Verilerin teslimi garanti edilir.
 - Göndericiye Alındı (*acknowledge*) onayı gönderilir.
 - *TCP* gibi güvenilir aktarım protokolleri kullanılır.
- Güvenilir Olmayan (*unreliable*):
 - Verilerin teslimi garanti edilmez.
 - Alındı onayı gönderilmez.
 - *UDP* gibi güvenilir olmayan aktarım protokolleri kullanılır.



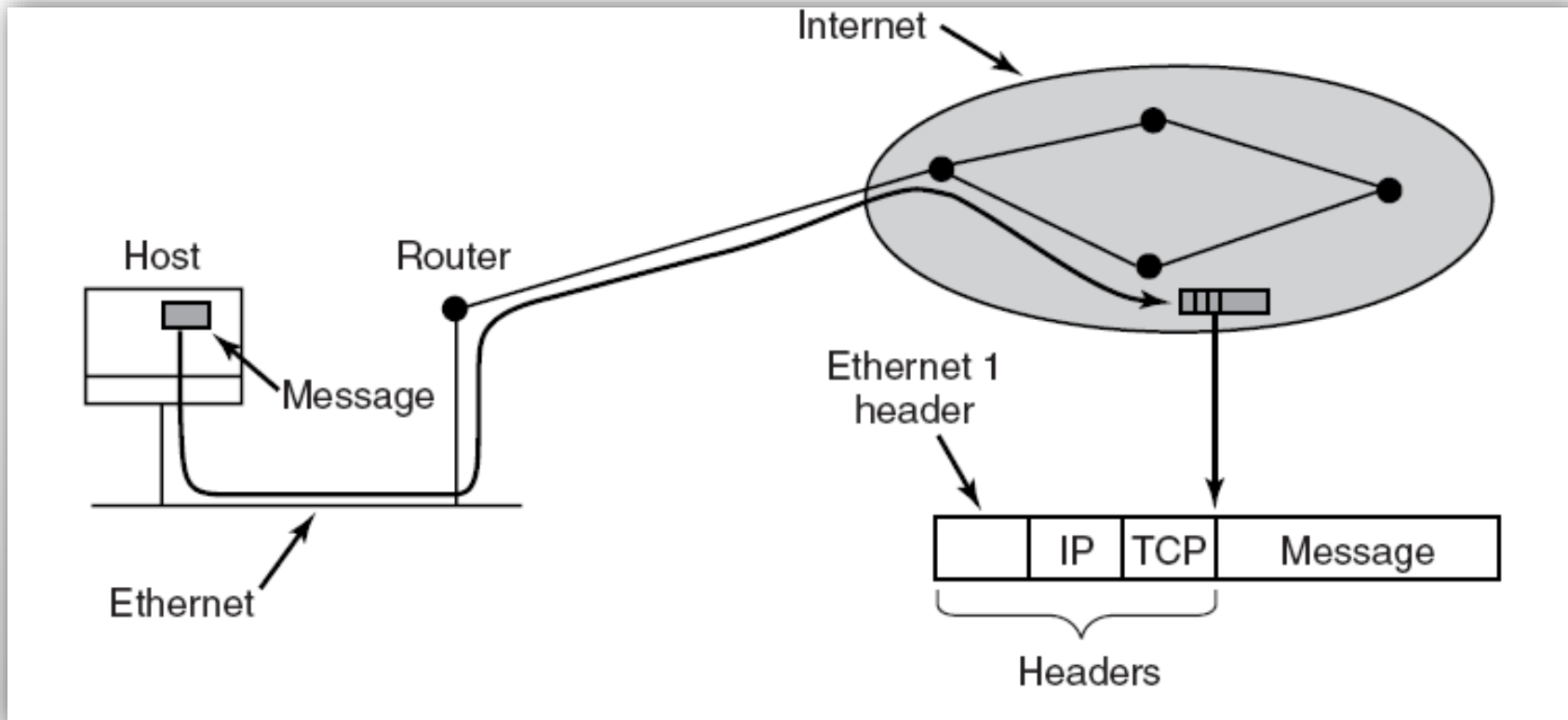
Bağlantı Odaklı ve Bağlantısız Servisler

- Bağlantı Odaklı (*connection oriented*):
 - İki düğüm arasında güvenilir bir bağlantı kurulur.
 - Akış kontrolü, hata düzeltme ve yeniden iletim mekanizmaları var.
 - Veri teslimatı, veri bütünlüğü garanti edilir.
 - Örnekler: *TCP, SCTP*
- Bağlantısız (*connectionless*):
 - Kalıcı bir bağlantı kurulmaz.
 - Veri teslimatı garanti edilmez.
 - Bağlantı kurma ve sonlandırma maliyeti yok.
 - Yüksek verim, düşük gecikme süresi sağlar.
 - Örnekler: *UDP, IP*



Ağ Protokolleri

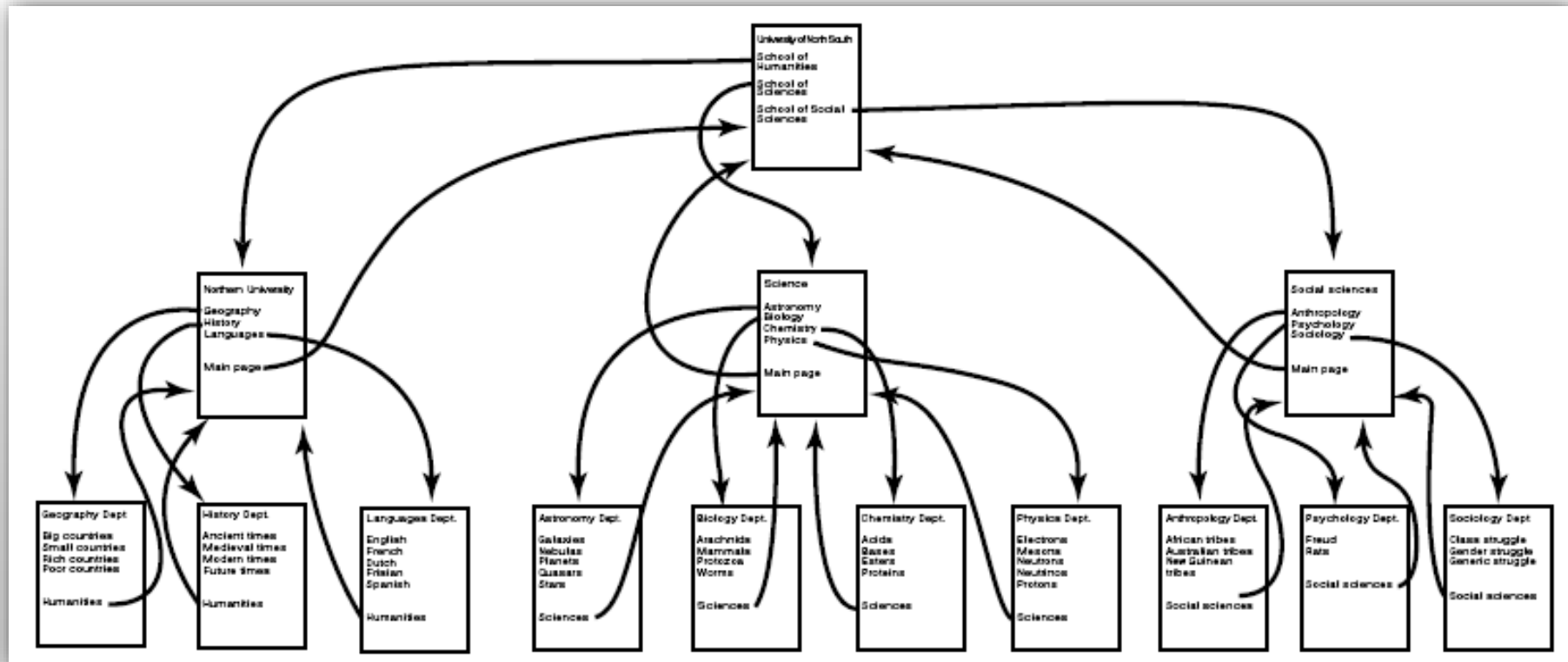
- Her bir katmanda pakete bir başlık eklenir.





Belge Tabanlı Ara Yazılım

- Web, belgelerden oluşan yönlendirilmiş büyük bir çizgedir.
- *document based middleware.*





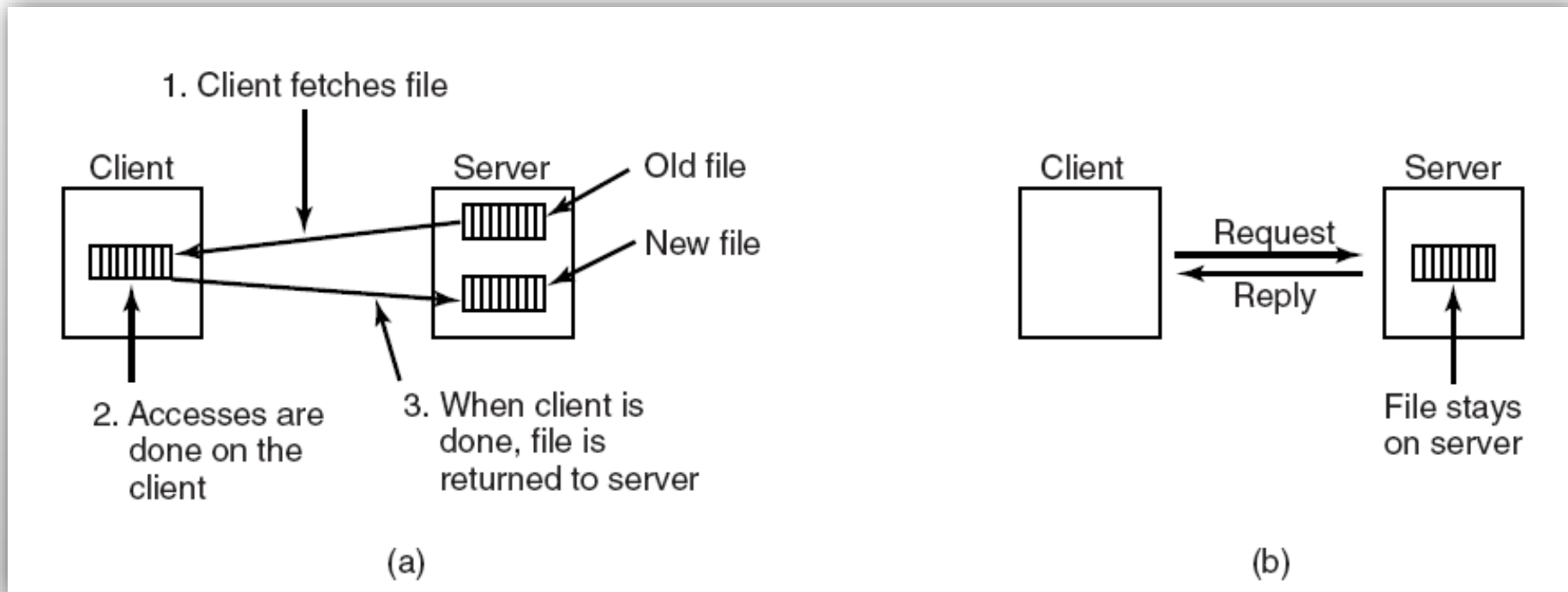
Tarayıcı Nasıl Çalışır

- Tarayıcı *DNS* sorgusu ile www.xxx.org alan adının *IP* adresini ister.
- *DNS*, *XX.XXX.XX.XXX* ile yanıt verir.
- Tarayıcı, *XX.XXX.XX.XXX* adresi *80* portuna *TCP* bağlantısı kurar.
- *xxx.html* dosyasını isteyen bir istek gönderir.
- www.xxx.org sunucusu *xxx.html* dosyasını gönderir.
- *TCP* bağlantısı sonlandırılır.
- Tarayıcı *xxx.html* içindeki metni görüntüler.
- Tarayıcı *xxx.html* içindeki resimleri alır ve görüntüler.



Dosya Sistemi Tabanlı Ara Yazılım

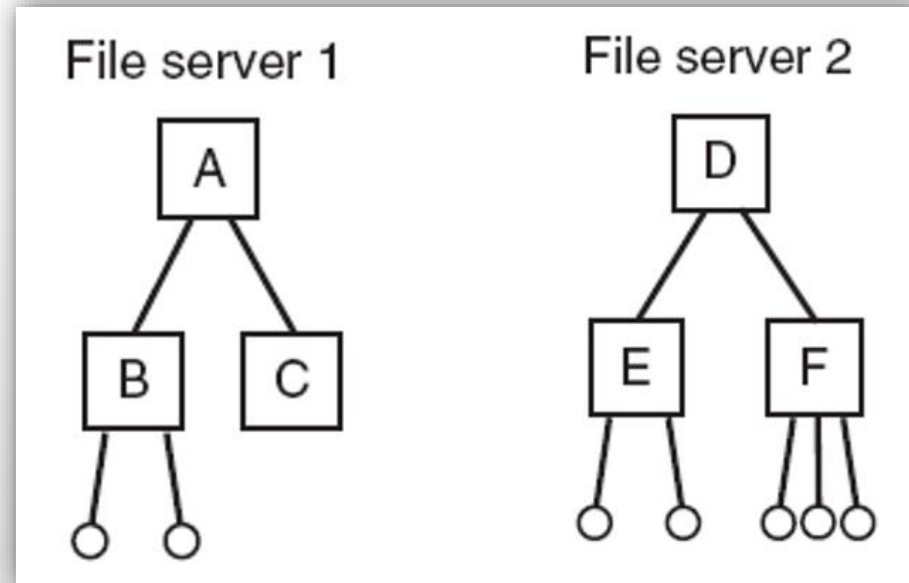
- (a) Yükleme/indirme (*upload/download*)
- (b) Uzaktan erişim (*remote access*)





Dizin Hiyerarşisi

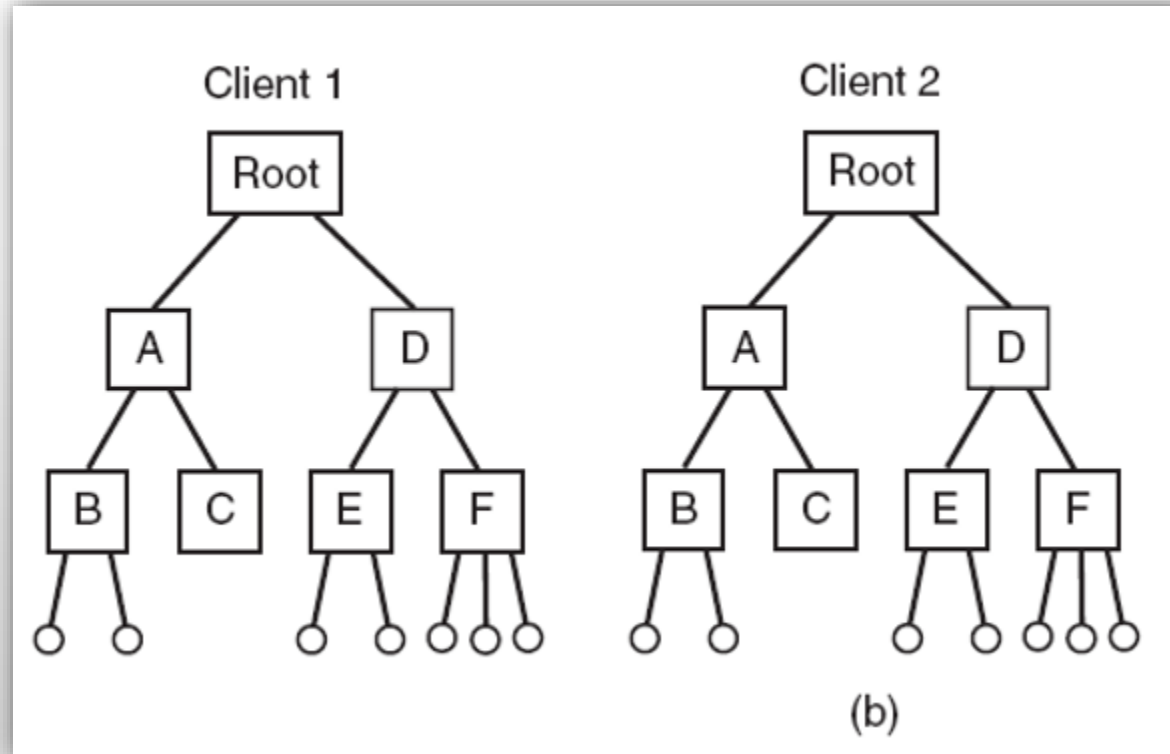
- (a) İki dosya sunucusu. Kare dizinleri ve daire dosyaları temsil eder.





Dizin Hiyerarşisi

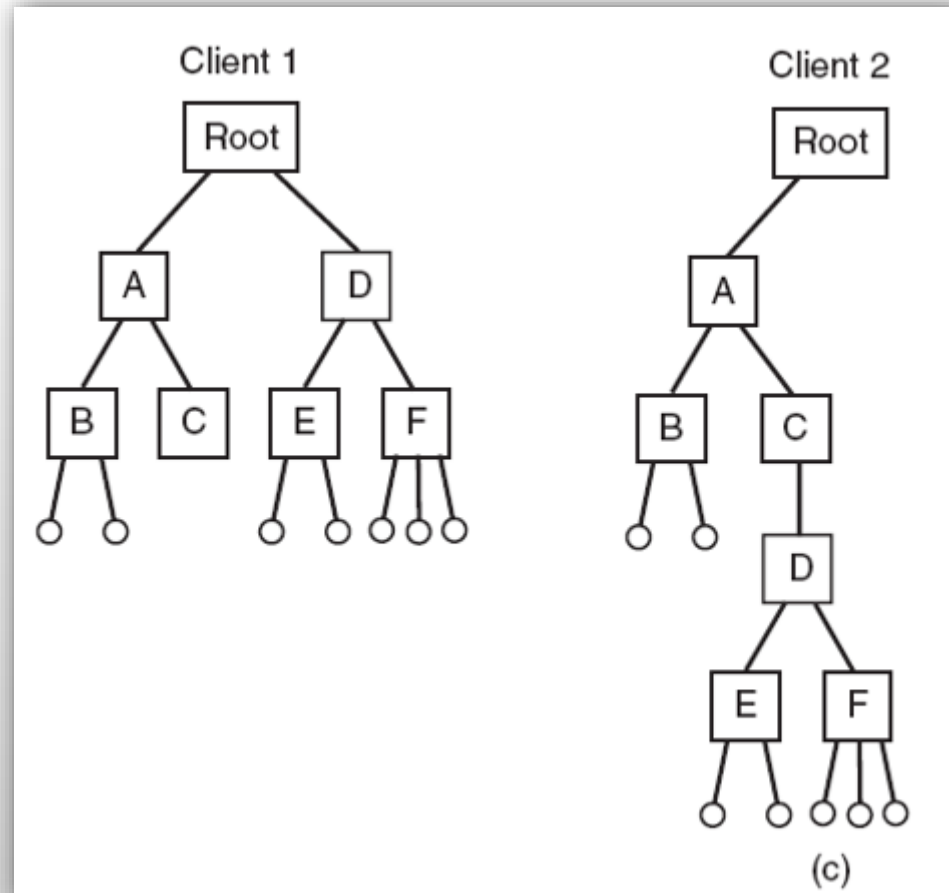
- (b) Tüm istemciler aynı dosya sistemi görünümüne sahip.





Dizin Hiyerarşisi

- (c) Farklı istemciler farklı dosya sistemi görünümüne sahip.





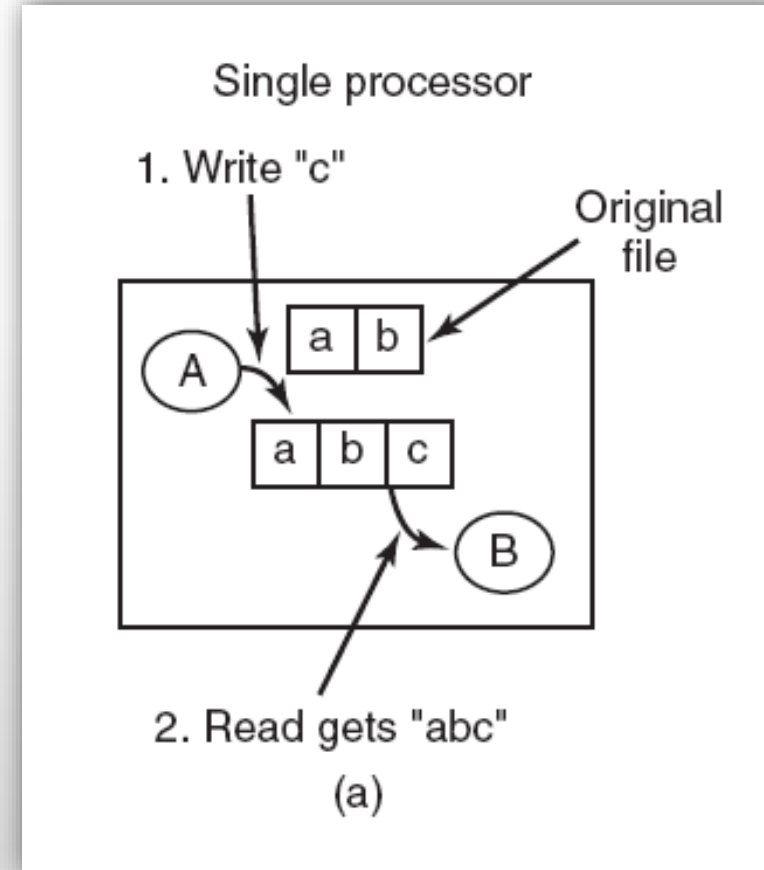
Adlandırma Şeffaflığı

- Dağıtık sistemlerde dosya/dizin adlandırma:
 - Makine + yol,
 - /makine/yol veya makine:yol gibi.
 - Uzak dosya sistemini, yerel dosya hiyerarşisine bağlama (*mount*).
 - Tüm makinelerde aynı görünen tek bir ad alanı (*namespace*).



Dosya Paylaşımının Semantiği

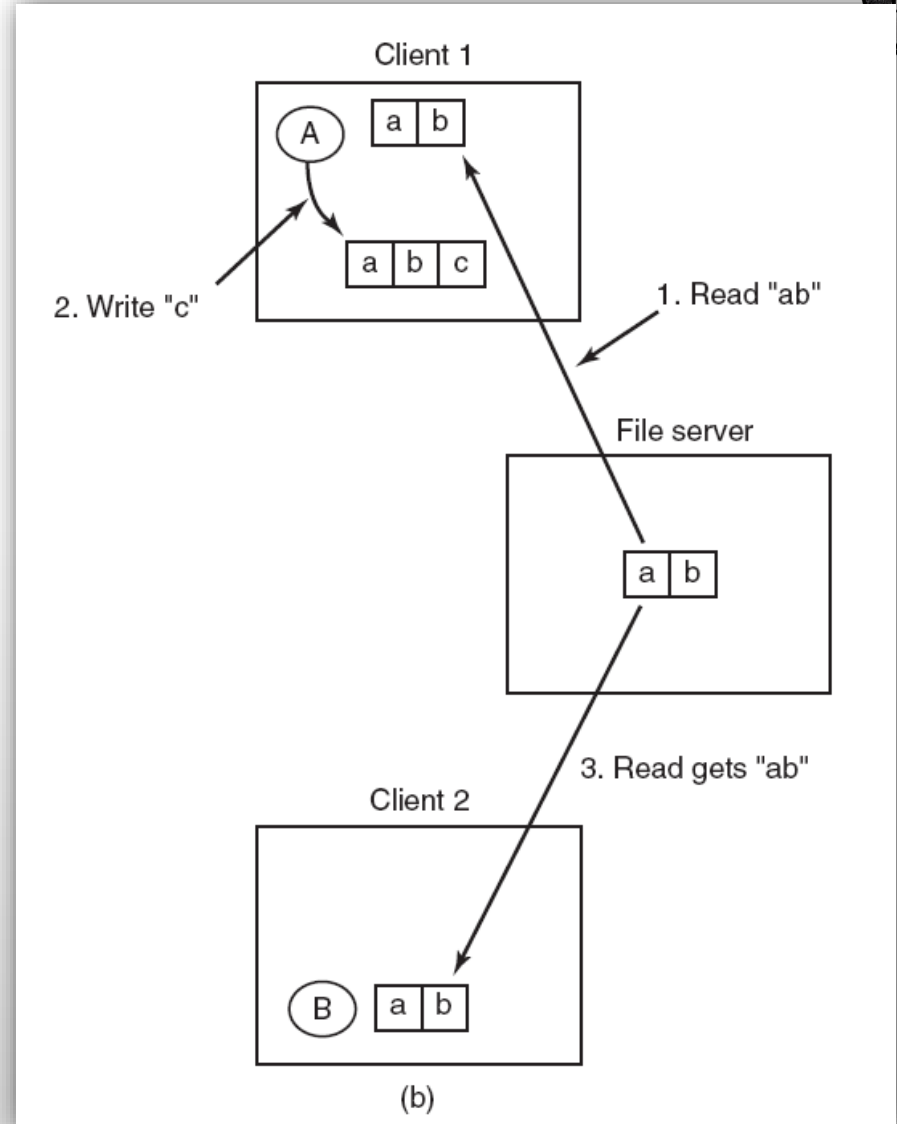
- (a) İşlemlerin sıralı olması tutarlılığı sağlar.





Dosya Paylaşımının Semantiği

- (b) Önbelleğe alma tutarsızlığına neden olur.





CORBA (Common Object Request Broker Architecture)

- Dağıtık yazılım bileşenleri arasında,
 - birlikte çalışabilirlik, ölçeklenebilirlik ve yeniden kullanılabilirlik sağlayan *ara katman* teknolojisi.
- **Nesne yönelimli:** (*object oriented*)
 - Nesne, sınıf, arayüz gibi kavramlara dayalı.
- **Platformdan bağımsız:** (*platform independent*)
 - Nesne ve arayüzleri tanımlamak için, farklı platform ve işletim sistemlerini destekleyen standart bir dil (*IDL*) kullanır.
- **Ağ şeffaflığı:** (*network transparency*)
 - Bir soyutlama düzeyi sağlayarak, nesnelerin ağdaki konumlarından bağımsız birbirleriyle iletişim kurmasına izin verir.



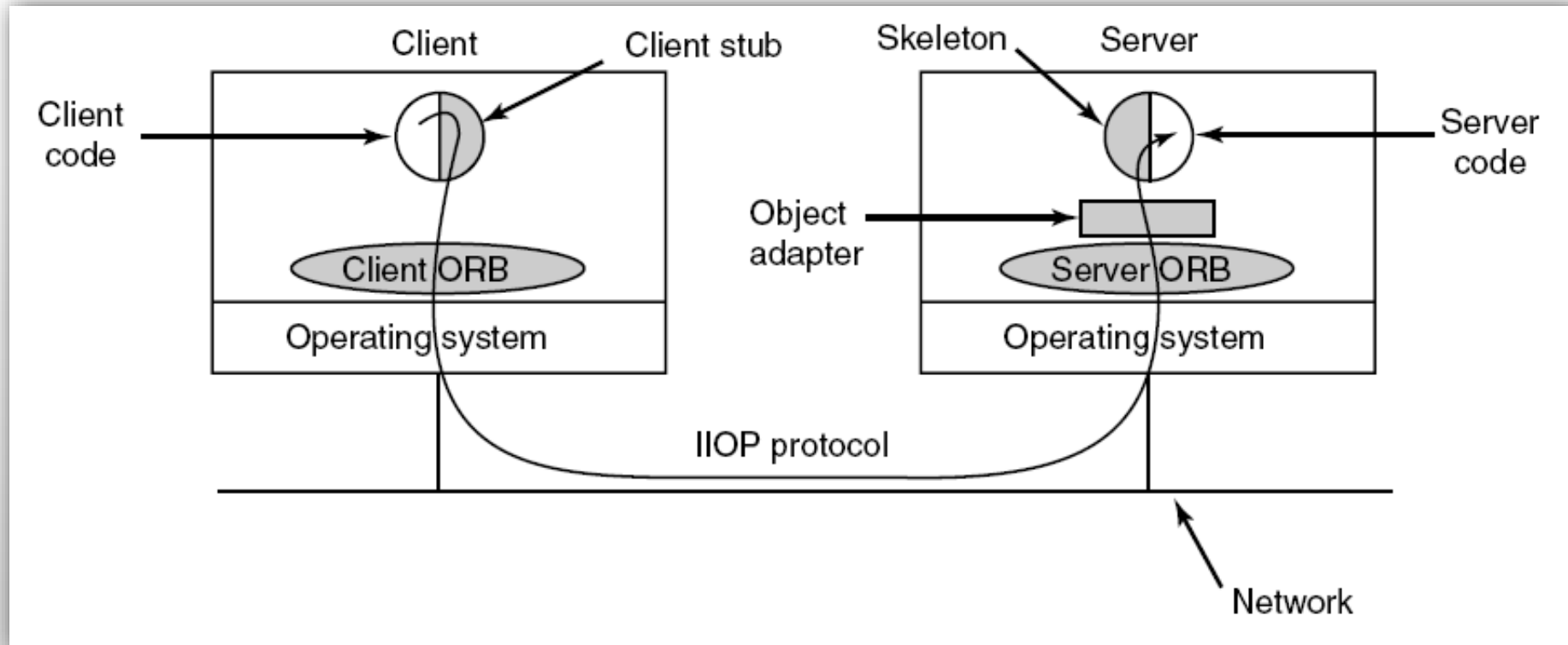
CORBA (Common Object Request Broker Architecture)

- **Nesne İstek Aracısı (ORB):** (*object request broker*)
 - İstemci ve sunucu arasında aracı görevi görür, iletişimi yönetir.
- **Arayüz Tanımlama Dili (IDL):** (*interface description language*)
 - Nesne, arayüz ve metotları tanımlamak için kullanılır.
- **Dinamik Çağırma Arayüzü (DII):** (*dynamic invocation interface*)
 - Nesne metotlarının keşfedilmesi ve çağrılmasına olanak tanır.
- **Statik Çağırma Arabirimi (SII):** (*static invocation interface*)
 - DII'nin önceden derlenmiş sürümü, performansı artırır.



CORBA (Common Object Request Broker Architecture)

- CORBA temel unsurları.





Linda: Koordinasyon Tabanlı Ara Yazılım

- *Coordination-Based Middleware.*
- Dağıtık bilgi işleme için kullanılır.
- Verilerin içeriğine göre saklanabileceği *demet (tuple)* kavramını kullanır.
- Demetler, dağıtık bir veritabanı gibi davranır.
- Demetler ile süreçler arasında *senkronizasyon ve iletişim* sağlanır.
- Linda, paralel işleme, işbirliğine dayalı çalışma ortamları gibi süreçlerin koordinasyon ve iletişim gerektirdiği uygulamalarda kullanılır.



Linda: Koordinasyon Tabanlı Ara Yazılım

- Bağımsız süreçler, soyut bir *demet* (*tuple*) aracılığıyla iletişim kurar.
- *Tuple* veri yapısı,
 - her biri bir tür değer olan,
 - bir veya daha fazla alandan oluşur.

```
("abc", 2, 5)  
("matrix-1", 1, 6, 3.14)  
("family", "is-sister", "Stephany", "Roberta")
```



Eşleşen Demetler (Tuples)

- Üç koşul karşılanırsa demetler eşleşir.
 - Şablon (*template*) ve demet aynı sayıda alana sahiptir.
 - Karşılıklı gelen alanlar aynı türdedir.
 - Şablondaki her sabit veya değişken değer, demet alanı ile aynıdır.



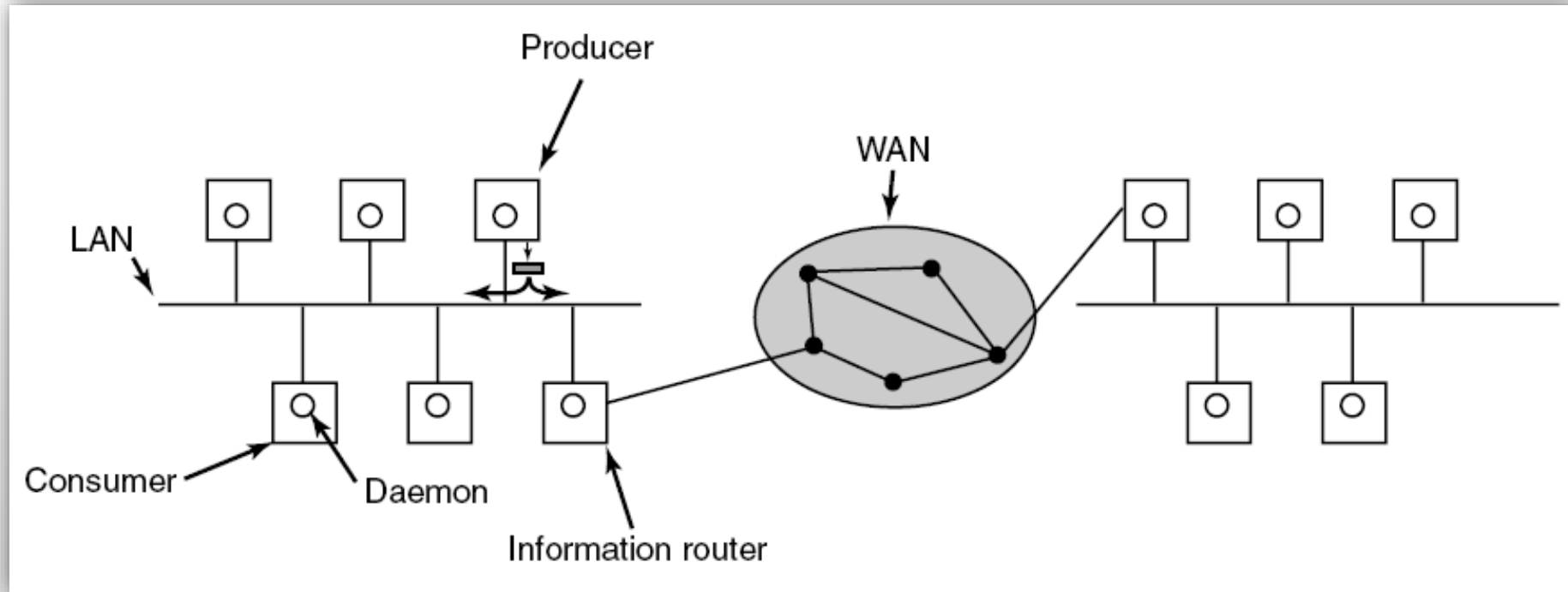
Yayınla Abone Ol Dağıtık Sistemler

- *Publish Subscribe Mechanism.*
 - Üretici bir konuda mesaj yayınlar,
 - Tüketici ilgilendiği konulara abone olur.
- Süreçlerin kimlik ve iletişim protokollerini bilmeye gerek yoktur.
- Aracı, üreticiden gelen mesajı alır, abonelere gönderir.
- Gönderici ve alıcıları birbirinden ayırır. (*bağımsızlık*)
- Aboneler dinamik olarak eklenebilir ve kaldırılabilir. (*ölçeklenebilir*)
- Yeni konular eklenebilir ve bunlara abone olunabilir. (*esnek*)
- Mesajlar abonelere teslim edilene kadar aracıda saklanır. (*güvenilir*)



Yayınla Abone Ol Mimarisi

- Yayınla abone ol mimarisi. (*publish/subscribe*)



Jini



- *Sun Microsystems* tarafından geliştirilmiştir.
- Aygıtlar önceden yapılandırma olmaksızın, bir ağ üzerinden, dinamik olarak birbirlerini keşfedebilir, kullanabilir.
- Birlikte çalışabilirlik için, *Java RMI (Uzaktan Metot Çağırma)* kullanır.
- Dağıtık sistemler için esnek, ölçeklenebilir ve güvenli bir altyapı sağlar.
- Sistem genelini etkilemeden, aygıtlar ve hizmetler eklenebilir, kaldırılabilir, yeniden yapılandırılabilir.
- Merkezi olmayan ve dinamik uygulamaların geliştirilmesini sağlar.



SON