YONGA-ÜSTÜ-AĞLAR İÇİN HATALARA DAYANIKLI UYARLANABİLEN YÖNLENDİRME ALGORİTMASI TASARIMI

FAULT-TOLERANT ADAPTIVE ROUTING ALGORITHM DESIGN FOR NETWORK-ON-CHIPS

ANIL İPEK

DOÇ. DR. SÜLEYMAN TOSUN

Tez Danışmanı

Hacettepe Üniversitesi

Lisansüstü Eğitim – Öğretim ve Sınav Yönetmeliğinin

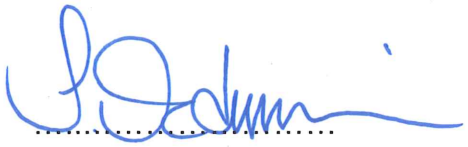
Bilgisayar Mühendisliği Anabilim Dalı İçin Öngördüğü

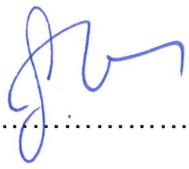
YÜKSEK LİSANS TEZİ

olarak hazırlanmıştır.

2019

Anıl İPEK' in hazırladığı **"Yonga-üstü-Ağlar için Hatalara Dayanıklı Uyarlanabilen Yönlendirme Algoritması Tasarımı"** adlı bu çalışma aşağıdaki jüri tarafından **BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI'** nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Prof. Dr. Suat Özdemir
Başkan ..... 

Doç. Dr. Süleyman Tosun
Danışman ..... 

Doç. Dr. Sevil Şen
Üye ..... 

Doç. Dr. Ahmet Burak Can
Üye ..... 

Dr. Öğr. Üyesi Murat Aydos
Üye ..... 

Bu tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafından **YÜKSEK LİSANS TEZİ** olarak ... / ... / ..... tarihinde onaylanmıştır.

Prof. Dr. Menemşe GÜMÜŞDERELİOĞLU
Fen Bilimleri Enstitüsü Müdürü

ETİK

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

11/02/2019

İmza

ANIL İPEK

YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezimin/raporumun tamamını veya herhangi bir kısmını, basılı (kâğıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma iznini Hacettepe Üniversitesi'ne verdiğimi bildiririm. Bu izinle üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet hakları bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları bana ait olacaktır.

Tezin kendi orijinal çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim Kurulu tarafından yayınlanan "***Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge***" kapsamında tezim aşağıda belirtilen koşullar haricince YÖK Ulusal Tez Merkezi / H. Ü. Kütüphaneleri Açık Erişim Sisteminde erişime açılır.

- ☐ Enstitü / Fakülte yönetim kurulu kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren 2 yıl ertelenmiştir.

- ☐ Enstitü / Fakülte yönetim kurulu gerekçeli kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren .... ay ertelenmiştir.

- ☐ Tezim ile ilgili gizlilik kararı verilmiştir.

11 / 02 / 2019

(İmza)

ANIL İPEK

ÖZET

YONGA-ÜSTÜ-AĞLAR İÇİN HATALARA DAYANIKLI UYARLANABİLEN YÖNLENDİRME ALGORİTMASI TASARIMI

Anıl İPEK

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Danışmanı: Doç. Dr. Süleyman TOSUN
Şubat 2019, 58 sayfa

Entegre devre sistemlerinde; teknoloji ilerledikçe çekirdek sayısı artmakta, transistor boyutları gittikçe küçülmektedir. Bu durum, üretilen söz konusu yapıları hatalara karşı daha hassas bırakmaktadır. Başarımı ön planda tutan tasarımcılar, olası olumsuzluklarda tolerans gösterebilecek daha karmaşık algoritmalara ihtiyaç duymakta ve klasik yaklaşımlardan uzak durmaktadır. Yonga-üstü-Ağ (YüA) kavramı burada devreye girerek ölçeklenebilirlik ve gecikme süresi gibi alanlarda devreye esneklik kazandırarak üstünlük sağlamaktadır. Tez çalışması kapsamında da çok düğümlü yapılarda yaşanan tıkanıklık ve geçici / kalıcı hata gibi performansı etkileyen sorunlar ele alınarak bir yönlendirme algoritması tasarlanması hedeflenmiştir. Güçlü makine öğrenme mekanizmasıyla HARAQ, öncül yöntem olarak değerlendirilmiş ve tespit edilen sahte tıkanıklık problemi olasılıksal bir şekilde bertaraf edilerek algoritmaya hataya dayanıklılık özelliği kazandırılmıştır. Anlatılanların bir sonucu olarak, benzer akademik çalışmalarda pek rastlanmayan hem tıkanıklık farkında hem hata kaldırabilen, uyarlanabilir HAFTA yöntemi ortaya konmuştur. Deney aşamasında, yaygın trafik modelleri ile

birtakım benzetimler gerçekleştirilmiş ve tasarlanan algoritmanın başarısı hakkında, belirlenen ölçütler açısından değerlendirme yapılmıştır.

Anahtar Kelimeler: Yonga-üstü-Sistem (YüS), Yonga-üstü-Ağ (YüA), Yönlendirme Algoritması, Hataya Dayanıklılık, Tıkanıklık Farkındalığı, Uyarlanabilirlik, Sahte Tıkanıklık.

ABSTRACT

FAULT-TOLERANT ADAPTIVE ROUTING ALGORITHM DESIGN FOR NETWORK-ON-CHIPS

Anıl İPEK

Master of Science, Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. Süleyman TOSUN
February 2019, 58 pages

As the technology advances in integrated circuit systems, the number of cores increases and dimensions of transistors are getting smaller. This leaves the relevant structures more susceptible to errors. Designers, who prioritize performance, avoid classical approaches and need more complicated algorithms that can tolerate possible adversities. The concept of the Network-on-Chip (NoC) comes into play, giving flexibility to the circuit in areas such as scalability and delay time. Within the scope of the thesis study, it is aimed to design a routing algorithm by tackling the problems affecting the performance such as congestion and temporary / permanent error in multi-node structures. HARAQ has been evaluated as a premise method with its powerful machine learning mechanism. The identified bogus jam issue has been eliminated in a probabilistic way and the algorithm is strengthened by fault tolerance feature. As a result of the described, congestion-aware and error-tolerant adaptive HAFTA method has been introduced, which is uncommon in similar academic studies. In the experimental phase, some simulations were carried out with widespread traffic models and the

success of the designed algorithm was evaluated in terms of the determined criteria.

Keywords: System-on-Chip (SoC), Network-on-Chip (NoC), Routing Algorithm, Fault Tolerance, Congestion Awareness, Adaptivity, Dummy Congestion.

TEŞEKKÜR

Lisansüstü eğitimim süresince bilgisi ve deneyimiyle çalışmalarıma yön veren, içten yaklaşımı ve esirgemediği desteğiyle çekindiğim anlarda ve konularda yanımda olduğunu sonuna kadar hissettiren, en önemlisi de aşıladığı güven ve eşsiz sabrıyla takdirimi kazanan değerli hocam Sayın Doç. Dr. Süleyman TOSUN'a,

Bu tezin tamamlanmasında 117E130 numaralı proje kapsamında maddi destek sağlayan TÜBİTAK'a,

Tanıdığım andan bugüne her koşulda bana arka çıkan ve beni hoş gören, yaşantısındaki birçok alanda fedakârlık gösteren ve hakkını tam olarak ödeyemeyeceğim, irili ufaklı bütün zorlukları beraber aştığım sevgili eşime,

Hayatım boyunca, bugünlere gelmem için hayal kuran ve gücü yettiğince katkılarını gösteren, yüreklendirici nutuklarıyla beni motive eden babama,

Sonsuz Teşekkürler…

Anıl İPEK
Şubat 2019, Ankara

İÇİNDEKİLER

ŞEKİLLER

SİMGELER VE KISALTMALAR

Simgeler

β	Sahte Tıkanıklık Olasılığı Parametresi
μ	Hata Olasılığı Parametresi
μG	Geçici μ Parametresi
μK	Kalıcı μ Parametresi

Kısaltmalar

2B	İki Boyutlu
CATRA	Congestion Aware Trapezoid-based Routing Algorithm
DBAR	Destination-Based Adaptive Routing
DBP	Default Backup Path
ECC	Error-Correcting Codes
EDXY	Enhanced Dynamic XY
FRA	Fuzzy-based Routing Algorithm
HAFTA	Highly Adaptive Fault-Tolerant Algorithm
HARAQ	Highly Adaptive Routing Algorithm using Q-learning
HT	Hata Toleransı
IP	Intellectual Property
MiCoF	Minimal-path Connection-retaining Fault-tolerant approach
NoP	Node-on-Path
PARS	Path-Aware Routing Scheme
YüA	Yonga-üstü-Ağ
YüS	Yonga-üstü-Sistem

1. GİRİŞ

Entegre üretiminde kullanılan devre elemanlarının boyutlarının Moore'un transistor yasasına paralel olarak her birkaç yılda bir küçülmesi sonucunda, milyonlarca transistor tek bir yonga üstüne dizilebilmektedir [1]. Bunun doğal bir sonucu olarak, daha karmaşık ve yoğun tasarımlar tek bir yonga üstünde gerçekleştirilebilmektedir. Tüm sistem bileşenlerinin tek bir yonga üstüne gerçekleşmesiyle oluşan tasarım, Yonga-üstü-Sistem (YüS) olarak adlandırılmaktadır. Yalnız, bu kadar çok sistem bileşeninin yongalar üstüne gerçekleşmesi neticesinde, klasik veri yolu tabanlı ve noktadan noktaya tabanlı haberleşme yöntemleri, düşük performans ve eşleme problemleri nedeniyle kullanışsız hale gelmekte ve bu tip geleneksel topolojilere gittikçe rağbet azalmaktadır. Özellikle de geleceğin mimarilerinin çok sayıda yüksek kapasiteli işlemcilerden oluşacağı ve işlemciler arası haberleşmenin, veri aktarımının şimdikine nazaran oldukça fazla olacağı düşünüldüğünde, daha verimli bir haberleşme yönteminin yongalar üstünde gerçekleştirilmesi kaçınılmaz olmuştur. Bundan dolayı, bu yüzyılın başında Yonga-üstü-Ağ (YüA) adı verilen yeni bir haberleşme yöntemi sunulmuştur [2, 3]. Yonga-üstü-Ağlar, etkin yapısıyla esneklik ve yeniden kullanılabilirlik gibi sorunları ortaya koymakta ve çözmektedir [4].

Yonga-üstü-Ağ haberleşmesinde bulunan bileşenler, klasik ağ teknolojilerini kullanarak yönlendirici düğümler vasıtasıyla paketler göndererek iletişimi sürdürürler. Böylece, veri yolu tabanlı haberleşmedeki veri yolu bir işlemci tarafından kullanılırken diğer işlemcinin beklemesi problemi ortadan kaldırılmış olur ve tüm işlemciler, hedef bileşene aynı anda verilerini gönderebilirler. Ancak alt mikron yarı iletkenler gibi yüksek frekanslarda çalışan tümleşik devreler kusurlara açıktır [5]. Dolayısıyla YüA'lar, bu tip hatalarla da ilgilenmelidir. Bu hatalar ise ikiye ayrılır: geçici ve kalıcı. Geçici olanları, bir başka deyişle yumuşak hatalar, voltaj dalgalanmaları gibi anlaşılmaz nedenlere sahiptir ve bulmak zordur. Kalıcı hatalar, yıpranma kusurları veya üretim kusurları ile ilgilidir. Çoğunlukla algoritmalar bu tarz katı hatalara odaklanır ancak topolojide oluşabilecek bir veya birden fazla geçici / kalıcı hata, standart yönlendirme

algoritmasını, paketlerin hedeflerine ulaşmasını garanti edebilme adına yetersiz bırakabilir. Bundan ötürü, mimari normal olarak çalışırken her ikisi de bertaraf edilmelidir.

Hata toleransı (HT) teknikleri ayrıca ikiye ayrılır: belirlenimci ve uyarlanabilir [6]. Birincisi verimsizdir çünkü belirlenimci yaklaşımlar, kısır yol alternatifleri nedeniyle daha fazla gecikme yaratır ve buna bağlı olarak yükü dengeleyemezler. İkinci yaklaşım ise tıkanıklık vakasına karşı daha dayanıklıdır ve çeşitli rota seçeneklerine sahip olduğundan kusur dayanıklılık özelliğini iyileştirir.

HT ağları normalde, bir hatayla karşılaşana kadar paketleri iletir. Arıza anında, kusurun etrafında yeniden yönlendirme meydana gelir ve yaklaşım, en kısa olmayan yolu seçebilir. Bu, ağırlıklı olarak algoritmalar için bazı performans bozulmalarına neden olur. Bu nedenle, yol seçimi belirsizliğini ortadan kaldırmak ve optimize etmek için akıllı mekanizmalar seçilmelidir. Takviyeli öğrenme bunlardan biridir. İyi sonuçlandırılmış eylemleri teşvik etme ilkesinden kaynaklanır [7]. Bu ailenin bir üyesi, çıkış port seçimini temsil eden ve makine öğrenmesi perspektifinden yükleri dengeleyen Q-Yönlendirme yöntemidir. Anahtar başına gecikme kestirim tablolarından oluşur ve bu şekilde, seçme kısmı kolaylaşır. Bu yaklaşım sadece dağıtım süresini azaltmakla kalmaz, aynı zamanda ağ üzerindeki sıkışmaları gidererek arızaları bertaraf etmeyi de destekler.

Bu çalışmada, bahsedilenler ışığında sunulan çözüm; Yüksek Uyarlamalı Hatalara Dayanıklı Algoritma (Highly Adaptive Fault-Tolerant Algorithm) olarak adlandırılan ve tamamen uyarlanabilir bir yönlendirme algoritmasıdır. Ayrıca yaklaşım, kilit problemine yol açmamasının yanı sıra mükemmel bir adaptasyonla sonuçlanan, geniş bir izin verilen dönüş grubu ile oldukça esnek bir yapı sunar. Q-Öğrenme yöntemini kullanmaya ek olarak, iki büyük katkıya odaklanılmıştır: verimli bir hata toleransı mekanizmasını eklemlemek ve "sahte tıkanıklık" sorununu tanımlamak / gidermek.

Tezin tamamı 6 bölüm olarak düzenlenmiştir. İkinci bölüm ilgili çalışmaları ele almaktadır. Üçüncü bölümde, çalışmada kullanılan temel kavramlara ve mimari modele değinilip kısaca probleme yer verilmiştir. Bölüm 4'te geliştirilen yönlendirme algoritması ve barındırdığı ilkeler ayrıntılı olarak açıklanmıştır. Bölüm 5, değerlendirme aşamasını kullanılan ortamla birlikte açıklarken son kısımda, varılan sonuç belirtilmiş ve tartışması yapılmıştır.

2. KAYNAK ÖZETLERİ

Son yıllarda artan işlemci gücü ihtiyacı, çok işlemcili sistemleri kaçınılmaz kılmıştır. Bu çok işlemcili sistemler de genellikle ağ haberleşmesi yöntemiyle haberleşmektedir. Ancak çok işlemcili sistemlerin fazla güç tüketimi sonucu, yonganın süreç içerisinde maruz kaldığı termal etkiler ve transistor boyutlarının küçülmesi neticesi artış gösteren fabrikasyon kaynaklı kalıcı hatalar, yonganın hatasız çalışmasını riske atmaktadır. Ayrıca, radyasyon etkilerine daha duyarlı olmalarından kaynaklanan geçici hatalar da yongaların doğru çalışmasını tehdit eder olmuştur. Bu nedenlerle son yıllarda, hataya dayanıklı YüA mimarileri ve uyarlamalı yönlendirme algoritmaları gibi konular, araştırmacıların ilgisini çekmiştir. Haliyle ilgili çalışmalar iki dalda irdelenebilir: tıkanıklığa duyarlı ve hataya dayanıklı. Görülüyor ki araştırmaların birçoğu, hataya müsamaha göstermenin yüksek maliyeti ve karmaşıklığı ya da sıkışıklıkları atlatmaya ilişkin uyarlamalı bilgi toplama / yorumlama zorluğu nedeniyle her iki amacı da hedeflememektedir.

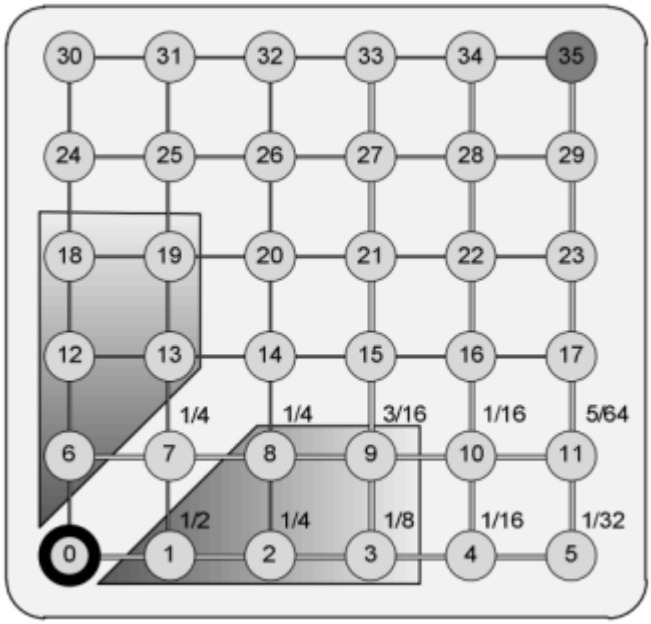
2.1. Tıkanıklığa Duyarlı Algoritmalar

FRA [8] ve DBAR [9] gibi algoritmalar, enerji verimliliği ve sembolik gecikme değerleri sağlamak için yalnızca en kısa yolları dikkate almaktadır. Bununla birlikte, rota belirlemedeki bu kısıtlamaları nedeniyle düşük uyarlamalıdır ve yeterince yük dağıtamazlar.

Öte yandan, ağın trafik bilgisi, tıkanıklığa aykırı yaklaşımlarda iyice ortaya çıkan tıkanıklık bilincinde çok önemlidir [10]. Bazı algoritmalar yalnızca yerel verileri dikkate alırken, diğerleri CATRA [11], PARS [12] ve NoP [13] gibi bölgesel kavramları kullanır. Her ne kadar bölgesel olanlar kullanışlı bilgiler açısından en iyisi gibi görünse de, hem performansı küresel tarzın altındadır hem de ölçeklenebilirliği kördüğüm bir konudur.

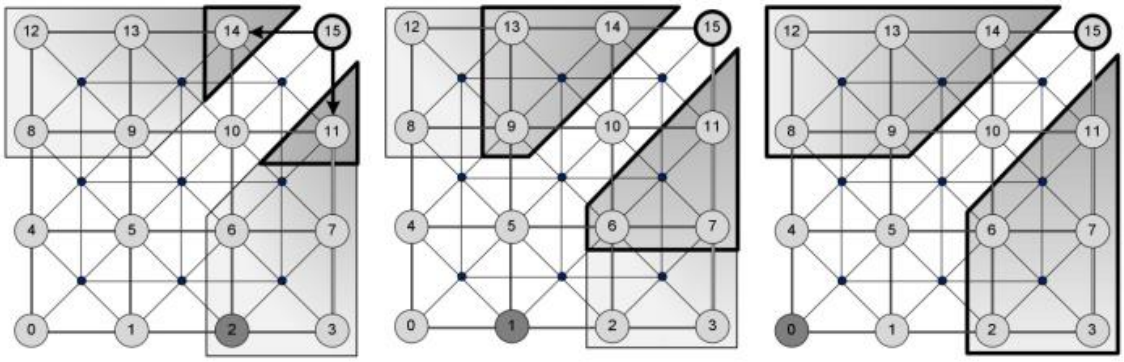
CATRA algoritmasının detayına inildiğinde ise, olasılık teorisine dayanan bir yaklaşımla karşılaşılıyor. Düğüm, rotada var olma ihtimali hesaplanıp yalnızca en

kısa yol üzerinde bir alternatifse ve eskimeyecek bir veri paylaşabilme yakınlığındaysa dikkate alınıyor. İşte burada, yamuk tabanlı bir yöntem ele alınmış.



Şekil 2.1. CATRA yönteminde paketlerin orta düğümlerden geçme olasılığı

Şekil 2.1'de de görüldüğü üzere, 0'dan 35'e giderken yeğlenebilecek iki en kısa yol üzerindeki yamuklar ele alınıyor. 7 ve 14 ortak düğüm olduğu için, kıyas hesabına katılmıyor.



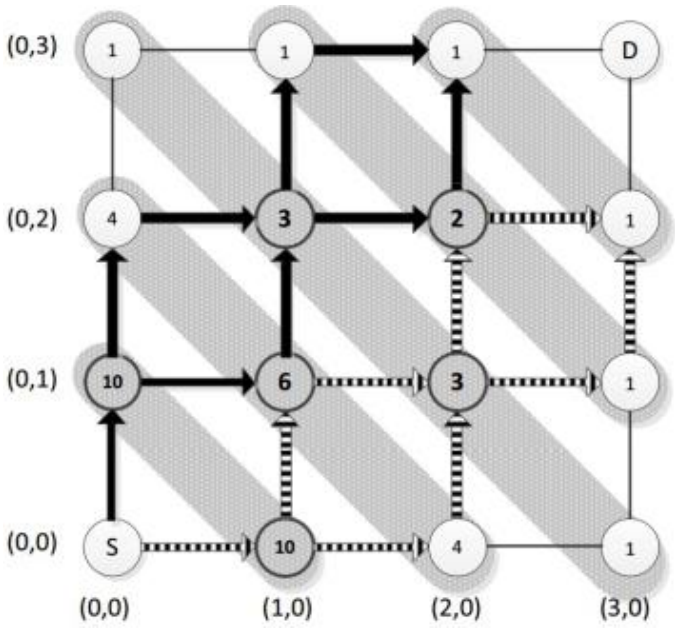
Şekil 2.2. CATRA algoritmasında sıkışıklık hesaplama örnekleri

Algoritma, kaynağa göre hedefin konumunu temel alarak sıkışıklık kıyasını daraltıp genişletme esnekliğine de sahip. Özetlersek, hedefle kaynak arasındaki

patikayı uzatacak bir yamuk alanı oluşuyorsa kırpılıyor. Mesela Şekil 2.2'de, hedef 1 no'lu düğüm olduğunda 2, 3, 8 ve 12 dikkate alınmıyor.

PARS algoritmasının detayında ise, "uyarlanabilirlik" adı altında bir metrik bulunmakta. Düğümlerin sıcak (yoğun trafik anları) veya hatalı olduğu durumlardan kaçınmak ve devrenin uyarlanabilirlik kabiliyetini pekiştirmek için ortaya atılan bu parametre, ilgili düğümün hedef koordinata olan uzaklığına göre hesaplanıyor.

Bu sistemin örgü topolojideki her düğüm için uygulanmış bir örneği, Şekil 2.3 üzerinden görülebilir. Burada, düğümlerin hedefe olan sekme uzaklığına göre gölge şeritlerle sınıflara ayrıldığı ve genellikle kuşbakışı yörünge üzerinde kalan ara düğümlerin uyarlanabilirlik değerinin kenar düğümlerden daha yüksek olduğu yorumu yapılabilir.



Şekil 2.3. Uyarlanabilirlik değerleriyle birlikte bir örgü topoloji

Ayrıca, yonga üzerindeki ağlara orijinal yaklaşımlar getiren bazı algoritmalar da mevcuttur. Örneğin, DBP [14] algoritmasında bulunan yedekleme yolları veya EDXY [15] algoritmasında bulunan ayrı kablolar gibi kimi çeşitli şemalar ekstra bileşenlere sahiptir ancak bu durum, harcamaları olumsuz yönde etkilemektedir. Her anahtarın ağ verilerini kabul edilebilir boyutta bir bellekte muhafaza ettiği ve

bu verileri de kendi yöntemiyle güncellediği tablo tabanlı sistemler de vardır [16]. Buradaki en kritik nokta, tabloyu olabildiğince küçültmek ve ağı taşırmayacak veya veri tazeliğini sekteye uğratmayacak en iyi şekilde geri bildirimleri dengelemektir.

2.2. Hataya Dayanıklı Algoritmalar

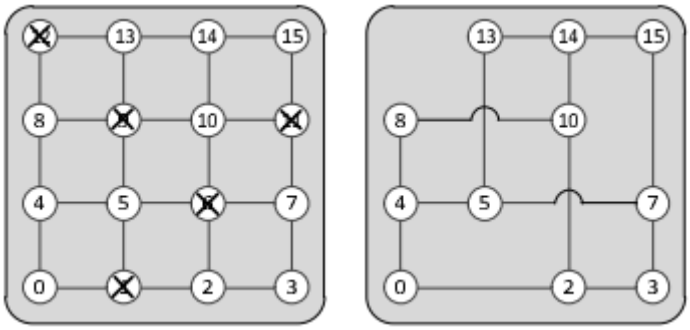
Literatürde daha önce yayımlanan çalışmalardan bazıları, hataların yonga üstünde düzeltilmesi yöntemini seçerken bazıları ise, yonganın hatalı bölümünü kullanmamak adına verilerin yönlendirme yollarını değiştirme seçeneğini tercih etmişlerdir. İlk yaklaşım, genellikle belleklerdeki hataların düzeltilmesinde kullanılmış ve bunun için maliyeti yüksek olan hata düzeltme kodlarından (ECC) faydalanılmıştır. İkinci yaklaşımda ise, çoğunlukla Yonga-üstü-Ağ mimarilerindeki kalıcı link hataları ve yönlendirici düğümlerde oluşabilecek port hataları ele alınmıştır.

Genellikle hataya dayanıklı yaklaşımlar, minimum olmayan rotaları kullanırlar ve bu esneklikleri sayesinde hatalı yönlendiricilerin veya bağlantıların üstesinden gelirler [17, 18]. Ne yazık ki, hataya dayanıklı algoritmaların birçoğunun karmaşıklık ile ilgili problemleri vardır ve bunun yanında, yalnızca statik, kalıcı kusurlara odaklanırlar.

Mojtaba Valinataj vd. hatalı düğümlerin etrafından dolaşmak suretiyle paketleri yönlendiren bir YüA algoritması [19] öne sürmüştür. Ne yazık ki mekanizma, çok fazla komşu yönlendiriciyi bilgilendirme zorunluluğundan dolayı sıkıntı çekmektedir. Buna bir alternatif getiren MiCoF algoritmasında da iddia edildiği üzere [20], yeniden yönlendirme bu tür yaklaşımlar için yeni sıcak noktalar oluşturur.

MiCoF algoritması, bir yönlendirici mantığının hatalı çalışması durumunda paketi ters yöne basitçe ileten bir yönlendirici mimarisi kullanarak 2 boyutlu (2B) örgü ağ topolojileri için YüA yapısını muhafaza etme fikrini ele alır. Arızalı bir yönlendirici, daha sonra basit bir şekilde gelen paketleri ileten uzun bir tel olarak kabul edilebilir. Bu şekilde; hatalı yönlendiricilerin etrafında hiçbir paket yönlendirilmesi

gerekmez, yönlendirme algoritması basitleşir ve her bir kaynak-hedef çifti arasında en kısa yolun seçilmesine izin verilerek orijinal performans seviyesinin korunması hedeflenir. Uyarlanabilir bir yönlendirme algoritması olan MiCoF, en uygun rotayı bulmak için yalnızca bitişik yönlendiricilerin hata durumuna ihtiyaç duyar.



Şekil 2.4. Yönlendiricilerde hata olduğunda MiCoF yaklaşımının görünümü

Şekil 2.4'te de görüldüğü üzere kusurlu yönlendiriciler, kanala gelen paketin geldiği doğrultuyu bozmadan iletim yapabilir durumda Yonga-üstü-Ağ akışını devam ettirebiliyor.

Hata kaldırabilmeyi ele alan bir diğer çalışma da Jie Wu tarafından ortaya atılmıştır [21]. Yalnız buradaki dezavantaj, daha önceden hata veya hataların tespit edilmiş olması gerektiğidir. Yani kullanılamayacak yönlendirici düğümler, önceden bilinmelidir.

David Fick vd. de yayımladıkları bir makalede [22], hatalara dayanıklılık özelliği üzerinde durmuşlardır. Bu çalışma da birçok HT algoritmasına benzer bir şekilde, yönlendirici düğümlere hatanın varlığını tespit eden bir mekanizma ekleyerek paketleri, yeni rotalarına yönlendirme işlemi yapmaktadır. Literatürde görülen birtakım diğer çalışmalarsa sadece örgü tipindeki topolojilerde, link hatalarına karşı iyi sonuçlar veren uyarlanabilen yönlendirme algoritmaları üzerinde durmuştur [23].

Bu tezin amacı, çok sayıda akademik çalışmayı bir araya getirerek düşük ek yük ile tıkanıklık farkındalığını ve hata toleransını bir arada kapsayan, en az

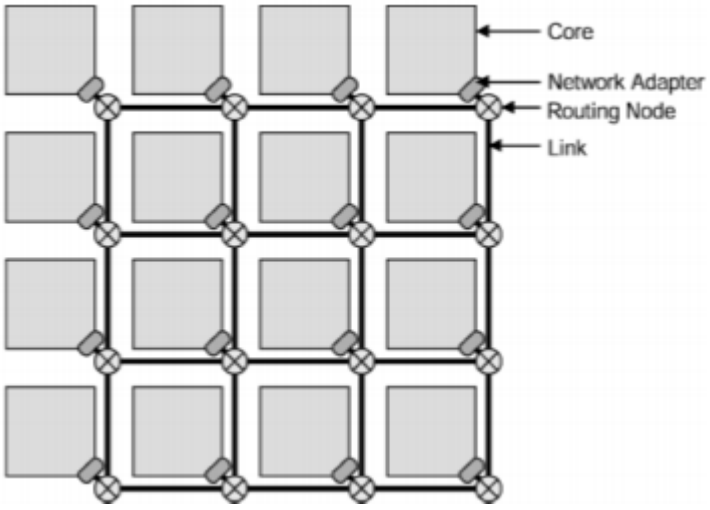
kısıtlanmış yaklaşımı tasarlamaktır. Bunu yapmak için, minimal olmayan yönlendirmeler ve yerel olmayan trafik bilgilerinin kullanılması gibi diğer öncül yaklaşımların bazı avantajlarını göz önünde bulunduruyoruz. Bu noktada, iyi yapılmış bir yönlendirme algoritmasının tasarlanmasını uyarlanabilirlik, karmaşıklık ve maliyet ödünleşmesini sıkı tutmakla ilişkilendirmek gerektiği söylenebilir.

3. YONGA-ÜSTÜ-AĞLAR VE YÖNLENDİRME TEMELLERİ

Bu kısımda, tez çalışmasında faydalanılan kavramların temel özelliklerine değinilmiştir.

3.1. Yonga-üstü-Ağ Mimarisi ve Bileşenleri

Yonga-üstü-Ağ yapısı üzerine gösterilen eğilimin son yıllarda artmasından ve sebeplerinden, önceki bölümlerde söz edilmişti. Bu yeni nesil yonga içi haberleşme tekniğinin detayında, bilgisayar ağlarında görülen paket anahtarlama yönteminin yattığı söylenebilir. Böylelikle, düğümler arası veri iletişiminin daha dağıtık ve alternatifli bir mekanizmaya oturtulması sağlanarak performans, ölçeklenebilirlik ve işlevsellik anlamında çeşitli avantajlar kazandırılmıştır. Mimarinin ayrıntısına hâkim olunabilmesi açısından, bu bölümde bileşenleri irdelenecektir. Bahsi geçen elemanların tümü; çekirdek (core), ağ bağdaştırıcısı (network adapter), yönlendirici düğüm (routing node) ve bağ (link) Şekil 3.1'de görülebilir.



Şekil 3.1. YüA bileşenlerinin örgü topoloji üzerinde gösterimi

3.1.1. Çekirdek (IP Blok)

IP, fikri mülkiyet olarak adlandırabileceğimiz ve mimarinin ana yapısını oluşturan blok elemandır. Ağ üzerinde gönderilen ve alınan paketler içerisindeki veriyi meydana getiren ve işleyen, asıl bileşendir. Buna örnek olarak, işlemciler verilebilir. Mimari, çekirdeklerin birbirine bağlanmasını ve haberleşmesini esas alır.

3.1.2. Ağ Bağdaştırıcısı

Çekirdeklerin YüA'ya dâhil olmasını ve bağlanmasını sağlar. IP blok süreçleri ile şebekedeki iletişimi birbirinden soyutlayarak bir ara katman vazifesi gerçekleştirir. Bu rolüyle, bilgisayar ağlarında yer alan Ethernet bağdaştırıcılarını andırmaktadır.

3.1.3. Yönlendirici Düğüm

Paketlerin bağdaştırıcılar kanalıyla dolaşıma girmesinden çekirdeklere iletilmesine kadar olan aşamada, ağ üzerindeki yönlendirmeleri yöneten elemandır. Bu çevrimde karar verme mekanizması, tasarlanan yönlendirme algoritmasına göre hareket eder.

3.1.4. Bağ (Link)

Yönlendirici düğümleri fiziksel olarak birbirine bağlayan ağ bileşenidir. Tek kanallı olabileceği gibi topolojinin hedefine göre (dayanıklılık, gecikme, güvenilirlik gibi) çok kanallı da kurgulanabilir.

3.2. Yönlendirme Temelleri

Yongalar, genel amaçlı veya uygulamaya özel, ideal iletişim başarımı amacıyla belirli bir topoloji üzerine kurulur. Ancak bu hayali gerçeğe dönüştürme anlamında yönlendirme kavramının payı oldukça büyük yer tutar. Zira kaynak göndericiden

hedef alıcıya verinin hangi güzergâhı izleyeceğini belirleyen, yönlendirme algoritmasıdır.

Bu noktada, hataya dayanıklılık gibi devreye katılabilecek birtakım kabiliyetler düşünülebilir fakat yönlendirme algoritmaları için başlıca iki denge unsuru vardır: iyi bir verimlilik (yükün topolojide mümkün olduğunca eşit bir şekilde dağıtılmasına dayanır) ve asgari gecikme süresi (çoğu zaman en kısa yola tekabül eder). Bahsedilen değerler üzerine bir fedakârlık terazisi kuran ihmalkâr (oblivious) algoritmalar var olduğu gibi trafik şekline göre kendini adapte edebilen algoritmalar da mevcuttur.

Sınıflandırma anlamında, hedef alıcı sayısı ve yönlendirme kararının alındığı yer gibi farklı kategoriler bulunsa da en sık karşılaşılan kıstas, algoritmanın uyarlanabilir olup olmadığıdır ve bu anlamda, iki tipten söz etmek mümkündür: belirlenimci ve uyarlanabilen. Geleneksel olarak adlandırabileceğimiz belirlenimci (deterministic) algoritmalar, tek bir kati yol belirleyip buna uygun yönlendirme yapılması şeklinde özetlenebilecek pratiklikte ve tekdüze trafikler için oldukça etkili gözükmektedir. Gel gelelim, olası ağ tıkanıklıklarındaysa adı üzerinde, uyarlama becerisinden yoksun. Modern yüz, uyarlanabilen algoritmalarsa birden fazla yol belirleyip kıstasa uygunu seçerek ilerlediğinden, daha esnek bir bakış açısı taşıyor. Böylece, birtakım engellere takılma durumunu olabildiğince bertaraf ediyor.

3.2.1. Belirlenimci ve İhmalkâr Algoritmalar

Bir miktar bahsedilen belirlenimci algoritmalar, kaynaktan hedefe tek yol çizip yönlendirme olarak da buna harfiyen uyan, haliyle basit ve bir o kadar da maliyeti düşük YüA yöntemlerindendir. Ancak herhangi bir ağ tıkanıklığını gözetmediği gibi karşılaşılan trafiği de monoton olmadığı müddetçe, pek de eşit bir yük dengesine oturtamamaktadır. Bunun da en yaygın bilinen örneği, alışılagelmiş boyut-sıralı-yönlendirme (dimension-order-routing) yapan e-küp algoritmasıdır. Sırasıyla boyut içi yönlendirmeleri bitiren (muhtemel asgari dönüş) ve dolayısıyla paket trafiğini merkeze yığan algoritma, yük dengeleme konusunda sınıfta kalmaktadır.

Akraba sayılabilecek ihmalkâr algoritmalarsa yük dengeleme veya yol kısaltma açısından biraz aşama kaydetmiş olsa da ağın durumunu dikkate almadan yönlendirme kararı veren yordamlardandır. Bundan dolayı da verimlilik ve gecikme anlamında, her ikisini birden en iyileştirmeyi çok fazla başarabildikleri söylenemez. Yine de en kısa yola yani hıza öncelik veren belirlenimci e-küp algoritmasına, yük dengesinde alternatif olarak çeşitli algoritmalar örnek gösterilebilir.

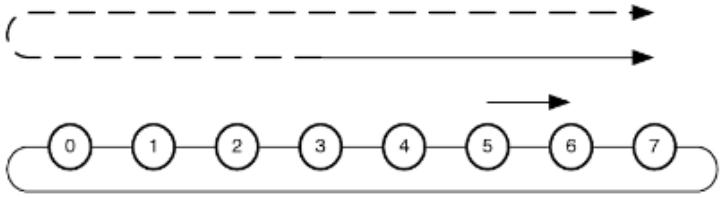
Valiant'ın rastgele algoritması, alışılagelmiş kaynaktan hedefe yönlendirmesini ikiye böler. Birinci evrede, kaynaktan sonraki öncelikli durağı hedef düğüm yerine, gelişigüzel bir düğüm seçer ve paketin buraya gönderilmesini sağlar. İkinci evrede de varılan rastgele düğümden asıl hedefe doğru bir yol çizilmesini öngörür. Her iki evrede de e-küp kullanır fakat bu gelişigüzellik, yük dengesinin de hemen hemen eşit dağılmasına yardımcı olur. Diğer yandan ise, gecikme neredeyse iki katına çıkmaktadır çünkü hedef, kaynağın yanı başında bile olsa rastgele bir ara düğüme uğranması zorunluluğu bulunmaktadır.

Valiant'ın rastgele algoritmasındaki gecikme dezavantajını gidermek için yükü dengelenmiş ihmalkâr algoritma önerilmiştir. İlk evrede, hedefin bulunduğu çeyrekten (örgü ve simit topolojiler için kare alanın dörtte biri) ara düğüm seçilmesi sağlanır. Böylece, yük dengesinden bir miktar feragat edilse de gecikme süresi, kabul edilebilir bir mertebeye taşınmış olur. Dolayısıyla, bu yöntemin e-küp ile Valiant'ın rastgele algoritmasının ortasında durduğu söylenebilir.

3.2.2. Uyarlanabilen Algoritmalar ve Sıkışıklık Sorunu

Yukarıda sözü edilenlerin aksine, trafik durumunu göz önünde bulunduran (örneğin kuyruktaki meşguliyet) ve buna göre yolu belirleyen, uyarlama kabiliyetine sahip algoritmalardır. Bu özelliği ile belirlenimci ve ihmalkâr algoritmalardan farklılaşır, verimlilik / gecikme en iyileştirmesinde bir adım öne çıkar. Diğer taraftan algoritmanın bu yetenekleri, yapının karmaşıklığını da artıran özelliklerdir.

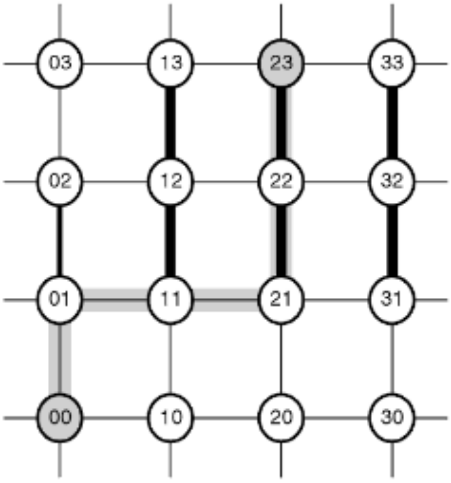
Uyarlanabilir yönlendirmenin mantığını kavramak adına, Şekil 3.2'yi yorumlayalım. 3 düğümü, 7 düğümüne bir paket göndermek istiyor ve 4-6 yolu ya da 2-0 yolu üzerinden, şeklinde iki seçeneği mevcut. Bu esnada da 5 ve 6 düğümleri arasında, bant genişliğini şişirecek bir trafik bulunmakta. Temel soru: bu bilgi, 3'ün daha iyi olan yolu seçebilmesi için 3'e doğrudan ulaşamadığına göre nasıl gelmeli?



Şekil 3.2. Örnek yüzük topoloji

İlgili trafiğin zıt yönünde, "ters basınç" ismi verilen bir teknik ile uçtan uca bu iletilebilir. Ancak bilgi paylaşımı düğüm 3'e ulaştığı anda, söz konusu yoğun ağ trafiği sona ermiş hatta öteki güzergâh üzerinde benzer bir sıkıntı başlamış olabilir. Bu sebepten ötürü, iletilen tıkanıklık bilgisinin güncelliği de oldukça önemli.

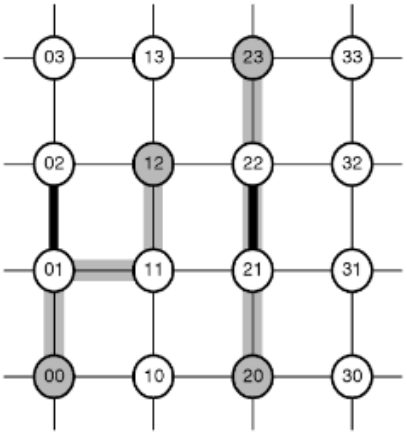
Bir de bilginin yerelliği konusunda ters tepmiş Şekil 3.3'teki örneği inceleyelim. Kabaca 00 düğümü, 23 düğümüne veri içerikli paketler göndermek istiyor ve rota üzerindeki 01 düğümü, üstteki hafif sıkışıklık yüzünden alternatif yolu tercih ediyor fakat bu durum, daha ağır bir sıkışıklıkla karşılaşılıp pahalıya patlamasına neden oluyor.



Şekil 3.3. Bilginin yerelliği probleminin oluştuğu bir örgü topoloji

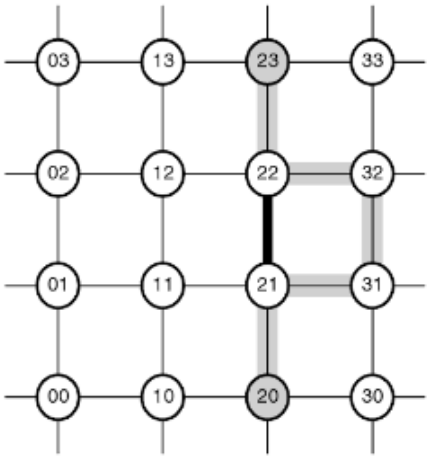
Minimal uyarlanabilen algoritmalar, isminden de anlaşıldığı gibi, kaynak ile hedef arasındaki en kısa yolları yeğleyen, uyarlanabilen bir algoritma çeşididir. Eğer birden fazla birbirine eşdeğer güzergâh varsa, yapısı gereği trafik açısından rahat olanı tercih eder. Bunu yaparken de “üretken vektör” sisteminden faydalanır. Bu vektör, (+x, -x, +y, -y) şeklindedir ve hedefe giderken kullanılacak doğrultuları belirler.

Minimalin zaafını da içeren bir örnekle devam edelim. Şekil 3.4’te yer alan şemada, 00 düğümünden 12 düğümüne giden paket akışı (1, 0, 1, 0), yoğun olan 01-02 yolunu tercih etmeyerek hem trafikten hem yoldan kazanmıştır. Ancak 20 düğümünden 23 düğümüne giderken eşdeğer bir yol bulunamadığından, trafiğe razı olunmuştur.



Şekil 3.4. Minimal uyarlanabilen algoritmaların iyi ve kötü yanları

Tam uyarlanabilen algoritmalar ise minimal yaklaşımın aksine, yolu uzatma pahasına da olsa, trafik durumunu birinci öncelik olarak değerlendiren, uyarlanabilen bir algoritma çeşididir. Bu sayede, en kısa yol haricindeki alternatifler de dâhil olmak üzere bütün güzergâhları gözeterek “tam” adını hak eder. Şekil 3.5’teki örnekte, minimalde yaşanan sıkıntının atlatıldığı rahatlıkla görülebilir.



Şekil 3.5. Tam uyarlanabilen algoritmaların trafiği gözetmesi

Minimale benzer bir mantıkla hareket ederek trafik durumları birbirine eşit olduğunda kısa yolu tercih eder. Dolayısıyla “yollar açıksa” minimal algoritmalar gibi davranır lakin değilse, “üretken olmayan” vektörleri de göz önünde bulundurur. Burada dikkat edilmesi gereken bir nokta ise her çeşit yol seçilebileceğinden, tekrar tekrar aynı düğüme gelinmesi ve kilit sorunlarının oluşması riskidir. Çözüm, U dönüşünü yasaklamak veya tümden bir dönüş modeli önermek olabilir.

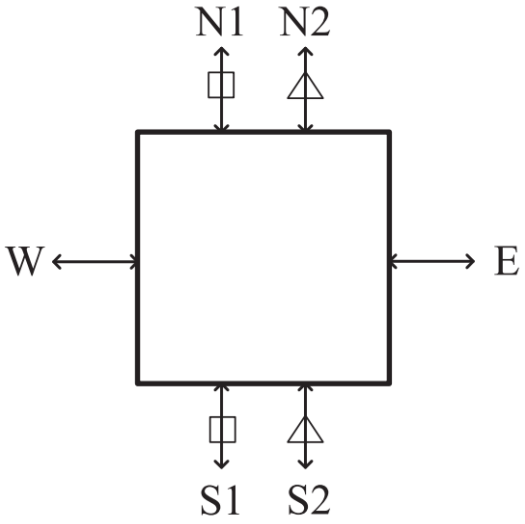
4. GELİŞTİRİLEN YÖNLENDİRME ALGORİTMASI

Bu kısımda, tez çalışmasının asıl odağını oluşturan yüksek uyarlama kabiliyetli ve Yonga-üstü-Ağ üzerinde yaşanabilecek muhtemel hatalara karşı dayanıklı algoritma tasarımının detayları verilecektir. Sırasıyla; kullanılan yönlendirici modeli, faydalanılan öncül algoritma ve önerilen algoritma kapsamındaki çözümlerle beraber ilişkili problemlerden bahsedilecektir.

4.1. Yönlendirici Modeli

2 boyutlu (XY) Yonga-üstü-Ağlarda sıradan bir yönlendiricide temel olarak kuzey, güney, doğu, batı ve yerel olarak yön başına bir giriş-çıkış kanalı temsil edilir. Ancak bu yapı, kilitlenme problemleri gibi bazı önemli nedenler için değişik de tercih edilebilir. Bu noktada, sanal kanal terimi gündeme gelir ve fiziksel kanallar bunlara bölünür.

Sanal kanal kullanımının diğer yöntemlere karşı yararı olsa da miktarının artışı, Yonga-üstü-Ağlar için maliyetleri olumsuz yönde etkilemektedir [24, 25]. Bu nedenle, tasarlanan algoritma bunlar için dengeyi bulacak en uygun çözümü hedeflemiştir: çift Y ağları (double-Y networks) [26, 27]. Yapı, X boyutu adına hiç sanallaştırma yapmazken Y boyutu boyunca yalnızca iki sanal kanaldan meydana gelir. Bu noktada, sanallaştırma yapılacak eksen için X'in seçilmemesi örgü gibi simetrik topolojilerde sonucu değiştirmemekle beraber, uygulamaya özgü olarak çift kanal noktası değiştirilebilir. Yaklaşım, toplam 7 kanal içeren (yerelle birlikte) ve Şekil 4.1'de görülebilecek bu yönlendirici tipi üzerinde geliştirilmiştir. Resimdeki modelde yer alan kanallardan N1 kuzey-1, N2 kuzey-2, E doğu, S1 güney-1, S2 güney-2 ve W batı kanallarını ifade etmektedir.



Şekil 4.1. Çift Y ağı

4.2. HAFTA Yaklaşımı

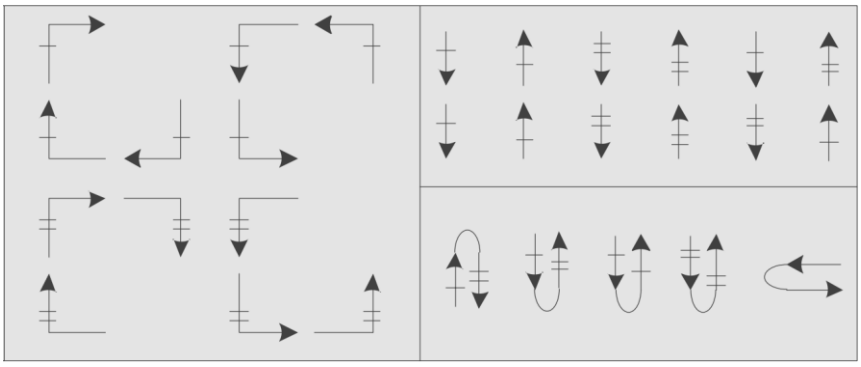
Bu bölümde, Yonga-üstü-Ağlarda yönlendirme yapabilen Yüksek Uyarlamalı Hatalara Dayanıklı Algoritma (HAFTA) sunulmaktadır. Sıradan yöntemlerin aksine bu tasarım, sadece tıkanıklık farkındalık kabiliyetini değil, aynı zamanda hata dayanıklılığı kabiliyetini de kapsar. Algoritma, özgün modeli ile yön başına düşen trafik sıkışıklıklarını keşfeder ve keşfettiği bu verileri bulundurduğu referans tablosunun tazeliğini, yepyeni bir olasılıksal yaklaşımla sürdürür. Ayrıca yaklaşım, ağdaki hatalı noktaları akılcı bir şekilde tespit eder ve nihayetinde rota, yüksek uyumluluğa bağlı olarak ilerler.

4.2.1. HARAQ Yöntemi

Bahsedilen hatalara dirençli uyarlanabilen yönlendirme tasarısı, HARAQ yöntemini temel almaktadır [28]. Bu yaklaşım, benzer şekilde çift Y ağının üzerinde konumlandırılan bir makine öğrenme yaklaşımı ile tıkanıklık farkında (congestion-aware) bir model sunmaktadır.

İlk olarak, ağda ne ölü kilit (deadlock) ne de canlı kilit (livelock) problemlerinin meydana gelemeyeceğini garanti altına alır. Bu hipotez, belirli kurallarla sınırlandırılmış bir dönüş modeli ve numaralandırma mekanizması tarafından

desteklenir [29]. Bunların ışığında da olası trafik akışının herhangi bir döngü ile sonuçlanmaması sağlanmış ve hedefe ulaşması güvence altına alınmış olur. Aynı zamanda, yol çeşitliliğini en üst düzeye çıkarmak için minimal olmayan bir yönlendirme tekniği de uygular ve bu doğrultuda, 180 derecelik dönüşlerin yapılabilmesine dahi izin verilir. Dolayısıyla, algoritmanın sıkışıklık durumlarında uyarlanabilirliği yüksek tutulmuş olur. Şekil 4.2'de HARAQ yönteminde izin verilen dönüşlerin tümü, çift Y ağına uygun bir biçimde görülebilir.



Şekil 4.2. Dönüş modeli

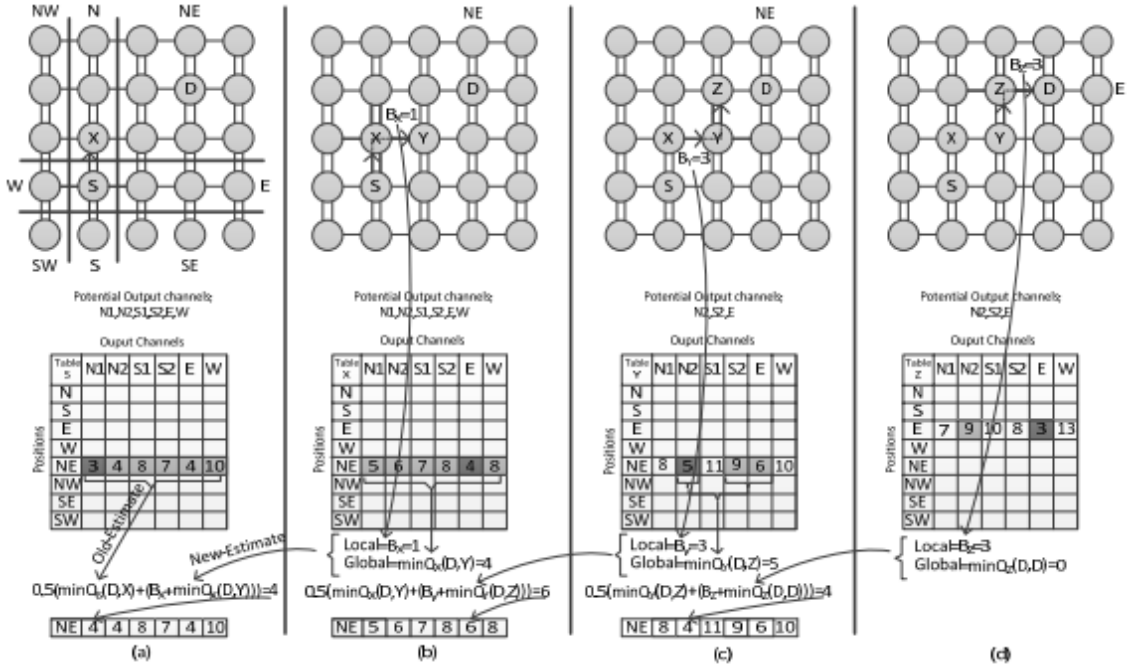
İkinci olarak, algoritmanın ana hedefi tıkanıklık farkındalığı, Q-Öğrenme yöntemiyle gerçekleştirilir [30]. Her bir yönlendirici; kuzey, güney, doğu, batı, kuzeydoğu, kuzeybatı, güneydoğu ve güneybatı yönleri doğrultusunda, muhtemel kanalların gecikme tahminleri için özelleştirilmiş bir Q-Tablosu muhafaza eder. Bölge tabanlı yönlendirme (Regional Routing, R-Yönlendirme) olarak adlandırılan yöntem, yön yaklaşımından ziyade ağdaki bütün anahtarlar için ayrı ayrı değerler tutan büyük Q-Tablolu diğer temel yönlendirmelere göre verimli bir Q-Tablosuna sahiptir. Yön kavramı ise tahmin edilebileceği üzere, geçerli anahtara göre hedef anahtarın konumuyla açıklığa kavuşturulur. Ayrıca düğümler, bir sonraki düğümün tıkanıklık durumu ile ilgili geri bildirimlerin tablodaki mevcut geçmiş verilerle harmanlanmasına göre gecikme değerini hesaplar ve Q-Tablosunu günceller. Daha sonra çıkış kanalı en asgari değer üzerinden, diğer bir deyişle en az sıkışık olandan, minimal olmayan bir yoldan olsa bile seçilmiş olur [31].

Son olarak HARAQ yöntemi, diğer Q-Yönlendirme türevlerinde de olduğu gibi, değer güncelleme kuralını güçlü bir şekilde sergilemektedir. Örnek vermek

gerekirse, algoritma tüm ağın trafik durumu hakkında daha iyi bilgi sağlamak için geri besleme akışlarında yerel ve küresel tıkanıklık verilerini kullanmaktadır [32]. Ek olarak, minimal olmayan tasarı, sıkışıklığın çok az olduğu veya hiç olmadığı durumlarda (başlangıç fazları gibi) uzun rotaların değerini 0'dan farklı puanlayarak minimal yolların avantajını korur. 4 bitlik bir Q-Tablosu yapısında bahsi geçen kısıtlamalarla alakalı ayrıntıya inmek gerekirse, en kısa kanalların başlangıç değeri "0000" dizisi ve minimal olmayan rotaların başlangıcı da "1000" dizisidir. Bununla birlikte, güncelleme doğruluğu gayet iyi olsa dahi, yeterli tazelik sağlanıyor mu sorusunun cevabı, bu tez çalışması kapsamında açığa kavuşturulacaktır.

Uygulanan tekniğe ait ilkelerin daha iyi bir şekilde anlaşılabilmesi adına, hedef D için S kaynak anahtarında bir mesajın üretildiği Şekil 4.3 örneği incelenmiştir. Modele göre, yerelde üretilen bir mesajı kuzeydoğudaki bir hedef anahtara iletmek için altı çıkış kanalı seçilebilir (yani N1, N2, S1, S2, E ve W). Şekil 4.3 (a)'da yer alan Q-Tablosunun, olası her bir çıkış kanalından kuzeydoğu bölgesine mesaj gönderme tahmini gecikmelerini gösterdiği görülebilir. N1 çıkış kanalı, daha düşük tahmini gecikmeye sahip olduğundan, mesaj bu kanaldan gönderilir. X anahtarında mesaj, S1 giriş kanalı tarafından alınır (Şekil 4.3 (b)). Benzer şekilde, bu sefer E çıkış kanalı en düşük gecikmeye sahiptir ve bu nedenle mesajın Y anahtarına iletilmesi için E seçilir. Bu esnada, yerel ve küresel tıkanıklık değerleri anahtar S'ye döndürülmelidir. İlgili mesajın Y anahtarına aktarımından önce X anahtarının giriş tamponunda (input buffer) bekleme süresi, yerel bilgi olarak geçer (Bx). X anahtarından, anahtar Y üzerinden (E kanalı) hedef bölgeye (kuzeydoğu) mesajların minimum tahmini yönlendirme gecikmesi de genel gecikme olarak kabul edilir (minQx (D, Y)). Bu iki bilginin toplam değeri, S anahtarından D noktasına giden yol hakkında yeni bir gecikme tahmini sağlar. S anahtarındaki Q-Tablosunun karşılık gelen değeri (yani satır: NE, sütun: N1), eski ve yeni gecikme tahmininin ortalaması alınarak güncellenir. Anahtar Y'de mesaj, batı giriş kanalı üzerinden alınır (Şekil 4.3 (c)). En düşük gecikme süresi olan çıkış kanalı, dönüş modeli gözetilerek üç olası çıkış kanalı arasından seçilir (yani N2, S2 ve E). Paketin iletilmesi üzerine, bir önceki adıma benzer olarak, sıkışıklık bilgisi X anahtarına geri döndürülür ve bu yeni gecikme süresi sayesinde gerekli güncelleme sağlanır. Son olarak mesaj, Z anahtarına S2 giriş kanalından

gelir (Şekil 4.3 (d)). Bu mesaj, N2 veya E çıkış kanalları üzerinden iletilerek hedefe ulaşabilir. Çıkış kanalı E, mesajı yönlendirmek için seçilir. Z anahtarındaki yerel gecikme, yine giriş arabelleğindeki bekleme süresi olurken; hedefe yönelik küresel gecikme, mesaj bir sonraki atlamada hedefe ulaştığından 0'a eşittir. Gecikme değerleri, Y anahtarına geri döndürülür ve Q-Tablosu güncellenir.



Şekil 4.3. Q-Tablolarının güncellenme süreci

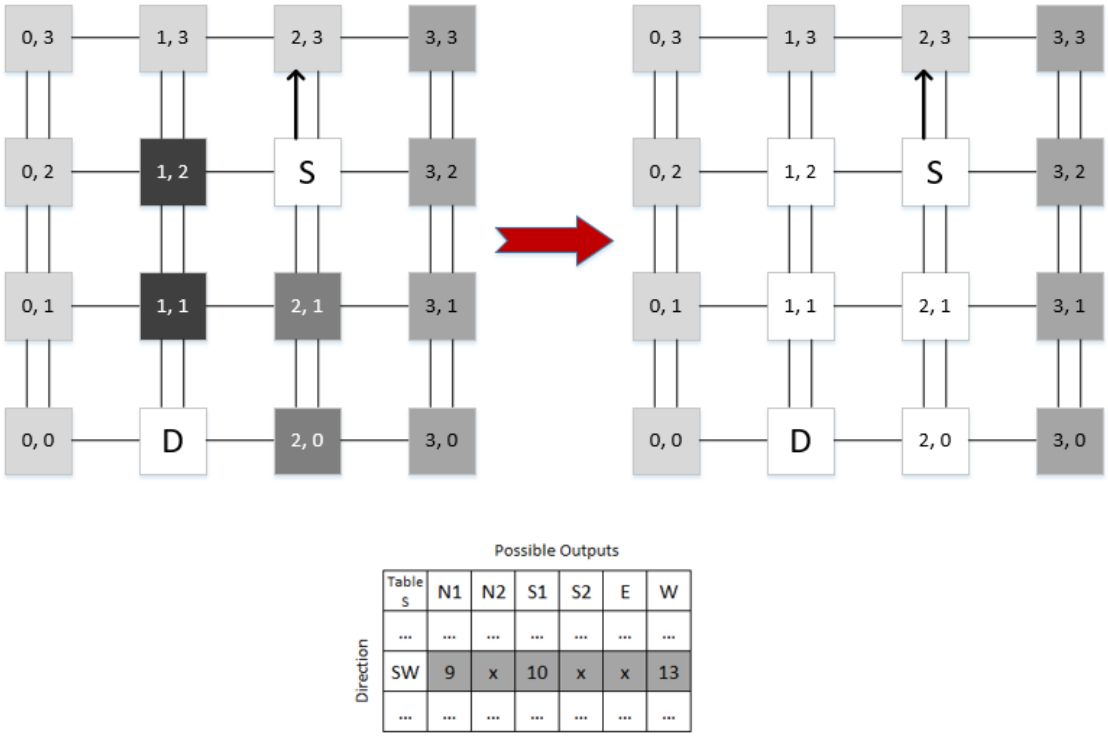
4.2.2. Sahte Tıkanıklık Sorunu

HARAQ algoritmasının güçlü yanlarına rağmen Q-Tablosu güncelleyicileri olarak bir sonraki düğümlerin seçimi, aynı yönlendiricilerde aynı yönler için aynı çıkışlar tekrar tekrar seçildikten sonra büyük bir problemi de beraberinde getirmektedir: sahte tıkanıklıklar. Bu, yalnızca seçilen ardıllarla yalnızca paket aktarım anlarında güncellemeler yapılmasından kaynaklanır. Enerji verimliliği açısından kesinlikle harika olan bu durum, çoktandır seçilmemiş ardılların Q-Tablosunda yer alan gecikme değerlerinin uzun süre güncellenmemesi sorununu doğurur. Söz konusu düğümlerin bu verilerinin taze hakikatler mi yoksa eskimiş yalanlar mı olduğu sorusuna yöntemin verebileceği bir cevap yoktur. Bu belirsizliğe bağlı olarak da gecikme tablolarında bir veya daha fazla eskimiş ve görece yüksek girdi varsa, bu değerler tablo arayıcısı tarafından, bağ üzerindeki tıkanıklık giderilmiş olsa bile

ne yazık ki “tıkanmış çıkış” olarak değerlendirilir. Bu sorunun üstesinden gelmek adına, minimal yollar sunan sanal kanalları gözetmek için periyodik bir kontrol yaklaşımı önerilmiştir.

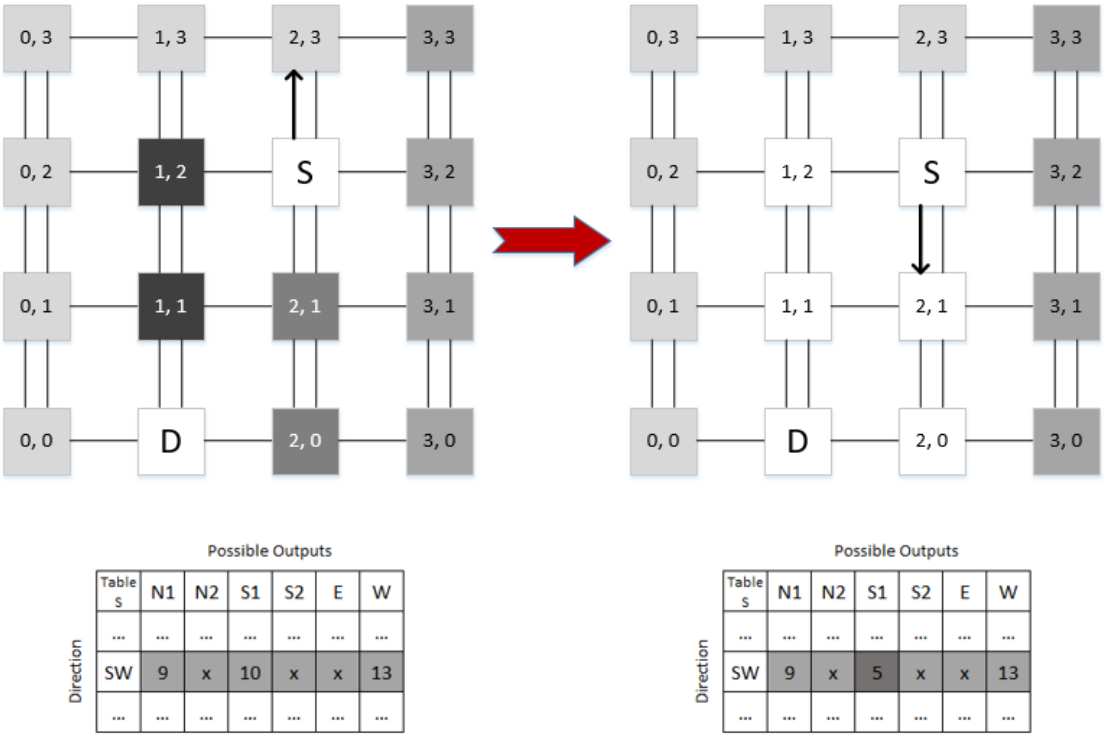
Mekanizmada yönlendirici, Q-Tablosunun anlık minimumundan bağımsız olarak her β transferde bir minimal bir çıkış kanalı seçer (Bir istisna: minimum değer minimal ise, bu periyodik seçim aşaması atlanır). Böylece gönderici, gerçek bir teslimattan sonra, seçilen kanal üzerinde gecikme değeri için tam bir kontrol sağlar. Yakın zamanda seçilmemiş ardılların belirli bir olasılık dâhilinde seçilmesine olanak tanınması ile referans tablosundaki sahte tıkanıklıklar en aza indirgenir. Yonga ağı, olası sahte tıkanıklık giderimi ile nefes alma şansı yakalar. Ayrıca, sıkışıklığı giderilmiş minimal bir yolun ıskalanması tehlikesi de azalır.

HAFTA yönteminde yer alan ve β anında yapılan seçim, basit bir olasılıksal model ile alakalıdır. Ortaya konulan tez çalışması kapsamında, bu parametrenin değeri 4 olarak kabul edilmiştir. HARAQ algoritmasındaki sahte tıkanıklık sorununu sayısal bir örnek ile açıklamak gerekirse, Şekil 4.4 dikkate alınabilir. Burada kaynak anahtar (2,2) kendisine göre güneybatıda konuşlanan (0,1) koordinatlı hedef anahtara, tabloya göre minimum gecikme değerini taşımasına rağmen güzergâh açısından minimallik sağlamayan kuzey-1 kanalı üzerinden bir iletim gerçekleştiriyor. Minimal yolların trafik yoğunluğu yaşaması nedeni ile böyle bir tercih meydana geliyor (daha koyu tonlu düğümlerde, veri yolu daha sıkışık). Ancak belli bir süre sonra, önceki seçim kuzey-1 linkinde sıkışma miktarı artmadıysa (tablodaki SW:N1 değeri sabitse), aynı kaynak-yön çifti (2,2 düğümü ve güneybatı) aynı akışı (N1) takip edecektir. Talihsiz bir şekilde tasarı, güncelleyip düzeltemediği sahte tıkanıklıkların tablosunda bulunması sebebiyle, trafik açısından rahatlamış minimal yolları (batı ve güney-1) pas geçecektir.



Şekil 4.4. HARAQ'ta sahte tıkanıklık durumu ve tablonun sabit kalışı

Anahtar, HAFTA algoritmasının kullanıldığı benzer bir senaryo altında ise, β-1 (burada 3 olmuş oluyor) aktarımın ardından gerçekleşecek ilk veri transferinde, dönüş modeline bağlı olarak güney-1 (S1) ve batı (W) sanal kanalları gibi seçilebilir minimallerden bir çıktıyı tercih edecektir. Kendi Q-Tablosundaki gecikme değerleri, bu çıkış-yön çiftleri için sırasıyla 10 ve 13 ise eğer, böyle bir durumda anahtar, ters orantı kurarak 13/23 olasılıkla güney-1 ve 10/23 olasılıkla batı kanalını seçecektir. Kurgunun güney-1 kanalı seçilerek devam ettiği ve güzergâh üzerindeki tıkanıklık geri bildirimlerinin 0 olduğu bir durumda, oluşabilecek Q-Tablosu ile elde edilen yönlendirme avantajı Şekil 4.5 üzerinden rahatlıkla görülebilir. Daha sonrasında karşılaşılacak olası aynı akışlarda da tablosunu güncelleyen bir HAFTA algoritmasının problemi bertaraf ederek ilerleyeceği ve HARAQ yöntemini performans olarak geride bırakacağı yorumu yapılabilir.



Şekil 4.5. HAFTA'nın sahte tıkanıklık farkındalığı ve tabloyu güncellemesi

4.2.3. Hataya Dayanıklılık Eklentisi

Hataların göz ardı edilmesi veya bertaraf edilmemesi, ağda gerçekçi olmayan performanslara sebebiyet verir. Böylesine yöntemlerde, bir bağ üzerinde veya yönlendiricide hata meydana geldiğinde paket kayıp miktarı önemli ölçüde artar. Bu çalışmada; HARAQ dâhil olmak üzere, çoğu yönlendirme algoritmasından farklı olarak hem hataya dayanıklılığı hem de tıkanıklık farkındalığını dikkate alan bir tasarım hedeflenmiştir. Algoritma, kalıcı ve geçici kusurların da ikisini birden ele almak için geliştirilmiştir. Bunun haricinde, HAFTA'nın hataya toleransı, fazladan bir eklenti biçiminde geliştirilerek algoritmanın her iki şekilde de performansının gösterilmesi için esnek bırakılmıştır.

Detayına girmek gerekirse sunulan yöntem, μ-geçici (μG) ve μ-kalıcı (μK) olarak 2 farklı parametre sunmaktadır. Birincisi, geçici bir hatanın varsayımı için bir eşik anlamına gelir. İkincisi ise, benzer şekilde, kalıcı hatalar için bir diğer eşik değerini belirler.

Bir yönlendirici, aynı sanal kanalda art arda μG kez aynı bit aktarımını algılarsa, o kanalın geçici olarak arızalı olduğu tespitinde bulunur. Bu vaziyetin μK defa ardışık olarak tekrar etmesi durumunda kanal, sürekli olarak hatalı şeklinde yorumlanır. Eğer yinelemeler μK eşik değerine ulaşmadan kesilirse, yani bir sonraki aktarımda farklı bit algılanırsa, μ değerleri için sayaçların sıfırlanması sağlanır.

Çeşitli amaçlar ve hesaplar doğrultusunda, μ parametreleri farklı değerler alabilir. Bu çalışmada μ değerleri 4 olarak seçilmiştir. Buna göre ağda, bir kanalın yanlışlıkla kalıcı hatalı olarak işaretlenmesi (yalancı pozitif) açıkça uzak bir ihtimaldir. Mevcut durumda oran, peş peşe 17 kere aynı bitin gönderilmesi olasılığından hesapla yaklaşık % 0.0015'tir. Dolayısıyla ihtimalin ihmal edilebilir olduğu söylenebilir. Ek olarak belirtildiği üzere, daha yüksek bir başarım hedefleniyorsa μ değerleri için başka atamalar yapılabileceği gibi, parametrelerin (μG ve μK) birbirinden farklı olması da sağlanabilir.

Sayaç mekanizmasını ve hataya dayanıklılık eklentisini açıklığa kavuşturmak için bir örnek vermek gerekirse, Şekil 4.6 üzerinde tasvir edilen akış takip edilebilir. Bu noktada, bir çıkış kanalı peş peşe aynı biti (sent bit = 0) gönderiyor olsun. Benzerlik art arda devam ettikçe, ikinci gönderimden yani ilk aynılıktan itibaren (succession = 1), μG (μ-temp) aktarım başına döngüsel olarak 1 artar (mod 4) ve her 4 μG artışına karşılık μK değeri (μ-perm), 1 yükselir (μ-temp = 0 anlarında). Böylesine μK parametresinin arttığı zamanlarda algoritma, bu ardışıklığın rastgele olmayacağını düşünüp linkte / kanalda bir geçici hata (soft error) olduğu yorumunda (interpretation) bulunur ve paketi, başka bir çıkıştan yeniden gönderir. Sonrasında seri hala bozulmazsa şayet, bu düzen tekrara uğrar ve aynılık sayısı 16'ya ulaştığında (succession = 16) ilgili düğüm artık, söz konusu sanal kanalı daimi hatalı kipe (permanent error) çeker. Yönlendirici, bu çıkışı kullanmayı sürekli olarak terk eder.

Yine belirtmekte fayda var ki bahsedilen sekans herhangi bir aşamada sürdürülememiş ve çıkış kanalı farklı bir bit göndermiş olsaydı, her iki μ için de sıfırlama meydana gelecekti (μ değerlerinin 0-0 olduğu yeni bir ilk satıra dönüş, sent bit = 1).

Sent Bit	Succession	μ-temp	μ-perm	Interpretation
0	0	0	0	-
0	1	1	0	-
0	2	2	0	-
0	3	3	0	-
0	4	0	1	Soft Error
0	5	1	1	-
0	6	2	1	-
0	7	3	1	-
0	8	0	2	Soft Error
0	9	1	2	-
0	10	2	2	-
0	11	3	2	-
0	12	0	3	Soft Error
0	13	1	3	-
0	14	2	3	-
0	15	3	3	-
0	16	0	0	Permanent Error

Şekil 4.6. Hata varsayımı

4.2.4. Geliştirilen Sözde Kod

Algoritmayı genel hatlarıyla ortaya koyan iyi bir sözde kod; tasarımcılar, geliştiriciler ve alıntı yapacaklar için analiz bakımından büyük bir önem taşır. Klasik akışın aksine, HAFTA algoritması için de böyle açıklayıcı, kullanışlı bir referans oluşturulmuştur. Bir önceki bölümde belirtildiği üzere, hataya dayanıklılık eklentisi ve ana fonksiyondaki ona ait fazladan kontroller, altı çizili bir formda ve ayrı olarak yazılmıştır.

```
InCh: Giriş Kanalı, OutCh: Çıkış Kanalı
Dir: Hedef Yön
Q-Dir: İlgili Yön için Q-Tablosundaki Değerler Dizisi
Turn(InCh,Dir): Seçilebilir Çıkışları Döndür
Heal(OutputArray): Hatasız Kanalları Döndür
Size(OutputArray): Dizinin Boyutunu Döndür
DropJam(): Paketi Düşür ve Sıkışıklık Bildir
Min(Q-Dir,T): Q-Minimum olan Çıkışı Döndür
ProbMin(Q-Dir,T): Olasılıksal Minimali Döndür
IsShort(OutCh,Dir): Minimallik için Boolean Döndür

LOOP1

1- IF Dir={local} THEN OutCh={local}; EXIT
2- ELSE T=Turn(InCh,Dir); T=Heal(T);
     3- IF Size(T)=0 THEN DropJam(); EXIT
     4- ELIF Size(T)=1 THEN OutCh=T[0]; EXIT
     5- ELSE M=Min(Q-Dir,T); IM=IsShort(M,Dir);
          6- IF IM=1 THEN OutCh=M; EXIT
          7- ELSE β = β+1 (mod 4);
               8- IF β != 0 THEN OutCh=M; EXIT
9- ELSE P=ProbMin(Q-Dir,T); OutCh=P; EXIT
```

Şekil 4.7. HAFTA algoritmasının çıkış kanalı seçimi

HAFTA algoritmasına ait sözde kod, üç bölüm halinde incelenmiştir. İlk olarak yönlendirici, rotasyon yönergelerine uygun bir şekilde, paketin istikametini ve kullanılabilir kanalları saptar (Adım 1 ve 2). Eğer hata eklentisi devredeyse saptama evresinde hatalı kanallar, seçilebilir çıkışlara dâhil edilmez. Sonrasında, şayet kullanılabilir bir çıkış yoksa paket düşürülerek tıkanıklık geri bildirimi yapılır (Adım 3). Yalnızca 1 adet seçilebilir kanal varsa da algoritma koşulmadan doğrudan çıkış kanalı ataması yapılır (Adım 4). Arda kalan bütün koşullar altında, Q-Tablosunun minimumu gözetilerek bir çıkış belirlenir (Adım 5). Seçilen çıkış, minimal bir rota üzerindeyse β modeline girilmeden kanal tayini sonuçlandırılır

(Adım 6). Bunun haricinde kalan daha uzun yol (nonminimal) seçimleri halinde; minimum olmayan minimallerin, tıkanıklık olduğu tahmini yapılan en kısa yolların, 1/β olasılıkla (β, 4 alınmıştır) seçilmesi için periyodik bir sayaçla devam edilir (Adım 7). β parametresi, eşik değere ulaşmadıysa seçim korunurken (Adım 8) ulaşması durumunda bölüm 4.2.2'de anlatılan yöntem uyarınca başka bir çıkış ataması yapılır (Adım 9). Sürecin tümü, Şekil 4.7 üzerinden daha net görülebilir. Güç tüketiminin azaltılması adına, kanal seçiminin kesinleştirilmesinin ardından geriye kalan satırların işlenmemesi için çıkış fonksiyonlarının (EXIT) yerleştirildiği dikkat çekmektedir.

```
Pre(OutCh): Önceki Gönderilen Biti Döndür
Now(OutCh): Şimdi Gönderilen Biti Döndür
Zero-μ(): μ Değerlerini Sıfırla
TmpFoul(OutCh): Kanalı Geçici Olarak Hatalı İşaretle
PrmFoul(OutCh): Kanalı Kalıcı Olarak Hatalı İşaretle

LOOP2

1- IF Pre(OutCh)!=Now(OutCh) THEN Zero-μ(); EXIT
2- ELSE μG = μG + 1 (mod 4)
     3- IF μG != 0 THEN EXIT
     4- ELSE μK = μK + 1 (mod 4)
          5- IF μK != 0 TmpFoul(OutCh); JMP STEP2@LOOP1
          6- ELSE PrmFoul(OutCh); JMP STEP2@LOOP1


Bir geri bildirim (q-new) geldiğinde, Q-Tablosundaki
ilgili girişi şu şekilde güncelle:
q-value = (1-α) * q-old + α * q-new
```

Şekil 4.8. HAFTA algoritmasının hata tolerans ve geri bildirim akışı

İkincil olarak iletim fazı, hataların bertaraf edilmesi ve dolayısıyla hata tespiti için çok önemlidir [33]. Mekanizma, bir önceki gönderim ile aynı bit aktarımını arar

(Adım 1). Varsa, akış μ sayaçlarının kontrolüyle devam eder (Adım 2, 3 ve 4) ve eşik değerlere gelinirse arıza saptaması ile sonuçlanır (Adım 5 ve 6). Bu hipotezin altında yatan mantık ise, söz konusu çıkış kablosunda belli bir bite kilitlenmenin gerçekleşmesi olasılığıdır. Döngüde yapılan 5. ya da 6. adımdaki işaretlemelerin ardından algoritma, ilgili bu kanalları bir kez veya sonsuza kadar sağlıksız olarak kabul eder ve birinci döngüye dönmek suretiyle (JMP) güvenilir bir çıkış arar. Ayrıca sıfırlama yaklaşımı, hiçbir hata olmaması veyahut ardışık olmayan benzerlik ortaya çıkması gibi durumlarda, μ değerleri üzerinde devreye girer (Adım 1).

Son aşamada ise, Q-Yönlendirme modellerinin çoğunda yaygın olarak kullanılan geri bildirim formülü uygulanmaktadır [34]. β prensibi nedeni ile emsal algoritmalardan daha fazla güncelleme yapan yaklaşımın dengelenmesi için α, öğrenme oranı, 0.4 olarak seçilmiştir. Bahsedilen adımların sözde kod biçemi, Şekil 4.8 üzerinden görülebilir.

5. DENEYSEL SONUÇLAR

Bu bölümde, sunulan Yüksek Uyarlamalı Hatalara Dayanıklı Algoritmanın (HAFTA) başarımını ortaya koymak için kodlamaların yapılmasının ardından, birtakım testler gerçekleştirilmiş ve de temel alınan HARAQ algoritmasıyla kıyaslanabilmesi amacı ile çeşitli trafik modelleri, benzetim aracı yardımıyla değerlendirilmiştir. Deneyler tekdüze profil, sıcak nokta profili ve kasırga profili sırasıyla verilmiş ve hemen öncesinde, hazırlanan ölçüm ortamı ayrıntılarıyla açıklanmıştır.

5.1. Ölçüm Ortamı

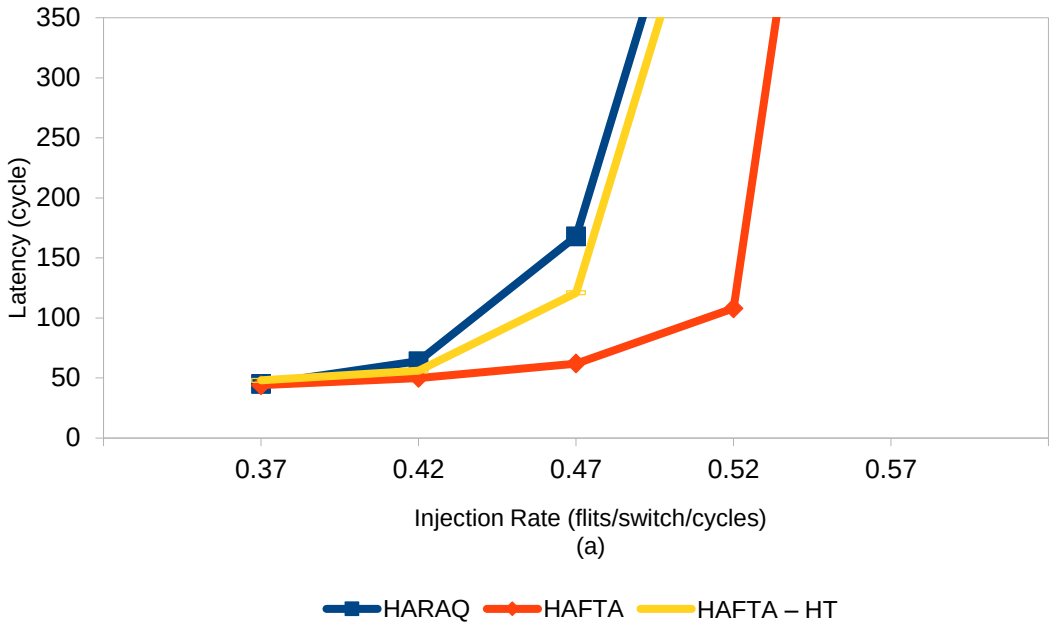
Kılavuz kabul edilen HARAQ algoritması ve HAFTA yöntemi gerçeklenmiştir. Bunun yanı sıra, HT eklentisine sahip olmayan (HT özelliği devre dışı bırakılmış) HAFTA, ağda bir hatanın yaşanmadığı şartlardaki başarısını ispatlamak için de başka bir kıstas olarak ölçüme dâhil edilmiştir. Esnek bir geliştirme ortamı için modülerliği sayesinde popülerliğini koruyan Yonga-üstü-Ağ benzetim aracı BookSim uygun görülmüştür [35]. Ağların hem gecikme hem de verim analizleri karakterize edilmiştir. Bu noktada gecikme hesaplamalarında, paketin hedefe varış süresi ele alınırken verim hesaplamalarında, gönderilen paketlerin hedefe varış oranı irdelenmiştir.

Paket büyüklüğü 10 parçacık (flit) şeklinde tanımlanmış, bant genişliği bir hat için çevrim (cycle) başına 1 parçacık olarak ayarlanmıştır. Ayrıca, her giriş kanalında 8 parçacıklık arabellek (buffer) boyutu vardır ve yönlendiriciler için veri genişliği 64 bit olarak belirlenmiştir. Yapılan bütün deneyler, "çevrimde anahtar başına enjekte edilmiş parçacıklar (parçacık / anahtar / çevrim)" birimine sahip olan artırımlı akıtma oranları (injection rate) kullanılarak gerçekleştirilmiş ve grafikler, algoritmaların ayrışan bölümlerine dikkat çekebilmek adına yakınlaştırılmış stilde gösterilmiştir. Dahası, değerlendirmeyi kararlı bir yakınsamaya sokabilmek için 12.000 ısınma döngüsü ve 200.000 örnekleme döngüsü olacak şekilde tanımlanan birtakım devirlerden faydalanılmıştır. Ortalama performans değeri, ısınma süresi sonrasında ve örneklemeler üzerinden alınmıştır. Ağın sergilediği

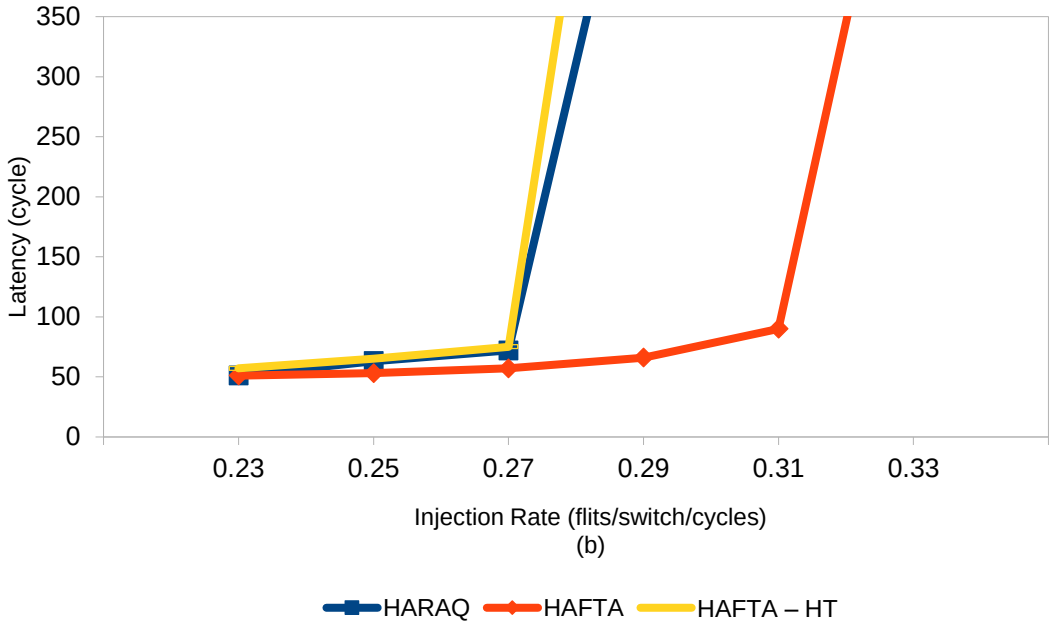
hem dengeli hem de stresli davranışları test edebilmek amacıyla sıcak nokta (hotspot), tekdüze rastgele (uniform random) ve kasırga (tornado) trafik modelleri dikkate alınmıştır [36].

5.2. Tekdüze (Uniform) Profil Deneyi

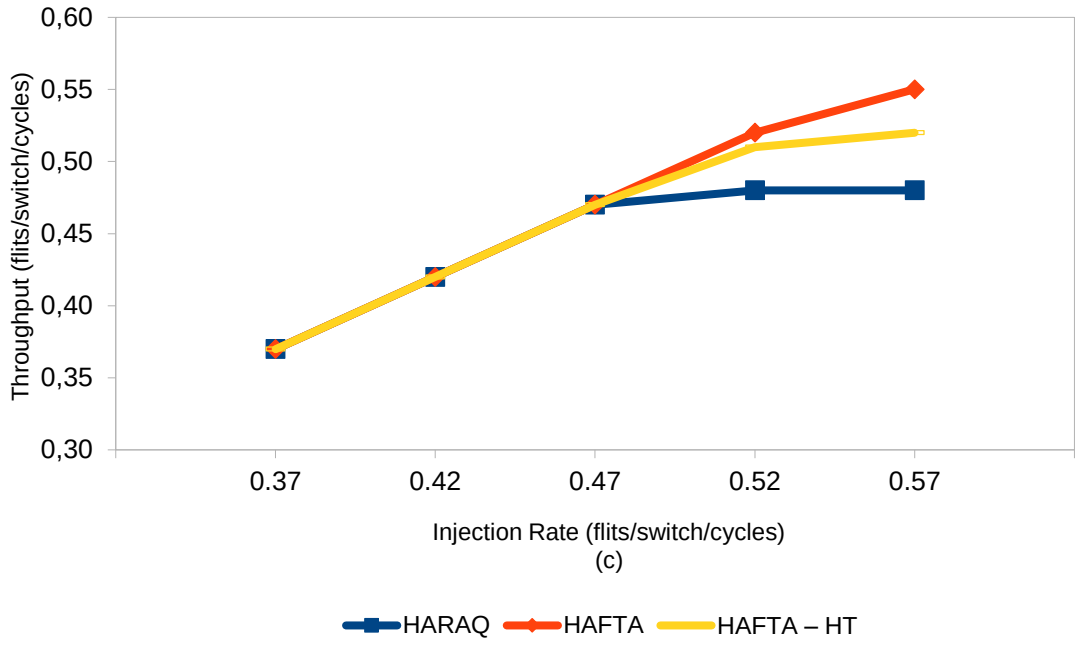
Tekdüze profilde, bir başka ifade ile düz dağılımda, her kaynağın her hedefe gönderdiği trafik miktarı eşittir. Bu durum, trafik yükünü idealleştirir ve böylece algoritmaların saf performansı gözlemlenebilir. Şekil 5.1, Şekil 5.2, Şekil 5.3 ve Şekil 5.4'te, 5x5 ve 8x8 örgü topolojileri için hem gecikme hem de verimin ortalama değerleri tasvir edilmiştir. Görüldüğü üzere; bütün algoritmalar düşük, hatta orta yüklerde benzer davranmaktadır ancak HAFTA, daha yüksek yüklerde gittikçe farklılaşmaktadır. Zira β modeli, özellikle ağır yüklere maruz kalındığında rüştünü ispat etmektedir. HARAQ yöntemi ise artan bir eğilimle minimal olmayan, daha uzun, kanalları seçmekte ve bu da sahte tıkanıklık olaylarını arttırmaktadır. Öte yandan, hataya dayanıklılık eklentisi ve rastgele bir hata ile test edilen HAFTA-HT algoritması (sarı renk), genel olarak başarılı bir grafik sergilemektedir. Bununla beraber, görece zayıf ölçeklenebilirlik nedeniyle daha büyük ağlarda biraz geride seyretmektedir. Her iki topoloji benzetiminin ortalama grafik değerleri göz önünde bulundurulduğunda HAFTA, HARAQ algoritmasını gecikmede % 30, verimde % 7 geride bırakmaktadır.



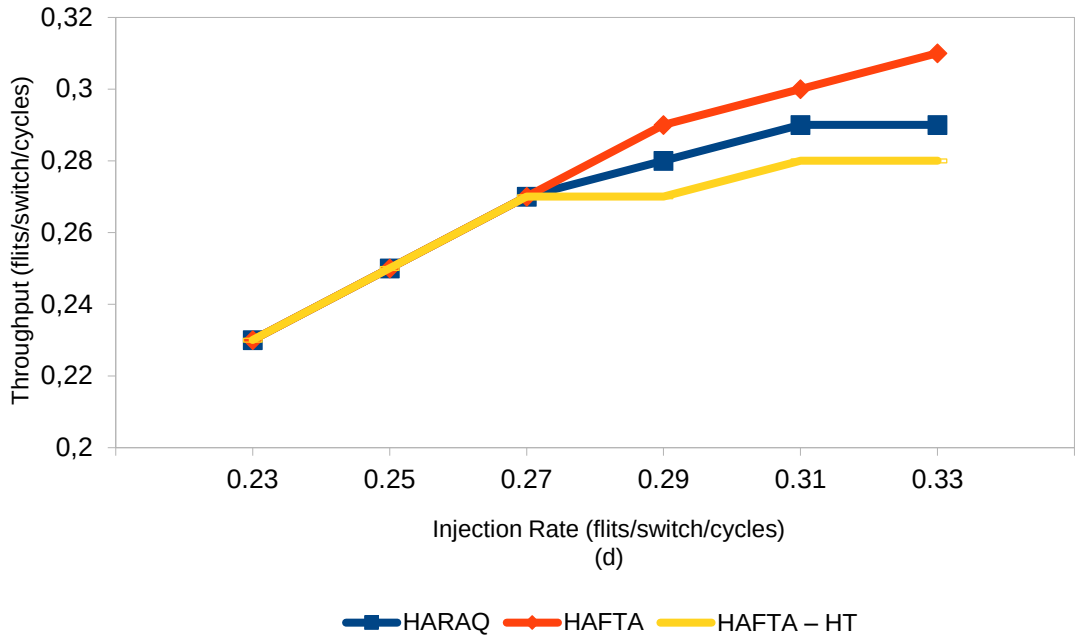
Şekil 5.1. Tekdüze model altında 5x5 örgüsü için gecikme analizi



Şekil 5.2. Tekdüze model altında 8x8 örgüsü için gecikme analizi



Şekil 5.3. Tekdüze model altında 5x5 örgüsü için verim analizi

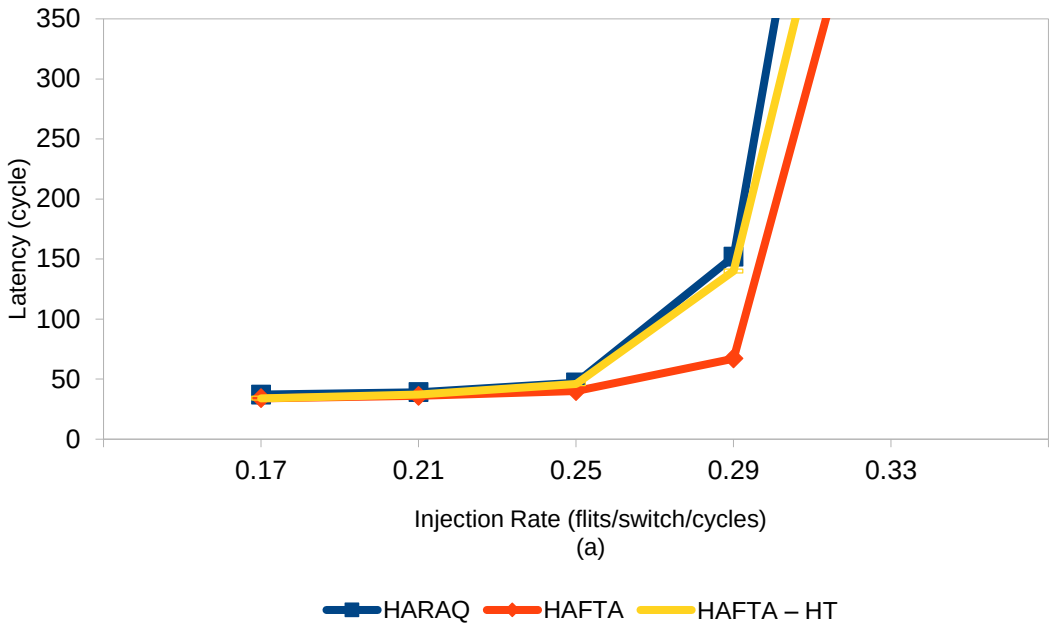


Şekil 5.4. Tekdüze model altında 8x8 örgüsü için verim analizi

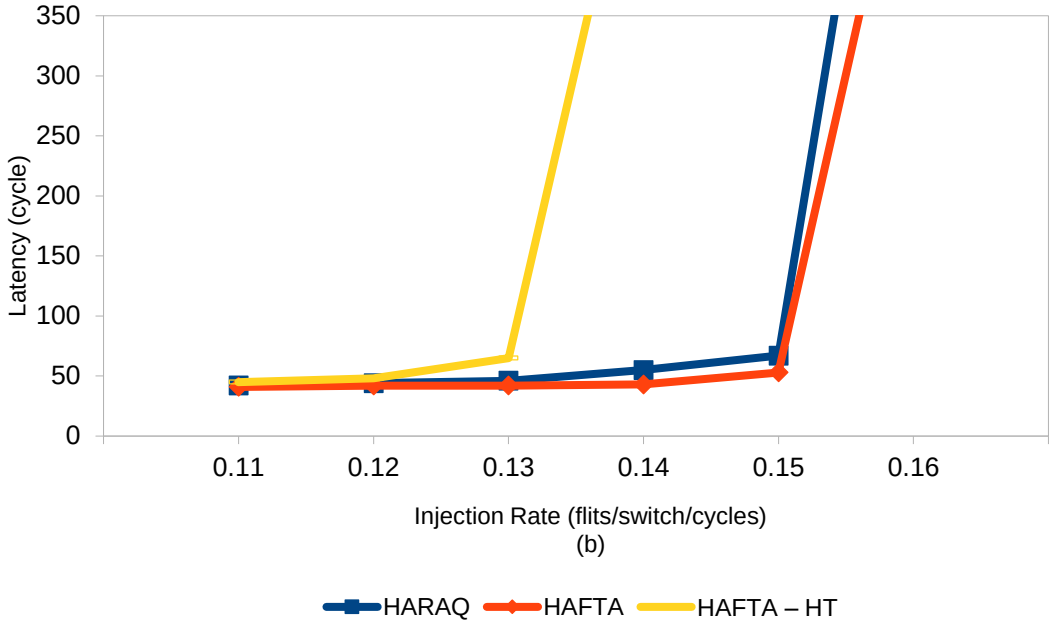
5.3. Sıcak Nokta (Hotspot) Profili Deneyi

Bu trafik altında akış, sıcak nokta olarak seçilen anahtar veya anahtarlar dışında düzgün dağılım şeklinde devam eder. Model, sıcak noktayı ise gönderim hedefi

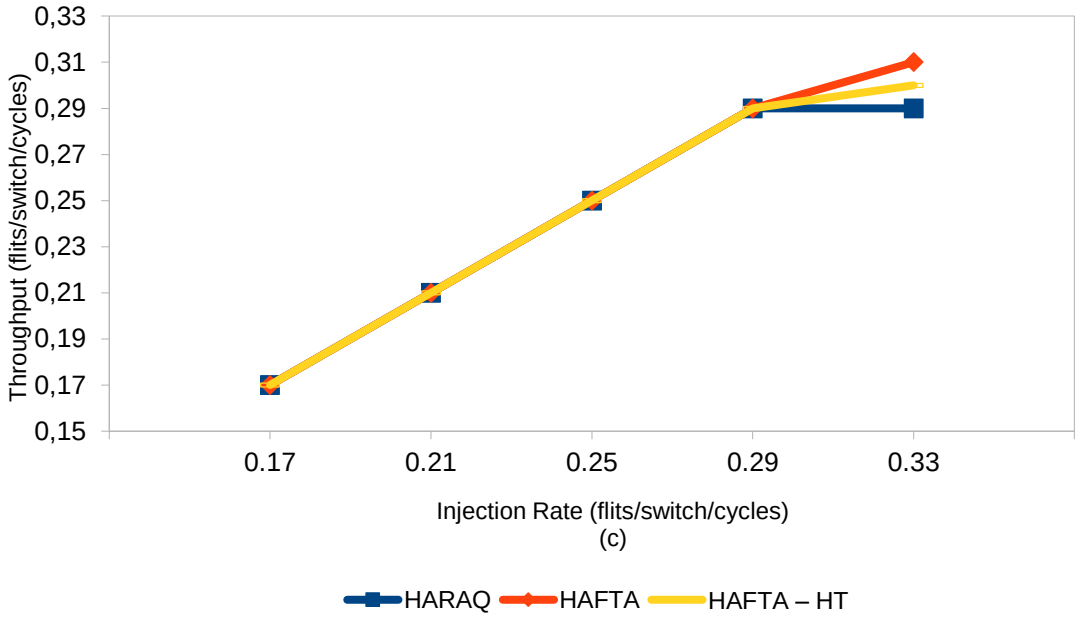
için ekstra bir H olasılığıyla seçer. Benzetimler, 5x5 örgüde (3,3) düğümü ve 8x8'lik örgüde (4,4) düğümü olan merkezi tekil sıcak noktalar aracılığıyla gerçekleştirilmiştir. Şekil 5.5, Şekil 5.6, Şekil 5.7 ve Şekil 5.8 üzerinde, H = 0.1 koşulunda, belirtilen trafiğin değerlendirilmesi gösterilmektedir. Görüldüğü üzere, küçük topolojide bütün algoritmalar hemen hemen aynı performansı sergilerken HAFTA'nın HARAQ'a üstünlüğü, tekdüze profile göre bir miktar daralma eğilimindedir. Bu eğilim, topoloji genişledikçe devam etmekte ancak HAFTA'yı geride bırakacak noktaya varamamaktadır. Nitekim β modelinin, trafik ağırlığının belirli noktalara toplandığı durumlarda avantajının minimize edildiği söylenebilir ancak algoritmanın geneline de aşırı yük getirdiği söylenemez. Öbür yandan rastgele hata altında kusura toleranslı HAFTA-HT, büyüyen topolojiyle beraber ölçeklenebilirlik açısından sorgulanabilir görülmektedir fakat her şeye rağmen, başarım bakımından kabul edilebilir gözükmektedir.



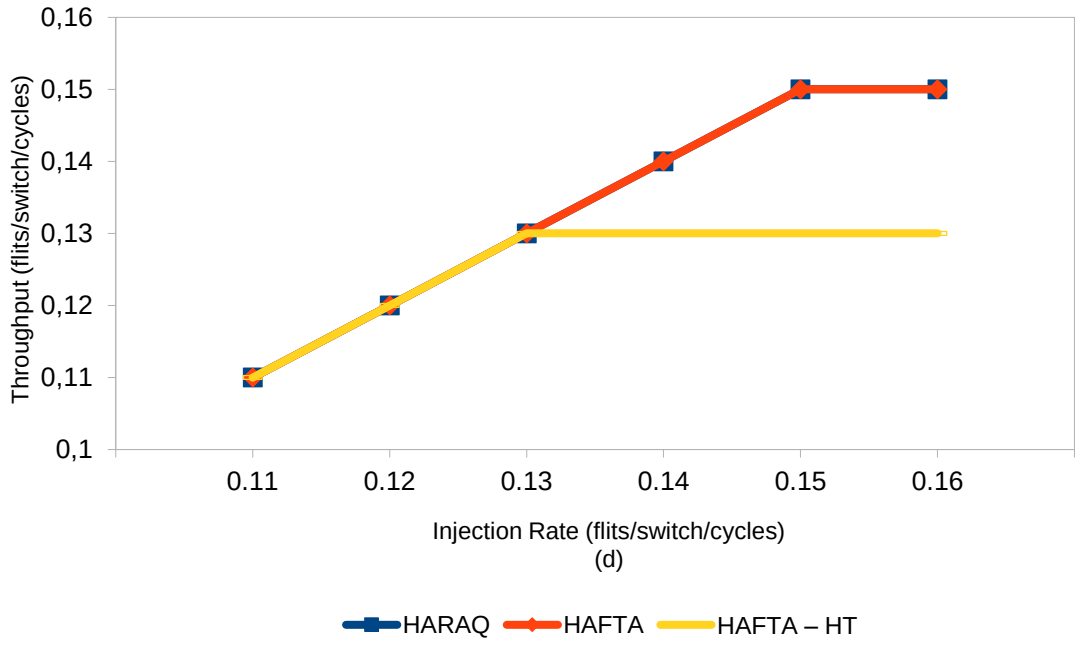
Şekil 5.5. Sıcak nokta modeli altında 5x5 örgüsü için gecikme analizi



Şekil 5.6. Sıcak nokta modeli altında 8x8 örgüsü için gecikme analizi



Şekil 5.7. Sıcak nokta modeli altında 5x5 örgüsü için verim analizi

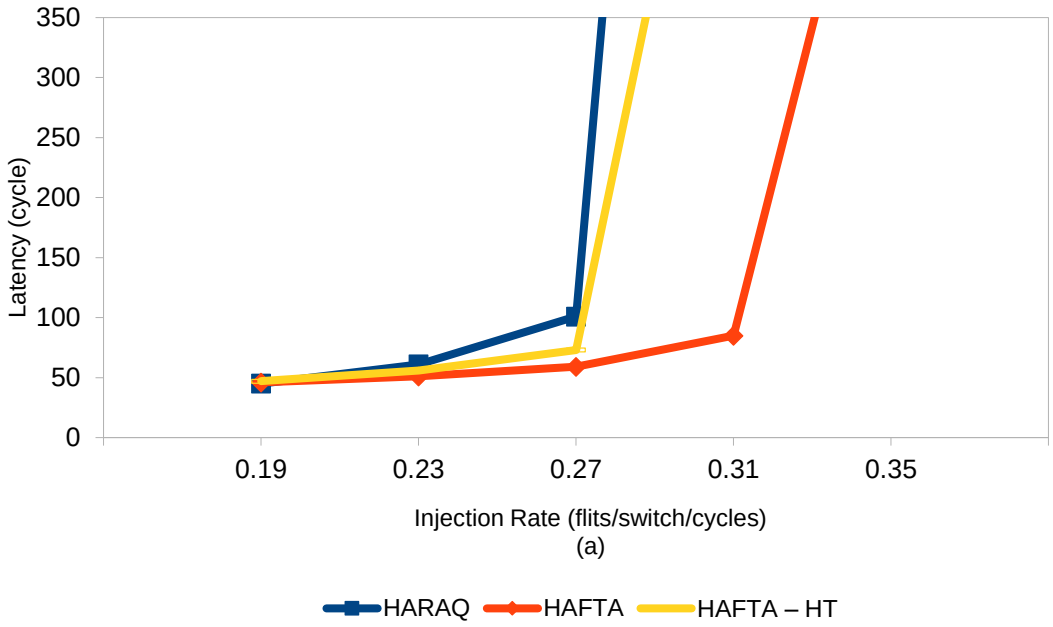


Şekil 5.8. Sıcak nokta modeli altında 8x8 örgüsü için verim analizi

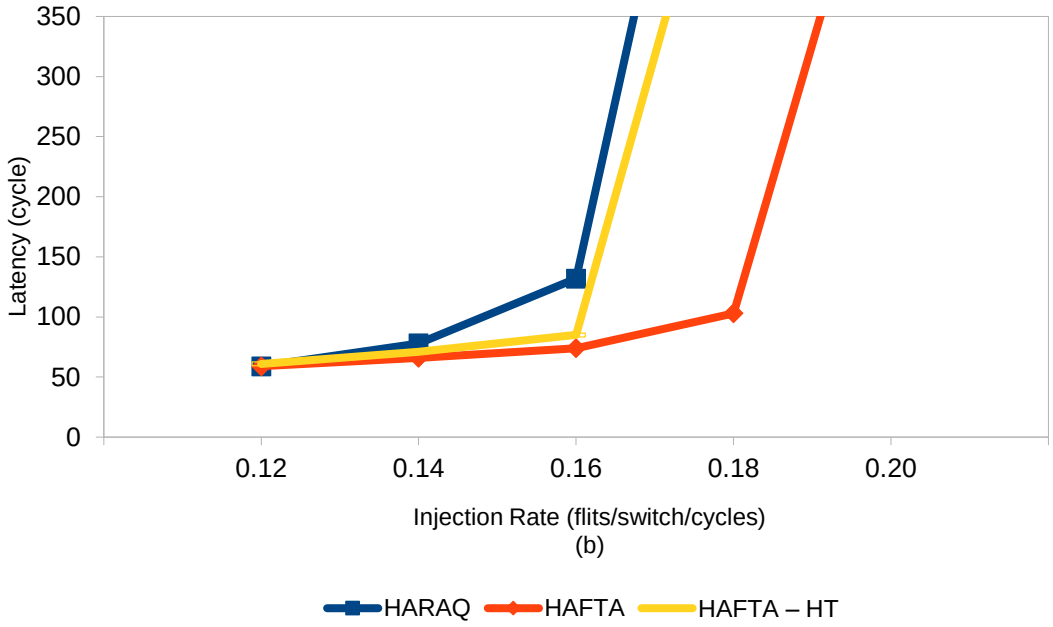
5.4. Kasırga (Tornado) Profili Deneyi

Bir kasırga trafik modeli altında her anahtar, kaynağın her koordinatına göre yarıçapın yarısından fazla olan (mod yarıçapında) düğüme paket trafiği gönderir. Bu örüntü modeli simit topolojilerini idealleştirir ancak genellikle örgü topolojide daha az sıkışık köşeler, dengeli kenarlıklar ve daha sıkışık bir göbek oluşturur. Bununla birlikte, geleneksel modeller arasında öngörülemeyen profillerden biri olarak kabul edilmektedir [37]. Değerlendirme aşamasında ise, hem gecikme hem de verim hesaplamaları 5x5 ve 8x8 topolojileri için tamamlanmış; Şekil 5.9, Şekil 5.10, Şekil 5.11 ve Şekil 5.12 üzerinde gösterilmiştir. Ölçeklenebilirlik ve başarı derecelendirmesi açısından, diğer iki profile göre daha istikrarlı sonuçlar elde edilmiştir. Söz konusu trafik için ölçüm sonuçları, özellikle yüksek yüklerde, HAFTA algoritmasının HARAQ algoritmasını hatalı veya normal koşullarda geçtiğini kanıtlar niteliktedir. Bunun nedeni, profilin aynı yönde hareket etmesi ve sonuç olarak sahte tıkanıklık sayısının maksimuma çıkarılmasıdır. Böyle olunca HAFTA yönteminin avantajları, HARAQ sisteminde pek dikkate alınmayan ve yükü düzgünleştirerek μ modelini (HT) de destekleyen β yaklaşımı açısından ortaya çıkmaktadır. Sayısal olarak bakıldığında da HAFTA, HARAQ algoritmasını

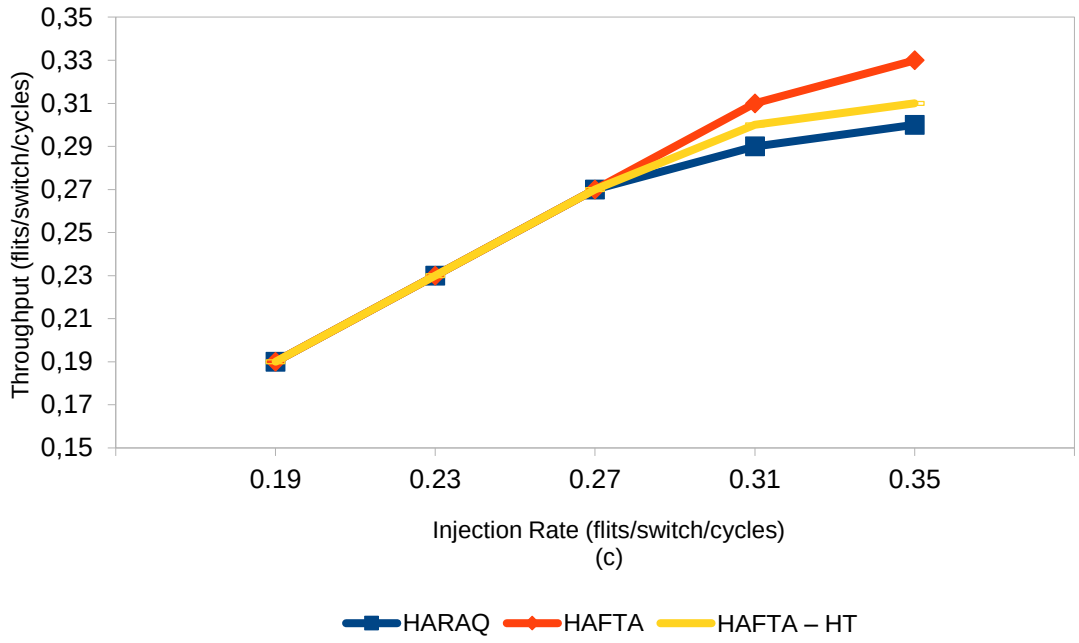
(her bir şema için ortalamalar gözetildiğinde) gecikme konusunda yaklaşık % 30 ve verim konusunda % 8 aşar.



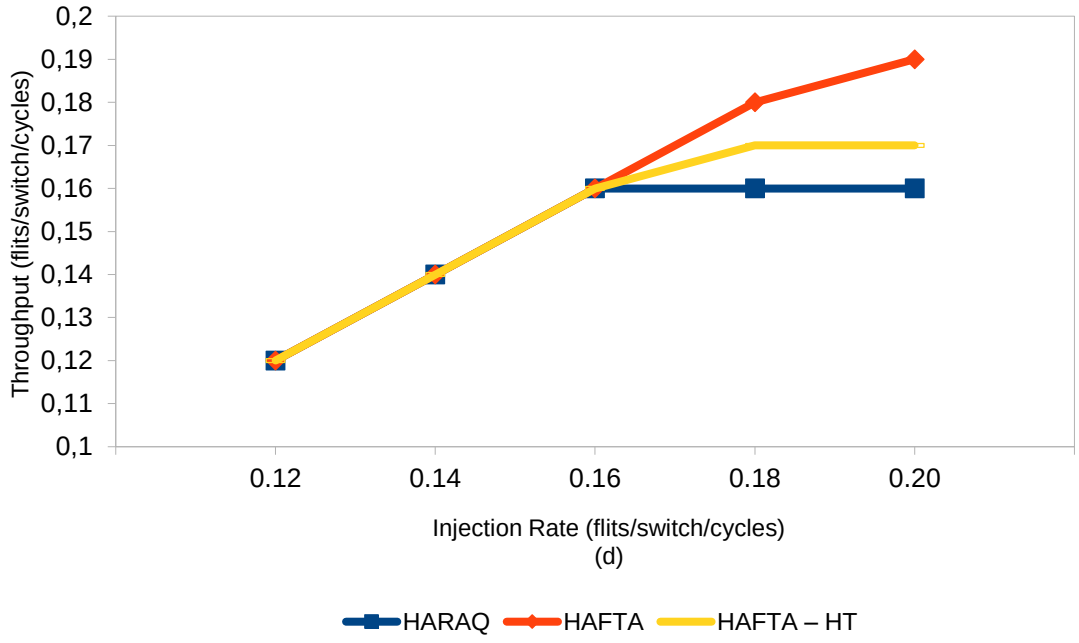
Şekil 5.9. Kasırga modeli altında 5x5 örgüsü için gecikme analizi



Şekil 5.10. Kasırga modeli altında 8x8 örgüsü için gecikme analizi



Şekil 5.11. Kasırga modeli altında 5x5 örgüsü için verim analizi



Şekil 5.12. Kasırga modeli altında 8x8 örgüsü için verim analizi

6. SONUÇ VE TARTIŞMA

Bu tez çalışmasında, HAFTA isimli, hataya dayanıklılığı göz önünde bulunduran tamamen uyarlanabilir bir YüA algoritması sunulmuştur. Aynı zamanda önerilen yöntem, çift Y ağı ile maliyet ve performans dengesini de korumaktadır. Ayrıca yaklaşım, paketleri de kısır döngüler oluşturmadan etkili bir şekilde çevirmektedir. Sahte tıkanıklık gerçekleştiğinde, gerekli olmayan uzun yolların seçilmesinden kaçınacak bir mekanizmadan faydalanmaktadır. Benzer şekilde, kusur esnekliği, olasılıksal bir temele dayanan sayaç yöntemi ile elde edilir. Bunlar, hem kusursuz hem de hatalı koşullarda tıkanıklığı azaltan sağlam bir ağı temellendirmektedir. Sonuçta yapılan deney ve analizler, HAFTA'nın çeşitli trafik profilleri altında sağlam bir yönlendirme düzeneği olduğunu göstermektedir.

Yonga-üstü-Ağlar için geliştirilen yönlendirme algoritması HAFTA, daha önceki mevcut çalışmalar ile kıyaslandığında, gecikme başarımının yanı sıra topolojilerde baş gösterebilecek geçici ve kalıcı hataları da göz önüne alarak benzerlerinden farklı bir noktaya oturmuştur. Yöntem kapsamında faydalanılan asgari düzeyde sanal kanal ve olasılıksal makine öğrenmesi yaklaşımlarıyla otomasyon ve verimlilik anlamında, modern bir bakış açısı sunmuştur.

Tasarlanan algoritmanın bir adım daha öteye gitmesi adına, bazı iyileştirmeler ve ek çalışmalar yapılabilir. Bunlar; tasarımın donanımsal ortamda gerçekleştirilerek sağlıklı bir enerji modeli çıkarılması, yönlendirme tablosunda ara yönlere nazaran daha az düğüme ilişkin gecikme tahmini sağlayan ana yönlerin budanarak yönlendirme tablosunun küçültülmesi ve yalnızca bir sonraki düğümden gelen güncelleme verisinin tam anlamıyla küreselleştirildiği bir mekanizma kullanılması gibi alt başlıklar olabilir.

KAYNAKLAR

[1] R. R. Schaller, “Moore’s law: past, present and future,” in IEEE Spectrum, 34 (6) (**1997**) 52-59.

[2] W. J. Dally and B. Towles, “Route packets, not wires: On-chip interconnection networks,” in Proc. Design Automation Conference, **2001**, pp. 684-689.

[3] L. Benini and G. De Micheli, “Networks on chips: A new SoC paradigm,” Computer, 35 (1) (**2002**) 70-78.

[4] W. Tsai, D. Zheng, S. Chen, and Y.H. Hu, “A fault-tolerant NoC scheme using bidirectional channel,” in Proc. DAC, **2011**, pp. 918-923.

[5] M. R. Kakoee, M. Daneshtalab and S. Safari, “HW/SW architecture for soft-error cancellation in real-time operating system,” in IEICE Electronics Express, 4 (23) (**2007**) 755-761.

[6] J. Duato, S. Yalamanchili and L. Ni, “Interconnection networks: an engineering approach,” Morgan Kaufmann Publishers, **2003**.

[7] X. Dai, C. K. Li and A. B. Rad, “An approach to tune fuzzy controllers based on reinforcement learning for autonomous vehicle control,” in Proc. IEEE Transactions on Intelligent Transportation Systems, 6 (3) (**2005**) 285-293.

[8] M. Dehyadegari, M. Daneshtalab, M. Ebrahimi, J. Plosila and S. Mohammadi, “An adaptive fuzzy logic-based routing algorithm for networks-on-chip,” in Proceedings of 13th IEEE/NASAESA International Conference on Adaptive Hardware and Systems (AHS), **2011**, pp. 208-214.

[9] S. Ma, N. Enright Jerger and Z. Wang, “DBAR: An efficient routing algorithm to support multiple concurrent applications in networks-on-chip,” in Proc. of ISCA, **2011**, pp. 413-424.

[10] C. J. Glass and L. M. Ni, “The turn model for adaptive routing,” in Proceedings of 19th Annual International Symposium on Computer Architecture, **1992**, pp. 278-287.

[11] M. Ebrahimi, M. Daneshtalab, P. Liljeberg, J. Plosila and H. Tenhunen, “CATRA-congestion aware trapezoid-based routing algorithm for on-chip networks,” in Proc. 15th ACM/IEEE Design, Automation, and Test in Europe (DATE), **2012**, pp. 320-325.

[12] X. Chang, M. Ebrahimi, M. Daneshtalab, T. Westerlund and J. Plosila, “PARS – An efficient congestion-aware routing method for networks-on-

chip," in Proceedings of 16th IEEE International Symposium on Computer Architecture and Digital Systems (CADS), **2012**, pp. 166-171.

[13] G. Ascia, V. Catania, M. Palesi and D. Patti, "Implementation and analysis of a new selection strategy for adaptive routing in networks-on-chip," IEEE Transaction on Computers, 57 (6) (**2008**) 809-820.

[14] M. Koibuchi, H. Matsutani, H. Amano and T. M. Pinkston, "A lightweight fault-tolerant mechanism for network-on-chip," in Proc. 2nd ACM/IEEE Int. Symp. Networks-on-Chip (NOCS), **2008**, pp. 13-22.

[15] P. Lotfi-Kamran, A. M. Rahmani, M. Daneshtalab, A. Afzali-Kusha and Z. Navabi, "EDXY - A low cost congestion-aware routing algorithm for network-on-chips," Journal of Systems Architecture, 56 (7) (**2010**) 256-264.

[16] C. Feng, Z. Lu, A. Jantsch, J. Li and M. Zhang, "A reconfigurable fault-tolerant deflection routing algorithm based on reinforcement learning for network-on-chip," in Proc. of International Workshop on Network on Chip Architectures (NoCArc), **2010**, pp. 11-16.

[17] S. P. Kim and T. Han, "Fault-tolerant adaptive wormhole routing in 2D mesh," in Proc. of IEICE Trans. Inform, **1998**, pp. 1064-1072.

[18] F. Chaix, D. Avresky, N. E. Zergainoh and M. Nicolaidis., "A fault-tolerant deadlock-free adaptive routing for on chip interconnects," in Proc. of DATE, **2011**, pp. 1-4.

[19] M. Valinataj, S. Mohammadi, J. Plosila, P. Liljeberg and H. Tenhunen, "A reconfigurable and adaptive routing method for fault-tolerant mesh-based networks-on-chip," in Proc. International Journal of Electronics and Communications (AEU), 65 (7) (**2011**) 630-640.

[20] M. Ebrahimi, M. Daneshtalab, J. Plosila and H. Tenhunen, "Minimal-path fault-tolerant approach using connection-retaining structure in networks-on-chip," in Proceedings of 7th International Symposium on Networks-on-Chip (NOCS), **2013**, pp. 1-8.

[21] J. Wu and D. Wang, "Fault-tolerant and deadlock-free routing in 2-D meshes using rectilinear-monotone polygonal fault blocks," in Proc. of Parallel Algorithms Appl., **2005**, pp. 99-111.

[22] D. Fick, A. DeOrio, G. Chen, V. Bertacco, D. Sylvester and D. Blaauw. "A highly resilient routing algorithm for fault-tolerant NoCs," in Proceedings of the Conference on Design, Automation and Test in Europe (DATE), **2009**, pp. 21-26.

[23] M. Palesi, S. Kumar and V. Catania, "Leveraging partially faulty links usage for enhancing yield and performance in networks-on-chip," IEEE

Transactions on Computer-Aided Design of Integrated Circuits and Systems, 29 (3) (**2010**) 426-440.

[24] M. Ebrahimi, M. Daneshtalab and J. Plosila, “High performance fault-tolerant routing algorithm for NoC-based many-core systems,” in Proc. of Parallel, Distributed and Network-Based Processing (PDP), **2013**, pp. 462-469.

[25] M. Ebrahimi, M. Daneshtalab, J. Plosila and F. Mehdipour, “MD: Minimal path-based fault-tolerant routing in on-chip networks,” in Proc. of ASP-DAC, **2013**, pp. 35-40.

[26] M. Ebrahimi, M. Daneshtalab, P. Liljeberg, J. Plosila and H. Tenhunen, “LEAR – A low-weight and highly adaptive routing method for distributing congestions in on-chip networks,” in Proc. of 20th IEEE Euromicro Conference on Parallel, Distributed and Network-Based Processing (PDP), Feb. **2012**, pp. 520-525.

[27] C. Wang and N. Bagherzadeh, “Design and evaluation of a high throughput QoS-aware and congestion-aware router architecture for network-on-chip,” in Microprocessors and Microsystems, **2014**, pp. 304-315.

[28] M. Ebrahimi, M. Daneshtalab, F. Farahnakian, P. Liljeberg, J. Plosila, M. Palesi and H. Tenhunen, “HARAQ: Congestion-aware learning model for highly adaptive routing algorithm in on-chip networks,” in Proc. of NOCS, **2012**, pp. 19-26.

[29] C. J. Glass and L. M. Ni, “Maximally fully adaptive routing in 2D meshes,” in Proc. of Parallel Processing, **1992**, pp. 101-104.

[30] C. J. C. H. Watkins and P. Dayan, “Q-learning,” in Proc. Machine Learning, **1992**, pp. 279-292.

[31] M. Majer, C. Bobda, A. Ahmadinia and J. Teich, “Packet routing in dynamically changing networks on chip,” in Proc. IPDPS, **2005**, p. 8.

[32] F. Farahnakian, M. Ebrahimi, M. Daneshtalab, P. Liljeberg and J. Plosila, “Q-learning based congestion-aware routing algorithm for on-chip network,” in Proceedings of 2th IEEE International Conference on Networked Embedded Systems for Enterprise Applications (NESEA) , Australia, Dec. **2011**, pp. 1-7.

[33] M. Radetzki, C. Feng, X. Zhao and A. Jantsch, “Methods for fault tolerance in networks-on-chip,” in Proc. ACM Comput Surv (CSUR), 46 (1) Oct. (**2013**) 1-38.

[34] J. A. Boyan and M. L. Littman, “Packet routing in dynamically changing networks: A reinforcement learning approach,” in Proc. of Advances in Neural Information, **1994**, pp. 671-678.

[35] N. Jiang, D. U. Becker, G. Michelogiannakis, J. Balfour, B. P. Towles, D. E. Shaw, J. Kim and W. J. Dally, “A detailed and flexible cycle-accurate network-on-chip simulator,” in Proc. Int. Symp. Performance Analysis of Systems and Software, **2013**, pp. 86-96.

[36] M. A. Kinsy, S. Khadka and M. Isakov, “PreNoc: Neural network based predictive routing for network-on-chip architectures,” in Proc. of GLSVLSI, **2017**, pp. 65–70.

[37] W. J. Dally and B. P. Towles, “Principles and practices of interconnection networks,” Morgan Kaufmann Publishers, San Francisco, CA, **2004**.

HACETTEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
YÜKSEK LİSANS/~~DOKTORA~~ TEZ ÇALIŞMASI ORJİNALLİK RAPORU

HACETTEPE ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI BAŞKANLIĞI'NA

Tarih: 11/02/2019

Tez Başlığı / Konusu: YONGA-ÜSTÜ-AĞLAR İÇİN HATALARA DAYANIKLI UYARLANABİLEN YÖNLENDİRME ALGORİTMASI TASARIMI

Yukarıda başlığı/konusu gösterilen tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bölümler d) Sonuç kısımlarından oluşan toplam 58 sayfalık kısmına ilişkin, 08/02/2019 tarihinde ~~şahsım~~/tez danışmanım tarafından *Turnitin* adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % 5 'tir.

Uygulanan filtrelemeler:
1- Kaynakça hariç
2- Alıntılar hariç/~~dâhil~~
3- 5 kelimeden daha az örtüşme içeren metin kısımları hariç

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü Tez Çalışması Orjinallik Raporu Alınması ve Kullanılması Uygulama Esasları'nı inceledim ve bu Uygulama Esasları'nda belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini saygılarımla arz ederim.

Tarih ve İmza

11.02.2019

Adı Soyadı: Anıl İPEK
Öğrenci No: N14329624
Anabilim Dalı: Bilgisayar Mühendisliği
Programı:
Statüsü: ☒ Y.Lisans ☐ Doktora ☐ Bütünleşik Dr.

DANIŞMAN ONAYI

UYGUNDUR.

Doç. Dr. Süleyman TOSUN
(Unvan, Ad Soyad, İmza)

ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı	:	Anıl İPEK
Doğum Yeri	:	Alaşehir/MANİSA
Medeni Hali	:	Evli
E-posta	:	anil.ipek@yandex.com
Adres	:	TÜBİTAK BİLGEM - SGE (Siber Güvenlik Enstitüsü), Mustafa Kemal Mah. 2151. Cad. No: 154 Kat: 9 06530 Çankaya/ANKARA

Eğitim

Lise	:	T.C. Ziraat Bankası Fen Lisesi, Balıkesir, 2009
Lisans	:	İTÜ, Telekomünikasyon Mühendisliği, İstanbul, 2013

Yabancı Dil

İngilizce

İş Deneyimi

2013 – 2016	:	Sistem Mühendisi, Empatel Telekom Hizmetleri
2016 – 2017	:	Sistem Mühendisi, Net İletişim
2017 –	:	Araştırmacı, TÜBİTAK BİLGEM – SGE

Deneyim Alanları

TCP/IP, Linux Sistem Yönetimi, IP Santral ve Çağrı Merkezi Projeleri (VoIP), XMPP, Siber Güvenlik, Sızma Testi ve Güvenlik Denetimi