

T.C.
BEYKENT ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**YAZILIM TANIMLI
AĞ KONFEDERASYONU
MİMARİ TASARIMI**

Doktora Tezi

Tezi Hazırlayan:
Hale DÖNERTAŞLI

İstanbul, 2020

T.C.
BEYKENT ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**YAZILIM TANIMLI
AĞ KONFEDERASYONU
MİMARİ TASARIMI**
Doktora Tezi

Tezi Hazırlayan:

Hale DÖNERTAŞLI

Öğrenci No:

140821002

Danışman:

Dr. Öğr. Üyesi Zeynep ALTAN

İstanbul, 2020

YEMİN METNİ

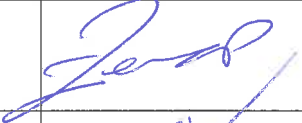
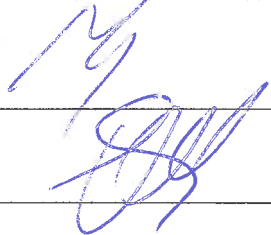
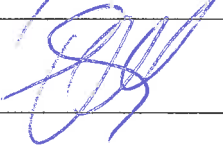


Doktora Tezi olarak sunduđum “Yazılım Tanımlı Ağ Konfederasyonu Mimari Tasarımı” başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmamın içinde kullandıkları her yerde bunlara atıf yapıldığını belirtir ve bunu onurumla dođrularım. 04/02/2020

Aday: Hale DÖNERTAŞLI



TEZ ONAYI

Beykent Üniversitesi Lisansüstü Eğitim Enstitüsü **Bilgisayar Mühendisliği Doktora** öğrencisi **140821002** no'lu **Hale DÖNERTAŞLI**'nin hazırladığı **"Yazılım Tanımlı Ağ Konfederasyonu Mimari Tasarımı"** konulu DOKTORA TEZİ ile ilgili TEZ SAVUNMA SINAVI Lisansüstü Öğretim ve Sınav Yönetmeliğinin ilgili maddesi uyarınca 30.01.2020 günü saat 14.00'da yapılmış, sorulan sorulara alınan cevaplar sonucunda adayın tezinin **KABULU**.....'ne OYBİRLİĞİ/OYÇOKLUĞU'yla karar verilmiştir.

JÜRİ ÜYESİ	KANAATI	İMZA
Dr. Öğr. Üyesi Zeynep ALTAN (Danışman) (Beykent Üniversitesi)	Başarılı	
Doç. Dr. Rüya ŞAMLI (Üye) (İstanbul Üniversitesi-Cerrahpaşa)	Başarılı	
Dr. Öğr. Üyesi Ediz ŞAYKOL (Üye) (Beykent Üniversitesi)	Başarılı	
Doç. Dr. Selçuk SEVGEN (Üye) (İstanbul Üniversitesi-Cerrahpaşa)	Başarılı	
Dr. Öğr. Üyesi Talat FİRLAR (Üye) (Beykent Üniversitesi)	Başarılı	

Adı ve Soyadı : Hale DÖNERTAŞLI
Danışmanı : Dr. Öğr. Üyesi Zeynep ALTAN
Türü ve Tarihi : Doktora, 2020
Alanı : Bilgisayar Mühendisliği
Anahtar Kelimeler : Yazılım Tanımlı Ağ, Ağ İşlevi Sanallaştırması, Bulut Ağ Mimarisi, Tedarikçi Bağımsızlığı, AçıkAkış, Temsilî Durum Aktarımı Uygulama Programlama Arayüzü

ÖZ

YAZILIM TANIMLI AĞ KONFEDERASYONU MİMARİ TASARIMI

Mobil iletişimin evrimi ve günlük yaşamımıza entegrasyonu kullanıcı verilerinin hızla artmasına ve yeni hizmet ihtiyaçlarının ortaya çıkmasına sebep olmuştur. Yazılım Tanımlı Ağ (YTA) ve Ağ İşlevi Sanallaştırması (AİS) olmak üzere iki yeni kavramın ortaya çıkması ağ işlevlerinin gelişimini ve ağ mimarilerinin evrimini kökten değiştirmiştir. Bu kavramlar düşük maliyetli bulut ağ mimarileri tasarlayan sürücüler olarak kabul edilirken ve sistemin uçtan uca çalışması için tedarikçi bağımlılığı yaratmayan, ağ katmanları içi ve arasındaki uyuma dayalı mimari tasarımlara sınırlı olarak dikkat çekilmiştir. Birlikte çalışabilmeyi sağlamak amacıyla açık kaynak kodlu standartlar yayınlanmıştır; fakat standartlar her kullanım durumu için ayrıntılı açıklamalar içermemektedir ve müşteri desteği temin edilmediği için operatörler bu standartları uygulamaya dönüştürerek risk almak istememektedir. Tedarikçiler tarafından geliştirilen YTA mimari çözümleri ticari amaçla tasarlandıkları için yalnızca kendi donanımları ya da ortak oldukları donanımlar ile birlikte çalışmaktadır ve bu mimariler tamamen yada kısmen tedarikçi bağımlıdır. Bu nedenlerle telekomünikasyon şirketleri mimari tasarımlarında kısır döngüye girmektedir. 5N teknolojisinin daha fazla bant genişliği sunmasıyla birlikte Nesnelerin İnterneti (Nİ) kullanıcıları artmış ve bu durum telekomünikasyon şirketlerinin içinde bulunduğu kısır döngünün yarattığı baskıyı daha da arttırmıştır. Bu çalışmada bulut içinde kullanım durumu örneklendirilmiş YTA Konfederasyonu mimarisinin tedarikçi bağımlılığını çözecek katman içi ve katman arasında uyum ile çalışan Temsilî Durum Aktarımı (TEDA) Uygulama Programlama Arayüzleri (UPA) ve AçıkAkış deneyici mesajları sunulmaktadır.

Name ve Surname : Hale DÖNERTAŞLI
Supervisor : Assistant Professor Dr. Zeynep ALTAN
Degree and Date : PhD., 2020
Major : Computer Engineering
Key Words : Software Defined Network, Network Function
Virtualization, Cloud Network Architecture, Vendor Independent, Openflow,
Representational State Transfer Application Programming Interface

ABSTRACT

SOFTWARE DEFINED NETWORK CONFEDERATION ARCHITECTURAL DESIGN

The evolution of mobile communication and its integration into our daily lives has led to a rapid increase in user data and new service innovations. The emergence of two new concepts Software Defined Network (SDN) and Network Function Virtualization (NFV) has radically changed the evolution of network functions and the evolution of network architectures. While these concepts are considered as drivers for designing low-cost cloud network architectures, limited attention has been paid to architectural designs based on the harmony between and within the network layers that do not create vendor dependency for the system to operate end-to-end. Open-source standards have been published to ensure interoperability, but the standards do not contain details for each use case, and operators do not want to take risks by implementing these standards as no customer support is provided. Because SDN architectural solutions developed by suppliers are designed for commercial purposes, they only work with their own hardware or associated equipment and these architectures are wholly or partly supplier dependent. Telecommunication companies are entering a vicious circle in their architectural designs. With 5G Technology offering more bandwidth, users of IoT (Internet of Things) increased and this increased the pressure of the vicious circle of telecommunication companies. In our study the cloud use case which solves the vendor dependence is exemplified, SDN Confederation architecture with Representational State Transfer Application Protocol Interface (REST API) and OpenFlow experimenter messages working in conformity between and within the layers are presented.

İÇİNDEKİLER

	Sayfa No.
ÖZ	i
ABSTRACT	ii
TABLolar LİSTESİ	v
ŞEKİLLER LİSTESİ	vi
KISALTMALAR	vii
GİRİŞ	1
1. LİTERATÜR TARAMASI	5
1.1 Açık Ağ Kurumu YTA Standartları ve İlgili Çalışmaları	5
1.2 Elektrik ve Elektronik Mühendisleri Enstitüsü YTA Çalışmaları	6
1.4 YTA İle İlgili Tez Çalışmaları	10
1.5 Özel Kullanım İçin Geliştirilmiş YTA Mimarileri	11
1.6 Ticari YTA Mimarileri.....	13
2. YTA KONFEDERASYONU	16
2.1. YTA Konfederasyonu Mimarisi	16
2.1.1 Uygulama katmanı.....	20
2.1.2 Kontrol katmanı.....	24
2.1.3 Altyapı katmanı	28
2.2. YTA Konfederasyonu Mimarisi Kullanım Örneği	29
2.2.1 Servis ve uygulamalar	32
2.2.2 Donanım kümeleri	33
2.2.3. Konfigürasyon tabloları.....	35
2.2.4 Trafik akışı ve yönlendirme.....	38
2.3. YTA Konfederasyonu Mimarisi İle Benzeri YTA Mimarilerinin Karşılaştırılması	47
2.3.1 Açık kaynak kod desteği	47
2.3.2 Katmanlar arası uyum.....	48
2.3.3 Katman içi uyum	51
2.3.4 Tedarikçi bağımsızlığı	53

3. YTA KONFEDERASYONU VE BENZERİ ÇÖZÜMLERİN NİTEL DEĞERLENDİRMESİ	58
3.1. Metod	58
3.2. Ölçütler.....	59
3.3. Matris	63
SONUÇ	66
KAYNAKÇA	68
EKLER	76



TABLolar LİSTESİ

	Sayfa No.
Tablo 1: YTA Konfederasyonu Bulut Varsayımları.....	35
Tablo 2: Orkestrasyon Servis Zinciri Tanımı	36
Tablo 3: Orkestrasyon İlk Adım Konfigürasyonları.....	36
Tablo 4: Orkestrasyon ve Denetleyici Yönlendirme Bilgileri	37
Tablo 5: Konfederasyon Denetleyicisindeki Kümeler.....	38
Tablo 6: Küme Denetleyicisindeki Anahtarlar	38
Tablo 7: YTA Mimarileri Değerlendirme Özeti	59
Tablo 8: YTA Mimarileri Değerlendirme Matrisi	64

ŞEKİLLER LİSTESİ

	Sayfa No.
Şekil 1: YTA Konfederasyonu Mimarisi	17
Şekil 2: Telekomünikasyon Şirketlerinin Yaşadığı Kısır Döngü	18
Şekil 3: Klasik SFZ Yapısı	19
Şekil 4: YTA Konfederasyonu SFZ Yapısı	19
Şekil 5: Veri Merkezleri İçin YTA Konfederasyonu Bulutu	31
Şekil 6: YTA Mimarisi Trafik Akış Diyagramı	40
Şekil 7: Nokia Nuage YTA Mimarisi	49
Şekil 8: Cisco UMA YTA mimarisi	50
Şekil 9: Big Switch Networks BBA YTA Mimarisi	50
Şekil 10: Juniper Contrail YTA Mimarisi	52
Şekil 11: Google YTA Mimarisi	53
Şekil 12: Cisco UMA SGYAA Mimarisi	54
Şekil 13: Juniper Contrail Kapsülleme Mimarisi	55
Şekil 14: Amazon NetBond YTA Mimarisi	56
Şekil 15: Amazon Verizon YTA Mimarisi	56
Şekil 16: Facebook Altoona YTA Mimarisi	57
Şekil 17: YTA Mimari Değerlendirme Anket Katılımcıları	58

KISALTMALAR

- 3NOP - 3. Nesil Ortaklık Projesi (3. Generation Partnership Project - 3GPP)
- 5N - 5. Nesil Kablosuz Teknoloji (5. Generation Wireless Technology - 5G)
- 6N - 6. Nesil Kablosuz Teknoloji (6. Generation Wireless Technology - 6G)
- AçıkAkış (OpenFlow)
- AçıkYığın (OpenStack)
- AçıkGünIşığı (OpenDayLight)
- AA - Akıllı Ağlar (Intelligent Networks - IN)
- AAK - Açık Ağ Kurumu (Open Networking Foundation - ONF)
- AA Lab - Açık Ağ Laboratuvar (Open Networking Lab - ON Lab)
- AAV - Açık Akış Yolu (Open Flow Path - OFP)
- AFS - Ağ Fonksiyon Sanallaştırması (Network Function Virtualization - NFV)
- AGE - AçıkAkış Genişletilebilir Eşleme (OpenFlow Extensible Match - OXM)
- AEBİB - Amazon Elastik Bilgi İşlem Bulutu (Amazon Elastic Compute Cloud - EC2)
- AH - Akış Hedefleri (Flow Objectives - FO)
- AİS - Ağ İşletim Sistemi (Network Operating System - NOS)
- AKİŞ - Açık Kaynak Ağ İşletim Sistemi (Open Source Network Operating System - ONOS)
- ASA - Açık Sistem Arabağlantısı (Open Systems Interconnection - OSI)
- ASAS - Ara Sistemden Ara Sisteme (Intermediate System to Intermediate System - ISIS)
- ASB - Ağ Servis Başlığı (Network Service Header - NSH)
- ASGYK - Ağ Sanallaştırma Genel Yönlendirme Kapsülleme (Network Virtualization Generic Routing Encapsulation - NVGRE)
- ATSE - Avrupa Telekomünikasyon Standartları Enstitüsü (European Telecommunications Standards Institute - ETSI)
- AVMB - Açık Veri Merkezi Birliği (Open Data Center Alliance - ODCA)

BBA - Büyük Bulut Ağı (Big Cloud Fabric - BCF)

ÇPED - Çok Protokollü Etiket Değişirme (Multi Protocol Label Switching - MPLS)

DABYP - Dinamik Ana Bilgisayar Yapılandırma Protokolü (Dynamic Host Configuration Protocol - DHCP)

DAP - Dosya Aktarım Protokolü (File Transfer Protocol - FTP)

EEME - Elektrik ve Elektronik Mühendisleri Enstitüsü (Institute of Electrical and Electronics Engineers - IEEE)

GAA - Geniş Alan Ağı (Wide Area Network - WAN)

GBF - Geniş Bant Forumu (Broadband Forum - BBF)

GDA - Güneye Doğru Arayüz (South Bound Interface - SBI)

GKA - Güvenli Kabuk (Secure Shell - SSH)

İAGC - İnternet Araştırma Görev Gücü (Internet Research Task Force - IRTF)

İAYÖ - İlk Açık Yöne Öncelik (Open Shortest Path First - OSPF)

İDP - İletim Denetim Protokolü (Transmission Control Protocol - TCP)

İDP/İP - İletim Denetim Protokolü/İnternet Protokolü (Transmission Control Protocol/Internet Protocol - TCP/IP)

İDS - İçerik Dağıtım Sistemi (Content Distribution System - CDN)

İMGC - İnternet Mühendisliği Görev Gücü (Internet Engineering Task Force - IETF)

İP - İnternet Protokolü (Internet Protokolü - IP)

İSAGP - İç Sınır Ağ Geçidi Protokolü (Interior Border Gateway Protocol - IBGP)

KAP - Kapsayan Ağaç Protokolü (Spanning Tree Protocol - STP)

KDA - Kuzeye Doğru Arayüz (North Bound Interface - NBI)

KDP - Kullanıcı Datagram Protokolü (User Datagram Protocol - UDP)

KREA - Kablosuz Radyo Erişim Ağı (Wireless Radio Access Network - WRAN)

KYAA - Kablosuz Yerel Alan Ağı (Wireless Local Area Network - WLAN)

MEF - Metro Ethernet Forumu (Metro Ethernet Forum - MEF)

MEK - Medya Erişim Kontrolü (Media Access Control - MAC)

MOYV - Veri Merkezi Olarak Yeniden Tasarlanan Merkezi Ofis (Central

Office Re-architected as a Datacenter - CORD)

MÜAP - Metin Üstü Aktarım Protokolü (Hypertext Transfer Protocol - HTTP)

Nİ - Nesnelerin İnterneti (Internet of Things - IoT)

OAAF - Optik Ağlar Arası Forumu (Optical Internetworking Forum - OIF)

OTÜ - Orijinal Tasarım Üreticileri (Original Design Manufacturers - ODM)

RÜA - Raf Üstü Anahtar (Top of Rack - ToR)

SAİ - Sanal Ağ İşlevleri (Virtual Network Functions - VNF)

SGO - Standart Geliştiren Organizasyon (Standard Developing Organization - SDO)

SGP - Sınır Geçit Protokolü (Border Gateway Protocol - BGP)

SFZ - Servis Fonksiyonu Zincirleme (Service Function Chaining - SFC)

SGYAA - Sanal Genişletilebilir YAA (Virtual Extensible LAN - VXLAN)

SHD - Sanallaştırılmış Hizmetler Denetleyicisi (Virtualized Services Controller - VSC)

SHP - Sanallaştırılmış Hizmetler Platformu (Virtualized Services Platform - VSP)

SHR - Sanallaştırılmış Hizmetler Rehberi (Virtualized Services Directory - VSD)

SÖA - Sanal Özel Ağ (Virtual Private Network - VPN)

SÖB - Sanal Özel Bulut (Virtual Private Cloud - VPC)

SYA - Sanal Yönlendirme ve Anahtarlama (Virtual Routing and Switching - VRS)

SYAA - Sanal YAA (Virtual LAN - VLAN)

TEDA - Temsilî Durum Aktarımı (Representational State Transfer - REST)

TEÇİ - Telekomünikasyon Endüstrisi Çözümleri İttifakı (Alliance for Telecommunication Industry Solutions - ATIS)

TTŞ - Tablo Tipi Şablonlar (Table Type Patterns - TTP)

TUD - Tip Uzunluk Değer (Type Length Value - TLV)

UMA - Uygulama Merkezli Altyapı (Application Centric Infrastructure - ACI)

UPA - Uygulama Programlama Arayüzü (Application Programming Interface - API)

UTB - Uluslararası Telekomünikasyon Birliği (International Telecommunication Union - ITU)

ÜİAB - Üçlü İçerik Adreslenebilir Bellek (Ternary Content Addressable Memory - TCAM)

YAA - Yerel Alan Ağı (Local Area Network - LAN)

Yİ - Yorum İsteği (Request for Comments - RFC)

YNSYPA - Yeni Nesil Servis Yer Paylaşım Ağı (Next Generation Service Overlay Network - NGSON)

YTA - Yazılım Tanımlı Ağlar (Software Defined Networks - SDN)

YTAAG - Yazılım Tanımlı Ağlar Araştırma Grubu (Software Defined Networks Research Group - SDNRG)



GİRİŞ

Son on yılda benimsenen iki büyük teknoloji sanallaştırma ve bulut bilgi işlem bilgi teknolojisi üzerinde çok büyük bir etki yaratmıştır. Sanallaştırma bir işletim sistemi, bir sunucu, bir depolama aygıtı veya ağ kaynaklarının gerçek sürümü yerine sanal versiyonun yaratılmasıdır. Bulut bilgi işlem şirketlerin bilgisayar hesaplama gücünü, depolama veya bir uygulamayı servis olarak kendi sanal altyapısını gibi bulut üzerinden kullanabilmesidir. Bu iki teknoloji ağ mühendislerine ve uygulama mimarlarına kendi donanımlarından maksimum verimi almalarını sağlamıştır. Bununla birlikte zaman içerisinde uygulamaların kullandığı temel veri ağı sanallaştırma ve bulut bilgi işlemeye göre daha az gelişmiştir. Bant genişliği her geçen gün artarken ağ uygulamalarının kullanımı da artmıştır, fakat ağ kurulumunun bu artışa nasıl uyarlanacağı kurgulanmamıştır. Ağ oluşturma alanındaki destek eksikliği uygulama tasarımcılarına otomasyon kısıtlamaları getirmiştir. Bunun üzerine ağ cihazları üreticileri servis sağlayıcılarının ve işletmelerin ihtiyaç duyduğu esneklik ve kullanım kolaylığını artıracak çözümler geliştirmiştir. Bu çözümlerin amacı merkezi kontrol, programlanabilirlik, orkestrasyon ve sanallaştırma ihtiyaçlarına cevap vermektir. Yazılım Tanımlı Ağlar (YTA) kavramı bu şekilde doğmuştur.

YTA bir protokol, protokol dizisi veya standartlaşmış bir teknoloji değildir; bir ağ mimarisidir. YTA şu anda mevcut olan ağ mimarisine farklı bir yaklaşım ile daha esnek, daha yönetilebilir ve daha iyi gözlemlenebilir yeni bir yapı ortaya koymayı hedeflemektedir. Bunu ağ katmanlarını birbirinden ayırarak gerçekleştirmektedir. YTA merkezi bir denetleyicide çalışan yazılımlarla ağın etkin bir şekilde programlanmasını sağlamaktadır. Ağ anahtarlarının ve yönlendiricilerinin tablolarını lokal olarak programlamaktadır. Bu da ağ aygıtlarının trafiği nasıl yöneteceği konusunda kendi kararlarını vermeleri anlamına gelmektedir. Trafik iletme kararları Kapsayan Ağaç Protokolü (KAP), İlk Açık Yöne Öncelik (İAYÖ) ve Sınır Geçit Protokolü (SGP) gibi kontrol platform protokolleri tarafından bildirilmektedir. Ancak bu geleneksel ağ protokollerinin esnekliği sınırlıdır. Bu protokollerin çalışabilmesi için yönlendirme alanına katılan tüm ağ cihazlarının protokol standardı ile aynı

kurallara uyması gerekmektedir. Bu durum ağ uygulamaları için kısıtlama getirmektedir.

Farklı donanımları ve çözümleri birbirleriyle çalıştırabilmek için YTA mimarisinde yukarıdan aşağıya ve aşağıdan yukarıya olmak üzere farklı stratejiler geliştirilmiştir. Altyapıda performans, desteklenen özellikler ve arayüzlerin tedarikçiler için çeşitlilik göstermesinin yanı sıra üst katmanda da aynı şekilde orkestrasyon ve servisler için gereken performans ihtiyacı ve desteklenen arayüzler operatörler için farklılık göstermektedir. Bu nedenle her altyapı ve servis ihtiyacına cevap verebilen denetleyici geliştirmek bu konuda standart geliştirmede olduğu gibi oldukça zordur. Alt katmanda ve üst katmanda çok maliyetli değişiklikler yapmaya gerek duymadan sadece denetleyiciye ek özellik getirerek farklı katmanların birbirini anlayabilecekleri bir noktada buluşturmak mümkündür. Bunu Açık Ağ Kurumu (AAK) ve benzeri organizasyonlar YTA mimarisi için hem yukarıdan aşağıya hem de aşağıdan yukarıya sunduğu standartlar ile çözmeye çalışmıştır, fakat bu çözümler aşağıda listelenen nedenler yüzünden hedeflenen beklentileri karşılamamıştır:

- Yukarıdan aşağıya izlenen yaklaşım: Servisler için geliştirilen arayüzler anahtar mimarisi ve kapasitesinden bağımsız, altyapı seviyesindeki ayrıntıları içermeyen genel temsili modeller şeklinde tasarlanmıştır. Örnek olarak AçıkAkış (OpenFlow) protokolü Üçlü İçerik Adreslenebilir Bellek'i (ÜİAB) kullanmaktadır, fakat anahtarın geri kalan mimarisini kullanmamaktadır. Trafik akışlarının karakterleri önceden bilinmediği için, optimum çözümler üretebilmek çok güçtür. Bunu ancak Google, Amazon ve Facebook gibi şirketler kendi özel ağ trafiği için donanım üreterek sağlamaktadır.

- Aşağıdan yukarıya izlenen yaklaşım: Bu yaklaşımda anahtarlar denetleyiciye mimarisi hakkındaki ayrıntıları göndermektedir. Bunun için her anahtara ait yönlendirme mimarisinin denetleyici tarafından anlaşılması gerekmektedir. Fakat her farklı anahtar için, yönlendirme mimarisini ortak bir dilde anlatmak imkansızdır. AçıkAkış protokolünün önerdiği çoklu tablo kullanımı için ÜİAB'ye ihtiyaç vardır. Yapılan anahtar performans testlerine göre Pica8 P-3290 yazılım ve donanım trafik akış tablolarının senkronize

olamaması bu duruma örnektir. ÜİAB çok maliyetli olduğu için bu işlem yığınlar halinde yapılmaktadır ve yığın içindeki komutların sıralı olduğu garanti edilememektedir. Buna ek olarak, eğer anahtarın işlemcisi meşgul ise denetleyici ile iletişimi donmakta ve komutları kaldığı yerden devam alamamaktadır.

YTA ve Ağ Fonksiyon Sanallaştırması (AFS) tabanlı mimarilere talep artarken AAK, Avrupa Telekomünikasyon Standartları Enstitüsü (ATSE), Elektrik ve Elektronik Mühendisleri Enstitüsü (EEME), İnternet Mühendisliği Görev Gücü (İMGC), Uluslararası Telekomünikasyon Birliği (UTB), 3. Nesil Ortaklık Projesi (3NOP), AçıkYığın (OpenStack) ve AçıkGünIşığı gibi organizasyonlar birbiriyle neredeyse aynı standartları geliştirmeye başlamışlardır. Fakat ağ operatörleri açısından mimarinin baştan sona çalışabilmesi için eksiklikler vardır. Bu kapsamda açık kaynak kod organizasyonları tarafından geliştirilen çözümlerin YTA mimarisinin uygulama ve altyapı katmanları ile bağlantılı arayüzleri olmadığı için operatörler bu konuda destek alamamıştır. Benzeri amaçla Google, Facebook ve Amazon tarafından tasarlanan YTA tabanlı mimariler sadece bu üç büyük firmanın kendi altyapılarına uyumlu geliştirilmiştir ve ayrıntılar gizli tutulmaktadır. Bunun yanı sıra Cisco Uygulama Merkezli Altyapı (UMA), Jupiter Contrail, Big Switch Networks Big Cloud Fabric ve Nokia Nauge gibi tedarikçiler talebe kısmen cevap vermektedir. Tedarikçiler tarafından geliştirilen YTA mimari çözümleri ticari amaçla tasarlandıkları için yalnızca kendi donanımları ya da ortak oldukları donanımlar ile birlikte çalışmaktadır ve bu mimariler tamamen yada kısmen tedarikçi bağımlıdır. Bu nedenlerle telekomünikasyon şirketleri mimari tasarımlarında kısır döngüye girmektedir. YTA Konfederasyonu mimarisi bu kısır döngüyü kırmak amacıyla tasarlanmıştır.

Uygulamada karşılaşılan en kritik mimari eksiklik, altyapı ve servisleri bulut ağlarda ağ dilimleri formunda dinamik olarak bütünleştirememektir. Literatürde buna en yakın çalışma İMGC tarafından yayınlanan Servis Fonksiyonu Zincirleme (SFZ) ve AAK standardı olan Tablo Tipi Şablonlar (TTŞ) standartlarıdır. Fakat bu standartlar tek başına bulut ağlarda modüler olarak servis, uygulama entegrasyonunu, 5. Nesil

Kablosuz Teknoloji'nin (5N) doğası gereği ihtiyaç duyduğu ağ dilimleri içerisinde uçtan uca çalıştırmaya yeterli değildir. YTA Konfederasyonu mimarisinde

AçıkAkış deneyici mesajları kullanılarak, denetleyiciye TTŞ ve SFZ'nin birbirleriyle konuşmasını sağlayan yeni bir yetenek eklenmiştir. Farklı donanımların kümelendiği, orkestrasyondan gelen bilgilerin konfederasyon denetleyicisi tarafından işlendiği YTA Konfederasyonu mimarisinin var olan çözümlere uçtan uca çalışabilmesi için katkıları; uygulama katmanını sisteme dahil eden SFZ için Temsilî Durum Aktarımı (TEDA) Uygulama Programlama Arayüzü (UPA) kodlaması, altyapı katmanını sisteme dahil eden TTŞ kodlaması ve bu yeni yetenekleri denetleyiciye kazandıran AçıkAkış deneyici mesajlarıdır. Tasarlanan mimari en alt katmandaki cihazları böl ve yönet kavramı ile kümelere ayırdıktan sonra arayüzler arasındaki iletişimi kapsülleme tekniği ile birbirine bağlayarak farklı donanımların uygulama katmanında çalışan servisler için uyumluluğunu ve ağ yönetim kolaylığını artırabilmeyi hedeflemektedir.

Bu tez çalışmasında YTA Konfederasyonu mimarisinin değerlendirilmesi için anket sistemi oluşturulmuştur. Bu sisteme bilgisayar ağları disiplininde deneyim sahibi olan, mimari tasarımcısı, yazılım mühendisi, test mühendisi ve operasyon mühendisi katılmıştır. Tasarlanan yeni YTA Konfederasyonu mimarisi, benzerlik teşkil eden Google'un Jupiter, Facebook'un Altoona, Amazon'un Amazon Elastik Bilgi İşlem Bulutu (AEBİB), Cisco'nun UMA, Nokia'nın Nuage, Juniper'in Contrail ve Big Switch Networks'un Big Cloud Fabric YTA mimarileri ile karşılaştırılmıştır. Bu mimarilerin her biri 14 farklı ölçüte göre anket yolu ile nitel olarak değerlendirilmiştir. Bu ölçütler Protokol Kapsamı, Kullanılabilirlik, Servis Oluşturma, Uygulama Oluşturma, AFS Desteği, Servis ve Uygulama Entegrasyonu, Kolay Yapılandırma ve Sağlama, Modülerlik, Böl ve Yönet Mekanizması, Kapsülleme Yöntemi, Farklı Donanım Mimarilerinin Kümelenmesi, Tedarikçi Bağımsızlığı, Trafik Akışı Esnekliği, Bulut Entegrasyonu, YTA Olmayan Bağlantı Desteği olarak belirlenmiştir.

Çalışmanın Giriş Bölümünden sonra, 1. Bölümde Literatür Tarama yapılmaktadır. 2. Bölümde ise YTA Konfederasyonu mimarisinin katmanları, kullanım örneği ve benzeri mimariler ile değerlendirilmektedir. Bölüm 3 YTA Konfederasyonu mimarisinin nitel olarak anket yoluyla karşılaştırılması ve sonuçların özet olarak sunulduğu değerlendirme matrisi ile devam etmektedir. Çalışma Sonuç Bölümü ile tamamlanmaktadır.

1. LİTERATÜR TARAMASI

1.1 Açık Ağ Kurumu YTA Standartları ve İlgili Çalışmaları

AAK birlikte çalışabilirliği artırabilmek için üç farklı formatta standartlar yayınlamaktadır. Bunlar teknik özellikler, protokol tanımları ve modellemeler olarak sınıflandırılmıştır. AAK standartları teknik tavsiyeler ve uygulama arayüzlerini içermektedir. AAK tarafından tanımlanmış standartlara YTA Konfederasyonu çalışmasında geniş ölçüde yer verilmiştir. Özellikle açık veri yolu projesi kısmında bulunan TTŞ çalışmada kullanılmıştır¹. İlerleyen bölümlerde TTŞ'lerin tek başına yetersiz olduğu alanlar irdelenmiş ve çözüm önerileri sunulmuştur. AAK'nin yayınladığı bu çalışmayla ilgili standartlar ve haberler aşağıda özetlenmiştir:

- Standart oluşturmak ve bunu yayınlanabilir hale getirmek uzun zaman isteyen bir süreçtir ve bir standart yayınlanana kadar YTA kullanıcıları artan talepler doğrultusunda yeni bir uygulamaya geçmektedir. Buna ek olarak standartlar çoğunluğa hitap etmek amacıyla yazılmıştır ve operatörler için pratikte kullanılabilir ayrıntıya sahip değildir. Telekomünikasyon şirketlerinin standartların yanı sıra talep gören kullanım durumları için de mimari tasarım olarak çözümlere ihtiyacı vardır. YTA Konfederasyonu çalışması bu doğrultuda mimari tasarım olarak oluşturulmuştur.
- Birçok Standart Geliştiren Organizasyonunun (SGO) YTA konusunda aynı probleme aynı çözümü üreten çalışması olduğu gibi uygulamada karşılaşılan bir çok ihtiyaca henüz çözüm üretilmemiştir². Çalışmada bu noktalar gözden geçirilmiş ve çözümler örneklendirilerek mimari tasarım olarak sunulmuştur.
- AA Lab. açık kaynak kod geliştirdiği için değişen taleplere bir ölçüde daha hızlı cevap vermektedir. Operatörler, ağ cihazlarını satan şirketlerin tek eline düşmemek için açık kaynak kodlu projelere ilgi göstermektedir, fakat bu

¹ ONF, *OpenFlow Table Type Patterns*, <https://www.opennetworking.org/wp-content/uploads/2013/04/OpenFlow%20Table%20Type%20Patterns%20v1.0.pdf>, Erişim Tarihi: 07/12/2018

² Joel Halpern, (December 2014), *Standards collisions around SDN*, IEEE Communications Magazine, IEEE, 2014, 10-15

durum kendilerine özgü çıkan sorunlar için müşteri hizmeti konusunda zorluk yaratmaktadır³. YTA Konfederasyonu mimarisi tamamen açık kaynak kodu kullanan arayüzler ile tasarlanmıştır ve tedarikçi bağımlılığı yaratan noktalara çözüm üretilmiştir.

- YTA mimarisini kullanmaya başlayan operatörler yazılım ağırlıklı bir mimariyle karşılaşınca sık ve merkezi güncellemeler karşısında sıkça trafik kesintisi yüzünden zorluk yaşamaktadır. YTA öncesi ağlarda bu kadar çok güncelleme görülmemektedir. Bu problemi çözmek amacı ile YTA Konfederasyonu mimarisi modüler olarak tasarlanmıştır.

- YTA mimarisi ağ teknolojisinde yeni bir çağ yaratmıştır ve benzer eğilimler telekomünikasyon alanında çıkan omurga ve yaprak mimarisinde de görülmektedir. YTA mimarisinin geleceğini tahmin etmek için referans olmaktadır.

1.2 Elektrik ve Elektronik Mühendisleri Enstitüsü YTA Çalışmaları

EEME tarafından YTA konusunda yapılmış en ayrıntılı çalışma kapsamlı ankettir⁴. AAK, ATSE, Açık Veri Merkezi Birliği (AVMB), EEME, Geniş Bant Forumu (GBF), İMGC, İnternet Araştırma Görev Gücü (İAGC), Metro Ethernet Forumu (MEF), Optik Ağlar Arası Forumu (OAAF), Telekomünikasyon Endüstrisi Çözümleri İttifakı (TEÇİ) ve UTB organizasyonlarının YTA ile ilgili çalışmaları listelenmiştir. Bunlardan AAK'nın yaptığı çalışmalar YTA Konfederasyonu mimarisine en yakın ayrıntıları içermektedir ve bu benzerlikler bir önceki bölümde irdelenmiştir.

YTA Konfederasyonu mimarisi çalışmasıyla ilgili EEME' nin yayınladığı bir diğer çalışma da YTA için ikinci nesil bulut ağlarının standartlaşmasıdır. YTA ağ merkezleşmesini ve sanallaştırmasını bulut yapısına taşıyarak bulut yapısının ikinci

³ Dan Meyer, *Standards remain ongoing open source challenge for operators*, <https://www.rcrwireless.com/20170125/carriers/standards-remain-ongoing-open-source-challenge-for-operators-tag2-tag99>, Erişim Tarihi: 09/02/2019

⁴ Kreutz et al., (20 May 1998), *Software-Defined Networking: A Comprehensive Survey*, MELECON '98. 9th Mediterranean Electrotechnical Conference, IEEE, 2002

neslini yaratmıştır. Bu neslin standartlaştırılması konusu aşağıdaki beş kategori altında incelenmiştir:

- YTA ile AçıkAkış Evrimi ve Standartlara bakış: Bu yayın AçıkAkış protokolünün AAK tarafından nasıl tanımlandığını anlatmaktadır. Bir standardın oluşabilmesi için sırayla dört aşama gerçekleşmektedir: öneri evresi, kuluçka evresi, prototip evresi ve AAK çapında inceleme. Standartlar için en çok bulut ve servis kalitesi uygulamalarına yer verilmiştir. YTA Konfederasyonu mimarisi çalışmasında bulut kullanım durumu ele alınmıştır, ek olarak AçıkAkış protokolünün 2014 yılından sonraki sürümleri için ayrıntılara yer verilmiştir⁵.

- Teknoloji ve Endüstri Odağını Açık Kaynak Standart Testleriyle Hizalama: AAK test ve birlikte çalışabilirlik grubu tekil kullanıcı girdilerinin teknolojiyi tek yönlü geliştirdiğine şahit olduktan sonra adil olarak test yazılımları gerçekleştirmek için tüm üyelere bilgi olarak ortak bir test yazılımı yaratmak için çalışmışlardır. Uyumluluk ve birlikte çalışabilirlik testlerine ait raporlara çalışmada yer verilmiştir. Farklı karakterdeki anahtarları birbirleriyle çalıştırmak yerine YTA Konfederasyonu mimarisi çalışmasında kümeleme yöntemi kullanılmıştır. Bu yöntem mimariyi daha modüler ve tedarikçiden bağımsız hale getirmiştir⁶.

- Servis Fonksiyon Zinciri: Ağ Servis Başlıkları ile Servis Düzlemi Oluşturma : Ağ servislerinin kapsülleme yöntemiyle mimariyi oluşturan farklı altyapı teknolojilerinden bağımsız olarak çalışabilmesi için geliştirilmiş bir uygulamadır. YTA Konfederasyonu mimarisi çalışmasında makaledeki Ağ Servis Başlığı'nın (ASB) çalışma prensibine benzer kapsülleme yöntemi izlenmiştir. Buna ek olarak birlikte çalışabilirliği sağlamak için SFS ve TTŞ entegrasyonu yapılmıştır⁷.

⁵ Tourrilhes et al., (24 November 2014), *SDN and OpenFlow Evolution: A Standards Perspective*, Computer, IEEE, 2014, 22 - 29

⁶ Rick Bauer, Ron Milford, Li Zhen, (24 November 2014), *Aligning Technology and Market Drivers in an Open Source Standards Testing Program*, Computer, IEEE, 2014, 30-36

⁷ Paul Quinn, Jim Guichard, (24 November 2014), *Service Function Chaining: Creating a Service Plane via Network Service Headers*, Computer, IEEE, 2014, 38-44

- Açık Kaynak Ağ Kontrol Düzlemiyle Buluştuğunda: YTA ağ fonksiyonlarını daha çok yazılıma doğru yönlendirdiği için, açık kaynak kod üreten komiteler bu konuda uygulama arayüzleri yazmışlardır. Açık kaynak kodun getirdiği esneklik hızlı çözümler üretmeyi sağlamış, fakat standartlaştırma kontrolünü de azaltmıştır. Geliştirilen UPA'lar farklı mimarilere göre farklı şekilde birleşerek çalışmaktadır ve YTA Konfederasyonu mimarisi çalışmasında da farklı UPA'ların nasıl birlikte çalıştığı örneklerle anlatılmıştır⁸.

- Hücreleme ile Artıklı YTA Kurulumu: Geleneksel bir ağ mimarisinden YTA mimarisine geçilmek istendiğinde bir anda bütün ağı, YTA kabiliyetli cihazlarla değiştirmenin mümkün olduğu durumlar vardır. Bu yazıda, hibrid ağlarla ilgili metodlar anlatılmaktadır. YTA kapasiteli ve diğer cihazlar ait oldukları farklı KAP'lara atanarak denetleyiciye bildirim yapılmaktadır. Bu yazının sonunda bu yöntemin henüz geniş bir kitle tarafından kabul edilen bir standart olmadığına, fakat buna aday olduğuna dikkat çekilmektedir. Bu yöntem klasik YTA mimarileri için iyi bir metoddur, fakat servis akışını aksatmaktadır. YTA Konfederasyonu mimarisi çalışmasında mimari kümelere bölüdüğü ve uygulama katmanı kontrol katmanından konfederasyon denetleyicisi ile ayrıldığı için bu yöntemde ihtiyaç yoktur⁹.

1.3 ScienceDirect YTA Makaleleri

YTA Mimari Standardizasyon: AAK YTA ile ilgili dağıtım gereksinimleri, farklı ağ türleri, arayüzler, bilgi modelleri, AçıkAkış protokol standardı, uyum ve sertifikasyon konularında çalışmaktadır. AAK standartları geliştirir ve diğer standart geliştiren organizasyonlarla işbirliği yaparak en çok talep gören ağ uygulamalarını destekler. AFS mimarisi YTA tabanlı çalışan destekleyici sanallaştırma mimarisidir . Burada gösterilen mimaride ikiden fazla denetleyici hiyerarşisi vardır ve bu yapı 5N

⁸ Rothenberg et al., (24 November 2014), *When Open Source Meets Network Control Planes*, *Computer*, IEEE, 2014, 46-54

⁹ Canini et al., (24 November 2014), *Software-Defined Networks: Incremental Deployment with Panopticon*, *Computer*, IEEE, 2014, 56-60

gibi teknolojiler için dilimlenebilir değildir. YTA Konfederasyonu mimarisi çalışmasında farklı görevlere sahip denetleyici hiyerarşisi ikinin üstünde değildir. Böylece fonksiyonellik artarken ölçeklenebilirlik azalmamaktadır¹⁰.

Ağ Servisi Orkestrasyon Standardizasyonu: Bu makale YTA mimarisinde Kuzeye Doğru Arayüz (KDA) standartlarına yoğunlaşmaktadır. KDA standartları düşük seviye ve yüksek seviye olmak üzere ikiye ayrılmaktadır. Düşük seviye standardı Güneye Doğru Arayüz'lerin (GDA) heterojenliğini homojen birimler haline getirmiştir: ağ bağlantı bilgileri ve teknolojiye ait özellikler tanımlanmaktadır, fakat bu parametreler kısıtlıdır ve Ağ İşletim Sistemi (AİS) için özel modeller geliştirilmektedir. YTA Konfederasyonu mimarisi çalışmasında bu çalışmadan farklı olarak orkestrasyon farklı karakterlere sahip anahtar kümeleriyle ilgili bilgiye sahiptir. Böylece hibrid ağ kullanımı mümkün hale gelmektedir¹¹.

Nesnelerin İnterneti Altyapısı için Bulut Tabanlı YTA ve AFS Mimarileri: Bu makalede YTA ve AFS mimarileri içinde kullanılan 4G teknolojilerinin donanım ve enerji maliyet tasarrufları nicel olarak değerlendirilmiştir. Makalenin odak noktası bulut yapısı içindeki klasik YTA mimarisinin altyapı katmanıdır. YTA Konfederasyonu mimarisi çalışmasında enerji tasarrufu konusundan ziyade katmanlar arası uyum incelenmiştir¹².

AFS senaryosunda YTA tabanlı SFZ mekanizması ve servis uygulaması: Bu makale AçıkAkış tabanlı SFZ mekanizmasının tasarımı ve bulut AFS servis uygulanmasını açıklanmaktadır. YTA Konfederasyonu mimarisi çalışmasında bu çalışmadan farklı olarak uygulama katmanındaki TEDA UPA'lar şeklinde tanımlanan

¹⁰ Sibylle Schaller, Dave Hood, (November 2017), *Software defined networking architecture standardization*, Computer Standards & Interfaces, ScienceDirect, 2017, 197-202

¹¹ Rotsos et al., (November 2017) *Network service orchestration standardization: A technology survey*, Computer Standards & Interfaces, ScienceDirect, 2017, 203-215

¹² Mamdouh Alenezia, Khaled Almustafa, Khalim Amjad Meerj, (March 2019), *Cloud based SDN and NFV architectures for IoT infrastructure*, Egyptian Informatics Journal, ScienceDirect, 2019, 1-10

SFZ elemanları altyapı katmanında bulunan farklı anahtarlara ait TTŞ'ler ile eşleştirilmiştir¹³.

YTA ve AFS kullanarak 5N ağ dilimleme: Sınıflandırma, mimariler ve gelecekteki zorluklarla ilgili bir anket: Bu makalede YTA ve AFS mimarileri kullanarak 5N ağ dilimleme çözümleri incelenmiş ve birbirleriyle karşılaştırılmıştır. Ağ dilimleme YTA ve AFS ile aynı aileden gelen bir tür sanal ağ mimarisidir ve paylaşılan fiziksel altyapı üzerinde birden fazla sanal ağ oluşturulmasına olanak tanımaktadır. YTA Konfederasyonu mimarisi çalışmasında altyapı katmanındaki kümelerin oluşturduğu konfederasyon yapısı ile 5N ağ dilimleme kavramı benzerlik göstermektedir, aradaki fark; ağ dilimleme kavramı sadece yatay dilimlemeyi desteklerken, konfederasyon yapısı hem yatay hem de dikey dilimlemeyi mümkün kılmaktadır¹⁴.

1.4 YTA İle İlgili Tez Çalışmaları

YTA Mimarilerinde Birlikte Çalışabilirlik: Vishal Mishra tarafından yazılmış olan lisans tezi YTA mimarisinin temel tanımını ve avantajlarını incelemektedir. Daha önceden yapılmış olan çalışmalara ait olan YTA mimarisinin güvenlik ve performans değerlerini içermektedir¹⁵.

Uygulama Gereksinimine Göre YTA Destekli Bant Genişliği ve Gecikmeye Dayalı Rota Optimizasyonu: Pongsakorn Chupala tarafından yazılmış yüksek lisans tezi YTA mimarilerinin performansını artırmaya yönelik yönlendirme optimizasyonlarını incelemektedir¹⁶.

¹³ Trajkovska et al., (November 2017), *SDN-based service function chaining mechanism and service prototype implementation in NFV scenario*, Computer Standards & Interfaces, ScienceDirect, 2017, 247-265

¹⁴ Barakabitze et al., (11 February 2020), *5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges*, Computer Networks, ScienceDirect, 2020

¹⁵ Vishal Mishra, *Interoperability in Software Defined Networking*, Lisans Tezi, Department of Electronics and Communication Engineering National Institute of Technology, Rourkela, 2015

¹⁶ Pongsakorn Chupala, *Overseer: SDN-Assisted Bandwidth and Latency Aware Route Optimization based on Application Requirement*, Yüksek Lisans Tezi, Nara Institute of Science and Technology, 2015

YTA ve AFS ile Servis Olarak Ağ Mimarisi: M.P.V. Manthena tarafından yazılmış yüksek lisans tezinde servis sağlayıcılarının YTA ve AFS tabanlı bulut ağları için servis olarak ağ uygulaması önerilmiştir¹⁷.

YTA Bulut Platformlarının Entegrasyonu: Açık Kaynak Yaklaşım: Francesco Foresta tarafından yazılmış yüksek lisans tezinde AçıkYığın ile bulut ağ oluşturma yetenekleri incelenmiştir¹⁸.

Ağ İşlevi Sanallaştırmasına Yeni Bir Yaklaşım: Aurojit Panda tarafından yazılmış doktora tezinde tip kontrolü ve resmi doğrulama gibi statik tekniklerin kullanımına dayalı alternatif AIS mimarisi sunulmuştur¹⁹.

1.5 Özel Kullanım İçin Geliştirilmiş YTA Mimarileri

Google'ın veri merkezi ağı dünyanın en büyüklerinden birini temsil etmektedir ve YTA mimarilerinin en gelişmiş şekli Google tarafından kullanılmaktadır. Google'ın altyapısının muazzam büyüme hızı, veri merkezi ağına YTA mimarisine yönelmenin ana motivasyonu olmuş ve Jüpiter doğmuştur. Yeni nesil YTA veri merkezi olan Jüpiter, mevcut en büyük fabriğin 6 katından daha büyük bir ölçektir. Önceki mimarilerden farklı olarak, yeni ağ teknolojisinin aşamalı olarak kurulması için bir gereksinimler belirlenmiştir. Oldukça büyük ölçekten dolayı, ağdaki hem planlanmış hem de planlanmamış olayların yönetimi eskisine göre daha zor olacaktır. Yaprak omurga yapısını kullanan Jupiter mimarisinde yaprak ve omurga arasında orta blok bulunmaktadır. Google, geleneksel ağ mimarisi büyük ölçekli gereksinimleri karşılarken, performans gereksinimi ve maliyetle artmıştır. Sunucu başına bant genişliği ortalama 100Mbps ile sınırlı kalırken ve karşılanamayan trafik çoğalmıştır. Veri merkezlerinde konuşlandırılan tüm omurga ve yaprak mimarileri, bu mimarin

¹⁷ M.P.V. Manthena, *Network-as-a-Service Architecture with SDN and NFV A Proposed Evolutionary Approach for Service Provider Networks*, Yüksek Lisans Tezi, Delft University of Technology, 2015

¹⁸ Francesco Foresta, *Integration of SDN Framework and Cloud Computing Platforms: An Open Source Approach*, Yüksek Lisans Tezi, Bologna University, 2017

¹⁹ Aurojit Panda, *A New Approach to Network Function Virtualization*, Doktora Tezi, Berkeley University, 2017

sorunlu varyantlarına çözüm bulmak amacıyla geliştirilmiştir. Mevcut ticari çözümler ölçek, yönetim ve maliyet gereksinimlerini karşılayamazken, sunucu başına artan bant genişliği, sunucu başına düşen maliyeti ve küme ölçeğini önemli ölçüde artırmaya devam etmiştir. Bu nedenle kendi anahtarlarını üretmiştir. Google, kontrol düzlemini nasıl oluşturacağı konusunda, mevcut İAYÖ, Ara Sistemden Ara Sisteme (ASAS), SGP, vb. protokolleri kullanan kendine özgü bir yazılım geliştirmiştir. Jupiter mimarisi Google'un özel ihtiyaçlarına göre geliştirilmiştir, bulut yapısıyla ilgili olan bölümünün çoğu statiktir ve başka bir ağda kullanılmak üzere uyumlu değildir²⁰. Google YTA altyapısını tedarikçilerden bağımsız olabilmek için geliştirmiştir. Bununla birlikte harici ağ katmanları hala pahalı ve kısıtlı tedarikçi ekipmanlarını kullanmaktadır.

Facebook'u sürekli olarak kullanan 1.35 milyardan fazla kişi kesintisiz her zaman açık bir site performansına güvenmektedir. Arka planda gerçek zamanlı bir deneyimi mümkün kılan birçok alt sisteme sahip ağ bunu mümkün kılmaktadır. Facebook'un kullandığı omurga ve yaprak mimarisinin YTA mimarisiyle birleşen özel bir versiyonu Altoona artan talepleri karşılayabilmek için yaratılmıştır²¹. Facebook Altoona mimarisini var olan protokolleri kendine özgü altyapısı için değiştirerek kullanmıştır. Facebook Altoona mimarisi ile operasyon ve değişiklikleri hızlandırmak için gereken yapılandırmaları aygıtlara otomatik olarak dağıtmak ve ağda yeni aygıtların rollerini keşfetmek için basit ve sağlam mekanizmalar kurmuştur. Facebook ağının büyümesini kısıtlayan tedarikçi bağımlılığının önüne geçebilmek için Altoona YTA mimarisi geliştirilmiştir.

Amazon Web Servisleri (AWS) dünyanın en kapsamlı ve geniş kabul görmüş bulut platformudur ve küresel olarak veri merkezlerine YTA altyapılı servis sunmaktadır. En hızlı büyüyen girişimler en büyük şirketler ve önde gelen devlet

²⁰ Singh et al., *Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network*, <https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43837.pdf>, Erişim Tarihi: 04/08/2018

²¹ Alexey Andreyev, *Introducing data center fabric, the next-generation Facebook data center network*, <https://engineering.fb.com/production-engineering/introducing-data-center-fabric-the-next-generation-facebook-data-center-network/>, Erişim Tarihi: 04/08/2018

kurumları dahil milyonlarca müşteri altyapılarını güçlendirmek, daha çevik olmak ve maliyetleri düşürmek için AWS kullanmaktadır. Amazon sanal bulut kullanımını kullanıcılarına özelleştirilebilir şekilde sunmaktadır. Bunun için de AWS talep üzerine ağ kaynaklarını hızlı bir şekilde sağlamak için gerekli olan YTA için kendi ağ donanımını geliştirmiştir. Bu müşterilerinin uygulamaları AWS' ye taşınması ve veri merkezlerinin kapasitesini artırmasına olanak tanımaktadır²². Amazon ağ alt yapısının en üst katmanında bulunan İçerik Dağıtım Sistemi (İDS) aracılığıyla çeşitli ağ ve güvenlik hizmetleri sunmaktadır. Bu servisler her kullanıcıya özel şekliyle AEBİB sanal makinelerinde çalışmaktadır. AWS müşteri sitelerine bağlı birden fazla Sanal Özel Bulut (SÖB) dağıtımını desteklemektedir. Eğer şirketler hibrit bir bulut kurmayı ve iş yüklerini genel ve özel bulutların arasında güvenli bir şekilde taşımayı düşünüyorlarsa, Amazon'un hizmetlerinden faydalanabilmektedirler. AWS' de çalışan özel bir bulut standart bilgi işlem platformunda sanal olarak çalışan, dinamik servis yerleştirmesine izin veren elastik bir platformdur. Bu kapsamda katman 4 ila 7 ağ servisleri trafik akışları ile dinamik olarak eşleşmektedir ve Sanal Genişletilebilir YAA (SGYAA) gibi kapsülleme sistemleri Sanal Özel Ağ'lar (SÖA) arası bağlantılar için tercih edilen bir yaklaşımdır.

1.6 Ticari YTA Mimarileri

Cisco UMA günümüzün geleneksel ağlarının gereksinimlerini ve uygulama faktörlerinin ağa getirdiği talepleri karşılamak için tasarlanmış yeni bir veri merkezi mimarisidir²³. Cisco UMA fiziksel ya da sanal bağlantı uçlarının, Uç Nokta Grupları'nda (EPG) birleştirilerek, bağlantının tanımlandığı uygulama merkezli politikayı kullanmaktadır. Bağlantı son kullanıcı bir EPG ile diğeri arasında bir sözleşme ilişkisi belirlediğinde tanımlanmaktadır. Cisco UMA'nın ana gücü açık kaynak kodlu UPA'ları ve açık OpFlex protokolünü kullanmasıdır.

²² AWS, *AWS Well-Architected Framework*, https://d1.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf, Erişim Tarihi: 04/08/2019

²³ Cisco, *Getting Started with Cisco Application Centric Infrastructure (ACI) in the Small-to-Midsize Commercial Data Center Cisco Validated Design*, <https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/guide-c07-733638.pdf>, Erişim Tarihi: 04/08/2018

Nokia'nın bir giriřimi olan Nuage Networks veri merkezi ađ altyapısını sanallařtıran ve bilgi iřlem kaynaklarını birbirine bađlayan YTA ürünüdür. Nuage Networks Sanallařtırılmıř Hizmetler Platformu (SHP), çok kullanıcı bulut veri merkezi için eksiksiz ađ servisleri çözümdür ve üç temel iřlevden oluřmaktadır. Bunlar Sanallařtırılmıř Hizmetler Rehberi (SHR), Sanallařtırılmıř Hizmetler Denetleyicisi (SHD), Sanal Yönlendirme ve Anahtarlamadır (SYA). Nuage Networks SHP servisler düzlemini, kontrol düzlemini ve veri merkezi ađının iletim düzlemini birleřtirmektedir. Bunu politikalar, uygulama tercihleri ve řebeke servis gereksinimleri hakkında BT merkezli bir bakıř açısı kullanarak yapmaktadır. Nuage Networks SHP, uygulama ve ađ UPA'ları aracılıđıyla programlanabilirlik sađlamakta ve yalın metal anahtarları da dahil olmak üzere mevcut altyapıyı sanallařtırıp otomatik hale getirmektedir²⁴.

Juniper Networks'ün Contrail denetleyicisi, yüksek ölçeklenebilirliđe sahip sanal ađların oluřturulmasını düzenlemek için YTA mimarisini kullanan açık bulut ađ bileřenidir. Juniper Contrail denetleyicisi bulutu genel, özel ve hibrit sanal ađ iřlevlerini servis zincirleme ile otomatik řekilde çalıřır hale getirmektedir²⁵.

Big Switch Networks'un Büyük Bulut Ađı (BBA) YTA mimarisi hibrit bulut dađıtımlarını düzene sokmak için ortak bulut ađını benimseyen bir mimari olarak türünün ilk örneđidir. Uygulamaların nerede bulunduđuna bakılmaksızın iřletmelerin ortak ađ iletiřimine ihtiyaçları vardır. Geleneksel üç katmanlı model eski model ađlarda kullanılmak üzere tasarlanmıř ve genellikle sanal sunucular gibi aygıtların yerini kısıtlayan bölmelere bölünmüřtür. BBA YTA mimarisi çeřitli yönlendiriciler ve omurga yaprak mimarisine dayanan fiziksel anahtarlardan oluřmaktadır²⁶. BBA YTA mimarisi AçıkAkıř protokolünün özelleřtirilmıř bir versiyonunu kullanmaktadır. BBA denetleyicisi uygulama odaklı tek bir programatik arayüz aracılıđıyla çeřitli bulut

²⁴ Nuage Networks, *Unconstrained Datacenter Networks For The Cloud Era*, <https://onestore.nokia.com/asset/165438>, Eriřim Tarihi: 04/01/2020

²⁵ Juniper Networks, *Contrail Architecture*, <https://www.juniper.net/us/en/local/pdf/whitepapers/2000535-en.pdf>, Eriřim Tarihi: 04/01/2020

²⁶ Big Switch Networks, *Big Cloud Fabric Cloud-First Networking Delivers Self-Service Automated Fabric with VPCs ON-PREM*, https://www.bigswitch.com/sites/default/files/datasheet_-_bcf-ec-v5.2.pdf, Eriřim Tarihi: 04/01/2020

yönetim platformları ile entegrasyonu desteklemektedir. BBA harici bağlantılar için SGP protokolünü desteklemektedir. BBA denetleyicisi ağ içi bağlantıları, omurga ve yaprak ağında SGP kullanımını ve ana yönlendiriciye olan bağlantıları sağlamaktadır.



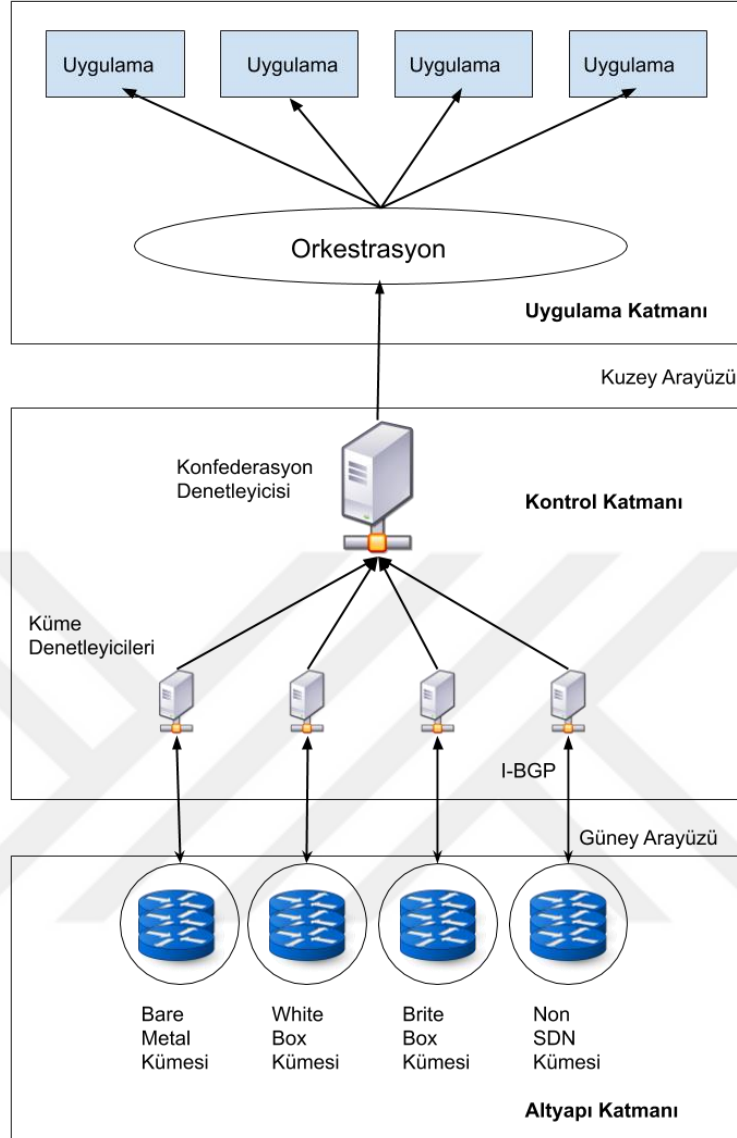
2. YTA KONFEDERASYONU

2.1. YTA Konfederasyonu Mimarisi

Tasarlanan YTA Konfederasyonu mimarisi klasik YTA ve benzeri YTA mimarilerinde olduğu gibi uygulama, kontrol ve altyapı katmanlarından oluşmaktadır²⁷. Şekil 1’de görüldüğü gibi YTA Konfederasyonu mimarisinin klasik YTA mimarisi ve benzeri diğer mimarilere göre en önemli temel üstünlüğü, altyapı katmanında kümeler içinde gruplanmış olan farklı özelliklere sahip donanımları uygulama katmanındaki servisler ile entegre etmesidir. Bu entegrasyon klasik YTA mimarilerinin sahip olduğu kısıtlamalara çözüm bulmak amacıyla yapılmıştır. Klasik YTA mimarisi kontrol katmanına odaklanmıştır ve bu katmanın yukarıda uygulama katmanı ve aşağıda altyapı katmanı ile uyumlu çalışması için gerekli olan arayüzleri kapsamamaktadır²⁸. Bu çalışmanın klasik YTA mimarisinden bir diğer fiziksel farkı da orkestrasyon ve küme denetleyicisi arasında servis ve uygulama yönetim kolaylığını sağlamak amacıyla tasarlanan konfederasyon denetleyicisinin işlevidir. YTA Konfederasyonu mimarisi fiziksel farklılıklarının yanı sıra, protokol kullanımı ve uygulamaları konusunda da klasik YTA ve benzeri çözüm mimarilerinden ayrılmaktadır.

²⁷ ONF, *Software-Defined Networking: The New Norm for Networks*, <https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf>, Erişim Tarihi: 05/03/2017

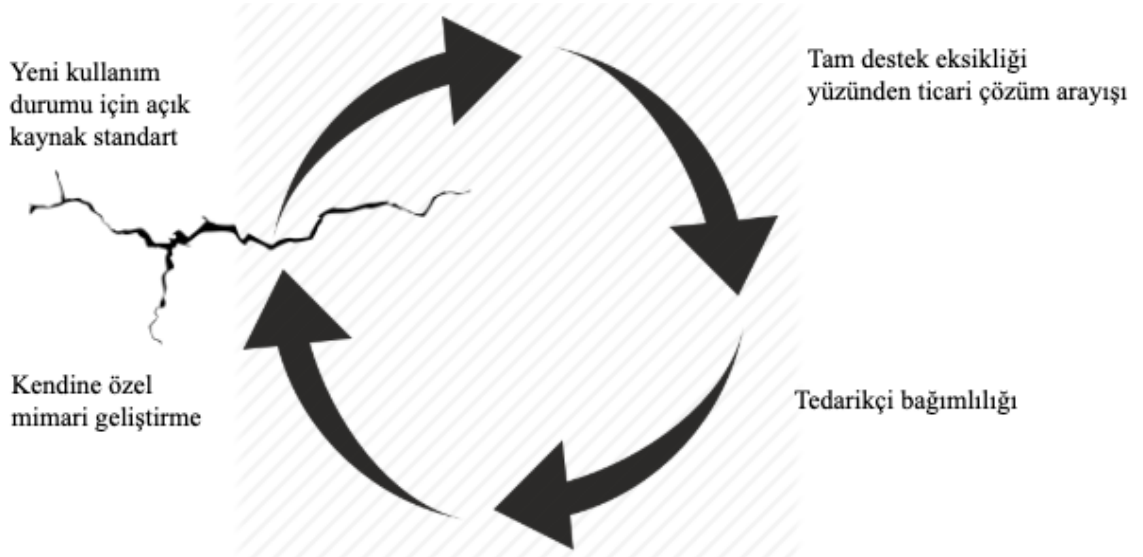
²⁸ Sven van der Meer, Eduard Grasa, *SDN Architectural Limitations: Towards a Full Software Network Vision*, IEEE Softwarization, IEEE, 2016



Şekil 1: YTA Konfederasyonu Mimarisi

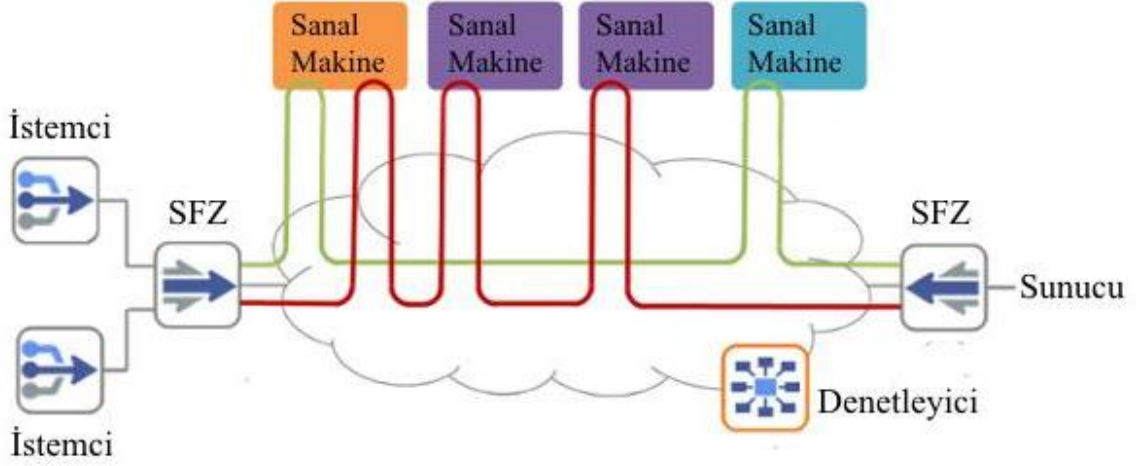
Standart geliştiren organizasyonlar en yeni kullanım durumlarını açık kaynak destekli standartlar şeklinde yayınlamaktadırlar. Bu standartlar telekomünikasyon şirketlerinin ihtiyacını baştan uca tam çözemediği gibi müşteri hizmetinden de yoksundur. Bu nedenle yeni kullanım durumu desteği için telekomünikasyon şirketleri ticari çözüm arayışına gitmektedirler fakat, bu durum tedarikçi bağımlılığı gibi yeni bir problem ile birlikte gelmektedir. Çünkü ticari YTA mimarisi sunan üreticiler pazar payını kaybetmemek için kısmen ya da tam olarak tedarikçi bağımlılığı yaratan mimariler üretmektedirler. Bunun sonucunda YTA kullanıcıları Google, Facebook ve

Amazon gibi kendi özel mimarilerini üretmektedirler. Bu kısır döngü Şekil 2’de görüldüğü üzere yeni bir kullanım durumunun ortaya çıkışına kadar devam eder. YTA Konfederasyonu mimarisi bu kısır döngüyü kırmak amacıyla tasarlanmıştır.



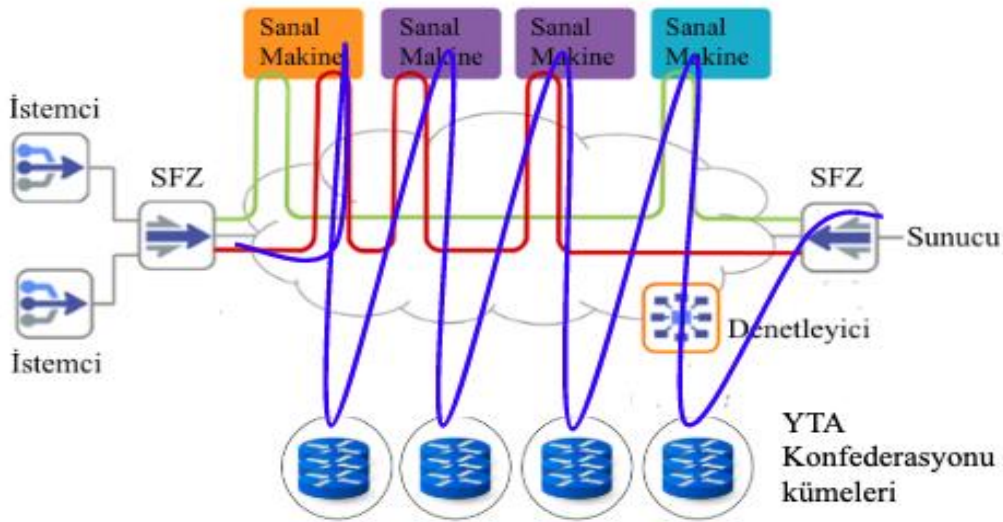
Şekil 2: Telekomünikasyon Şirketlerinin Yaşadığı Kısır Döngü

Literatürdeki benzer çözümler AAK standardı olan TTŞ ve EEME standardı olan SFZ’dir. TTŞ yapısı donanım bağımsızlığını desteklemek için AAK ve diğer YTA öncüleri tarafından geliştirilmiştir. Bir kontrol profili veya AçıkAkış şablonu gibi işlev gören TTŞ mantıksal yönlendirme hattını AçıkAkış terimleriyle tanımlamaktadır fakat, TTŞ yapısı yalnızca altyapı katmanında çalışmaktadır ve bu kısıtlı alan özellikle YTA mimarilerinin bulut içinde kullanımını sınırlamaktadır. SFZ ağ servislerinin kapsülleme yöntemiyle mimariyi oluşturan farklı altyapı teknolojilerinden bağımsız olarak çalışabilmesi için geliştirilmiş bir uygulamadır fakat, bu yapı Şekil 3’te görüldüğü üzere yalnızca uygulama katmanında çalışmaktadır ve yine YTA mimarilerinin bulut içinde kullanımını kısıtlamaktadır.



Şekil 3: Klasik SFZ Yapısı

YTA Konfederasyonu mimarisinde AçıkAkış deneyci mesajları kullanılarak, denetleyiciye TTŞ ve SFZ'nin birbirleriyle konuşmasını sağlayan yeni bir yetenek eklenmiştir. Farklı donanımların kümeleniği, orkestrasyondan gelen bilgilerin konfederasyon denetleyicisi tarafından işlendiği YTA Konfederasyonu mimarisinin var olan çözümlere sistemin Şekil 4'te görüldüğü üzere uçtan uca çalışabilmesi için katkıları; uygulama katmanını sisteme dahil eden SFZ için TEDA UPA kodlaması, altyapı katmanını sisteme dahil eden TTŞ kodlaması ve bu yeni yetenekleri denetleyiciye kazandıran AçıkAkış deneyci mesajlarıdır.



Şekil 4: YTA Konfederasyonu SFZ Yapısı

Bu bölümde YTA Konfederasyonu arayüzleri ile oturum, sunum, taşıma ve ağ katmanlarında çalışan servis fonksiyonları tanımlanmış ve protokol kullanımı farklılıkları ayrıntılı bir şekilde irdelenmiştir. Açık kaynak olarak geliştirilen tüm arayüz fonksiyonları AçıkAkış protokolü ve Temsilî Durum Aktarımı (TEDA) Uygulama Programlama Arayüzü (UPA) tarafından yönlendirilmek üzere işlenmektedir²⁹. Bu mesajların içerikleri ayrıntılı bir şekilde alt bölümlerde gösterilmiştir. YTA Konfederasyonu mimarisinin uygulama, kontrol ve altyapı katmanları arasında AçıkAkış iletisi özet olarak üst katmandan alt katmana aşağıdaki gibi taşınmaktadır:

- Orkestrasyon kaynaktan gelen trafik ile akış eşleştirmesi yaparak fonksiyon zinciri kimliğini tanımlamaktadır. Bu bilgiyi sanal makinelere ve konfederasyon denetleyicisine iletmektedir.
- Konfederasyon denetleyicisi orkestrasyondan gelen bilgileri işleyerek ilgili küme denetleyicileri iletişime geçerek yönlendirme bilgilerini gönderir.
- Küme denetleyicisi konfederasyon denetleyicisinden gelen yönlendirme bilgilerine göre kümedeki anahtarları programlar.

2.1.1 Uygulama katmanı

Uygulama katmanında servis fonksiyonlarının çalıştığı sanal makinalar ve orkestrasyon bulunmaktadır. Orkestrasyon kuzey arayüzünden sanal makinalara, güney arayüzünden de kontrol katmanında bulunan konfederasyon kontrolcüsüne bağlıdır. AFS YTA ile birlikte çalışır ve ağ fonksiyonlarının sanallaşmasını sağlamaktadır³⁰. YTA Açık Sistem Arabağlantısı (ASA) ağ ve veri bağlantısı katmanında, AFS ise taşıma, oturum, sunum ve uygulama katmanında çalışmaktadır³¹. YTA ve AFS arasında birçok standart vardır, fakat literatür tarama kısmında

²⁹ McKeown et al., (March 2008), *OpenFlow: enabling innovation in campus networks*, ACM SIGCOMM Computer Communication Review, ACM, 2008

³⁰ Hawilo et al., (December 2014), *NFV: State of the Art, Challenges and Implementation in Next Generation Mobile Networks (vEPC)*, IEEE Network, IEEE, 2014, 18-26

³¹ Bora et al., *OSI Reference Model: An Overview*, International Journal of Computer Trends and Technology, 2014

irdelendiği üzere bu standartlar her sistemin baştan sona çalışabileceğine dair ayrıntıları kapsamamaktadır.

Uçtan uca hizmetlerin sunulması genellikle çeşitli ağ servis işlevleri gerektirmektedir. Bunlar güvenlik duvarları gibi geleneksel ağ servisi işlevlerinin yanı sıra uygulamaya özgü işlevleri de içermektedir. Sıralı servis fonksiyonları setinin tanımı, somutlaştırılması ve bunların içinden geçen trafiğin yönlendirilmesi SFZ olarak adlandırılır³². YTA Konfederasyonu mimarisinde uygulama katmanında bulunan orkestrasyon ağ servislerinin gereksinimleri trafik sınıflarına ve Servis Fonksiyon Zinciri'ne (SFZ) dönüştürmektedir. Bunu yaparken de servisler için başlangıç noktası, bitiş noktası, servis işlem sırasını ve politika kısıtlamalarını içeren fonksiyon zinciri gereksinimlerini ayarlanmaktadır. Geleneksel ağlarda İnternet trafiği bir güvenlik duvarı üzerinden yönlendirilmektedir. SFZ farklı olarak trafiği sanal bloklara bölerek yönlendirmektedir. Eğer bir paket belirli bir şart serisiyle eşleşirse, belirli fonksiyon zincirinden geçmekte ve yönlendirme YTA bulutunda gerçekleşmektedir. YTA Konfederasyonu mimarisinde aşağıdaki SFZ elemanları kullanılmıştır:

- Servis fonksiyonu: Güvenlik duvarı, yük dengeleyici gibi fonksiyonlardır.
- Akış sınıflandırıcı: Bir akışın servis fonksiyon zincirine girip girmeyeceğine karar vermektedir.
- Arayüz: Ağ cihazlarının servis fonksiyonla eşleşen arayüzlerini tanımlamaktadır.
- Arayüz zinciri: Akış sınıflandırıcı ve arayüz ikilisinden oluşan bir dizidir.

Trafik akışı sınıflandırılmak üzere ilk olarak orkestrasyona yönlendirilir. Ağ operatörü bir kaynak ile bir hedef arasındaki trafik akışına uygulanması gereken farklı karakterdeki sıralı servis fonksiyonlarının gereksinimlerini belirlemektedir. Orkestrasyon bu gereksinimleri almakta ve konfederasyon denetleyicisine göndermek üzere TEDA UPA'ya çevirmektedir. TEDA UPA aşağıdaki birimleri içermektedir:

³² Joel Halpern, Carlos Pignataro, *Service Function Chaining (SFC) Architecture*, <https://tools.ietf.org/html/rfc7665>, Erişim Tarihi: 05/11/2019

- Servis fonksiyon zinciri: Bir kaynak ve bir hedef arasındaki trafiğe hangi servis fonksiyonu işlem sırasının uygulanması gerektiğini belirtmektedir. Kaynak formülü, hedef formülü ve bir operasyondan oluşmaktadır.
- Uç nokta formülleri: Boole operatörlerinden oluşan bir ifadeyi ve işlenenleri oluşturmaktadır. Kaynak ve hedef uç nokta tanımlayıcıları ile birlikte kullanıldığında bir trafik akışını temsil etmektedir.
- Uç nokta tanımlayıcıları: Uç noktayı kısmen tanımlamak için kullanılan bir tür ve değer vardır. Son nokta tanımlayıcı kimliği bitiş noktasının cebir ifadesinde bir terim olarak kullanılmaktadır. Bitiş noktası tanımlayıcısının tipi İP adresi veya kullanıcı tanımlı tip gibi temel bir ağ tamamlayıcısıdır.
- Operasyon: Kaynak bitiş noktası ve hedef bitiş noktası ile eşleşen trafiğe uygulanan işlemi belirtmektedir.
- Trafik akışı sınıflandırma kuralları: Her bir akışa ait uygulanması gereken servis dizisini içermektedir.

Orkestrasyon önceden yapılan konfigürasyonlara göre paket için gerekli olan servis fonksiyon zincirinin tanımını bulur. Orkestrasyon servis zinciri içindeki yönlendirme bilgisini kuzey arayüzünden sanal makinelere TEDA UPA biçiminde aşağıdaki temsili gösterimde olduğu gibi paylaşır:

```

"request": {
  "path_information": {
    "kind": "service_function_chain",
    "flow_type": <flow_type>,
    "path_id": <path_id>,
    "path_index": <path_index>,
  }
  "hop_information": {
    "type": "VM",
    "name": <name>,
  },
  "next_hop_information": {
    "type": <type>,
    "name": <name>,
  }
}

```

Orkestrasyon servis zinciri içindeki yönlendirme bilgisini güney arayüzünden konfederasyon kontrolcüsü ile TEDA UPA biçiminde aşağıdaki örnek gösterimde olduğu gibi paylaşır:

```
"request": {
  "path_information": {
    "kind": "service_function_chain",
    "flow_type": <flow_type>,
    "path_id": <path_id>,
    "path_index": <path_index>,
  }
  "hop_information": {
    "type": "Cluster",
    "name": <name>,
  },
  "next_hop_information": {
    "type": <type>,
    "name": <name>,
  }
}
```

Yukarıda görülen TEDA UPA kodlarında talep (request) orkestrasyondan, sanal makinelere gönderilen taleptir. Bu talep sanal makinenin sunucudan istemciye giden trafiğin yolculuğunda nerede olmasını gerektiren komuttur. Bu komuttaki yolculuğun bütün durakları servis fonksiyon zinciri (service_function_chain) tarafından tanımlanmaktadır. Bu bir yolculuk tipidir ve bunun gibi başka formata ve kurallara sahip yolculuk tipi tanımlanmaktadır. Bir ağda farklı trafik karakterleri için tanımlanmış servis fonksiyon zinciri yapılandırılmaktadır. Farklı servis fonksiyonlarını ayırt etmek için yol kimliği (path_id) ve hangi akış için hizmet ettiklerini bildirmek için de akış kimliği (flow_id) kullanılmaktadır. Bu iki kimliğin birleşmesiyle birlikte hangi trafik akışının hangi servis fonksiyon zincirinde yola çıktığı belirlenmiş olmaktadır. Bu mesaj orkestrasyondan sanal makineye gönderilen bir talep olduğu için, sanal makinenin servis fonksiyonu içinde hangi sırada görev alacağı da yol indeksi (path_index) aracılığıyla tanımlanmaktadır. Tüm bu parametreler yol bilgisi (path_information) yapısıyla orkestrasyondan sanal makinelere gönderilmekte ve böylece her sanal makine ya da küme hizmet sırasının ne zaman kendine geleceğini bu yapı sayesinde öğrenmektedir. Bir sanal makine ya da küme gelen talebin kendisine ait olduğunu adım bilgisinden (hop_information)

bilmektedir. Eğer tip (type) ve ad (name) kendisiyle eşleşiyorsa bu talep kendisi tarafından işlenmektedir. Talep işlendikten sonra iki seçenek vardır; ya trafik hedefe ulaşmıştır ya da trafik bir sonraki adıma yönlendirilmektedir. Eğer bir sonraki adım bilgisi (next_hop_information) boş (NULL) ise trafik hedefe ulaştırılmakta, eğer bir sonraki adım bilgisi başka bir sanal makine yada kümeye ait bilgi içeriyorsa trafik o adıma yönlendirilmektedir. Bu döngü sonraki adım bilgisi NULL olana kadar devam etmektedir.

2.1.2 Kontrol katmanı

Kontrol katmanında konfederasyon denetleyicisi ve küme denetleyicileri bulunmaktadır. Konfederasyon denetleyicisi ve küme denetleyicileri birbirleri ile AçıkAkış anahtar protokolünü kullanarak iletişim kurmaktadır. AçıkAkış anahtar protokolü açık kaynak kodludur ve YTA mimarisinde bulunan bağlantıları ve akımları kontrol eden ara birimdir. Konfederasyon denetleyicisi orkestrasyondan gelen talepleri ağ mimarisine bağlı yollara dönüştürerek servislerin kısıtlama yaşamadan bir uçtan diğer uca ulaştırılmasını sağlamaktadır. Konfederasyon denetleyicisi, orkestrasyon ve küme denetleyicisinden aldığı bilgileri birleştirerek zincir başlığı kapsülleme ve taşıma yeteneklerini kodlamaktadır.

AçıkAkış YTA mimarisi programcılarının yeni ağ teknolojilerini ve davranışlarını tanımlama mekanizmaları sağladığı için bu tez çalışmasında tercih edilmiştir. Geleneksel YTA mimarilerine ek yetenekler eklemek için AçıkAkış Deneyci mesajları (experimenter) mesajları kullanılmıştır³³. Deneyci alanlarına ait akış eşleştirme değerleri `oxm_class = OFP_XMC_EXPERIMENTER` sabiti kullanılarak `oxm_header` aracılığı ile aşağıdaki kodda gösterildiği gibi tanımlanmaktadır. YTA Konfederasyonunun eklediği kapsül bilgisinden sonra `sizeof(struct ofp_oxm_experimenter_header)` komutu ile yeni protokol başlığı hesaplanmakta ve küme denetleyicisi tarafından veri paketlerine eklenmektedir. Eğer AçıkAkış 1.5 versiyonu kullanılıyorsa ek protokol bilgileri oturum ya da sunum

³³ Nikolai A. Snow, Venkat R. Dasari, Billy E. Geerhart, (September 2018), *OpenFlow Experimenter Labels for Encoding Adaptive Network Functions*, IEEE 39th Sarnoff Symposium, IEEE, 2018

katmanına yerleştirilmektedir³⁴. AçıkAkış protokolünün daha üst versiyonları için ek protokol bilgileri oturum, sunum, taşıma ya da ağ katmanlarından birine yerleştirilmektedir.

```
struct ofp_oxm_experimenter_header {
    uint32_t oxm_header;
    uint32_t experimenter;
};
OFP_ASSERT(sizeof(struct ofp_oxm_experimenter_header) ==
8);
```

AçıkAkış protokolünün 1.4 versiyonu, uygulamaların kaynak ve hedefe ait İDP veya Kullanıcı Datagram Protokolü (KDP) port numaralarına göre dinamik olarak sınıflandırılmasını desteklemektedir³⁵. AçıkAkış protokolünün YTA Konfederasyonu mimarisine ait değişiklikleri destekleyebilmesi için İDP ve KDP başlık alanlarını aşağıdaki gibi işlemesi gerekmektedir. Böylece küme denetleyicileri konfederasyon denetleyicisinden aldığı talimata göre YTA düzlemindeki anahtarları programlamaktadır. Her bir YTA yönlendirme düzlemi ögesi gelen trafiği oturum katmanındaki bağlantı noktalarına kadar sınıflandırır ve küme denetleyicisi tarafından programlanan yönlendirme tablosuna dayanarak yönlendirme kararları almaktadır.

Kaynaktan gelen trafik akışının eşleştirilmesi için paket içindeki yol kimliği aşağıdaki AçıkAkış deneyci kodunda görüldüğü gibi (path_id) ve yol indeksi (path_index) alanları kullanılmaktadır. Eşleştirme operasyonları struct ofp_ofb_experimenter_sfch_match yapısıyla gerçekleştirilmektedir.

³⁴ ONF, *OpenFlow Switch Specification Version 1.5.1*, <https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf>, Erişim Tarihi: 07/11/2019

³⁵ ONF, *OpenFlow Switch Specification Version 1.4.0*, <https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.4.0.pdf>, Erişim Tarihi: 07/11/2019

```

struct ofp_oxm_experimenter_sfch_path_id {
    uint32_t operation;
    uint32_t path_id;
};

struct ofp_oxm_experimenter_sfch_path_index {
    uint32_t operation;
    Uint8_t path_index;
};

struct ofp_action_experimenter_sfch {
    uint32_t operation;
    uint32_t path_id;
    uint8_t path_index;
    uint8_t metadata[];
};
OFP_ASSERT(sizeof(struct
ofp_action_experimenter_header)==8);

struct oxm_ofb_experimenter_sfch_match {
    OXM_OF_SFCH_PATH_ID,
    OXM_OF_SFCH_PATH_INDEX,
};

```

Deneyci mesaj bilgilerini AçıkAkış protokolü içerisinde tanımlamak için TUD kodlama şeması kullanılmaktadır. TUD veri iletişim protokollerinde isteğe bağlı bilgi ögesi için kullanılan bir kodlama şemasıdır ve aşağıdaki AçıkAkış kodunda görülen ofp_action_experimenter_header yapısının içinde tip (type), uzunluk (length), değer (value) bilgileri ile tanımlanmıştır³⁶. AGE sınıfı TUD kodlamasının ilk dört baytı deneyciye ait tanımlayıcıyı içerir ve bu yapı OFPAT_EXPERIMENTER biçiminde tanımlanmaktadır. Akış eşleştirme alanları, AçıkAkış AGE TUD kodlaması kullanılarak tanımlanmıştır. Her AGE TUD 5 ila 259 bayt arasında bir uzunluğa sahiptir. Bir AGE TUD kodlamasının ilk 4 baytı başlıktır ve ardından girişin gövdesi gelmektedir. AçıkAkış ekleri anahtarları İDP/KDP portlarına göre yönlendirmesini ve küme denetleyicilerinin veri bağı ve ağ katmanlarında Medya Erişim Kontrolü (MEK),

³⁶ ONE, *OpenFlow Switch Specification Version 1.3.2*, <https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.3.2.pdf>, Erişim Tarihi: 07/11/2019

İnternet Protokolü (İP), Çok Protokollü Etiket Değişirme (ÇPED), SYAA bilgilerini işleyebilmesini sağlamaktadır.

```
struct ofp_action_experimenter_header {
    uint16_t type;
    uint16_t len;
    uint32_t experimenter;
};

OFP_ASSERT(sizeof(struct
ofp_action_experimenter_header)==8);
```

Konfederasyon denetleyicisi orkestrasyondan aldığı TEDA UPA biçimindeki servis fonksiyon zinciri bilgilerini Akış Açık deneyci mesajlarını aşağıdaki eylemleri uygulayarak küme kontrolcülerine gönderir:

```
struct ofp_experimenter_sfch_action {
    OFPAT_PUSH_SFCH,
    OFPAT_POP_SFCH,
    OFPAT_SET_SFP,
    OFPAT_SET_SFI,
    OFPAT_INC_SFI,
    OFPAT_SET_MD,
};
```

Servis zinciri başlığını bir paketin üzerine itmek veya SFZ'yi bir paketten çıkarmak için yukarıdaki Akış Açık kodundan görüldüğü gibi deneyci eylemleri kullanılacaktır. Konfederasyon denetleyicisi bir SFZ'yi giden bir pakete yerleştirmek için orkestrasyona ya da küme kontrolcüsüne push SFZ eylemi gönderir. Konfederasyon kontrolcüsü giden paketlerden bir SFZ kaldırmak için pop SFZ eylemi gönderir.

Küme denetleyicileri, konfederasyon denetleyicisinden aldıkları servis fonksiyon zinciri bilgilerine göre anahtarların akış tablolarını aşağıdaki gibi flow-mod komutu aracılığıyla programlamaktadır.

```
flow-mod
cmd=add,table=<table_no>,eth_dst=<eth_dst>,vlan_vid=<vlan_vid>,apply:output=
<output_port>
```

2.1.3 Altyapı katmanı

YTA Konfederasyonu mimarisi birlikte çalışabilirliği ve kolay yönetilebilirliği artırmak için Şekil 1’de görüldüğü gibi farklı karakterdeki anahtarlardan oluşan altyapıyı kümelere bölmektedir. YTA Konfederasyonu mimarisinde yukarıda bahsedildiği üzere trafiğin akması için her yaprak anahtar omurga anahtarların her birine birebir bağlıdır. Yaprak anahtarlar ve bunların üst bağlantıları arasında herhangi bir aşırı abonelik yoksa engelleyici olmayan bir yapı elde edilmektedir³⁷. Bu sistem denetleyici başına düşen yükü azalttığı için AçıkAkış protokolünün önerdiği çoklu tablo kullanıma gerek kalmamakta ve böylece yüksek maliyetli olan ÜİAB’nin getirdiği kısıtlamalara çözüm olmaktadır³⁸.

YTA Konfederasyonu mimarisi en az AçıkAkış 1.4 protokolünü destekleyen herhangi bir sanal anahtar ile uyumludur. Harici kontrol cihazlarının birden fazla tablo kullanarak paket işleme hattı oluşturabilmesi ve her tablonun birden fazla girişle programlanabilmesi için AçıkAkış 1.5 protokol sürümünün tercih edilmelidir. Bu tablolar herhangi bir eşleşme alanı kombinasyonunu, Sanal Genişletilebilir YAA (SGYAA), Sanal YAA (SYAA) ve Ağ Sanallaştırma Genel Yönlendirme Kapsülleme (ASGYK) gibi birden fazla kapsülleme yapan sanal ağ teknolojisini desteklemektedir.

Bu çalışmada farklı karakterdeki donanımların küme denetleyicileri ile iletişim kurabilmesi için TTŞ yapısı kullanılmıştır. TTŞ yapısı donanım bağımsızlığını desteklemek için AAK ve diğer YTA öncülleri tarafından geliştirilmiştir. Bir kontrol profili veya AçıkAkış şablonu gibi işlev gören TTŞ mantıksal yönlendirme hattını aşağıdaki örnek kodlamada görüldüğü gibi AçıkAkış terimleriyle tanımlamaktadır.

³⁷ Maciej Kúznar, Peter Perešini, Dejan Kostić, (04 March 2015), *What you need to know about SDN control and data planes*, International Conference on Passive and Active Network Measurement, 2015, 347-359

³⁸ Long et al., (20 May 2012), *Research on TCAM-based Openflow switch platform*, International Conference on Systems and Informatics (ICSAI), IEEE, 2012

```

"Meter_table":{
  "meter_types": [
    {"name":"ControllerMeterType",
      "bands": [{"type": "DROP", "rate":<rate> , "burst":<burst>}]
    },
    {"name":"TrafficMeter",
      "bands": [{"type": "DSCP_REMARK", "rate":<rate>
, "burst":<burst>},
        {"type": "DROP", "rate":<rate> , "burst":<burst>}]
    }
  ],
  "built in meters": [
    {"name": "ControllerMeter", "meter_id":1,
      "type": "ControllerMeterType", "bands": [{"rate":<rate>,
"burst":<burst>}]},
    {"name": "AllArpMeter", "meter id":2,
      "type": "ControllerMeterType", "bands": [{"rate":<rate>,
"burst":<burst>}]}
  ]
}

```

Denetleyici ve anahtar bağlantı zamanında uygun gelen TTŞ'yi seçmekte ve çalışma zamanından önce hangi AçıkAkış mesajlarının desteklenmesi gerektiğine karar vermektedir. TTŞ'ler ihtiyaç duyulan işlevleri tanımlamak için YTA uygulama geliştiricileri tarafından oluşturulmakta ve anahtar tedarikçileri tarafından aygıtlarının neler yapabileceğini tanımlamak için kullanılmaktadır. Her bir TTŞ'nin benzersiz bir adı vardır, anahtarlar ve denetleyiciler desteklediği TTŞ'leri tanıtabilmektedir. Böylece farklı özellikteki anahtar kümeleri birlikte çalışabilmektedir. AAK birçok ürün grubunda desteklenen TTŞ setini tanımlamak için anahtar ve yonga üreticileri ile birlikte çalışmaktadır³⁹.

2.2. YTA Konfederasyonu Mimarisi Kullanım Örneği

Bu kısımda YTA Konfederasyonu mimari katmanları bir örnek üzerinden bulut yapısı içinde incelenmiştir. Bulut yapısı ön yatırımın ortadan kaldırılması, operasyonel giderlerin azaltılması, talep üzerine bilgi işlem kaynaklarının, esnek ölçeklendirmenin ve bilgi teknolojisi için kullanım başına ödeme iş modelinin oluşturulması gibi temel kavramlar üzerine inşa edilmiş, yaygın olarak kabul gören bir bilgi işleme paradigması olarak ortaya çıkmıştır ve bu sebeplerden dolayı telekomünikasyon şirketleri

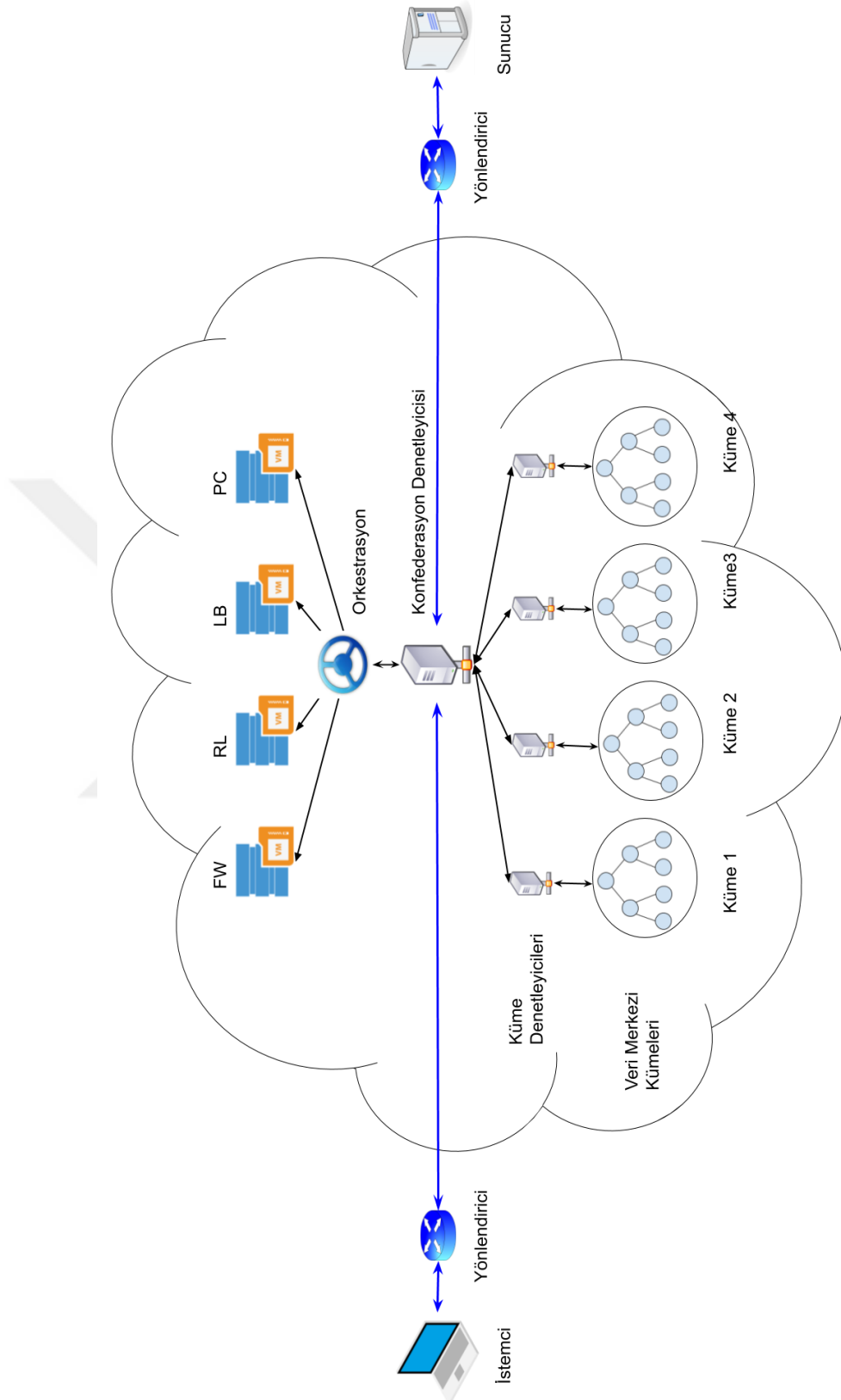
³⁹ ONE, *Simplifying OpenFlow Interoperability with Table Type Patterns (TTP)*, <https://www.opennetworking.org/wp-content/uploads/2014/10/sb-TTP.pdf>, Erişim Tarihi: 08/11/2019

tarafından tercih edilmektedir⁴⁰. Örnekte bu çalışma Şekil 5'te görüldüğü gibi bulut yapısı içerisinde ayrıntılı olarak açıklanmaktadır. Alt bölümlerde servis ve uygulamalar, donanım kümeleri, konfigürasyon tabloları, trafik akışı ve yönlendirme ayrıntılı olarak irdelenmiştir. Bulut yapısının seçilmesinin sebebi küresel ağ trafiği bileşenleridir. Küresel ağ trafiği bileşenine bakıldığında ağ trafiğinde 2015-2020 yılları arasında % 22'lik büyüme oranı görülmektedir⁴¹. Bu trafik Geniş Alan Ağı (GAA) mobil ağlar ve omurga bağlantıları üzerinden seyahat eden veri merkezleri içindeki ağ trafiğini temsil etmektedir. Günümüzün modern uygulamalarının giderek artan veri miktarını yönlendirmek için telekomünikasyon şirketleri tarafından bulut bilgi işlem kullanılmaktadır.

YTA Konfederasyonu mimarisinde orkestrasyon, konfederasyon denetleyicisinin kuzey sınırındaki arayüzü aracılığıyla gerekli değişiklikleri talep etmekte, konfederasyon denetleyicisi de istenen değişiklikleri AçıkAkış protokolü aracılığıyla küme denetleyicisine iletmektedir. Servis işlevleri AçıkAkış protokolü desteklemiyorsa veya AçıkAkış protokolü ağ denetleyicileri alanı altında değilse, konfederasyon denetleyicisinin böyle durumlar için önceden yapılandırılmış varsayılan servis işlevleri için küme denetleyicisi ile doğrudan etkileşime girmesi gerekmektedir.

⁴⁰ Siamak Azodolmolky, Philipp Wieder, Ramin Yahyapour, (27 June 2013), *SDN-based cloud computing networking*, 15th International Conference on Transparent Optical Networks (ICTON), IEEE, 2013

⁴¹ Barnett et al., *Cisco Global Cloud Index 2015–2020*, https://www.cisco.com/c/dam/m/en_us/service-provider/ciscoknowledgenetwork/files/622_11_15-16-Cisco_GCI_CKN_2015-2020_AMER_EMEAR_NOV2016.pdf, Erişim Tarihi: 08/11/2019



Şekil 5: Veri Merkezleri İçin YTA Konfederasyonu Bulutu

2.2.1 Servis ve uygulamalar

YTA Konfederasyonları mimarisini bulut sistemi içinde örneklendirirken çalışmasındaki ağ işlevleri sanal olarak konumlandırılmıştır. Sanallaştırma birden fazla heterojen ağ mimarisinin paylaşılan fiziksel altyapı katmanı üzerinde birlikte çalışmasını sağlayarak daha esnek ve daha kolay yönetilebilir ağ yapısı sunmaktadır⁴². Bunun için de çalışmada AFS YTA mimarisi ile birlikte kullanılmış ve ilerleyen bölümlerde bu yapı üzerinden konfigürasyon tabloları tanımlanmıştır. AFS servis fonksiyonlarını yerine getirmek üzere ağ donanım yerine yazılım aracılığıyla yöneterek sanallaştırma teknolojilerini kullanan bir ağ mimarisidir. AFS Sanal Ağ İşlevleri (SAİ) yapı taşlarından oluşmaktadır. AFS sanallaştırılmış servis fonksiyonlarını yönlendiriciler, anahtarlar, sunucular veya bulut bilgi işlem sistemleri gibi donanım ağ altyapısının üstünde bir yada daha fazla sanal makinede çalıştırmaktadır. SAİ'ler orkestrasyon tarafından yönetilmekte ve denetlenmektedir. Orkestrasyon YTA denetleyicisinin bir parçası olabileceği gibi ayrı bir modül olarak da mimariye dahil edilebilmektedir. Bir YTA denetleyicisi için SAİ sadece bir başka kaynak, ağda kontrol edilebilir bir düğümdür. YTA mimarisi servis fonksiyonlarını sunabilmedeki esnekliği SAİ'ler sayesinde artmış, YTA ve AFS birbirini tamamlayan mimariler hale gelmişlerdir⁴³.

Bu çalışmada ağ fonksiyonlarına örnek olarak güvenlik duvarı, yük dengeleyici, yönlendirici filtresi ve ebeveyn kontrolü kullanılmıştır. Güvenlik duvarı güvenilmeyen bir bölge ve güvenilir bir bölge arasındaki trafik akışlarına ağa erişim izni veren veya reddeden bir ağ fonksiyonudur⁴⁴. Yük dengeleyici gelen trafiği birden fazla yola dağıtarak hedefe ulaştırmaktadır⁴⁵. Yönlendirici filtresi belirli rotaların

⁴² N. M. Mosharaf Kabir Chowdhury, Raouf Boutaba, (31 July 2009), *Network virtualization: a state of the art and research challenges*, IEEE Communications Magazine, IEEE, 2009, 20-26

⁴³ ONF, *TR-518 Relationship of SDN and NFV*, https://www.opennetworking.org/wp-content/uploads/2014/10/onf2015.310_Architectural_comparison.08-2.pdf, Erişim Tarihi: 09/12/2019

⁴⁴ Jarrod N. Bakker, Ian Welch, Winston K.G. Seah, (10 November 2016), *Network-wide virtual firewall using SDN/OpenFlow*, IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), IEEE, 2016

⁴⁵ Hailong Zhang, Xiao Guo, (29 November 2014), *SDN-based load balancing strategy for server cluster*, IEEE 3rd International Conference on Cloud Computing and Intelligence Systems, IEEE, 2014

konfigürasyonuna göre yerel rota veri tabanına eklenip eklenmeyeceğine karar vermektedir⁴⁶. Ebeveyn kontrolü ağ erişim cihazlarının trafik, konum ve kullanıcıya tepki veren içerik filtrelerine sahip olabilme yeteneğidir⁴⁷.

2.2.2 Donanım kümeleri

YTA mimarilerinde ağırlıklı olarak yalın metal (bare metal), beyaz kutu (white box) ve markalı yalın metal (brite box) anahtarları kullanılmaktadır⁴⁸. Yalın metal anahtarlar Orijinal Tasarım Üreticileri (OTÜ) tarafından üretilen ve üzerinde AİS olmayan anahtarlardır. Beyaz kutu anahtarlar yalın metal anahtarlara AİS eklenmiş anahtarlardır. Markalı yalın metal anahtarlar beyaz kutu anahtarların markalı versiyonudur ve avantajları müşteri hizmetleriyle birlikte geldikleri için daha güven verici olmaları, dezavantajı ise daha maliyetli olmalarıdır. Bir de geleneksel ağ mimarilerinde kullanılmış, fakat YTA mimari desteği olmayan anahtarlar vardır.

Bu çalışmada yalın metal, beyaz kutu, markalı yalın metal ve YTA mimari desteği olmayan dört tip anahtar örneği dört farklı konfederasyon kümesini temsil etmek üzere kullanılmıştır. Bu farklı özellikteki anahtarlar YTA mimarisinin popüler hale gelmesinden sonra anahtarların telekomünikasyon ihtiyaçlarına göre evrim geçirmesiyle ortaya çıkmıştır. Tercih edilmelerinin sebebi servis sağlayıcılarının kullanım durumuna göre değişmektedir. YTA Konfederasyonu mimarisinde farklı platformlar aşağıdaki kritik noktalar için kümelere ayrılmıştır:

- Yeni çıkan donanımlara yatırım yapan şirketler önceki altyapılarını başka fonksiyonlara servis amacıyla kullanmayı tercih etmektedirler. Örneğin tedarikçi tekelinden kurtulmak için daha ucuz ve performanslı yalın metal

⁴⁶ Andreas Blenk, Arsany Basta, Wolfgang Kellerer, (15 May 2015), *HyperFlex: An SDN virtualization architecture with flexible hypervisor function allocation*, IFIP/IEEE International Symposium on Integrated Network Management (IM), IEEE, 2015

⁴⁷ Habibi et al., (19 June 2014), *Personalizing the home network experience using cloud-based SDN*, Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, IEEE, 2014

⁴⁸ Victor Lopez, Oscar Gonzalez de Dios, Juan Pedro Fernandez-Palacios, (7 March 2019), *Whitebox Flavors in Carrier Networks*, Optical Fiber Communications Conference and Exhibition (OFC), IEEE, 2019

anahtarlar alan bir operatör, bu cihazları yüksek performans gerektiren ses ve görüntü trafiği için kullanmak isterken, eski donanımları ağdan tamamen çıkartmak yerine daha düşük bant genişliği gerektiren fonksiyonlar için kullanmaktadır.

- Cihaz seviyesinde değişiklik küme seviyesindeki değişikliklere göre daha sık olacağı için zor, yavaş ve riskli ağ güncellemelerini de artırmaktadır. Duraksız trafik talepleri bu durumda kesintiye uğrar ve operasyon maliyetini de artırmaktadır.

- Her operasyonu otomasyon sistemine dönüştürmeye çalışırken bu denetleyici için kuzey güney arayüzlerini ve anahtar işleyişini daha karmaşık hale getirmektedir. Karmaşıklık arttıkça bu durum anahtarı aşırı yükler, dolayısıyla denetleyici ve anahtar arasındaki akış tablosunda uyumsuzluklar oluşmaktadır.

- Birçok servis sağlayıcısının ağları yüksek ölçüde YTA olmayan anahtar içermektedir. YTA mimarisine geçerken bu cihazları asgari statik bir çabayla, diğerlerinden farklı bir küme içinde kullanılabilir hale getirmek mümkündür.

Hibrid YTA, hem YTA istemcisi olan ve hem de olamayan anahtarların birlikte YTA mimarisinde olabilmesini sağlayan yaklaşımdır. Bu konuda birçok yöntem geliştirilmiştir. Çalışmadaki mimariye uyum sağlaması açısından sadece denetleyici (controller only) metodu seçilmiştir⁴⁹. Bu metotta YTA mimarisine uyumlu olmayan anahtarlar denetleyiciye İç Sınır Ağ Geçidi Protokolü (İSAGP) bağlantısı yaparak tüm sisteme dahil olmuştur. Bunun için denetleyici ve anahtarda bir konfigürasyon değişikliği yeterlidir.

⁴⁹ Sandhya, Yash Sinha, K.Haribabu, (15 December 2017), *A survey: Hybrid SDN*, *Journal of Network and Computer Applications*, ScienceDirect, 2017, 35-55

2.2.3. Konfigürasyon tabloları

Bu kısımda verilen tablolar bulut ağındaki küme altyapılarına ve servis ihtiyaçlarına göre oluşturulmaktadır. Tablo 1’de trafik çeşitleri, sanal makineler ve konfederasyon kümelerine ait varsayımlar listelenmiştir. Bu varsayımlar örnek amacı ile bulut yapısında en çok kullanılan trafik çeşitleri, sanal makine fonksiyonları ve donanım tiplerine göre seçilmiştir.

Tablo 1: YTA Konfederasyonu Bulut Varsayımları

Varsayımlar	
Trafik çeşitleri	<ul style="list-style-type: none">• Metin Üstü Aktarım Protokolü (MÜAP)• Video• Güvenli Kabuk (GKA)• Dosya Aktarım Protokolü (DAP)
Sanal makine fonksiyonları	<ul style="list-style-type: none">• Güvenlik Duvarı (GD)• Hız Sınırlayıcı (HS)• Yük Dengeleyici (YD)• Ebeveyn Kontrolü (EK)
Donanım tipleri	<ul style="list-style-type: none">• K1: yalın metal anahtar kümesi• K2: beyaz kutu anahtar kümesi• K3: markalı yalın metal anahtar kümesi• K4: YTA desteği olmayan anahtar kümesi

Tablo 2’de orkestrasyonda yapılandırılacak servis zinciri tanımı gösterilmiştir. Bu tanıma göre uygulama sütununda bulunan farklı servis tiplerinin sırasıyla hangi sanal makineleri ve kümeleri ziyaret edeceği belirtilmiştir. Örnek olarak video trafik tipine göre daha az bant genişliği gerektiren MÜAP uygulaması daha az kapasiteli olan anahtar K2, K3 kümelerinden ve güvenlik duvarından, yük dengeleyiciden geçmektedir. Öte yandan daha çok bant genişliğine ihtiyaç duyan video trafiği sunucudan istemciye olan yolculuğunda daha çok kapasiteli anahtarlardan oluşan K1 kümesinden ve hız sınırlayıcı sana makineden geçmektedir. Buradaki tanımlar örneklendirme amacıyla yapılmıştır ve özel ihtiyaçlara göre operatörler tarafından farklı servis zinciri tanımları yapılabilmektedir.

Tablo 2: Orkestrasyon Servis Zinciri Tanımı

Uygulama	GD	K1	YD	K2	K3	HS	EK	C4
MÜAP	X		X	X	X			
video		X				X		
GKA	X			X			X	
DAP	X					X		X

Tablo 3'te orkestrasyon ilk adım konfigürasyonları verilmiştir. Bu tablo her trafik tipinin servis tipini ve sunucudan istemciye doğru yolculuğa çıkan trafiğin ilk adımını gösteren örnek bilgileri içermektedir. Servis tipi bilgisi akış eşleştirmede kullanılmaktadır ve her paketin içinde bu bilgi bulunmaktadır.

Tablo 3: Orkestrasyon İlk Adım Konfigürasyonları

Uygulama	Servis Tipi	İlk Adım
MÜAP	4	GD
video	8	K1
GKA	12	GD
DAP	16	GD

Tablo 4'te orkestrasyon ve denetleyiciye air yönlendirme bilgileri verilmiştir. Orkestrasyonda sanal makinelere ait servis tipi ve sonraki adım bilgileri bulunmaktadır. Küme denetleyicilerinde ise her kümeye ait servis tipi ve sonraki adım bilgileri bulunmaktadır. Bu bilgilere göre akan trafik sonraki adımlara yönlendirilmektedir.

Tablo 4: Orkestrasyon ve Denetleyici Yönlendirme Bilgileri

Sanal Makine Bilgileri			Küme Bilgileri		
Sanal Makine	Servis Tipi	Sonraki Adım	Küme	Servis Tipi	Sonraki Adım
GD	4	YD	K1	4	GD
GD	8	K1	K1	8	HS
GD	12	K2	K1	12	GD
GD	16	HS	K1	16	GD
YD	4	K2	K2	4	K3
YD	8	K1	K2	8	K1
YD	12	GD	K2	12	EK
YD	16	GD	K2	16	GD
HS	4	K3	K3	4	son
HS	8	son	K3	8	K1
HS	12	K2	K3	12	K2
HS	16	K4	K3	16	GD
EK	4	GD	K4	4	K3
EK	8	HS	K4	8	HS
EK	12	GD	K4	12	EK
EK	16	GD	K4	16	son

Tablo 5’te konfederasyon denetleyicisindeki kümeler gösterilmiştir. Burada örnek olarak endüstride en çok kullanılan yalın metal, beyaz kutu, markalı yalın metal ve YTA desteği olmayan anahtar kümelerine yer verilmiştir. Operatörler tarafından farklı karakterdeki anahtarları içeren kümeler oluşturulabilmektedir.

Tablo 5: Konfederasyon Denetleyicisindeki Kümeler

Küme Numarası	Küme Adı
1	Yalın metal
2	Beyaz kutu
3	Markalı yalın metal
4	YTA desteği olmayan

Tablo 6’da her kümede yer alan temsili anahtarlar MEK adresleriyle birlikte örnek olarak verilmiştir. Buradaki küme anahtarlarına ait MEK adreslerinin küme denetleyicisi tarafından AçıkAkış protokolü ile dinamik olarak öğrenilmesi önerilmektedir.

Tablo 6: Küme Denetleyicisindeki Anahtarlar

MEK	Küme Numarası
00-14-22-01-23-45	1
00-14-22-01-23-46	1
00-14-22-02-23-47	2
00-14-22-02-23-48	2
00-14-22-03-23-49	3
00-14-22-03-23-50	3
00-14-22-04-23-51	4
00-14-22-04-23-52	4

2.2.4 Trafik akışı ve yönlendirme

Konfigürasyon tablolarındaki bilgiler doğrultusunda Şekil 6’da görüldüğü gibi YTA Konfederasyonu omurga ve yaprak mimarisinde istemci içeriğe ulaşmak için düğmeye basmakta ve talep erişim ağına yönlendirilmektedir. Erişim ağındaki yönlendirici veya güvenlik duvarı bu eşleştirmeyi kaydetmekte ve uygulama katmanındaki orkestrasyona iletmektedir. Orkestrasyon SFZ’deki uygulama ve servis

eşleştirme konfigürasyonlarına göre trafiği yönlendirmektedir. Eğer servis fonksiyonu zincirinde sanal makinalar varsa trafik güvenlik duvarı, hız sınırlayıcı, yük dengeleyici veya ebeveyn kontrolü gibi sanal makinelerinden birine yönlendirilmekte ve işlem sonucunda sanal makinalar orkestrasyona bildirim göndermektedir. Benzeri şekilde eğer servis fonksiyonu zincirinde kümelere özel bir yönlendirme varsa önce konfederasyona ait denetleyici geçen trafiği servis ve TTŞ eşleştirmesi için sorgulamaktadır. Sorgulama olumsuz ise paket ağdan düşürülmektedir. Eğer sorgulama olumlu ise AçıkAkış tablosunda TTŞ ve MEK eşleştirmesi yapılmakta ve eşleştirme eğer olumlu ise trafik konfederasyon kümelerine yönlendirilmektedir. Eğer eşleştirme olumsuz ise küme denetleyicisine MEK öğrenme işlemini başlatması için talep mesajı gönderilmektedir. MEK öğrenme işlemi başarıyla tamamlandığında küme denetleyicisi paketi işlemektedir. İşlem başarıyla tamamlanmadığında paket ağdan düşürülmektedir. Bu döngü servis fonksiyonları zincirindeki son adıma ulaşana kadar devam etmekte ve son adımda paket hedefe ulaştırılmaktadır.

Buna göre küme denetleyicisi ile anahtarlar arasındaki arayüzün kapsüllemeyi işleyecek şekilde geliştirilmesi gerekmektedir. YTA Konfederasyonu mimarisinin konfigürasyon tablolarına göre yapılandırıldığı varsayılırsa, trafik akışı aşağıdaki gibi gerçekleşmektedir:

- Küme denetleyici güney uygulama arayüzü Tablo 6'daki gibi MEK adres öğrenme işlemini tamamlamaktadır.
- Konfederasyon denetleyici güney uygulama arayüzüne Tablo 5'te görüldüğü gibi küme bilgisi verilmektedir.
- Küme denetleyici kuzey uygulama arayüzü Tablo 6'ki anahtarlar için TTŞ bilgisini orkestrasyon güney uygulama arayüzüne göndermektedir. Temsili örnek açısından TTŞ'ler farklı özellikte anahtar barındıran kümeler için aşağıdaki kodda görüldüğü gibi yapılandırılmaktadır. Aşağıdaki kodda ölçüm tablosu (meter_table) ölçüm tiplerini (meter_types) ve var olan ölçümleri (built_in_meters) içermektedir. Her ölçüm tipinin ismi (name) ve bant (bands) tanımı bulunmaktadır. Bant tip (type), hız (rate), patlama (burst) bilgileri ile tanımlanır. Tüm bu değerler anahtarın gelen trafiği hızına göre nasıl yönettiğini belirtir. Bunun gibi anahtara ait birçok özellik uygun TTŞ'ler aracılığı ile tanımlanmaktadır.

- K1 kümesinden video trafik akışı için TTŞ tanımı:

```
"Meter_table":{
  "meter_types": [
    {"name":"ControllerMeterType",
     "bands": [{"type": "DROP", "rate":
"1000..10000", "burst":"50..200"}]
    },
    {"name":"TrafficMeter",
     "bands": [{"type": "DSCP_REMARK", "rate":
"10000..500000", "burst":"50..500"},
              {"type": "DROP", "rate": "10000..500000", "burst":
"50..500"}]
    }
  ],
  "built_in_meters": [
    {"name": "ControllerMeter", "meter_id":1,
     "type": "ControllerMeterType", "bands": [{"rate":2000, "burst":
75}]},
    {"name": "AllArpMeter", "meter_id":2,
     "type": "ControllerMeterType", "bands": [{"rate":1000,
"burst":50}]}
  ]
}
```

- K2 kümesinden MÜAP trafik akışı için TTŞ tanımı:

```
"Meter_table":{
  "meter types": [
    {"name":"ControllerMeterType",
     "bands": [{"type": "DROP", "rate":
"1000..10000","burst":"50..200"}]
    },
    {"name":"TrafficMeter",
     "bands": [{"type": "DSCP REMARK", "rate":
"1000..50000","burst":"50..500"},
               {"type": "DROP", "rate": "1000..50000", "burst":
"50..500"}]
    }
  ],
  "built in meters": [
    {"name": "ControllerMeter", "meter_id":1,
     "type": "ControllerMeterType","bands": [{"rate":2000, "burst":
75}}},
    {"name": "AllArpMeter", "meter id":2,
     "type": "ControllerMeterType","bands": [{"rate":1000,
"burst":50}}]
  ]
}
```

- K3 kümesinden GKA trafik akışı için TTŞ tanımı:

```
"Meter table":{
  "meter types": [
    {"name":"ControllerMeterType",
     "bands": [{"type": "DROP", "rate":
"100..1000","burst":"50..200"}]
    },
    {"name":"TrafficMeter",
     "bands": [{"type": "DSCP REMARK", "rate":
"1000..50000","burst":"50..500"},
               {"type": "DROP", "rate": "1000..50000", "burst":
"50..500"}]
    }
  ],
  "built_in_meters": [
    {"name": "ControllerMeter", "meter_id":1,
     "type": "ControllerMeterType","bands": [{"rate":2000, "burst":
75}}},
    {"name": "AllArpMeter", "meter id":2,
     "type": "ControllerMeterType","bands": [{"rate":1000,
"burst":50}}]
  ]
}
```


- K4 kümesinden DAP trafik akışı için TTŞ tanımı:

```
"Meter_table":{
  "meter_types": [
    {"name":"ControllerMeterType",
     "bands": [{"type": "DROP", "rate":
"100..1000","burst":"50..200"}]
    },
    {"name":"TrafficMeter",
     "bands": [{"type": "DSCP REMARK", "rate":
"1000..50000","burst":"50..500"},
               {"type": "DROP", "rate": "1000..50000", "burst":
"50..500"}]
    }
  ],
  "built in meters": [
    {"name": "ControllerMeter", "meter id":1,
     "type": "ControllerMeterType","bands": [{"rate":2000, "burst":
75}}],
    {"name": "AllArpMeter", "meter_id":2,
     "type": "ControllerMeterType","bands": [{"rate":1000,
"burst":50}}]
  ]
}
```

- Orkestrasyon kuzey uygulama arayüzüne ağ yöneticisi tarafından Tablo 2’deki gibi servis zinciri bilgileri verilmektedir.
- Orkestrasyon, fonksiyonları yerine getiren sanal makineleri başlatma talebi göndermektedir ve her SAI’yi Tablo 3’deki gibi ilk adım bilgileri ile programlamaktadır. Orkestrasyonun denetleyiciyi nasıl programladığına dair ayrıntılar YTA/AFS orkestrasyon durum çalışmalarında işlenmektedir.⁵⁰
- Bu maddeler sonucunda YTA Konfederasyon bulutundaki tüm katmanlarda servislerin istenilen sanal makinelerden ve kümelerden geçerek hedefe ulaştırılacakları bilgisi programlanmıştır.

Yukarıdaki hazırlıklar tamamlandıktan sonra sunucu ve istemci arasındaki örnek olarak seçilen MÜAP trafik akışı aşağıdaki gibi olmaktadır:

- MÜAP trafik akışı sunucudan istemciye gitmek üzere başlamaktadır. Akış sınıflandırılmak üzere ilk olarak orkestrasyona yönlendirilir.

⁵⁰ Ordonez-Lucena et al., (12 May 2017), *Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges*, IEEE Communications Magazine, IEEE, 2017, 80-87

- Orkestrasyon, önceden yapılan tanımlara göre paket için gerekli olan servis fonksiyon zincirinin tanımını bulur. Tablo 2'ye göre MÜAP trafiği servis fonksiyon zincir kimlik numarası 4 ve sırasıyla yönlendirileceği duraklar GD, YD, K2, K3 şeklinde tanımlanmıştır.

- Orkestrasyon servis zinciri içindeki yönlendirme bilgisini kuzey arayüzünden sanal makinelerle TEDA UPA biçiminde aşağıdaki örnek gösterimde olduğu gibi paylaşmaktadır:

- Sanal makine GD'ye gönderilen temsili TEDA UPA

```
"request": {
  "path_information": {
    "kind": "service_function_chain",
    "flow_type": "HTTP",
    "path_id": "1",
    "path_index": "0"
  }
  "hop_information": {
    "type": "VM",
    "name": "FW"
  },
  "next_hop_information": {
    "type": "VM",
    "name": "LB"
  }
}
```

- GD paketi işlemekte ve küme denetleyicisine göndermektedir.
- Sanal makine YD'ye gönderilen temsili TEDA UPA

```
"request": {
  "path_information": {
    "kind": "service_function_chain",
    "flow_type": "HTTP",
    "path_id": "1",
    "path_index": "1"
  }
  "hop_information": {
    "type": "VM",
    "name": "LB"
  },
  "next_hop_information": {
    "type": "Cluster",
    "name": "C2"
  }
}
```

- Orkestrasyon servis zinciri içindeki yönlendirme bilgisini güney arayüzünden konfederasyon kontrolcüsü ile TEDA UPA biçiminde aşağıdaki örnek gösterimde olduğu gibi paylaşmaktadır:

```
"request": {
  "path_information": {
    "kind": "service_function_chain",
    "flow_type": "HTTP",
    "path_id": "1",
    "path_index": "2"
  }
  "hop_information": {
    "type": "Cluster",
    "name": "C2"
  },
  "next_hop_information": {
    "type": "Cluster",
    "name": "C3"
  }
}

"request": {
  "path_information": {
    "kind": "service_function_chain",
    "flow_type": "HTTP",
    "path_id": "1",
    "path_index": "3"
  }
  "hop_information": {
    "type": "Cluster",
    "name": "C3"
  },
  "next_hop_information": {
    "type": NULL,
    "name": NULL
  }
}
```

- Konfederasyon denetleyicisi orkestrasyondan aldığı TEDA UPA biçimindeki servis fonksiyonu bilgilerini AçıkAkış protokolü deneyici TUD'larına çevirerek küme kontrolcülerine göndermektedir.

- K2 küme denetleyicisine gönderilen servis fonksiyon zinciri eşleştirme operasyonları ve eylemleri aşağıdaki gibi örneklendirilmiştir:

```
oxm_ofb_experimenter_sfch_match.OXM_OF_SFCH_PATH_ID = 1;
oxm_ofb_experimenter_sfch_match.OXM_OF_SFCH_PATH_INDEX = 2;

ofp_experimenter_sfch_action.OFPAT_PUSH_SFCH = 3;
ofp_experimenter_sfch_action.OFPAT_POP_SFCH = 2;
ofp_experimenter_sfch_action.OFPAT_SET_SFP = 1;
ofp_experimenter_sfch_action.OFPAT_SET_SFI = 2;
ofp_experimenter_sfch_action.OFPAT_INC_SFI = 1;
```

- K3 küme denetleyicisine gönderilen servis fonksiyon zinciri eşleştirme operasyonları ve eylemleri aşağıdaki gibi örneklendirilmiştir:

```
oxm_ofb_experimenter_sfch_match.OXM_OF_SFCH_PATH_ID = 1;
oxm_ofb_experimenter_sfch_match.OXM_OF_SFCH_PATH_INDEX = 3;

ofp_experimenter_sfch_action.OFPAT_PUSH_SFCH = NULL;
ofp_experimenter_sfch_action.OFPAT_POP_SFCH = 3;
ofp_experimenter_sfch_action.OFPAT_SET_SFP = 1;
ofp_experimenter_sfch_action.OFPAT_SET_SFI = 3;
ofp_experimenter_sfch_action.OFPAT_INC_SFI = NULL;
```

- Küme denetleyicileri, konfederasyon denetleyicisinden aldıkları TUD'lardaki değerlere göre anahtarlara aşağıdaki gibi akış eklemektedir:

- K2

```
flow-mod
cmd=add,table=1,eth_dst=00:14:22:01:23:47,vlan_vid=10,
apply:output=1
```

- K3

```
flow-mod
cmd=add,table=1,eth_dst=00:14:22:01:23:49,vlan_vid=10,
apply:output=1
```

- Küme denetleyicisi ilk adım bilgisi ile paketi kapsülleme tekniğini kullanarak konfederasyon kümesine göndermektedir. Bu durumda yeni ilk adım YD'dir.
- Küme denetleyicisi YD bilgisine göre MÜAP trafiğini K2 ve K3 kümeleri üzerinden paylaştırarak hedefe ulaştırmaktadır.

2.3. YTA Konfederasyonu Mimarisi İle Benzeri YTA Mimarilerinin Karşılaştırılması

Bu bölümde çalışmanın ve benzeri diğer mimarilerin açık kaynak kod desteği, tedarikçi bağımsızlığı, YTA mimarisinin katman içi ve katman arası uyumluluk özellikleri karşılaştırılmıştır. YTA Konfederasyonu mimarisi ile kendi kullanımına özel geliştirilmiş YTA mimarilerinden Google, Facebook ve Amazon YTA mimarileri, ticari mimarilerden de Cisco, Nokia, Juniper ve Big Switch Networks YTA mimarileri karşılaştırılmıştır. Mimarilere ait genel bilgiler ve kaynakçalar literatür tarama bölümünde verilmiştir.

2.3.1 Açık kaynak kod desteği

Amazon, Google ve Facebook bazı açık kaynak organizasyonları desteklemektedirler, fakat mimarileri kendi ihtiyaçlarına özel ve gizli olduğu için hangi açık kaynak kodlu protokolü ne kadar kullandıkları konusunda net bir bilgi yoktur. Bununla birlikte Amazon AçıkYığın'ın en büyük destekçilerinden biridir. AçıkYığın bir kümedeki ağ ortamını simüle etmek için mekanizma sağlamaktadır⁵¹. Amazon sanallaştırılmış ağ ortamlarını programlanabilir bir şekilde desteklemek için AçıkYığın protokollerinin özelleştirilmiş bir versiyonunu kullanmaktadır. Ticari YTA mimarilerine bakıldığında açık kaynak kodlu olarak Nokia Nauge bir tek AçıkYığın

⁵¹ Omar Sefraoui, Mohammed Aissaoui, Mohsine Eleluldj, (October 2012), *OpenStack: Toward an Open-Source Solution for Cloud Computing*, International Journal of Computer Applications, 2012, 38-42

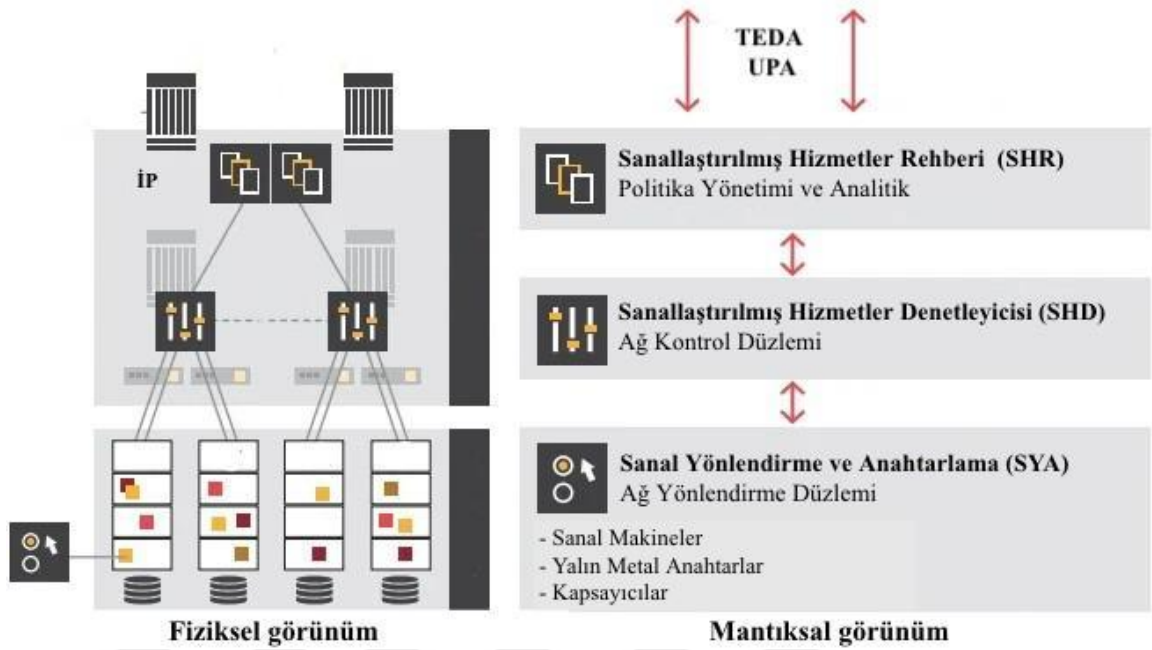
ile çalışabilmektedir. Cisco UMA açık kaynak kodlu AçıkAkış protokolü ile uyumludur ancak, YTA uygulamalarına adapte olacak mimariyi geliştirirken AAK standartlarına bağlı kalmamıştır. Contrail Networking farklı tedarikçileri desteklemek için Apache v2.0 açık kaynaklı lisansını kullanmaktadır. Bunun dışında Contrail Networking açık kaynak koda kısmen destek vermektedir.

2.3.2 Katmanlar arası uyum

Uygulama katmanı ve kontrol katmanı uyumu göz önüne alındığında ticari mimarilerin bu konuda sınırlı kaldığı görülmektedir. Şekil 7’de Nokia Nuage’nin bulut içerisinde YTA mimarisi görülmektedir. SHR modülü bu mimarinin uygulama katmanında yer almaktadır. Sanal makinelerle çalışmak üzere TEDA UPA kullanımını desteklemektedir, fakat SHR modülü açık kaynak kodlu değildir ve her yaratılan UPA’yı okumamaktadır. Örnek olarak Dinamik Ana Bilgisayar Yapılandırma Protokolü (DABYP) yapılandırması için tanımlanan IP aralığından bir IP adresi almak için gerekli olan UPA’yı desteklememektedir. Buna benzer örnekler YTA ağlarının uçtan uca çalışabilmesi için problem yaratmaktadır. YTA Konfederasyonu mimarisinde Bölüm 2 Kısım 2.1.1’de gösterildiği gibi orkestrasyondan gelen TEDA UPA mesajları konfederasyon denetleyicisinde AçıkAkış deneyici mesajlarına çevrildiği için bu problem çözülmüştür.

Şekil 7’de görüldüğü üzere Nokia Nuage’nin SYA modülü sanal fonksiyon oluşturma, taşıma ve silme fonksiyonlarını yerine getirmektedir⁵². Fakat bu ağ yönetimini zorlaştırmaktadır, çünkü orkestrasyon işlevleri uygulama katmanında gerçekleşmelidir. YTA Konfederasyonu mimarisinde bölüm 2 kısım 2.1’de gösterildiği gibi sanal fonksiyon işlemleri orkestrasyon tarafından uygulama katmanında yapılmaktadır.

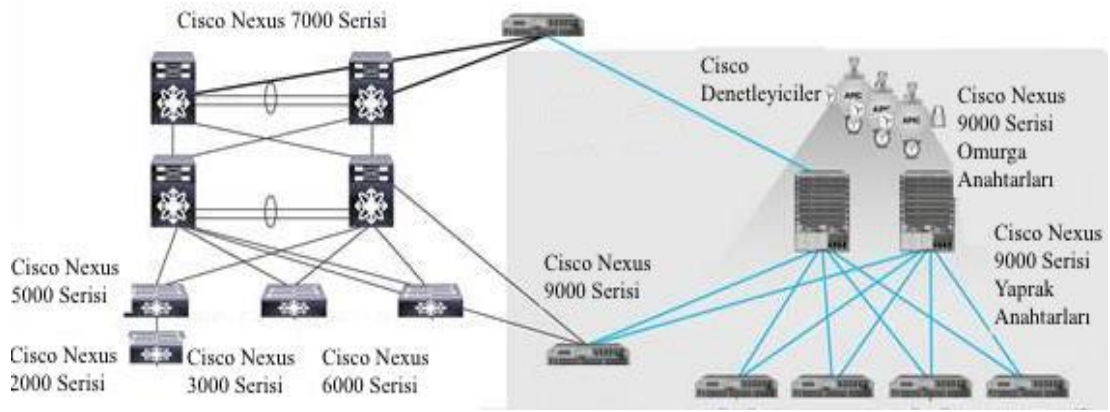
⁵² Nuage Networks, *Nuage Networks Product Architecture*, <https://www.nuagenetworks.net/wp-content/uploads/2015/08/NuageNetworks-Architecture-White-Paper.pdf>, Erişim Tarihi: 01/12/2020



Şekil 7. Nokia Nuage YTA Mimarisi

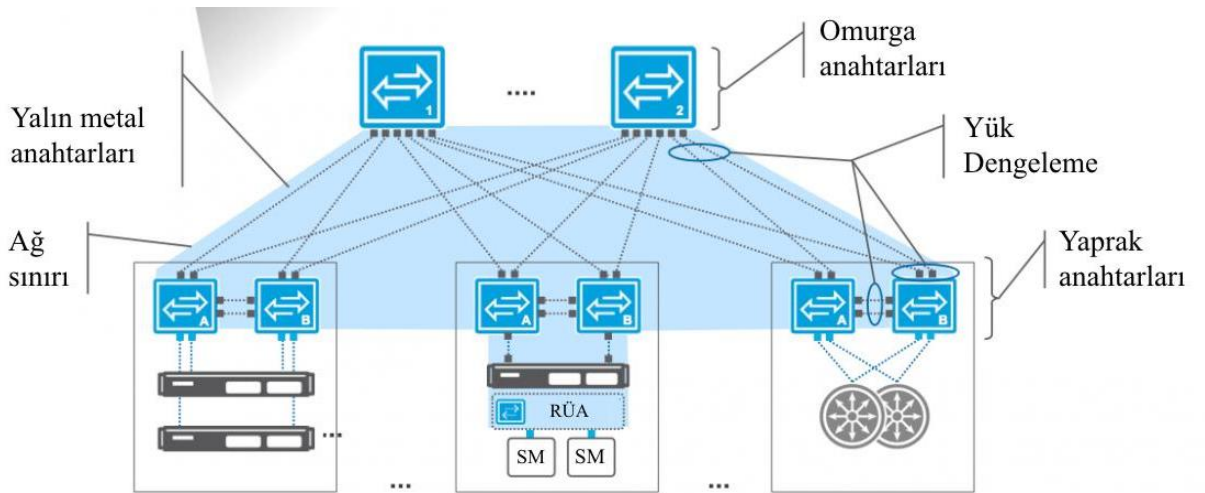
Şekil 8’de görüldüğü üzere Cisco Nexus 9000 serisi anahtarlar bir tek mevcut Cisco Nexus anahtarlarla uyumluluk ve tutarlılık içinde Cisco NX işletim sistemi aracılığıyla çalışabilmektedir⁵³. UPA'nın uygulama politikası odaklı servislerinden ve altyapı otomasyon özelliklerinden tam olarak yararlanmak için UMA modu kullanılmaktadır. Bu durum servis bulut entegrasyonunu engellemektedir. YTA Konfederasyonu mimarisinde Bölüm 2 Kısım 2.4’de örneklendirildiği gibi servislerin bulut entegrasyonu TEDA UPA’lar ve TTŞ’ler ile katmanlar arası korunmaktadır.

⁵³ Cisco, *Transform Your Business and Protect Your Cisco Nexus Investment While Adopting Cisco Application Centric Infrastructure*, <https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-731822.pdf>, Erişim Tarihi: 01/12/2020



Şekil 8. Cisco UMA YTA mimarisi

Cisco UMA YTA mimarisi AçıkAkış protokolünü kullanarak kontrol ve altyapı katmanını senkronize etmektedir. Trafik AçıkAkış denetleyicisi tarafından sınıflandırılmakta ve yönlendirilmektedir. Bu durum uygulama ve altyapı uyumunu kapsamamaktadır. Bu nedenle uygulama katmanındaki servisler, altyapı katmanındaki farklı karakterdeki anahtar kümeleriyle eşleştirilerek yapılandırılmamaktadır. YTA Konfederasyonu mimarisinde Bölüm 2 Kısım 2.2.3'te örneklendirildiği gibi trafik orkestrasyondaki altyapı katmanındaki kümelere göre yapılmış konfigürasyonlara göre sınıflandırılır.



Şekil 9. Big Switch Networks BBA YTA Mimarisi

Şekil 9’da görüldüğü gibi BBA denetleyicisi AçıkYığın ile uyumlu YTA denetleyicisi için uygulama şablonu sunmaktadır⁵⁴. Fakat bu şablonlar altyapı katmanı ile uyumlu değildir ve bu durum bulut yapısının uçtan uca çalışmasını aksatmaktadır. Bu duruma çözüm olarak çalışmada önerilen her küme için servis fonksiyonları tanımlanmıştır. Bölüm 2 Kısım 2.4’te örneklendirildiği gibi kümeler ve servis fonksiyonları kullanım durumlarına göre değişiklik gösterecektir. Her bir küme operatörün ihtiyacına göre trafik taleplerini karşılamak için TTŞ’leri SFZ’ye dönüştürerek çeşitli fonksiyonlarla eşleştirmektedir.

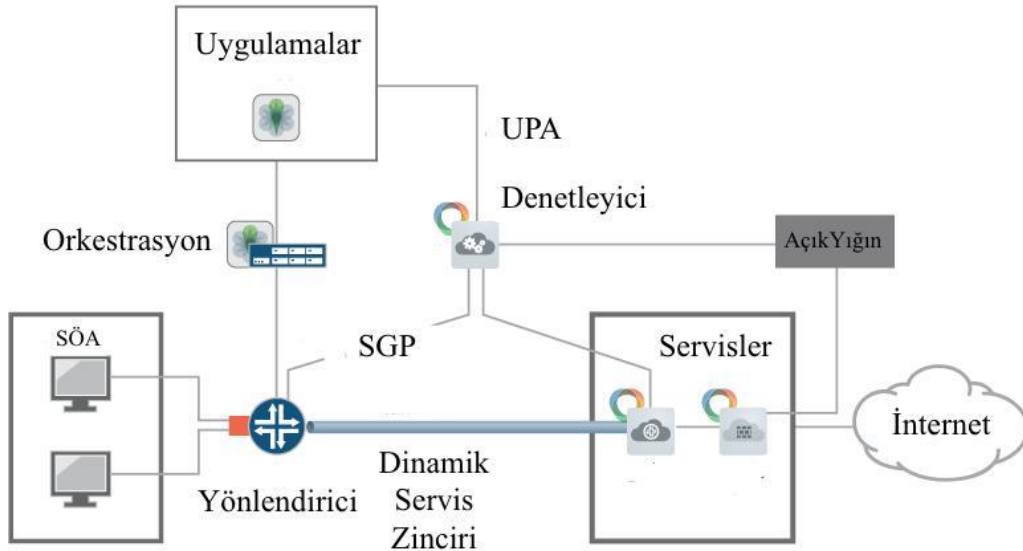
2.3.3 Katman içi uyum

YTA Mimarisinin altyapı katmanında farklı karakterdeki donanımların kümelenerek aynı bulut içinde çalışması telekomünikasyon şirketleri için kritik bir fonksiyondur. Hemen hemen her yıl önemli miktarda bant genişliği artışı destekleyen yeni donanımlar piyasaya sürülmektedir. Bu durum karşısında İnternet servis tedarikçileri ağ geliştirme durumlarına göre farklı karakterdeki donanımlardan oluşan hibrid ağlara sahip olmuşlardır. Operatörler daha az kapasiteli anahtarları tamamen ağdan çıkartmak yerine düşük bant genişliği ile çalışabilen uygulama tipleri için kullanmak istemektedirler.

Bazı ticari YTA mimarilerinin altyapı katman içinde tamamen uyumlu çalışmadığı görülmektedir. Nuage Networks’ ün YTA destekli veri merkezi çözümü fiziksel ve sanal cihazları desteklemek için X86 tabanlı açık mimari üzerine inşa edilmiştir. Fakat YTA Konfederasyonları mimarisinde Bölüm 2 Kısım 2.2.2’de örneklendirildiği gibi yalın metal veya beyaz kutu gibi farklı karakterdeki donanımları ayrı denetleyiciler altında kümelemediğinden desteklediği anahtar çeşidi oldukça sınırlıdır.

⁵⁴ Big Switch Networks, *Big Cloud Fabric Next Generation Data Center Switching Platform*, http://go.bigswitch.com/rs/974-WXR-561/images/What_is_BCF_WP_JAN17.pdf, Erişim Tarihi: 01/13/2020

Uygulama katmanı içindeki uyuma bakıldığında servis ve uygulamalar ile orkestrasyon arasındaki esneklik eş zamanlı trafik akış yönetimi için önemli bir fonksiyondur. Şekil 10’da görüldüğü üzere Juniper Contrail denetleyicisi herhangi bir standart İP omurga ve yaprak mimarisi ile çalışmayı desteklemekte ve konfigürasyon işlemlerini gerçekleştirmek için standart protokolleri kullanılmaktadır⁵⁵. Contrail Networking aygıtları bir veya daha fazla rolle ilişkilendirerek farklı ağ sanallaştırma işlemlerini uygulamak için yapılandırma şablonlarını kullanmaktadır. Fakat bu işlev yeni sanal ağlar eklemek, yeni bitiş noktaları ve daha fazlası gibi iş akışlarını sağlamak için UPA'dan gelen tüm istekleri kabul etmemektedir. YTA Konfederasyonu mimarisinin uygulama katmanındaki orkestrasyon tarafından işlenen SFZ bilgilerinin konfederasyon denetleyicisine direk olarak aktarabilinmesi sayesinde servis fonksiyonlarının somutlaştırılması, yapılandırılması ve kullanımdan kaldırılması gibi operasyonlar şeffaf olarak gerçekleştirilmektedir.

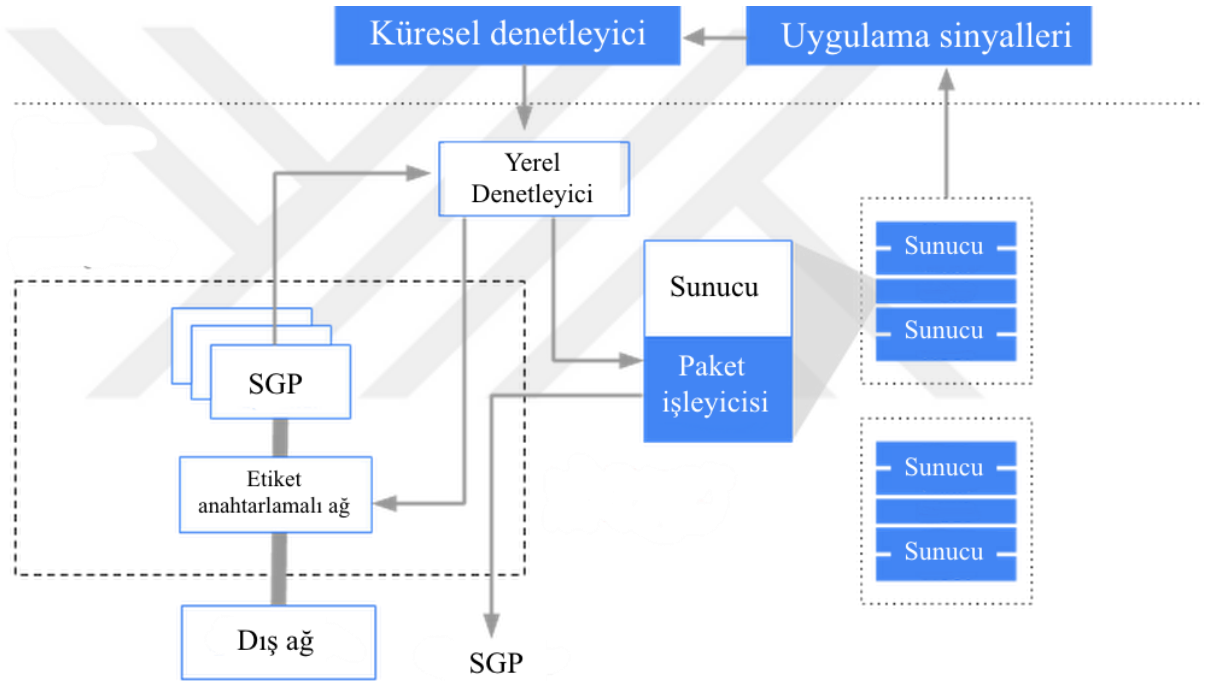


Şekil 10. Juniper Contrail YTA Mimarisi

Uygulama katmanı içinde karşılaşılan uyumsuzluklar trafik akış esnekliğini meydana getirir. Şekil 11’de görülen Google’un Espresso YTA mimarisi artan trafik

⁵⁵ Juniper Networks, *Contrail Cloud Platform Architecture*, https://www.juniper.net/documentation/en_US/contrail-cloud10.0/information-products/pathway-pages/contrail-cloud-platform-architecture-10.0.pdf, Erişim Tarihi: 01/13/2020

talepleri için tasarlanmıştır ve talepleri karşılamada başarılı olmuştur, fakat modülerlik eksikliği ve bazı baştan tahmin edilemeyen mimari kısıtlamalar yüzünden uygulama odaklı olmakta sıkıntılar yaşamıştır⁵⁶. Örnek olarak yaklaşık 100 sunucunun olduğu bir raf ve işlenmesi için her raf başına saniyede 10000 yeni akış olduğunda, YTA denetleyicileri anahtarlara gönderilmek üzere saniyede 1.000.000.000 akış denetimi yapılacaktır. Trafik akışlarının bu şekilde tasarlanması uygulama odaklı yapılandırmayı güçleştirmiştir. YTA Konfederasyonları mimarisinde konfederasyon denetleyicisi ve küme denetleyicisi olmak üzere ikili hiyerarşi vardır ve bu denetleyici başına düşen yükü hafifleterek akış esnekliği sağlamaktadır.



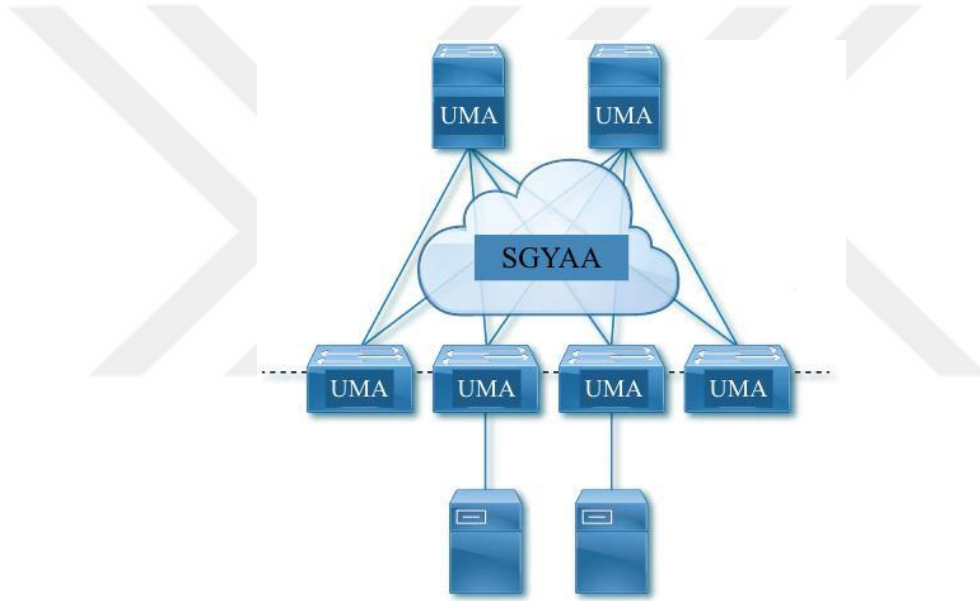
Şekil 11. Google YTA Mimarisi

2.3.4 Tedarikçi bağımsızlığı

Kendi kullanımına özel geliştirilmiş YTA mimarilerinden Google, Facebook ve Amazon YTA mimarileri tedarikçi bağımlılığı ile karşılaşmış ve çözüm olarak

⁵⁶ Yap et al., (August 2017), *Taking the Edge off with Espresso: Scale, Reliability and Programmability for Global Internet Peering*, The Conference of the ACM Special Interest Group, 2017

ağlarının bazı bölümleri için gerekli olan donanımları kendi üretmişlerdir⁵⁷. Ticari YTA mimarilerinde tedarikçi bağımlılığı yaratan yapılar görülmektedir. Örnek olarak Şekil 12’de görülen Cisco UMA YTA yapısı ASGYK kullanarak SYAA ve SGYAA dahil olmak üzere birçok farklı kapsülleme türünü uygulamaktadır. Aşağıdaki şekilde görüldüğü gibi Cisco UMA YTA mimarisinde Cisco SGYAA kapsülleme yöntemini kullanmıştır⁵⁸. Pratikte bu mimaride Cisco SGYAA desteklemeyen başka bir yaprak veya omurga anahtarı kullanılamamaktadır ve bu durum tedarikçi bağımlılığı yaratmaktadır. YTA Konfederasyonları mimarisi Bölüm 2 Kısım 2.2’de örneklendirildiği gibi sadece açık kaynak kodlu arayüzlerin UPA’ları üzerinden tanımlandığı için tedarikçi bağımlılığı yaratmamaktadır.



Şekil 12. Cisco UMA SGYAA Mimarisi

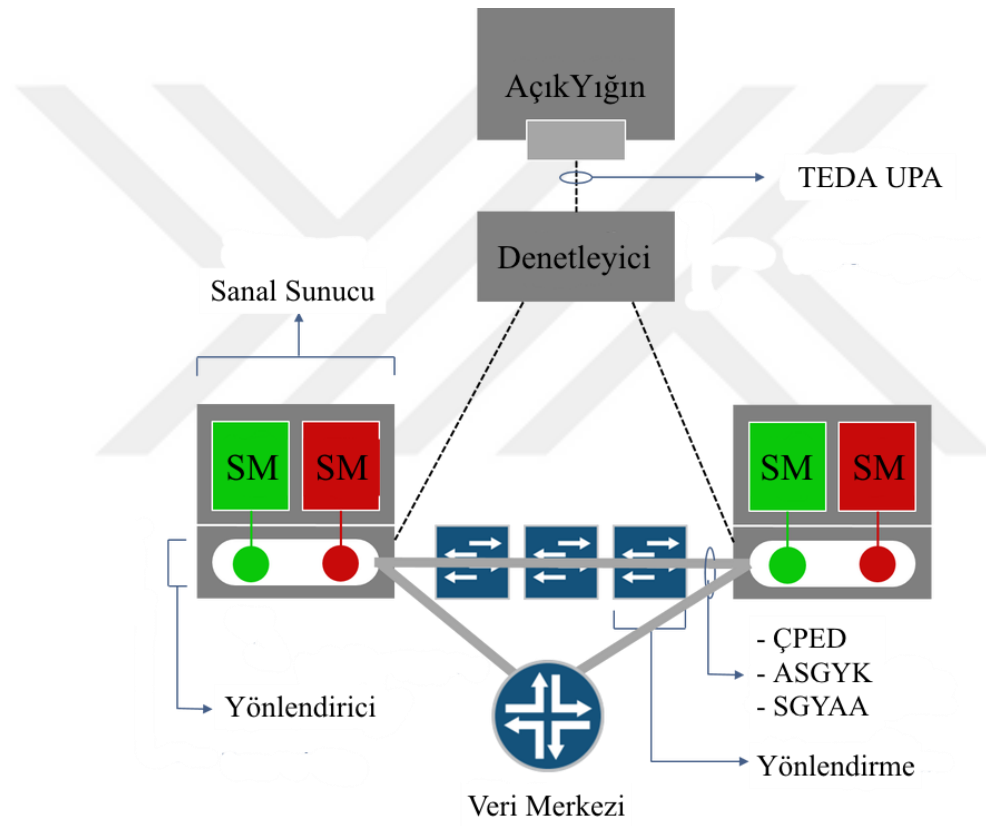
Benzeri durum Nuage Networks YTA mimarisinde de görülmektedir. Mimari ağın etki alanını her sunucunun atandığı politika bölgelerine ayrılmaktadır. Denetleyici tüm SYA araçları üzerinde konfigürasyon gerçekleştirir. Bulut ağına

⁵⁷ Isogai et al., (31 May 2012), *Global-scale experiment on multi-domain Software Defined Transport Network*, The 10th International Conference on Optical Internet (COIN2012), IEEE, 2012

⁵⁸ Cisco, *Cisco ACI Multi-Site Architecture White Paper*, <https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-739609.html>, Erişim Tarihi: 01/14/2020

birden fazla denetleyici eklendiğinde denetleyiciler birbirleriyle konuşması için ASGYK sonlandırılmasını desteklemesi gerekmektedir. AAK denetleyici standartları içinde böyle bir zorunluluk bulunmamaktadır. Dolayısıyla bu durum tedarikçi bağımlılığı getirmektedir.

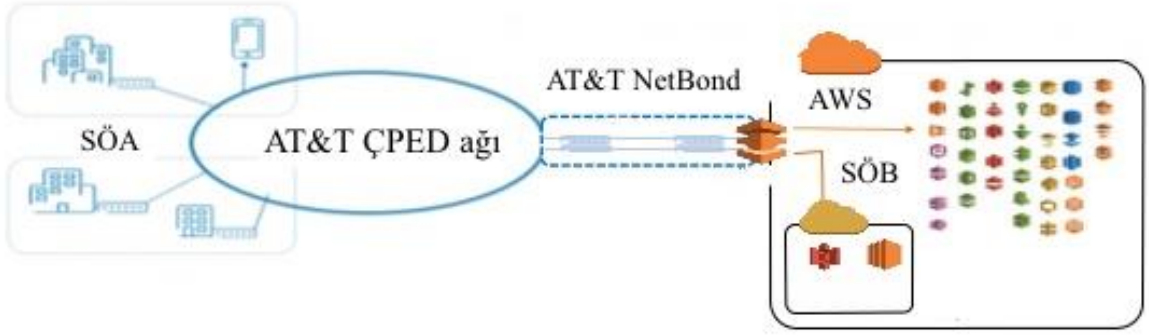
Big Cloud Fabric BBA YTA altyapısı da Dell EMC, EdgeCore beyaz kutu ve HPE Altoline ürünlerini desteklemektedir. Bunların dışında birçok anahtar ile çalışmadığı için mimaride kısmen tedarikçi bağımlılığı vardır.



Şekil 13. Juniper Contrail Kapsülleme Mimarisi

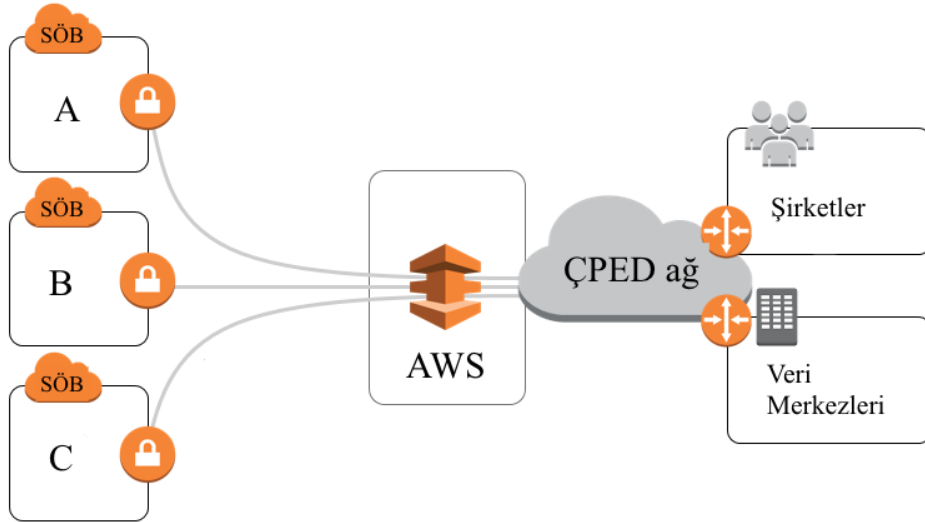
Şekil 13'te görüldüğü üzere Juniper Contrail hibrit yapıyı kısmen destekleyen bir buluttur⁵⁹. ASGYK ve SGYAA gibi ağ kapsülleme standartlarını desteklemektedir, fakat protokol başlığı sadece kendi cihazları ya da partner cihazları ile uyumludur. Bu durum tedarikçi bağımlılığı yaratmaktadır.

⁵⁹ Juniper Networks, *Contrail for Service Providers*, <https://www.juniper.net/us/en/local/pdf/whitepapers/2000529-en.pdf>, Erişim Tarihi: 01/14/2020



Şekil 14. Amazon NetBond YTA Mimarisi

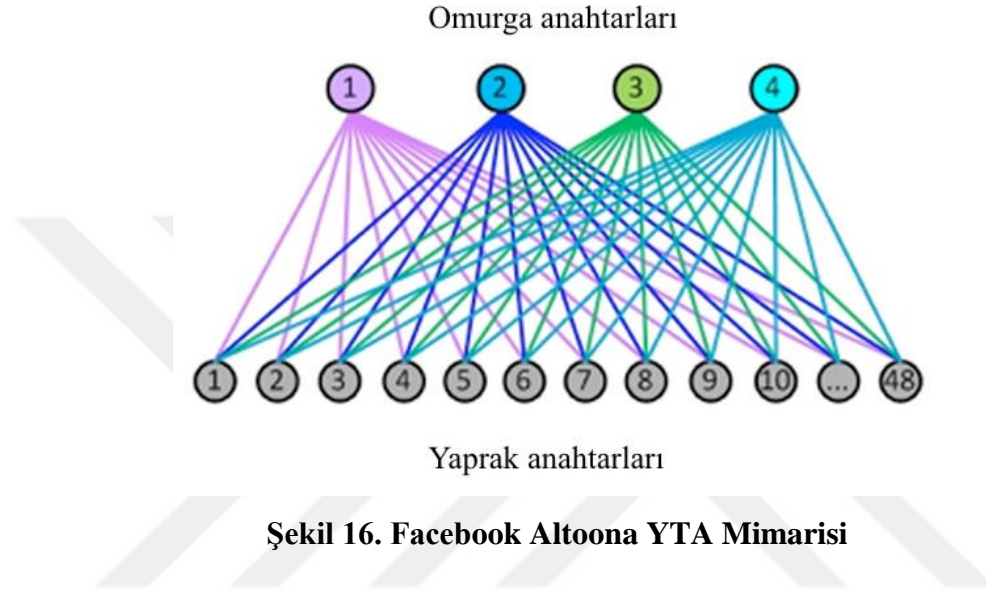
Şekil 14'te görüldüğü gibi Amazon YTA tabanlı mimariler için servis olarak oluşturulan NetBond iş uygulamalarını ÇPED bağlantısı ile sunarak AWS bulut erişimini geliştirmektedir⁶⁰. Beanfield Metroconnect Amerika Birleşik Devletleri ve Kanada'daki müşterilerin ağlarını doğrudan AWS bulutuna bağlamak için gereken arayüzü sunmaktadır. Fakat bu durum tedarikçi ve lokasyon konusunda kısıtlamalar getirmektedir.



Şekil 15. Amazon Verizon YTA Mimarisi

⁶⁰ AWS, *AT&T NetBond for Cloud with Amazon Web Services*, <https://www.business.att.com/content/dam/attbusiness/briefs/cloud-att-netbond-with-aws-brief.pdf>, Erişim Tarihi: 01/14/2020

Amazon sanal makineleri müşteri ağına bağlamak için kaplama ağı (overlay network) teknolojisini kullanmaktadır⁶¹. Kaplama ağları başka bir şebekenin üzerine yerleştirilen ağlardır. Bunlardan bir tanesi de Şekil 15’de görüldüğü gibi Verizon Secure Cloud Interconnect’in ÇPED üzerinden müşterilerinin kurumsal ağlarını AWS’ye bağlandığı şebekedir. Bu uygulama belirli lokasyonlarda tedarikçi kısıtlaması getirmektedir.



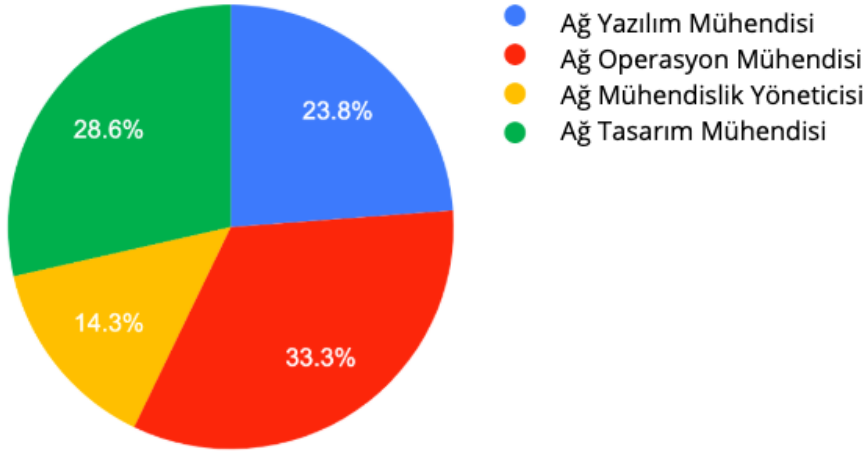
Facebook Altoona YTA mimarisi Şekil 16’da görüldüğü gibi omurga ve yaprak anahtarlarından oluşmaktadır. Facebook’tan internete olan trafik miktarı her geçen gün daha fazla kişi bağlandığında yeni ürün ve servisler oluşturulurken sürekli artmaktadır. Arka planda servis katmanları vardır ve uygulamalar bu katmanlara göre dağıtılmaktadır. Fakat uygulama katmanı ve altyapı katmanı dinamik servis oluşturmak için birbir entegre değildir. YTA Konfederasyonunun küme ve arayüzleri sayesinde konfigürasyon çalışmaları yapı bloklarını ve yönlendirme mantığını tanımlamak için gerekli minimum seviyedeki ayarları kullanarak cihaz düzeyinde yapılmaktadır. Böylece yönlendirme ilkeleri Bölüm 2 Kısım 1.3’te gösterildiği gibi platforma özgü formlara dönüştürümekte ve anahtarlar bu şekilde programlanmaktadır.

⁶¹ AWS, *AWS Network Connectivity Over MPLS*, <https://aws.amazon.com/partners/success/verizon/>, Erişim Tarihi: 01/14/2020

3. YTA KONFEDERASYONU VE BENZERİ ÇÖZÜMLERİN NİTEL DEĞERLENDİRMESİ

3.1. Metod

YTA Konfederasyonun ve en çok benzeyen yedi çözüm mimarisi ile birlikte on beş kategoride, Şekil 17’de görüldüğü gibi ağ yazılım, operasyon, tasarım mühendisi ve yöneticilerinden oluşan kırk üç profesyonelin katıldığı anket aracılığı ile değerlendirilmiştir. Ankette sorular Tam-Kısmi-Özel ya da Mükemmel-Ortalamanın üstünde-Ortalama-Ortalamanın altında-Zayıf seçenekleri ile sunulmuştur.



Şekil 17. YTA Mimari Değerlendirme Anket Katılımcıları

YTA Konfederasyonu EK-1’de verilen ankete göre, Protokol Kapsamından Tam, Kullanılabilirlikten Açık Kaynak Kodlu, Servis Oluşturmadan Tam Dinamik, Uygulama Oluşturmadan Tam Dinamik, Servis ve Uygulama Oluşturmadan Tam, Kapsülleme Yönteminden Tam, Bulut Entegrasyonundan Tam, Trafik Akışı Esnekliğinden Mükemmel, AFS Desteğinden Mükemmel, Kolay Yapılandırma ve Sağlamadan Mükemmel, Modülerlikten Mükemmel, Böl ve Yönet Mekanizmasından Tam, Farklı Donanım Mimarilerinin Kümelenmesinden Tam, YTA Olmayan Bağlantı Desteğinden Tam, Tedarikçi Bağımsızlığından Tam olarak değerlendirilmiştir.

3.2. Ölçütler

YTA Konfederasyonu mimarisi, benzerlik teşkil eden Google'un Jupiter, Facebook'un Altoona, Amazon'un AEBİB, Cisco' nun UMA, Nokia'nın Nuage, Juniper'in Contrail ve Big Swich Networks'un Big Cloud Fabric YTA mimarileri ile ağ yazılım, operasyon, tasarım mühendisi ve yöneticilerinden oluşan uzman grup tarafından karşılaştırılmıştır. Karşılaştırma aşağıdaki tabloda özetlenmiştir.

Tablo 7: YTA Mimarileri Değerlendirme Özeti

YTA Mimarisi	Protokol Desteği	Uygulama Odaklı ve Modüler	Esnek ve Kullanım Kolaylığı	Tedarikçi Bağımsızlığı
YTA Konfederasyonu	<ul style="list-style-type: none"> - AçıkAkış, TTŞ ve SFZ ile uyumlu - AAZ protokol paket standartları ile senkronize - Dış bağlantı desteği - Özelleştirilebilir mimari 	<ul style="list-style-type: none"> - Çoklu ortam omurga-yaprak orkestrasyonu - Servis-küme eşleştirme desteği - Farklı hizmet türlerini yapılandırma desteği 	<ul style="list-style-type: none"> - Hiyerarşik kontrol sistemi - Kolay konfigürasyon ve bakım - Anında servis ve uygulama kurallarını değiştirebilme 	<ul style="list-style-type: none"> - Servis yapılandırma katman soyutlama - Çok tedarikçili karma ortam desteği - AAK deneysel obje tanımlaması - Böl ve yönet küme mekanizması
Google Jupiter	<ul style="list-style-type: none"> - Özelleştirilmiş AçıkAkış desteği - Statik servis ve uygulama yapılandırması - Dış bağlantı desteği - Detaylar şirkete ait gizli bilgi 	<ul style="list-style-type: none"> - Sınırlı uygulama odaklı düzen - Güncellemede çevik olmayan yapı - Araçlarda standart eksikliği - Detaylar şirkete ait gizli bilgi 	<ul style="list-style-type: none"> - Çok karmaşık denetleyici düzlemi - Esnek uygulama yapılandırması için statik mekanizma - Kullanımı kolay yapılandırma araçları - Detaylar şirkete ait gizli bilgi 	<ul style="list-style-type: none"> - Dış bağlantıların olduğu yerlerde tedarikçi bağımlılığı - YTA omurga-yaprak mimarilerinde tamamen bağımsızlık - Detaylar şirkete ait gizli bilgi
Facebook Altoona	<ul style="list-style-type: none"> - Temel kavramlarda geliştirilmiş YTA mimarisi - AAK uyumluluğu yok - Dış bağlantı desteği - Detaylar şirkete ait gizli bilgi 	<ul style="list-style-type: none"> - Tekrarlayan modüler omurga-yaprak mimarisi - Uygulama odaklı servis katmanı - Eş zamanlı uygulama ekleme desteği - Detaylar şirkete ait gizli bilgi 	<ul style="list-style-type: none"> - Kolay ve hızlı veri merkezi kablolama sistemi - Göreceli olarak basit omurga-yaprak mimarisi - Esnek yapılandırma araçları - Detaylar şirkete ait gizli bilgi 	<ul style="list-style-type: none"> - Dış bağlantıların olduğu yerlerde tedarikçi bağımlılığı - YTA omurga-yaprak mimarilerinde kısmen bağımsızlık - Detaylar şirkete ait gizli bilgi
Amazon AEBİB	<ul style="list-style-type: none"> - AçıkYığın destekçisi - Dış bağlantı desteği - Detaylar şirkete ait gizli bilgi 	<ul style="list-style-type: none"> - Araçlarda standart eksikliği - Güncellemede çevik olmayan yapı - Detaylar şirkete ait gizli bilgi 	<ul style="list-style-type: none"> - Çok karmaşık denetleyici düzlemi - Esnek uygulama yapılandırması için statik mekanizma - Detaylar şirkete ait gizli bilgi 	<ul style="list-style-type: none"> - ASGYK, SGYAA ve ÇPED kapsülleme problemi - Detaylar şirkete ait gizli bilgi
Cisco UMA	<ul style="list-style-type: none"> - Oplex protokol suit desteği var - AAK uyumluluğu yok - Cisco NX-OS ile tek başına modunda çalışan konfigürasyonlar - Dış bağlantı desteği 	<ul style="list-style-type: none"> - Uygulama odaklı fonksiyonellikte sınırlı - Bulut servis entegrasyon problemi - Uygulama ve altyapı katmanı uyumsuzluğu 	<ul style="list-style-type: none"> - Farklı kapsülleme türleri için uyumlu - Transparan uç nokta grupları desteği - Kolay güncelleme ve büyüme fonksiyonu 	<ul style="list-style-type: none"> - Açık kaynak kod kullanan tedarikçilerle uyumlu - Oplex kullanan tedarikçilerle kısmen uyumlu mimari - Cisco SGYAA problemi

Nokia Nuage	- Kısmen AçıkAkış ile uyumlu - ÇPED protokolü desteği - Dış bağlantı desteği	- Open vSwitch tasarımı servis kapsülleme desteği - Uygulama alanlarını SÖA'lara ayırarak yönetme - SHR modülü TEDA UPA ile uyumsuzluğu	- SHR fonksiyonu ile servis dağıtım yapılandırması - SYA fonksiyonunun uygulama katmanı ile uyumsuzluğu - Altyapı katman uyumsuzluğu	- Tedarikçi bağımlılığı yaratan mimari - Açık kaynak kod desteği yok - ASGYK sonlandırılması problemi
Juniper Contrail	- İç ve dış bağlantı desteği - Kapsülleme protokol desteği - Standart açık kodlar ile uyumlu - Kısıtlı TEDA UPA desteği	- Ağ sanallaştırma mimarilerini uygulamak için yapılandırma şablonları - Hibrit bulut platformunda uygulama taşınabilirliği	- Sanallaştırılmış ağ fonksiyonları için dinamik servis zinciri - Ağ fonksiyonlarını açık ağ standartları ile birbirine bağlayan yapı - Kolay altyapı kullanım sistemi	- Tedarikçi bağımlılığı yaratan mimari - Apache v2.0 açık kaynaklı lisans desteği - ASGYK ve SGYAA kapsülleme problemi
Big Switch Networks Big Cloud Fabric	- AçıkAkış ile uyumlu - İç ve dış bağlantı desteği	- Modüler uygulama şablon desteği - Uygulama odaklı TEDA UPA uyumu - Şablonların altyapı katmanı ile uyumsuzluğu	- K2 / K3 ağı için kolay kullanım - Ağı kademeli olarak büyümeye odaklı esnek YTA mimari tasarımı	- Dell EMC, EdgeCore white box ve HPE Altoline ürün desteği - HPE platformu olarak SimpliVity ile entegre çalışma

Mimariler, Protokol Kapsamı, Kullanılabilirlik, Servis Oluşturma, Uygulama Oluşturma, AFS Desteği, Servis ve Uygulama Entegrasyonu, Kolay Yapılandırma ve Sağlama, Modülerlik, Böl ve Yönet Mekanizması, Kapsülleme Yöntemi, Farklı Donanım Mimarilerinin Kümelenmesi, Tedarikçi Bağımsızlığı, Trafik Akışı Esnekliği, Bulut Entegrasyonu, YTA Olmayan Bağlantı Desteği ölçütleri ile nitel olarak anket yoluyla değerlendirilmiştir. Değerlendirme ölçütleri aşağıda açıklanmıştır:

- Protokol Kapsamı, bir YTA ağı üzerinden iki veya daha fazla cihaz arasındaki iletişimi tanımlayan kurallar ve prosedürler oluşan resmi standartlar ve politikalarıdır. Bu ölçüt için cevap seçeneği ankette Tam, Kısmi, Özel seçenekleri ile sunulmaktadır.

- Kullanılabilirlik, mimarinin ne şekilde sunulduğunu ölçmektedir. Bu ölçüt için cevap seçeneği ankette Açık Kaynak Kodlu, Ücretli, Özel seçenekleri ile sunulmaktadır.

- Servis Oluřturma, bir ađın alıřmasını kolaylařtıran özelliktir. Genellikle, ađın ASA modelindeki uygulama katmanında alıřan ađ protokollerine dayanan bir sunucu tarafından sađlanmaktadır. Bu lut iin cevap seeneđi ankette Tam Dinamik, Kısmi Dinamik, Statik, Yok seenekleri ile sunulmaktadır.

- Uygulama Oluřturma, ađdaki bir noktadan diđerine veri aktarımı yapmak iin İnternet veya diđer ađ donanım altyapısını kullanma yeteneđidir. Bu lut iin cevap seeneđi ankette Tam Dinamik, Kısmi Dinamik, Statik, Yok seenekleri ile sunulmaktadır.

- Servis ve Uygulama Entegrasyonu, tm bulut kmelerini ve bulut ierisindeki sanal makineleri, trafik sınıflandırma gereksinimlerine gre alıřan uygulamalarla btnleřtirme yeteneđidir. Oluřturulan uygulamaların temel servislerle eřleřtirilmesine izin veren iřlevdir. Bu lut iin cevap seeneđi ankette Tam, Kısmi, Yok seenekleri ile sunulmaktadır.

- Kapslleme Yntemi, bir protokolden veri alma ve bařka bir protokole evirme iřlemidir, bylece veriler bir ađda devam edebilir. Bu lut iin cevap seeneđi ankette Tam, Kısmi, Yok seenekleri ile sunulmaktadır.

- Bulut Entegrasyonu, gerek zamanlı veri ve iřlem alıřveriři iin eřitli uygulamaları, sistemleri, depoları ve ortamları birbirine bađlayan sistemdir. Bu lut iin cevap seeneđi ankette Tam, Kısmi, Yok seenekleri ile sunulmaktadır.

- Trafik Akıřı Esnekliđi, trafik akıřı kurallarını tanımlama ve bunları nceden tanımlanmıř servislere, uygulamalara ve kmelere atama yeteneđidir. Bu lut iin cevap seeneđi ankette Mkemmel, Ortalamanın stnde, Ortalama, Ortalamanın altında, Zayıf seenekleri ile sunulmaktadır.

- AFS Desteđi, ađ servisleri oluřturan, dađıtan ve iřleten yeni nesil yontemdir. Ađ fonksiyonlarını tescilli donanım cihazlarından ayırma iřlemidir. Bu olđüt için cevap seęeneđi ankette Mükemmel, Ortalamanın üstünde, Ortalama, Ortalamanın altında, Zayıf seęenekleri ile sunulmaktadır.

- Kolay Yapılandırma ve Sađlama, bir ađın kontrollerini, akıřını ve ęalıřmasını YTA bulutunun ađ iletiřimini kolayca desteklemek için ayarlama iřlemidir. Bu olđüt için cevap seęeneđi ankette Mükemmel, Ortalamanın üstünde, Ortalama, Ortalamanın altında, Zayıf seęenekleri ile sunulmaktadır.

- Modülerlik, bađımsız bileřenlerden veya birleřtirilebilecek parçalardan oluřan sistem bileřenlerinin derecesidir. Bu olđüt için cevap seęeneđi ankette Mükemmel, Ortalamanın üstünde, Ortalama, Ortalamanın altında, Zayıf seęenekleri ile sunulmaktadır.

- Böl ve Yönet Mekanizması, bir ađ mimarisinin ęok dallı özyinelemesine dayanan paradigmayı tasarlayan yontemdir. Bu olđüt için cevap seęeneđi ankette Tam, Kısmi, Yok seęenekleri ile sunulmaktadır.

- Farklı Donanım Mimarilerinin Kümelenmesi, donanım gruplarının farklı özelliklere sahip ęalıřma ortamlarının tek bir sistem olarak görünmesini sađlayacak řekilde birbirine bađlanmasını sađlayan yontemdir. Bu olđüt için cevap seęeneđi ankette Tam, Kısmi, Yok seęenekleri ile sunulmaktadır.

- YTA Olmayan Bađlantı Desteđi, YTA yeteneđi olmayan bir kümeye bađlanma yeteneđidir. Bu olđüt için cevap seęeneđi ankette Tam, Kısmi, Yok seęenekleri ile sunulmaktadır.

- Tedarikçi Bađımsızlıđı, farklı donanım veya yazılım platformlarını seęebilme özgürlüğünü sađlayan bir özelliktir. Bu olđüt için cevap seęeneđi ankette Tam, Kısmi, Yok seęenekleri ile sunulmaktadır.

3.3. Matris

YTA hakkında birçok açık kaynak destekli standart ve kullanım durumları yayınlanmıştır. Literatür tarama bölümünde bahsedildiği üzere bu standart ve kullanım durumlarının ortak yanları olduğu gibi, birbirleriyle çelişen, beraber çalışmayan, katman içi ve arası uyumlu olmayan durumları da vardır. Buna ek olarak özellikle telekomünikasyon şirketlerinin ihtiyaç duyduğu sistemin baştan sona çalışabilmesi için gerekli olan birçok nokta için standartlaşmış açık kaynak destekli çözümler bulunmamaktadır. Ticari YTA ürünleri bu noktalara çözüm getirirken pazar paylarını kaybetmemek için tedarikçi bağımlılığı yaratmışlardır. YTA birçok ağ alanında hem akademi hem de endüstriden büyük ilgi görmüştür. Bölüm 2, Kısım 2.3'te YTA Konfederasyonları mimarisi ve benzeri mimariler açık kaynak kod desteği, katman içi, katman arası ve tedarikçi bağımsızlığı konularında literatürde var olan kaynaklar doğrultusunda karşılaştırılmıştır. Bu kısımda Tablo 8'de özetlenen endüstri liderleriyle yapılmış anket sonuçlarının literatür baz alınarak yapılan değerlendirmelerdeki karşılığı sunulmuştur.

YTA Konfederasyonları mimarisinin anket sonuçlarına göre özel ve ticari mimari çözümlere kıyasla protokol kapsamı tamdır ve kullanılabilirlik durumu açık kaynak koddur. Buna paralel olarak Bölüm 2 Kısım 2.1'de gösterilen TEDA UPA'lar ve AçıkAkış deneyci mesajları protokol desteği konusunda tamamen açık kaynak kod ile tanımlandığından dolayı standartlar doğrultusunda protokol desteği tamdır. Bu husustaki literatüre ait karşılaştırma ayrıntıları Bölüm 2 Kısım 2.3.1'de verilmiştir.

Tablo 8: YTA Mimarileri Değerlendirme Matrisi

Mimari	Google Jupiter	Facebook Altoona	Amazon AEBİB	Cisco UMA	Nokia Nuage	Juniper Contrail	Big Switch Networks Big Cloud Fabric
Ölçüt							
Protokol Kapsamı	Özel	Özel	Özel	Kısmi	Kısmi	Kısmi	Kısmi
Kullanılabilirlik	Özel	Özel	Özel	Ücretli	Ücretli	Ücretli	Ücretli
Servis Oluşturma	Kısmi Dinamik	Kısmi Dinamik	Kısmi Dinamik	Statik	Yok	Kısmi Dinamik	Kısmi Dinamik
Uygulama Oluşturma	Kısmi Dinamik	Kısmi Dinamik	Tam Dinamik	Statik	Statik	Kısmi Dinamik	Kısmi Dinamik
Servis ve Uygulama Entegrasyonu	Kısmi	Kısmi	Tam	Kısmi	Yok	Kısmi	Kısmi
Kapsülleme Yöntemi	Kısmi	Kısmi	Tam	Kısmi	Yok	Kısmi	Yok
Bulut Entegrasyonu	Tam	Tam	Tam	Kısmi	Kısmi	Kısmi	Kısmi
Trafik Akışı Esnekliği	Ortalamanın üstünde	Ortalamanın üstünde	Ortalamanın üstünde	Ortalama	Ortalamanın altında	Ortalama	Ortalamanın altında
AFS Desteği	Ortalamanın üstünde	Ortalamanın üstünde	Ortalamanın üstünde	Ortalamanın altında	Ortalamanın altında	Ortalama	Ortalama
Kolay Yapılandırma ve Sağlama	Ortalamanın üstünde	Ortalamanın üstünde	Ortalamanın üstünde	Ortalama	Ortalamanın altında	Ortalama	Ortalama
Modülerlik	Ortalamanın üstünde	Ortalamanın üstünde	Mükemmel	Ortalama	Ortalamanın altında	Ortalama	Ortalama
Böl ve Yönet Mekanizması	Kısmi	Kısmi	Kısmi	Yok	Yok	Kısmi	Yok
Farklı Donanım Mimarilerinin Kümelenmesi	Tam	Kısmi	Kısmi	Yok	Yok	Kısmi	Yok
YTA Olmayan Bağlantı Desteği	Kısmi	Kısmi	Kısmi	Yok	Yok	Kısmi	Kısmi
Tedarikçi Bağımsızlığı	Kısmi	Kısmi	Kısmi	Kısmi	Yok	Kısmi	Kısmi

Yine anket sonuçlarına göre YTA Konfederasyonu mimarisinin servis oluşturma ve uygulama oluşturma kabiliyeti tam dinamik, servis ve uygulama entegrasyonu ise tamdır. Diğer mimarilere bakacak olursak özel YTA mimarilerinin bu konuda kısmi desteğinin olduğu görülmektedir. Bunun sebebi özel mimarilerin ihtiyaç duyduğu kullanım durumlarının kısıtlı olmasıdır. Ticari mimarilere bakarsak da bazıları bu özellikleri ya kısmi olarak desteklemekte ya da hiç desteklememektedir.

Bölüm 2 Kısım 2.3.3'te anlatıldığı gibi bu parametreler uygulama katmanı içindeki uyum kapsamı altındadır. YTA Konfederasyonu mimarisin uygulama katmanındaki çalışma ayrıntıları Bölüm 2 Kısım 2.1.1'de verilmiştir.

Kapsülleme yöntemi ve bulut entegrasyonu hem katmanlar içi hem de katmanlar arası uyumu gerektirmektedir. Özel ve ticari mimarilerden bazıları kapsülleme yöntemini kısmi olarak desteklerken, bulut entegrasyonunu özel mimariler tam, ticari mimariler ise kısmi olarak desteklemektedir. Bu konuda YTA Konfederasyonu mimarisi Bölüm 2 Kısım 2.3.2 ve 2.3.3'te anlatıldığı üzere tam uyumludur, ayrıntılar ise Bölüm 2 Kısım 2.1'de anlatılmıştır.

Anket parametrelerinin bazılarının YTA bulut uygulamasında birbirleriyle kesiştiği görülmektedir. Trafik akış esnekliği, AFS desteği, kolay yapılandırma ve modülerlik ölçütlerinin mükemmel derecede çalışabilmesi açık kaynak kod desteği, katmanlar içi ve arası uyum özelliklerinin olmasını gerektirir. Anket sonuçlarına göre bu konuda özel mimariler mükemmel yada ortalamanın üstünde değer alırken, ticari mimariler ortalama ya da ortalamanın altında değer almışlardır. Bu ölçütlerde mükemmel değer alan YTA Konfederasyonu mimarisinin bulut içinde nasıl çalıştığına dair ayrıntılar Bölüm 2, Kısım 2.2'de açıklanmıştır. Diğer YTA mimarileri ile karşılaştırmaya ait kullanım durumları Bölüm 2, Kısım 2.3.1, 2.3.2 ve 2.3.3'te incelenmiştir.

Böl ve yönet mekanizması, farklı donanım mimarilerin kümelenmesi, YTA olmayan bağlantı desteği ve tedarikçi bağımsızlığı, tüm katmanların kendi içlerinde ve aralarında uyumlu çalışmasına ek olarak, açık kaynak kod desteği de gerektirmektedir. YTA Konfederasyonu mimarisi böl ve yönet mekanizması ve farklı donanım mimarilerinin kümelenmesi ölçütlerinden tam değer almıştır ve bu konuya ait ayrıntılar Bölüm 2 Kısım 2.1.3 ve 2.2.2'de anlatılmıştır. Tedarikçi bağımsızlığı konusu başlı başına Bölüm 2, Kısım 2.3.4'te ele alınmıştır ve anket sonuçlarından da görüldüğü üzere YTA konfederasyonu mimarisi bu özelliği tam desteklemekte, diğer mimariler ise ya kısmi desteklenmekte ya da hiç desteklenmemektedir.

SONUÇ

Bulut bilgi işlem, YTA ve ağ sanallaştırma veri merkezlerinin ağa bağlanma şeklini tamamen değiştirmiştir. Veri merkezleri arasındaki bağlantılar da değişirken, esnek ve uygun maliyetli bağlantı sağlamak için YTA kontrolünde yalın metal, beyaz kutu ve markalı yalın metal platformlarını kullanma eğiliminde bir artış görülmektedir. YTA dünyasındaki hızlı değişiklikler operatörlerin ağa farklı özelliklere sahip birçok cihaz eklemelerine neden olmuştur. Standartları geliştiren organizasyonlar bu değişim hızına adapte olmakta zorluk çekmişlerdir. Çünkü bir standart oluşturmak çok zaman almakta ve o zaman içerisinde standart geliştirilen teknoloji yerini yenisine bırakmaktadır.

YTA teknolojileri zamanla olgunlaşırken farklı yapılar ve platformlar için uygulama ve servis bütünlüğü konularında boşluklar oluşmuştur. Bu boşlukları YTA mimarisine yatırım yapan şirketler kendi ağ yapılarına uygun bir şekilde çözmüştür, fakat artan internet hızı ile zamanla talepler değişmiş ve yeni ihtiyaçlar ortaya çıkmıştır. Bu ihtiyaçlardan en önemlisi servis ve uygulamaları bulut ile entegre bir şekilde farklı YTA mimarilerini içeren kümeler ve YTA olmayan kümeler ile çalıştıran bir çözüm mimarisidir. Literatürde buna en yakın çalışma EEME tarafından geliştirilen SFZ mimarisidir. SFZ'nin amacı ağ operatörlerinin bir servis mimarisi oluşturmalarını ve ağ üzerinde bir servis fonksiyonu yolunu başlatmalarını sağlayacak bir dizi mimari yapı bloğu geliştirmektir, fakat SFZ mimarisi farklı özelliklerdeki YTA kümeleri için gereken akış esnekliğine dair bir çözüm içermemektedir. Bu çalışmada YTA Konfederasyonu mimarisi altyapı yeteneklerini servis fonksiyonu zinciri ile eşleştiren, uygulamaya dayalı bir ağ mimarisi olarak önerilmiştir.

YTA Konfederasyonları mimarisinin anket sonuçlarına ve değerlendirmelere göre özel ve ticari mimari çözümlere kıyasla TEDA UPA'lar ve AçıkAkış deneyici mesajlarının açık kaynak standartlar ile tanımlanmasından dolayı protokol desteği tamdır. YTA Konfederasyonu mimarisinde tam olarak desteklenen kapsülleme yöntemi ve bulut entegrasyonu hem katmanlar içi hem de katmanlar arası uyumu sağlamaktadır. Özel ve ticari mimarilerden bazıları kapsülleme yöntemini kısmi olarak

desteklemekte, bulut entegrasyonunu özel mimariler tam, ticari mimariler ise kısmi olarak desteklemekle beraber tedarikçi bağımlılığı yaratmaktadırlar.

YTA Konfederasyonu mimarisi farklı karakterdeki anahtarları servis zinciriyle entegre ederek operatörlerin farklı kullanım durumları için ağlarını kümelere bölmeye izin vermektedir. Bu fonksiyon ağ mimarisine modülerlik kazandırırken 5N gibi yatay ağ dilimlemeye ihtiyaç duyan teknolojilerin ihtiyaç duyduğu altyapıyı da oluşturmaktadır. Donanım konusundaki gelecek keşiflere paralel olarak 5N' den 6. Nesil Kablosuz Teknoloji'ye (6N) geçiş aşamasında mimarinin yatay dilimleme fonksiyonuna ek olarak dikey dilimleme fonksiyonunu da desteklemesi gerekmektedir. Dikey ağ dilimleme fonksiyonunun YTA Konfederasyonu mimarisine kazandırılması gelecek çalışmalar kapsamındadır.

KAYNAKÇA

İNTERNET KAYNAKLARI

Andreyev A., **Introducing data center fabric, the next-generation Facebook data center network**, <https://engineering.fb.com/production-engineering/introducing-data-center-fabric-the-next-generation-facebook-data-center-network/>, Erişim Tarihi: 04/08/2018

AWS, **AT&T NetBond for Cloud with Amazon Web Services**, <https://www.business.att.com/content/dam/attbusiness/briefs/cloud-att-netbond-with-aws-brief.pdf>, Erişim Tarihi: 01/14/2020

AWS, **AWS Network Connectivity Over MPLS**, <https://aws.amazon.com/partners/success/verizon/>, Erişim Tarihi: 01/14/2020

AWS, **AWS Well-Architected Framework**, https://d1.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf, Erişim Tarihi: 04/08/2019

Barnett T, Jain S., Sumits A., Andra U., Khurana T., **Cisco Global Cloud Index 2015–2020**, https://www.cisco.com/c/dam/m/en_us/service-provider/ciscoknowledgenetwork/files/622_11_15-16-Cisco_GCI_CKN_2015-2020_AMER_EMEAR_NOV2016.pdf, Erişim Tarihi: 08/11/2019

Big Switch Networks, **Big Cloud Fabric Cloud-First Networking Delivers Self-Service Automated Fabric with VPCs ON-PREM**, https://www.bigswitch.com/sites/default/files/datasheet_-_bcf-ec-v5.2.pdf, Erişim Tarihi: 04/01/2020

Big Switch Networks, **Big Cloud Fabric Next Generation Data Center Switching Platform**, http://go.bigswitch.com/rs/974-WXR-561/images/What_is_BCF_WP_JAN17.pdf, Erişim Tarihi: 01/13/2020

Cisco, **Cisco ACI Multi-Site Architecture White Paper,**

<https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-739609.html>, Erişim Tarihi: 01/14/2020

Cisco, **Getting Started with Cisco Application Centric Infrastructure (ACI) in the Small-to-Midsize Commercial Data Center Cisco Validated Design,**

<https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/guide-c07-733638.pdf>, Erişim Tarihi: 04/08/2018

Cisco, **Transform Your Business and Protect Your Cisco Nexus Investment While Adopting Cisco Application Centric Infrastructure,**

<https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-731822.pdf>, Erişim Tarihi: 01/12/2020

Halpern J., Pignataro C., **Service Function Chaining (SFC) Architecture,**

<https://tools.ietf.org/html/rfc7665>, Erişim Tarihi: 05/11/2019

Juniper Networks, **Contrail Cloud Platform Architecture,**

https://www.juniper.net/documentation/en_US/contrail-cloud10.0/information-products/pathway-pages/contrail-cloud-platform-architecture-10.0.pdf, Erişim Tarihi: 01/13/2020

Juniper Networks, **Contrail Architecture,**

<https://www.juniper.net/us/en/local/pdf/whitepapers/2000535-en.pdf>, Erişim Tarihi: 04/01/2020

Juniper Networks, **Contrail for Service Providers,**

<https://www.juniper.net/us/en/local/pdf/whitepapers/2000529-en.pdf>, Erişim Tarihi: 01/14/2020

- Meyer D., **Standards remain ongoing open source challenge for operators**,
<https://www.rcrwireless.com/20170125/carriers/standards-remain-ongoing-open-source-challenge-for-operators-tag2-tag99>, Erişim Tarihi: 09/02/2019
- Nuage Networks, **Unconstrained Datacenter Networks For The Cloud Era**,
<https://onestore.nokia.com/asset/165438>, Erişim Tarihi: 04/01/2020
- Nuage Networks, **Nuage Networks Product Architecture**,
<https://www.nuagenetworks.net/wp-content/uploads/2015/08/NuageNetworks-Architecture-White-Paper.pdf>,
Erişim Tarihi: 01/12/2020
- ONF, **OpenFlow Switch Specification Version 1.3.2**,
<https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.3.2.pdf>, Erişim Tarihi: 07/11/2019
- ONF, **OpenFlow Switch Specification Version 1.4.0**,
<https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.4.0.pdf>, Erişim Tarihi: 07/11/2019
- ONF, **OpenFlow Switch Specification Version 1.5.1**,
<https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf>, Erişim Tarihi: 07/11/2019
- ONF, **OpenFlow Table Type Patterns**, <https://www.opennetworking.org/wp-content/uploads/2013/04/OpenFlow%20Table%20Type%20Patterns%20v1.0.pdf>,
Erişim Tarihi: 07/12/2018
- ONF, **Simplifying OpenFlow Interoperability with Table Type Patterns (TTP)**,
<https://www.opennetworking.org/wp-content/uploads/2014/10/sb-TTP.pdf>,
Erişim Tarihi: 08/11/2019
- ONF, **Software-Defined Networking: The New Norm for Networks**,
<https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf>, Erişim Tarihi: 05/03/2017

ONF, **TR-518 Relationship of SDN and NFV**,

https://www.opennetworking.org/wp-content/uploads/2014/10/onf2015.310_Architectural_comparison.08-2.pdf,
Erişim Tarihi: 09/12/2019

Singh A., Ong J., Agarwal A., Anderson G., Armistead A., Bannon R., Boving S.,
Desai G., Felderman B., Germano P., Kanagala A., Provost J., Simmons J.,
Tanda E., Wanderer J., Hölzle U., Stuart S., Vahdat A., **Jupiter Rising: A
Decade of Clos Topologies and Centralized Control in Google's
Datacenter Network**,

<https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43837.pdf>, Erişim Tarihi: 04/08/2018

YAYINLANMIŞ BİLDİRİLER

Alenezia M., Almustafa K., Meerj K. A., (March 2019), **Cloud based SDN and
NFV architectures for IoT infrastructure**, Egyptian Informatics Journal,
ScienceDirect, 2019, 1-10

Azodolmolky S., Wieder P., Yahyapour R., (27 June 2013), **SDN-based cloud
computing networking**, 15th International Conference on Transparent
Optical Networks (ICTON), IEEE, 2013

Bakker J. N., Welch I., Seah W. K. G., (10 November 2016), **Network-wide virtual
firewall using SDN/OpenFlow**, IEEE Conference on Network Function
Virtualization and Software Defined Networks (NFV-SDN), IEEE, 2016

Bauer R., Milford R., Zhen L., (24 November 2014), **Aligning Technology and
Market Drivers in an Open Source Standards Testing Program**,
Computer, IEEE, 2014, 30-36

- Barakabitze A. A., Ahmad A., Mijumbi R., Hines A., (11 February 2020), **5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges**, Computer Networks, ScienceDirect, 2020
- Blenk A., Basta A., Kellerer W., (15 May 2015), **HyperFlex: An SDN virtualization architecture with flexible hypervisor function allocation**, IFIP/IEEE International Symposium on Integrated Network Management (IM), IEEE, 2015
- Bora G., Bora S., Singh S., Arsalan S. M., **OSI Reference Model: An Overview**, International Journal of Computer Trends and Technology, 2014
- Canini M., Feldmann A., Levin D, Schaffert F., Schmid S., (24 November 2014), **Software-Defined Networks: Incremental Deployment with Panopticon**, Computer, IEEE, 2014, 56-60
- Chowdhury N. M., Boutaba R, (31 July 2009), **Network virtualization: a state of the art and research challenges**, IEEE Communications Magazine, IEEE, 2009, 20-26
- Gharakheili H. H., Bass J., Exton L., Sivaraman V., (19 June 2014), **Personalizing the home network experience using cloud-based SDN**, Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, IEEE, 2014
- Isogai A, Fukuda A., Masuda A., Hiramatsu A., (31 May 2012), **Global-scale experiment on multi-domain Software Defined Transport Network**, The 10th International Conference on Optical Internet (COIN2012), IEEE, 2012
- Halpern J., (December 2014), **Standards collisions around SDN**, IEEE Communications Magazine, IEEE, 2014, 10-15
- Hawilo H., Shami A., Mirahmadi M., Asal R., (December 2014), **NFV: State of the Art, Challenges and Implementation in Next Generation Mobile Networks (vEPC)**, IEEE Network, IEEE, 2014, 18-26

- Lopez V., Gonzalez de Dios O., Fernandez-Palacios J. P., (7 March 2019), **Whitebox Flavors in Carrier Networks**, Optical Fiber Communications Conference and Exhibition (OFC), IEEE, 2019
- Kreutz D., Ramos F. M. V., Verissimo P. E., Rothenberg C. E., Azodolmolky S., Uhlig S., (20 May 1998), **Software-Defined Networking: A Comprehensive Survey**, MELECON '98. 9th Mediterranean Electrotechnical Conference, IEEE, 2002
- Kúzniar M., Perešini P., Kostić D, (04 March 2015), **What you need to know about SDN control and data planes**, International Conference on Passive and Active Network Measurement, 2015, 347-359
- Long F., Sun Z., Zhang Z., Chen H., Liao L., (20 May 2012), **Research on TCAM-based Openflow switch platform**, International Conference on Systems and Informatics (ICSAI), IEEE, 2012
- McKeown N., Anderson T. E., Anderson T., Balakrishnan H., Parulkar G., Peterson L. L., Rexford J., Shenker S. J., Turnere J., (March 2008), **OpenFlow: enabling innovation in campus networks**, ACM SIGCOMM Computer Communication Review, ACM, 2008
- Ordenez-Lucena J., Ameigeiras P., Lopez D., Ramos-Munoz J. J., Lorca J., Folgueira J., (12 May 2017), **Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges**, IEEE Communications Magazine, IEEE, 2017, 80-87
- Rotsos C., King D., Farshad A., Bird J., Fawcett L., Georgalas N., Gunkel M., Shiomoto K, Wang A., Mauthe A., Race N., Hutchison D., (November 2017) **Network service orchestration standardization: A technology survey**, Computer Standards & Interfaces, ScienceDirect, 2017, 203-215
- Rothenberg C. E., Chua R., Bailey J., Winter M., Corrêa C. N. A., C. de Lucena S., Salvador M. R., Nadeau T. D., (24 November 2014), **When Open Source Meets Network Control Planes**, Computer, IEEE, 2014, 46-54

- Sandhya, Sinha Y., Haribabu K., (15 December 2017), **A survey: Hybrid SDN**, Journal of Network and Computer Applications, ScienceDirect, 2017, 35-55
- Schaller S., Hood D., (November 2017), **Software defined networking architecture standardization**, Computer Standards & Interfaces, ScienceDirect, 2017, 197-202
- Sefraoui O., Aissaoui M., Eleluldj M., (October 2012), **OpenStack: Toward an Open-Source Solution for Cloud Computing**, International Journal of Computer Applications, 2012, 38-42
- Snow N. A., Dasari V. R., Geerhart B. E., (September 2018), **OpenFlow Experimenter Labels for Encoding Adaptive Network Functions**, IEEE 39th Sarnoff Symposium, IEEE, 2018
- Quinn P., Guichard J., (24 November 2014), **Service Function Chaining: Creating a Service Plane via Network Service Headers**, Computer, IEEE, 2014, 38-44
- Tourrilhes J., Sharma P., Banerjee S., Pettit J., (24 November 2014), **SDN and OpenFlow Evolution: A Standards Perspective**, Computer, IEEE, 2014, 22-29
- Trajkovska I., Alexandros M., Christos K., Denis S., João B., Piyush S., George H., Thomas X., Bohnert M., Koumaras H., (November 2017), **SDN-based service function chaining mechanism and service prototype implementation in NFV scenario**, Computer Standards & Interfaces, ScienceDirect, 2017, 247-265
- Van der Meer S., Grasa E., **SDN Architectural Limitations: Towards a Full Software Network Vision**, IEEE Softwarization, IEEE, 2016
- Yap K., Motiwala M., Rahe J., Padgett S., Holliman M., Baldus G., Hines M., Kim T., Narayanan A., Jain A., Lin V., Rice C., Rogan B., Singh A., Tanaka B., Verma M., Sood P., Tariq M., Tierney M., Trumic D., Valancius V., Ying C., Kallahalla M., Koley B., Vahdat A., (August 2017), **Taking the Edge off**

with Espresso: Scale, Reliability and Programmability for Global Internet Peering, The Conference of the ACM Special Interest Group, 2017

Zhang H., Guo X., (29 November 2014), **SDN-based load balancing strategy for server cluster**, IEEE 3rd International Conference on Cloud Computing and Intelligence Systems, IEEE, 2014

TEZLER

Chupala P., **Overseer: SDN-Assisted Bandwidth and Latency Aware Route Optimization based on Application Requirement**, Yüksek Lisans Tezi, Nara Institute of Science and Technology, 2015

Foresta F., **Integration of SDN Framework and Cloud Computing Platforms: An Open Source Approach**, Yüksek Lisans Tezi, Bologna University, 2017

Manthana M.P.V., **Network-as-a-Service Architecture with SDN and NFVA Proposed Evolutionary Approach for Service Provider Networks**, Yüksek Lisans Tezi, Delft University of Technology, 2015

Mishra V., **Interoperability in Software Defined Networking**, Lisans Tezi, Department of Electronics and Communication Engineering National Institute of Technology, Rourkela, 2015

Panda A., **A New Approach to Network Function Virtualization**, Doktora Tezi, Berkeley University, 2017

EKLER

Ek-1: YTA Mimarileri Değerlendirme Anketi

SDN Architectural Evaluation Survey

The goal of this survey is to collect measurable metrics for the following SDN Architectures based on the parameters defined below according to the Industry Practices.

Evaluation Architectures and Parameters

SDN Architectures to be evaluated:

+++++

1. SDN Confederation
2. Google Jupiter
3. Facebook Altoona
4. Amazon EC2
5. Cisco ACI
6. Nokia Nuage
7. Juniper Contrail
8. Big Switch Networks Big Cloud Fabric

Parameters used in the survey:

+++++

1. Protocol Coverage
2. Availability to Use
3. Service Creation
4. Application Creation
5. NFV Support
6. The Service and The Application Integration
7. Easy Configuration and Provision
8. Modularity
9. Divide and Conquer
10. Encapsulation Method
11. Clustering Different Hardware Architectures
12. Vendor Independence
13. Traffic Flow Flexibility
14. Cloud Integration
15. NON-SDN Connectivity

1. Before we start, it is important for us to know your background. What is your Job Title?

Mark only one oval.

- Network Design Engineer
- Network Software Engineer
- Network QA Engineer
- Network Operation Engineer
- Other: _____

1. Protocol Coverage is formal standards and policies comprised of rules, procedures and formats that define communication between two or more devices over a SDN network. Which one of the following applies on evaluating each SDN architecture based on the Protocol Coverage for the Controller according to the Industry Practices?

2. 1. SDN Confederation

Mark only one oval.

- Full
- Partial
- Private
- Other: _____

3. 2. Google Jupiter

Mark only one oval.

- Full
- Partial
- Private
- Other: _____

4. 3. Facebook Altoona

Mark only one oval.

- Full
- Partial
- Private
- Other: _____

5. 4. Amazon EC2

Mark only one oval.

- Full
- Partial
- Private
- Other: _____

6. 5. Cisco ACI

Mark only one oval.

- Full
- Partial
- Private
- Other: _____

7. 6. Nokia Nuage

Mark only one oval.

- Full
- Partial
- Private
- Other: _____

8. 7. Juniper Contrail

Mark only one oval.

- Full
- Partial
- Private
- Other: _____

9. 8. Big Switch Networks Big Cloud Fabric

Mark only one oval.

- Full
- Partial
- Private
- Other: _____

2. Availability to Use

This is to measure to see if the architecture is available to use by costumers. Which one of the following applies on evaluating each SDN architecture based on the Availability to Use according to the Industry Practices?

10. 1. SDN Confederation

Mark only one oval.

- Free to Use (Open Source)
- Comes with Price
- Private
- Other: _____

11. 2. Google Jupiter

Mark only one oval.

- Free to Use (Open Source)
- Comes with Price
- Private
- Other: _____

12. 3. Facebook Altoona

Mark only one oval.

- Free to use (Open Source)
- Comes with Price
- Private
- Other: _____

13. 4. Amazon EC2

Mark only one oval.

- Free to use (Open Source)
- Comes with Price
- Private
- Other: _____

14. 5. Cisco ACI

Mark only one oval.

- Free to use (Open Source)
- Comes with Price
- Private
- Other: _____

15. 6. Nokia Nuage

Mark only one oval.

- Free to use (Open Source)
- Comes with Price
- Private
- Other: _____

16. 7. Juniper Contrail

Mark only one oval.

- Free to use (Open Source)
- Comes with Price
- Private
- Other: _____

17. 8. Big Switch Networks Big Cloud Fabric

Mark only one oval.

- Free to use (Open Source)
- Comes with Price
- Private
- Other: _____

3.
Service
Creation

Service Creation is the capability that facilitates a network operation. It typically is provided by a server based on network protocols running at the application layer in the Open Systems Interconnection (OSI) model of the network. Which one of the following applies on evaluating each architecture based on the Service Creation according to the Industry Practices?

18. 1. SDN Confederation

Mark only one oval.

- The architecture allows service creation in full dynamic way
- The architecture allows service creation in partial dynamic way
- The architecture allows service creation in static way
- The architecture doesn't allow service creation
- Other: _____

19. 2. Google Jupiter

Mark only one oval.

- The architecture allows service creation in full dynamic way
- The architecture allows service creation in partial dynamic way
- The architecture allows service creation in static way
- The architecture doesn't allow service creation
- Other: _____

20. 3. Facebook Altoona

Mark only one oval.

- The architecture allows service creation in full dynamic way
- The architecture allows service creation in partial dynamic way
- The architecture allows service creation in static way
- The architecture doesn't allow service creation
- Other: _____

21. 4. Amazon EC2

Mark only one oval.

- The architecture allows service creation in full dynamic way
- The architecture allows service creation in partial dynamic way
- The architecture allows service creation in static way
- The architecture doesn't allow service creation
- Other: _____

22. 5. Cisco ACI

Mark only one oval.

- The architecture allows service creation in full dynamic way
- The architecture allows service creation in partial dynamic way
- The architecture allows service creation in static way
- The architecture doesn't allow service creation
- Other: _____

23. 6. Nokia Nuage

Mark only one oval.

- The architecture allows service creation in full dynamic way
- The architecture allows service creation in partial dynamic way
- The architecture allows service creation in static way
- The architecture doesn't allow service creation
- Other: _____

24. 7. Juniper Contrail

Mark only one oval.

- The architecture allows service creation in full dynamic way
- The architecture allows service creation in partial dynamic way
- The architecture allows service creation in static way
- The architecture doesn't allow service creation
- Other: _____

25. 8. Big Switch Networks Big Cloud Fabric

Mark only one oval.

- The architecture allows service creation in full dynamic way
- The architecture allows service creation in partial dynamic way
- The architecture allows service creation in static way
- The architecture doesn't allow service creation
- Other: _____

4.
Application
Creation

Application Creation is the capability to utilize the Internet or other network hardware infrastructure to perform transferring data from one point to another within the network. Which one of the following applies on evaluating each architecture based on the Application Creation according to the Industry Practices?

26. 1. SDN Confederation

Mark only one oval.

- The architecture allows application creation in full dynamic way
- The architecture allows application creation in partial dynamic way
- The architecture allows application creation in static way
- The architecture doesn't allow application creation
- Other: _____

27. 2. Google Jupiter

Mark only one oval.

- The architecture allows application creation in full dynamic way
- The architecture allows application creation in partial dynamic way
- The architecture allows application creation in static way
- The architecture doesn't allow application creation
- Other: _____

28. 3. Facebook Altoona

Mark only one oval.

- The architecture allows application creation in full dynamic way
- The architecture allows application creation in partial dynamic way
- The architecture allows application creation in static way
- The architecture doesn't allow application creation
- Other: _____

29. 4. Amazon EC2

Mark only one oval.

- The architecture allows application creation in full dynamic way
- The architecture allows application creation in partial dynamic way
- The architecture allows application creation in static way
- The architecture doesn't allow application creation
- Other: _____

30. 5. Cisco ACI

Mark only one oval.

- The architecture allows application creation in full dynamic way
- The architecture allows application creation in partial dynamic way
- The architecture allows application creation in static way
- The architecture doesn't allow application creation
- Other: _____

31. 6. Nokia Nuage

Mark only one oval.

- The architecture allows application creation in full dynamic way
- The architecture allows application creation in partial dynamic way
- The architecture allows application creation in static way
- The architecture doesn't allow application creation
- Other: _____

32. 7. Juniper Contrail

Mark only one oval.

- The architecture allows application creation in full dynamic way
- The architecture allows application creation in partial dynamic way
- The architecture allows application creation in static way
- The architecture doesn't allow application creation
- Other: _____

33. 8. Big Switch Networks Big Cloud Fabric

Mark only one oval.

- The architecture allows application creation in full dynamic way
- The architecture allows application creation in partial dynamic way
- The architecture allows application creation in static way
- The architecture doesn't allow application creation
- Other: _____

5. NFV Support

Network functions virtualization support provides a new way to create, distribute, and operate networking services. It is the process of decoupling the network functions from proprietary hardware appliances so they can run in software on standardized hardware. How would you rank each architecture based on the NFV support according to the Industry Practices? (1: Excellent - 2: Above Average - 3: Average - 4: Below Average - 5: Very Poor)

34. 1. SDN Confederation

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

35. 2. Google Jupiter

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

36. 3. Facebook Altoona

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

37. 4. Amazon EC2

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

38. 5. Cisco ACI

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

39. 6. Nokia Nuage

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

40. 7. Juniper Contrail

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

41. 8. Big Switch Networks Big Cloud Fabric

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

6. Service and Application Integration

Service and Application Integration is the ability to integrate all the clusters and the virtual machines inside or outside of cloud with the applications running according to the traffic classification needs. It is the function which allows mapping created applications to the underlying services. Can you evaluate if each architecture has Service and Application Integration support according to the Industry Practices?

42. 1. SDN Confederation

Mark only one oval.

- Full
- Partial
- None
- Other: _____

43. 2. Google Jupiter

Mark only one oval.

- Full
- Partial
- None
- Other: _____

44. 3. Facebook Altoona

Mark only one oval.

- Full
- Partial
- None
- Other: _____

45. 4. Amazon EC2

Mark only one oval.

- Full
- Partial
- None
- Other: _____

46. 5. Cisco ACI

Mark only one oval.

- Full
- Partial
- None
- Other: _____

47. 6. Nokia Nuage

Mark only one oval.

- Full
- Partial
- None
- Other: _____

48. 7. Juniper Contrail

Mark only one oval.

- Full
- Partial
- None
- Other: _____

49. 8. Big Switch Networks Big Cloud Fabric

Mark only one oval.

- Full
- Partial
- None
- Other: _____

7.
Configuration
and Provision

Configuration and Provision is the process of setting a network's controls, flow and operation to support the network communication of the SDN cloud easily. How would you rank each architecture based on the easy Configuration and Provision according to the Industry Practices? (1: Excellent - 2: Above Average - 3: Average - 4: Below Average - 5: Very Poor)

50. 1. SDN Confederation

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

51. 2. Google Jupiter

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

52. 3. Facebook Altoona

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

53. 4. Amazon EC2

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

54. 5. Cisco ACI

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

55. 6. Nokia Nuage

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

56. 7. Juniper Contrail

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

57. 8. Big Switch Networks Big Cloud Fabric

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

8. Modularity

Modularity is the degree of system's components are made up of relatively independent components or parts which can be combined. How would you rank each architecture based on the Modularity according to the Industry Practices? (1: Excellent - 2: Above Average - 3: Average - 4: Below Average - 5: Very Poor)

58. 1. SDN Confederation

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

59. 2. Google Jupiter

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

60. 3. Facebook Altoona

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

61. 4. Amazon EC2

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

62. 5. Cisco ACI

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

63. 6. Nokia Nuage

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

64. 7. Juniper Contrail

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

65. 8. Big Switch Networks Big Cloud Fabric

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

9. Divide
and
Conquer

Divide and Conquer is the method that designs paradigm based on multi-branched recursion of a network architecture. Can you evaluate if each architecture has the Divide and Conquer functionality according to the Industry Practices?

66. 1. SDN Confederation

Mark only one oval.

- Full
- Partial
- None
- Other: _____

67. 2. Google Jupiter

Mark only one oval.

- Full
- Partial
- None
- Other: _____

68. 3. Facebook Altoona

Mark only one oval.

- Full
- Partial
- None
- Other: _____

69. 4. Amazon EC2

Mark only one oval.

- Full
- Partial
- None
- Other: _____

70. 5. Cisco ACI

Mark only one oval.

- Full
- Partial
- None
- Other: _____

71. 6. Nokia Nuage

Mark only one oval.

- Full
- Partial
- None
- Other: _____

72. 7. Juniper Contrail

Mark only one oval.

- Full
- Partial
- None
- Other: _____

73. 8. Big Switch Networks Big Cloud Fabric

Mark only one oval.

- Full
- Partial
- None
- Other: _____

10.
Encapsulation
Method

Encapsulation Method is the process of taking data from one protocol and translating it into another protocol, so the data can continue across a network. Can you evaluate if each architecture has the support for the Encapsulation Method according to the Industry Practices?

74. 1. SDN Confederation

Mark only one oval.

- Full
- Partial
- None
- Other: _____

75. 2. Google Jupiter

Mark only one oval.

- Full
- Partial
- None
- Other: _____

76. 3. Facebook Altoona

Mark only one oval.

- Full
- Partial
- None
- Other: _____

77. 4. Amazon EC2

Mark only one oval.

- Full
- Partial
- None
- Other: _____

78. 5. Cisco ACI

Mark only one oval.

- Full
- Partial
- None
- Other: _____

79. 6. Nokia Nuage

Mark only one oval.

- Full
- Partial
- None
- Other: _____

80. 7. Juniper Contrail

Mark only one oval.

- Full
- Partial
- None
- Other: _____

81. 8. Big Switch Networks Big Cloud Fabric

Mark only one oval.

- Full
- Partial
- None
- Other: _____

11. Clustering
Different
Hardware
Architectures

Clustering Different Hardware Architectures is the method that allows the interconnection of hardware groups with different capabilities in a way that makes them appear to the operating environment as a single system. Can you evaluate if each architecture has the support for the Clustering Different Hardware Architectures Method according to the Industry Practices?

82. 1. SDN Confederation

Mark only one oval.

- Full
- Partial
- None
- Other: _____

83. 2. Google Jupiter

Mark only one oval.

- Full
- Partial
- None
- Other: _____

84. 3. Facebook Altoona

Mark only one oval.

- Full
- Partial
- None
- Other: _____

85. 4. Amazon EC2

Mark only one oval.

- Full
- Partial
- None
- Other: _____

86. 5. Cisco ACI

Mark only one oval.

- Full
- Partial
- None
- Other: _____

87. 6. Nokia Nuage

Mark only one oval.

- Full
- Partial
- None
- Other: _____

88. 7. Juniper Contrail

Mark only one oval.

- Full
- Partial
- None
- Other: _____

89. 8. Big Switch Networks Big Cloud Fabric

Mark only one oval.

- Full
- Partial
- None
- Other: _____

12. Vendor Independence

Vendor Independence is the set of feature that provide the freedom of being able to choose different hardware or software platform without interfering any vendor lock in. Which one of the following applies on evaluating each SDN architecture based on the Vendor Independence support according to the Industry Practices?

90. 1. SDN Confederation

Mark only one oval.

- Full
- Partial
- None
- Other: _____

91. 2. Google Jupiter

Mark only one oval.

- Full
- Partial
- None
- Other: _____

92. 3. Facebook Altoona

Mark only one oval.

- Full
- Partial
- None
- Other: _____

93. 4. Amazon EC2

Mark only one oval.

- Full
- Partial
- None
- Other: _____

94. 5. Cisco ACI

Mark only one oval.

- Full
- Partial
- None
- Other: _____

95. 6. Nokia Nuage

Mark only one oval.

- Full
- Partial
- None
- Other: _____

96. 7. Juniper Contrail

Mark only one oval.

- Full
- Partial
- None
- Other: _____

97. 8. Big Switch Networks Big Cloud Fabric

Mark only one oval.

- Full
- Partial
- None
- Other: _____

13.
Traffic
Flow
Flexibility

Traffic Flow Flexibility is the ability to define traffic flow rules and assign them to predefined service, applications and clusters. How would you rank each architecture based on the Traffic Flow Flexibility according to the Industry Practices? (1: Excellent - 2: Above Average - 3: Average - 4: Below Average - 5: Very Poor)

98. 1. SDN Confederation

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

99. 2. Google Jupiter

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

100. 3. Facebook Altoona

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

101. 4. Amazon EC2

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

102. 5. Cisco ACI

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

103. 6. Nokia Nuage

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

104. 7. Juniper Contrail

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

105. 8. Big Switch Networks Big Cloud Fabric

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

14. Cloud Integration

Cloud Integration is the system that connects various applications, systems, repositories, and environments for the real-time exchange of data and processes. Can you evaluate if each architecture has the support for the Cloud Integration according to the Industry Practices?

106. 1. SDN Confederation

Mark only one oval.

- Full
- Partial
- None
- Other: _____

107. 2. Google Jupiter

Mark only one oval.

- Full
- Partial
- None
- Other: _____

108. 3. Facebook Altoona

Mark only one oval.

- Full
- Partial
- None
- Other: _____

109. 4. Amazon EC2

Mark only one oval.

- Full
- Partial
- None
- Other: _____

110. 5. Cisco ACI

Mark only one oval.

- Full
- Partial
- None
- Other: _____

111. 6. Nokia Nuage

Mark only one oval.

- Full
- Partial
- None
- Other: _____

112. 7. Juniper Contrail

Mark only one oval.

- Full
- Partial
- None
- Other: _____

113. 8. Big Switch Networks Big Cloud Fabric

Mark only one oval.

- Full
- Partial
- None
- Other: _____

15. NON-SDN
Connectivity

NON-SDN Connectivity is the ability to get connectivity to a NON-SDN Cluster. Can you evaluate if each architecture has the support for the NON-SDN Connectivity according to the Industry Practices?

114. 1. SDN Confederation

Mark only one oval.

- Full
- Partial
- None
- Other: _____

115. 2. Google Jupiter

Mark only one oval.

- Full
- Partial
- None
- Other: _____

116. 3. Facebook Altoona

Mark only one oval.

- Full
- Partial
- None
- Other: _____

117. 4. Amazon EC2

Mark only one oval.

- Full
- Partial
- None
- Other: _____

118. 5. Cisco ACI

Mark only one oval.

- Full
- Partial
- None
- Other: _____

119. 6. Nokia Nuage

Mark only one oval.

- Full
- Partial
- None
- Other: _____

120. 7. Juniper Contrail

Mark only one oval.

- Full
- Partial
- None
- Other: _____

121. 8. Big Switch Networks Big Cloud Fabric

Mark only one oval.

- Full
- Partial
- None
- Other: _____

ÖZGEÇMİŞ

2003 yılı Beykent Üniversitesi Lisans Matematik ve Bilgisayar (Anadal) ve Yönetim Bilişim Sistemleri (Yandal) Bölümü mezunuyum. Yüksek Lisansımı Teknoloji Bölümü, Kent State Üniversitesinde 2005 yılında tamamladım. İş tecrübem sırasıyla yazılım mühendisi (Direct Response Technologies, ABD, 2006-2005), ağ mühendisi (ECI Telecom, ABD, 2010-2006), ağ test mühendisi (Google, ABD, 2012-2010), ağ operasyon mühendisi (Google, ABD, 2015-2012), teknik proje yöneticisi (Google, İrlanda, 2017-2015) ve bulut mimari (Rakuten, Japonya, 2020) şeklindedir. Google'da çalıştığım süre boyunca Google B4 (2012), Google Jupiter (2014), Google Technical Infrastructure (TI) (2006) ödüllerini aldım. PHP Zend Certified Engineer, Cisco Certified Network Associate (CCNA, 2009), Cisco Certified Internet Professional (CCIP, 2010) ve Cisco Certified Internetwork Expert (CCIE, Service Provider ,Written, 2010) sertifikalarına sahibim.

Aday: Hale DÖNERTAŞLI