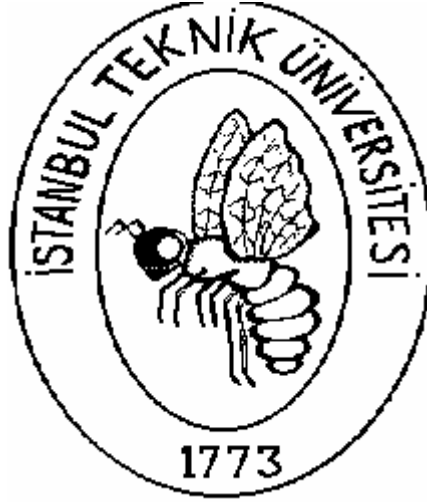


**İSTANBUL TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**BİLGİSAYAR MİMARİLERİNE YENİ
YAKLAŞIM DERSİ
AĞ İŞLEMCİ MİMARİLERİ**



Dönem Projesi

Öğretim Görevlisi

Prof. Dr. Bülent Örencik

Öğrenci

Tuğba Akbal

504061533

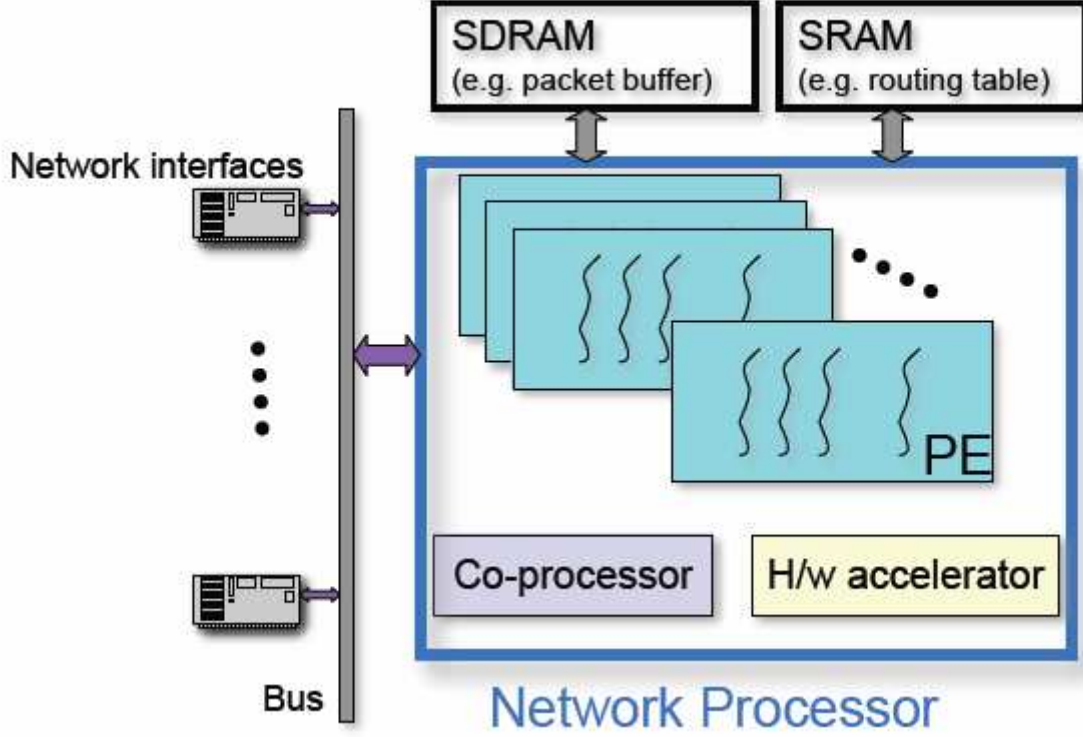
Aralık 2007

İÇİNDEKİLER

1. Genel Bilgi	3
1.1 Ağ İşlemciler Neden Gereklidir?	4
1.2 Ağ Uygulamalarının Esasları	5
1.2.1 Örüntü Eşleştirme.....	5
1.2.2 Arama	5
1.2.3 Hesaplama	5
1.2.4 Veri Kotarma.....	5
1.2.5 Kuyruk Yönetimi.....	6
1.2.6 Kontrol İşlemi	6
1.3 Ağ Uygulamalarının Genel Özellikleri	6
1.3.1 Ağ Protokol Tabakaları	6
1.3.2 Yoğun Veri ve Yoğun Dallanma.....	8
1.3.3 Paket Paralelliği.....	8
1.3.3.1 Paket Düzeyinde Paralellik	8
1.3.3.2 Paketler Arası Paralellik.....	8
1.3.4 Paket Bağımlılığı.....	8
2. Ağ İşlemcilerde Aşılması Gereken Sorunlar.....	9
2.1 Derin Paket Sınıflandırması İşlemi	9
2.2 Güvenlikle İlgili İşlemler	10
3. Ağ İşlemcilerin Sahip Olması Gereken Özellikler.....	10
3.1 Cep Bellek Tasarımı.....	12
3.2 Ana Bellek Tasarımı.....	12
3.3 Çok İplikli Mimariler	12
3.3.1 Süperskaler	13
3.3.2 Büyük İş Parçalı Çok İplikli.....	13
3.3.3 Küçük İş Parçalı Çok İplikli.....	13
3.3.4 SMT.....	13
3.3.5 CMP	14
3.4 Paket Bağımlılığına Çözümler	15
3.5 Örneklerle Ağ İşlemcilerin Genel İhtiyaçları.....	15
3.6 Hızlı Yol – Yavaş Yol Kavramları.....	16
3.7 Bir Ağ İşlemcisini Programlamak.....	16
4. Intel IXP Ailesi.....	17
4.1 Intel IXP2400 Ağ İşlemci Mimarisi.....	18
5. Yeni Nesil Ağ İşlemciler.....	19
6. Sonuç.....	20
7. Açıklamalar	21
8. Referanslar	22

1. Genel Bilgi

Ağ tabanlı uygulamaların gitgide karmaşıklaşması ve ağ trafiğinin artan yoğunluğu sebebiyle, ağ işlemcileri, ağ uygulamalarını hızlandırmak durumundadır. Paket paralelliği ve bağımlılığı, paketlerin sınıflandırılması, ağ uygulamalarının çekirdekleri ve yeni nesil uygulamalar için tasarımsal sorunlarla (derin paket sınıflandırma işi ve güvenlikle alakalı işler) karşı karşıya kalmıştır. Arzu edilen cep bellek ve ana bellek organizasyonu ve paket bağımlılıklarını çözmek için, çok işlemcili ve çok iplikli ortamlar kullanılır.



Ağ işlemcileri çok çeşitli ağ ekipmanları şeklindeki paketlerin üretilmesinde kullanılan özel entegre devrelerdir. Genel amaçlı ağ işlemci mimarileri, yerlerini uygulamada özel komutlu işlemcilere bırakmaktadır. Ağ paketleri, uygulamada çoğunlukla bağımsız ve işleme eş zamanlı olarak tabii tutulan birimlerdir. Ağ uygulamaları bazı spesifik özelliklere sahiptir: veri ve dallanma ağırlıklı olduklarından, düzensizlik, yüksek düzeyde paket paralelliği ve paketler arası bağımlılık söz konusudur. Üst düzey ağ uygulamaları tasarımlarında, bu özellikler detaylı olarak göz önünde bulundurulmalıdır. Ağ işlemciler, aynı zamanda yüksek programlama yeteneği gerektirirler çünkü uygulamalar sürekli gelişmekte ve farklı protokolleri içerebilmektedir. Algoritmalarda sürekli güncellemelere ve sistemde değişikliklere gidilmelidir. Buna ek olarak, yeni ağ araçları taşınabilir de olmalıdır. Örneğin ağ güvenliği uygulamalarında yeni nesil saldırı ve tehditlere karşı gerekli güncellemeler yapılmalıdır. Yeni nesil ağ uygulamaları bazı tasarımsal sorunlar içerir. Bunlar:

Paket Sınıflandırma: Performans kritik bir işidir ve QoS, URL karşılaştırma, virüs algılama, saldırıları tespit etme..vb işlerde gereklidir.

Güvenlikle Alakalı Sorunlar: Web anahtarları ve sunucuları için hayattır ve diğer uygulamalara göre çok daha fazla CPU gücü gerektirir.

gibi önemli sorunlar olabilmektedir.

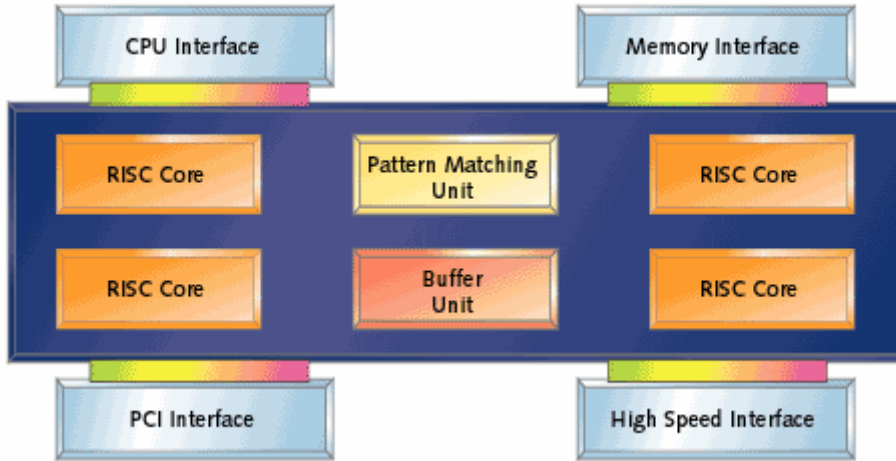
1.1 Ağ İşlemciler Neden Gereklidir?

Ağ işlemcileri, genel amaçlı mikroişlemciler gibi programlanabilir yongalardır, fakat ağ aygıtları içinde paket işlemek için gerekli olduklarından optimize edilmişlerdir. Ağ aygıtları gömülü sistemlerin gelişen bir sınıfıdır ve alışılmış internet ekipmanlarını içerir: yönlendiriciler, anahtarlar, güvenlik duvarları, VoIP köprüler gibi yeni araçlar, VPN geçitler, QOS uygulayıcılar, yük dengeleyiciler, önbellekleme makineleri gibi web-özel araçlar ve SSL hızlandırıcılar gibi.

Ağ işlemci mimarileri CPU mimarilerininin sabit ve sıkıcı karşılanmasına sebep olur. Farklı şirketlerin ağ işlemci tasarımcıları I/O ara yüzleri, bellek ara yüzleri ve programlama modelleri hakkında çok farklı düşünceler geliştirmektedir. Sistem mimarisi ve donanım hızı için gerekli olanlardan bahsetmemektedirler.

Aşağıda genel bir ağ işlemci diyagramı yer almaktadır. Bu özel amaçlı bir ağ işlemciyi temsil etmemektedir ancak genel anlamda içermesi gereken birimler yer almaktadır. Bunlar:

- Çok sayıda RISC çekirdekler
- Genel ağ işlemleri için atanmış donanımlar
- Yüksek hızda ana bellek ara yüzleri
- Yüksek hızda I/O ara yüzleri
- Genel amaçlı CPU için ara yüz



3 farklı yaklaşım var:

- 1.) Saf donanım: ayarlanabilir ASIC → en iyi performans, az esneklik
- 2.) Saf yazılım: programlanabilir → programlanabilme maksimuma çıkarılır
- 3.) Hibrit: programlanabilir ASIC → şu anki gelişim bu yöndedir, performans artarken esneklikte gerçekleşir, alışılmış ağ fonksiyonları donanımla kendi akışında gerçekleşir, saf yazılım yaklaşımından bile daha iyi ölçeklenebilir.

Ağ işlemcileri tarafından gerçekleştirilmesi beklenen işlemler:

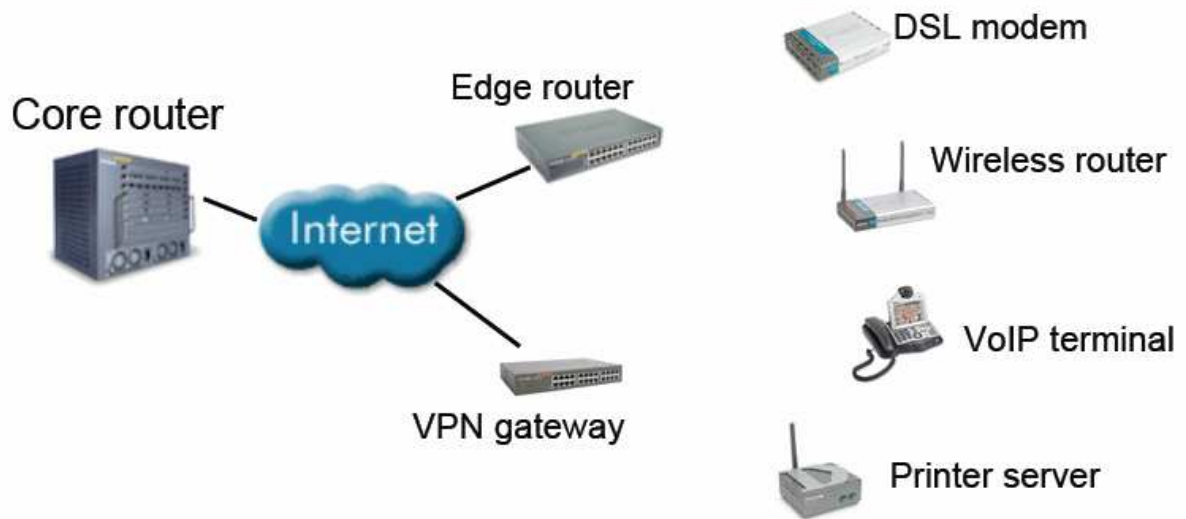
- Veri iletimi için gerekli işlemler (move, load, store, parse) çok fazla işlem gücü gerektirir
- Çok sık olarak kesme gerçekleştirilebilir özellikle de ağ gecikmeleri yüzünden tamponlama limitliyse. Örn: video iletişim uygulamaları
- Sonlu durum makinelerinde yer alan çok sayıda işlem

- Yönlendirme/anahtarlama için arama tablolarında etkin erişim sağlama
- Protokollere bağlı olarak, word ya da byte bağlantılı olmayan işlemler

Yukarıdaki işlevlerinden dolayı ağ işlemcileri, az yer kaplamalı, kod boyutu az olmalı ve az güç harcamalıdır. Ayrıca ağ işlemcileri, mimariden derleyici ve simülatöre kadar ayarlanabilir olmalıdır.

1.2 Ağ Uygulamalarının Esasları

Ağ uygulamaları, yönlendirici, temel anahtar, URL yük dengeleyiciler gibi yeni ekipmanlarda yer alabilirler. Tipik ağ uygulamaları, WAN/LAN anahtarları ve yönlendiriciler, Web cep bellek, yük dengeleme, VoIP ağ geçitleri, güvenlik duvarları ve şifreleme uygulamaları içerir. Ağ işlemcileri genel amaçlı işlemcilere göre çok farklı bir uygulama kümesi içerir çünkü gelen trafiği paket başına baz alır. Paket başlığı ve içeriği bilinçli bir şekilde ele alınmalıdır. Ağ işlemcilerin kullanıldığı alanlar:



olabilir.

1.2.1 Örüntü Eşleştirme

Regular expression ile çalışır ve girdisi bir paket alanıdır. Çıktı olarak ise, mantıksal bir değer döndürür. Bu çekirdek, arama tablolarında kullanılır.

1.2.2 Arama

Bir anahtar baz alınarak, tabloda yer alan spesifik bir girdinin araması gerçekleştirilir. Bu çekirdek, Ipv4ve Ipv6 yönlendirmede kullanılır.

1.2.3 Hesaplama

Bu çekirdek, ağır hesaplama yetisi gerektirir. Bu çekirdeğin basit bir hesaplaması, sağlama toplamı da olabilir.

1.2.4 Veri Kotarma

Bir paket başlığının değiştirilmesi fonksiyonları, veri kotarmanın bir parçası olarak değerlendirilebilir. Ipv4 yönlendirme içindeki TTL(yaşam süresi) alanının azaltılması buna bir örnek olabilir.

1.2.5 Kuyruk Yönetimi

Kuyruk yönetimi işini yapan çekirdektir. Paketlerin iptali, trafiği yönlendiren politikalar, paketlerin saklanması ve Servis Kalitesi uygulamaları gibi uygulamaları içerir.

1.2.6 Kontrol İşlemi

Bu çekirdek tablo güncellemeleri ve işlem anında gerekmeyen istatistikleri içerir.

1.3 Ağ Uygulamalarının Genel Özellikleri

İşlevlerine göre: “veri yolu”, “kontrol yolu” ve “yönetim yolu” olmak üzere 3’e ayrılır.

Veri Yolu: Yüksek performans gerektiren uygulamalardır. MAC ten veri alma ve veri iletimi, paket iletimi, sınıflandırılması, kuyruğa sokma ve zamanlama işi gibi işlerdir. Paketlerin iptal edilmemesi için, iletiminin çabuk gerçekleşmesi gerekmektedir. Paketler arası bağımlılık yoksa paketler paralel işletilebilir.

Kontrol Yolu: Bu uygulamalar daha az zaman kritiktir. Tablo idaresi, yönlendirme, sinyalleşme, politika yönetimi, ATM sanal devre kur/kaldır..vb Az veri paralelliği gerektiren, veri yolu işlerinin yürütülmesini gerçekleştirir.

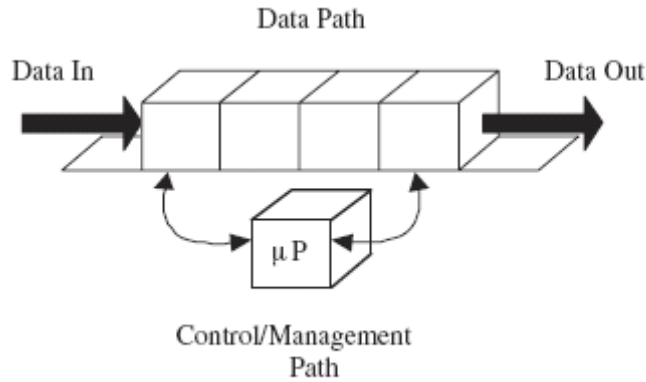
Yönetim Yolu: Sistemi başlangıç durumuna hazırlama/ayarlama ve yönetim protokolleri bu yola aittir. Yönetim yolu işleri, daha az veri paralelliğine ihtiyaç duyar.

Line-rate Path

- .Forwarding
- .Queuing/Scheduling
- .Data Transformation
- .Classification

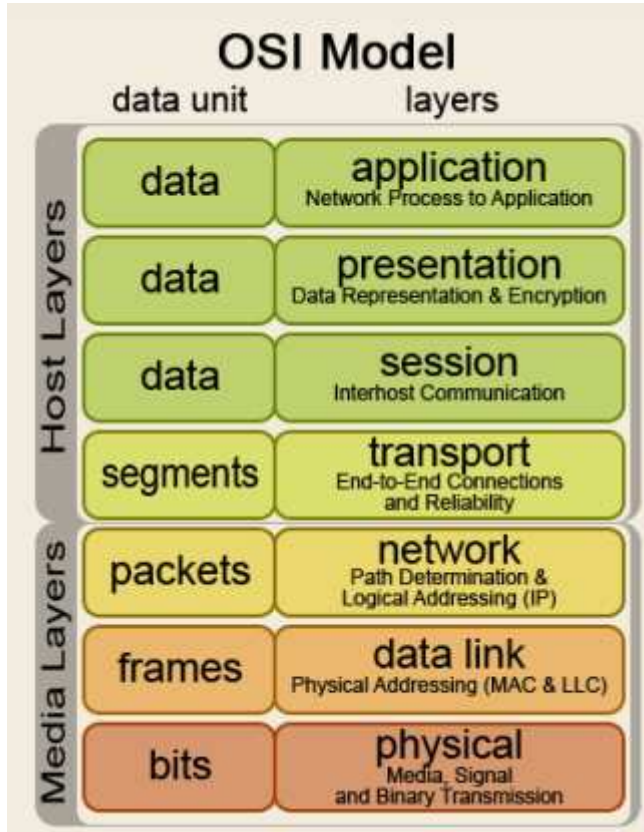
Slow-speed Path

- .Routing Processing
- .Error Processing
- .Statistics Reporting
- .Configuration, etc



1.3.1 Ağ Protokol Tabakaları

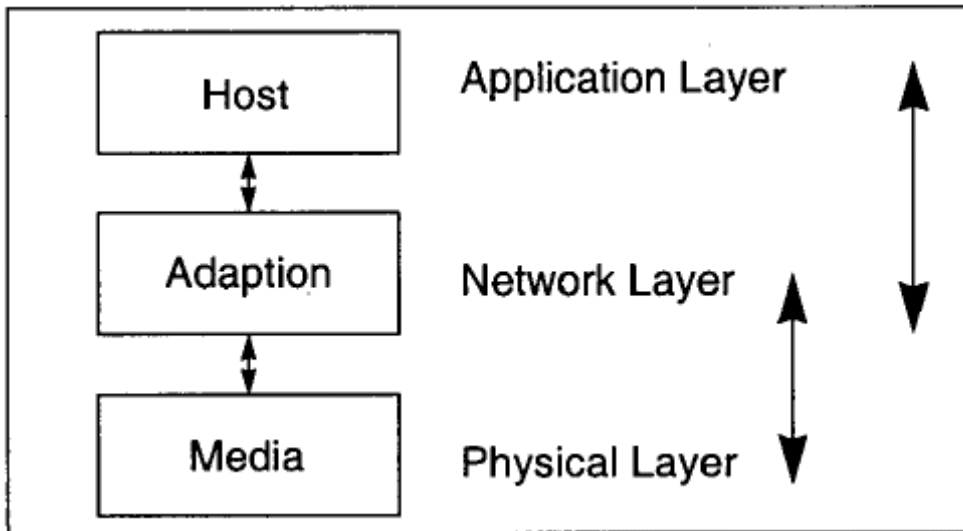
OSI/ISO iletişim modelleri baz alınacak olursa, 7 mantıksal tabaka söz konusudur. Bu tabakaların işlevleri, uygulamada karşılık gelen anlamlarıyla eşleştirilirse, tüm sistemi bu 7 tabakayı içerecek şekilde, çok üst düzeyde bir mimari ile tanımlamak mümkün olabilir.



Karmaşıklığı azaltmak için, birçok ağ çok sayıda tabakadan oluşacak şekildedir. ISO tarafından OSI ağ protokolleri modeli geliştirilmiştir. 7 tabakadan oluşur: fiziksel, veri bağı, ağ, iletim, oturum, sunum ve uygulama katmanları. OSI çok seyrek olarak bu 7 katmandan oluşur, örneğin TCP/IP sunucu ve oturum tabakalarını içermez. Bu tabakalı yapı, paketleri aynı anda işlemek için paralellik imkânı yaratır.

OSI ağ tabakalarının işlevlerine göre bölmelenmesi, durumdan duruma göre değişir. Daha düşük hızda bir ağ için OSI oturum tabakası, iletim tabakası ve ağ tabakası fonksiyonlarını da konak tabakaya izdüşürmek avantajlı olabilir. Yönlendirici ya da anahtarlar gibi yüksek hız gerektiren ekipmanlarda, iletim ve ağ tabakası fonksiyonlarını sistem adaptasyon tabakasına taşımak gereklidir. Dış Ortam tabakasının, modülasyon ve demodülasyon gibi fiziksel tabaka fonksiyonlarını gerçekleştirme tercih edilmektedir.

Sistem 1 den 7 ye kadar tüm tabakaları işleyecek protokollerin fonksiyonlarını gerçekler. Sistem mimarisini tanımlamak için, 3 sistem tabakasından bahsetmek yeterli olacaktır: konak tabaka, adaptasyon tabakası ve dış ortam tabakası. Her bir tabaka, amacına uygun şekilde ve uygulama alanına göre sınıflandırılır.



Konak tabaka, web browsing ve telnet oturumları gibi işlevleri yerine getirir. Bir ya da çok amaçlı işlemciden oluşur. Adaptasyon tabakası, bilgiyi fiziksel iletim için hazırlar ve uygulamaya hazır olan bilgileri konak tabakaya ulaştırır. Dış Ortam tabakası ise, bilginin fiziksel olarak iletimi ve alınması işlerini yürütmekle sorumludur. Donanım tabanlı veya bir

ya da daha fazla sinyal işleme algoritmaları için programlanan DSP(Sayısal Sinyal İşleme) içerecek şekilde gerçekleştirilebilirler.

1.3.2 Yoğun Veri ve Yoğun Dallanma

TL, ROUTE, DRR, NAT gibi programlar, fazla sayıda komut gerektirmektedir. Bu işlemler, yönlendirme tablosu üzerinde gerçekleşen ve veri ağırlıklı programlar olmalarına karşın, örneğin güvenlik duvarı uygulaması olan IPCHAINS bile CRC gibi bir uygulamadan 6 kat daha büyük bir dallanma oranı gösterir. Bu da gösteriyor ki, ağ uygulamaları veri yoğun çalıştıkları kadar, dallanma yoğun da çalışmaktadır.

	Load/Store Ratio(%)	Branch Ratio(%)	IPC	Branch Prediction Rate(%)	I-Cache L1 Miss Rate(%)	D-Cache L1 Miss Rate(%)	Unified L2 Miss Rate(%)
CRC	23.8	2.5	1.05	98.2	0.0	0.7	50.1
TL	76.5	5.2	1.88	94.2	0.0	1.3	0.8
ROUTE	75.6	5.4	1.90	95.1	0.0	1.3	0.9
DRR	74.9	5.7	1.85	94.4	0.0	1.3	1.0
IPCHAINS	13.9	15.3	1.70	95.3	0.0	0.1	55.6
NAT	76.1	5.1	1.90	95.2	0.0	1.4	0.9
DH	22.9	11.0	1.57	89.8	0.0	0.0	49.9
MD5	27.2	7.4	1.67	96.4	0.0	0.1	49.4
URL	26.5	11.9	0.71	93.7	0.0	6.8	8.5
Average	46.4	7.7	1.58	94.7	0.0	1.5	24.1

1.3.3 Paket Paralelliği

Ağ uygulamalarında, paket ana iş birimidir. Tipik internet paketleri boyutlarının dağılımları üzerine yapılan bir araştırmaya göre, paketlerin %50 sinden fazlası 44 byte üzerindedir. Paketler birbirinden bağımsız ya da eşzamanlı işleme tabi tutulabilir. Komut düzeyinde ve iplik düzeyinde paralellik dışında, ağ işlemcilerinde birçok paralellik söz konusudur.

1.3.3.1 Paket Düzeyinde Paralellik

Eğer gelen paketler, işlenmekte olan diğer paketlerden bağımsızsa, gelen paketleri ağ işlemcisinin çok sayıda bağımsız işletme biriminde eş zamanlı olarak yürütülmesi sağlanabilir. Bu şekildeki paralelliğe “paket düzeyinde paralellik” denir ve birçok ticari ağ işlemcisi, çok sayıda RISC çekirdeğinin tek bir işlemcide birleştirilmesi ile bunu gerçekleyebilir.

1.3.3.2 Paketler Arası Paralellik

Her paketin işlenmesi boyunca, bir paketi işleme sokacak işler eğer birbirinden bağımsızsa, bunlar paralel olarak yürütülebilir. Örneğin, kaynak MAC adres kotarması ve varış MAC adres kotarması, 2.tabakada eşzamanlı olarak yürütülebilir. Bu türlü paralelliğe “paketler arası paralellik” denir.

1.3.4 Paket Bağımlılığı

Paket bağımlılığı, doğru işlem sonucu elde etmek için gerekli olan senkronizasyondur. Çok işlemcili ya da çok iplikli bir ortamda, paketlerin peş peşe işlenmesi gerektiğinde, 2 paket arasında paket bağımlılığı olabilir ya da olmayabilir. Eğer paketler statik kurallarla işleme tabi tutuluyorsa, paket bağımlılığı söz konusu olmaz çünkü her pakette çalışan kod, kuralları değiştirme ihtiyacı duymaz. Bu nedenle, eğer paketler bağımsız olarak işleme sokuluyorsa, senkronizasyona ihtiyaç yok demektir. Eğer paketler aynı TCP bağlantısından ise ve şifreleme amaçlı bellekte durum değiştirme ya da iki paket arasında TCP durum düzenlemesi yapılıyorsa, bir paket bağımlılığı söz konusu demektir. Paket bağımlılığı aynı zamanda trafik

yönetimi sayaçları, yönlendirme ya da adres dönüşüm tablolarında gerçekleşir. Eğer bir paket bağımlılığı varsa, bir çeşit senkronizasyon gerçekleşmelidir.

Paket bağımlılığı analizi, belli bir paket çerçeve boyu söz konusuysen ele alınan bir paketin, diğer paketlerle olan bağımlılığını ölçer. Sonuç olarak, 100 paket için sadece TCP akışının olduğu bir çalışma alanında, %14.7 gibi bir paket bağımlılığı söz konusudur [3]. Başka deyişle, 100 paketin o anda kullanıldığı varsayımı ile, sonraki paket ancak ortalama 14.7 paket işleme sokulduktan sonra işleme alınabilir. Paralel ağ işlemcilerde yüksek performansın yakalanması konusunda, paket bağımlılığı çok kritik bir sorun haline gelmiştir.

Öte yandan, web anahtarları için çok ciddi paket bağımlılıkları mevcuttur. Çok sayıda sınıftan paketler alındıkları sırada işlenmelidir. Web anahtarı bir TCP bağlantısının anlamsallığını sağlar yani aldığı paketleri URL i çözümleyene dek tamponlar. İsteği bir web sunucuya iletirken, web anahtarı gönderilen fakat henüz alındığı bildirilmeyen paketleri saklamalıdır. Böylece bu paketler yeniden gönderilebilir.

Bu bağımlılıklara rağmen, bir web anahtarı hala paralellikten faydalanabilir mi? Nasıl faydalanabilir? Eğer paketler sıralanabilirse, belli bir bağlantıya has paketler aynı RISC çekirdeğe gider, bu yüzden bu bağlantı için paketler sırayla işlenir ve paketler arası bağımlılıklar gözlenir.

Eğer bir ağ işlemcisi değerlendirilirse, ne çeşit paketler arası bağımlılıklara sahip olduğu ve ağ işlemcilerin bunları ne şekilde çözeceğine dikkatli bir şekilde karar verilmelidir. Ağ işlemcileri çok yüksek hızdaki, paketler arası bağımlılıklar için hazırlıklı olmayan trafikler için tasarlanmıştır. Bu nedenle uygulama düzeyinde iş gören ağ aygıtları için uygun olmayacaktır.

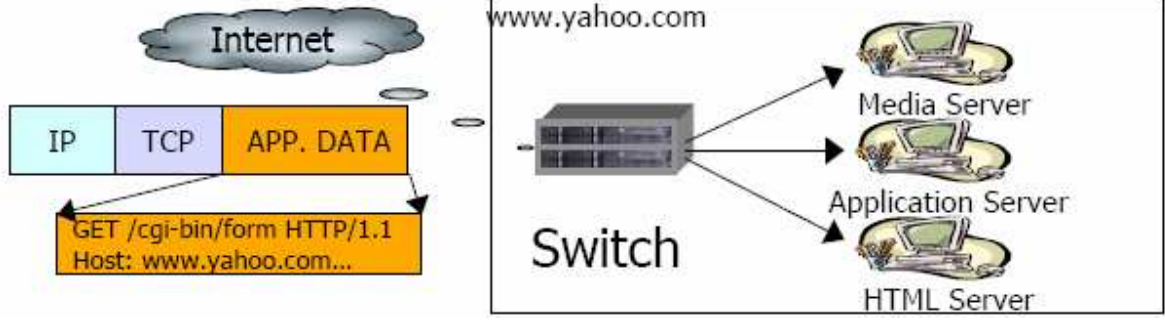
2. Ağ İşlemcilerde Aşılması Gereken Sorunlar

2.1 Derin Paket Sınıflandırması İşlemi

Paket sınıflandırma işi, veri yolu işlemlerinin ilk adımındır. Sınıflandırma türleri: tek alanlı sınıflandırma işi, çok alanlı sınıflandırma işi ve derin paket sınıflandırma işi olarak ele alınabilir. Tek alanlı sınıflandırma işi, paket başlığı içerisindeki tek bir alanın incelenmesi ile gerçekleşir ve paket iletimi uygulamaları bu türdür. Çok alanlı sınıflandırma işi, güvenlik duvarı paket filtreleme ya da InterServ ve DiffServ tarzı QoS uygulamalarında olduğu gibi, paket filtresi tabanlı uygulamalarda birçok alanın ele alınması ile gerçekleşir. Derin paket sınıflandırması işi, paket başlığı olduğu kadar paket içindeki verinin içeriğinin de incelenmesi ile gerçekleşir. Bu tür sınıflandırma, sunucunun yük dengelemesi, saldırı tespiti, virüs tarama gibi içerik-bilinçli uygulamaların önem kazandığı bu dönemde, çok daha kritikleşmiştir.

Örneğin Web anahtarlama işi içerik bilinçli gerçekleşmelidir:

- Yalnız bir Sanal IP
- 5. Tabaka tabanlı yola çıkmış paketler
 - IP ve TCP ye ek olarak, uygulama verisi de incelenir
- 4. Tabaka üzerinden anahtarlamanın avantajları
 - Daha iyi yük dengeleme: içerik türüne göre dağıtım gerçekleştirilir
 - Hızlı cevap: cep benzerliğinden faydalanılır
 - Daha iyi kaynak yönetimi: veritabanı bölümlenir



Tüm bu uygulamalar, çok yüksek hızlarda gerçekleşmelidir. Alışılmış arama tabloları gibi çözümler, derin paket sınıflandırması için yetersizdir çünkü onlar tek alanlı sınıflandırma için kullanılan ideal yöntemlerdir.

2.2 Güvenlikle İlgili İşlemler

Web anahtarları ve sunucuların önem kazanma nedenleri olan e-ticaret, banka ve finansal işlemler gibi hayati uygulamalar, güvenlikle alakalı işlemlerin önem kazanmasına yol açmaktadır. SSL ve TLS güvenlikle alakalı işlemlere örnek olarak verilebilir. Genellikle, güvenlikle alakalı işler CPU yükü fazla olan ve diğer ağ uygulamalarına göre daha fazla CPU gücü gerektiren uygulamalardır. Örneğin SSL diğer ağ uygulamalarına göre 5-7 kat fazla hesapsal bir yük gerektirir. HTTPS hareketinin toplam işleme süresinin ortalama %70'i, SSL in güvenli web hareketlerinde işlenmesi için harcanır [3]. Güvenlikle alakalı işlemler de, çok yüksek hız gerektiren işlemlerdir.

3. Ağ İşlemcilerin Sahip Olması Gereken Özellikler

Ağ işlemcileri doğası gereği heterojendir. Uygulamaya özel komut kümesi içeren elemanlarda, yüksek veri oranı içeren basit işler yürütülür. Uzun ve daha karmaşık yürütme zincirleri, homojen çok işlemcilerdeki yazılımlara görev olarak atanır. Bu durumda, koşma anı zamanlama metotları, farklı iş zincirlerinin ihtiyacı olan uygun kaynakları eşit şekilde bölüştürür. Kaynaklar lokal belleği bölüştürerek ayrılır. Aynı kaynak üzerinde yürütülen birbirine bağımlı işler, iletişim kurmakta zorlanmaz, hatta farklı kaynaklar üzerindeki işlerde bir iletişim yolu üzerinden haberleşir. Bir iletişim isteği, iletilecek byte miktarı ile ayırt edilebilir. Farklı paket yolları içeren bir heterojen mimari aşağıda yer almaktadır. Sağdaki mantıksal bir sistem mimarisi yapısıdır. İşlemi uygulayan birimler, sonraki işleri yürütmek üzere paketleri depolayan birleşik bellekler içerir. İş sırası gereği, bir paket seçilir ve yürütülmeye başlanır. O anki paketin işlenmesi bir başka paket için bir göreve aktarılır. Bir görevin yürütülmesinden sonra bir paket o anki kaynağın girdi kuyruğuna sokulur ya da sistem yolu kullanılarak diğer kaynaklara yeniden dağıtılır.

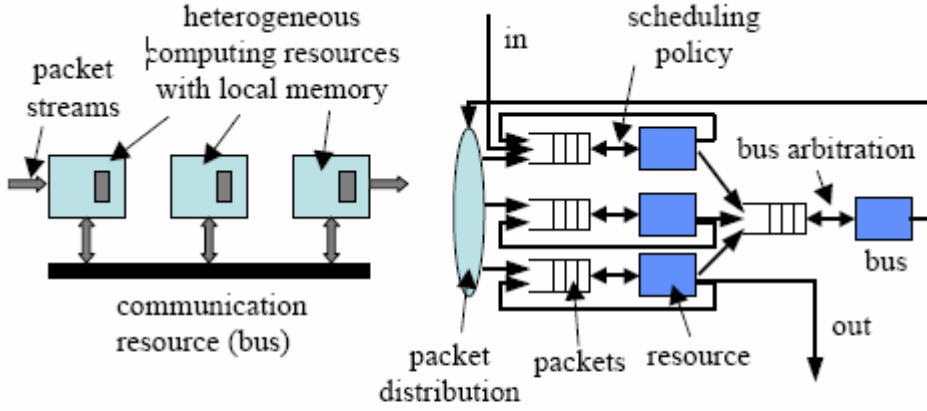


Figure 2. Example of a physical (left) and logical (right) structure of a network processor architecture.

Son 10 yılda, yüksek bant genişliğine sahip ağlar, ağ elemanları bakımından evrim geçirmiştir. İlk tasarımlar CPU'yu harici kullanmıştır. Ancak, genel amaçlı CPU'lar ağ programlama için ideal değildir. Programlanabilir olmaları önemli bir unsur olduğundan, kayan noktalı birimleri kullanılmamış, çok fazla veri cep belleğine ve çok küçük ana bellek bant genişliklerine sahip olmuşlardır. Ayrıca, bant genişliği için mevcut talep, CPU hızına oranla daha fazla artmaktadır. Ağ elemanları tasarımcıları yeni nesil CPU'ların aygıtlarının hızlarını arttırmaları için bekleyememektedir. Hızlı yol-yavaş yol tasarımında dahi, problemler devam etmektedir. Hızlı yollar daha nasıl hızlandırılabilirler?

Bu soruya en iyi cevap, bir ASIC tasarlamaktır. İyi tasarlanmış ASIC'ler, CPU'lardan daha hızlı olabilirler fakat geliştirmesi oldukça pahalı ve zordur. Ayrıca, ASIC'ler genellikle kısıtlı programlanabilirlik özelliğine sahiptir ve protokoller ya da ara yüzler değiştiğinde yeniden tasarlanmalıdır. Ağ işlemci şirketleri, ASIC'ler ve CPU'lar arasında bir köprü kurarak, bir CPU gibi programlanabilen ve ASIC kadar hızlı olabilen bir aygıt sağlamayı hedeflemektedir.

Özetle ağ işlemciler:

- Yüksek performanslı paket işleme yeteneği ve programlanabilirliği birleştiren
- Çok çeşitli ağ işlemci uygulamalarını gerçekleştirebilen
- Etkin kaynak kullanım zorluğunu aşabilen
- Ağ alanında artan istek ve gelişmeleri karşılayan (yüksek bant genişliği, düşük gecikme, yüksek veri oranları, yaygınlaşan ağ anlayışı): artan bant genişliği her zaman iyi sonuç vermez!
- Ağ bant genişliği işlemci hızını geride bırakmaktadır: donanım kaynak kullanımını optimize edebilen
- Bir ağ içindeki paketleri iletmek için gerekli fonksiyonların entegre edildiği programlanabilir yongalar olan
- Gerçek zamanlı işleme, saklama ve ileme, güvenlik, anahtarlama yapısı, IP paket işleme gerçekleştiren
- Gecikme sorununu donanım içindeki ve programlanabilir elemanlardaki paket işleme fonksiyonlarını uygulayarak çözebilen
- Mimari sayesinde hız artışı: paralel dağıtık işleme, iş hatları ile sağlayabilen

- İşlem kapasitesi, esneklik ve ölçeklenebilirlik özelliklerine sahip
- Ağ servisine eklenmesi gereken: derin paket inceleme ve öncelikli veri oranı özelliklerini içeren
- Çalışma süresi ve maliyetinin en aza indirgenmesi: kodun yeniden kullanılması ve ölçeklenebilirlik hedefleyen şekilde gerçekleştirilmelidir.

3.1 Cep Bellek Tasarımı

Ağ işlemcilerde ana sorunlardan biri de, paket sınıflandırma işini çok hızlı gerçekleştirebilmek için cep belleği doğru şekilde tasarlayabilmektir. Yönlendirme tablosu arama işi, bir önek/maskeleye çiftleri tablosundaki bir paketin verilen bir varış IP adresi ile en uzun önekleri kıyaslayarak bulmaktır. Paket grupları içindeki yeteri kadar zamanda yöreselliğin bulunma sebebi, bir yönlendirme tablosu cebinin kullanımını doğrulamaktır.

Paket sınıflandırma işi veri yükü fazla olan bir iştir. Derin paket sınıflandırma işi, daha önce de bahsedildiği üzere, paketlerin veri içerikleri dâhil incelemeye alındığından çok fazla bellek erişimi gerektirir. Bu nedenle, derin paket sınıflandırma sorununu ortadan kaldırmak için, yönlendirme tablosu cep tasarımı konusunda daha fazla araştırma yapılmalıdır.

3.2 Ana Bellek Tasarımı

Paket tamponları ağ işlemlerinin temeli olan paket değiştirmede kullanılır. Var olan paket tampon tasarımları, SRAM/DRAM'ın karışımı olan bir yaklaşım ile en kötü durum bant genişliğini gerektirir. Yeni nesil yönlendiriciler yüzlerce ara yüzden daha fazlasını desteklemeye ihtiyaç duyacak, paket tamponları birkaç Gb kapasitede olacak, bant genişliği yüzlerce Gb/s olacaktır. Bu nedenle, yeni nesil yönlendiriciler için tasarlanmış paket tamponları için, yüksek hızlara erişmenin öncesinde daha fazla araştırma gereklidir.

3.3 Çok İplikli Mimariler

Çok iplikli mimariler, bellek erişim gecikmelerini tolere etmek için geliştirilmiştir. Bir iplikten gelen bir komut, uzun gecikme süreli bir işleme neden oluyorsa, iplik inaktif edilir ve bu ipliğin komut cebindeki başka komutlarının alınması durdurulur. Bu esnada diğer ipliklerin komutları işlenmek üzere iş hattına alınmaya devam eder. Sonuçta, çok iplikli mimariler tek bir işlemcinin işlevsel birimlerinin birçok iplik tarafından ortak kullanılmasını sağlar. Bu özellikle süperskaler makinelerde, ek işlevsel birimler eklenerek yüksek performans elde etme zorluklarını ortadan kaldırırken, bu birimlerin az kullanımından kaynaklanan sıkıntılar göze çarpmaktadır.

Birçok ağ işlemcisi, paralel olarak koşturulan çok sayıda işlemci çekirdekleri içerir. Intel'in IXP'si gibi ağ işlemcileri, birçok içerik için donanım desteği içerir. Peki zamanlama işini ne üstlenir? Aşağıdaki şekilde, 6 paket 4 çekirdekli ağ işlemcilerine yönlendirilmiştir.

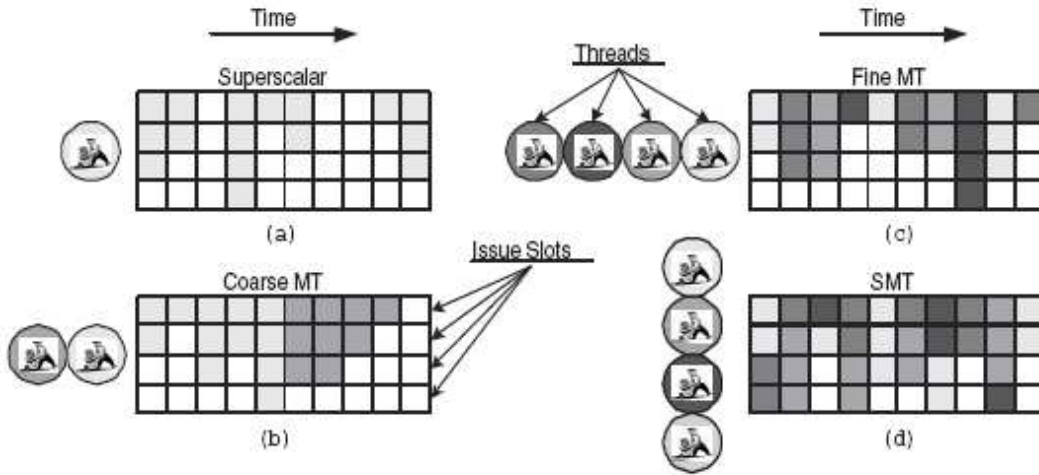


Peki hangi paketler hangi çekirdekte işlenecektir? Bazı ağ işlemcileri içinde, buna donanımla karar verilir. Diğerlerinde bu işi yazılım üstlenmiştir. Uygulamalar ve algoritmalara bağlı

olarak, hangi paketin hangi çekirdeğe gideceğinin kontrol edilmesi yeteneği önemli bir ihtiyaçtır. Diğerleri için, donanım zamanlama hızı hayatidir.

3.3.1 Süperskaler

Şekilde(a) süperskaler bir iş akışının, çok iplik desteği olmadan çalışması gösterilmektedir. Sıklıkla gecikmeye yol açan komut ıskası gibi bir durumda, süperskaler makine her saat çevriminde çalışmaz ve tüm işlemcinin bekletilmesine sebep olur. Bu sebepten dolayı, işlevsel birimlerin az kullanımı sorunu söz konusu olur.



3.3.2 Büyük İş Parçalı Çok İplikli

Şekilde(b) görüldüğü gibi, ipliklerin değiştirilmesi ve işlenen komutlardan her saat çevriminde yalnız bir tanesi alınması sağlanır. Avantajı, kısa ve uzun süreli gecikmeleri saklamasıdır. Dezavantajı ise, işlenmeye hazır olan komutların işlenmesini yavaşlatmasıdır çünkü diğer ipliklerden gelen komutlar yüzünden işlenmeye hazır olan komutlar bekletilir. Her saat çevriminde yalnızca bir iplik işleme sokulur, hiç tamamen boş işe yer verilmez, fakat bireysel saat çevrimlerinde çok fazla sayıda boş bekleme süresi yer alır.

3.3.3 Küçük İş Parçalı Çok İplikli

Şekilde(c) iri parçalı çok iplikli bir iş akışı gözükmektedir. Bu durumda, L2 cep ıskası durumunda olduğu gibi, uzun süreli gecikmelerde o an işleyen iplik değiştirilir. Bu modelde, işlenmeye hazır komutların işlenme süresi yavaşlatılır çünkü diğer ipliklerin işlenmesi uzun sürmektedir ve bu modelin en önemli avantajıdır. Dezavantajı ise, kısa süreli işlerde kendini gösterir çünkü kısa süreli sık gecikmeler, iş hattının başlama süresini uzatır. İpliklere geçmek için gerekli başlama süresinin uzaması sebebi ile tamamen boş durumda bekleme süresi uzar. Aynı zamanda, her saat çevrimindeki boş kalma süresi aşırı artar çünkü her saat çevriminde yalnızca tek bir komut işlenebilir.

3.3.4 SMT

Şekilde(d) görülmektedir. Tek bir saat çevriminde, çok sayıda iplikten gelen çok sayıda komut işlenir ve dinamik olarak ardarda TLP ve ILP düzeyinde kullanılır. Saklayıcı yeniden isimlendirme ve dinamik zamanlama sayesinde, bağımsız ipliklerden gelen çok sayıda komut aralarında hiçbir bağımlılık olmaksızın işleme sokulur. SMT aşırı donanımsal maliyet ve büyük mimari değişimlere yol açmaksızın, performansı yükseltebilir. Detaylı bir açıklama aşağıda yer alacaktır. Bellek gecikmeleri, paket işleme gibi ağ uygulamalarında potansiyel bir

sorun olduğundan, çok iplikli yapı bellek gecikmelerini saklamak ve yüksek işlem hızı elde etmek için ideal bir çözümdür.

Süperskaler ve VLIW de olduğu gibi, modern çok işli işlemciler tek iplikli programlara göre daha fazla işlevsel birim içerdiğinden daha etkin kullanılabilir. Çok sayıda iplik, tek bir işlemcinin işlevsel birimlerini ortak kullanabilir. SMT çok ipliklilerin bir türüdür ve tek bir çevrimde aralarında hiçbir bağımlılık olmayan çok sayıda ipliğin birçok komutunu aynı anda işleyebilir. Dinamik olarak işleri TLP ve ILP bazında sıraya sokar. SMT’de amaç, uzun bellek gecikmeleri ve her iplik arasında kısıtlı paralellik sorunları çerçevesinde, işlemci kullanımını artırmaktır.

Temel SMT modeli aynı anda çok sayıda paralel ipliğin aktif olmasına izin verir. SMT mimarisinde, temel süperskaler iş hatları değişmemektedir, fakat her bağımsız ipliğin bağlamını saklayacak program sayaçları, komut kuyukları ve dallanma öngörülerini gibi kaynaklar paylaşmaktadır. SMT işlemcilerde, çok sayıda aktif ipliğin komutları işlendiğinden ve peş peşe yürütüldüğünden, işlevsel birimler, komut kuyukları, yeniden isimlendirme saklayıcıları, cep bellekler, TLB’ler gibi paylaşılan tüm donanımsal kaynaklar, her çevrimde iplikler tarafından elde edilmeye çalışılır. Eğer iplikler aynı kaynaklar için talepte bulunursa, bu çakışmaya sebep olur ve daha az sayıda komutun işlenmesiyle sonuçlanır. Eğer iki paralel iplik, SMT işlemcilerdeki birbirini tamamlayan iki kaynağa çatışmasız ihtiyaç duyarsa, kaynakların kullanımı artacaktır. SMT piyasada yaygın olarak kullanılmaktadır. Compaq Alpha 21464, Hyperthreading teknolojisi içeren Intel Pentium 4 işlemci ve IBM Power5 SMT tabanlı işlemciler kullanılmaktadır.

SMT performansı aşırı donanımsal maliyet ve büyük mimari değişimler gerektirmeden artırdığı için, yüksek performans ve paralel teknoloji gerektiren geleceğin ağ işlemcilerinde kullanılacak bir mimari olamaya uygundur. SMT işlemciler, FGMTI CMP, SS işlemcilerle göre aynı kaynağa sahipken, daha üstün performans sergiler (3 farklı çalışma alanında denemeler yapıldığında (IP iletimi, MD5, 3DES) ve ağ uygulamalarının karakteristik özellikleri yer almadığında). Aynı bir denemede SMT tabanlı, sıraya koyma işi için donanım kuyuklarının ve iplik işleme düzeyinde yük dengeleme mekanizmalarının yer aldığı bir ağ işlemcisi ele alındığı, SMT’nin denendiği bir simülasyon ortamında, 2 farklı karşılaştırma maksatlı program farklı senaryolarla denendiğinde (IP arama ve MD5), SMT’nin ağ sunucuları üzerindeki etkisi ölçülmüştür. Bu etkiler Web, FTP ya da Hyperthreading içeren IntelIXon kullanan dosya sunucuları üzerinde gözlemlenebilir. SMT desteğinin iş yüklerine hassas olduğu ve OS ağırlıklı iş yüklerinde çok fazla bir fayda sağlamadığı görülmüştür.

CNP810SP SMT yeteneklerini içeren bir ağ işlemcisidir. Bu işlemci, eş zamanlı 8 ipliğe kadar yürütebildiğini ve her ipliğin her çevrimde 0 ila 3 komut yürütebildiğini göstermektedir. Buna rağmen, bu işlemci piyasada tutulmamaktadır.

3.3.5 CMP

CMP’ler, aynı birim üzerinde tüm mimari kaynakları içeren çok işlemcilerdir. CMP TLP’yi farklı iplikleri farklı işlemcilerde paralel olarak yürüterek gerçekleştirirken, SMT TLP’yi birbirinden farklı geniş bir yelpazeden iplikleri tek bir işlemci üzerinde gerçekleştirir. CMP genel amaçlı işlemcilerden daha basit olan tek iplikli işlemcilerden oluşur ve aynı anda çok sayıda iplik paralel olarak, çok sayıda tek iplikli işlemci üzerinde yürütülür. Eğer bir uygulama, ipliklere bölünemiyorsa, tek iplikli işlemci çekirdekleri tam verimle kullanılamazlar. En kötü durumda, eğer uygulama alt ipliklere bölünemiyorsa, sadece tek bir iplik işlemci çekirdeğine iş olarak atanır. SMT’de bahsedildiği gibi: SMT ağ

uygulamalarında, büyük taneli paralel programları ortalama %60 yürütebilen CMP'ye göre daha üstün performans sergiler. Bunun sebebi, SMT'nin donanım kaynaklarından dinamik olarak yer alır, oysa ki CMP aynı donanım kaynaklarını statik olarak bölmeler. Bu da SMT'nin sistem kaynaklarını CMP'ye göre daha etkin kullanması anlamına gelir. Ancak CMP'nin benzer çalışma alanlarında, daha fazla sayıda tek iplikli işlemci çekirdekleri içerir çünkü bir tek iplikli işlemci çekirdeği ona eş bir SMT işlemciden daha az çalışma alanına ihtiyaç duyar. Eğer uygulama, bir tek-iplikli işlemci çekirdeği üzerinde yürütülen iplikler, etkin şekilde çok sayıda ipliğe bölünebilirse, SMT den daha başarılı sonuçlar elde edilebilir. Tek-iplikli işlemci çekirdekleri SMT işlemcilerden daha basit oldukları için, tasarımı ve optimize etmesi SMT'lere göre daha kolaydır. Paket düzeyinde paralellik içeren ağ uygulamalarında, spesifik bir paket işleme işini yürütmek için tasarlanmış bir tek-iplikli işlemci çekirdeği, paketleri paralel olarak işlemek için iş hattı mantığını uygulayabilir. En son çıkan ağ işlemcilerden Intel IXP2800, karmaşık algoritmaları, derin paket denetlemelerini, trafik yönetimini ve iletimini yüksek hızlarda gerçekleştirmek için, tek bir yonga üzerine bir paralel işlem tasarımı yerleştirmiştir. Mimari açıdan, 17 tane 32 bitlik bağımsız çok-iplikli mikro makine ile Intel Xscale çekirdeğin birleşimidir. Mikro makineler, yüksek hız gerektiren ASIC'lerdeki işlemci gücünü karşılamak için kullanılır.

IBM PowerNP de aynı zamanda CMP stilinde ağ işlemcisidir. 16 tane basitleştirilmiş RISC piko işlemci ve bir PowerPC çekirdekten oluşur. Her bir piko işlemci 2 donanım izleğini ve komut al/komut çöz/komut yürüt gibi 3 aşamalı iş hattını destekler. Piko işlemciler veri yolu işleri için ve PowerPC çekirdek kontrol/yönetim işleri için kullanılır.

3.4 Paket Bağımlılığına Çözümler

İki farklı yaklaşımla da bu sorun çözümlenebilir: donanımsal ve yazılımsal. Yazılımsal yaklaşımda, bağımlılığı olan paketleri işlemek için koda yazılım kilitleri konur ve o an yürütülmekte olan bağımlı paketin işi bitene dek kilit kaldırılmaz. Ancak yazılım kilitleri uygulamada oldukça zordur çünkü paket işleme kodları genelde çok büyüktür ve kodu analiz ederek kilitleri doğru yere yerleştirmek oldukça zahmetlidir. Ölümcül kilitlenmelerle karşılaşılabilir. Donanımsal yaklaşımda, paket sınıflandırma donanımı kullanılır. Ancak donanımsal yaklaşım performansı düşürebilir çünkü tüm paketleri ihtiyaç olsun olmasın ard arda işler. Paket bağımlılığını çözmek için Melvin, bir "yazma tablosu", bir "okuma tablosu", bir "aktif paket listesi" ve bir "çatışma öngörü mantığı"ndan oluşan kavramsal bir donanım çözümü getirmiştir. Yazılımda ise herhangi bir düzenleme gerektirmemiştir.

3.5 Örneklerle Ağ İşlemcilerin Genel İhtiyaçları

Tüm ağ aygıtları uygulamada aynı ihtiyaçlara sahip değildir. Ancak aralarında birçok benzerlik mevcuttur. Örneğin bir yönlendiricinin ve web anahtarının paket işlemek için gerekli iş adımları ele alınabilir. Bu çekirdek, zaman kritik işlere veri düzeyinde işler denir.

Yönlendiriciler internetin en çok iş gören kısımlarıdır. Yönlendirici çok çeşitli ağ ara yüzlerinden paketleri kabul eder ve onları iptal eder veya bir ya da birden fazla ara yüze yollar. Paketler bir düzine ya da daha fazla sayıda yönlendiriciden geçerek internet üzerinde kendi yollarını bulurlar. Bir IP yönlendirme algoritması şu şekilde olabilir:

- Bağlantı katmanı başlığı silinir
- IP başlığındaki varış IP adresi bulunur
- Bir sonraki gidilecek noktadaki IP adresi için, bir arama tablosu yapılır
- Bir sonraki adımda gidilecek bağlantı katmanının adresi belirlenir

- Pakete bağlantı katman başlığı eklenir
- Paket gönderilmek için kuyruğa alınır
- Paket gönderilir ya da iptal edilir(eğer bağlantı tıkanmışsa)

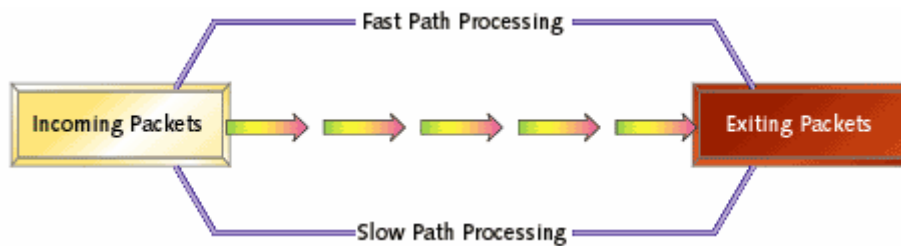
Web anahtarları ise, karşılaştırılacak olursa yeni bir ağ aygıtıdır. Popüler bir web sitesinin cevap verme yeteneğini arttırmak için, tek bir web sunucudan daha fazla kullanarak gerekli sorunları çözer. Bir web anahtarı, URL'in kendisi dâhil, çok çeşitli ağ parametreleri tabanlı farklı sunuculardan gelen HTTP isteklerini yönlendirebilir. Örneğin, tüm güvenli HTTP istekleri bu istekleri hızla sonuçlandırmak için, şifreleme donanımı içeren özel web sunucuları ile iletilebilir. Bir web anahtarlama algoritması:

- Gelen TCP bağlantısını kabul edilir(3 yollu el sıkışma)
- Gelen TCP veri dizisi tamponlanır(TCP/IP protokolü)
- İstenen URL i bulmak için veri dizisi çözümlenir
- İsteğin iletileceği yeri belirlemek için bir arama tablosu yapılır
- Web sunucusu ile TCP bağlantısı açılır(3 yollu el sıkışma)
- Tamponlanan istek gönderilir(TCP/IP protokolü)

Verilen bir bant genişliğinde, web anahtarlama işi çok fazla ihtiyaç gerektirir ve yönlendirme işlemlerinden daha fazla durum gerektirir. Çünkü yönlendirici paketleri, web anahtarları ise bağlantıları gerçekleştirir.

3.6 Hızlı Yol – Yavaş Yol Kavramları

Veri ve kontrol düzeylerinin farklı ihtiyaçları sıklıkla hızlı yol- yavaş yol tasarımında çözümlenir. Bu türlü tasarımda, paketler ağ aygıtına girdiği anda, varış adresleri ve portları değerlendirilir ve inceleme baz alınarak, içten hızlı yada yavaş yola yönlendirilir. Minimum ya da normal işleme tabi tutulacak olan paketler hızlı yola, sıra dışı yada karmaşık işlemlerden geçecek paketlerse yavaş yola alınır. Hızlı yol paketleri veri düzeyindeki işlere karşı gelmekteyken, yavaş yol paketleri kontrol seviyesi işlerine karşı gelir. İşlenen paketler, hem yavaş hem de hızlı yolu terk ederek aynı ağ ara yüzüne doğru ilerlemektedir.



Paketleri bu şekilde ayrı yollardan işlemek esneklik kazandırır. Ancak yavaş yol işlem birimi bir CPU ile gerçekleştirirken, hızlı yol birimi bir FPGA, ASIC, yardımcı işlemci veya bir başka CPU tarafından gerçekleştirilmelidir. Bu mimari oldukça güçlüdür çünkü böylelikle, basit ve zaman kritik uygulamaları donanım düzeyinde ve karmaşık algoritmaları yazılım düzeyinde çözümler.

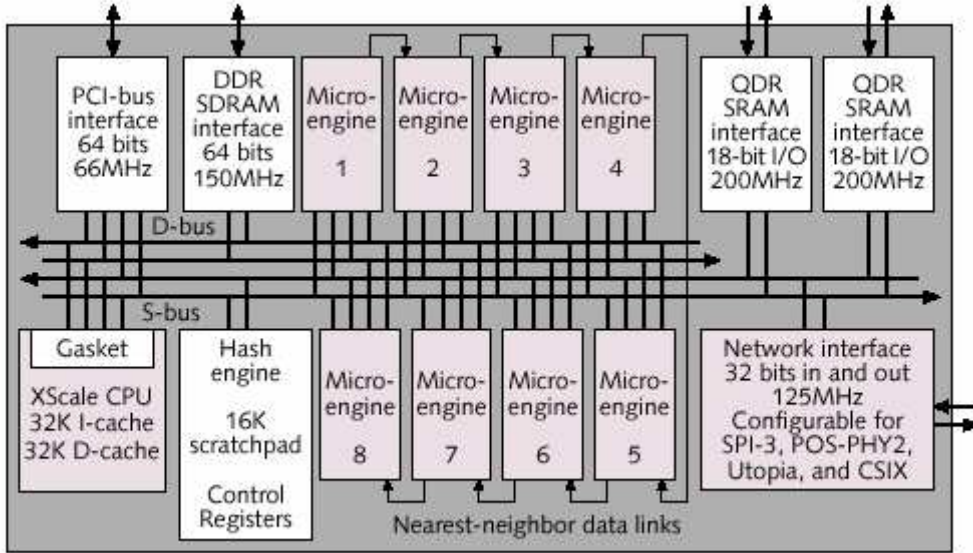
3.7 Bir Ağ İşlemcisini Programlamak

Ağ işlemcileri genel amaçlı işlemcilerden çok farklı oldukları için, programcılar için en önemli soru, bu işlemcilerin nasıl programlanacağıdır. Çok sayıda RISC çekirdek ve donanım hızlandırma birimleri nasıl daha etkin kullanılabilir? Birçok yönden, ağ işlemci mimarileri 10

yıl öncesinin paralel işlem mimarileri olarak gözükmetedir. Programcılar paralel işlem mimarilerinin koşma gücünü uzun süre önce denemiş fakat başarılı olamamıştır. Vektör işlemci süper bilgisayarlar hava sümülasyonu gibi özel amaçlı uygulamalarda kullanılmış fakat programcılar bu bilgisayarları genel amaçlı uygulamalarda bile başarılı bir şekilde kullanamamıştır.

Ağ işlemcilerin daha iyi olmasını düşündürecek bir neden var mı? Elbetteki var. Ağ işlemciler genel amaçlı işlemleri hızlandırmak için uğraşmamaktadır. Ağ işlemleri, genel amaçlı işlemlerden çok daha farklı belli özellikler içerir. Ağ işlemcileri genel amaçlı işlemciler göre daha az kod içerir. Veriler arasında daha az bağımlılık vardır. Örneğin yönlendiriciyi ele alacak olursak, eğer bir yönlendirici n paket alırsa(n küçük bir sayı), bu paketleri birbirinden bağımsız olarak işleyebilir. Bunu ifade etmenin bir başka yolu da, paketlerin yönlendiricilerin durumlarını değiştirmeyeceğidir. Bunun için istisna ayarlama paketleri, ya da yönlendirme protokol paketleri olabilir. Ancak, bu bağımlılıklarda bile gevşek davranılabilir. Eğer yönlendirme tablolarını güncellemesi gereken bir paket alırsa, birkaç paketi daha işlemeyen önce tabloyu güncellemesi için geçerli bir sebep yoktur.

4. Intel IXP Ailesi



- Core & Edge-IXP2800, Edge & Access-IXP2400, CPE-IXP425
- Mikro makine teknolojisi
 - Sakla ve ilet mimarisi
 - Yüksek paralel tasarım
 - Yüksek hızda veri işleme
- Xscale teknolojisi
 - Kontrol düzeyinde entegre uygulama işleme
 - Veri yapılarını güncelleme ve yönetme
 - Dış ortam ve anahtarlarına aygıtlarını kurma ve denetleme
 - Süper iş hattı teknolojisi sayesinde, çok aşamalı yüksek etkinlikte iş hattı sürme mimarisi
- IXA(Internet Exchange Architecture) yazılım taşınabilir tasarımı
 - Kolay kod geliştirme ve yeniden kod kullanımı
 - Modüler programlanabilir model: mikro makineler ve iplikler arasında en uygun uygulama bölmelenmesi

- Optimum mikro makine kütüphaneleri ve araçları
- Intel Xscale kaynak kod kütüphaneleri: birçok işlem ortamı arasında taşınılılırlık

Memory Element	Type	Capacity	Access Time (cycles)	Context Switch
Data Mem.	Local	2560 B	3	No
Scratchpad	Shared	16 KB	60	Yes
SRAM	Shared	≤ 256 MB	90	Yes
DRAM	Shared	≤ 2 GB	120	Yes

Table 1: Intel IXP series memory capacities and access times

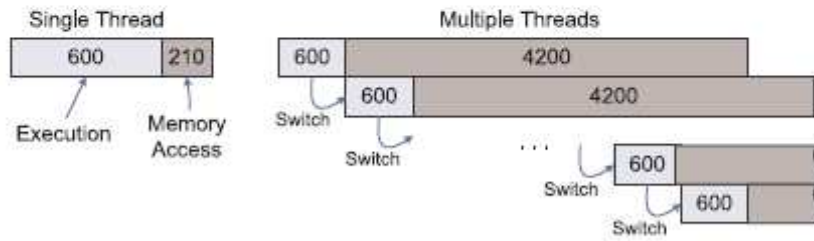
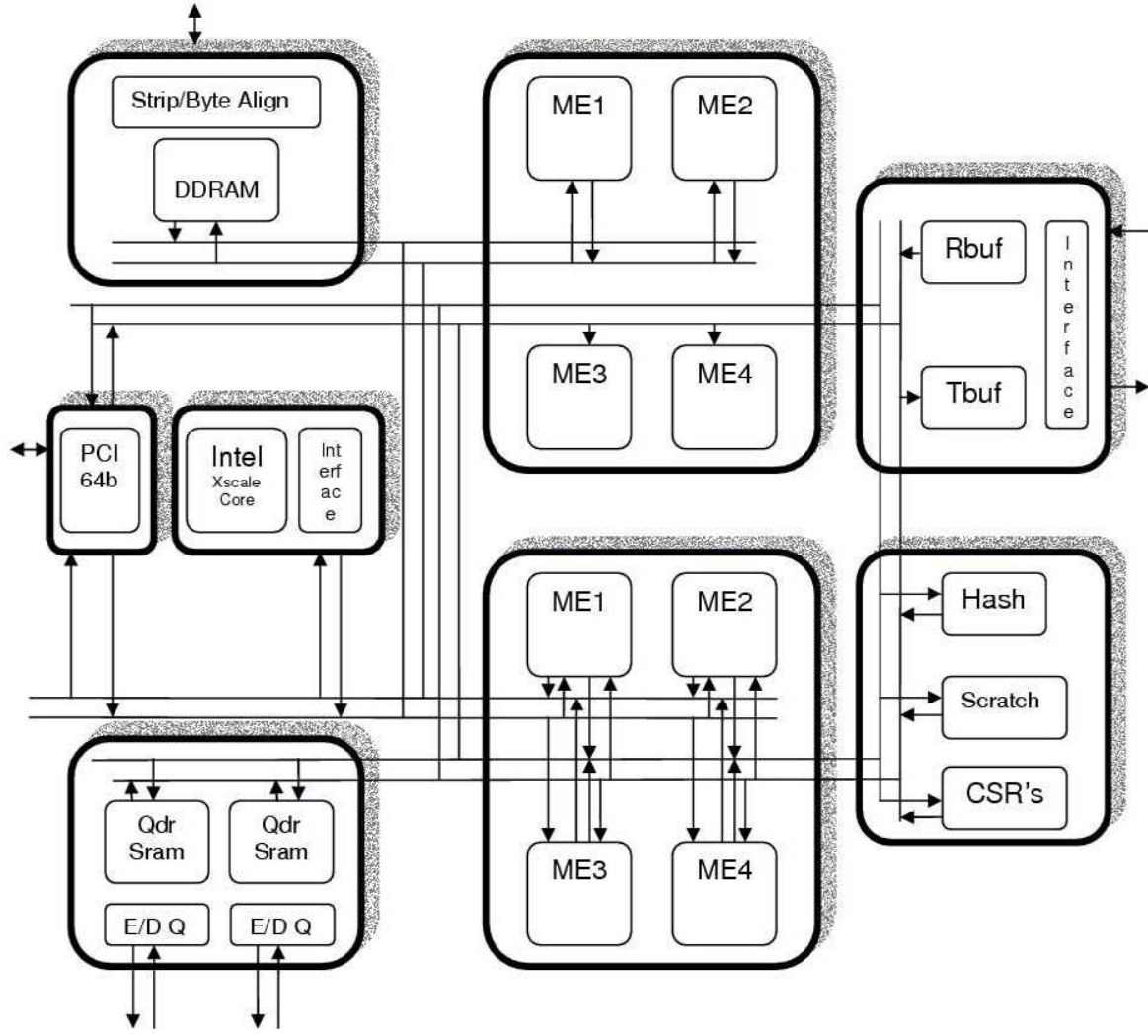


Figure 1: Optimization by multi-threading

4.1 Intel IXP2400 Ağ İşlemci Mimarisi

Intel IXP2400 işlemci CMP tabanlı çok iplikli mimariler sınıfına aittir. Bu mimari, 32 bitlik bağımsız mikro makineler ile bir Intel Xscale işlemci çekirdeğini birleştirerek, iki kümede gruplanmıştır.

Mikro makineler Intel'in ikinci nesil çok iplikli RISC işlemcileri tabanlıdır. Mikro makinelerdeki her iplik kendi saklayıcısı kümesini, program sayacını ve özel yerel saklayıcıların denetleyicisini içerir. Hızlı içerik yer değiştirme sayesinde, bir başka içerik hesaplama yaparken, ilk içerik I/O işlemleri için bekler. Her iplik 4 durumdan birinde bulunabilir: i) İnaktif: eğer uygulamalar tüm iplikleri kullanmak istemezse, ii) Hazır: iplik yürütmeye hazırdır, iii) Yürüt: Bu yürütme durumudur; bir iplik ancak bir komut bu izleğin durumunu değiştirene kadar ya da bir içerik yer değiştirme olduğunda bu durumda kalmayı sürdürür ve iv) Uyku: iplik dışarıdan gelen olayların gerçekleşmesini bekler. Sadece bir içerik, belli bir zamanda yürütülme durumunda kalır. Her mikro makine 640 tane 32 bit kelimelik yerel bellek içerir. Bu bellek mikro makineler üzerindeki ipliklere eşit olarak bölüştürülür. Her mikro makine 256 tane 32 bitlik genel amaçlı saklayıcı içerir ve bunlar program kontrolü altında yürütme yolundaki veriye işlemleri uygulamakta kullanılır.

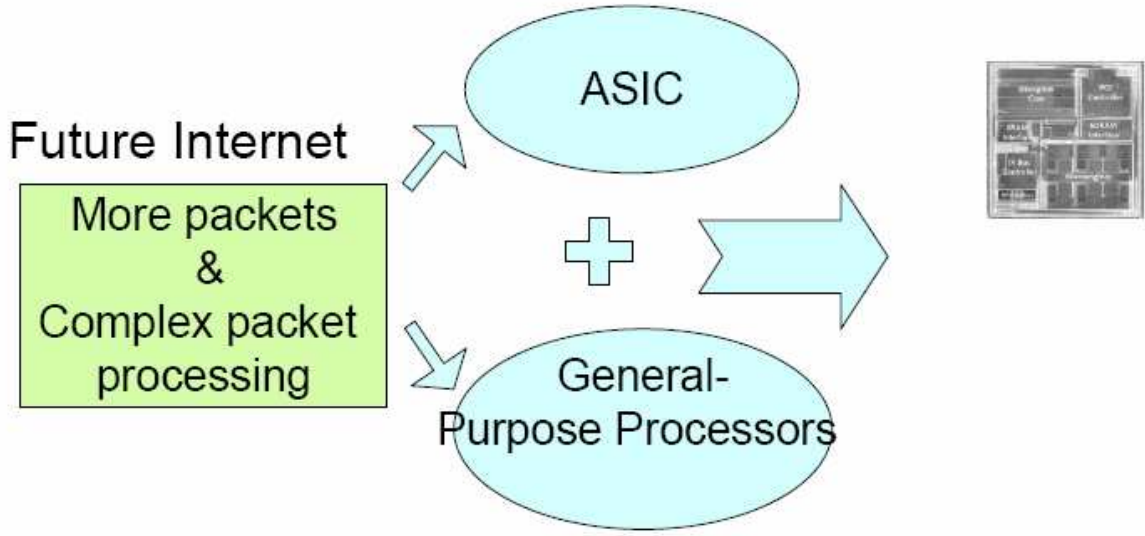


Her mikro makine 3 çeşit dış belleğe erişir: yazboz bellek, SRAM ve DRAM. Yazboz bellek yongayla tümleşik bellek, SRAM ve DRAM yongadan ayrı belleklerdir. IXP2400, 16KB'lık yazboz belleklere tüm mikro makinelerin erişebileceği bir ortak hızlı erişim belleğidir. IXP2400QDR SRAM'ın iki kanalını sağlar. SRAM denetleyici kuyruk, bağlantılı liste ve FIFO'ların uygulaması için kullanılan halka işlemleri için destek donanımına sahiptir. Aynı zamanda, atomik bit bazlı işler gibi işlemlerin senkronizasyonunu sağlar. IXP2400 2 GB'a kadar tek kanallı DRAM'i destekler. DRAM hızlı bir ara yüze sahip olduğu için, gelen paketleri tamponlamakta kullanılır. Bu ara yüz, paketleri dış ortamdan alan ve dış ortama ileten dış ortam anahtar sistemlerden oluşur.

5. Yeni Nesil Ağ İşlemciler

Yeni nesil ağ işlemciler:

- Yüksek işlem gücü
- Yüksek iletim hızı
- Programlanabilir
- Ölçeklenebilir
- Ağ uygulamaları için en iyi şekilde kullanılabilir olmalıdır.



6. Sonuç

Sonuç olarak, ağ uygulamalarında paralellik hem ağ protokollerinden hem de paketlerden türetilir. Ancak, bazı ağ uygulamaları paketlerin paralel şekilde işlenmesine engel olacak paket bağımlılıklarına sahip olabilir. SMT ve CMP modele, çok iplikli yapı entegre edilerek, ağ uygulamalarında paralellik sağlanabilir. Etkin paket bağımlılığına getirilen çözümler, paralel olarak paket işleyebilmek için ağ uygulamalarında paket bağımlılığı içeren paketlere uygulanmalıdır. Geleneksel ağ uygulamaları esasında, veri ve dallanma yükü ağır uygulamalardır. Ancak yeni nesil ağ uygulamalarının, kesinlikle hesapsal yükü de ağır olacaktır. Ağ uygulamalarının tüm bu özellikleri geleceğin ağ işlemcilerinin tasarlanması aşamasında kesinlikle göz önünde bulundurulmalıdır.

Mimari açıdan bakılacak olunursa, SMT ve CMP yeni nesil işlemcilerin temel mikro mimarileri olacaktır. SMT'nin kaynak kullanımının CMP'ye göre daha etkin olması sebebiyle SMT'de üstün performans sergiler. Öte yandan, CMP içinde benzer çalışma alanları için, SMT'den daha az yer kaplayacak şekilde birden fazla tek-iplikli işlemci çekirdeği kullanılabilir. Eğer uygulamalar etkin bir şekilde çok sayıda alt ipliğe bölünebiliyorsa, CMP SMT'ye göre daha üstün performans sergileyebilir. Yeni nesil veri yükü ağır uygulamaların yönlendirme tablosu cep tasarımı ve paket tampon tasarımı üzerinde daha fazla çalışma gereklidir.

Yeni nesil ağ işlemcileri, tek bir yongadan oluşan, daha etkin yönlendirme tablosu cepleri, paket tamponları ve paket bağımlılığına çözüm getiren, SMT işlemci çekirdeklerinden oluşan SMT mikro işlemciler olacaktır. Geleceğin ağ uygulamalarında veri yükü ve hesapsal yük fazla, dallanma oranı yüksek olacağından, SMT temel mikro mimari olarak seçilmelidir.

7. Açıklamalar

Anlaşılması açısından, raporda yapılan çeviri ve kabuller şu şekildedir:

- Thread → İplik
- Switch → Anahtar
- Router → Yönlendirici
- TLB(Translation Lookaside Buffer) → Adres dönüşüm tamponu
- CMP(Chip MultiProcessing)→ Tek yongada çok işlemci
- SMT(Simultaneous MultiThreading) →Eşzamanlı çok iplikli
- ASIC(Application Specific Integrated Circuit) → Uygulamaya özel tümleşik devre
- RISC(Reduced Instruction Set Computer) → Azaltılmış komut kümeli bilgisayar
- TLP(Thread-Level Parallelism) → İplik düzeyinde paralellik
- ILP(Instruction-Level Parallelism) → Komut düzeyinde paralellik
- Branch/Data intensive → Dallar/Veri yoğun
- Deep packet classification → Derin paket sınıflandırma
- Content aware → İçerik bilinçli

8. Referanslar

- [1] TKS LakshmiPriya, V.H. Prasad, D.Kannan, L.K. Singaram, G. Madhan, R.M. Sundaram, R.M. Prasad and R. Parthasarathi: Evaluating the Network Processor Architecture for Application-Awareness. By IEEE, 2007.
- [2] C. Ostler, K. S. Chatha and G. Konjevod: Approximation Algorithm for Process Mapping on Network Processor Architectures. By IEEE, 2007.
- [3] K. Yi and J. Gaudiot: Features of Future Network Processor Architectures, By IEEE, 2006.
- [4] V. Ramamurthi, J. McCollum, C. Ostler and K. S. Chatha: System Level Methodology for Programming CMP based Multi-threaded Network Processor Architectures. By IEEE, 2005.
- [5] L. Thiele, S. Chakraborty, M. Gries and S. Künzli: Design Space Exploration of Network Processor Architectures. By IEEE, 2002.
- [6] P. Crowley, M. E. Fiuczynski, J. Baer and B. N. Bershad: Characterizing Processor Architectures for Programmable Network Interfaces. By 2000 International Conference on Supercomputing, May, 2000.
- [7] X. Nie, L. Gazsi, F. Engel and G. Fettweis: A New Network Processor Architecture for High-speed Communications. By IEEE, 1999.
- [8] Y. Jun: Network Processors
<http://www.ecs.umass.edu/ece/wolf/courses/ECE697J/Fall2002/presentations/ECE697J-02-11-12.pdf>
- [9] M. Kohler: Network Processor Overview
<http://www.netrino.com/Articles/NetworkProcessors/index.php>
- [10] Y. Lou: Network Processor Architecture and Applications
<http://faculty.uml.edu/yluo/Publications/CAR-seminar.pdf>