T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

EŞ DÜĞÜMLER ARASI AĞLARDA ÇOKLU ORTAM VERİLERİNİN GERÇEK ZAMANLI İLETİMİ İÇİN YENİ BİR YÖNTEM

DOKTORA TEZİ

Halil ARSLAN

Enstitü Anabilim Dalı : ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ

Tez Danışmanı : Yrd. Doç. Dr. Sinan TÜNCEL

Şubat 2016

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

EŞ DÜĞÜMLER ARASI AĞLARDA ÇOKLU ORTAM VERİLERİNİN GERÇEK ZAMANLI İLETİMİ İÇİN YENİ BİR YÖNTEM

DOKTORA TEZİ

Halil ARSLAN

Enstitü Anabilim Dalı : ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ

Bu tez 02 / 02 / 2016 tarihinde aşağıdaki jüri tarafından oybirliği / ~~oyçokluğu~~ ile kabul edilmiştir.

Prof. Dr.
Etem KÖKLÜKAYA
Jüri Başkanı

Prof. Dr.
İsmail ERTÜRK
Üye

Doç. Dr.
Murat ÇAKIROĞLU
Üye

Doç. Dr.
Resul KARA
Üye

Yrd. Doç. Dr.
Sinan TÜNCEL
Üye

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Halil ARSLAN
02.02.2016

TEŞEKKÜR

Öncelikle, tez çalışmam boyunca beni yönlendiren, ilgi ve yardımlarını esirgemeyen ve her türlü destek ve teşvikini gördüğüm danışman hocam Yrd. Doç. Dr. Sinan TÜNCEL'e teşekkür ederim. Yine araştırmam ve akademik hayatım boyunca yardım ve desteklerini gördüğüm sayın hocam Prof. Dr. Hüseyin EKİZ ve bana emeği geçen tüm hocalarıma en kalbi saygılarımı sunarım. Deneysel çalışmalarım sürecinde verdiği destek ve katkı için Detay Danışmanlık A.Ş. yönetici ve çalışanlarına, çalışmalarım boyunca bilgi birikimlerinden ve rehberliklerinden faydalandığım değerli dostlarıma teşekkürlerimi sunarım.

Bu çalışmanın maddi açıdan desteklenmesine olanak sağlayan Sakarya Üniversitesi Bilimsel Araştırma Projeleri (BAP) Komisyon Başkanlığı'na (Proje No: 2012-50-02-051) ve doktora eğitimim boyunca "TÜBİTAK-BİDEB 2211-Yurt İçi Doktora Burs Programı" kapsamında burs imkanı sağlayan Türkiye Bilimsel ve Teknolojik Araştırma Kurumu'na teşekkür ederim.

Son olarak tez çalışmamı, desteklerini esirgemeyen değerli ailem ve hürmetle yad ettiğim babam merhum Memet Ali ARSLAN'a ithaf ediyorum.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR LİSTESİ

ALG : Uygulama katman geçişi
AOL : Amerika'da internet servis sağlayıcı
BOSH : HTTP üzerinden senkron çift yönlü akış
CAN : NAT geçişi algoritması
CIDR : Sınıfsız alanlararası yönlendirme
DMZ : Askerden arındırılmış bölge
EPD : Son nokta ayırıcısı
FC : Tam koni NAT
Firewall : Güvenlik duvarı
GSM : Mobil iletişim ve kullanılan ses kodlama standardı
H323 : IP tabanlı ses iletim protokolü
ICQ : Seni arıyorum anlamına gelen bir mesajlaşma programı
IETF : İnternet standartları geliştiren grup
ICE : NAT arkasındaki kullanıcılar için hibrit geçiş protokolü
ICMP : İnternet kontrol mesaj protokolü
IM : Anında mesajlaşma
ISP : İnternet servis sağlayıcısı
ITU : Uluslararası telekominikasyon standartları belirleyen kuruluş
IQ : Bilgi sorgusu
JID : Jabber kimliği
LAN : Yerel alan ağı
MPEG : Görüntü kodlama standardı
NAPT : Ağ adres port dönüştürücüsü
NAT : Ağ adres dönüştürücüsü
P2P : Eş düğümler arası
PRC : Port kısıtlamalı NAT

RC : Adres kısıtlamalı NAT

RFC : İnternet için standart oluşturma dokümanları

RTCP : Gerçek zamanlı aktarım kontrol protokülü

RTMFP : Gerçek zamanlı medya akış protokolü

RTMP : Gerçek zamanlı mesajlaşma protokolü

RTP : Gerçek zamanlı aktarım protokolü

RTSP : Gerçek zamanlı yayın protokolü

RTT : Gidiş dönüş süresi

SASL : Basit kimlik doğrulama ve günvelik katmanı

SBN : Durum tabanlı NAT geçişi

SCTP : Yayın kontrol aktarım protokolü

SDP : Oturum tanımlama protokolü

SILK : IETF tarafından geliştirilen genişband ses kodlaması

SIP : Oturum başlatma kuralları, IP tabanlı ses iletim protokolü

STUN : NAT üzerinden yerel-genel port eşleştirme protokolü

SYM : Tekrar edilemeyen port dönüşümlü NAT

TCP : Aktarım kontrol protokolü

TURN : NAT arkasındaki kullanıcılar için aktarım protokolü

UDP : Kullanıcı veribloğu protokolü

UPnP : Evrensel tak çalıştır

URL : Kaynağın tam yolu

VoIP : IP üzerinden ses iletimi

VPN : Sanal özel ağ

XMPP : Genişletilebilir mesajlaşma ve durum protokolü

WAN : Geniş alan ağı

WebRTC : Web tabanlı gerçek zamanlı iletişim

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

ÖZET

Anahtar kelimeler: Eş Düğümler Arası Ağlar, Gerçek Zamanlı Veri Aktarımı, Ağ Adres Dönüştürücü Geçişi, Genişletilebilir Mesajlaşma ve Durum Protokolü

İnternet kullanıcılarının gerçek zamanlı ortam verilerini paylaşma ihtiyacı her geçen gün artmaktadır. Artan bu ihtiyacın karşılanmasında, klasik istemci-sunucu mimarisi pek çok parametreden dolayı istenilen verimi sağlayamamaktadır. Bu nedenle kullanıcılar bulundukları ağ yapılarından bağımsız olarak birbirleri ile doğrudan iletişim kurabilmelidirler. Bu nedenle eş düğümler arası iletişim için sorun oluşturan durumların belirlenmesi ve araştırmacılar tarafından uygun yöntemlerin ortaya konulması gerekmektedir. Eş düğümler arası iletişimde karşımıza çıkan temel sorunların başında ağ adres dönüştürücü ve güvenlik duvarı gibi özel ağ oluşturan cihazların arkasındaki istemcilere kamusal ağdan erişilememesi gelmektedir. Bu sorunun çözümüne yönelik literatürde öne sürülen çözüm önerilerinin değişik avantaj ve dezavantajları bulunmaktadır. Bu çözüm önerilerinden İnteraktif Bağlantı Kurulumu ve Gerçek Zamanlı Medya Akış Protokolü, gerek internet altyapısından bağımsız oluşları, gerekse de dinamik yapılar için uygunlukları ile öne çıkmaktadırlar. Yapılan çalışma ile ağ adres dönüştürücü geçişi için tüm adımların tanımlandığı bir yöntem geliştirilerek eş düğümler arası ağlarda çoklu ortam verilerinin iletiminde uçtan uca tam bir model ortaya konulmuş ve "Durum Tabanlı Ağ Adres Dönüştürücü Geçişi" olarak isimlendirilmiştir. Geliştirilen model ile İnteraktif Bağlantı Kurulumu protokolünün bağlantı kurulum süresi, band genişliği ve paket kullanımı parametreleri iyileştirilmiştir.

A NEW METHOD FOR REAL TIME TRANSPORT OF MULTIMEDIA DATA IN PEER-TO-PEER NETWORKS

SUMMARY

Keywords: Peer to Peer, RTP, NAT Traversal, XMPP

The usage of peer-to-peer (P2P) networks that provide sharing of real-time environmental data by internet users is becoming more and more popular. As a result, it is necessary to identify the problems during P2P communication and to develop proper solutions. One of the major problems of P2P communication is that it is not possible to reach the clients behind devices that create private networks like network address translation (NAT) and firewalls from the public network. Among the solutions proposed for this problem, Interactivity Connectivity Establishment (ICE) and Real Time Media Flow Protocol (RTMFP) are the methods most preferred in the literature. These methods seem more attractive than other NAT traversal mechanisms since they are independent from internet infrastructure and are also appropriate for dynamic structures. However, they do have some disadvantages. With this thesis work, a new state-based end-to-end communication technique (SBN) for NAT traversal has been designed and realized. The performance of the designed method was evaluated against three criteria connectivity check delay, connection packet count and bandwidth and compared to the ICE method.

BÖLÜM 1. GİRİŞ

1.1. Giriş

Son yıllarda, mantıksal (overlay) ağ teknolojileri dikkat çeken araştırma alanları arasında yer almaktadır. Bu teknolojilerin temel amacı, yüksek boyutlu verilerin işlenmesi, dağıtımı, ölçeklendirilmesi gibi maliyetlerin iyileştirilmesine yönelik çözümler sunmaktır. Mantıksal ağ mimarilerilerinden özellikle eş düğümler arası (Peer to Peer - P2P) bilgisayar ağları, çoklu ortam verileri gibi büyük boyutlara sahip içeriğin işlenmesi ve dağıtımına yönelik önemli kazanımlar sağlamaktadır. P2P bilgisayar ağları, iki ya da daha fazla istemci arasında görevlerin ve işlerin paylaşılması temeline dayanan dağıtık uygulama mimarileri olarak tanımlanmaktadır [1]. Mimari açıdan geleneksel istemci-sunucu yaklaşımında sunucu, tedarikçi konumunda merkezde yer alırken, istemciler tüketici konumunda dağıtık olarak yapılandırılmaktadır. P2P bilgisayar ağlarında ise istemcilerin tümü hem tedarikçi hem de tüketici olarak merkezi olmayan mimariye sahiptirler. Bu yaklaşım kaynak paylaşımı ve içerik dağıtımında geleneksel istemci-sunucu yaklaşımına karşı yaygın ve başarılı bir seçenek olarak dikkat çekmektedir [1, 12].

İstemci-sunucu yaklaşımında merkezde yer alan sunucu, iletilecek bir veriyi istemci adeti kadar iletmek zorunda olduğu için sistem kaynaklarından maliyetli ve önemli olan bant genişliğini bu bağlamda tüketmektedir. Özellikle gerçek zamanlı veri transferi ihtiyacı duyulan uygulamalarda ise kurumsal internet altyapısının çift taraflı (upload, download) tüketilmesi anlamına gelmektedir. Bu durum, kurumsal internet altyapıları ile yüksek boyutlara sahip gerçek zamanlı verilerin iletiminin geleneksel mimariler ile gerçekleştirilmesini, gerek bant genişliği maliyeti, gerek sunucu yatırım maliyetleri, gerekse de servis kalitesi açısından çözümlenmesi gereken problemler olarak ortaya çıkarmaktadır.

Karşılıklı veri transferi gerektiren P2P bilgisayar ağları, basit dosya paylaşımı uygulamalarının yanında, çok kanallı işbirliği uygulamaları, içerik yönetimi ve anında mesajlaşma ürünleri ile geniş bir alanda karşılık bulmuştur. Bu tür mantıksal ağlar için anlık mesajlaşma servisleri, web tarayıcıların internet üzerindeki önemi gibidir [2]. Anında mesajlaşma (Instant Messaging – IM), çeşitli kullanıcılar arasında farklı cihaz türleri ile rahat iletişim sağlayabilen IP tabanlı bir uygulamadır. Günümüzde, bilgisayar-bilgisayar arası anlık metinsel mesajlaşma yanında, ses ve video da eklenebilen formu en çok bilinen şeklidir.

Anlık mesajlaşma servisleri 1996 yılında ortaya çıkan "seni arıyorum" (I seek you – ICQ) uygulaması ile birlikte internet kullanıcıları arasında hızla yaygınlaşmıştır. Bu teknolojinin popülaritesi ile çok geçmeden birçok firma, benzer ürünler (AOL, Yahoo, Live Messenger) geliştirmişlerdir. Bu uygulamaların her biri, geliştiren şirketler tarafından işletilen, kendilerine özgü bir ağ protokolüne bağlı olduğu için kendi kullanıcıları dışındaki kullanıcılar ile iletişim kuramamaktadır. Bu sorun merkezi olmayan anında mesajlaşma ağı ve protokolünün geliştirilmesi fikrini ortaya çıkarmıştır. Önceleri jabber günümüzde ise Genişletilebilir Mesajlaşma ve Durum Protokolü (Extensible Messaging and Presence Protocol – XMPP) olarak ifade edilen standart, 1999 yılından günümüze kadar gelişimini sürdürmeye devam etmiştir. XMPP protokolü ile özel sohbet ağlarındaki kullanıcılar, birbirleri ile servislerinin izin verdikleri ölçüde iletişim kurabilme olanağına sahiptirler. Bu durum, kişisel kullanıcılar için anından mesajlaşma pazarındaki XMPP destekli herhangi bir ürün ile kendi listesindeki kullanıcılarla iletişim kurabilmesini sağlamıştır. XMPP, Internet Engineering Task Force (IETF) tarafından yayımlanan RFC 3920 ve RFC 3921 ile tanımlanarak gelişimini sürdürmeye devam etmiştir. XMPP, ister iki sunucu, ister iki istemci arasında P2P haberleşme olanağı sağlayan ve birbirlerine bağlanabilen servisler arasında federe bir ağ oluşturmaktadır [3, 43].

P2P ağ uygulamaları, istemciler arasında uçtan uca iletişim kurabilmelerini gerektirir. Günümüzde ise pekçok kullanıcı internete, güvenlik duvarı (firewall) ve ağ adres dönüştürücüleri (Network Address Translators - NATs) arkasından bağlanmaktadırlar. Yazılım ve donanım olarak bulunabilen her iki sistemden

güvenlik duvarlarının temel görevi altağlardaki veri aktarımını kontrol etmek ve yetkisiz erişimleri engellemek iken, ağ adres dönüştürücülerinin temel görevi ise kamusal IP adresleri üzerinden akan ağ trafiğini özel IP adreslerine dönüştürmektir. Ağ adres dönüştürücüleri ve güvenlik duvarlarından kaynaklanan asimetrik dönüşüm ve port yetkilendirmeleri P2P ağ uygulamalarının uçtan uca iletişim kurabilme yeteğine önemli kısıtlar getirmektedir [4]. Bu durum P2P ağ uygulamalarının kayıt, keşif ve aktarım gibi çeşitli görevler için konumlandırılmış randevu sunucuları kullanımını ortaya çıkarmıştır. Günümüzde internet erişimindeki kolaylık ve mobilite, özellikle video gibi multimedya içeriğin istemciler arasında paylaşımına yönelik istekleri artırmaktadır. Bu bağlamda P2P iletişim temelinde çoklu ortam verilerinin istemciler arasında paylaşım uygulamalarının ihtiyaç duyduğu altyapı iyileştirmeleri ve mimari modeller önemli araştırma alanları oluşturmaya devam edecektir.

1.2. Tezin Amacı

Bu tez çalışması ile P2P bilgisayar ağlarında, ortam verilerinin gerçek zamanlı iletim modelinin oluşturulması amaçlanmıştır. Geliştirilen modelde, veri transferinin merkezi bir sunucu üzerinden yapılması yerine, iletişimine geçecek istemcileri doğrudan birbirleri ile bağlayacak altyapı tasarlanmış ve gerçeklenerek gerçek zamanlı aktarım protokolleri koşturulmuştur. Ortaya konan çalışma ile görsel (video) ve işitsel (audio) gerçek zamanlı ortam verilerinin yüksek kalitede, kurumsal internet altyapısını çok az kullanarak yüksek hızlarda istemciler arasında taşınabilmesine olanak sağlayacak altyapının kurulma süreci iyileştirilmektedir.

Bu tez çalışması ile ayrıca P2P bilgisayar ağlarında karşılaşılan NAT ve firewall gibi özel ağ oluşturan cihazlardan kaynaklanan iletişim problemlerinin çözümü literatür ile karşılaştırılmalı olarak ortaya konulmaktadır. Bu sorunun giderilmesi için RFC 3489 ile tanımlanan STUN (Simple Traversal of User Datagram Protocol Through Network Address Translators NATs) protokolü, RFC 5766 ile tanımlanan TURN (Traversal Using Relays around NAT) protokolü, RFC 5245 ile tanımlanan ICE (Interactive Connectivity Establishment) protokolü, RFC 7016 ile tanımlanan Gerçek

Zamanlı Medya Akış Protokolü (Real-Time Media Flow Protocol - RTMFP) ve Gerçek Zamanlı Mesajlaşma Protokolü (Real Time Messaging Protocol - RTMP) ile literatürde kullanılan benzer çalışmalar karşılaştırmalı olarak ele alınmaktadır [5, 6, 7, 8, 9]. Bağlantı garantisi, band genişliği kullanımı, aktarılan paket sayısı ve bağlantı gecikmesi gibi parametreler incelenerek geliştirilen yeni yöntemin verimliliği ortaya konulmaktadır. Geliştirilen yöntem ile NAT tiplerinden Symmetric NAT olarak adlandırılan modellerde de gerekli altyapıyı oluşturabilecek yaklaşım ortaya konularak P2P bilgisayar ağı iletişim problemleri en aza indirgenmektedir.

Gerçek zamanlı aktarım protokolü (Real-time Transport Protocol – RTP) ortam verilerinin uçtan uca taşınmasını sağlayan RFC 3550 ile tanımlanan UDP temelli iletimin gerçekleştirilmesini ortaya koymaktadır [10]. Geleneksel istemci-sunucu mimarilerinde sıklıkla kullanılan bu protokol ile video konferans, sesli iletişim, web tabanlı video yayınları gibi veri aktarım uygulamaları gerçekleştirilmektedir. Aktarım katmanında UDP kullanımı, TCP'ye göre ek başlıklar gerektirmediği için gerçek zamanlı uygulamalarda tercih edilmektedir [11, 12]. Ancak UDP'nin güvenlik duvarlarında engellendiği durumlarda RTP uygulamaları kullanılamamaktadır. Yapılan çalışma ile güvenlik duvarlarından kaynaklanan P2P ağ oluşturma sorunlarını giderecek nitelikte mümkün olan durumlarda aktarım katmanında UDP, aksi durumlarda TCP kullanımı adaptif olarak gerçeklenmiştir.

Gerçek zamanlı veri aktarım uygulamalarında UDP, TCP, RTP, RTMFP ve RTSP gibi protokoller kullanılmaktadır. Bu protokollerin yapıları gereği NAT ve güvenlik duvarı gibi özel ağ oluşturan yapılar arkasındaki düğümler için bağlantı ve verimlilik sorunları ortaya çıkmaktadır. NAT geçiş yöntemi olarak belirlenen tekniğe uygun aktarım ve uygulama katmanı protokolünün belirlenmesi bu noktada önem kazanmaktadır. Bu çalışma ile gerek NAT geçiş işlevinin, bağlantı garantisi, band genişliği kullanımı, düşük bağlantı kurulum zamanı, paket sayısı gibi parametrelerin verimliliği gerekse bu geçiş yöntemine uygun aktarım ve uygulama katmanı protokolünün kullanımı sağlanmaktadır.

Gerçek zamanlı ortam verilerinin P2P bilgisayar ağlarında aktarım uygulamalarının başka bir araştırma konusu ise ses ve video verilerinin sıkıştırma/kodlanma tekniklerdir. Bu noktada uygulama katmanı protokolleri video verileri için MPEG temelli Uluslararası Telekominikasyon Birliği (International Telecommunication Union - ITU-T) standartlarını [11, 13, 14, 15] tercih ederken, ses verileri için ITU-T standartlarından G serisi (G711, G721 vb.) kodlamalarla GSM, SILK, speex gibi pek çok standart [16, 17, 18, 44] kullanım alanı bulmaktadır. Erişilebilen çoklu ortam verisi kodlama teknikleri, ağ ve donanım parametreleri dikkate alınarak geri beslemeli adaptif yaklaşımlarla uygulamada kullanılmıştır. Kodlama teknikleri, servis kalitesi parametresini etkileyen önemli bir değişken olarak ortaya çıkmaktadır [18].

İşbirlikçi uygulamalarda P2P iletişim kurmak isteyen istemciler arasında kayıt, keşif ve durum yönetimi gibi temel koordinasyon işlevleri için randevu sunucularının kullanılması gerekmektedir. Çalışmanın genel çerçevesini oluşturan P2P bilgisayar ağlarında anlık mesajlaşma ve çoklu ortam verilerinin iletimi gerçeklenirken temel mesajlaşma protokolü olarak XMPP kullanılmıştır. P2P iletişim oluşturma adımlarının randevu sunucusu üzerinden yürütülebilmesi için randevu sunucusunun, geliştirilen NAT geçişi yöntemine uyumluluğu noktasında protokol eklentileri tanımlanmıştır.

Sonuç olarak bu tez çalışması ile P2P bilgisayar ağlarında ortam verilerinin gerçek zamanlı aktarımı için uçtan uca tam bir yöntem geliştirilmiştir. Bilgisayar ağlarının katmansal yapısı dikkate alınarak mimari model tasarlanmıştır. P2P iletişim sağlanmasında karşılaşılan NAT ve güvenlik duvarı kaynaklı engellerin aşılabilmesi için yeni bir verimli NAT geçiş yöntemi geliştirilmiş ve yöntemin verimliliği gerçek uygulama ortamında sayısal sonuçlarla gösterilmiştir. Geliştirilen NAT geçiş yöntemi durum tabanlı NAT geçişi (state-based end-to-end communication technique for NAT traversal – SBN) olarak isimlendirilmiştir. Yöntemin kullanıldığı uygulama mimarisi için aktarım katmanı gereksinimleri tanımlanmış ve buna uygun protokol seçimi sağlanmıştır. Uygulamanın koordinasyonu ve yönetimi için XMPP temelli bütüncül bir yapı ortaya konulmuştur.

1.3. Literatür Taraması

Literatürde, P2P bilgisayar ağlarında ortam verilerinin gerçek zamanlı iletimi konusunda birçok çalışma yapıldığı ve farklı yaklaşımlar sunulduğu görülmektedir. Özellikle son yıllarda, veri boyutlarındaki artış, bu verilerin geleneksel istemci-sunucu mimarisi üzerinden ortam verilerinin iletiminde güçlüklere neden olmaktadır. Yapılan çalışmalarda P2P veri transferi internet trafiğinde giderek arttığı görülmektedir. Labovitz ve arkadaşlarının çalışmalarına göre internet trafiğinin yaklaşık %20'sini P2P veri iletişimi oluşturmaktadır [19]. Harzog'un yaptığı bir çalışmaya göre 2014 yılında tüm internet trafiğinin sadece %25'inin web verilerinden oluşacağı öngörülmekte [20]. Cisco'ya ait çalışmalarda ise 2018 yılına kadar internet trafiğinin yaklaşık %50'sinin P2P iletişim uygulamlarından oluşacağı ifade edilmektedir. İnternet trafiğinin büyük bir kısmını IP üzerinden ses iletimi (VoIP), Internet TV, dosya paylaşımı (File Sharing) gibi P2P iletişim temelli uygulamaların tüketeceği öngörülmektedir [21, 22]. Veri boyutunun inanılmaz derecede artmasına rağmen istenilen bant genişliğinin sağlanamamasından dolayı, P2P bilgisayar ağları altyapı sorunlarının ve uygulama protokollerinin net olarak ortaya konulması ve problemlerin ele alınması gereksinimini gündeme getirmektedir.

P2P veri iletişimi açısından karşılaşılan en büyük problem IPv4 ile tanımlanan IP adreslerinin yetersizliğini gidermek ve özel ağ oluşturmak için kullanılan NAT ve Firewall gibi cihazların oluşturdukları ağlarda yer alan istemcilere, kamusal ağlardan erişilememesidir. Bu problemin giderilmesi için literatürde birçok çalışma ortaya konulmuştur. Rosenberg ve ark., yaptıkları çalışmada istemciler için özel ağ oluşturan NAT ve Firewall gibi cihazların arkasında bulunan düğümlerin NAT tiplerini ve bu düğümlere kamusal ağdan erişilebilir kılan kamusal IP ve portlarının belirlenmesine olanak sağlayan STUN protokolünü tanımlamışlardır [4]. Ancak bu tanımlamada ortaya konan yaklaşım Symmetric NAT tipindeki özel ağlarda gerekli iletişim olanağını sunamamaktadır.

Mahy ve ark., STUN protokolünde karşılaşılan Symmetric NAT tipindeki özel ağlardan kaynaklanan problemleri giderebilmek için Traversal Using Relays around

NAT (TURN) protokolünü tanımlamışlardır [5]. Bu protokol, temelde STUN ile çok büyük benzerlikler içerse de çok fazla bant genişliği gerektirmesi, TURN sunucusuna P2P iletişim sürecinde bağımlı kalınması, ve ek başlıklar gerektirmesi gibi gerekçelerden ötürü tek başına uygun bir çözüm oluşturamamıştır [23].

STUN ve TURN gibi protokollerde ortaya çıkan problemlerin giderilmesi için Rosenberg "Interactive Connectivity Establishment" (ICE) protokolünü RFC 5245'de tanımlamıştır [7]. Bu protokoldeki temel yaklaşım STUN ve TURN protokollerinin birlikte kullanılmasıyla tüm NAT tiplerinde P2P iletişim altyapısının sağlanmasına yöneliktir. Ancak protokolün temelini oluşturan yapılardaki olumsuzluklar, tüm NAT tipleri için iletişim ortamının sağlanması dışında devam etmektedir [24].

Topal ve ark., yaptıkları çalışmada yeni bir P2P UDP temelli dönüştürücü mimari önermişlerdir [25, 26]. Ancak bu çalışmalarında, getirdiği ilave yük ile mevcut IP paket yapısında yönlendirme özelliklerinin mevcut internet altyapısı açısından dağıtık sistemlere uygulanabilirliğine değinmemişlerdir.

Wang ve ark., ortaya koydukları araştırmalarında STUN protokolünün performansını artırmaya yönelik çalışmalarda bulunmuşlar ve bu çalışmalarını PS-STUN olarak isimlendirmişlerdir. Ancak istemcilerin rastlantısal Symmetric NAT tiplerinin arkasında olmaları durumlarda gerekli iyileşmeyi sağlayamadıklarını belirtmişlerdir [27]. Zhang ve ark., C-STUN isimli çalışmalarında belirttikleri algoritmada STUN sunucusunun yanında bir de süper düğüm önermişlerdir [28]. Müller ve ark., yaptıkları araştırma ile ICMP mesajlarını taklit eden otonom bir NAT dönüştürücü tasarlamışlardır [29]. Ancak ICMP paketlerinin engellendiği Firewall arkasındaki istemciler açısından tasarımın sonuçlarını ortaya koymamışlardır.

Tseng ve ark., ICE protokolünün temelini oluşturan STUN ve TURN protokollerinden kaynaklanan bağlantı kurulum gecikmesini gidermek için Context-Aware NAT (CAN) protokolünü önermişlerdir [30]. Bu çalışmada istemciler üzerinde çalışan ajan uygulamalar, düğümlerin ağ bilgilerini toplamakta ve oturum

başlatma protokolü (SIP) sunucusuna bildirmektedir [31]. İletişim kurmak isteyen düğümlerin arayüz bilgileri daha önceden bilindiği için en uygun yolun bulunması için oluşan gecikme dört temel adımda gerçekleştirilen kontrol ile ortadan kaldırılmıştır. Ancak her durumda ağ bilgilerinin SIP sunucusuna önceden bildirilmesi durumu modelin aslında tüm kullanıcılar için ilave yükler getirmesi durumunu ortaya çıkarmaktadır.

NAT geçiş tekniği olarak STUN, TURN ve bunların bir çatısı (framework) olan ICE protokollerinin yanısıra, UDP tabanlı veri akışı için Adobe firması tarafından tasarlanan RTMFP ve TCP tabanlı RTMP de önemli uygulama alanları bulmaktadır [32, 33, 34]. RTMFP protokolü, istemciler arasında mümkün olan doğrudan bağlantıların kurulumunu düşük gecikme süreleri ile sağlayabilirken, doğrudan bağlantı kurulumunun mümkün olmadığı durumlarda TCP tabanlı RTMP protokolünün kullanılmasını önermektedir.

NAT ve Firewall gibi özel ağ oluşturan cihazların arkasında yer alan istemcileri doğrudan haberleştirmeye yönelik altyapı çalışmaları literatürde yaygın çalışma alanlarından olmaya devam etmektedir. Ortaya konulan NAT geçiş yöntemleri ve bu yöntemlerin verimliliğinin iyileştirilmesi üzerine çalışmalar sürdürülmektedir. Bu noktada, ICE ve RTMFP yöntemleri avantaj ve dezavantajları ile birlikte ele alındığında, bağlantı kurulum süresinin düşük olduğu, istemciler üzerine ilave yükler getirmeyen ve UDP/TCP destekli yeni çalışmaların yapılması gerekliliği ortaya çıkmaktadır.

Bu tez çalışmasında istemciler arsasında iletişim problemlerinin giderilmesinin ardından uygulama aktarım katmanı için uygun protokolün belirlenmesi üzerine yapılan literatür çalışmaları sonucunda; gerçek zamanlı ortam verilerinin taşınması prensibi ışığında RTP protokolünün bu tür uygulamalar için kullanılmasına karar verilmiştir [12, 14, 15, 17, 18, 35]. RTP protokolü, IP ağları için gerçek zamanlı veri aktarım uygulamalarında sıkça kullanılan uygulama protokolüdür [18]. Literatürde RTP protokolüne yönelik farklı yaklaşımlar içeren çalışmalara rastlanmaktadır.

Güncel çalışmalardan elde edilen bulgular doğrultusunda bu çalışmanın uygulama katmanı tasarlanmıştır.

Costa ve ark., gerçek zamanlı mültimedya iletişim desteği için yeni bir P2P mimari önermişlerdir [36]. Oluşturdukları mimaride aktarım katmanı protokolü olarak SCTP (Stream Control Transmission Protocol) 'yi kullanmışlardır. Ancak bu çalışma ile NAT ve Firewall gibi ağ cihazları arkasındaki düğümlerin kamusal ağda iletişim kurmak istemeleri durumunda mevcut internet altyapısı dikkate alındığında iletişim kuramayacakları anlaşılmaktadır.

Chen ve ark., P2P bilgisayar ağları için canlı yayın yapan adaptif bir istemci yaklaşımı ortaya koymuşlardır [37]. Benzer şekilde Öztoprak hazırladığı tez çalışmasında çoğul ortam iletişim için melez içerik dağıtım ağları (CDN) üzerinde P2P ağ mimarisi sunmuştur [38].

Segui ve ark., multimedya grupları için RTP/RTCP tabanlı bir yaklaşım ortaya koymuşlardır [39]. Hazırladıkları çalışma ile RTCP tarafından sağlanan geribildirimleri kullanarak birçok kaynaktan gelen ortam verilerinin bir alıcı tarafından işlenmesi ile bir kaynaktan gelen ortam verisinin birçok alıcı tarafından işlenmesi süreçlerini senkronize etmişlerdir.

Burmeister ve ark., gerçek zamanlı aktarım kontrol protokolü geri bildirimlerinin kullanımı için genişletilmiş gerçek zamanlı aktarım protokolünün zamanlama kurullarını ve benzetim sonuçlarını yayınlamışlardır. Bu çalışmayı hem tek nokta bağlantıları (unicast), hem de çok nokta bağlantıları (multicast) için ortaya koymuşlardır. Bu çalışmada RTCP geribildirim paketlerinin ortalama 5 saniyede bir iletilmesi gerektiği vurgulanmıştır. Ayrıca her bir RTP oturumu için kullanılacak bant genişliğinin %5'inin RTCP paketleri için ayrılması gerektiği bildirilmektedir [40]. Pek çok çalışmada gerçek zamanlı aktarım kontrol protokolü başarım analizinde Burmeister ve ark., çalışması temel referans kaynağı olarak kullanılmıştır [39, 45, 46].

Khlifi ve ark., gerçek zamanlı aktarım protokolü üzerinde çoklu ortam akışları için adaptif tıkanıklık kontrol mekanizması geliştirmişler ve çalışmalarına ARTP ismini vermişlerdir. ARTP ile gerçek zamanlı aktarım protokolüne iki yeni parametre eklemişler ve geribildirim raporlarını yorumlamışlardır. Çalışmalarını NS2 benzetim ortamında değerlendirerek kayıp paket oranını ve bant genişliği kullanımını düşürdüklerini ifade etmişlerdir [41]. Ancak çalışmalarında P2P iletişim için herhangi bir altyapı sorunu olmadığı varsayımı ile hareket ettikleri görülmektedir. Veri transferinin gerçekleştiği düğümler için iletişim ortamının haritalandığı varsayımı ile analizlerini sunmuşlardır. Çalışmalarının gerçek ortamda ne derece başarılı sonuçlar verebildiği noktasında herhangi bir bulguya rastlanılmamıştır.

Granda ve ark., şirketlerde e-öğrenme platformlarının senkronizasyonu için bir ağ tekniği geliştirmişlerdir [42]. Çalışmalarında coğrafi olarak dağıtık yapıdaki şirketlerin hizmet içi eğitim işlevlerini yerine getirebilecekleri bir uygulama hazırlamışlardır. Uygulama ile şirketlerin çoklu yayım (multicast) altyapıları olmadığı durumlarda da grup iletişimini, RTP Aktarım (RTP Relay) sunucuları üzerinden verimli bir şekilde yapabileceklerini ifade etmişlerdir. Çalışmalarında geleneksek istemci-sunucu mimarisini kullanan Granda ve ark., bant genişliği ve işlemci kullanımı gibi parametrelerle önerdikleri ağ modelini değerlendirmişlerdir.

NAT ve Firewall gibi özel ağ oluşturan cihazlar üzerinden kamusal ağa erişmek için kullanılan STUN, istemcilerin özel ağ çıkışları için bir adet kamusal IP ve port sunmaktadır. Model üzerinde uygulama protokolü olarak kullanılacak RTP protokolü ise gerçek zamanlı veri transferi için bir port, geri bildirimlerde kullanılmak üzere alt bileşeni olan RTCP protokolü için ise RTP portunun bir fazlası olan diğer bir porta ihtiyaç duymaktadır. Bu durum iki farklı portun özel ağdan kamusal ağa geçiş için adreslenmesi gereksinimini ortaya koymaktadır. Ortaya konan bu problemin giderilmesi için tasarlanacak modelde RFC 3550 ile tanımlanıp daha sonra RFC 3551 ve RFC 5761 ile son hali ortaya konan RTP veri ve kontrol paketlerinin çoklanarak tek port üzerinden iletim yöntemi kullanılmıştır [47]. Bu yaklaşımla P2P iletişim altyapısında özel ağlardan kaynaklanan problemlerin en aza indirgenebileceği öngörülmektedir.

Sunulan tez çalışması ile uçtan uca tam bir tekil iletişim modeli açısından NAT ve Firewall cihazlarının oluşturduğu özel ağları da dikkate alarak, RTP ve RTCP protokollerinin bu altyapıya uygun koşturulduğu, konfigürasyon gerektirmeyen, geri beslemeli adaptif bir yaklaşımla sunulması hedeflenmektedir. Bu bağlamda kullanıcılacak metodolojiyi P2P bilgisayar ağları için koordine etmek, kimlik doğrulama, el sıkışma, durum yönetimi gibi süreçleri yönetmek için randevu sunucusu olarak XMPP protokolü kullanılmıştır.

XMPP ve P2P bilgisayar ağlarında çoklu ortam verilerinin gerçek zamanlı paylaşımına yönelik çalışmalar son yıllarda giderek artmaktadır. Özellikle bu çalışmanın temelini oluşturan uçtan-uca tam bir model geliştirilmesi noktasında, XMPP protokolünün geliştirilen NAT geçişi yöntemine uyumluluğu ve altyapı protokollerinin koşturulması için uygun eşleşme sağlaması önem arz etmektedir. Çalışma kapsamında gerçekleştirilen literatür taramasında, son zamanlarda XMPP protokolünün pek çok çalışmaya temel teşkil ettiği gözlemlenmektedir. Ancak bu protokolün sunuduğu önemli avantajlara rağmen P2P bilgisayar ağlarına uyumluluğuna yönelik tam bir model oluşturabilecek ilave protokole rastlanmamıştır.

Pankaj ve ark., çalışmalarında iş uygulamalarının teorik temellerini ortaya koymuşlar, uygun iş modelleri önermişler ve bu uygulamaların P2P iletişim bağlamında uygun altyapılar sunabileceğini vurgulamışlardır [48].

Ragavan ve ark., XMPP protokolü kullanarak çeşitli endüstriyel uygulamaları kontrol eden gerçek zamanlı veri toplama yeteneğine sahip bir uygulama geliştirmişlerdir. Bu çalışmalarında XMPP protokolünün gerçek zamanlı endüstriyel iş uygulamaları için önemli bir altyapı sunabileceğini ifade etmişlerdir [49].

Erlend ve ark., anında mesajlaşma servisi yanında XMPP protokolünü sağlık sektöründe kullanmak üzere incelemişler ve mobil platformlar için risk analizlerini ortaya koymuşlardır [50].

Sun ve ark., kurumsal kullanıcılar için XMPP tabanlı Bouncy Castle şifreleme kütüphanesine dayalı melez bir şifreleme algoritması ortaya koymuşlar ve bu algoritmayı anında mesajlaşma adımlarına uygulamışlardır [51].

Gomes ve ark., çalışmalarında XMPP tabanlı içerik yönetim mimarisi ortaya koymuş ve kullanıcı konum/lokasyon bilgilerinin sunumu için Bilgi/Sorgu (IQ) paketlerine GPS düğümü ekleyerek yeni bir model önermişlerdir [52].

Lübke ve ark., XMPP altyapısında mobil yazılım geliştiriciler için konum tabanlı grup yönetim hizmeti sunmuşlardır. Bu amaçla XMPP protokolüne uygun istemci-sunucu mimarisi ortaya koymuşlardır [53].

Khan ve ark., akıllı ev otomasyonları için zigbee tabanlı kablosuz algılayıcı ağlardan elde ettikleri ışık, sıcaklık ve termostat gibi sensör bilgilerini XMPP protokolü ile gerçek zamanlı, verimli ve güvenli bir şekilde servis tabanlı sistemlere iletebildiklerini ifade etmişlerdir. Burada XMPP protokolünün gerek gerçek zamanlı veri iletimi gerekse de erişim kontrolü noktasında bu tür uygulamalar için uygun bir platform olduğunu savunmuşlardır [54].

Yapılan araştırmalar göstermektedir ki XMPP, IP tabanlı iletişim platformları için gelişmeye devam edecek ve P2P ağ uygulamalarının gerek altyapı protokolü olarak kullanımı gerekse protokol eklentileri ile farklı uygulamalara uyumluluğu anlamında önemli kolaylıklar sağlayacaktır.

Yapılan literatür araştırmaları, bütünüyle ele alındığında, bu tez çalışmasının odaklandığı üç temel nokta olan, NAT geçişi, gerçek zamanlı veri iletimi ve sistemin koordinasyonunun sağlanmasında kullanılan randevu sunucusu ekseninde aşağıdaki çıkarımlar elde edilmiştir.

1. P2P bilgisayar ağları için öncelikle özel ağlardan kamusal ağlara çıkış için kullanılan yapılara karşı verimli bir NAT geçiş tekniği belirlenmelidir. Bu teknik istemcilerin birbirleri ile gerçek zamanlı ortam verilerinin paylaşımı

aşamasınında istemci-sunucu mimarisi kullanmaktan ziyade mümkün olan yapılar için istemci-istemci veri paylaşımı sağlamalıdır. Ayrıca kullanılacak teknik, mevcut IP ağları için ilave yapılandırma ve altyapı değişikliği gerektirmemelidir. Bu noktada ICE ve RTMFP protokollerinin kullandığı teknikler, belirli temel bir takım iyileştirmeler yapılarak NAT geçiş tekniği olarak kullanıldı. Bu teknikler, yeni bir algortma temelinde ele alındı, yeni bir yöntem önerildi ve bu yöntem bir uygulama modeli üzerinde kullanılarak değerlendirildi.

2. Ortam verililerinin aktarımı için yapılan değerlendirmelerde RTP protokolü, gerek yapısındaki esneklik, gerek servis kalitesi açısından geribesleme sağlayabilmesi, gerekse de NAT geçiş tekniklerinin öncelikle aktarım katmanında UDP bağlantıları tercih etmesinden dolayı, ortaya konan çalışmada öncelikli uygulama katmanı protokolü olarak tercih edildi. İncelenen araştırmalarda görülmektedir ki RTP/RTCP protokolü UDP temelli ortam verilirinin dağıtımında istenen servis kalitesi için ilave çalışmalara ihtiyaç duymaktadır. Bu çalışma ile sunum katmanında kullanılan ITU-T standartlarından ses ve görüntü kodlama teknikleri elde edilen geri beslemelerle otomatik belirlenerek servis kalitesinin artırımı sağlandı.
3. Son olarak P2P bilgisayar ağlarında kimlik doğrulama, yetkilendirme, el sıkışma, durum yönetimi gibi temel randevu işlevlerini yerine getirecek sunucu uygulaması için ilave protokoller geliştirildi.

P2P veri iletiminin temelini oluşturan anında mesajlaşma uygulamalarında bir standart oluşturmak için geliştirilen XMPP protokolü gerek ticari uygulamalarda gerekse de literatürde yoğun kullanım alanı bulmuştur. Bu protokolün genişletilebilir olması ilave protokollere uyumluluk noktasında önemli bir avantaj sağlamaktadır. Temel hedefi P2P bilgisayar ağlarında koordinasyonu sağlamak olan XMPP, istenilen parametrelerle istemciler arasında etkin genişleme olanağı sunmaktadır. Özellikle XMPP protokolünün istemciler arasında doğrudan veri iletimi gerektiren ortam verilerinin aktarılması için gerekli müzakere süreçlerini yönetmesi önemlidir. XMPP üzerinden oturum bilgilerinin taşınmasının yanında istemciler arasında NAT geçişi sağlayabilecek parametrelerin dağıtımı gerekmektedir. Bu bağlamda XMPP

ilave protokollerinden jingle-node olarak isimlendirilen NAT geçişi yöntemleri için oturum bilgilerinin istemciler arasında el değiştirme modeli tanıtılmıştır [55]. Bu protokol eklentisinin alt yapısı ICE protokolüne dayalı NAT geçiş tekniği için tasarlanmıştır [56]. NAT geçiş tekniği için kullanılacak ağ arayüz bilgilerinin tanımlanan paket yapıları üzerinden taşınmasının yanında, oturum tanımlama bilgilerinin de kullanıldığı çalışmada RTP oturumlarının yanında ham UDP veri aktarımı da kullanılabilmektedir. XMPP'de NAT geçiş tekniği olarak sadece ICE'ın tanımlanması nedeniyle bu protokol bu çalışmada ortaya konan yeni algoritmayla yeniden ele alındı.

Yapılan tez çalışmasının amacına yönelik incelenen literatür çalışmaları sonucunda ortaya konan değerlendirmeler ışığında tezin organizasyonu aşağıda sunulmaktadır.

1.4. Tez Düzeni

Tezin ilk bölümünde, konuya genel bir bakış açısı kazandırmaya yönelik temel bilgiler verilmiş, çalışmanın amacı sunulmuş, literatürde daha önce benzer alanlarda yapılan çalışmalar değerlendirilerek tezin temel dayanakları ve araştırma alanları tanımlanmış, çalışma mimarisi oluşturulmuş ve diğer bölümlerin içeriği kısaca sunulmuştur.

Bölüm 2'de, P2P bilgisayar ağları modeli çerçevesinde, NAT ve Firewall gibi cihazlardan kaynaklanan problemlerin giderilmesine yönelik çözüm önerileri üzerinde durulmuştur. NAT cihazlarının yapısı ele alınarak P2P bilgisayar ağları açısından tanımlanmış çözüm önerileri sunulmuştur. Bu bölümde özellikle tezin temel araştırma alanı olarak belirlenen P2P bilgisayar ağları için NAT geçiş kavramı detaylandırılmıştır.

Bölüm 3'te, geliştirilecek yöntemin kullanıldığı uygulama için aktarım ve uygulama katmanı protokolleri tanıtılmıştır. Öncelikle gerçek zamanlı aktarım protokolü olan RTP ele alınmıştır. RTP protokolünün yapısı ve P2P bilgisayar ağlarına uygun modeli sunulmuştur. Bunlara ek olarak bu bölümde RTP protokolünün servis kalitesi

için geliştirilen gerçek zamanlı aktarım kontrol protokolü RTCP tanıtılmıştır. Sonrasında, Genişletilebilir Mesajlaşma ve Durum Protokolü olarak isimlendirilebilen XMPP tanıtılmıştır. XMPP protokolü ile P2P iletişim kurulmadan önce istemcilerin yapılandırılması ve birbirleri ile haberleşebilmeleri sağlanmıştır. Bu protokolün yapılandırılması ile sistemin istemciler arası koordinasyonu sağlayan randevu sunucusu ihtiyacı karşılanmıştır.

Bölüm 4'te, tezde önerilen P2P bilgisayar ağlarında gerçek zamanlı veri akış modeli ortaya konulmuştur. Uygulamanın sunulacağı bölüm olan Bölüm 4'te ayrıca bu model üzerinden ortam verilerinin istemciler arasında iletiminin gerçekleştirildiği laboratuar çalışmaları ve sonuçları sunularak model verimlilik analizi ele alınmıştır. Bu bölüm içerisinde uygulama ve uygulamanın üç temel araştırma alanı sunulmuştur. Yapılan tüm geliştirmeler literatürde ortaya konulan diğer bir model olan ICE protokolü üzerinden kıyaslanarak bant genişliği kullanımı, iletişim altyapısı, gecikme zamanı, kullanılan paket sayıları gibi parametrelerle değerlendirilmiştir.

Bölüm 5'te ise tezin sonuçları kapsamlı olarak değerlendirilmiş ve ileride yapılacak çalışmalar için önerilerde bulunulmuştur.

BÖLÜM 2. NAT ve NAT GEÇİŞİ

2.1. Giriş

Dördüncü nesil internet protokolü (IPv4) adres havuzunun belirli sayıda host'u içerebilen adres sınıflarına bölünerek dağıtılması yaklaşımı IPv4 adreslerinin hızlıca tükenmesine neden olmuştur. Daha az sayıda host gerektiren ağlar için çok daha büyük IP aralıkları içeren B ve C sınıfı IP adresleri dağıtılmıştır. Kullanılamayan IP adreslerinin daha verimli dağıtımı için Sınıfsız Alanlar-Arası Yönlendirme (Classless Inter-Domain Routing – CIDR) olarak adlandırılan yöntem tavsiye edilmiştir. Bu yaklaşım IPv4 adresleri için göreceli bir genişleme getirse de her geçen gün artan IPv4 adres talebine kesin bir çözüm üretememiştir. Kalıcı çözüm için daha geniş adres aralığı içeren internet protokolünün yeni sürümü olan IPv6'ya geçiş, aygıt yazılımlarının güncellenmesi, dönüştürücülerin verimliliği gibi nedenlerden dolayı internet omurgasında tam anlamı ile sağlanamamıştır. Bu sorunun, yeni internet protokolüne geçilinceye kadar mevcut IPv4 kamusal IP adreslerinin internete bağlanmak isteyen belirli istemciler arasında ortak kullanılması temeline dayanan ağ adres dönüştürücüleri ile ötelenebileceği önerilmektedir. [57, 58].

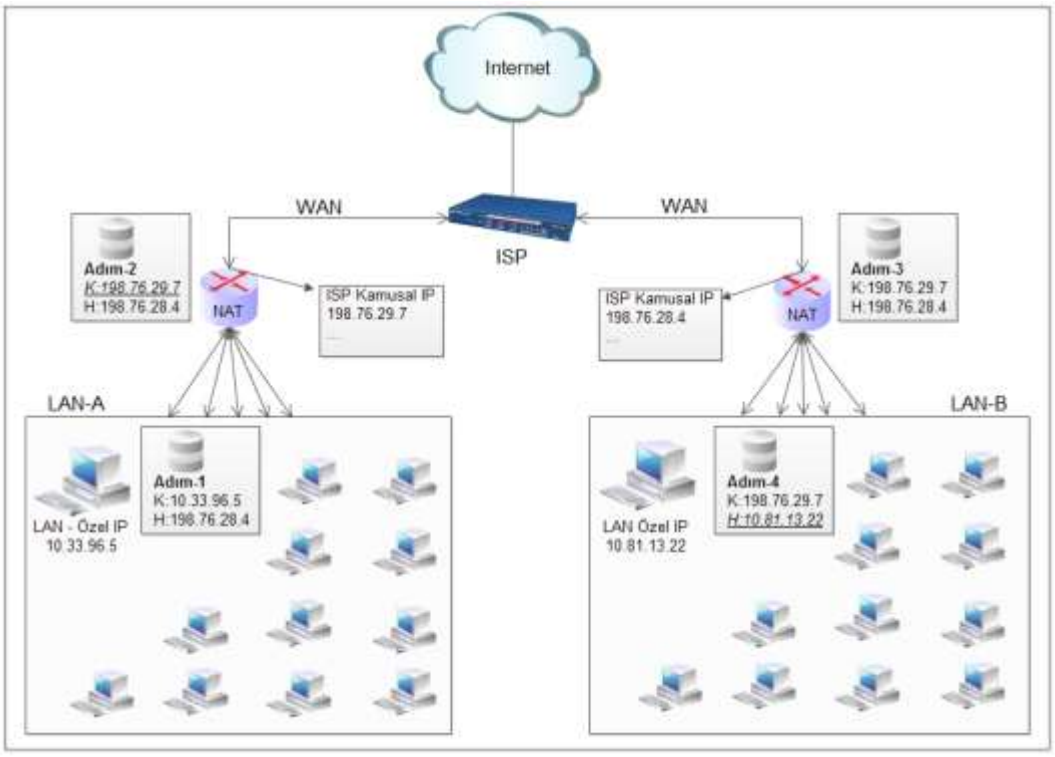
Bu bölümde, P2P bilgisayar ağlarında istemciler arası doğrudan bağlantı kurulabilirliği önündeki en temel problem olan NAT ve NAT geçişi kavramları ele alınmıştır. Mevcut NAT geçişi yöntemleri sunularak bu yöntemlerin avantaj ve dezavantajları belirtilmiştir.

2.2. NAT

Ağ adres dönüştürücüleri (Network Address Translation - NAT), 32 bitlik IPv4 adres sınırını genişletmek ve özel ağ oluşturmak için Internet Engineering Task Force

(IETF) tarafından ortaya konulan bir standarttır. NAT cihazlarının temel işlevi, internet servis sağlayıcıları (ISP) tarafından atanan kamusal IPv4 adresini, bu hizmet üzerinden internete bağlanan yerel ağlardaki tüm cihazlar için ortak kullandırılması üzerine oluşturulmuştur. O hat üzerinden internete bağlanan cihazlar kendi aralarında internette kullanılmayan özel IP adreslerine sahiptirler. NAT, kamusal ağ adreslerini özel ağ adreslerine, özel ağ adreslerini de kamusal ağ adreslerine çeviren bir ağ geçididir [59].

NAT aygıtları, ağ adresi dönüştürmenin dışında sistem yöneticilerine yerel ağları için ilave bir güvenlik katmanı sunmaktadır. İnternet omurgasında IPv6'ya geçilse bile yerel ağlarda sağladığı özelleştirilmiş yönetim ve tek bir IP adresi üzerinden birden fazla bilgisayarı internete bağlama yeteneği NAT'ların kullanımını sürdüreceğini göstermektedir [60].



Şekil 2.1. Ağ adres dönüşümü (NAT)

Ağ adres dönüştürücülerinin temel mimarisi en basit haliyle Şekil 2.1.'de görülmektedir. Şekilde belirtilen işlem adımları ele alındığında LAN-A'da yer alan

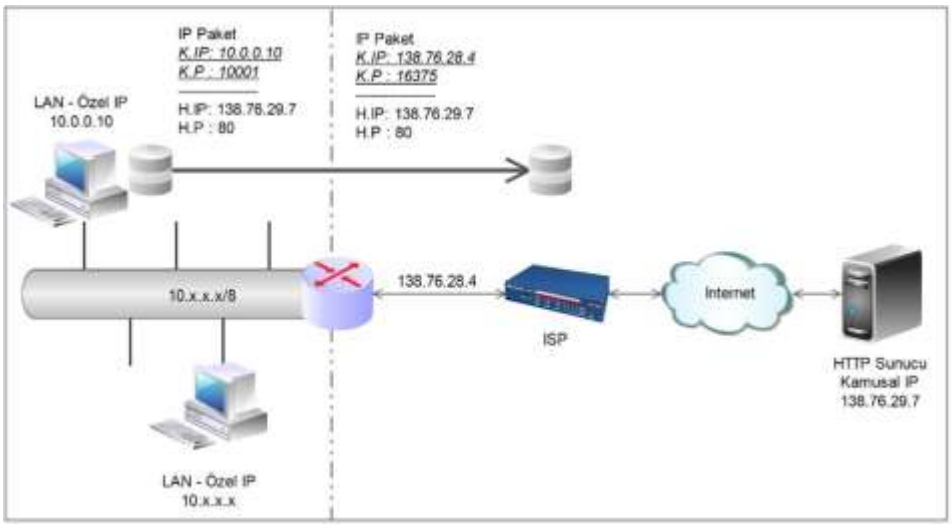
ve IP adresi 10.33.96.5 olan düğüm, LAN-B'de yer alan ve IP adresi 10.81.13.22 olan düğüme bir IP paketi iletmektedir. Bu aşamada gerçekleşen işlemler sırası ile aşağıdaki gibidir.

Adım-1'de kaynak düğüm ileteceği IP paketi için, hedef adres olarak LAN-B'nin tekil kamusal adresi olan 198.76.28.4'ü kullanırken, kaynak adres olarak kendi yerel özel adresini kullanır.

Adım-2'de iletilen IP paketi, LAN-A üzerinden WAN ağına aktarılırken ağ adres dönüştürücüsü, IP paketinin kaynak adresindeki yerel özel IP adresini, kendi tekil kamusal IP adresi olan 198.76.29.7'ye dönüştürür.

Adım-3 ve 4'te LAN-B'nin ağ adres dönüştürücüsü, WAN ağından aldığı IP paketini, yerel ağ üzerinden haritaladığı hedef düğümün özel IP adresi olan 10.81.13.22'ye dönüştürür.

Şekil 2.1'de belirtilen temel ağ adres dönüştürme işlevinin dışında NAT cihazları, iletişime geçecek iki servis arasında port dönüşümü yapmaktadır. NAT aygıtlarının bu işlevi, Ağ Adres Port Dönüşümü (Network Address Port Translation – NAPT) olarak isimlendirilmektedir. Ağ adres port dönüşüm işlemlerinin sunulduğu mimari Şekil 2.2.'de görülmektedir.



Şekil 2.2. Ağ adres port dönüşümü (NAPT)

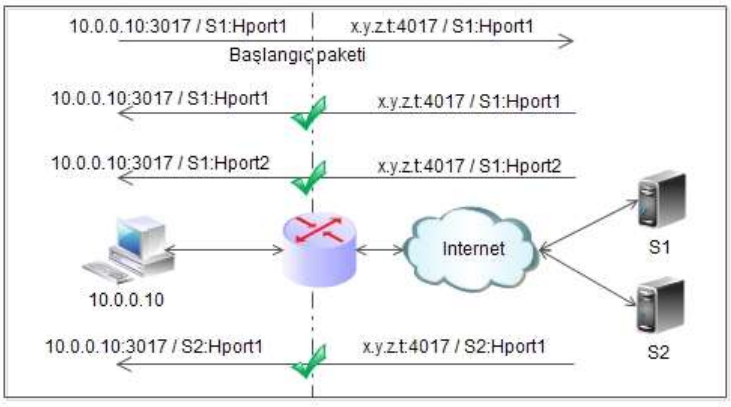
Şekil 2.2.'de sunulan modelde, yerel ağ adresi 10.0.0.10 olarak görülen istemcinin, kamusal ağ adresi 138.76.29.7 olan ve 80 portundan hizmet veren bir web sunucusu ile haberleşmesi kurgulanmıştır. Yerel ağdaki IP paketinin kaynak IP ve port değerleri yerel ağ içerisinde geçerli olan özel IP ve port bilgilerinden oluşmaktadır. Hedef adrese gönderilen IP paketi, WAN'a iletilmeden, kaynak IP değeri, o uç için ISP tarafından tahsis edilen tekil kamusal IP adresine dönüştürülürken, yerel port bilgisi ise WAN çıkışında boş olan ve 16 bit genişliğindeki bir port numarası ile eşlenir. Port numara aralığında yer alan 0-4096 arası değerler daha önceden bilinen servisler için tahsis edildiği için dönüşüm işlemlerinde kullanılmazlar.

2.3. NAT Davranışları

UDP temelli uygulamalar arasında ağ adres/port dönüşüm yöntemleri ve geri dönüş yeteneklerine göre NAT davranışları 4 grup altında tanımlanmıştır [4, 5, 23, 30, 65]. NAT geçişi çalışmaları için bu davranışların bilinmesi, karşılaşılan sorunların nedenleri ve çözüm yöntemleri açısından önemlidir.

2.3.1. Tam koni (Full Cone – FC)

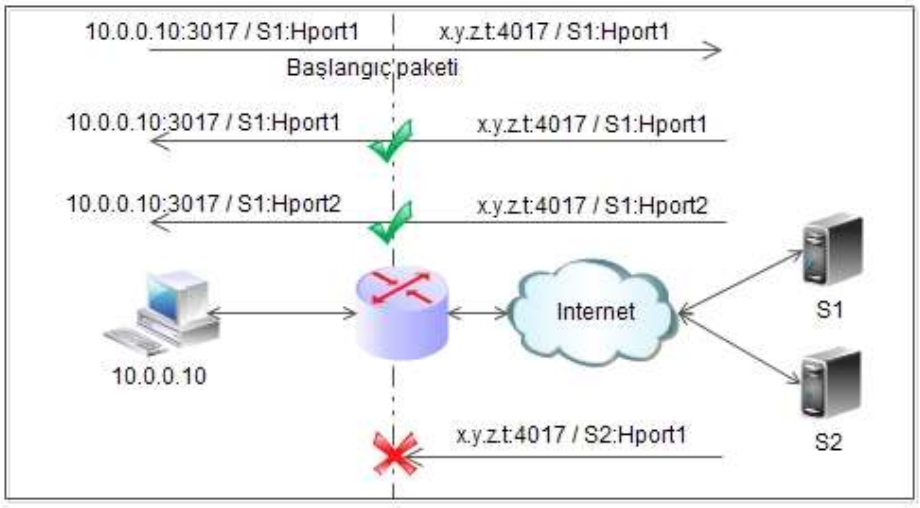
Bu tip ağ adres dönüştürücüler, tüm istekler için özel IP adres ve portlar ile aynı kamusal IP adres ve portları eşleştirirler. Eşleştirilen bu kamusal IP adresler ve portlar üzerinden yerel kullanıcılara erişmek mümkündür (Şekil 2.3.) [4, 5, 23, 30, 65].



Şekil 2.3. FC NAT mimarisi

2.3.2. Adres kısıtlamalı koni (Restricted Cone – RC)

Tüm istekleri full cone NAT gibi eşleştirir. Ancak sadece erişim sağlanan kamusal IP adresi üzerindeki herhangi bir port ile bağlantının kurulduğu yerel ağdaki özel IP adresine erişim sağlanabilir. Farklı kamusal IP adreslerinden gelen istekler dönüşüm işlemine tabi tutulmamaktadır (Şekil 2.4.) [4, 5, 23, 30, 65].

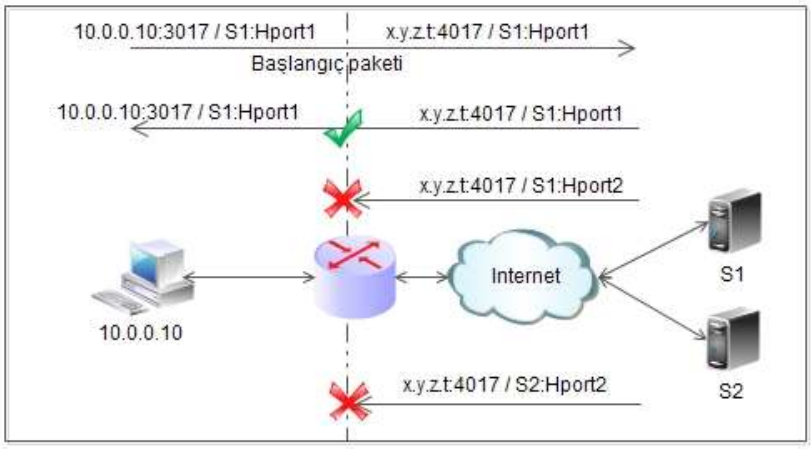


Şekil 2.4. RC NAT mimarisi

2.3.3. Port kısıtlamalı koni (Port Restricted Cone – PRC)

Eşleştirme restricted cone NAT gibidir. Ancak sadece haritalanan kamusal IP adresi ve port üzerinden, yereldeki özel IP adresine erişim sağlanabilir. Erişilmek istenen kamusal IP adresi ve port dönüşüm yapılan kamusal IP adresi ve porttan farklı olduğunda gelen istekler haritalanmamaktadır (Şekil 2.5.) [4, 5, 23, 30, 65].

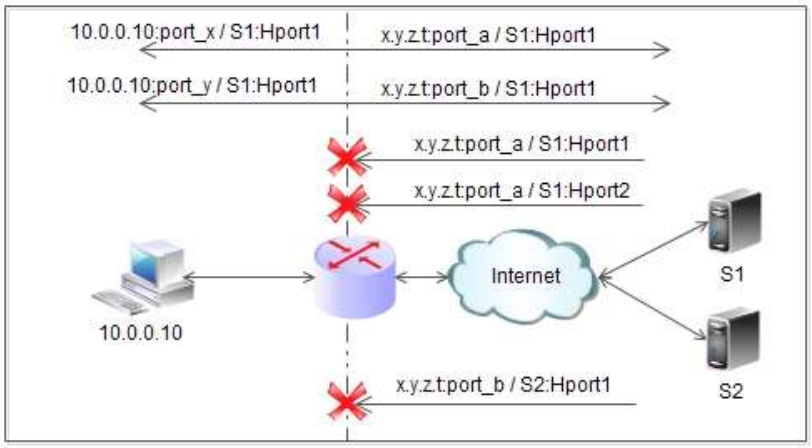
Genel anlamda koni NAT davranışına sahip tüm NAT tipleri, haritaladıkları IP ve port çiftleri üzerinden iletişim kurabilme olanığı sunmaktadır. Bu özellikleri ile koni NAT'lar, NAT geçişi yöntemleri açısından arkasında yer alan düğümler ile doğrudan bağlantı kurulabilir NAT'lar olarak tanımlanabilir.



Şekil 2.5. PRC NAT mimarisi

2.3.4. Simetrik (Symmetric – SYM)

Bu tip NAT'lar herbir iletişim için farklı port dönüşümü gerçekleştirirler. Aynı yerel özel IP ve port üzerinden farklı bir kamusal istek üretildiğinde rastlantısal yada artırımsal olarak belirlenen yeni bir port ile haritalanırlar. Sadece isteklerin alındığı kamusal IP adres ve port tarafından paket gönderilebilir (Şekil 2.6.) [4, 5, 23, 30, 65].



Şekil 2.6. SYM NAT mimarisi

Tanımlanan ilk üç NAT davranışı, koni-tip NAT olarak isimlendirilir [5, 65]. Bu davranış tipine sahip NAT'larda, ağ adres/port dönüşümüne uğrayan yerel özel IP:port ile kamusal IP:port eşleşmesi üzerinden, NAT yönlendirme tablosundaki eşleşme kaydı silininceye kadar iletişim devam etmektedir. Symmetric NAT'da ise

herbir bağlantı için yeni bir port ile kamusal IP:port eşleştirmesi rastlantısal yada artırımsal olarak yeni bir kayıt olarak tanımlanmaktadır.

Ağ adres dönüştürücülerin, gerek tekil kamusal IP adreslerinin, yerel ağlardaki pek çok düğüm tarafından ortak kullanılabilirliğini sağlaması, gerekse de sistem yöneticilerinin kontrollünü kolaylaştıran ve ilave bir güvenlik katmanı sunan özel ağ oluşturması gibi avantajlarının yanında, internetin uçtan-uca modeline uygun olmaması, tüm yerel – kamusal bağlantıların tek bir cihazın eşleme listesine mahkum olması, IP katmanındaki güvenlik politikalarını bozması gibi dez avantajları bulunmaktadır [61]. Bu dezavantajlardan en dikkat çekici olanı ise P2P iletişim kurmak isteyen istemcilerin birbirleri ile doğrudan haberleşme olanağı bulamamasıdır. SIP ve H323 gibi IP tabanlı ses iletimi uygulamaları için özel yapılandırmalar bazı NAT aygıtları üzerinde tanımlanmış olsa dahi bu durum tüm internet altyapısında kullanılan aygıtları kapsamamakta ve yeni uygulamalar için genel bir çözüm oluşturmamaktadır. P2P iletişimde ağ adres dönüştürücülerden kaynaklanan sorunların çözümü için geliştirilen yaklaşımlar literatürde NAT geçişi (NAT traversal) olarak isimlendirilmektedir.

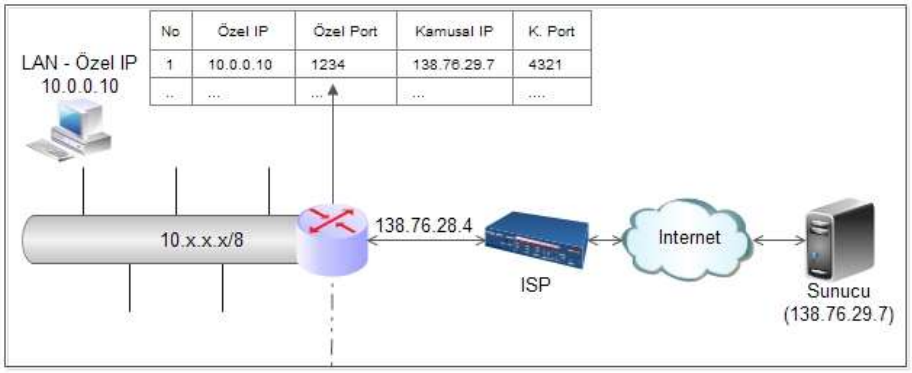
2.4. NAT Geçişi

Ağ adres dönüştürücüleri, temelde istemci/sunucu mimarileri için tasarlanmıştır. Ancak son yıllarda hızla gelişen P2P dosya paylaşımı (BitTorrent vb.), IP tabanlı sesli görüşme ve anında mesajlaşma (Skype, GTalk vb.), online oyun, video yayını gibi uygulamaların hızlı gelişimi, P2P iletişimde NAT geçişi sorununun çözümlenmesi gereken temel araştırma alanlarından olmasına neden olmuştur [62]. P2P iletişimde NAT geçişi sorununun çözümü için geliştirilen yaklaşımlar belirli ana başlıklar altında toplanabilir [65].

2.4.1. Manuel port yönlendirme

İletişim kurulmak istenilen kullanıcının yerel özel IP adresi, NAT üzerinden uygun port numarasıyla eşlenerek kamusal ağdan erişim sağlanabilir [63]. Bu yöntemde

özel adreslere erişim için manual yapılandırmaya gereksinim duyulmaktadır. Bu nedenle dağıtık istemciler arası ağ uygulamaları için uygulanabilir bir yöntem değildir. Genellikle NAT arkasındaki sunucuların kamusal ağdan erişilebilir olmalarını sağlamak için sistem yöneticileri tarafından kullanılırlar. Manuel port yönlendirmenin grafiksel sunumu Şekil 2.7.'de görülmektedir.

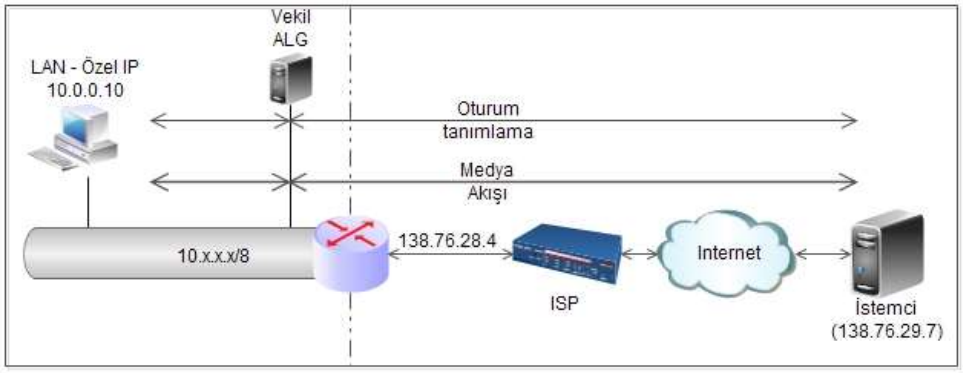


Şekil 2.7. Manuel port yönlendirme

2.4.2. Uygulama katman geçişi

Uygulama katman geçişi (Application-Level Gateway – ALG), P2P bilgisayar ağlarında istemciler arası vekil sunucu olarak kullanılmaktadır. Yerel ağ içerisinde bazen askerden arındırılmış bölgede (DeMilitarized Zone – DMZ) yada özel protokollere izin verilerek konumlandırılan ALG sunucuları, istemciler arasında oturum tanımlama protokolünü (Session Description Protocol – SDP) yönetirler. İletişim kuran düğümler arasındaki ortam verilerinden hangi düğümler arasında iletişim kurulacağının belirlenmesi gerekmektedir [64]. Bu yöntem, tüm NAT çeşitlerinde medya vekil sunucusuna ihtiyaç duyması, sadece bellirli özel uygulamaları desteklemesi [24] ve gerçek zamanlı iletişim için ilave başlıklar ve güvenlik ihlalleri oluşturmasından dolayı dağıtık uygulamalarda kullanım olanağı bulmamıştır [64].

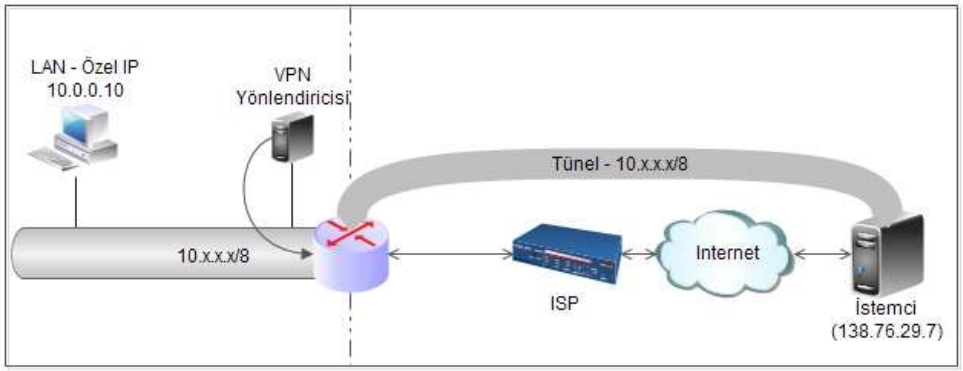
Uygulama katam geçiş yötemi örneği Şekil 2.8.'de görülmektedir.



Şekil 2.8. ALG mimarisi

2.4.3. Sanal özel ağ

P2P iletişim kurulacak düğümler arasında tünel oluşturan sanal özel ağ (Virtual Private Network – VPN) bağlantıları, sunucu tabanlı bir yöntemdir (Şekil 2.9.). Gelen istekleri kabul edebilmek için tüm yerel ağlarda tünel sunucusuna ihtiyaç duyulması, yüksek band genişliği gerektirmesi ve güvenlik gereksinimleri, bu yöntemi P2P ağ uygulamalarında etkisiz kılmıştır [11, 61].



Şekil 2.9. Sanal özel ağ

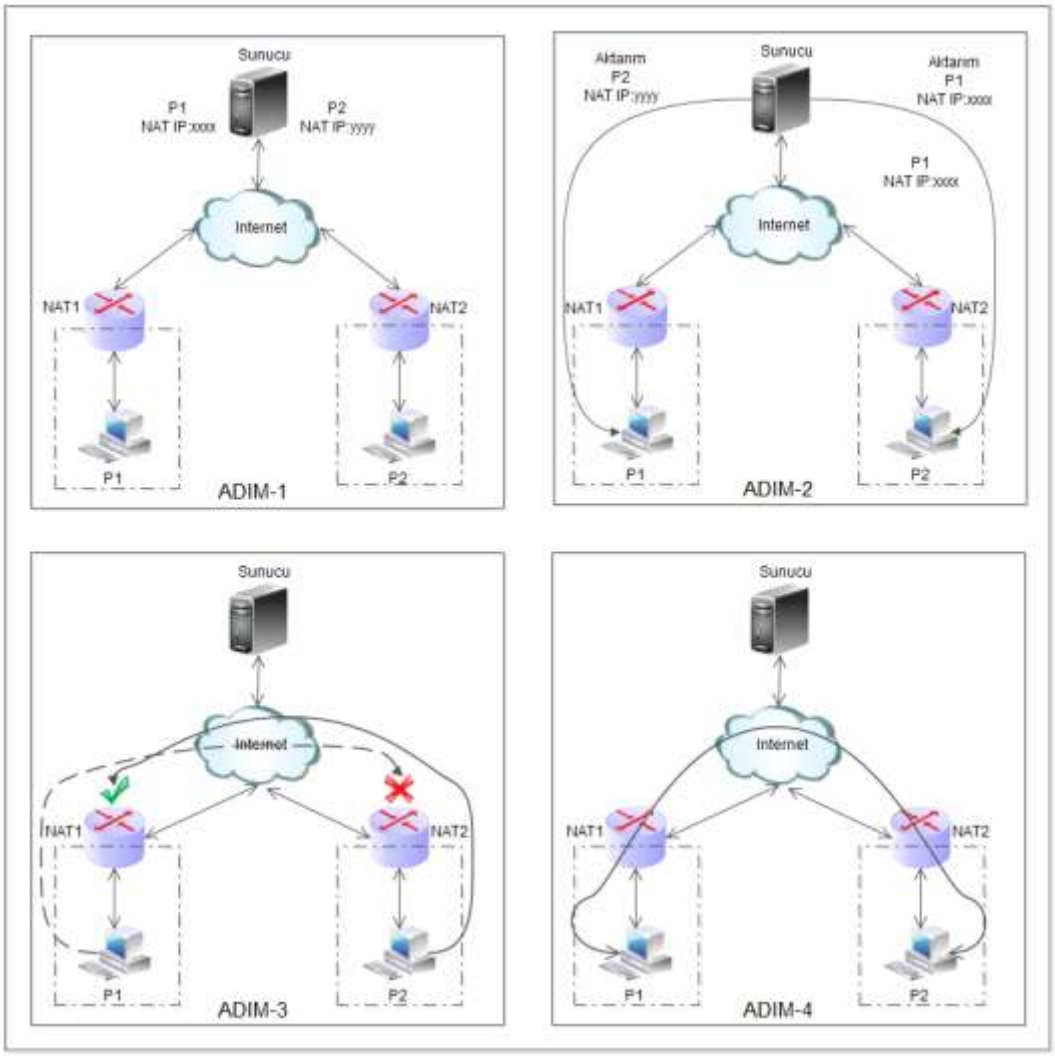
2.4.4. Evrensel tak çalıştır

Evrensel tak çalıştır (Universal Plug and Play – UPnP), çeşitli ağ cihazlarını P2P bağlamak için UPnP forumu tarafından geliştirilmiştir. UPnP desteği olan ağ cihazı otomatik olarak özel ağ adresi ile kamusal ağ adresini uygun port ile haritalamakta ve NAT geçişi sağlamaktadır [66, 67]. Son kullanıcı açısından yapılandırma

gerektirmiyor olması önemli bir avantaj olmasına karşın pek çok NAT üreticisi tarafından standart olarak desteklenmemesi yöntemin genel geçer bir çözüm oluşturmasını engellemiştir.

2.4.5. UDP kanal açma tekniği

Literatürde UDP hole punching olarak isimlendirilen bu teknik, ağ adres dönüştürücü arkasındaki istemcilerin, randevu sunucuları yardımıyla, doğrudan P2P temelli UDP oturum kurmalarını sağlamaktadır. Bu tekniğin çalışma adımları Şekil 2.10.'da görülmektedir.



Şekil 2.10. UDP kanal açma tekniği

Burada P1 ve P2 gibi iki istemcinin NAT1 ve NAT2 olarak isimlendirilen yerel ağlarda olduğu varsayılmıştır.

Adım1'de P1 ve P2 istemcileri, doğrudan bağlantı kuramayacakları için öncelikle randevu sunucusu üzerinde UDP oturum açarlar. Randevu sunucusu, P1 ve P2'nin NAT arayüzünden gelen kamusal IP ve port bilgilerini mevcut soket bağlantıları üzerinden elde eder.

Adım2'de randevu sunucusu P2'ile doğrudan bağlantı kurmak isteyen P1'e, P2'nin kamusal IP ve port bilgilerini aktarır. Benzer şekilde P2'ye de P1'in kamusal IP ve port bilgileri aktarılır. Bu aktarım sonucunda, doğrudan iletişim kurmak isteyen P1 ve P2, birbirlerinin kamusal IP ve port bilgilerini elde etmiş olurlar.

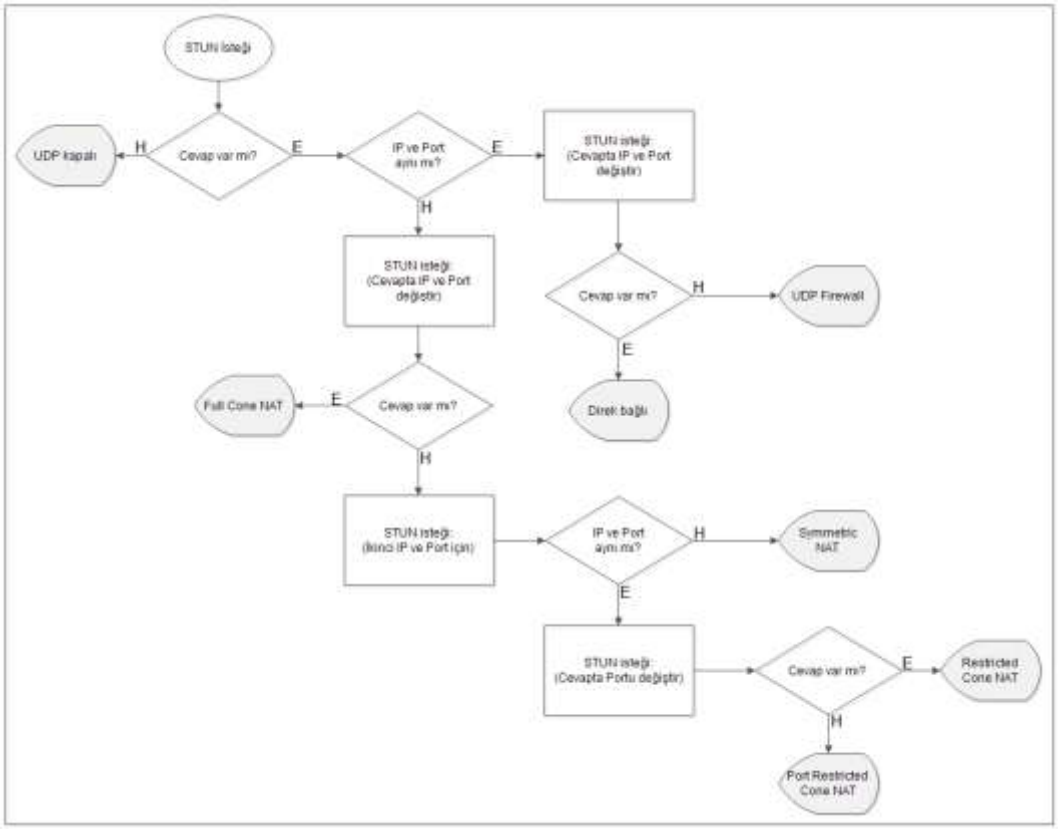
Adım3'de P1, P2'nin kamusal IP ve portu üzerinden UDP oturum kurmayı denese de P2'nin son bağlantı noktası sunucu ile olduğu için bu istek başarısız olur. Benzer şekilde P2, P1'in kamusal IP ve portu üzerinden UDP oturum kurmayı dener ve P1'in son bağlantı denemesi P2 ile olduğu için bu girişim başarılı olur. Bu aşamada P1 ve P2'nin NAT'ları kamusal IP ve port eşleştirmelerini birbirlerinin kamusal IP ve portlarıyla güncelleyecektir. Son aşamada P1 de P2 ile NAT eşleştirmeleri uygun olduğu için doğrudan UDP iletimi yapabilecektir. Bu adımdan sonra P1 ve P2 arasında UDP veri aktarımı devam eder [68, 69].

Bu teknik Symmetric NAT tipindeki ağ adres dönüştürücülerinde uygun port çiftlerini randevu sunucuları üzerinden değiştirse de bir sonraki hedefi değişen yeni istekte kamusal portlar yenileneceği için uygun eşleştirmeyi yapamaz. Ancak istemcilerden her ikisi birden Symmetric NAT yada biri Symmetric NAT diğeri Port Restricted NAT tipindeki ağ adres dönüştürücüleri arkasında olmadığı durumlarda ilave aktarım sunucusuna gereksinim duymadan uygun eşleştirmeyi yapabilmektedir [5].

Yöntemin sadeliği ve son kullanıcı için özel yapılandırma gerektirmemesi önemli bir avantaj iken uygun port eşleştirmelerin yapılamadığı durumların varoluşu bu yöntemin tek başına genel bir çözüm oluşturamamasına neden olmaktadır.

2.4.6. İnteraktif bağlantı kurulumu

İnteraktif bağlantı kurulum (Interactive Connectivity Establishment – ICE) protokolü [7], STUN [5] ve TURN [6] gibi temelde UDP kanal açma tekniğini kullanan NAT geçişi protokolleri için bir arayüz oluşturmuştur. Bu yaklaşımda doğrudan bağlantı kurulamayan durumlarda TURN sunucuları kullanılırken, diğer durumlarda STUN sunucusu üzerinden uygun kamusal–özel IP adresi ve port haritalaması yapılır.



Şekil 2.11. STUN protokolü NAT tipi belirleme algoritması

STUN protokolü RFC 3489 ile tanımlanmış ve daha sonra RFC 5389 ile oturum geçiş aracı olarak isimlendirilerek son halini almıştır. STUN protokolünün bu sürümünde temel mimaride bir değişiklik olmamasının yanında, bir uygulama

çerçevesi oluşturacak şekilde standartların tanımlanması, NAT tiplerinin belirlenmesinde ortaya çıkan sorunların giderilmesi, kamusal IP port eşleşmelerinde ortaya çıkan bazı uyumsuzlukların giderilmesi ve güvenlik noktasındaki iyileştirmeler gibi düzeltmeler ve standartlaştırmalar yapılmıştır [70].

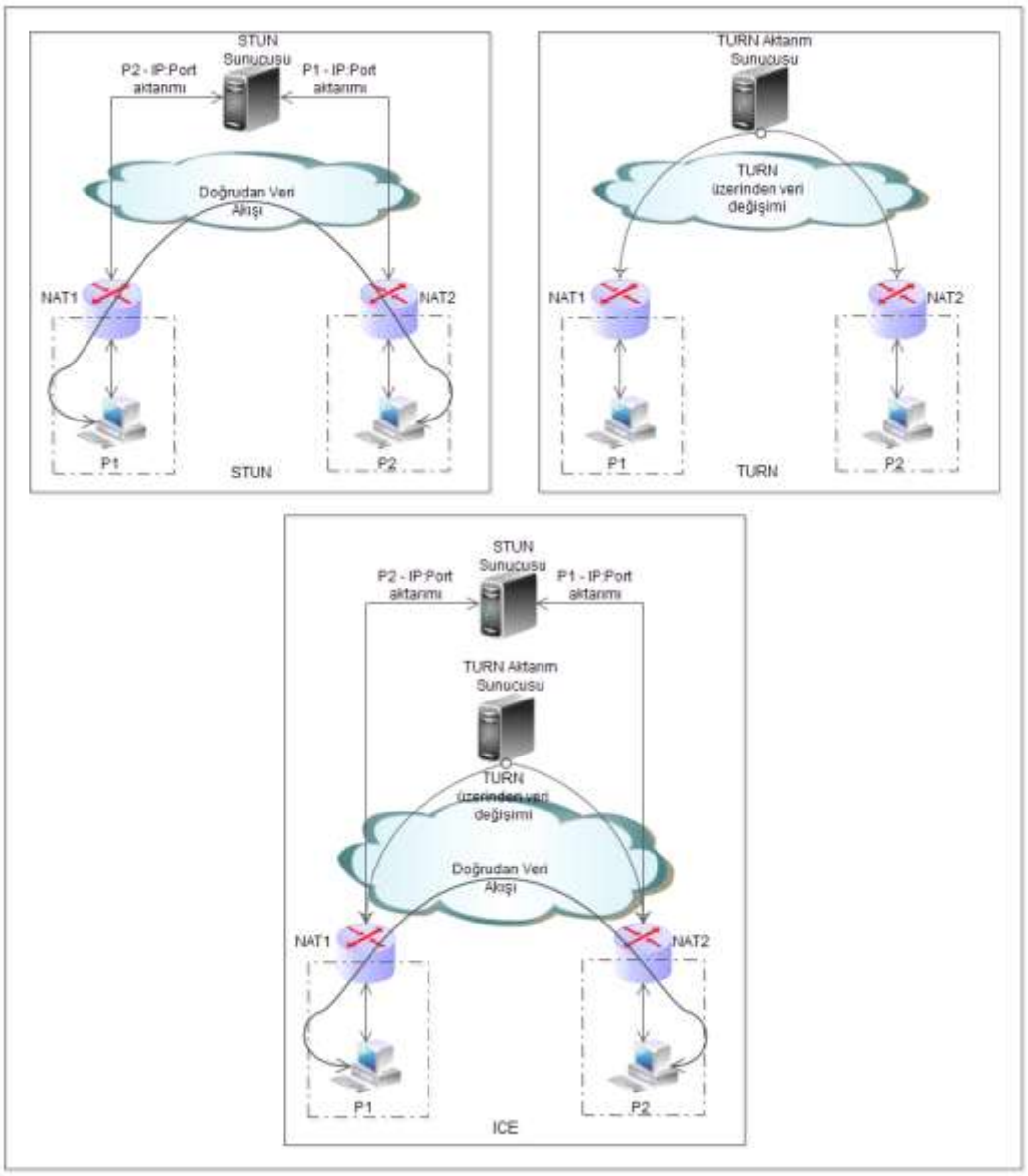
STUN protokolü istemcilerin NAT tiplerini belirleyebilmek için bir algoritma tanımlamıştır. Bu algoritma bağlamında NAT tipleri tanımlanırken istemcinin bağlantı arayüzleri de tanımlanabilmektedir [5]. STUN protokolünün NAT tiplerini belirlemek için tanımladığı algoritma Şekil 2.11.'de görülmektedir.

Bu algoritmada öncelikle istemci, STUN sunucusunun bilinen kamusal IP ve portuna (standardı UDP 3478) STUN bağlantı isteği (binding request) üretir. Eğer STUN sunucusundan cevap alınamıyorsa istemcinin UDP protokolü üzerinden iletişim kurulamayacağına karar verilir [5].

Eğer cevap alınabiliyor ise gelen cevap paketi içerisine STUN sunucusu tarafından yerleştirilen istek paketinin IP adres ve port kopyaları ile cevap paket başlığındaki IP adres ve port adresleri karşılaştırılır. İlgili veriler aynı ise istemcinin NAT arkasında olmadığı belirlenmiş olur. Bu aşamada istemci, STUN sunucusundan farklı IP ve port adresi ile NAT üzerinde kendi haritalandığı kamusal IP ve portuna bağlanmasını ister. Eğer bu isteği sonucunda STUN sunucusu kendisinde tanımlı ikinci bir IP ve farklı bir port üzerinden istemciye cevap dönebilmiş ise istemcinin internete NAT kullanmadan doğrudan bağlantığı belirlenmiş olur. Eğer cevap alınamadı ise istemcinin symmetric UDP firewall kullandığı anlaşılır [5].

İlk alınan cevaptan sonra, gelen cevap paketi ile paket içeriğindeki kopya IP ve port bilgileri uyuşmuyor ise istemcinin NAT arkasında oluduğuna karar verilir ve bir önceki adımda yapılan STUN sunucusunun farklı IP ve port üzerinden cevap isteği tekrarlanır. Farklı IP ve porttan gelen cevap paketi başlığındaki IP ve port ile paket içeriğindeki IP ve port aynı ise istemcinin full cone NAT arkasında olduğuna karar verilir. Veriler uyuşmuyor ise, STUN sunucusunun farklı IP ve port üzerinden istediği istekte kullandığı IP ve port numarasına ilk adımda olduğu gibi STUN

bağlantı isteği (binding request) üretilir. Bu istekten alınan paket başlığındaki IP ve port farklı ise istemcinin symmetric NAT arkasında olduğu belirlenir. Eğer cevap paketindeki başlık ile içerikteki kopya IP ve port aynı ise bu durumda istemcinin restricted cone NAT arkasında mı yoksa port restricted cone NAT arkasında mı olduğuna karar verebilmek için STUN sunucudan sadece yeni bir port kullanarak cevap dönmesi istenir. STUN sunucunun yeni belirlediği port üzerinden cevap alınmış ise istemcinin restricted cone NAT arkasında, eğer cevap alınamamış ise istemcinin port restricted cone NAT arkasında olduğuna karar verilir [5, 28].



Şekil 2.12. STUN, TURN ve ICE

STUN protokolünün yukarıda belirtilen NAT tiplerini belirleme algoritması sonucunda, P2P iletişim oluşturmak isteyen istemcinin belirlenen arayüz ve NAT davranış durumuna göre istemci full cone, restricted cone ve port restricted cone NAT arkasında iken doğrudan UDP bağlantı kurulabileceği, symmetric NAT arkasında olduğu durumda ise bağlantı UDP bağlantı kurulamayacağı belirlenmiş olur [27, 28].

STUN protokolü UDP kanal açma tekniğinde olduğu gibi itemcilerin doğrudan bağlantı kuramadıkları durumlarda işlevsiz kalmaktadır. Bu durumda TURN sunucusu üzerinden medya aktarımı yapılması önerilmiştir.

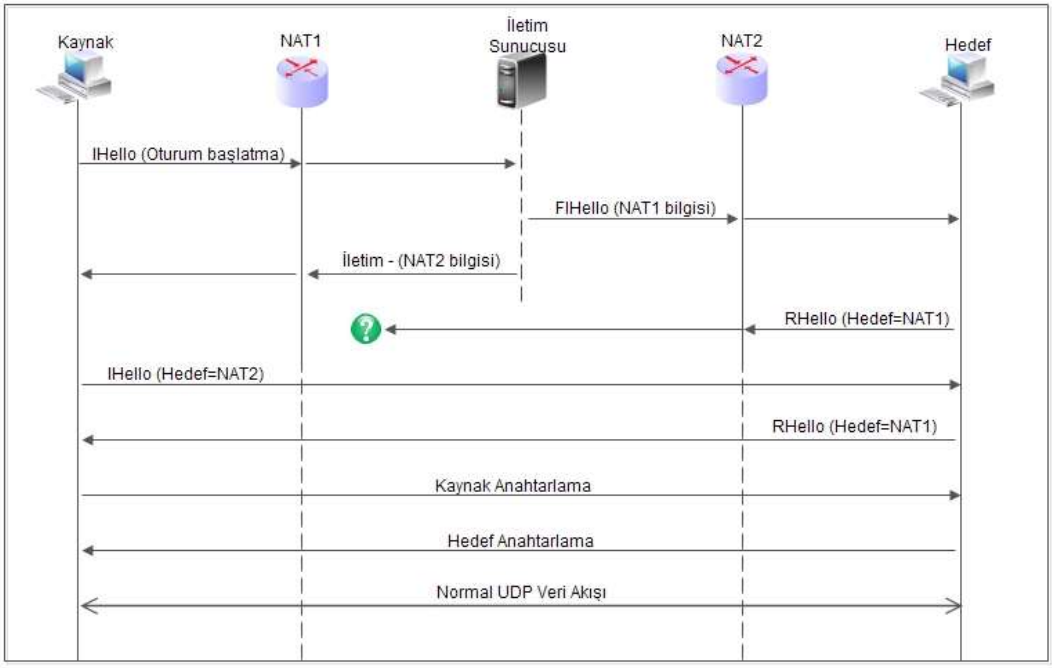
Sonuç olarak doğrudan bağlantının kurulabildiği durumlar için STUN, kurulamadığı durumlar için RFC 5766 ile tanımlanan TURN protokolünün kullanıldığı genel bir çözüm oluşturulmaya çalışılan ICE protokolü önerilmiştir [28, 71]. ICE protokolü, istemcinin en uygun bağlantı yolunu bulabilmesi için tüm ağ arayüzlerini, STUN ve TURN sunucuları üzerinden ürettiği üyelik çiftleri ile istemcide tanımlanan üyelik çiftleri üzerinden kontrol etme prensibine dayanmaktadır. Bu durum, yüksek bağlantı test süresi ve fazla kontrol paketi kullanması gibi dezavantajları ortaya çıkarmaktadır [30]. STUN, TURN ve ICE protokollerinin çalışma mimarileri Şekil 2.12.'de görülmektedir.

2.4.7. Gerçek zamanlı medya akış protokolü

Gerçek zamanlı medya akış protokolü (RTMFP) [8], ICE'ında temel aldığı STUN protokolü gibi UDP kanal açma tekniğini kullanmaktadır. Sadece UDP tabanlı doğrudan bağlantıları destekleyen protokol, düşük gecikme süresi, tıkanıklık kontrolü, media dağıtım sunucusuna ihtiyaç duymadan P2P veri dağıtımı ve veri önceliklendirmesi gibi temel sorunlar üzerine odaklanmıştır [33, 72]. NAT geçişi yöntemi olarak istemci UDP arayüz verilerini iletim ve yönlendirme işlevi gören randevu sunucuları üzerinden değiştirerek UDP kanal açmayı dener [72].

RTMFP protokolü doğrudan bağlantı kurulamayan ve UDP paketlerinin engellendiği NAT arkasındaki kullanıcılar için P2P iletişimi desteklemez [72]. Bu durumda, TCP tabanlı gerçek zamanlı mesajlaşma protokolünün (RTMP) kullanılması önerilmektedir [96]. Ancak RTMP, istemci-sunucu mimarisine dayanan, TCP tabanlı, yüksek band genişliği gerektiren yapısı ile P2P iletişim gerektiren gerçek zamanlı uygulamalar açısından verimsizlik oluşturmaktadır [34].

RTMFP protokolü NAT geçişi yöntemi olarak istemciler arasında, iletim sunucusundaki UDP oturumlarından, endpoint discriminator (EPD) verilerini birbirleri arasında aktardıktan sonra UDP kanal açma tekniğine uygun süreçleri gerçekleştirmektedir. EPD verileri EPD-tip ve EPD-veri alanlarından oluşmaktadır. EPD tip alanı, haberleşmenin istemci-sunucu veya istemci-istemci arasında kurulduğunu belirlemek için kullanılırken, EPD-veri alanı ise, istemci-sunucu durumunda iletim sunucusunun URL adresi, istemci-istemci durumunda ise eş kimliği değerlerini içerir [73].



Şekil 2.13. RTMFP oturum başlatma ve NAT geçişi

Protokolün oturum başlatma ve NAT geçişi algoritması Şekil 2.13.'te görülmektedir. Kaynak ve Hedef istemcilerin İletim sunucusunda kamusal NAT arayüzlerini

tanımlayan oturumları kaydedilmiştir. İletim sunucusu Kaynak istemcinin NAT arayüzlerini içeren IHello paketini FIHello paketi olarak Hedef istemciye iletir. Aynı zamanda Kaynak istemciye, NAT2'nin kamusal adreslerini içeren iletim paketini yeniden yönlendirir. Hedef FIHello paketine cevap olarak NAT1'in kamusal adresine RHello paketini iletir. Bu noktada eğer iletim sunucusundan önce RHello paketi gitmiş ise NAT1 bu paketi engelleyecektir. Bu durum yeniden yönlendirmeden sonra oluşmuş ise RHello paketi Kaynak istemciye ulaşacaktır. Bu aşamada oluşan port aktarımından dolayı Kaynak istemcinin NAT2'yi hedef alan IHello paketi Hedef istemciye ulaşacaktır. Devam eden süreçte Kaynak ile Hedef istemciler arasında rutin güvenlik süreçleri tamamlanır ve doğrudan UDP veri akışı başlatılır [8].

2.5. Sonuç

P2P uygulamaları açısından istemciler arası doğrudan bağlantı kurulumu bu tür uygulamaların en önemli gereksinimlerindendir. Ancak NAT arkasındaki kullanıcıların birbirleri ile doğrudan bağlantı kurabilmesi, kamusal-özel ağ geçişi yapılamadığı için mümkün olamamaktadır. Bu bölümde öncelikle NAT ve davranışları ifade edildikten sonra NAT geçişi sorununun giderilmesine yönelik geliştirilen yöntemler ele alınmıştır. Tanımlanan yöntemlerden pek çoğu ilave yapılandırmalar gerektirmesinden ötürü P2P uygulamaları için uygun yöntem olarak görülmemektedir. Bu yöntemlerden dağıtık mimarilere uygunluğu ve tüm NAT davranışları için bağlanılabilirlik garantisi sunan ICE, uygulama geliştiriciler ve araştırmacılar açısından en çok kabul gören yöntem olarak dikkat çekmektedir. Ancak ICE yöntemi, uzun bağlantı kurulum süresi ile yüksek paket ve band genişliği kullanımı gibi servis kalitesini etkileyen önemli dezavantajlara sahiptir. NAT geçişi için geliştirilen bir başka yöntem olan RTMFP ise koni tip NAT'lar için bağlantı kurulum verimliliği ile önemli iyileştirmeler ortaya koysa da simetrik NAT davranışına sahip NAT'larda bağlantı sağlayamaması önemli dezavantaj oluşturmaktadır. Bu tez çalışmasında geliştirilen yöntem için kullanılan teorik altyapının bilinmesi ve mevcut yöntemlerin tanımlanması çalışmanın değerlendirilebilmesi açısından önemlidir.

BÖLÜM 3. GERÇEK ZAMANLI AKTARIM

3.1. Giriş

P2P ağ uygulamalarında, gerçek zamanlı verilerin taşınması için aktarım katmanı protokollerinin bilinmesi ve bu protokollere uygun iletim kanallarının oluşturulması gerekmektedir. Ağ uygulamalarında aktarım katmanı protokolleri, uygulama katmanı protokolleri ile uyumlu olmalıdır. Bu protokoller, mimari açıdan farklı katmanlarda tanımlanmış olsalar da ağ uygulamaları açısından belirli gereksinimlerin birlikte sağlanmasını gerektirir. Bu bölümde öncelikle P2P ağ uygulamaları için düğümler arası veri aktarımında kullanılan protokoller ele alınmıştır. Aktarım katmanı protokollerin bilinmesi, geliştirilecek NAT geçişi yöntemini doğrudan ilgilendirmektedir. Ayrıca bu bölümde, geliştirilen NAT geçişi yönteminin ve gerçek zamanlı aktarım protokolünün gereksinimlerini sağlayan, düğümlerin randevulaşması ve el sıkışması gibi süreçlerin yönetildiği uygulama katmanı protokolü olarak kullanılan XMPP tanımlanmıştır. XMPP, geliştirilen yeni NAT geçişi yönteminde kullanılan oturum tanımlama parametrelerinin dağıtımında ve uygulamanın istemci-sunucu haberleşmesinde kullanılmıştır.

3.2. Aktarım Katmanı

Ağ uygulamalarında veri aktarım süreci, bütün bir modelin katmanlarından birisi olmasının yanında tüm mimarinin kalbi olarak tanımlanabilir. Aktarım katmanı, kaynak düğümden çıkan bir paketi, hedefe ulaşıncaya kadar güvenilir ve etkin olarak taşımalıdır. Genel konsept olarak aktarım, uygulamalar arasında uçtan-uca, bağlantı sağlanması olarak tanımlanabilir. IP ağları için aktarım katmanında kullanılan iki protokol TCP ve UDP'dir [16, 17]. Bu protokoller bağlantılı ve bağlantısız oluşları

ile gerek yapıları gerekse de işlevsellikleri bakımından farklı kullanım alanları bulmaktadır.

Gerçek zamanlı uygulamalarda aktarım protokolleri IP protokolü ile bağlantılı olarak taşınan verinin iletimini kontrol etmek için kullanılır. Gerçek zamanlı multimedia içeriğin taşınmasında kullanılan 3 temel uygulama katmanı protokolü UDP, TCP ve UDP temelli RTP'dir [16, 17]. Bu bölümde aktarım katmanında kullanılan bu protokoller detaylandırılacak ve yapılan çalışmanın bu katmandaki metodolojisi sunulacaktır.

3.2.1. Aktarım denetim protokolü

TCP / IP protokol kümesi, Aktarım Denetim Protokolü (Transmission Control Protocol – TCP) ve Kullanıcı Veribloğu Protokolü (User Datagram Protocol – UDP) olarak isimlendirilen iki aktarım katmanı protokolüne sahiptir. Günümüz internet trafiğinin yaklaşık %75'i web (HTTP), e-posta, dosya transferi, anlık mesajlaşma, online oyun ve bazı video akış uygulamaları (örneğin, YouTube) TCP tabanlı uygulamalardır. İnternet trafiğinin kalan %25'lik kısmı ise Alanadı İsim Sunucusu (Domain Name Server – DNS) ve gerçek zamanlı multimedia uygulamaları gibi UDP tabanlı uygulamalara aittir.

TCP protokolü, bağlantı yönelimli, noktadan noktaya güvenilir bir aktarım protokolüdür. Bağlantı yönelimli, bir aktarım oturumu başlamadan önce üç yönlü el sıkışma yoluyla gönderici ve alıcı arasında bir bağlantı kurmak anlamına gelir. TCP başlığında, SYN ve ACK bayrak bitleri ilk bağlantı kurulumunda kullanılır. Her TCP başlığı 16-bit kaynak port numarası, 16-bit hedef port numarası, 32-bit sıra numarası, 32-bit alındı numarası, 4-bit TCP başlık uzunluğu, FIN, SYN, RST, PSH, ACK, URG olarak isimlendirilen bayrak bitleri, 16-bit toplamsal hata denetimi, 16-bit acil işaretçisi ve seçenek alanlarından oluşmaktadır. Asgari TCP başlığı 20 byte (seçenek alanı kullanılmadığında) 'dır.

Sıra numarası ve onay numarası, gönderim akışında, gönderilen paket konumunu tanımlamak ve (ACK bayrağı ile) ilgili paketlerin alınmasını kabul etmek için kullanılır. Kayıp paketler için yeniden iletim bildirim mekanizması TCP paketlerinin güvenilir iletimi için anahtardır. 16-bit pencere boyutu, akış kontrolü (hızlı veya yavaş gönderme) üzerinden gönderme ve alma işlemlerini garanti eder. Tıkanıklık kontrol mekanizması, ağda tıkanıklık olasılığını gösterir bir kayıp paket olduğunda, gönderim bit hızını azaltarak ağ tıkanıklığını önleyecektir. TCP tıkanıklık mekanizması internetin sağlıklı çalışması için çok önemlidir.

<table>
<tr><th colspan="9">0-15</th><th>16-31</th></tr>
<tr><td colspan="9">16 bit kaynak port</td><td>16 bit hedef port</td></tr>
<tr><td colspan="10">32 bit sıra numarası</td></tr>
<tr><td colspan="10">32 bit onay numarası</td></tr>
<tr><td>Başlık Uzunluğu</td><td colspan="2">Saklı</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td>16 bit pencere boyu</td></tr>
<tr><td colspan="9">16 bit checksum</td><td>16 bit acil gösterici</td></tr>
<tr><td colspan="10">Seçenekler (gerekiyorsa)</td></tr>
</table>

Şekil 3.1. TCP paket başlığı

TCP, güvenilir ve tıkanıklık kontrollü bir aktarım sağlar. Fakat gerçek zamanlı iletişim uygulamalarında aktarım katmanında TCP kullanımı uç düğümler, erişim bağlantıları ve ağ kabuğunda çeşitli sorunlara yol açmaktadır. İşletim sistemleri genel olarak yeni bir TCP bağlantısı kabul etmek için azami düzeyde kısıtlamalar getirir. Örneğin eşlerden biri çok sayıda TCP bağlantı isteği aldığında diğer TCP istekleri için yanıt veremeyebilir. Yada aynı istemci üzerinde çalışan web tarayıcılar gibi diğer ağ uygulamaları önemli ölçüde bağlantı kayıplarıyla karşılaşabilir. Özellikle ev kullanıcıları gibi erişim yönlendiricileri ortak ve genellikle sınırlı kapasitede kaynağa sahip yerel ağ ortamlarında bir kullanıcı TCP bağlantısı üzerinden multimedya iletişim yapıyorken aynı yerel ağ içerisindeki farklı kullanıcılar kötü ağ bağlantısı ile karşılaşırlar [74]. Yine NAT arkasındaki kullanıcıların P2P iletişim kanalı kurulması noktasında NAT'ların TCP NAT geçişi verimlilikleri oldukça kötüdür. Bu durum neredeyse tüm veri iletimini P2P olmaktan çıkarıp medya sunucusu üzerinden

yürütülmesi sonucunu ortaya çıkarmaktadır [23-30]. Tüm bu nedenlerle son zamanlarda multimedia içeriğin TCP yerine UDP kullanılarak taşınması eğilimi ortaya çıkmıştır [74].

Yukarıdaki özellikleri nedeniyle TCP, ağırlıklı olarak e-posta, ftp ve web uygulamaları gibi gecikme duyarsız, yüksek güvenilirlik gerektiren uygulamalar için kullanılır. Alındı onayı, yeniden iletim, tıkanıklık kontrolü gibi özellikler gerçek zamanlı multimedia uygulamaları için uygun değildir. Bu durum gerçek zamanlı multimedia uygulamaları için UDP protokolünü, öncelikli tercih edilmesi gereken seçenek haline getirmiştir [16, 17].

3.2.2. Kullanıcı veribloğu protokolü

UDP, TCP ile karşılaştırıldığında yapısı ve fonksiyonları oldukça basit bir protokoldür [74]. Şekil 3.2'de görüldüğü gibi kaynak port numarası için 16 bit, hedef port numarası için 16 bit, paket uzunluğu için 16 bit ve geri kalan 16 bit ise hata tespiti sağlamak için ayrılmıştır. TCP'de mevcut olan bağlantı kurulum aşaması, akış kontrolü, tıkanıklık kontrolü ve yeniden iletim mekanizmaları UDP'de yoktur.

UDP, bağlantısız ve yeniden iletim mekanizması olmadığı için TCP'ye göre veri transferi daha hızlıdır. Ayrıca sıra numarası ve onay mekanizması UDP paket yapısında tanımlanmadığından, aktarılan paketlerin doğru sırası ve bir paketin alınıp alınmadığı bilinmemektedir. Bu durum UDP ile aktarımı, hızlı ama güvenilmez kılar. UDP'nin paket kaybı, gerçek zamanlı multimedia içeriğin aktarıldığı uygulamalarda bir ölçüde tolere edilebilir. UDP ayrıca noktadan noktaya ve çok noktaya yayın uygulamaları için uygundur.

Soket noktasında UDP paketleri kendi hedef soketi üzerinden hedef'e gönderilir. İletilen verinin hedefe ulaşıp ulaşmadığı ağın o anki durumuna bağlıdır. Ayrıca IP ağlarının doğası gereği, bazı paketler çoğaltılmış olabilir ve bazı paketler de sırası bozulmuş şekilde gelebilir. Bu durum gerçek zamanlı multimedia uygulamaları için iletilen verinin alıcı tarafında doğru sırayla oynatılması ile çözülebilmektedir [21].

Bu tür uygulama noktasındaki sorunları çözümlemek için Gerçek Zamanlı Aktarım Protokolü (Real-time Transport Protocol - RTP)'nün geliştirilmesi fikri ortaya çıkmıştır [16, 17].

0-15	16-31
16 bit kaynak port	16 bit hedef port
16 bit paket uzunluğu	16 bit checksum

Şekil 3.2. UDP paket başlığı

Herşeyden önce UDP, TCP'ye göre çok daha az bağlantı yükü doğurur. Ancak artan UDP temelli band genişliği kullanımı, ağ çöküşünü önlemek için kullanılan tıkanıklık kontrolü içermemesi nedeniyle ciddi endişeler ortaya çıkarmaktadır. Ayrıca UDP datagramlarının ağda kaybolması, iletilememesi, bozulması durumları ortaya çıkabilir. Bu durumda, multimedya sistemlerinin paket kayıplarını nasıl tolere edeceği önem kazanmaktadır. Bu konudaki en basit strateji, gerçek zamanlı veri iletimi temelinde kayıp paketleri görmezden gelerek görsel ve işitsel içeriğin kalitesini kullanıcı deneyimini en az etkileyecek düzeyde düşürmek şeklinde ifade edilebilir. Ancak çok fazla sayıda paket kayıpları ciddi oranda video oynatma performans kaybına yol açacaktır. Bu durumda P2P uygulamalar için ek karmaşıklıklar getiren kayıp UDP paketlerini yeniden isteme durumları ortaya çıkacaktır.

3.2.3. Gerçek zamanli aktarim protokolü

IP ağlarında ölçeklenebilirlik, hata ve tıkanıklık kontrolü gibi ihtiyaçlar, gerçek zamanlı aktarım protokolünün doğmasına yol açmıştır [12]. RTP başlangıçta 1996 yılında RFC 1889 [75] ile önerilmiş ve 2003 yılında RFC 3550 [76] ile sadeleştirilmiştir. RTP, UDP aktarım protokülü üzerinden gerçek zamanlı multimedia veri akışını yönetmeyi amaçlamaktadır. Protokol tasarımcılarının temel amacı güvenilmez bir aktarım katmanı üzerinden sağlam ve gerçek zamanlı

multimedia taşınımı için bir mekanizma kurmak olmuştur. Bu noktada temel felsefe uygulama katmanında çerçeveleme ve uçtan-uca presibidir [12].

Uygulama katmanında çerçevelemenin savunduğu temel nokta, verinin nasıl taşınması gerektiğine yalnızca uygulama karar verebilir. Böylece eğer bir hata olursa uygulama ne tür bir yanıt verebileceğine, verinin aktarımı ve kabulü noktasında karar verebilir. Örneğin, bazı durumlarda kayıp veri göz ardı edilebilirken bazı durumlarda ise bu kayıp verinin bir kopyasının yeniden iletimi istenebilir. Dolayısıyla aktarım protokolü ile uygulamanın çok yakın bir etkileşim halinde olması ile bunları gerçekleştirebilmesi mümkündür. Uygulama katmanında çerçeveleme ve aktarım katmanı protokolü olarak UDP kullanarak gerektiğinde veri kayıpları dikkate alınmazken gerektiğinde yeniden iletim ve ileri hata düzeltme gibi esneklikler elde edilmektedir.

RTP tarafından benimsenen diğer bir felsefe uçtan-uca prensibidir. Bu prensibe göre ağ yolu boyunca varolan sistemler verinin doğru iletimi için sorumluluk almaları gerekmez. Bu sistemler güvenilir olmayabilir. Veri iletiminin uç noktaları, alt sistemlerin yardımı olmaksızın teslim sürecinin güvenilirliğini sağlamak için uygulama tasarımcısı adına önemli çalışmalar yapmasını gerektirir.

RTP, UDP aktarım protokolü üzerinden gerçek zamanlı multimedya veri aktarımını sağlamayı amaçlamaktadır. RTP paketleri, yanlış sırada alıcıya ulaşmış paketler için sıra numarası sağlar. Bununla birlikte zaman damgası alanı ile alıcı tarafından doğru sırayla alınan ses ve görüntü paketlerinin doğru pozisyonda oynatılmasına imkan tanır. Ayrıca SSRC ve CSRC gibi diğer alanlar ile bir multimedia iletişimde yer alan ses veya görüntü kaynağını belirleyebilir. Daha sonra ele alınacak RTCP (RTP Kontrol Protokolü) ile RTP birlikte çalışır ve bir multimedia oturumun hizmet kalitesini izlemek ve devam eden bir oturumda katılımcılar hakkında bilgi toplayabilme olanağı oluşur. RTP paket başlığı Şekil 3.3.'te görülmektedir [16].

RTP paket başlığında yer alan ilk iki bit versiyon için ayrılmıştır. Sonrasında yer alan bir bitlik (P) alanı dolgu anlamındadır ve eğer paket sonunda doldurma oktetleri

(sekizlileri) kullanılmış ise bu alan kuruludur. RTP yükünün alıcı tarafında yeniden elde edilmesi sürecinde bu alanın değerine göre veri alanının hesaplanmasında dikkate alınır. Özellikle şifreli RTP paketlerinin kullanıldığı durumlarda dolgu alanı kullanılır [12, 16, 17].

<table>
<tr><th>0-1</th><th>2</th><th>3</th><th>4-7</th><th>8</th><th>9-15</th><th>16-31</th></tr>
<tr><td>Ver.</td><td>P</td><td>X</td><td>CC</td><td>M</td><td>PT</td><td>16 bit sıra numarası</td></tr>
<tr><td colspan="7">32 bit zaman damgası</td></tr>
<tr><td colspan="7">32 bit senkronizasyon kaynak kimliği (SSRC)</td></tr>
<tr><td colspan="7">32 bit sağlayıcı kaynak kimliği (CSRC)</td></tr>
<tr><td colspan="7">Ekstra başlıklar (gerekiyorsa)</td></tr>
<tr><td colspan="7">RTP yükü (payload)
(…)</td></tr>
</table>

Şekil 3.3. RTP paket başlığı

Doldurma alanı sonrasında yer alan (X-eXtension) uzantı alanı, RTP standart paket başlığı ve RTP yükü arasında CSRC alanı sonrasında ilave başlık uzantılarının yer alıp almadığını ifade eder. Ekstra başlık alanında 16 bitlik ektra başlık kimliği ve ekstra başlık uzunluğu tanımlanır ve RTP yükü bu başlık alanlarından sonra gelmektedir. Özellikle multimedia kodeklerin bır kısmı bu ekstra RTP başlığına ihtiyaç duymaktadır [16, 17].

Uzantı (X) alanı sonrasında yer alan 4 bitlik sağlayıcı sayısı (CC) alanı, CSRC tanımlayıcı sayısını ifade etmektedir. Bir RTP oturumunda en fazla 16 farklı sağlayıcı kaynak kimliği (CSRC) desteklenir [12].

Sağlayıcı sayıcı (CC) alanından sonra 1 bitlik işaretleyici (M) alanı yer almaktadır. İşaretleyici, özellikle ses ve video kodeklerin sessizlik ve I-Frame (anahtar çerçeve) gibi özel paketlerin tanımlanmasında kurulur [17].

İşaretleyici bayrağından sonra 7 bitlik yük tipi (PT) alanı yer almaktadır. Yük tipi alanı iletilen multimedia içeriğin kullandığı ses veya görüntü kodek tiplerini ifade etmektedir. G711 u-law ses kodeği için bu alanın değeri 0 atanmışken, GSM için 3, G729 için 19, vb. tanımlanmıştır. Benzer şekilde görüntü kodekleri için de örneğin JPEG için 26, H261 için 31 vb. tanımlanmıştır. Bu alana istenirse yeni kodekler ilave edilebileceği gibi H264 gibi dinamik kodeklerin değişken saat sinyallerini ifade edebilecek şekilde özelleştirilebilir [21].

RTP yük tipi alanından sonra 16 bit uzunluğunda sıra numarası alanı yer almaktadır. Sıra numarası alanı, RTP paketlerinin iletim sırasını ifade etmektedir. Bu alandaki değerler kullanılarak ağda kaybolan/iletilemeyen paketlerin belirlenmesi ve farklı sırada gelen paketlerin doğru sıra ile önbelleklenmesi sağlanmaktadır. Gönderici ağa bıraktığı her RTP paketi için sıra numarasını bir artırır. En büyük sıra numarasına ulaşan paketten (65535) sonra iletilecek yeni RTP paketi tekrar sıfırdan başlatılarak iletilir. Sıra numaralarının sıfırdan başlatılması yerine şifreleme saldırılarını önlemek için rastlantısal olarak üretilmesi RFC 3550'de tavsiye edilmiştir [18, 76].

RTP yük tipi alanından sonra yer alan 32 bit uzunluğundaki alan senkronizasyon için kullanılan zaman damgasıdır. Bu alan medya saat oranına göre ölçümlenir. Zaman damgası, bir paket içindeki medya verisinin ilk byte'ın örneklemesini temsil eder ve verinin oynatımını planlamak için kullanılır. Zaman damgası kodek hızında artan ve maksimum değer aşıldığında sıfırlanan 32-bitlik işaretsiz bir tam sayıdır. Zaman damgasının başlangıç değerinin sıfırdan başlatılması yerine rastgele belirlenmesi önerilir. Sıra numarasında olduğu gibi, bu önlem şifreli RTP oturumları için saldırıları daha zor hale getirir [12, 18, 76].

Zaman damgası alanından sonra yer alan 32 bit uzunluğundaki alan senkronize kaynak kimliğidir (SSRC). Bu alan bir RTP oturumundaki katılımcıların kimliğini tanımlar. Yani aynı SSRC numarasına sahip RTP paketinin aynı kaynaktan çıktığı kabul edilir. SSRC değeri bir RTP oturumuna katılan kaynak tarafından rastgele seçilen 32 bitlik bir değerdir. SSRC değeri yerel olarak belirlendiği için farklı iki kaynak aynı SSRC değerini belirlemiş olabilir. Bu durumda bu kaynaklardan biri

kendisi için belirlediği bir SSRC numarasına ait bir paket aldığında çakışmayı belirlemiş olur (Gerçek zamanlı aktarım kontrol protokolü aracalığıyla). Bu alan her bir ses ve video kaynağı için ayrı bir numara ile kimliklendirilir [12].

Standart bir RTP paket başlığında yer alan 32 bit uzunluğundaki son alan sağlayıcı kaynak kimliğini (CSRC) ifade eder. Bir RTP oturumunda kaç farklı sağlayıcı kaynak kimliğinin kullanıldığı CC alanında tanımlanmıştır. Bu alanın değerinin sıfır olduğu durumlarda tek bir sağlayıcı tarafından veri akışı sağlandığı anlamına gelirken en fazla 15 farklı sağlayıcıdan veri akışı sağlanabilmektedir [18].

RTP fonksiyonları aşağıdaki işlevleri sunmaktadır [18, 76];

1. IP ağlarında iletilen datagram paketleri farklı yönlendiriciler üzerinden iletim sırası değişerek alıcaya ulaşabileceğinden dolayı veri paketlerinin alıcı tarafından doğru sıra ile sıralanabilmesi için sıra numarası sunar.
2. Alıcı tarafından veri kaynağının ve yükün tanımlanması için yük içeriğinin türü ve kaynağı tanımlanır.
3. Gerçek zamanlı multimedia içeriğin iletiminde alıcının iletilen verileri doğru sırada alması yeterli olmamaktadır. Sıralamanın yanında zamanlama ve senkronizasyon gereklidir. Örneğin görüntü ve ses verilerinin doğru zamanlama ile senkronize edilmesi gerekir. RTP zamanlama ve senkronizasyon işlevlerini yerine getirebilme işlevselliğine sahiptir.
4. RTP, aktarımların izlenebilmesine imkan sunmaktadır. İletilen verinin dağıtım verimliliğini ve kalitesini izleyebilme ve gönderen tarafa sağladığı geribildirim olanakları ile ölçeklenebilirliği ve servis kalitesini sağlamaktadır.
5. RTP, bir RTP oturumu üzerinden farklı kaynaklardan oluşan trafiği tek bir akış halinde birleştirerek, heterojen bir trafik sunabilmektedir.

RTP çerçeve yapısı, çok az ilave protokol ihtiyacı duyan ve pek çok senaryo için yeterli olacak şekilde tasarlanmıştır. Bu tasarım temel olarak multimedia içeriğin aktarımı için modellenmiştir. Bu modele göre RTP, gerekli tüm oturum yönetim işlevlerini sağlar. Ancak RTP fonksiyonları aktarım katmanı içinde farklı protokoller

ile etkileşime girer. Herhangi bir servis kalitesi (QoS) garantisi ve veri iletim güvenilirliği sağlamaz. Sadece, alıcıdan vericiye bilgi akışının kontrolünü izleyen yardımcı özellikler sağlar. RTP güvenilir teslimat ve servis kalitesi noktasında alt katman protokollerine ihtiyaç duymaktadır. RTP diğer protokollerin üstlendiği servis kalitesi fonksiyonlarından yoksundur fakat yük türü, kimlik, zaman damgaları ve sıralama bilgileri ile gerçek zamanlı verilerin taşınmasını sağlar.

Her RTP paketi bir sıra numarası ve zaman damgası içerir. Gelen paketlerin sıra numarası incelendiğinde her paketin doğru pozisyonda sıralanması sağlanır. Ayrıca ulaşılamayan sıra numaraları, kayıp paketlerin oranının belirlenmesinde kullanılır.

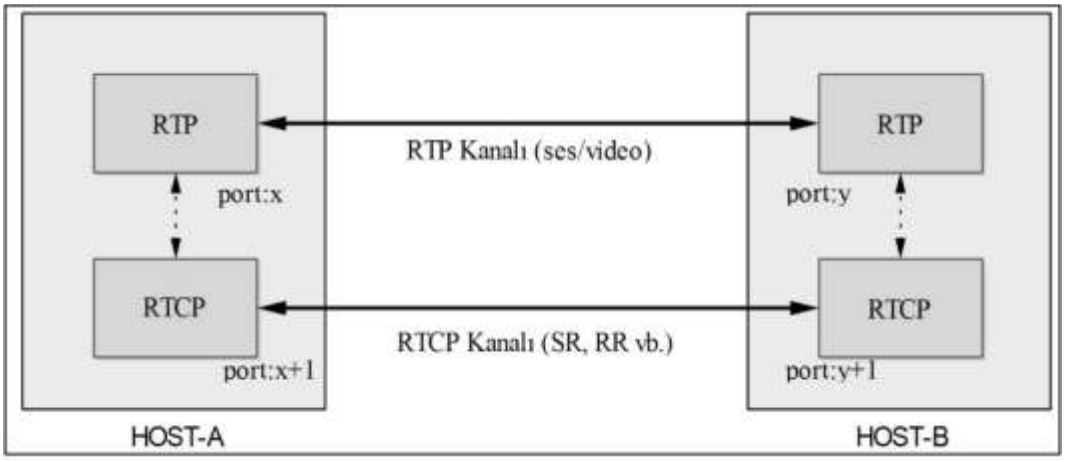
Benzer şekilde zaman damgası alanı kullanılır. İlk zaman damgası paket gönderildiği zaman kaynak tarafında oluşturulur ve ikinci zaman damgası paket alındığında alıcı tarafında üretilir. İki zaman damgası arasındaki fark, bu paketin ağdaki gecikme süresi hakkında bilgi verir. Bu bilgi, RTP paketleri arasındaki bekleme süresinin ve kodlama hızının belirlenmesinde kullanılır. Alıcı, tanımlanan gecikme (jitter) sürelerinden yararlanarak ağ trafiğinden kaynaklanan gecikmeden etkilenmeyen bir tamponlama oluşturabilir. Ayrıca zaman damgaları kaynak ve hedef senkronizasyonunu kolaylaştırdığı gibi paketlerin doğru zamanlama ile oynatılmasında gereklidir. Kaynak senkronizasyonu özellikle aynı anda birden çok veri tipi için (görüntü ve ses) kullanılır. RTP oturumları aynı anda birden çok içerik türünün taşınmasına izin verir. Bu içerikler alıcıya geldiğinizde, farklı içerik türlerinin senkronizasyon alanları doğru oynatma için kullanılır. RTP burada bahsedilen fonksiyonları yerine getirmek için gerçek zamanlı aktarım kontrol protokolüne (RTCP) ihtiyaç duyar.

3.2.4. Gerçek zamanlı aktarım kontrol protokolü

RTCP, RTP için tasarlanmış ilave bir protokoldür ve Şekil 3.4.'te görüldüğü gibi birlikte kurgulanmış bir kanal yapısına sahiptir. RTCP, ses ve video veri akışlarının senkronizayonunu, iletim istatistilerini, alım kalitesini ve alıcı bilgilerini periyodik olarak raporlar. RFC 3550'de belirtildiği gibi RTCP iki temel işleve sahiptir.

Bunlardan birincisi, multimedia içerik dağıtım kalitesi hakkında geri bildirim sağlamaktır. Bu işlev RTCP alıcı ve gönderici raporları aracılığı ile sağlanır. İkinci temel işlevi ise ses ve video senkronizayonunu sağlamak için RTP akışları için zaman damgalarını eşlemektir.

RTCP protokolünün temel işlevi olan medya dağıtım kalitesi hakkındaki geri bildirim fonksiyonunu icra etmek için RTCP-SR (gönderici raporu) ve RTCP-RR (alıcı raporu) olarak isimlendirilen paket yapıları tanımlanmıştır. RTCP-SR paketi, RTP akışlarını senkronize etmek, gönderilen paketlerin ve byte sayılarının genel bilgisini bildirmek ve iki nokta arasındaki gidiş-dönüş gecikme (RTT) süresinin hesaplanması için kullanılır. RTCP-RR paketi, alınan medya kalitesini periyodik olarak geri bildirir. Bu bilgi, bir ağ tıkanıklığı durumunda iletim hızını ayarlamak için kullanılır [75, 76].



Şekil 3.4. RTP / RTCP kanal yapısı

RTCP her RTP akışını tanımlamak için Takma İsim (CNAME) olarak adlandırılan bir tanımlayıcı kullanır. Bir RTP oturumuna yeni bir kaynak eklendiğinde oturumdaki paydaşlara bu kaynağın dağıtımı için RTCP-SDES (kaynak tanımlayıcı) paketi kullanılır. Benzer şekilde bir RTP oturumundan bir katılımcının ayrıldığının duyurulması için RTCP-BYE (Ayrılma) paketi tanımlanmıştır. Ayrıca bir SSRC kimliğinin çakışması durumunda da RTCP-BYE paketi kullanılır. Son olarak uygulamaya özgü fonksiyonları tanımlamak için kullanılabilecek RTCP-APP (Uygulama tanımlama paketi) tanımlanmıştır [75, 76].

Temel kural olarak bir oturumda RTCP için ayrılan bant genişliği toplam RTP bant genişliğinin yüzde 5'i ile sınırlandırılır [18, 19].

3.3. Genişletilebilir Mesajlaşma ve Durum Protokolü (XMPP)

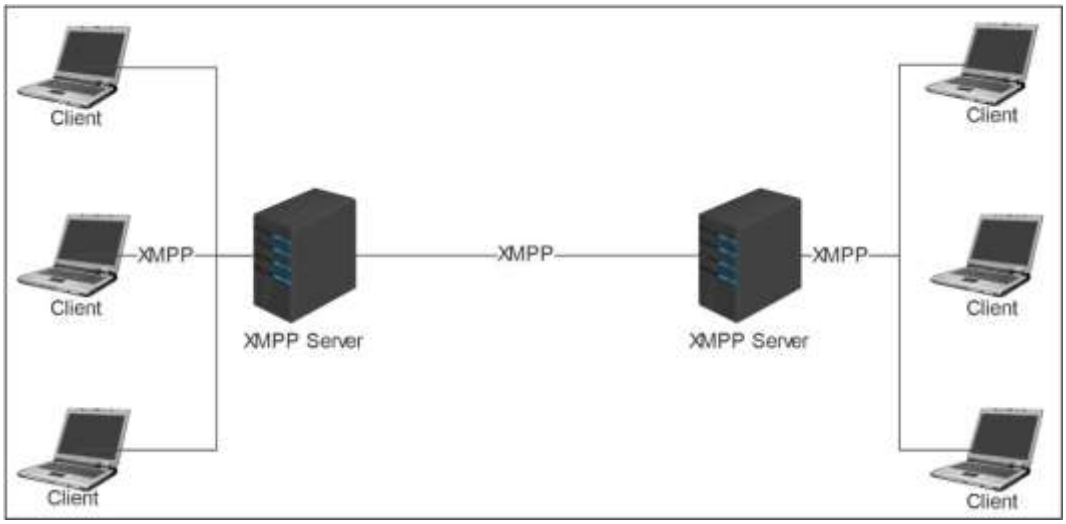
Anında mesajlaşma (IM), kullanıcılar arasında, farklı cihaz türleri ile kullanılabilen ve kolay iletişim sağlayan IP tabanlı bir uygulamadır. Günümüzde yaygın olarak bilinen formu, bilgisayar-bilgisayar arası anlık metinsel mesajlaşma yanında, ses ve video da eklenebilen şeklidir. Anında mesajlaşma servisleri, ICQ ile birlikte 1996 yılından beri internet kullanıcıları arasında hızla yaygınlaşmıştır [77]. Bu teknolojinin gösterdiği başarı nedeniyle birçok firma benzer ürünler (AOL, Yahoo, Live Messenger) üzerine çalışmalarını yönlendirmiştir. Bu uygulamaların her biri, uygulamayı geliştiren şirket tarafından işletilen, kendine özgü bir ağ protokolüne bağımlıdır. Bunun sonucu olarak sadece aynı protokolü kullanan cihazlar/yazılımlar birbirleri ile haberleşebilmektedir. Bu sorun, merkezi olmayan anında mesajlaşma ağı ve protokolünün geliştirilmesi fikrini ortaya çıkartmıştır. 1999 yılında jabber, günümüzde ise XMPP olarak anılan Genişletilebilir Mesajlaşma ve Durum Protokolü [43, 78] Internet Engineering Task Force (IETF) tarafından yayımlanan RFC 3920 ve RFC 3921 ile tanımlanarak gelişimini sürdürmeye devam etmiştir.

XMPP, genişletilebilir işaretleme dili (XML) tabanında, gerçek zamanlı sayılabilecek, mesajlaşma, durum yönetimi ve istek-yanıt hizmetleri için geliştirilmiş bir uygulama protokolüdür [41]. XMPP teknolojisi Jeremie Miller tarafından 1999 yılında anında mesajlaşma hizmeti için ortaya konulmuştur [79]. XMPP geliştiricileri başlangıçta bir anlık mesajlaşma sistemi geliştirmeye odaklanmışlardır. Ancak XML'in genişletilebilir yapısı sayesinde protokol yapısı hızla sadece anlık mesajlaşma özellikleri sağlamakla kalmayıp anlık mesajlaşma yanında güvenilir bir altyapıya ihtiyaç duyan uygulama geliştiriciler için pek çok farklı uygulama için altyapı protokolü haline gelmiştir. Sonuç olarak XMPP anında mesajlaşma özellikleri yanında, bildirimler, beyaz tahta uygulamaları, multimedia ortamları, sosyal ağ, online oyun vb. geniş bir yelpazede kullanılır olmuştur. Bu süreç geliştiriciler tarafından XMPP çekirdek protokollerinin yanında çok sayıda uzantı protokollerinin

geliştirilmesine yol açmıştır. Bu uzantılar, açık standartlar altında, XMPP uzantı protokolleri (XEP) olarak yayınlanmaktadır.

XMPP, gerek iki sunucu, gerekse iki istemci arasında, P2P haberleşme olanağı sağlayabilen ve birbirlerine bağlanabilen servisler arasında otonom ağlar oluşturabilmektedir [3]. XMPP protokolü ile özel sohbet ağlarındaki kullanıcılar, birbirleri ile servislerinin izin verdiği ölçüde iletişim kurabilme olanağına sahiptirler. Bu durum, kullanıcılar için anında mesajlaşma pazarındaki XMPP destekli herhangi bir ürün ile kendi listesindeki kullanıcılarla iletişim kurabilmelerini sağlamıştır.

XMPP, istemciler arasında veri iletimi için, XML paket formatını kullanan gerçek zamanlı iletişim protokolüdür [43]. Temelde XMPP, gerçek zamanlı olarak sayılabilecek şekilde istemciler arasında, XML tabanlı veri iletimi için uygun bir iletişim ortamı sağlamaktadır. XMPP, web sunucuları ve e-posta hizmetlerinde olduğu gibi, merkezi olmayan bir istemci-sunucu mimarisi sunar. Şekil 3.5.'te görüldüğü gibi, merkezi olmayan istemci-sunucu mimarisi ile isteyen herkes kendi XMPP sunucusunu güvenlik ve ölçeklenebilirlik noktasında yönetebilirken, istemciler yalnızca kullanıcı deneyimlerine odaklanarak dağıtık olarak geliştirilebilirler.



Şekil 3.5. XMPP istemci-sunucu mimarisi

Popüler tüm internet teknolojileri, bağlantı kurulumu, birlikte çalışabilme ve iletişim için uygun yollar sunan çeşitli yeteneklere sahip mimarilere sahiptir. Örneğin www için geliştirilen Apache gibi web sunucu hizmet programları ve bu hizmetleri kullanan milyonlarca Firefox vb. gibi tarayıcı kullanan istemciler vardır. Yada pek çok e-posta hizmet sunucu programı ve bunlarla birlikte çalışabilme, bağlatı kurulumu ve iletişim gibi mimarisel yeteneklere sahip e-posta istemcileri mevcuttur. Tüm bunlara benzer şekilde anlık mesajlaşma ve durum yönetimi temelinde hizmet veren ejabberd [81] ve Openfire [80] vb. gibi sunucu hizmet yazılımları ve Jitsi [82], Adium [83], Pidgin [84] gibi istemci yazılımları XMPP protokolü üzerinde popüler internet teknolojilerinin sahip oldukları temel mimariye uyumlu bir şekilde gelişmektedir [79]. Ancak bu noktada web yada e-posta mimarileri ile XMPP mimarisi arasında temel farklılıklar da ortaya çıkmaktadır. XMPP farklı fedaratif yapılar üzerinden sunucular arası haberleşme sağlarken web mimarisi için bu durumdan standart olarak bahsedilemez. E-posta hizmetlerinde ise bir e-posta sunucusu farklı bir federasyonda tanımlı bir e-posta hesabına e-posta iletimi için birden fazla sunucu üzerinden geçmesi gerekebilirken XMPP için doğrudan federasyonlar arası iletişim kurulumu mümkündür.

3.3.1. XMPP adresleme

XMPP ağı üzerindeki her varlık JID (jabber kimliği) olarak isimlendirilen tekil bir adrese sahip olmalıdır. Bir JID kimliği, user@domain/resource (kullanıcı@alanadı.com/kaynak) yapısında, yerel ad, alan adı ve kaynak bildirimlerine sahiptir [3, 79, 85].

Kullanıcı alanı, e-posta hesabında olduğu gibi ilgili alan adı içerisindeki tekil bir kullanıcıyı kimliklendirmek için kullanılır. Bu ad, genellikle pek çok serviste e-posta hesabı ile aynı kullanılır. Jabber kimliğini oluşturan bileşenlerden kullanıcı adı ve alan adı kısımları büyük-küçük harfe duyarsız olmalarına karşın kaynak alanı büyük-küçük harfe duyarlıdır.

Her XMPP kimliğinde bulunması gereken alan adı alanı, tüm XMPP ağı içerisinde o varlığa ait alt ağı tanımlamak için kullanılır. Kullanıcı adı ve alan adı alanının bileşkesiyle (kullanıcı@alanadı.com) elde edilen jid yapısı, çıplak (bare) jid olarak isimlendirilir. Benzer jid yapısı çok kullanıcılı mesaj odalarının isimlendirilmesi için de kullanılır. Ancak bir jabber kimliği sadece alan adı ile ifade edilebilir. Bu durumda ifade edilen jabber kimliği o federasyonu adreslemektedir.

Kaynak bildirimi ise, bir kullanıcının farklı istemciler üzerinden bağlantı kurabildiği durumlarda hangi istemcinin hangi kaynağı kullandığını tanımlamak için tanımlanmaktadır. Bu tanımlama, bağlantı kurulan istemci yazılımını ifade etmek için kullanılabileceği gibi kullanıcınının bu oturumun oluşturulduğu aygıta özel mesajlaşmalar ve sorgulamalar yapmak için de özelleştirilebilir. Bir kullanıcı aynı anda farklı kaynaklar üzerinden benzer oturumlar açabildiği gibi istenirse bu durum engellenebilmektedir. Bir jabber kimliğini oluşturan bu üç bileşenin yer aldığı adres yapısı tam (full) jid olarak isimlendirilir. Temel olarak bare jid XMPP federasyonundaki o kullanıcıyı ifade ederken, full jid anlık oturumu ifade etmektedir.

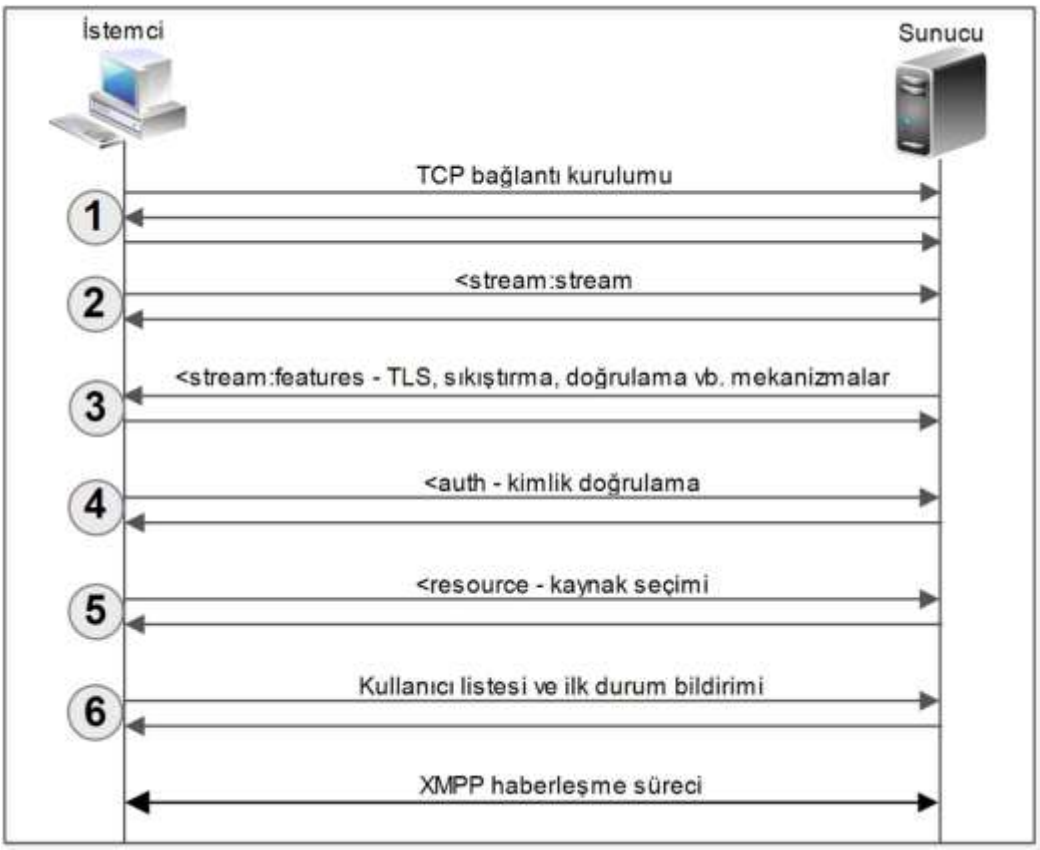
3.3.2. XMPP sinyalleşme

İstemci-sunucu arasındaki uzun ömürlü bir TCP bağlantısı açıldıktan sonra veri akışı, <stream:stream> çifti ile başlar ve sonra XML paket formatında devam ederek </stream:stream> çifti ile sona erer. Bu yapı webde olduğu gibi <html>…</html> yapısı olarak düşünülebilir. Ancak burada kurulan TCP bağlantısı klasik web tabanlı istemci-sunucu modelinde olduğundan farklı olarak istek cevapları döndükten sonra kapatılmaz ve yaşatılmaya devam eder. Bu durum her yeni istek için yeniden TCP bağlantısı kurulumu oluşturmadan neredeyse gerçek zamanlı bir iletişim altyapısı üzerinde sistemin koşturulmasına olanak tanır.

En genel anlamda bir XMPP sinyalleşme süreci yapısal olarak aşağıdaki gibidir;

1. Sunucu ile uzun ömürlü TCP bağlantının kurulması,
2. XML akışlarının başlatılması,

3. Çeşitli XML şemalarının paylaşımı,
4. Sunucu üzerinde istemci kimliğinin doğrulanması,
5. Kaynak seçiminin sağlanması,
6. Anında mesajlaşma süreci öncesi kullanıcı listelerinin alınması ve ilk durumun sunucuya bildirimi şeklinde gerçekleşir.



Şekil 3.6. XMPP sinyalleşme adımları

İlk 6 adımdan sonra gerçek anlamda XMPP haberleşmesi sürdürülmektedir. Bu aşamaya kadar gerçekleşen süreç; bir XMPP sunucu üzerinde bir istemcinin sinyalleşme adımları Şekil 3.6.'da sunulmuştur.

3.3.3. XMPP paket yapıları

İstemci-Sunucu arasında XMPP sinyalleşme başlığı altında anlatılan adımlar gerçekleştirildikten sonra bu bağlantı üzerinden XMPP'de tanımlı XML paket yapıları üzerinden iletişim akışı sürdürülür. Bu XML paket yapıları XML Stanza

olarak isimlendirilir. Bu terim, XMPP'de diğer ağ protokollerinde kullanılan paket yapılarında olduğu gibi istemci-sunucu arasındaki iletişimin temel birimleri olarak ifade edilebilir. İletişim paket yapılarının XML stanza olarak alt birimlere ayrılmasında bu paket tiplerinin hem yapısal hem de işlevsel olarak farklı öznitelikler ve davranışlar sergilemesinin yanında, kurulan iletişim modeli açısından farklı isim uzayları gibi nesnel ayrıştırmalara olanak sağlamasıdır. Bu paket yapıları temel olarak [3, 43, 79];

1. XMPP'de XML paket formatı olarak mesaj (<message>), durum (<presence>) ve bilgi/sorgu (<iq>) olmak üzere üç temel tanımlama mevcuttur. Bu paket yapılarının her biri sunucu tarafından farklı yönlendirildiği gibi istemciler tarafından da farklı işlenir [43].
2. Tanımlanan bu 3 farklı XML stanza için kendi sorgu yapılarına özgü, tanımlanması gereken öznitelik:değer çiftlerinden oluşan zorunlu bağımlılıkları vardır. Dolayısıyla bu öznitelik:değer tanımlamaları alıcılar tarafından paketin nasıl işleneceğini belirler.
3. Bu temel tanımlamalar içerisinde taşınan XML yükleri yada alt elemanları her bir paket yapısına özgü ve çoğunlukla otomatik olarak oluşturulan XML elemanlarıdır.

XMPP'de tanımlanan 3 farklı paket yapısı için ortak olan 5 farklı öznitelik mevcuttur. Bu öznitelikler tüm paket yapılarında aynı anlamı ifade etmektedir ve tüm XMPP federasyonunda değiştirilmeden sürdürülegelmektedir [43].

Kime (to) özniteliği: İlgili paketinin alıcısını tanımlayan JID'dir. Kime özniteliğinin tanımlanmadığı paketlerde sunucu bu paketin hedefinin kendisi olduğuna karar verir ve işler. İstemci, sunucuya ilettiği bir paket için kime özniteliğini tanımlamak zorunda olmasa da sunucu kime özniteliğini tüm XMPP paketleri için zorunlu olarak tanımlamalıdır. Benzer şekilde kime özniteliğinin değeri çıplak (bare) JID olarak tanımlanmış ise bu paketi sunucu işleyecektir. Eğer kime değeri tam (full) JID olarak tanımlanırsa bu paket doğrudan o kullanıcıya yönlendirilecektir.

Kimden (from) özniteliği: İlgili paketin kim tarafından gönderildiğini tanımlayan JID'dir. Bir istemci kimden özniteliği olmayan bir paket alırsa bu paketin istemcinin bağlı olduğu sunucudan geldiğini kabul eder. Ancak sunucuya gelen bir XMPP paketinde kimden alanı geçersiz yada tanımlanmamışsa bu durumda sunucu istemciye "geçersiz kimden" (<invalid-from/>) cevabı dönecektir. Benzer şekilde henüz kimlik doğrulaması yapılmamış istemcilerin ilettikleri paketler için de sunucu tarafından kimlik doğrulanmadı (<not-authorized/>) hatası ile cevap verilecektir. Oluşan bu iki hata durumunda da istemci-sunucu arasındaki mevcut TCP bağlantısı sonlandırılacaktır. Bu kontroller, yetkisiz erişimleri engellemek için alınmış ilave güvenlik önlemleridir.

```
İ:
<message from='kullanıcı@ornek.com' to='kullanıcı2@ornek.com'>
   <body>Selam</body>
</message>

S:
<stream:error>
   <invalid-from xmlns='urn:ietf:params:xml:ns:xmpp-streams'/>
   </stream:error>
</stream:stream>
```

Şekil 3.7. Hata paket yapısı

Şekil 3.7.'de görülen örnek hata paketinde "to" da belirtilen bir kullancıya iletilmek üzere sunucuya bir mesaj paketi iletmiştir. Ancak ilgili kullanıcı XMPP ağında olmadığı için paketi üreten kullanıcıya geçersiz-kimden hatası dönülerek oturumu sonlandırılmıştır.

Kimlik (id) özniteliği: XMPP paketlerinde oluşan cevap paketlerinin eşleştirilmesinde/kimliklendirilmesinde yardımcı olmak için bilgi/sorgu (iq) paketi hariç seçimli olarak kullanılır. Bir XMPP paketi için cevap paketi oluşturulacak ise herhangi bir karışıklığı önlemek için aynı kimlik (id) değerine sahip bir cevap paketi ile cevaplanmalıdır. Özellikle art arda istenen paketler için cevap paketleri farklı sıralarda ulaşabilir. Bu durumda hangi cevabın hangi istek için geldiğinin belirlenmesi açısından kimlik (id) özniteliği önem arzetmektedir.

Tip (type) özniteliği: Mesaj (<message>), durum (<presence>) ve bilgi/sorgu (<iq>) paketlerinin içeriği yada amacı hakkında bilgiler ifade etmektedir. Üç tip paket yapısı için ortak olan tip özniteliği hata (error) değeridir. Tip (type) özniteliğinin alabileceği değerler bu üç paket yapısı için ayrı ayrı özelleştirilmiş anlamlar ifade etmektedir ve ilgili paket yapılarının detaylandırıldığı bölümlerde ele alınacaktır.

XML dil (xml:lang) özniteliği: W3C tarafından tanımlanan [86] ve RFC 2277 [87] ve RFC 3066'da [88] açıklandığı gibi görüntülenebilir bir XML verisi içeriyorsa bu verinin uluslararasılaştırılması için tanımlanır. Bu değer XML karakterlerinin varsayılan dilini ifade eder. XML dil tanımı, alt etiketler tarafından değiştirilebilir. Genellikle XMPP oturumlarında XML akışlarının başlatıldığı <stream:stream …> etiketi altında bir kez tanımlanan öznitelik, diğer XML alt bileşenlerince miras alınır. Bu özniteliğin hiç tanımlanmadığı durumlarda sunucunun varsayılan dil tanımı kullanılır.

Mesaj (<message>) paketi temelde, anlık mesajlaşma sistemleri için istemciler arasında mesaj iletiminde kullanılabildiği gibi, özelleştirilmiş her türlü bilgiyi de istemciler arasında aktarmak için kullanılabilir. XMPP mesaj paket mimarisi, e-posta sistemlerinde olduğu gibi gönder ve unut prensibine dayanır. Yani, iletilen bir mesajın e-posta sistemlerinde olduğu gibi alıcıya ulaşıp ulaşmadığı konusunda bir geri bildirimde bulunulmaz. Ancak mesaj alındı bildiriminin istenildiği uygulamalar için XEP-0184 protokol eklentisi geliştirilmiştir [90, 91]. Örnek bir mesaj paket yapısı Şekil 3.8.'de görülmektedir.

```
<message
    from='kullanıcı1@ornek.com/kaynak'
    id='svs123456'
    to='kullanıcı2@ornek.com'
    type='chat'
    xml:lang='en'>
  <body>Selam</body>
</message>
```

Şekil 3.8. Mesaj paket yapısı

Tanımlanan mesaj paketi içerisindeki özniteliklerden tip (type) alanı bu paket yapısına özgü değerler içermektedir. Mesaj paket tipi (type) olarak normal, chat, groupchat, error ve headline tanımlanmıştır.

Tip değerinin "normal" tanımlandığı mesaj paketleri, birebir chat görüşmesi yada grup chat dışındaki mesaj iletileri için tanımlanmıştır. Mesaj paketinin tip alanı için varsayılan değeri "normal" olarak tanımlanmıştır. Mesajların arşivlenmediği bu tip paketler genellikle çevrimdışı bir kullanıcıya iletilmek üzere gönderilen mesajı ifade etmek için kullanılır.

Birebir sohbet iletilerinin gönderilmesi için kullanılan "chat" mesaj tipi, çevrimiçi bir kullanıcıya doğrudan gönderilen mesajı ifade ederken çevrimdışı kullanıcılara iletilen mesajlar mesaj arşivinde saklanarak ilgili kullanıcının çevrimiçi olduğu durumlarda yeniden iletilebilir. Anlık mesajlaşma (IM) uygulamalarının en sık kullandığı mesaj tipidir.

Çok kullanıcılı bir sohbet ortamının bir sohbet odası üzerinden grup iletimi (multicast), altyapısı ile iletilerin yönlendirilmesi için geliştirilmiş "groupchat" mesaj tipi, bir chat odasına iletilmek üzere gönderilen mesaj tipi olarak tanımlanmıştır. Bir kullanıcıya özel mesaj iletmek için "chat" tipi kullanılırken bir sohbet odasındaki tüm alıcılara mesaj iletmek için "groupchat" tipi kullanılır.

Başlık "headline" mesajları ise genellikle web sayfası bildirimleri olarak kullanılan RSS'ler gibi bilgi amaçlı iletilen, cevap beklenilmeyen yada otomatik üretilen, bildirim, uyarı, duyuru vb. mesajlar için kullanılmaktadır. Bu tip mesajlarda alıcı (to) için çıplak (bare) JID'in kullanıldığı durumlarda sunucunun bu kullanıcıya ait tüm kaynaklara bu mesajı iletmesi beklenirken, bu alıcının kaynaklarından en az birine mesajın iletimi zorunludur. Alıcının tam (full) JID olarak tanımlandığı bu tip mesaj paketlerinde sunucu bu kaynağa ilgili mesajı iletmek zorundadır. Aksi durumda sunucu mesajı ya reddetmiştir yada hata döndürecektir.

Son olarak “error” mesaj tipi, hata oluşan bir iletiye cevap durumda kullanılır. Örneğin
iletilen mesaj sonrasında mesajı ileten kullanıcının XMPP ağında bulunamaması durumunda sunucu iletiyi gönderen istemciye “error” mesaj tipi ile cevap verecek ve oturumunu sonlandıracaktır.

Mesaj paketi içerisinde isteğe bağlı olarak tanımlanan <body></body> elementi, okunabilir iletilerin tanımlanması için kullanılır. <body> elementinin xml:lang dışında özniteliği yoktur. Bir mesaj paketi içerisinde birden fazla <body> elementi xml:lang özniteliği ile bir içeriğin farklı dillerdeki değerini ifade etmek için kullanılabilir. Benzer şekilde mesaj paketi içerisinde isteğe bağlı olarak tanımlanabilen bir başka alt element <subject></ subject> elementidir. Okunabilir bir mesaj içeriğinin başlığı olarak tanımlanan <subject> alt elementi, <body> elementinde olduğu gibi xml:lang özniteliği ile birden fazla tanımlanarak ilgili başlığın farklı dillerdeki versiyonlarını ifade etmek için kullanılabilir.

İki kullanıcı arasındaki her bir mesajlaşma oturumunu ayrı ayrı kimliklendirmek için <thread>tekil değer</thread> alt elementi kullanılabilir. İş parçacığı olarak isimlendirebileceğimiz bu alt element, tekil bir değerle, diğer konuşma oturumlarından ayrıştırılabilir. İş parçacığı elementi ebeveyn “parent” özniteliği ile bir konuşma oturumunun alt bileşeni olarak da kimliklendirilebilir. Ebeveyn özniteliğinin değeri, ilgili konuşma oturumunun iş parçacağı değeri ile ilişkilendirilmelidir. Bir mesaj paketi içerisinde bir tane iş parçacığı alt elementi yer alabilir.

Durum (presence) paketi, bir kullanıcının XMPP ağında varlığını kontrol eder ve raporlar. Kısaca XMPP ağındaki bir kullanıcının çevrimiçi, meşgul, çevrimdışı gibi durumlarını yönetmek için kullanılan paket yapısıdır. Böylece kullanıcıların birbirlerinin erişilebilirliğini kontrol edebilmeleri sağlanır. Bir istemcinin durum paketleri, sadece o istemcinin onaylamış olduğu iletişim listesindeki diğer kullanıcılar arasında dağıtılmaktadır. Ayrıca durum paketi bir kullanıcının diğer kullanıcı ve gruplara aboneliklerini bildirmek ve sonlandırmak için kullanılır.

Geleneksel anlık mesajlaşma sistemlerinde durum paketlerinin oluşturduğu trafik mevcut trafiğin büyük çoğunluğunu oluşturmaktadır. Genellikle iki kullanıcı arasında ilitişimi etkinleştirmek için iki tarafın birbirlerinin durumundan haberdar olmaları gerekir. Başkaca istemci-sunucu haberleşmesi olan e-posta vb. sistemlerde iletinin gönderildiği kişinin durumunun bilinmesi anlık olarak önem arz etmemektedir. E-posta sistemlerinin aksine anlık mesajlaşma sistemlerinde mesajın iletildiği kullanıcının anlık durumu (çevrimiçi, çevrimdışı, meşgul, dışarda, vb.) mesajı ileten kullanıcı tarafından bilinmelidir. Örnek bir durum paket yapısı Şekil 3.9.'da görülmektedir. Şekilde görüldüğü gibi durum paketinde tip (type) alanı tanımlanmamış ise bu durum paketini üreten kullanıcının iletişim için uygun durumda olduğu anlamına gelir. Bu durumdan tip (type) alanının varsayılan değerinin mevcut (avaliable) olduğu anlamı çıkarılmamalıdır. Tip (type) alanı için herhangi bir varsayılan değer tanımlanmamıştır.

```
<presence xmlns='jabber:client'>
   <show>away</show>
   <priority>0</priority>
   <status>Dışarıda</status>
</presence>
```

Şekil 3.9. Durum paket yapısı

Durum paketlerinin alabileceği tip (type) değerlerinin ifade ettiği anlamlar özetlenecek olursa;

1. Hata (error): Gönderilen bir durum paketinin işlenmesiyle ilgili bir hata oluştu ise type=error ve <error/> alt elementi ile durum bildirimine cevap verilir.
2. Araştırma (probe): Sunucu tarafından bir kullanıcının o anki durumunu sorgulamak için oluşturulan durum kontrol paketleridir.
3. Mevcut-değil (unavailable): Bir kullanıcının sunucu ile oturumunu sonlandırmadan önce iletişim için uygun olmadığını bildiren durum paketidir.

4. Abonelik isteği (subscribe): İsteği üreten kullanıcının isteğin iletildiği kullanıcı için iletişim listesine (roster) abone olmak için kullanılan durum paketidir.
5. Abonelik onayı (subscribed): Abonelik isteğinin kabul edildiğinin, abonelik isteğinde bulunan kullanıcıya bildirildiği durum paketidir.
6. Abonelikten ayrılma isteği (unsubscribe): İsteği üreten kişinin, alıcının aboneliğinden ayrıldığının bildiriminin yapıldığı durum paketidir.
7. Abonelik iptali (unsubscribed): Abonelik isteğinin reddedildiği yada varolan aboneliğin iptal edildiğini ifade eden durum paketidir.

Eğer bir durum paketinin tip (type) alanı yukarıdaki değerlerin dışında bir değer ile kurulursa alıcı yada bir ara yönlendirici tarafından <bad-request/> hata paketi ile cevap verilir. Durum paketleri <show/>, <priority/> ve <status/> alt elementleri içerebilir. Bu alt elementler bir varlığın durumu hakkında daha detaylı bilgi vermek için tanımlanmıştır.

İsteğe bağlı olarak tanımlanabilen <show/> alt elementi mevcut olan bir varlığın alt durumunu tanımlamak için kullanılır. Bir durum paketi içerisinde isteğe bağlı olarak sadece bir tane <show/> alt elementi tanımlanabilir ve bu alt elementin değeri “away”, ”chat”, “dnd” ve “xa” olabilir. Bu tanımlardan “away” tanımı varlığın kısa bir süre için dışarda olduğunu, “chat” tanımı varlığın sohbet için uygun olduğunu, “dnd” tanımı varlığın meşgul olduğunu ve son olarak da “xa” tanımı varlığın uzun bir süre dışarda olduğunu ifade etmektedir. Eğer bir durum paketi <show/> alt elementi içermiyorsa bu durumda varlığın çevrimiçi veya mevcut durumda olduğu kabul edilir.

Benzer şekilde <show/> alt elementi gibi isteğe bağlı olarak tanımlanabilen <status/> alt elementi varlığın <show/> alt elementinde görülen durumsal tanımının açıklayıcı metni olarak kullanılabilir. Örneğin <show>xa</show> şeklinde dışarda olarak durumunu ayarlayan bir kullanıcı, aynı zamanda <status>Öğle yemeğinde</status> şeklinde durumu ile ilgili açıklayıcı bir detay tanımlayabilir. Bu açıklama alanı xml:lang dil tanımlaması ile çoğullanarak farklı dillerde açıklama girilmesi

sağlanabilir. Benzer şekilde tip (type) alanında ifade edilen abonelik isteği (subscribe) ve mevcut-değil (unavailable) durum paketlerinde tanımlama yapmak için kullanılabilir.

Son olarak öncelik <priority> alt elementi, bir kullanıcıya ait kaynakların öncelik seviyesini ayarlamak için kullanılan ve -128 ile +127 arasında bir tamsayı değer alabilen tanımlayıcıdır. Bir durum paketinde isteğe bağlı olarak en fazla bir tane öncelik alanı tanımlanabilir. Öncelik alanının tanımlanmadığı durum paketlerinin öncelik değeri sıfır olarak kabul edilir. Öncelik değerinin aldığı değerlere göre bir mesaj paketinin bir kullanıcının birden çok kaynağı arasında hangisine dağıtılacağına karar verilir. Bir kullanıcının çıplak (bare) JID'ine bir mesaj paketi iletildiğinde bu kullanıcının çevrimiçi negatif olmayan herhangi bir kaynağı yok ise bu mesaj paketi daha sonra dağıtılmak üzere çevrimdışı mesaj olarak saklanır ve <service-unavailable/> paketi ile mesajı ileten kullanıcıya cevap dönülür. Eğer bu kullanıcının bir tane negatif olmayan önceliğe sahip kaynağı var ise bu mesaj paketi bu kaynağa dağıtılacaktır. Eğer bu kullanıcının iki tane negatif önceliğe sahip olmayan kaynağı varsa bu mesaj paketi özel yapılandırmalarla belirlenen daha yüksek önceliğe sahip kaynağa yada tüm kaynaklara iletilecektir.

XMPP ağında oturum açan bir kullanıcının uygun önkoşul sinyalleşmelerini gerçekleştirdikten ve iletişim listesinde yer alan kullanıcıları (roster) sunucudan istedikten sonra kendi durumunun iletişim kurulmaya hazır olduğunu sunucuya bildirmek için ilk durum paketi (initial presence) olarak isimlendirilen durum paketini bildirmek zorundadır.

```
<presence xmlns='jabber:client'>
   <priority>1</priority>
</presence>
```

Şekil 3.10. İlk durum paketi

Şekil 3.10.'da görüldüğü gibi ilk durum paketi, öncelik (priority) alt elementine benzer, gösterge (show) ve durum (status) alt elementleri de içerebilir. Ancak bu üç alt element de isteğe bağlı tanımlanabilmektedir.

İlk durum paketini alan sunucu bu kullanıcıya abone olan diğer kullanıcılara paketin kimden (from) kısmına kullanıcının tam (full) JID'ini yerleştirerek dağıtımda bulunur. İlk durum bildiriminde bulunan kullanıcının iletişim listesinde bulunan kullanıcıların abonelik tanımları "both" yada "from" olduğunda durum bildirimleri dağıtılacakken, abonelik tanımları "none" yada "to" olan kullanıcılara durum bildirimleri dağıtılmayacaktır. Şekil 3.11.'de kullanıcı1@ornek.com/kaynak1'den gelen ilk durum bildiri, bu kullanıcının iletişim listesinde (roster) yer alan ve dağıtım durumları "both" yada "from" olarak işaretlenmiş örnek kullanıcılara dağıtımları görülmektedir.

```
S:
<presence from='kullanıcı@ornek.com/kaynak1'
          to='kullanıcı2@ornek.com'>
     <priority>1</priority>
</presence>

S:
<presence from='kullanıcı@ornek.com/kaynak1'
          to='kullanıcı3@ornek.com'>
     <priority>1</priority>
</presence>


S:
<presence from='kullanıcı@ornek.com/kaynak1'
          to='kullanıcı4@ornek.com'>
     <priority>1</priority>
</presence>

S:
vb..
```

Şekil 3.11. İlk durum paketinin sunucudan dağıtımı

İlk durum bildiriminin dağıtıldığı kullanıcılar bu bildirimin kimden geldiğine bakarak ilk durum bildiriminin gereğini yapmaları beklenir. Bu bildirimin geldiği kullanıcıların iletişim listesinde bildirimde bulunan kullanıcının çıplak (bare) JID'i

tanımlı ise kullanıcı listesi arayüzüne kullanıcının durumu yansıtılır. İlgili kullanıcı, diğer kullanıcıların iletişim listesinde olmadan da mesaj ve bilgi/istek paketleri üzerinden haberleşebileceği için böyle bir iletişim ekranı aktif ise yine bildirimde bulunan kullanıcının durumu ilgili arayüzlere yansıtılır. Bu durumların hiçbirinin olmadığı durumlarda ilk durum bildirim paketi dikkate alınmaz.

İlk durum bildiriminde (initial presence) bulunan kullanıcı aynı yaklaşımla iletişim listesinde tanımlı kullanıcılara yayınlanmak üzere alt durum bildirimlerinde bulunabilir. Burada süreç ilk durum bildiriminde izlenen adımlardaki şekilde sürdürülür. Şekil 3.12.'de görüldüğü gibi ilk durum bildiriminden sonra aynı kullanıcı durumunun dışarda (away) olarak ayarlanmasını isteyebilir. Bu durumda sunucu bu bildirimi oturum yönetimine işleyeceği gibi ilk durum bildirimine uygun olarak kullanıcının iletişim listesine dağıtacaktır.

```
İ:
<presence>
   <show>away</show>
</presence>

S:
<presence from='kullanıcı@ornek.com/kaynak1'
          to='kullanıcı2@ornek.com'>
     <show>away</show>
</presence>

S:
<presence from='kullanıcı@ornek.com/kaynak1'
          to='kullanıcı3@ornek.com'>
     <show>away</show>
</presence>

S:
vb..
```

Şekil 3.12. Durum bildirim yapısı

XMPP ağındaki bir kullanıcı, sunucu ile oturumunu sonlandırmadan önce kullanıcının sunucuya mevcut-değil (unavailable) durum bildiriminde bulunması beklenir. Mevcut-değil durum bildiriminde bulanan kullanıcı isterse <status></status> alt elementi arasında oturumu sonlandırma nedenini kendisine abone olan kullanıcılara dağıtabilir.

```
İ:
<presence type='unavailable'>
   <status>Durum mesajı</status>
</presence>

S:
<presence from='kullanıcı@ornek.com/kaynak1'
          to='kullanıcı2@ornek.com'
          type='unavailable'>
     <status>Durum mesajı</status>
</presence>

S:
<presence from='kullanıcı@ornek.com/kaynak1'
          to='kullanıcı3@ornek.com'
          type='unavailable'>
     <status>Durum mesajı</status>
</presence>

S:
vb..
```

Şekil 3.13. Mevcut-değil (unavailable) durum bildirim yapısı

Şekil 3.13.'te bir mevcut-değil (unavailable) durum bildiriminin istemciden üretimi ve bu durum bildiriminin bu kullanıcıya abone olan kullanıcılara sunucu tarafından dağıtımı ifade edilmiştir.

Bilgi/Sorgu (<iq>) paketi, istemci-sunucu arasında iletilmek istenen bir verinin, mesaj ya da durum paket yapılarına uymadığı durumlarda bilgi/sorgu paketi olarak kullanılması için tasarlanmıştır. Bilgi/sorgu paketleri genel olarak istek/cevap paketi olarak konumlandırılmıştır. Bu paket yapısı özelleştirilerek, protokolün genişletilebilirliği sağlanır. Bu paket yapısı özellikle HTTP mimarisinde kullanılan GET, POST, PUT vb. yöntemlerinde olduğu gibi istek-yanıt etkileşimini sağlamaya yönelik iş akışları için bir yapı sağlamaktadır.

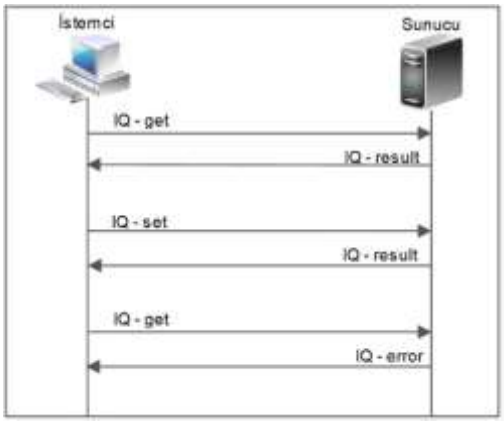
Şekil 3.14.'te görülen örnek bilgi/istek peketinde "kullanıcı@ornek.com/kaynak1" kaynağı kullanıcının iletişim listesini (roster) sunucudan talep etmektedir. Id olarak ifade edilen değer bu pakete üretilecek cevabın takibi için kullanılmaktadır. Ardıl bir şekilde istenebilecek bilgi/sorgu paketlerinin farklı sıralarda gelme yada bir kısmının gelememesi durumunda hangi cevabın hangi isteğe karşılık geldiğinin belirlenmesinde "id" değeri önem arz etmektedir.

```
İstemci isteği:
<iq from='kullanıcı@ornek.com/kaynak1'
    id='d3079'
    type='get'
    xmlns='jabber:client'>
 <query xmlns='jabber:iq:roster'/>
</iq>

Sunucu cevabı:
<iq to='kullanıcı@ornek.com/kaynak1'
    id='d3079'
    type='result'>
 <query xmlns='jabber:iq:roster'>
   <item jid="kullanıcı1@ornek.com"/>
   <item jid="kullanıcı2@ornek.com"/>
   <item jid="kullanıcı3@ornek.com"/>
   ...
 </query>
</iq>
```

Şekil 3.14. Bilgi/istek paketi örneği

Bu paket yapısında tip (type) alanı istekleri ve sorgulamaları bildirmek için kullanılan "get", bir değeri ayarlamak yada kurmak için kullanılan "set", başarılı bir şekilde gerçekleştirilen "get" veya "set" paketlerinin geri dönüş cevabını ifade etmek için kullanılan "result" ve son olarak "get" veya "set" paketlerinin işlenmesinde yada dağıtımında ortaya çıkan hataları bildirmek için kullanılan "error" değerlerini içerebilir. Bilgi/sorgu (IQ) paketlerinin tip (type) özniteliğinin yapısal olarak davranış biçimini ifade etmek için ağ akış modeli Şekil 3.15.'te sunulmuştur.



Şekil 3.15. Bilgi/istek paketi veri akış modeli

Yukarıdaki örnek IQ paket yapıları ve veri akış modelinde de görüldüğü gibi IQ–get yada IQ–set paketleri isteği üreten ve cavaplayan varlıklar arasında bilinen ve özel olarak tanımlanmış XML yükler içerebilir. Bu durum bir alıcı tarafından işlenebilecek her türden verinin iletimini mümkün hale getirmektedir. XMPP'nin sunduğu esnek paket yapısı çalışmanın pek çok noktasında özelleştirilerek kullanılacaktır. Sunulan tüm bu süreç XMPP protokolü üzerinden gerçek zamanlı anlık mesaşlama uygulamalarının yanında istenilen pek çok uygulamanın bu altyapı üzerinde inşa edilmesine imkan sağlamaktadır.

3.3.4. XMPP'de güvenlik ve oturum

XMPP, varsayılan olarak TLS (Transport Layer Security) şifreleme desteği sunar. XMPP sunucusu tarafından TLS desteği sağlandığında istemciler TLS oturum başlatabilirler. Böylelikle istemci-sunucu arasındaki tüm veri akışı TLS kullanılarak şifrelenmiş olur. Bu aşamadan sonra istemci SASL (Simple Authentication and Security Layers) protokolünü kullanarak md5, plain, token gibi bir takım güvenlik mekanizmalarından uygun olanı seçerek kimlik doğrulaması yapabilir [3, 79, 89]. Kimlik doğrulama mekanizmalarının sunucu tarafından bildirimi ve sonrasında istemci tarafından md5 mekanizmasının kullanılarak kimlik bilgilerinin bildiriminin yapıldığı örnek paket yapıları Şekil 3.16'da görülmektedir.

```
Sunucu:
<stream:features>
 <starttls xmlns="urn:ietf:params:xml:ns:xmpp-tls">
 </starttls>
 <mechanisms xmlns="urn:ietf:params:xml:ns:xmpp-sasl">
   <mechanism>PLAIN</mechanism>
   <mechanism>DIGEST-MD5</mechanism>
   <mechanism>INTERNAL</mechanism>
 </mechanisms>
 <auth xmlns="http://jabber.org/features/iq-auth"/>
</stream:features>

İstemci:
<auth mechanism='DIGEST-MD5'
      xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
   NzE4OGJkMDdiMzAwODQwOTIxZDgwZWYyNjE0ZDEyZGI=
</auth>
```

Şekil 3.16. SASL mekanizması

Şekil 3.16'da görülen örnek kimlik doğrulama sürecinde, sunucu kimlik doğrulama mekanizmaları ve TLS gibi sunucu özellikleri istemciye bildirilmektedir. İstemci bu mekanizmalardan kendisi için uygun olan mekanizmayı seçerek oturum başlatma-kimlik doğrulama sürecini gerçekleştirmektedir. Sunucu tarafından uygun doğrulama yapılabilen kullanıcı için <success/> cevabı dönecektir.

3.4. Sonuç

P2P uygulamalarında düğümler arasında çoklu ortam verilerinin iletimi için aktarım katmanın protokollerinin bilinmesi ve bu protokollere uygun yaklaşımların geliştirilmesi gerekmektedir. Aktarım katmanı protokollerinden UDP, gerçek zamanlı veri iletiminde TCP'ye göre sağladığı önemli avantajlardan ötürü öncelikle tercih edilen protokoldür. Ancak ölçeklenebilirlik, hata ve tıkanıklık kontrolü gibi ihtiyaçlar UDP tabanlı RTP protokolünü ortaya çıkarmıştır. Günümüzde yetenekleri geliştirilerek pek çok gerçek zamanlı iletim uygulamalarının temel protokolü olmayı sürdürmektedir.

Ağ uygulamalarında sistemin bütünlüğünü sağlayan ve tüm katmanların kullandığı verilerin oluştuğu katman uygulama katmadır. XMPP, gerçek zamanlı çoklu ortam verilerinin eş düğümler arasında iletimi için geliştirilen uygulama ve düğümlerin doğrudan iletişim kurabilmesini sağlayan yeni NAT geçişi yöntemi için uygun bir uygulama protokolüdür. XMPP'nin özelleştirilebilirliği ve genişletilebilirliliği, geliştirilen yeni NAT geçişi yöntemi ve P2P uygulamaları için ihtiyaç duyulan el değiştirmeleri ve randevulaşmayı sağlayabilmektedir.

BÖLÜM 4. GELİŞTİRİLEN NAT GEÇİŞİ YÖNTEMİ ve UYGULAMASI

4.1. Giriş

Bu bölümde, tez çalışmasının temel araştırma alanını oluşturan P2P bilgisayar ağlarında ortam verilerinin iletim altyapısı için yeni bir durum tabanlı NAT geçişi ve uygulama çalışmaları sunulmaktadır. Çalışmanın ilk aşamasında P2P bilgisayar ağları için merkezi yönetim gereksinimlerini gerçekleştirmek ve P2P iletişim altyapısı için randevulaşma, el sıkışma vb. koordinasyon süreçlerinin yürütülmesi için gerekli merkezi sunucu mimarisi ele alınmaktadır. Geliştirilen sunucu uygulaması XMPP uygulama protokolü üzerinde inşa edilmiştir. XMPP protokolünün genişletilebilir yapısı P2P bilgisayar ağları için ihtiyaç duyulabilecek ilave protokollerin entegrasyonunda önemli standardizasyon ve ölçeklenebilirlik sağlamaktadır. Özellikle P2P iletişim uygulamalarında ortam verilerinin paylaşımı için önemli gereksinimler arasında sayabileceğimiz, kullanıcı listelerinin paylaşılması, durum yönetimi, multimedia içerik dışında kalan içeriğin güvenilir dağıtımı, randevulaşma ve el sıkışma süreçlerindeki içerik değişiminin sağlanması gibi adımlar bu tasarım üzerinde inşa edilmiştir.

Çalışmanın ikinci aşamasında ise NAT arkasındaki istemcilerin P2P iletişim kurabilmeleri için gerekli NAT geçişi için yeni bir yöntem tasarlanmıştır. Tasarlanan bu yöntem ile Bölüm 2.4.6.'da ifade edilen NAT geçişi yaklaşımlarından biri olan interaktif bağlantı kurulumu (ICE) protokolü, bağlantı kurulum süresi, paket kullanım oranı ve band genişliği kullanım değerleri gibi servis kalitesini doğrudan etkileyen parametreler bağlamında karşılaştırılmakta ve geliştirilen modelin avantajları ifade edilmektedir. Yapılan çalışma ile ICE protokolünün verimsiz kaldığı bağlantı süresi, paket sayısı ve band genişliği kullanımı gibi parametrelerin iyileştirilmesine yönelik durum tabanlı yeni bir yöntem önerilmiştir.

Son aşamada XMPP protokolü için önerilen protokol eklentileri ile NAT geçişi için ele alınan yeni tasarım bir uygulama mimari modeli üzerinde geliştirilmiştir. P2P bilgisayar ağlarında ortam verilerinin paylaşımı için internet tasarım felsefesine uygun olarak uçtan uca tam bir model ortaya konulmuştur.

4.2. Özelleştirilmiş Bilgi/Sorgu Paketi

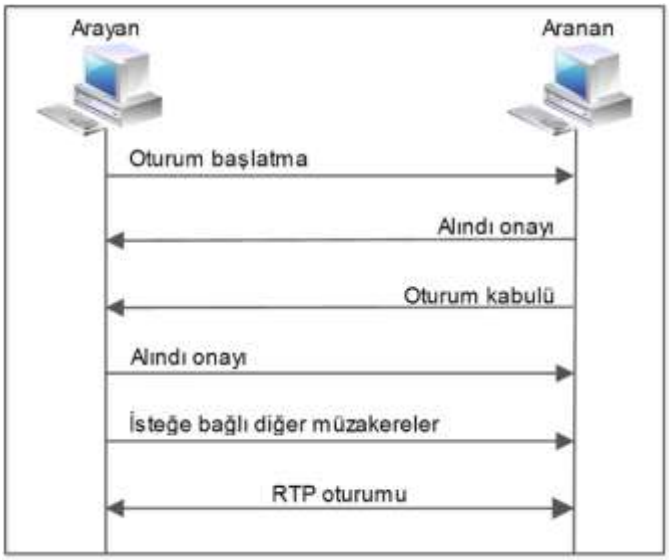
Bölüm 3.3.'te ele alınan XMPP Protokolü'nün XML temelli paket yapılarından bilgi/sorgu paketinin genişletilebilir ve özelleştirilebilir yapısı üzerinden sorgu yada istek tabanlı yeni paketler oluşturabilmek mümkündür. Bu aşamada, P2P bilgisayar ağları için genel anlık mesajlaşma fonksiyonalitelerinin yanında ihtiyaç duyulabilecek istemci-sunucu haberleşme tanımları için yeni özelleştirilmiş bilgi/sorgu paketleri tanımlanmıştır.

Yeni tanımlanan paket yapılarından biri, doğrudan iletişim kurmak isteyen iki düğüm arasında kullanılan oturum tanımlama paketleridir. Çalışmanın temel araştırma alanlarından NAT geçişi yönteminin ve RTP protokolünün ihtiyaç duyduğu özelleştirilmiş Oturum Tanımlama Protokol (SDP [92]) içeriğinin istemciler arasında değişiminin gerçekleştirilmesi sağlanmıştır.

XMPP çekirdek protokolünün [89] dışında özellikle anlık mesajlaşma süreçlerinde ihtiyaç duyulabilecek pek çok ilave protokol geliştirilmiş ve bu ilave protokoller protokol uzantıları (XEP) [93] altında tanımlanmıştır. Çalışmada geliştirilen NAT geçişi yönteminin ihtiyaç duyduğu randevulaşma süreçleri için XMPP çekirdek protokolünün tasarım sürecinde herhangi bir tasarım yapılmamıştır. Ancak daha sonra XEP-0167 [94] ile tanımlanan ve Jingle RTP Session olarak isimlendirilen protokol eklentisi standart UDP ve ICE protokolü için gerekli SDP değişimini gerçekleştirmek için tasarlanmıştır.

Bu protokol eklentisi, çalışmamızın da aktarım protokolü olarak kullandığı Gerçek Zamanlı Aktarım Protokolü üzerinden çoklu ortam verilerinin iletiminde kurulmak istenen oturum müzakeresi için uygulama altyapısı sunmaktadır. XEP-0167 ile

istemciler arasında, ortam verilerinin iletimi temelinde oturum başlatmak ve SDP içeriğinin el değişiminin gerçekleştirilebilmesi için XML paket yapıları tanımlanmıştır. Bu sürecin ağ akış modeli Şekil 4.1.'de görülmektedir.



Şekil 4.1. XEP-0167 RTP oturumu

XEP-0167 RTP oturum modelinde, multimedia oturum başlatmak isteyen istemci "Arayan", multimedia oturumun muhatabı olan istemci ise "Aranan" olarak isimlendirilmiştir.

Bu oturum sürecinde arayan-sunucu-aranan arasındaki müzakere ve randevulaşma sürecinde kullanılan bilgi/sorgu paket yapılarından oturum başlatma isteği paket örneği Şekil 4.2.'de görülmektedir. Örnekte arayan istemci, bir sesli görüşme isteğinde bulunmaktadır. Bu görüşmede, arayan istemcinin ses verisini kodlamada kullanabileceği yada desteklediği ses-kodlayıcılarının listesi ve ICE NAT geçişi için kullanılacak arayüz üyeliklerinin tanımları, tanımlanan XML paket formatına uygun olarak iletilmektedir.

Oturum başlatma istek paketi, XMPP sunucusu üzerinden aranan istemciye iletilir. Aranan istemci öncelikle bu paketin alındı onayını paket kimlik numarasını koruyarak arayan istemciye bildirilecektir.

```
<iq from='arayan@ornek.com/kaynak'
    id='ih28sx61'
    to=aranan@ornek.com/kaynak'
    type='set'>
 <jingle xmlns='urn:xmpp:jingle:1'
         action='session-initiate'
         initiator='arayan@ornek.com/kaynak '
         sid='a73sjjvkla37jfea'>
  <content creator='initiator' name='voice'>
    <description xmlns='urn:xmpp:jingle:apps:rtp:1' media='audio'>
      <payload-type id='0' name='PCMU'/>
      <payload-type id='97' name='speex' clockrate='8000'/>
      <payload-type id='18' name='G729'/>
    </description>
    <transport xmlns='urn:xmpp:jingle:transports:ice-udp:1'
               pwd='asd88fgpdd777uzjYhagZg'
               ufrag='8hhy'>
      <candidate component='1'
                 foundation='1'
                 generation='0'
                 id='el0747fg11'
                 ip='10.0.1.1'
                 network='1'
                 port='8998'
                 priority='2130706431'
                 protocol='udp'
                 type='host'/>
      <candidate <!—diğer üyeler -->/>
    </transport>
  </content>
 </jingle>
</iq>
```

Şekil 4.2. XEP-0167 Arayan oturum başlatma paketi

Arayan istemciye iletilecek alındı onay paket örneği Şekil 4.3.'te görülmektedir. Onay paketi, kullanıcının isteğine bırakılabileceği gibi multimedia oturumun kişi onayına bırakılmadan başlatılmasının istenebileceği uygulamalarda otomatik olarak da yapılabilir.

```
<iq from='aranan@ornek.com/kaynak'
    id='ih28sx61'
    to='arayan@ornek.com/kaynak'
    type='result'/>
```

Şekil 4.3. XEP-0167 Aranan alındı onayı

Eğer kullanıcı, bu oturum isteğini kabul etmek istemiyor ise alındı onayı paketinden hemen sonra ivedilikle bir meşgul paketi ile oturum başlatma isteğini reddedebilir. Arayan kullanıcının oturum başlatma isteğine aranan kullanıcı tarafından RTP

oturumunu sonlandırmak için gönderdiği örnek meşgul paketi Şekil 4.4.'te görülmektedir.

```
<iq from='aranan@ornek.com/kaynak'
    id='ch3vs61d'
    to='arayan@ornek.com/kaynak'
    type='set'>
  <jingle xmlns='urn:xmpp:jingle:1'
          action='session-terminate'
          initiator='arayan@ornek.com/kaynak'
          sid='a73sjjvkla37jfea'>
    <reason><busy/></reason>
  </jingle>
</iq>
```

Şekil 4.4. XEP-0167 Aranan meşgul paketi

Alındı onay paketi, aranan kullanıcının oturum başlatma isteğini kabul etme yada etmeme durumuna bakılmaksızın, önceki talep paketinin iletiminin gerçekleştiğinin istemciler arasında bildirimleri için tüm süreç boyunca gerçekleştirilmelidir.

Aranan kullanıcı, örnek sesli görüşme isteğini kabul etmesi durumunda arayan kullanıcıya oturum kimliğini (sid) muhafaza ederek, ses verisinin kodlanacağı türleri öncelik sırasına dikkat ederek ve ICE NAT geçişi için kullanılacak arayüz üyelikleri ile bildirir. Arayan kullanıcının sesli görüşme isteğine, aranan kullanıcı tarafından dönülen oturum kabul paketi örneği Şekil 4.5.'te görülmektedir.

Örnekte, aranan kullanıcının öncelikle speex kodlayıcısını tercih ettiği anlaşılmaktadır. Kodlayıcı tanımlamalarının yanında onay paketi NAT geçişi için kullanılabilecek ağ arayüz üyeliklerini de içermektedir. Aranan tarafından üretilen oturum onay paketi, arayan kullanıcıya ulaştığında, oturum başlatma paketine dönülen alındı onay paketinde olduğu gibi oturum onay paketinin alındı onayı, arayan kullanıcı tarafından aranan kullanıcıya dönülecektir. Bu süreçlerin karşılıklı arayan-aranan arasında tamamlanması ile ortam verilerinin iletimi için oturum başlatma ve müzakere süreci gerçekleştirilmiştir.

```
<iq from='aranan@ornek.com/kaynak'
    id='i91fs6d5'
    to='arayan@ornek.com/kaynak'
    type='set'>
 <jingle xmlns='urn:xmpp:jingle:1'
         action='session-accept'
         initiator='arayan@ornek.com/kaynak'
         responder='aranan@ornek.com/kaynak'
         sid='a73sjjvkla37jfea'>
  <content creator='initiator' name='voice'>
   <description xmlns='urn:xmpp:jingle:apps:rtp:1' media='audio'>
     <payload-type id='97' name='speex' clockrate='8000'/>
     <payload-type id='18' name='G729'/>
   </description>
   <transport xmlns='urn:xmpp:jingle:transports:ice-udp:1'
              pwd='YH75Fviy6338Vbrhrlp8Yh'
              ufrag='9uB6'>
     <candidate component='1'
                foundation='1'
                generation='0'
                id='or2ii2syr1'
                ip='192.0.2.1'
                network='0'
                port='3478'
                priority='2130706431'
                protocol='udp'
                type='host'/>
   </transport>
  </content>
 </jingle>
</iq>
```

Şekil 4.5. XEP-0167 Aranan onay paketi

4.2.1. Oturum tanımlama bilgilerinin haritalanması

Oturum tanımlama protokolü (SDP), multimedia oturumlarının başlatılması sürecinde medya detaylarının, aktarım adreslerinin ve diğer oturum tanımlama parametrelerinin tanımlanmasını sağlamaktadır [92]. SDP protokolü multimedia içeriğin nasıl taşındığına bakmaksızın standart bir medya tanımlama profili sağlar. Dolayısıyla SDP protokolü aktarılmak istenen medya tanımlarının müzakeresi dışında aktarılacak verinin UDP, RTP, SIP, RTSP, HTTP vb. yapılardan hangisi ile taşındığıyla ilgilenmez. SDP protokolü, taşınacak medya için, medya tipi (video, audio, vb), aktarım protokolü tanımı (RTP, UDP, SIP, H.320 vb.), medya formatı (H.264, MPEG, speex vb.), multicast iletişim adresi, aktarım portu, uzak adres ve portu gibi tanımlamaların ilgili düğümler arasında paylaşılabilmesi için ortak bir paket formatı sunmaktadır. İstemciler arasında medya tanımları için bir standart oluşturmak üzere tasarlanan SDP paket formatı, ascii kodlardan ve her değerin bir satır içerisinde anahtar=değer şeklinde tanımlanması ile oluşmaktadır. Bu tanımlama

yapısının XMPP protokolüne uygun bilgi/sorgu paket formatına dönüşüm örneği Şekil 4.6.'da görülmektedir.

```
SDP Tanımı:
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 width=800; height=600; profile-level-id=42A01E;

XML Haritası:
<description xmlns='urn:xmpp:jingle:apps:rtp:1' media='video'>
  <payload-type id='98' name='H264' clockrate='90000'>
    <parameter name='width' value='800'/>
    <parameter name='height' value='600'/>
    <parameter name=' profile-level-id' value='42A01E'/>
  </payload-type>
</description>
```

Şekil 4.6. Medya oturum tanımlarının haritalanması

Oturum tanımlamalarının haritalandığı örnekte, medya türü olarak "m=video …" satırı ile bunun bir gerçek zamanlı video/ses konferans ve iletişim uygulamaları için tanımlanan profillerde yer alan bir video içeriği olduğu, "a=rtpmap:98…" satırı ile bunun dinamik bir yük tipine sahip olduğu ve kodlama tekniği olarak H264 kodlayıcı ile 90000 bit oranında kodlandığı, "a=fmtp:98…" satırı ile bu kodlama tekniğine özgü parametrelerin dinamik bir yük tipinde, 800 piksel genişliğinde ve 600 piksel yüksekliğinde ve H264 temelband versiyon 3.0'ı kullandığını ifade etmektedir. Bu tanımlamaların bilgi/sorgu paketinin "<description ..>" alt elementi içerisinde tanımlanabilmesi ile medya tanımlarının XMPP altyapısına uygun olarak dağıtımı sağlanmıştır. Kullanılan H264 video kodlama tekniği dinamik bir kodlama desteği sunmakta ve video tabanlı gerçek zamanlı iletim uygulamaları için servis kalitesi noktasında önemli kazanımlar sağlamaktadır. Geliştirilen uygulamada video kodlayıcı olarak H264 tercih edilmiştir.

4.2.2. Aktarım arayüzlerinin haritalanması

Oturum tanımlama protokolü (SDP), multimedia içeriğin taşınması için oluşturulacak oturumlarda iki düğüm arasında aktarılacak medya tanımları ve özellikleri için bir profil sunar. Ancak bu tanımlarda yer alan aktarım için kullanılacak arayüz bilgileri (adres, port) NAT arkasındaki istemciler için genellikle erişilemezdir. SDP paketinde

tanımlanan aktarım arayüzleri doğrudan iletişim olanağı olmayan bu kullanıcılar için çoğunlukla dikkate alınmazlar. RTP protokolünü istemciler arasında doğrudan iletişim kurabilecekleri arayüzlerden bağlamak gecikme, paket kaybı vb. servis kalitesini etkileyen pek çok parametreden dolayı bu çalışmanın ana amacını oluşturmaktadır. Bu problemi gidermek için ortaya konan literatür çalışmaları ve yöntemler Bölüm 2'de ele alınmıştır. XMPP protokolü bu yöntemlerden ICE protokolü için aktarım arayüzlerinin tanımlandığı üyelik bilgilerinin, istemciler arasında değişimi için bir paket tanımı geliştirmiştir [94]. Bu üyelik bilgileri, iki istemci arasında doğrudan iletişim kurulabilme potansiyeline sahip IP:port çiftlerinden oluşmaktadır. İstemciler arasında aktarım arayüzlerinin el değiştirilmesi için bilgi/sorgu paketinde haritalanmıştır. İstemciler arasında el değiştiren potansiyel iletişim arayüzleri üzerinden ICE protolünün bağlantı kontrol süreci yürütülmektedir. Örnek bir potansiyel bağlantı arayüzlerinin bilgi/sorgu paket formatına dönüştürülmesi Şekil 4.7.'de sunulmuştur.

```
SDP Tanımı:
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
a=candidate:1 1 UDP 2130706431 10.0.1.1 8998 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45664 typ srflx
                        raddr 10.0.1.1 rport 8998
XML Haritası:
<transport xmlns='urn:xmpp:jingle:transports:ice-udp:1'
           pwd='asd88fgpdd777uzjYhagZg'
           ufrag='8hhy'>
     <candidate component='1'
                foundation='1'
                generation='0'
                id='e10747fg11'
                ip='10.0.1.1'
                network='1'
                port='8998'
                priority='2130706431'
                protocol='udp'
                type='host'/>
     <candidate component='1'
                foundation='2'
                generation='0'
                id='y3s2b30v3r'
                ip='192.0.2.3'
                network='1'
                port='45664'
                priority='1694498815'
                protocol='udp'
                rel-addr='10.0.1.1'
                rel-port='8998'
                type='srflx'/>
</transport>
```

Şekil 4.7. Aktarım arayüzlerini haritalanması

Örnekte STUN ve TURN sunucuları üzerinden elde edilen potansiyel bağlantı arayüzleri ve ICE tanımlarının yer aldığı SDP formatı <transport> elementi altında tanımlanmıştır. Her arayüz tanımı için bir <candidate> elementi yer almaktadır. Bu tanımlar, istemciler arasında XMPP üzerinden el değiştirdikten sonra ICE bağlantı kontrolü aşamasında tekrar standart SDP formatına dönüştürülmektedir.

XMPP protokolünde P2P iletişim kurabilmek için ortaya konulan paket tanımlamaları doğrudan iletişim kurulumu dışında yalnızca ICE protokolü için özelleştirilmiştir. Bu çalışmada önerilen, yeni NAT geçişi yönteminin istemciler arasında el sıkışma ve müzakere süreçlerinde ihtiyaç duyduğu oturum tanımlama bilgilerinin dağıtımı için yeni bilgi/istek paket yapıları tanımlanmıştır. Yeni NAT geçişi yönteminde ihtiyaç duyulan özel IP:port, kamusal IP:port, EPD-Near ID ve relay adres tanımları için tanımlanan özelleştirilmiş SDP bilgi/sorgu paket formatı Şekil 4.8.'de sunulmuştur.

```
<transport xmlns='jabber:client'
           pwd='6rmhqlqc703rqj7o7gic8mt97a'
           ufrag='4fnnq1a61aual6'>
 <address component='1'
          local-addr='192.168.2.13'
          public-addr='176.239.98.35'
          local-port='10782'/>
 <nearID component='2'
         nearId='c81f339ebe2bf87981d3cf1becb7ecbafbb65cdcd4…'/>
 <candidate component='3'
            foundation='1'
            generation='0'
            id='se6232f12c'
            ip='192.168.10.211'
            network='1'
            port='64902'
            priority='2815'
            protocol='udp'
            rel-addr='176.239.98.35'
            rel-port='10001'
            type='relay'/>
</transport>
```

Şekil 4.8. Özelleştirilmiş aktarım arayüzü tanımlama bilgilerinin haritalanması

Özelleştirilmiş potansiyel aktarım arayüzleri yapısı, geliştirilen NAT geçişi yönteminin ihtiyaç duyduğu oturum bilgilerini içermektedir. Bunlardan <address…> alt elementi ile ifade edilen tanımda istemcinin yerel IP adresi, UDP portu ve XMPP

sunucusunda oturum başlatma sürecinde sunucu tarafından sağlanan ve istemcinin NAT arayüzünü ifade eden kamusal IP adresi bilgilerini içermektedir.

İstemcinin NAT arayüzlerinin tanımlandığı birinci alt elementten sonra <nearID…> etiketi ile istemcinin Cumulus [95] sunucusu üzerinden elde ettiği EPD tanımı ifade edilmiştir. Bu tanımlama NAT arayüzlerinin tanımlandığı ilk kısımda olduğu gibi geliştirilen NAT geçişi yönteminde kullanılmak üzere istemciler arasında el değiştirecektir.

```
<iq from='arayan@ornek.com/kaynak'
    id='ha58mr54'
    to=aranan@ornek.com/kaynak'
    type='set'>
 <jingle xmlns='urn:xmpp:jingle:1'
         action='session-initiate'
         initiator='arayan@ornek.com/kaynak '
         sid='32sdlw3341kfs33'>
  <content creator='initiator' name='video'>
    <description xmlns='urn:xmpp:jingle:apps:rtp:1' media='video'>
      <payload-type id='98' name='H264' clockrate='90000'>
        <parameter name='width' value='800'/>
        <parameter name='height' value='600'/>
        <parameter name=' profile-level-id' value='42A01E'/>
      </payload-type>
    </description>
    <transport xmlns='jabber:client'
               pwd='6rmhqlqc703rqj7o7gic8mt97a'
               ufrag='4fnnq1a61aual6'>
      <address component='1'
               local-addr='192.168.2.13'
               public-addr='176.239.98.35'
               local-port='10782'/>
      <nearID component='2'
              nearId='c81f339ebe2bf87981d3cf1becb7ecbafbb65cd…'/>
      <candidate component='3'
                 foundation='1'
                 generation='0'
                 id='se6232f12c'
                 ip='192.168.10.211'
                 network='1'
                 port='64902'
                 priority='2815'
                 protocol='udp'
                 rel-addr='176.239.98.35'
                 rel-port='10001'
                 type='relay'/>
    </transport>
  </content>
 </jingle>
</iq>
```

Şekil 4.9. Özelleştirilmiş bilgi/sorgu paketi

Son olarak ICE protokolü için tasarlanan potansiyel iletişim arayüzlerinin tanımlandığı <candidate …> tanımı referans alınmıştır. Ancak bu tanımda tüm potansiyel iletişim arayüzlerinin tanımlanması yerine sadece TURN server üzerinden elde edilen aktarım-röle (relay) arayüz tanımı kullanılmıştır. Gerek multimedia paylaşım oturumunda taşınacak medya bilgisinin tanımlandığı oturum tanımlama protokolünün, gerekse de yeni NAT geçişi yönteminde kullanılan potansiyel aktarım arayüz bilgilerinin yer aldığı özelleştirilmiş bilgi/sorgu paket yapısı Şekil 4.9.'da görülmektedir.

Şekilde sunulan özelleştirilmiş bilgi/sorgu paketi XMPP çekirdek protokolü için geliştirilen protokol eklentilerinden XEP-0167'de [94] tanımlanan bilgi/sorgu paket formatına uygun olarak tasarlanmıştır.

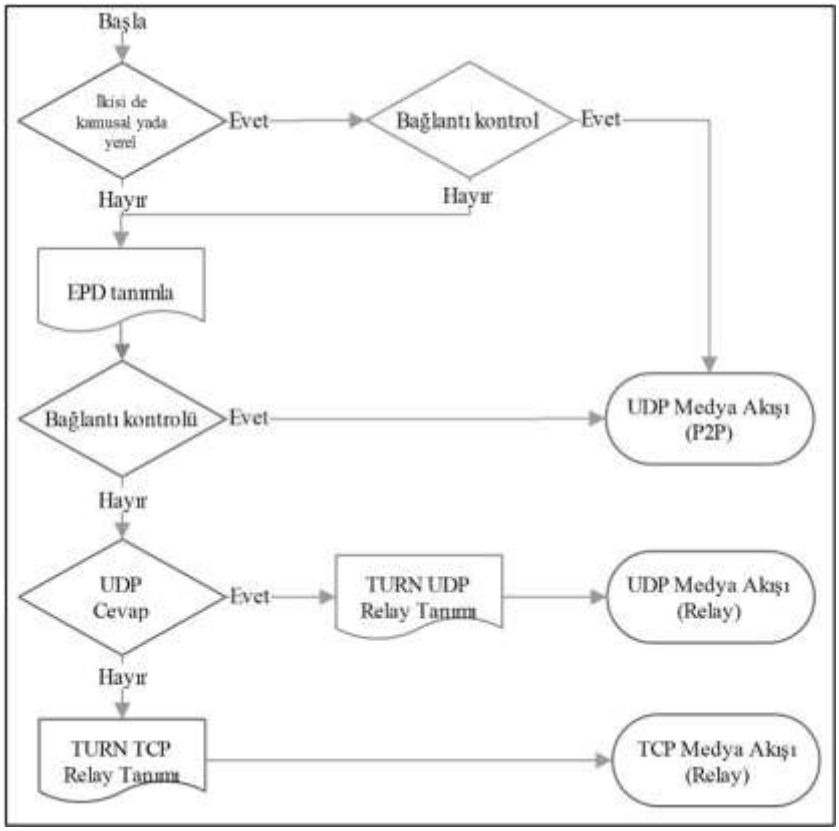
4.3. Önerilen Durum Tabanlı NAT Geçiş Yöntemi

P2P bilgisayar ağlarında ortam verilerinin istemciler arasında dağıtımındaki temel problem, NAT arkasındaki kullanıcıların birbirleri ile RTP oturum başlatabilmesinin önündeki doğrudan iletişim kurabilecek arayüzlere erişilememesidir. Bu problemin temel nedeni RTP oturumu için tanımlama bilgileri sunan oturum tanımlama protokolü kullanılarak istemciler arasında el değiştirilen aktarım arayüz bilgilerinin (IP:port çifti) kullanılamamasıdır.

Bahse konu bu durum literatürde NAT geçişi olarak ele alınmaktadır. Bölüm 2.4.'de detaylandırılan NAT geçişi yöntemleri, altyapı değişikliği gerektirmeleri, manual tanımlama gereksinimleri ve bazı NAT davranışları altında sonuç alınamaması gibi gerekçelerden dolayı ev kullanıcıları ve lokasyon bağımsız kullanıcılar için uygulama olanağı bulamamaktadır. Bu modeller arasından ICE protokolü, gerek altyapı değişikliği istememesi, gerekse de farklı NAT davranışları arkasındaki kullanıcılar için uygun NAT geçişi sağlaması ile öne çıkmaktadır. Ancak ICE protokolü, tüm yerel ve uzak arayüz çiftlerini tek tek deneyerek bağlantı kontrolü yapmasından kaynaklanan yüksek bağlantı kurulum gecikmesine neden olması,

yüksek band genişliği ve paket kullanım oranları ile dikkat çeken araştırma alanlarındandır [30].

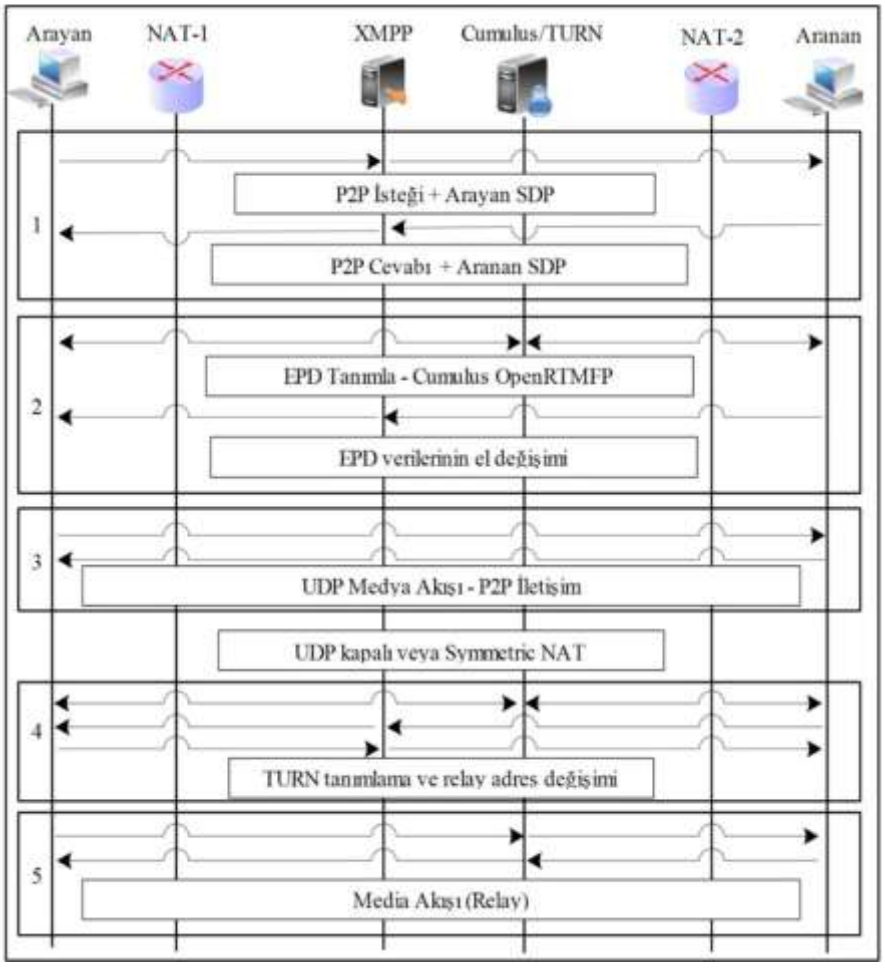
Tüm bu yöntemler üzerinde yapılan incelemeler sonucunda yeni bir durum tabanlı NAT geçişi yöntemi tasarlanmıştır ve SBN (State Based NAT Traversal) olarak isimlendirilmiştir. SBN yönteminin, P2P bilgisayar ağlarında çoklu ortam verilerinin iletim uygulamaları için önemli bir kazanım sağlayacağı öngörülmüştür. Yapılan çalışma sonucunda elde edilecek NAT geçişi yöntemi, alt yapı değişikliği gerektirmemesi, manual müdahale ihtiyacı olmaması, tüm NAT davranışları için uygun iletişim arayüzü sunmasının yanında, bağlantı gecikmesi, paket sayısı ve band genişliği kullanımı gibi parametreler yönünden mevcut yöntemlerden daha iyi sonuçlar ortaya koymalıdır.



Şekil 4.10. SBN algoritması

Ortaya konulan bu parametreler dikkate alınarak tasarlanan NAT geçişi yönteminin algoritma akış diyagramı Şekil 4.10.'da görülmektedir. Bu algoritmaya göre,

öncelikle P2P iletişim kurmak isteyen düğümler birbirlerine göre yerel yada kamusal olma durumlarına göre doğrudan iletişim kurmayı denerler. Eğer belirlenen kamusal yada yerel arayüzler üzerinden iletişim kurulumu başarılı ise P2P medya akışı başlatılır. Aksi taktirde düğümler Cumulus Server [95] üzerinden uygun UDP çiftleri için denetlenmektedir. Eğer belirlenen bu arayüzler üzerinden bağlantı sağlanabilir ise belirlenen IP ve port verisi Real-time Transport Protocol (RTP) aktarım arayüzü olarak tanımlanır. Düğümler arasında P2P bağlantı kurulamadığı durumlarda TURN server üzerinden UDP relay arayüzler belirlenir ve RTP veri akışı UDP Relay Server üzerinden sürdürülür. İstemcinin UDP aktarımının kapalı yada güvenlik duvarlarınca engellendiği durumlarda UDP bağlantı kurulumu mümkün olmayacaktır. Bu durumda TCP relay arayüzler tanımlanır ve veri akışı TCP olarak sürdürülür. ICE ile tasarlanan SBN yöntemleri arasındaki farkın anlaşılmasına yönelik veri akış diyagramı Şekil 4.11.'de görülmektedir.



Şekil 4.11. SBN yöntemi ağ akış diyagramı

İki istemci arasında, ortam verilerinin iletimi için iletişim isteğinin oluşması ve medya akışının başlatılması adımları görülmektedir. Bu adımlar, Arayan'ın, Aranan ile P2P iletişim kurma isteği üretmesi üzerine kurgulanmıştır. Şekil 4.11.'de gösterilen SBN yönteminin ağ akış adımları ele alındığında;

1. Arayan olarak isimlendirilen istemci, 4096'nın üzerindeki herhangi bir boş port üzerinden UDP soket açar ve yerel IP:port ve kamusal IP bilgilerini içeren NAT arayüzlerini içeren özel oturum tanımlama bilgileri ile birlikte P2P iletişim isteğini (özelleştirilmiş bilgi/sorgu paketi) XMPP randevu sunucusuna iletir [80, 98]. Randevu sunucusu, gelen iletişim isteğine Arayan istemcinin kamusal IP adresi ve yerel IP:port bilgilerini içeren NAT arayüzlerini içeren özelleştirilmiş bilgi/sorgu paketi olarak Aranan istemciye iletir. Bu istek Aranan istemci tarafından kabul edilirse (Arayan istemcinin UDP soket'ine benzer bir UDP soket açarak), randevu sunucusu üzerinden, Arayan istemciye Aranan istemcinin özelleştirilmiş bilgi/sorgu paketi ile birlikte cevap paketi döner. Bu aşamada birinci adım için gerekli müzakere süreci ve içerik değişim süreci tamamlanmıştır. Özelleştirilmiş oturum tanımlama paketinde tanımlanan içerik üzerinden istemcilerin bağlantı kontrolleri gerçekleştirilir. Belirlenen üyelik çiftleri üzerinden iletişim sağlanabilmesi durumunda süreç tamamlanır. Bu durumda, istemcilerin yerel yada kamusal ağda olma durumlarına göre SBN yöntemi, düğümler arasında doğrudan iletişimi başlatmış olacaktır. Bağlantının başarısız olması durumunda ikinci aşamayla NAT geçişi süreci devam edecektir.
2. Arayan ve Aranan istemciler, Cumulus (UDP port 1935 dinliyor) Server üzerinden EPD verilerini tanımlar ve NearID'leri üretir. Üretilen NearID'ler XMPP server üzerinden özelleştirilmiş bilgi/sorgu paketleri aracılığı ile Arayan ve Aranan istemciler arasında el değiştirilerek FarID'ye dönüştürülür.
3. Bu aşamada RTMFP protokolünün kullandığı UDP delik açma yöntemi temelinde bağlantı testleri geçekleştirilecektir. Bağlantı kurulum denemesinin başarılı olması durumunda üretilen iletişim kanalları üzerinden doğrudan medya akışı Arayan ve Aranan istemciler arasında sürdürülecektir. Bağlantı kurulumunun başarısız olduğu durumda sonraki aşamaya geçilecektir.

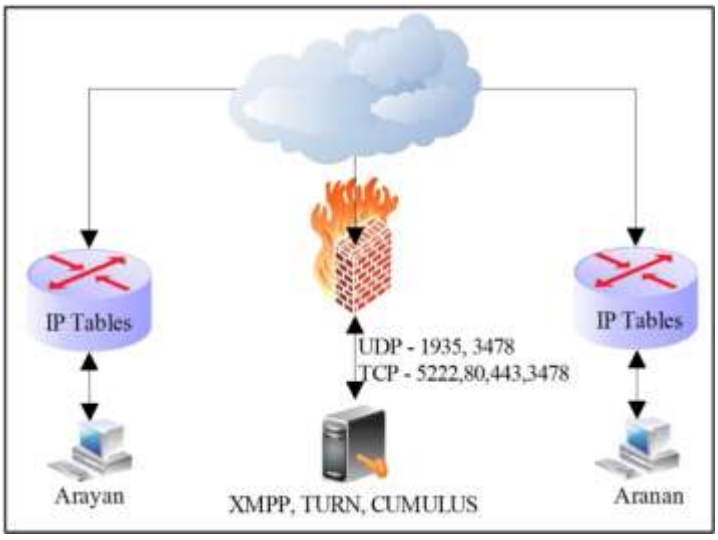
4. Bu aşama, istemciler arasında doğrudan iletişimin kurulamadığı durumdur. Bu durum istemcilerden herhangi birinin Symmetric NAT arkasında iken diğerinin ya Symmetric NAT yada Port Restricted NAT arkasında olması ile ifade edilebileceği gibi istemcilerden herhangi birinin yada ikisinin güvenlik duvarı ile UDP aktarımının engellendiği durum olarak da ortaya çıkabilir. Bu aşamada doğrudan iletişim kurulumu tekrar denenmeyecektir. İstemcilerin doğrudan iletişim kuramayacağı bu durumlar da TURN server üzerinden medya aktarımı denenecektir. Öncelikle istemcilerin TURN server üzerinde tanımlanan relay arayüzleri, özelleştirilmiş bilgi/sorgu paketi ile XMPP randevu sunucusu üzerinden değiştirilir.
5. ICE protokolünde ifade edilen bağlanılabilirlik kontrol süreci yalnızca relay adres çiftleri üzerinden denetlenir. El değiştirilen relay arayüzler yerel-uzak aday çiftleri olarak bağlantı kontrol sürecinde kullanılır. Relay arayüz çiftlerinin başarılı olması durumunda bu arayüzler istemcilerin RTP aktarım arayüzleri olarak kullanılacaktır.

SBN yönteminin gerçek ortamda denetlenmesi ve performans değerlendirmesinin yapılabilmesi için bir uygulama geliştirilmiş ve yöntem bu uygulama üzerine entegre edilmiştir.

4.4. SBN Yönteminin Gerçeklenmesi ve Başarım Analizi

SBN yönteminin başarım analizlerini değerlendirmek için Şekil 4.12'de sunulan ağ topolojisi üzerinde çalışan uygulama geliştirilmiştir. Geliştirilen uygulama ile P2P bilgisayar ağları uygulamalarında ortam verilerinin iletiminin uçtan uca tam bir model sunabilmesi adına istemciler arsında randevu sunucusu olarak XMPP kullanılmıştır. Randevu sunucusu uygulamanın merkezinde yer alarak, yüksek bant genişliği ihtiyacı olmayan, oturum açma, yetkilendirme, kullanıcı listelerini sunma, durum değişikliklerini yönetme, anında mesajlaşma gibi işlevleri yerine getirmektedir. Bunların dışında randevu sunucusunun temel işlevi, istemciler arasında, doğrudan iletişim için gerekli oturum tanımlama parametrelerini

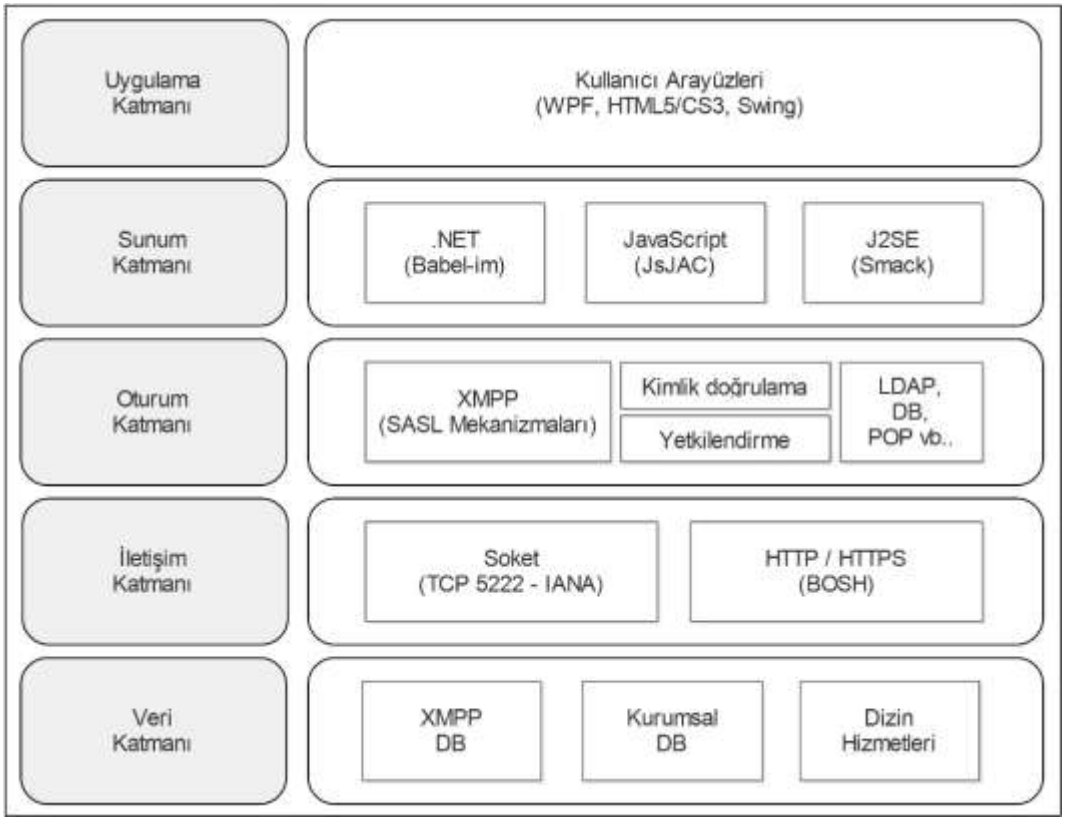
yönetmektir. Oturum tanımlama parametlerinin istemciler arasında dağıtımı için özelleştirilmiş bilgi/sorgu paketleri kullanılmaktadır.



Şekil 4.12. Uygulama ağ modeli

Uygulanan senaryoda, istemcilerin kamusal ağ çıkışları, IP Tables (Linux Debian 7) ile yönetilmiştir. IP Tables üzerinden farklı NAT davranış biçimleri modellenmiş (Full Cone NAT, Restricted Cone NAT, Port Restricted Cone NAT ve Symmetric NAT) ve tüm NAT tipleri karşılıklı olarak belirtilen ağ topolojisinde denenmiştir. Senaryoya göre XMPP altyapısına sahip randevu sunucusu TCP-5222, 80 ve 443 portu üzerinden hizmet vermektedir. XMPP server TCP soket bağlantıları için 5222 portundan hizmet vermektedir. İstemcilerin HTTP yada HTTPS bağlatıları için TCP 80 ve 443 portları kullanılmaktadır. HTTP bağlantıları için Internet Information Server (IIS) üzerinden ters vekil sunucusu kullanılmıştır. Ters vekil sunucu, gelen URL'leri tanımlanan düzenli ifadeler bağlamında dönüştürmekte ve HTTP isteklerini istenilen porta yönlendirebilmektedir. Böylelikle XMPP sunucusu üzerinde farklı ağ modellerinden oturum açmak isteyen kullanıcılar için erişilebilirlik sağlanmıştır. Öncelikle Arayan ve Aranan istemci XMPP üzerinden oturum açmakta ve kullanıcı listelerine ve bu kullanıcıların durumlarına erişmektedir. XMPP üzerinde oturum açan kullanıcı, oturum başlatma sürecinin son aşamasında kendine ait NAT arayüz tanımlarını XMPP üzerinden elde etmekte ve saklamaktadır. Bu bilgiler SBN yönteminin birinci adımında kullanılmaktadır. XMPP üzerinde oturum açan

kullanıcılar aynı zamanda iletişim listesinde bulunan kullanıcıların durum değişikliklerini görebilmekte ve kullanıcının çevrimiçi, çevrimdışı, dışarda, meşgül vb. durumları kullanıcı listesi arayüzüne yansıtılmaktadır. Kurgulanan test senaryosuna göre Arayan istemci NAT-1 arkasında iken NAT-2 arkasında bulunan Aranan istemci ile P2P bağlantı kurmak istemektedir (video görüşme isteği). Modelde bulunan TURN server [100] UDP-3478, TCP-80 ve 443 numaralı portları dinlemektedir. Modelde yer alan TURN server istemcilerin relay arayüzlerini tanımlamak için kullanılmaktadır. Diğer bir sunucu hizmeti olan Cumulus OpenRTMFP servisi UDP-1935 portundan hizmet vermektedir. Cumulus servisi istemcilerin NearID-FarID'ler üzerinden NAT geçişi sağlayabilmeleri için gerekli EPD verilerini sağlamaktadır. Geliştirilen uygulamanın katmansal mimarisi Şekil 4.13.'te sunulmuştur. Uygulama 5 temel katmandan oluşmaktadır.



Şekil 4.13. Uygulama katmansal modeli

Aşağıda, uygulamada yer alan katmanların işlevleri maddeler halinde açıklanmıştır.

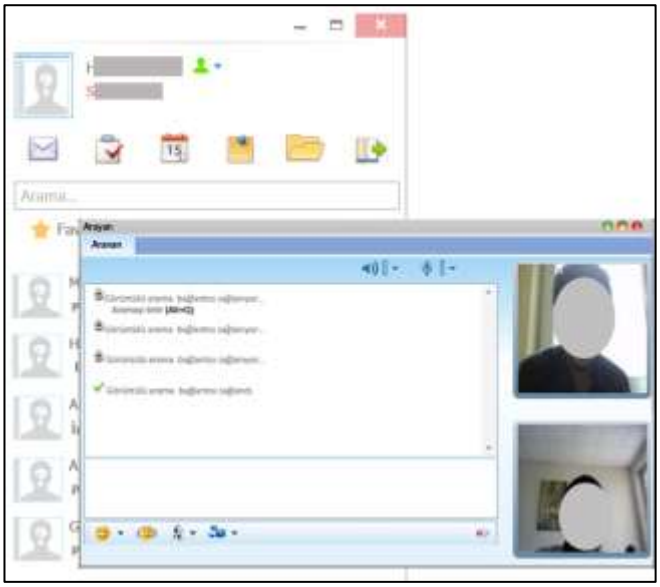
Uygulama katmanı: En üst katmanda yer alan bu katman, kullanıcı arayüzünü ifade etmektedir.

Sunum katmanı: Arayüz katmanı altında XMPP istemci kütüphaneleri ve bunların gerektirdiği dönüşümlerin yapıldığı katmandır.

Oturum katmanı: Sunum katmanından alınan verilerin XMPP sunucusuna iletimi için gereken oturum süreçlerinin yönetildiği, kimlik doğrulama ve yetkilendirme süreçlerinin yürütüldüğü katmandır.

İletişim katmanı: Bu katmanda istemci-sunucu arası bağlantılar yönetimi gerçekleştirilir. İstemci-sunucu arasında TCP tabanlı soket, HTTP yada HTTPS bağlantılar bu katman aracılığıyla gerçekleştirilir.

Veri katmanı: En alt katmanda yer alan ve uygulamanın ihtiyaç duyduğu bilgilerin organizasyonundan sorumludur. Şekil 4.13.'te sunulan katmansal mimariye uygun olarak geliştirilen test uygulamasının ana ekran görüntüsü ve çoklu ortam verilerinin gerçek zamanlı iletimi için kullanılan haberleşme ekranı görüntüsü Şekil 4.14.'te gösterilmiştir.



Şekil 4.14. Uygulama ekranları

Şekil 4.12.'de sunulan ağ modelinde, geliştirilen SBN yönteminin yer aldığı uygulama ile NAT geçişi tekniği olarak ICE protokolünü kullanan açık kaynak kodlu VoIP yazılımı Jitsi'nin 2.2.4603 sürümü ayrı ayrı koşturulmuştur [99]. Performans değerlendirilmesinde kullanıcılacak veriler, Arayan istemci üzerinde koşturulan Wireshark Network Analyzer yazılımının 1.8.4 sürümü ile elde edilmiştir [97]. Uygulamaların denetlenmesi ve performanslarının değerlendirilebilmesi için kurulan yöntemlerin çalıştırıldıkları sistemlere ait konfigürasyon ve donanım detayları Tablo 4.1.'de görülmektedir.

Tablo 4.1. Sistem konfigürasyonu

Bileşen	Özellikler
İstemci platformları	Windows 7 Ultimate 64 bit i7 8 GB RAM'li notebook, wireless ethernet
Sunucular	Randevu Sunucusu: Windows Server 2008, Xeon 3.1 GHz işlemci, 4 GB RAM TURN Server, Cumulus Server: Linux Ubuntu 13.04, Xeon 3.1 GHz işlemci, 4 GB RAM, çift ethernet
Uygulama Geliştirme Araçları	Sunucu: Java İstemci GUI: C# İlave protokoller: ActionScript
Sunucu Band Genişliği	5 Gbit Metro İnternet
İstemci Band Genişliği	8 Gbit ADSL İnternet
NAT aygıtları	4 farklı NAT tipi için Linux Debian 7 sürümü üzerinde IPTables ile tanımlanmıştır. (bridge mode-PPPoE bağlantı). Donanım, intel i3 cpu'lu, 4 GB RAM, çift ethernet.

NAT geçişi yöntemi olarak SBN kullanan uygulama ile ICE yöntemini kullanan Jitsi yazılımı Şekil 4.12.'de belirtilen ağ topolojisinde koşturulmuş, istemcilerin farklı NAT tipleri arkasında iken NAT geçişi performansları temelinde;

- Bağlantı kurulum süresi,
- Paket kullanım sayısı,
- Band genişliği kullanımı

gibi parametreler arayan istemci üzerinden izlenmiştir. Uygulamaların NAT geçişi başarımları, ICE ve SBN için kullanılan farklı NAT davranışları için bağlantı modelleri ve kurulan bağlantı tipleri Tablo 4.2'de görülmektedir. Tüm NAT tipleri arkasında iki yöntem de aynı davranışları sergilemektedir. Hem ICE hem de SBN, arayan ve aranan istemcinin SYM-SYM yada PRC-SYM arkasında olmaları durumunda TURN sunucu üzerinden relay bağlantı kurulabilmektedir. Diğer 13 farklı durumda ise doğrudan bağlantı kurulabilmiştir.

Tablo 4.2. Yöntemlerin NAT geçişi performansı

		Aranan			
		FC	**RC**	**PRC**	**SYM**
Arayan	**FC**	P2P	P2P	P2P	P2P
	RC	P2P	P2P	P2P	P2P
	PRC	P2P	P2P	P2P	RELAY
	SYM	P2P	P2P	RELAY	RELAY

Değerlendirmelerde, 4 farklı NAT davranış biçiminde, SBN ile ICE protokolü için doğrudan iletişim isteği üretilmiştir. Üretilen oturum başlatma istekleri sonucunda, RTP medya akışı başladığındaki bağlantı kurulum süresi, bant genişliği kullanımı ve kullanılan kontrol paket sayıları olmak üzere üç kategoride izlenmiştir.

SBN yönteminin test edilmesi için geliştirilen uygulama ile NAT geçiş yöntemi olarak ICE protokolünü kullanan Jitsi yazılımı 16 farklı durum için elde edilen uç değerleri değerlendirme dışı bırakmak için 10'ar kez çalıştırılmıştır. İki uygulamanın da 16 farklı durum ve izlenen 3 farklı performans parametreleri kapsamında oluşan değerler tablosu elde edilmiştir. Elde edilen ölçüm değerlerinin maksimum ve minumum değerleri değerlendirme dışı tutularak ölçüm sonuçlarının ortalamalarından oluşan veriler değerlendirme parametreleri için kullanılmıştır.

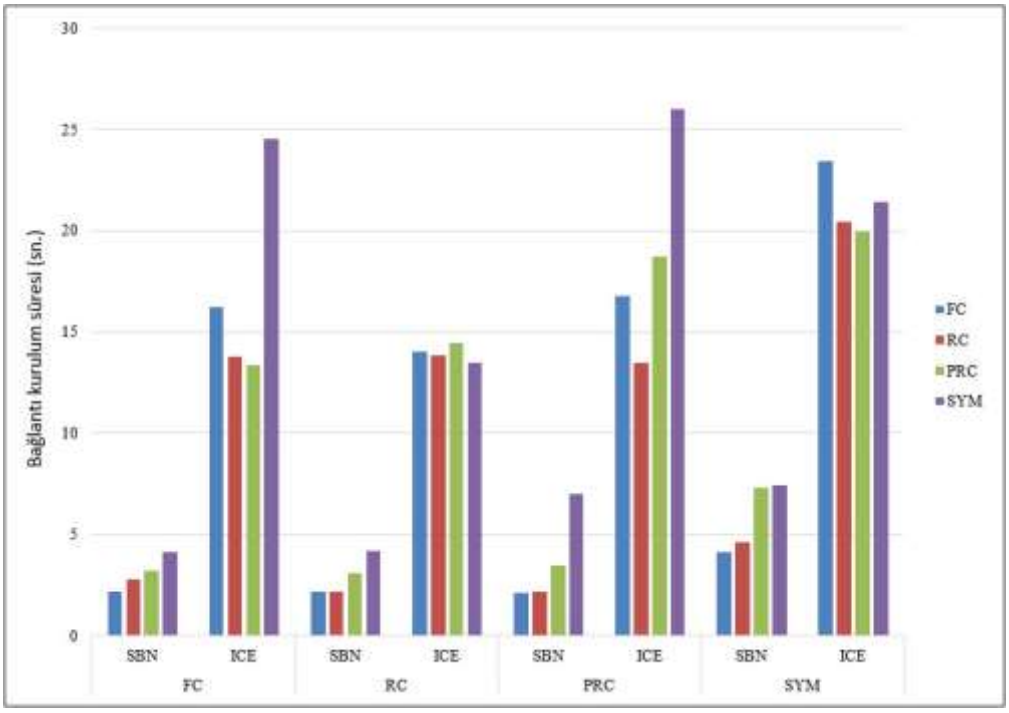
4.4.1. Bağlantı kurulum süresi

Yapılan literatür incelemelerinde, farklı özel ağlarda yer alan istemciler arasında P2P iletişimin ICE protokolü kapsamında verimli bir şekilde başlatılabilme süresinin 4-5 saniye civarında olduğu ifade edilmektedir [30]. Ancak gerçek uygulama ortamlarında gecikmenin çok daha fazla olduğu görülmektedir. Bağlantı kurulum süresinin ölçümü için uygulama modelinde sunulduğu gibi Arayan ve Aranan istemci olarak konumlandırılan bilgisayarlarda SBN yöntemini kullanan uygulama ve ICE yöntemini kullanan Jitsi yazılımı ayrı ayrı 10'ar kez çalıştırılmıştır. Uygulamalar yürütülürken istemciler üzerindeki ağ kaynaklarını ve bilgisayar performansını etkileyebilecek tüm servisler kapatılmıştır. Arayan istemci üzerinden ağ trafiği izlenmiş ve video görüşme isteğinin randevu sunucusuna iletilmesinden Arayan ile Aranan istemci arasında video akışının başlaması ile ağ izlenmesi sonlandırılmıştır. Elde edilen ağ akışları ilgili düğümler arasındaki IP adresi ve port numarası ve protokol tipi ile filtrelenerek paketlerin zaman damgaları çıkarılmıştır. Video görüşme isteğinin üretildiği ilk paket ile video verisinin iletildiği ilk paketin zaman damgaları arasındaki fark bağlantı kurulum zamanı olarak kayıt altına alınmıştır. Belirtilen bu süreç iki farklı NAT geçiş yöntemi için ayrı ayrı 10'ar kez tekrarlanmıştır. Her bir model için elde edilen minimum ve maksimum değerler atıldıktan sonra geriye kalan 8 farklı bağlantı kurulum zamanı verilerinin ortalamaları alınmış ve sonuçlar iki modelin ortaya koyduğu performansları kıyaslayabilecek şekilde grafiğe yansıtılmıştır. İzlenen performans kriterine özgü ölçüm değerlerinin ortalamalarını ve uç değerlerini ifade eden veriler, Tablo 4.3.'te görülmektedir.

Tablo 4.3. Bağlantı kurulum süresi ölçümleri

Bağlantı Kurulumu (sn.)	**FC**		**RC**		**PRC**		**SYM**	
	ICE	**Model**	**ICE**	**Model**	**ICE**	**Model**	**ICE**	**Model**
FC	16,23 ±1,58	2,18 ±0,11	15,42 ±0,98	2,78 ±0,28	12,5 ±0,41	3,24 ±0,3	13,79 ±1,46	4,15 ±0,25
RC	14,02 ±0,77	2,16 ±0,11	13,85 ±0,39	2,21 ±0,1	14,45 ±0,48	3,13 ±0,09	13,49 ±0,45	4,21 ±0,16
PRC	16,75 ±,45	2,12 ±0,19	13,45 ±1,165	2,18 ±0,155	18,73 ±0,7	3,44 ±0,14	26,01 ±2,16	7,01 ±0,13
SYM	23,45 ±1,34	4,12 ±0,25	20,45 ±0,92	4,63 ±0,125	19,95 ±1,44	7,29 ±0,09	21,43 ±1,08	7,45 ±0,14

Tablo 4.2'de sunulan 16 farklı senaryo için ICE ve SBN yöntemlerinde elde edilen bağlantı kurulum süreleri Şekil 4.15.'te görülmektedir. Grafikteki değerler bu çalışmanın başlıca amacına ulaşılıp ulaşılmadığını göstermesi açısından önemlidir. Bağlantı kurulum süresi açısından önerilen SBN yöntemin ICE yöntemine oranla tüm senaryolarda ortalama %77'lik önemli bir iyileşme sağladığı görülmektedir.



Şekil 4.15. Bağlantı kurulum süresi karşılaştırması

Bağlantı kurulum süresinin değerlendirildiği bu analizde, önerilen SBN yönteminin ICE yöntemine göre ortaya koyduğu başarımın temel nedeni olarak ICE yönteminin potansiyel tüm ağ arayüzleri, bağlanılabilirlik açısından taramasından kaynaklanmaktadır.

Ayrıca bağlantı biçimi olarak aynı yöntemin kullanıldığı relay bağlantı durumunda ICE ile SBN arasındaki bağlantı kurulum süresindeki farklılığın temel nedeni olarak SBN yönteminde tespit edilen üyelik çiftlerinden sadece relay adres çiftlerinin bağlantı kontrolü sürecine dahil edilmiş olmasıdır.

4.4.2. Paket kullanım sayısı

Bağlantılarda kullanılan kontrol paketlerinin sayısının azlığı bant genişliğinin efektif kullanımı açısından önemlidir. Özellikle STUN tabanlı NAT geçişi yöntemleri mevcut ağ arayüzlerinin tümü için ayrı ayrı kontrol paketleri ile bağlantı başarım kontrolleri yapmaktadır. Bu yaklaşım büyük sayıda paket kullanımı ortaya çıkarmaktadır. ICE yöntemi öncelikle alt yapısında STUN protokolünü kullanarak doğrudan bağlantı kurabilme potansiyeline sahip arayüzleri tanımlar.

Bu arayüzler Host, Server Reflexive ve Relayed Candidates olarak isimlendirilir.

Host arayüzleri istemcinin fiziksel ağ arayüzündeki bağlantıları ifade eder. Ancak bu arayüzler fiziksel yada VPN gibi sanal arayüzler olabilir. Aynı zamanda ağ bağlantısında kullanılmayan ve otomatik olarak üretilmiş pek çok arayüz tanımı olabilmektedir. Aslında bu tanımların pek çoğu gereksiz yere kontrol edilmektedir. Zaten iki istemci arasında el sıkışma süreçlerini gerçekleştirmek için bir randevu sunucusu üzerinde bağlantı kurmak zorunda olan istemciler bu kullanılabilir arayüzlere erişebilmektedir.

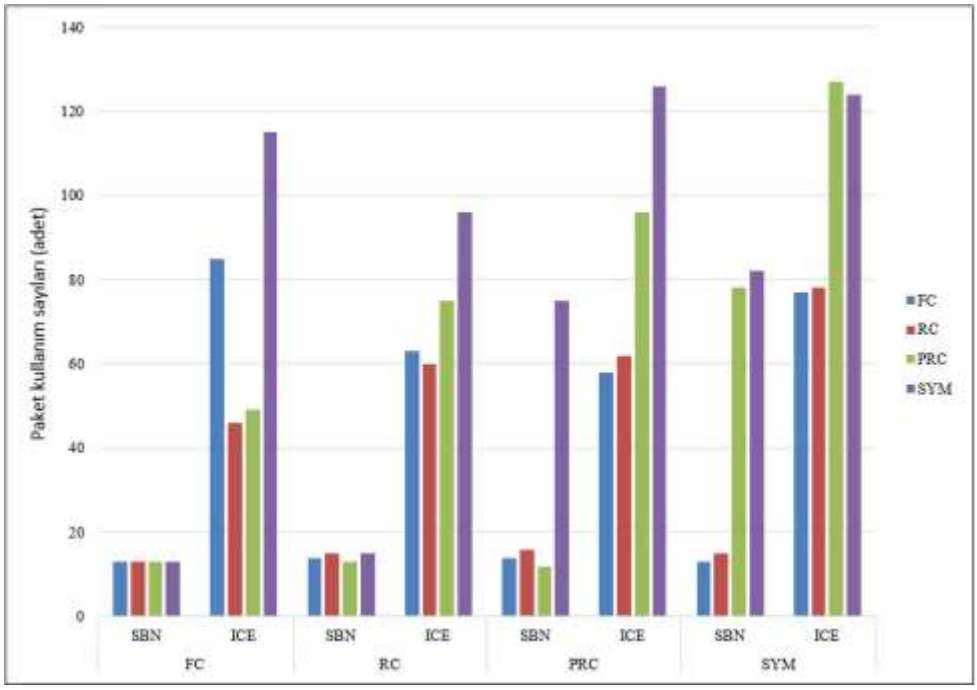
Server Reflexive adresler ise NAT arkasındaki istemcinin kamusal IP yada NAT arayüzünün IP'si ve NAT tarafından haritalanan kamusal UDP portu olarak tanımlanır. Bu arayüzler, STUN protokolünün de temelini oluşturan UDP kanal açma tekniğinde kullanılmaktadır. Ancak STUN sunucusu gerek RTT sürelerinin fazlalığı gerekse de fazlaca paket alışverişi gerektirmesi nedeniyle geliştirilen modelde kullanılmamıştır.

Son olarak tanımlanan Relay adresler istemciler arasında doğrudan bağlantının kurulamadığı durumlarda röle sunucuları üzerinden bağlantının haritalanması için kullanılmaktadır. ICE yönteminde olduğu gibi SBN yönteminde de relay adres çiftleri en düşük öncelik değerine sahiptir ve son aşamada bu arayüzler üzerinden bağlantı kurulumu denenir. Çünkü röle sunucuları doğrudan bağlantı kurulumlarına göre daha fazla oranda gecikmeye yol açacaktır.

Tablo 4.4. Paket kullanım sayısı ölçümleri

Paket Kullanımı (adet)	FC		RC		PRC		SYM	
	ICE	**Model**	**ICE**	**Model**	**ICE**	**Model**	**ICE**	**Model**
FC	85 ±20	13 ±2,5	46 ±7,5	13 ±2	49 ±11	13 ±2	115 ±12,5	13 ±2
RC	63 ±8,5	14 ±3	60 ±7,5	15 ±3	75 ±5	13 ±1,5	96 ±6,5	15 ±2,5
PRC	58 ±4,5	14 ±2	62 ±5,5	16 ±2	96 ±36	12 ±1	126 ±9	75 ±7,5
SYM	77 ±7,5	13 ±1,5	78 ±4	15 ±3	127 ±7	78 ±4,5	124 ±7	82 ±5,5

İzlenen performans kriterine özgü ölçüm değerlerinin ortalamalarını ve uç değerlerini ifade eden veriler, Tablo 4.4.'te görülmektedir. Önerilen SBN yöntemi ile ICE'nin bağlantı kurulumu için üretmiş olduğu kontrol paket sayıları Şekil 4.16.'da sunulmuştur.



Şekil 4.16. Paket kullanım karşılaştırması

Her iki yöntem için de kullanılan paket sayılarının belirlenmesi, bağlantı kurulum sürelerinin belirlendiği bölümde kullanılan yöntemle elde edilmiştir. Şekilde görüldüğü gibi örneğin Arayan ve Aranan istemcilerin FC NAT türünü kullanırken bağlantı modeli olarak ICE tercih edildiğinde kullanılan paket sayısı 76, SBN

yöntemi tercih edildiğinde ise 14 olarak gerçekleşmiştir. Bu farkın farklı NAT davranışlarında da benzer şekilde olduğu tespit edilmektedir.

SBN yönteminin, doğrudan bağlantı kurulabildiği senaryolarda ICE yöntemine oranla paket kullanım sayısında %66'lık bir iyileşme sağladığı görülmektedir. Geliştirilen SBN yöntemi sonucu elde edilen paket kullanım sayılarındaki iyileştirmeler, benzer şekilde bant genişliği kullanımı ve bağlantı kurulum zamanı parametrelerini doğrudan etkilemektedir.

4.4.3. Bant genişliği kullanımı

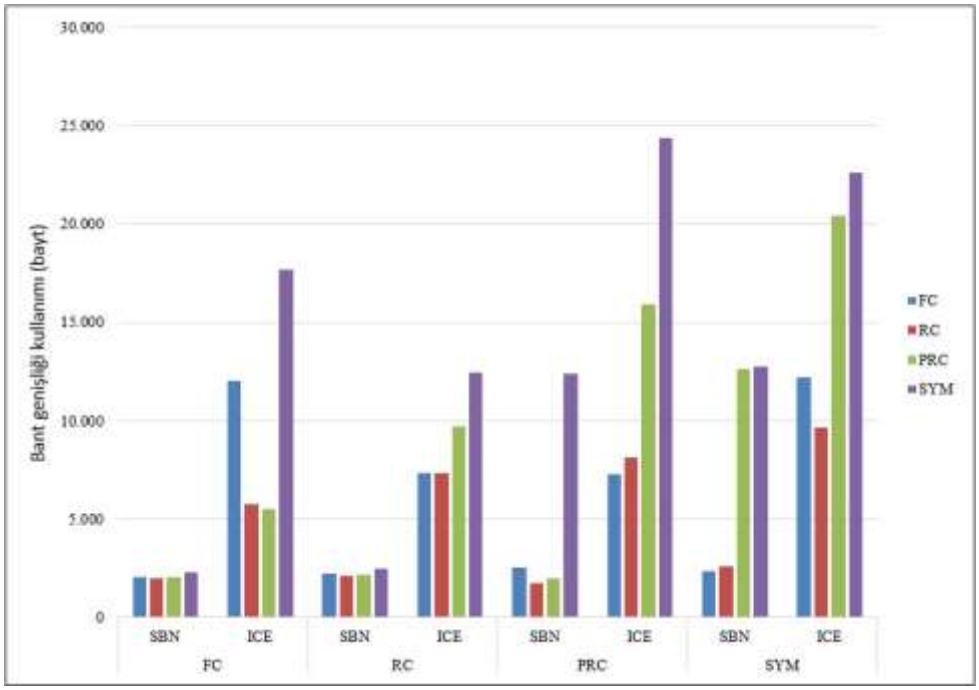
Bağlantı kurulumu için kontrol edilecek arayüz sayısının azaltılması kullanılan paket sayısını düşürmekte ve kullanılan bant genişliği oranları da bu durumlara bağlı olarak düşmektedir.

Tablo 4.5. Bant genişliği kullanımı ölçümleri

Band Genişliği (bayt)	FC		RC		PRC		SYM	
	ICE	**Model**	**ICE**	**Model**	**ICE**	**Model**	**ICE**	**Model**
FC	12015 ±552,5	2050 ±260	5790 ±284	1990 ±60	5546 ±813,5	2060 ±100	17660 ±1065	2280 ±170
RC	7340 ±196	2230 ±80	7345 ±130,5	2103 ±29	9685 ±70,5	2165 ±140	12432 ±555	2450 ±200,5
PRC	7265 ±175	2530 ±125	8110 ±324	1730 ±200	15928 ±7200	1992 ±70	24350 ±1990	12406 ±1035,5
SYM	12187 ±760	2358 ±97	9652 ±124	2620 ±125	20390 ±526,5	12634 ±457	22610 ±554	12725 ±301

İzlenen performans kriterine özgü ölçüm değerlerinin ortalamalarını ve uç değerlerini ifade eden veriler, Tablo 4.5.'te görülmektedir.

Şekil 4.17.'de ICE ve SBN yönteminin kullandığı bant genişliği değerleri sunulmuştur. Şekilde sunulan verilerin elde edilme yöntemi olarak bağlantı kurulum sürelerinin belirlenmesinde kullanılan yöntem kullanılmıştır. Kullanılan bant genişliği değerlerinde yaklaşık %59'luk bir azalma sağlanmış olması SBN yönteminin sağladığı önemli bir avantajdır.



Şekil 4.17. Bant genişliği kullanım karşılaştırması

4.5. Sonuç

P2P uygulamalarında düğümler arasında doğrudan bağlantı kurulumu bu tür uygulamalarda paylaşılan veri boyutu dikkate alındığında çok önemli bir gereksinimdir. Bu sorunun çözümüne yönelik geliştirilen ICE yöntemi önemli kazanımlar sağlamıştır. Ancak ICE NAT geçişi yöntemi, gerek servis kalitesi ve kullanıcı deneyimi açısından gerekse uygulamalarda zaman aşımı oluşturabilecek boyutlarda gecikmelerin ortaya çıkması gibi parametreler açısından değerlendirilmelidir. Bu sorunların giderilmesine yönelik geliştirilen SBN yöntemi, kullanıcı deneyimini ve servis kalitesini etkileyen bağlantı kurulum zamanı kriterine göre ICE yöntemine göre %77'lik bir performans artışı sağlamıştır. Benzer şekilde paket kullanım adetleri için ortalama %66'lık bir iyileşme getirdiği ve son olarak band genişliği kullanım değerleri incelendiğinde ortalama %59'luk bir düşüş sağladığı görülmektedir. Bu değerler de göstermektedir ki yapılan çalışma sonucunda yeni bir yöntem olarak önerilen SBN, hedeflenen verimlilik artışını sağlamanın yanında, geliştirilen uygulama ile de gerçek ortamda uçtan-uca tam bir uygulama modeli ortaya koymuştur.

BÖLÜM 5. SONUÇLAR VE DEĞERLENDİRME

Yapılan tez çalışması ile ilk olarak P2P bilgisayar ağları uygulamaları için önemli bir problem olan NAT geçiş yöntemleri kapsamında literatür çalışmaları ele alınmıştır. Yapılan değerlendirmeler sonucu mevcut NAT geçiş yöntemlerinin pek çoğu mevcut internet altyapısında değişiklik gerektirmesi, manuel yapılandırma ihtiyaçları ve uçtan uca tam bir model oluşturamamaları nedeniyle özellikle son kullanıcı noktasında yaygın kullanım şansı bulamamaktadır. Özellikle ALG, VPN, uPnP ve manual port yönlendirmesi gibi yöntemler belirtilen sorunlardan dolayı gerçek zamanlı iletişim altyapısı gerektiren uygulamalar için yazılım geliştiricilerince tercih edilmemektedir. Bunların dışında kalan STUN, TURN, ICE ve RTMFP gibi daha güncel yöntemler son zamanlarda yoğun kullanım ve araştırma alanı bulan yöntemler olarak dikkat çekmektedir.

Belirtilen yöntemlerden ICE ise STUN ve TURN protokolleri için bir çatı protokolüdür. İstemciler arasında öncelikle, alt yapısında UDP kanal açma tekniğini kullanan STUN protokolü üzerinden bağlantı kurulumu denenir. Bu aşamada STUN protokolü istemcilerin doğrudan bağlantı kurabilecekleri ağ arayüzlerinin tanımlanmasında kullanılmaktadır. Doğrudan bağlantı kurulumunun mümkün olmadığı durumlarda TURN server üzerinden elde edilen relay arayüzler devreye girmektedir. Ancak ICE, protokolü tanımladığı tüm bu potansiyel bağlantı arayüzleri için genel bir durum kontrolü yapmak yerine mevcut arayüzlerin tümü için başarılı bağlantı kurulumu yapılabilir olup olmadığını kontrol etmektedir. Bu yaklaşım örneğin 10 tane potansiyel bağlantı arayüzü tanımlanan iki istemci arasında tüm bağlantı çiftleri için çapraz kontrolleri gerektirir. Yapılan bu kontrol yöntem olarak, oluşturulmaya çalışılan soket üzerinden gönderilen isteğe cevap alınıp alınamadığının belirli bir zaman aşımı süresinde denetlenmesiyle gerçekleştirilir. Her arayüz için gerçekleştirilen bu bağlanılabilirlik kontrolü istemciler arasındaki

bağlantı kurulum zamanını, kontroller için ihtiyaç duyulan paket sayısını ve bant genişliği kullanımını doğrudan etkilemektedir. Yaygın kullanım alanı ve mevcut internet altyapısına uyumluluğu ile önemli bir ihtiyaca cevap vermekte olan ICE yönteminin belirtilen parametreler çerçevesinde iyileştirilmesi gerekmektedir.

Yapılan çalışmanın hedeflediği çıktılardan biri ICE yönteminde karşılaşılan bu dezavantajların iyileştirilmesi olmuştur. Bu iyileştirmelerin yapılabilmesi için Adobe firması tarafından geliştirilen ve yeni bir NAT geçiş yöntemi öneren RTMFP protokolü incelenmiştir. STUN protokolüne benzer bir altyapı üzerinde kurgulanan bu protokolün temel hedefi istemciler arasında verimli bir NAT geçişi sağlamaktır. Kullandığı yöntem olarak, Aktarım sunucusu olarak isimlendirilen ve düşük RTT süreleri ile çalışan sunucular üzerinden istemcilerin NAT arayüzlerini haritalamak ve UDP kanal açma tekniği ile NAT arkasına mevcut UDP arayüzler ile geçiş yapmaktır. Ancak RTMFP protokolü doğrudan bağlantı kurulamayan istemciler arasında bir NAT geçiş yöntemi önermemektedir. İki istemci arasında doğrudan bağlantı kurulamadığı durumda TCP tabanlı ve yüksek gecikme ile gerçek zamanlı uygulamalar için ciddi sunucu ve bant genişliği maliyeti getiren RTMP protokolünün kullanılmasını önermektedir [96]. RTMP protokolünün TCP tabanlı oluşu, tüm medya içeriğinin dağıtımının merkezi medya sunucuları üzerinden yapılıyor oluşu, yüksek gecikme süresi, merkezi bant genişliği ihtiyacı gibi maddi ve teknik zorluklar ortaya çıkmasına neden olmaktadır.

Çalışmanın birinci araştırma alanını oluşturan verimli bir NAT geçiş yönteminin belirlenmesinde mevcut yöntemlerin hiçbiri belirtilen problemleri tek başına çözmekte yeterli olamamaktadır. Ancak gerek ICE gerekse de RTMFP protokolünün önemli avantajları bulunmaktadır. Bu nedenle geliştirilen yeni NAT geçişi yönteminde mevcut protokollerin sağladığı avantajlar kurulan model için referans alınması gereken noktalar olarak belirlenmiştir. Bu yaklaşım sonucunda geliştirilen ve SBN olarak isimlendirilen yöntem için bir uygulama geliştirilmiştir. Bu uygulama gerçek ortamda denenmiş ve NAT geçişi yöntemi olarak ICE yöntemini kullanan Jitsi yazılımı ile karşılaştırılmıştır. Yapılan değerlendirmelerin birinci aşamasında öncelikle hedeflenen çıktı, SBN yönteminin, ICE yönteminde olduğu gibi tüm NAT

davranış tiplerinde bağlantı kurulum başarımını sağlayabilmesi olmuştur. Bu noktada UDP kanal açma tekniğinin başarısız olduğu istemcilerden herhangi birinin Port Restricted Cone NAT arkasında iken diğer istemcinin Symmetric NAT arakasında yada her iki istemcinin de Symmetric NAT arkasında olduğu durumlarda TURN sunucusu üzerinden elde edilen relay arayüzler kullanılmıştır. Ancak NAT arkasındaki istemcilerin doğrudan bağlantı kurabildikleri diğer durumlarda ICE'ın kullandığı tüm arayüzleri sırasıyla UDP kanal açma için denemek yerine açık kaynak kodlu RTMFP projesi olan Cumulus (Aktarım) sunucusu üzerinden elde edilen EPD verileri kullanılmıştır. Cumulus sunucusu gerek NAT arayüzlerinin haritalanmasında gerekse de RTT sürelerindeki verimlik noktasında STUN sunucularına göre daha verimlidir [32]. Dolayısıyla doğrudan bağlantı kurulumunun öncelikli olduğu bu tür uygulamalarda (pek çok ev kullanıcısı için doğrudan bağlantı kurulumu, kullanılan NAT tiplerinden dolayı mümkündür.) UDP kanal açma tekniğinin verimliliği sağlanmıştır. Sonuç olarak geliştirilen SBN yöntemi hedeflenen iyileştirmeleri sağlamanın yanında istemciler arasında bağlantı kurulum başarımı ile de en çok tercih edilen mevcut yöntemlerden biri olan ICE yöntemiyle aynı davranışları sergilemiştir.

Çalışmanın asıl hedefi P2P bilgisayar ağlarında çoklu ortam verilerinin iletimi için uçtan uca tam bir model önermektir. Bu hedef doğrultusunda istemciler arasında randevulaşma, müzakere ve oturum tanımlama bilgilerinin istemciler arasında el değişiminin sağlanmasından sorumlu istemci-sunucu mimarisine dayalı uygulama protokolü gerekmektedir. Bu uygulama protokolü pek çok anında mesajlaşma servisinin de uygulama protokolü olarak kullandığı XMPP olarak belirlenmiştir. Bu protokolün tercih edilmesinin nedenleri arasında, protokolün XML paket yapısına dayalı genişletilebilirlik sağlaması, neredeyse gerçek zamanlı bir istemci-sunucu iletişimi sunuyor olması, kendi içinde bağımsız ancak diğer XMPP otonom sistemleri ile entegre olabiliyor olmasının yanında pek çok istemci yazılımı ve uygulama geliştirme arayüzü sunması sayılabilir.

Çalışma sürecinde XMPP protokolünün gerek çekirdek tanımları gerekse de sonradan geliştirilen XEP olarak ifade edilen ilave protokolleri incelenmiştir.

Yapılan incelemeler sonucunda XMPP kullanıcıları arasında P2P iletişim için randevulaşma, müzakere ve oturum tanımlama bilgilerinin el değiştirmelerini sağlayan XEP-0167 ilave protokolünün geliştirildiği belirlenmiştir. Bu ilave protokol XMPP paket yapılarından bilgi/sorgu paketlerini özelleştirerek, istemciler arasındaki medya oturumu öncesinde gerekli veri akışını XMPP sunucusu üzerinden sağlamaktadır. Ancak bu ilave protokolde tasarlanan bilgi/sorgu paket yapısı yalnızca ICE yöntemi için bir haritalama sunmaktadır. Bu durum geliştirilen SBN yöntemi ile XMPP uygulama protokolü arasında bir uyumsuzluk oluşturmaktadır. Bu uyumsuzluğun giderilmesine ve XMPP ağındaki istemciler arasında gerçek zamanlı medya akış oturumu öncesinde gerekli bilgi transferlerinin sorunsuz sağlanabilmesi adına yeni bir özelleştirilmiş bilgi/sorgu paketinin tasarlanması gerekliliği ortaya çıkmıştır.

Önerilen SBN yönteminin ihtiyaç duyduğu oturum tanımlama bilgilerinin XMPP sunucusu üzerinden istemciler arasında el değişminin sağlanması için özelleştirilmiş bilgi/sorgu paketi tasarlanmıştır. Bu paket yapısı medya oturumu başlatmak isteyen kullanıcılara ait NAT arayüzlerini, UDP kanal açma tekniğinde kullanılacak EPD verilerini ve doğrudan bağlantı kurulamadığı durumlarda bağlantının röle sunucuları üzerinden yapılmasını sağlayan Relay arayüzlerini içermektedir. Özelleştirilmiş bilgi/sorgu paketi ile gerçek zamanlı medya oturumu başlatmak isteyen iki istemci arasındaki randevulaşma, müzakere ve oturum tanımlama bilgilerinin el değişimini sağlanmıştır.

Sonuç olarak yapılan bu tez çalışması ile internetin de temel tasarım felsefesi olan uçtan uca tam bir model ortaya konulmuştur. Bu model P2P bilgisayar ağlarında ortam verilerinin iletimi için NAT geçişi performansı ve randevulaşma süreçlerine önemli yaklaşımlar ve yenilikler getirmektedir. Yapılan çalışma sonucunda SBN olarak isimlendirilen yeni bir durum tabanlı NAT geçiş yöntemi önerilmesinin yanında, bu yöntemin bir uygulama protokolü üzerinde yürütülebilmesi için gerekli modeller de ortaya konulmuştur.

Yapılan bu tez çalışması ile gerek NAT geçişi süreçlerinin anlaşılması, gerek gerçek zamanlı iletişim ihtiyacı duyan yazılımlar için bir model önermesi, gerekse de pek çok uygulama tarafından altyapı protokolü olarak kullanılan XMPP protokolünün anlık mesajlaşma uygulamaları dışında başka çalışma alanlarında kullanımına yönelik fikirler ortaya çıkarabileceği düşünülmektedir.

5.1. Tartışma ve Öneriler

Çalışma sonucunda ortaya konulan SBN [106] yöntemi, NAT geçişi yöntemi olarak ICE protokolünü kullanan WebRTC [104] gibi HTML5 ile gündeme gelen yeni nesil teknolojilere uyarlanması üzerinde çalışmalar yapılabilir. Özellikle WebRTC ile gündeme gelen ve ICE yönteminin verimsiz kaldığı bağlantı kurulum süresi gibi parametrelerin iyileştirilmesini hedefleyen Trickle-ICE [101] modelinin HTML5 dışındaki uygulama mimarileri için de kullanılabilir hale getirilmesi önem arz etmektedir. Özellikle XMPP protokolü için geliştirilen XEP-0167 ilave eklentisi temelinde HTML5 ile birlikte gündeme gelen bu yeni yaklaşımın ilave protokol olarak XMPP protokolüne eklenmesi değerli bir çalışma alanı olacaktır. Özellikle mobil cihazların yaygınlaşması ve HTML5 ile birlikte gelen yeni teknolojilere olan ilginin artması neticesinde günümüz yazılım geliştiricileri için tarayıcı tabanlı bu teknoloji ile diğer platformların çapraz bir şekilde iletişim kurabilmeleri önemli bir yenilik oluşturacaktır.

Yapılan tez çalışmasında geliştirilen model için uygulama protokolü olarak kullanılan ve özelleştirilmiş bilgi/sorgu paket tanımları ile genişletilebilirliği vurgulanan XMPP protokolü, gerçek zamanlı altyapıya ihtiyaç duyan pek çok çalışma için kullanılabilir. Gerçekleştirilen çalışmalar sonucunda ortaya konulan özelleştirilmiş XML paket yapıları ve geliştirmeler XMPP ilave protokolü olarak XMPP standartları vakfına (XMPP Standards Foundation) kaydedilerek yaygınlaştırılması sağlanabilir. Bu protokolün, özellikle kurumsal uygulamalar için dağıtık ve gerçek zamanlı iş uygulamaları yaklaşımı getirebileceği öngörülmektedir. Pek çok kurumsal yapı için sorun teşkil eden kamusal anlık mesajlaşma servisleri bu altyapı üzerinden kurumsal hale getirilebilir ve kurumların bu altyapı üzerinden iş

yapış biçimleri kurgulanabilir. Neredeyse gerçek zamanlı bir XML paket iletimi sağlayan XMPP [105] protokolü benzer pek çok araştırma alanında kurgulanabilecektir.

KAYNAKLAR

[1] Oram, A., Peer to Peer: Harnessing the Power of Disruptive Technologies, O'Reilly & Associates, Inc., USA, ISBN: 0-596-00110-X, 2001.

[2] Flenner, R., Abbott, M., Boubez, T. Java P2P Unleashed, Sams Publishing, USA, ISBN: 0-672-32399-0, 2002.

[3] Moffitt, J., Professional XMPP Programming with JavaScript and jQuery, Wiley Publishing, Inc., USA, ISBN: 978-0-470-54071-8, 2010.

[4] Tarkoma, S., Overlay Networks Toward Information Networking, Auerbach Publications Taylor & Francis Group, USA, ISBN: 978-1-4398-1373-7, 2010.

[5] Rosenberg, J., Weinberger, J., Huitema, C., Mahy, R., STUN - Simple Traversal of User Datagram Protocol Through Network Address Translators, Internet Engineering Task Force, RFC: 3489, Mart 2003.

[6] Mahy, R., Matthews, P., Rosenberg, J., Traversal Using Relays around NAT: Relay Extensions to Session Traversal Utilities for NAT, Internet Engineering Task Force, RFC: 5766, Nisan 2010.

[7] Rosenberg, J., Interactive Connectivity Establishment: A Protocol for Network Address Translator Traversal for Offer/Answer Protocols, Internet Engineering Task Force, RFC: 5245, Nisan 2010.

[8] Thornburgh, M., Adobe's Secure Real-Time Media Flow Protocol, Adobe, Internet Engineering Task Force, RFC: 7016, Kasım, 2013.

[9] Parmar, H., Thornburgh, M., Adobe's Real Time Messaging Protocol, Adobe, Aralık, 2012.

[10] Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V., RTP: A Transport Protocol for Real-Time Applications, Internet Engineering Task Force, RFC: 3550, Temmuz, 2003.

[11] Simpson, W., Video Over IP IPTV, Internet Video, H.264, P2P, WebTV and Streaming: A Complete Guide to Understanding the Technology, Elsevier Inc., USA, ISBN:978-0-240-81084-3, 2008.

[12] Perkins, C., RTP: Audio and Video for the Internet, ISBN: 0-672-32249-8, Addison Wesley, 2003.

[13] Sadka, A. S., Compressed Video Communications, ISBN 0-470-84312-8, John Wiley & Sons, Ltd., UK, 2002.

[14] Firestone, S., Ramalingam, T., Fry, S., Voice and Video Conferencing Fundamentals, ISBN-13: 978-1-58705-268-2, Cisco Systems, Inc., USA, 2007.

[15] Richardson, I. E. G., H.264 and MPEG-4 Video Compression, ISBN:0-470-84837-5, John Wiley & Sons Ltd., UK, 2003.

[16] Sun, L., Mkwawa, I. H., Jammeh, E., Ifeachor, E., Guide to Voice and Video over IP For Fixed and Mobile Networks, ISBN: 978-1-4471-4905-7, Springer, UK, 2013.

[17] Perea, R. M., Internet Multimedia Communications Using SIP, ISBN: 978-0-12-374300-8, Morgan Kaufmann Publishers, USA, 2008.

[18] Agbinya, J. I., IP Communications and Services for NGN, ISBN: 978-1-4200-7090-3, Taylor and Francis Group, USA, 2010.

[19] Labovitz, C., Lekel-Johnson, S., Mcpherson, D., Oberheide, J., Jahanian, F., Internet Inter-Domain Traffic SIGCOMM, 2010.

[20] Harzog, B., Net Neutrality and the Cloud, http://www.virtualizationpractice.com/blog/?p=8794, Erişim Tarihi: 01.02.2014, 2001.

[21] White Papers, VNI, Cisco Visual Networking Index: Forecast and Methodology, http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html, Erişim Tarihi: 18.09.2014, Haziran 2014.

[22] White Papers, VNI, The Zettabyte Era – Trends and Analysis, http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/VNI_Hyperconnectivity_WP.html Erişim Tarihi: 18.09.2014, Haziran 2014.

[23] Perreault, S., NAT and Firewall Traversal with STUN/TURN/ICE, 2008-09-04, http://www.viagenie.ca/publications/, Erişim Tarihi: 25.12.2013.

[24] Chen, Y. C., Jia, W. K., Challenge and solutions of NAT traversal for ubiquitous and pervasive applications on the Internet, The Journal of

Systems and Software 82, p:1620–1626, 2009.

[25] Topal, C., Design and Implementation of a Border Router for Seamless Peer To Peer (P2p) Udp Communication Using IPv4+4 Addresses, Master of Science Thesis, Anadolu University, 2008.

[26] Topal, C., Akınlar, C., Secure seamless peer-to-peer (P2P) UDP communication using IPv4 LSRR option and IPv4+4 addresses, Computers and Electrical Engineering 35 p. 115–125, 2009.

[27] Wang, Y., Lu, Z., Gu, J., Research on Symmetric NAT Traversal in P2P Applications, Proceedings of the International Multi-Conference on Computing in fthe Global Information Technology, 2006.

[28] Zhang, Z., Wen, X., Zheng, W., A NAT Traversal Mechanism for Peer-To-Peer Networks, International Symposium on Intelligent Ubiquitous Computing and Education, 2009.

[29] Müller, A., Evans, N., Grothoff, C., Autonomous NAT Traversal, Peer-to-Peer Computing (P2P) IEEE Tenth International Conference, 2010.

[30] Tseng, C., Lin, C., Yen, L., Liu, J., Ho, C., Can: A context-aware NAT traversal scheme, Journal of Network and Computer Applications 36, p. 1164–1173, 2013.

[31] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E., SIP: Session Initiation Protocol, Internet Engineering Task Force, RFC: 3261, Haziran, 2002.

[32] Kaufman, M., RTMFP Overview for IETF77 TSV AREA, http://www.ietf.org/proceedings/10mar/slides/tsvarea-1.pdf, Erişim Tarihi: 20.11.2013.

[33] Sathiaseelan, A., Fairhurst, G., TCP-Friendly Rate Control (TFRC) for bursty media flows, Computer Communications 34, p. 1836–1847, 2011.

[34] Xue, L., Wen, F., Fan, C., Wang, J., Wang, X., Group Audio Application with Flash Multicast Streaming Based on RTMFP, The 2nd International Conference on Computer Application and System Modeling, Published by Atlantis Press, Paris, France, p. 41 – 44, 2012.

[35] Dwivedi, H., Hacking VoIP: Protocols, Attacks, and Countermeasures, ISBN-13: 978-1-59327-163-3, No Starch Press Inc, CA, USA, 2009.

[36] Costa, D. G., Fialho, S. V., A P2P Architecture to Support Mobile Real-Time Multimedia Communications, Journal of Multimedia, Vol. 5, No. 5, p. 514 – 521, 2010.

[37] Chen, T. H., Liu, S. C., Chen, J. H., Adaptive Receiver-Driven Approach in P2P Live Streaming Networks, Computers and Electrical Engineering 36, p. 1002 – 1013, 2009.

[38] Öztoprak, K., Hybrid CDN P2P Architecture For Multimedia Streaming, Doktora Tezi, Middle East Technical University, Ankara, Türkiye, 2008.

[39] Segui, F. B., Cebollada, J. C. G., Mauri, J. L., An RTP-RTCP baset approach for multimedia group and inter-stream synchronization, Multimed Tools Appl 40, p. 285 – 319, 2008.

[40] Burmeister, C., Hakenberg, R., Miyazaki, A., Ott, J., Sato, N., Fukunaga, S., Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback: Results of the Timing Rule Simulations, Network Working Group, RFC: 4586, Temmuz 2006.

[41] Khlifi, H., Gregoire, J. C., ARTP: A buffer-aware rate control protocol for media streaming, Computer Networks 51, p. 1601 – 1615, 2007.

[42] Granda, J. C., García, D. F., Nuño, P., Suárez, F.J., An efficient networking technique for synchronous e-learning platforms in corporate environments, Computer Communications 33, p. 1752 – 1766, 2010.

[43] Saint-Andre, P., Ed., Extensible Messaging and Presence Protocol (XMPP): Core, Network Working Group, RFC: 3920, Ekim 2004.

[44] Karapantazis, S., Pavlidou, F. N., VoIP: A comprehensive survey on a promising technology, Computer Networks 53, p. 2050 – 2090, 2009.

[45] Kim, J., Choi, H. J., Lee, S. H., Park, S. H., Song, M. S., Chang, H., Implementation of Quality of Service Control and Security Based on Real-Time Transport Protocol, Proceedings of IEEE IC-BNMT, p. 10 – 13, 2013.

[46] Liu, C., Wang, Y. K., Hannuksela M. M., Chen, Y., Sujeet, M., Gabbouj, M., RTP/AVPF Compliant Feedback for Error Resilient Video Coding in Conversational Applications, ISCIT 2009, IEEE, p. 218 – 223, 2009.

[47] Perkins, C., Westerlund, M., Multiplexing RTP Data and Control Packets on a Single Port, Internet Engineering Task Force, RFC: 5761, Nisan 2010.

[48] Pankaj, P., Hyde, M., Rodger, J. A., P2P Business Applications: Future and Directions, Communications and Network 4, p. 248 – 260, 2012.

[49] Ragavana, S. V., Kusnantoa, I. K., Ganapathyb, V., Service Oriented Framework for Industrial Automation Systems, Procedia Engineering 41, p. 716 – 723, 2012.

[50] Bønes, E., Hasvold, P., Henriksen, E., Strandenæs, T., Risk analysis of information security in a mobile instant messaging and presence system for healthcare, International Journal of Medical Informatics 76, p. 677 – 687, 2007.

[51] Sun, X., Du, Z., Chen, R., A Secure Cross-platform Mobile IM System for Enterprise Applications, 2011 International Conference on Uncertainty Reasoning and Knowledge Engineering, p. 158 – 161, 2011.

[52] Gomes, D., Gonçalvesy, J. M., Santosy, R. O., Aguiar, R., XMPP based Context Management Architecture, IEEE Globecom 2010 Workshop on Enabling the Future Service-Oriented Internet, p. 1372 – 1377, 2010.

[53] Lubke, R., Schuster, D., Schill, A., MobilisGroups: Location-based Group Formation in Mobile Social Networks, Second IEEE Workshop on Pervasive Collaboration and Social Networking, p. 502 – 507, 2011.

[54] Khan, A. A., Mouftah, H. T., Secured Web Services for Home Automation in Smart Grid Environment, 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2012.

[55] XEP-0166: Jingle, draft, v1.1, 2009-12-23, http://xmpp.org/extensions/xep-0166.html, Erişim Tarihi: 08.01.2014.

[56] XEP-0176: Jingle ICE-UDP Transport Method, draft, v1.0, 2009-06-10, http://xmpp.org/extensions/xep-0176.html, Erişim Tarihi: 08.01.2014.

[57] Egevang, K., Francis, P., The IP Network Address Translator (NAT), Network Working Group, RFC: 1631, Mayıs 1994.

[58] Srisuresh, P., Egevang, K., Traditional IP Network Address Translator (Traditional NAT), Network Working Group, RFC: 3022, Ocak 2001.

[59] Tanenbaum, A. S., Computer Networks, ISBN: 0-13-066102-3, Prentice Hall, 2003.

[60] Kuhn, D. R., Walsh, T. J., Fries, S., Security Considerations for Voice Over IP Systems, Recommendations of the National Institute of Standards and Technology, Special Publication 800-58, p. 54, Ocak 2005.

[61] Hain, T., Architectural Implications of NAT, Network Working Group, RFC: 2993, Kasım 2000.

[62] Cuevas, R., Cuevas, A., Cabellos-Aparicio, A., Jakab, L., Guerrero, C., A collaborative P2P scheme for NAT Traversal Server discovery based on topological information, Computer Networks 54, p. 2071 – 2085, 2010.

[63] Chen, W. E., Huang, Y. L., Chao, H. C., NAT Traversing Solutions for SIP Applications, EURASIP Journal onWireless Communications and Networking, Volume 2008, Article ID 639528, p. 1 – 9, 2008.

[64] Sinnreich, H., Johnston, A. B., Internet Communications Using SIP Delivering VoIP and Multimedia Services with Session Initiation Protocol, Second Edition, ISBN: 978-0-471-77657-4, Wiley Publishing Inc, USA, 2006.

[65] Boucadair, M., Inter-Asterisk Exchange (IAX) Deployment Scenarios in SIP-Enabled Networks, ISBN: 978-0-470-77072-6, John Wiley & Sons, Ltd, UK, 2009.

[66] UPnP forum, http://www.upnp.org, Erişim Tarihi: 10.12.2013.

[67] Wang, X., Research on P2P-SIP based VoIP system enhanced by UPnP technology, The Journal of China Universities of Posts and Telecommunications 17, p. 36 – 40, 2010.

[68] Ford, B., Srisuresh, P., Kegel, D., Peer-to-Peer Communication Across Network Address Translators, Proceedings of the 2005 USENIX Annual Technical Conference, Anaheim, CA, April, 2005.

[69] Srisuresh, P., Ford, B., Kegel, D., State of Peer-to-Peer (P2P) Communication across Network Address Translators (NATs), Network Working Group, RFC: 5128, Mart 2008.

[70] Rosenberg, J., Mahy, R., Matthews, P., Wing, D., Session Traversal Utilities for NAT (STUN), Network Working Group, RFC: 5389, Ekim 2008.

[71] Poikselka, M., Mayer, G., The IMS IP Multımedia Concepts and Services, Third Edition, ISBN 978-0-470-72196-4, John Wiley & Sons Ltd, UK, 2009.

[72] Kaufman, M., RTMFP Overview for IETF77 TSV AREA, Adobe Systems, 2009.

[73] Singh, K., P2P-SIP, Peer-to-peer Internet telephony using SIP, http://p2p-sip.blogspot.com.tr/2011/12/understanding-rtmfp-handshake.html, Erişim Tarihi: 02.11.2013.

[74] Hei, X., Liu, Y., Ross, K. W., IPTV over P2P Streaming Networks: The Mesh-Pull Approach, IPTV Systems, Standards and Architectures, IEEE Communications Magazine, p. 86-92, February 2008.

[75] Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V., RTP: A Transport Protocol for Real-Time Applications, Network Working Group, RFC: 1889, January 1996.

[76] Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V., RTP: A Transport Protocol for Real-Time Applications, Network Working Group, RFC: 3550, July 2003.

[77] Shigeoka I, Manning Publications Co, Instant Messaging in Java, 2002.

[78] Saint-Andre, P., Ed., Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence, Network Working Group, RFC: 3921, October 2004.

[79] Saint-Andre, P., Smith, K., Tronçon, R., XMPP: The Definitive Guide Building Real-Time Applications with Jabber Technologies, O'Reilly Media, Inc., April 2009.

[80] Ignite Realtime: Openfire XMPP Server, http://www.igniterealtime.org, Erişim Tarihi: 01.10.2015.

[81] Ejabberd: Robust, Scalable and Extensible XMPP Server, https://www.ejabberd.im/, Erişim Tarihi: 01.10.2015.

[82] Jitsi Meet - Web Conferences, https://jitsi.org/, Erişim Tarihi: 01.10.2015.

[83] Adium is a free instant messaging application, https://adium.im/, Erişim Tarihi: 01.10.2015.

[84] Pidgin, the universal chat client, https://pidgin.im/, Erişim Tarihi: 01.10.2015.

[85] Shigeoka I, Manning Publications Co, Instant Messaging in Java, 2002.

[86] Bray, T., Paoli, J., Sperberg-Mcqueen, C., Maler, E., Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C Recommendation, 26 November 2008.

[87] Alvestrand, H., IETF Policy on Character Sets and Languages, Best Current Practice 18, RFC 2277, January 1998.

[88] Alvestrand, H., Tags for the Identification of Languages, Best Current Practice 47, RFC 3066, January 2001.

[89] Saint-Andre, P., Extensible Messaging and Presence Protocol (XMPP): Core, Standards Track, RFC 6120, March 2011.

[90] Saint-Andre, P., Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence, Standards Track, RFC 6121, March 2011.

[91] Saint-Andre, P., Hildebrand, J., XEP-0184: Message Delivery Receipts, Standards Track, 2011-03-01.

[92] Handley, M., Jacobson, V., Perkins, C., SDP: Session Description Protocol, Standards Track, RFC 4566, July 2006.

[93] Saint-Andre, P., XEP-0001: XMPP Extension Protocols, Procedural, 2010-03-10.

[94] Ludwig, S., Saint-Andre, P., Egan, S., Mcqueen R., Cionoiu, D., XEP-0167: Jingle RTP Sessions, Standards Track, 2009-12-23.

[95] Cumulus: A complete open source and cross-platform RTMFP server, https://github.com/OpenRTMFP/Cumulus, Erişim Tarihi: 16.12.2014.

[96] Ma K.J., Bartos R., Bhatia S. A., Survey of Schemes for Internet-Based Video Delivery, Journal of Network and Computer Applications, 34, 1572-1586, 2011.

[97] Wireshark: Network protocol analyzer, http://www.wireshark.org, Erişim Tarihi: 25.04.2014.

[98] Smack API: Open Source XMPP (Jabber) client library for instant messaging and presence, http://www.igniterealtime.org/projects/smack/, Erişim Tarihi: 20.05.2014.

[99] ice4j: A Java implementation of the ICE protocol, http://code.google.com/p/ice4j/, Erişim Tarihi: 21.12.2013.

[100] TurnServer: This project open-source TURN server implementation, http://turnserver.sourceforge.net/, Erişim Tarihi: 21.12.2013.

[101] Ivov, E., Rescorla, E., Uberti, J., Trickle ICE: Incremental Provisioning of Candidates for the Interactive Connectivity Establishment (ICE) Protocol, http://tools.ietf.org/pdf/draft-ietf-mmusic-trickle-ice-02.pdf, Erişim Tarihi: 20.03.2015.

[102] Grigorik, I., High-Performance Browser Networking In: WebRTC, 1st ed. O'Reilly Media Inc, 309-362, 2013.

[103] WebRTC: Real-Time Communications (RTC) capabilities via simple APIs, http://www.webrtc.org, Erişim Tarihi: 01.04.2015.

[104] Arslan H., Tüncel S., Yüksek, A.G., Comparison of the Web Based Multimedia Protocols for NAT Traversal Performance, 2015 23nd Signal Processing and Communications Applications Conference (SIU), DOI: 10.1109/SIU.2015.7129979, Sayfa: 915-918, Mayıs 2015.

[105] Arslan H., Tüncel S., Gün, O. Extensible Messaging and Presence Protocol's Adaptation to Business Applications, 5th World Conference on Innovation and Computer Sciences - INSODE-2015, Mayıs 2015.

[106] Arslan H., Tüncel S., A New State-Based Connectivity Model for Peer-to-Peer Networks, IEICE Transactions on Information and Systems, DOI: 10.1587/transinf.2015EDP7220, Vol.E99-D, No:3, Mar. 2016.

ÖZGEÇMİŞ

Halil Arslan, 10.12.1982'de Sivas'ta doğdu. İlk, orta ve lise eğitimini Sivas'ta tamamladı. 2006 yılında Sakarya Üniversitesi Bilgisayar Sistemleri Öğretmenliği bölümünü bitirdi. 2006 yılında başladığı Sakarya Üniversitesi, Fen Bilimleri Enstitüsü Elektronik ve Bilgisayar Eğitimi Ana Bilim Dalı Yüksek lisans eğitiminden 2008 yılında mezun oldu. 2004-2009 yılları arasında Sakarya Üniversitesi Bilgi İşlem Dairesi Başkanlığı'nda web yazılım uzmanı olarak görev yaptı. Bu süre içerisinde görev yaptığı kurumun web yazılımları başta olmak üzere, akademik ve öğrenci işleri temelli uygulamalarının geliştirilmesinde görev aldı. Şu anda Cumhuriyet Üniversitesi, Sivas Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü'nde öğretim görevlisi olarak görev yapmaktadır.