

# Mobil Cihazlarda WebRTC Teknolojisi

## WebRTC Technology for Mobile Devices

Dr. Cengiz Toğay  
NETAŞ  
İstanbul, Türkiye  
ctogay@netas.com.tr

**Özetçe—** WebRTC, Web dünyasına entegre, standartlara dayanan, cihaz ve platformdan bağımsız gerçek zamanlı iletişim teknolojisidir. Teknoloji ilk olarak masaüstü sistemler için geliştirmekle birlikte mobil ortamlar giderek önem kazanmaktadır. Android işletim sistemi için Chrome ve Firefox tarayıcıları desteklemeye başlamıştır. Ayrıca, IOS ve Android işletim sistemlerinde WebRTC destekli doğal uygulamalar geliştirilebilmektedir. Bu çalışmada, WebRTC ile ilgili araştırma konuları ve test sonuçları sunulmaktadır.

**Anahtar Kelimeler —** Mobil, WebRTC

**Abstract—** WebRTC is a standardized, device and platform independent real-time communication technology which is integrated to web-world. As a first step, the technology is implemented for desktop systems and getting focus for mobile environments. Chrome and Firefox already support WebRTC on Android operating system. Also, WebRTC based native applications can be developed, in IOS and Android operating systems. In this study, WebRTC related research opportunities and test results are presented.

**Keywords —** Mobile, WebRTC

### I. GİRİŞ

obil akıllı cihazlar, 3G/4G teknolojileri ile çok daha hızlı İnternet kaynaklarına erişme imkanına sahip olmaktadır. İnternet hızındaki artışla birlikte, IP tabanlı iletişim günümüzde çeşitli uygulamaları (Skype, Google Talk, vb.) günlük hayatımızda görülmeye başlanmıştır. Bu konuda çıkan son teknolojilerden olan WebRTC (Web browsers with Real-Time Communications) [1][2], içerdği standartlar aracılığı ile NAT arkasındaki cihazlarda dahil olmak üzere bir dereceye kadar güvenli iletişim sunabilmektedir. WebRTC, tarayıcılar üzerinde HTML5 teknolojisi ile uyumlu arayüzler aracılığı ile herhangi eklenti ya da uygulama kurulmasına gerek kalmaksızın sesli/görüntülü görüşmenin yanı sıra veri paylaşımına imkan vermektedir. WebRTC teknolojisinde medya şifreli bir şekilde iletilmektedir. Teknolojide iletişimin kurulmasına yönelik tüm sinyalleşme faaliyetleri uygulamalara bırakılmaktadır. Bu şekilde uygulama ve sinyalleşmeden bağımsız olarak medyanın taraflarca

paylaşımı mümkün olmaktadır. WebRTC'yi diğer iletişim teknolojilerinden ayıran temel özellikleri; ücretsiz kodlara sahip olması, açık kaynak kodlu proje olması, Chrome ve Firefox tarafından destekleniyor olması, cihaz ve platformdan bağımsız iletişime imkan vermesi ve az sayıdaki JavaScript komutları ile Web tabanlı iletişimin kurulmasına olanak tanımasıdır. Medyanın WebRTC teknolojisi ile paylaşılmasında geliştirilecek çözümler, tarayıcılar üzerinde geliştirilecek web uygulamaları ya da doğrudan teknolojiyi bir kütüphane olarak içeren doğal uygulamalar şeklinde tanımlanabilir. Şu an hali hazırda, Microsoft Windows, Linux ve MacOS masaüstü işletim sistemlerinde yer alan Chrome ve Firefox tarayıcılarında WebRTC desteği bulunmaktadır. Ayrıca, eklenti (plugin) aracılığı ile İnternet Explorer'da da WebRTC uygulamaları kullanılabilmektedir. Mobil internet tarayıcılardan Chrome versiyon 29 ve Firefox versiyon 24, Android işletim sistemlerinde WebRTC desteğini sunmaktadırlar. IOS ortamında ise WebRTC destekli doğal uygulamalar geliştirilebilmektedir. 2016 yılı sonunda 3.5 milyar cihazın WebRTC desteğine sahip olması beklenmektedir [3]. Her ne kadar WebRTC ile internet ortamında iletişim hedefleniyor olsa da geleneksel ağlarda yer alan istemciler ile Ağ Geçitleri (Örn. SPiDR ve Mavenir WebRTC Gateway) aracılığı ile iletişim kurulabilmektedir. Dolayısı ile internet ortamında bir Web uygulamasından ya da WebRTC destekli uygulamadan, sabit telefon ya da VoIP telefonlar ile iletişim kurulması mümkündür. Bu çalışmada, WebRTC'nin temel özelliklerinin yanı sıra mobil dünyasında yer alması beklenen WebRTC'nin teknolojisinin önündeki temel problemler ve araştırma konularına değinilmektedir.

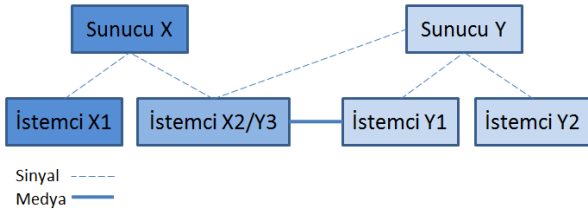
### II. WEBRTC

WebRTC (Web browsers with Real-Time Communications), mevcut standartları tarayıcı dünyasındaki üretici firmaların kabul ettiği bir teknoloji olarak bir araya getirmektedir [2]. Teknolojinin temel amacı, internet ortamında gerçek zamanlı iletişimin sağlanması amacıyla medyanın güvenli bir şekilde aktarılmasıdır. Medya ses, görüntü ya da veri paketlerinden oluşabilir. WebRTC kütüphanesinde her bir medya kendi Secure Real Time Transport Protocol

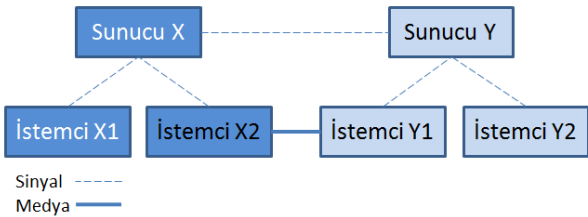
(SRTP) [4] ve SRTP Control Protocol (SRTCP) [4] portlarından aktarılabileceği gibi tüm medya çeşitleri ortak SRTP ve SRTCP portlarından da aktarılabilir. Gerçek zamanlı görüntü aktarımının yanı sıra Adobe Flash platformuna alternatif olarak değerlendirmek üzere videoların paylaşılmasında da kullanılması konusunda çalışmalar bulunmaktadır [5].

#### A. WebRTC'de Sinyalleşme

WebRTC, tamamen medyanın taşınması üzerine odaklanmaktadır. Sinyalleşmenin nasıl yapılacağı konusu uygulama geliştiricilerine bırakılmaktadır. Sinyalleşme WebRTC açısından: 1) kullanıcıların bir sunucuda yetkilendirilmelerini, 2) çağrı başlatma: WebRTC kütüphanesi aracılığı ile medya transferi için sadece kütüphane tarafından oluşturulan SDP paketlerinin karşı tarafa gönderilmesi 3) çağrı bilgilendirilmesini yakalama ve gelen SDP paketinin WebRTC kütüphanesine iletilmesi şeklinde sınıflandırılabilir. Kütüphane SDP paketleri aracılığı ile pakette yer alan bilgilerden yola çıkarak iletişimi başlatabilmektedir. Sinyalleşmede kullanılabilecek sunuculara SPiDR, Mavenir WebRTC Gateway verilebilir. Çağrı başlatma amacı ile istemcilerin sunuculara bağlanmasında, SIP, REST, WebSocket, XMPP protokolleri kullanılabilir. Çağrılar istemcilere bildirilmesinde genellikle Longpolling ve WebSocket teknolojilerinden faydalanılmaktadır.



Şekil 1. İki farklı uygulamaya ait istemcilere ait medya trafiği (İstemci X2 aynı zamanda Y3 istemcisi özelliğine sahip)



Şekil 2. İki farklı uygulamaya ait istemcilere ait medya trafiği (Sunucular birbirleri ile bağlantılı)

Şekil 1 ve 2'de iki farklı uygulamaya ait istemcilerin birbirleri ile iletişimine ait blok diyagramlar yer almaktadır. İstemciler ile sunucular arasında sinyalleşme gerçekleşirken medya aktarımında WebRTC Kütüphanesi kullanılmaktadır. Böylece uygulamadan bağımsız olarak istemciler arasında medya transferi gerçekleştirilebilir. Şekil 1'de yer alan mimaride X2 istemcisi aynı zamanda X ve Y uygulamalarının da istemcisi rolündedir. İstemci

X2/Y3, Y uygulamasına ait istemciler ile Sunucu Y aracılığı ile sinyalleşme işlemini gerçekleştirip iletişim medya trafiği başlatabilir. Şekil 2'de yer alan mimaride ise sunucular arasında alan tabanlı yönlendirme söz konusudur. İstemci X2 kendi sunucusu aracılığı ile Sunucu Y ve sonrasında istemci Y1 ile iletişim kurabilir. WebRTC'nin sağladığı standartlar ile uygulama, işletim sistemi ve cihaz bağımsız iletişim kurulabilmesi mümkündür. WebRTC kütüphanesi, web arayüzünde az sayıdaki JavaScript komutu ile uygulama geliştirmeye izin verirken kütüphane arayüzünü kullanan uygulamalar daha derin detayda bilgiye ihtiyaç duyulmasına neden olmaktadır [6]. Bu şekilde isteğe bağlı uygulama konusunda ilerleme sağlanırken konu ile ilgili uzmanlığa olan gereksinimi arttırmaktadır. Bu nedenle, özellikle sinyalleşmeyi gerçekleştirecek sunucular da var ise, sinyalleşme aktivitelerinin yanı sıra WebRTC arayüzünün de uygulama geliştiricilerinden saklanması durumunda, birkaç günlük çalışma ile mevcut uygulamaya WebRTC teknolojisi ile sesli ve görüntülü görüşme imkanı kazandırılmaktadır. WebRTC mobil cihazlarda kullanılmaya başlanması ile birlikte masaüstü cihazlarda yaşanmayan bazı sorunların ortaya çıkmaktadır. Bunlar arasında mobil cihazların işlemci gücü ve kablosuz bağlantıdan kaynaklı ağ iletişim problemleri sayılabilir. WebRTC 200 milisaniyeden fazla ağ kaynaklı gecikmenin olması durumunda bant genişliği kullanımında özellikle görüntülü iletişimde problemler yaşanmaktadır [7]. Mobil cihazları, sahip oldukları "Fast Dormancy"[8][9] özelliği ile kablosuz modemlerde bekleme moduna geçerek bağlantının kopmasına neden olabilmektedir. Bekleme moduna geçme süresi cihazdan cihaza değişmekle birlikte Web Socket/Longpolling bağlantılarında PING/PONG ya da timeout sürelerinin fazla belirlenmesi durumunda bildirimler kullanıcıya erişilememesi durumları ortaya çıkmaktadır. Sürenin kısa tutulması ise özellikle internet bağlantılarının aktif olarak kullanıldığı durumlarda cihazın pilinin gereğinden hızlı tükenmesine neden olmaktadır. Fast Dormancy özelliği belirli servis sağlayıcılar tarafından desteklenmekte ve internet bağlantısını nadir kullanan kullanıcılar için avantajlar sunmaktadır.

#### B. WebRTC'de Yer Alan Temel Standartlar ve Yöntemler

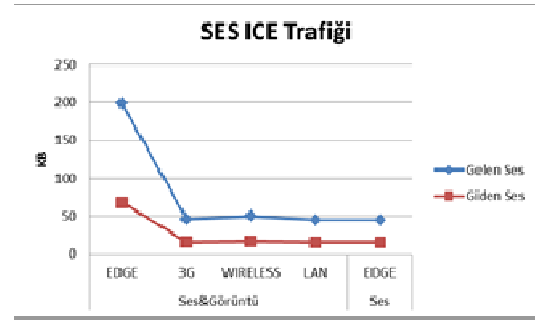
Medya, ağ üzerinde UDP paketleri üzerinde SRTP ile hedefe iletilmektedir. Paketlerin şifrelenmesinde AES 128-bit şifreleme anahtarı ve 80-bit yetkilendirme etiketi kullanılmaktadır. Medyanın güvenli bir şekilde hedefe ulaştırılmasında medyanın şifrelenmesinde kullanılacak anahtarın da güvenli bir şekilde hedefe iletilmesi esastır. Anahtarın aktarılmasında Source Description RTP Packet (SDPS) [10] ya da Datagram Transport Layer Security (DTLS) [11] metodları kullanılmaktadır. SDPS metodunda anahtar doğrudan sinyalleşme kanalından aktarılmaktadır. Sinyalleşmenin güvenliğinin sağlanması medyanın güvenliği açısından önem arz etmektedir. DTLS

metodunda, sinyalleşme kanalından üretilen sertifikanın parmak izi gönderilirken, medya için açılan portlardan sertifika gönderilir. Medya kanalından gelen sertifikanın, sinyalleşme tarafından gelen parmak izi ile uyumlu olması durumunda SRTP şifrelemesi için üretilen anahtar paylaşılır. Gerçek zamanlı medya aktarımında paket kayıpların bağlı olarak ses ve görüntüde problemlerin yaşanması muhtemeldir. Buna önlem olarak Real Time Transport Protocol (RTCP)-based feedback (RTCP/AVPF) [12] ile paket kayıp istatistiklerinin gönderen tarafa iletilmesi ile kayıp oranının azaltılması ve mümkünse kayıp paketin tekrar üretilmesi suretiyle yerine konması sağlanır. Alternatif olarak ileriye doğru hata kontrol mekanizmaları kullanılabilir [13]. Interactive Connectivity Establishment (ICE) [14]: İstemci bir iletişim başlatmak amacıyla STUN/TURN[15] sunucuları aracılığı ile elde ettiği IP/PORT bilgilerini birer aday olarak hedef kullanıcıya iletir. Hedef istemci, kullanıcıya erişebileceği IP/PORT ikililerini belirli bir önceliklendirme mekanizması aracılığı ile deneyerek kaynak istemciye (TURN var ise) NAT arkasında olsa dahi ulaşma imkanına kavuşur. Ayrıca, ICE Connectivity Check (STUN) mesajları belirli aralıklarla (tarayıcılar ve sürümleri arasında farklılık göstermekte) bağlantının NAT cihazlarında aktif tutulması için hedefe gönderilir.

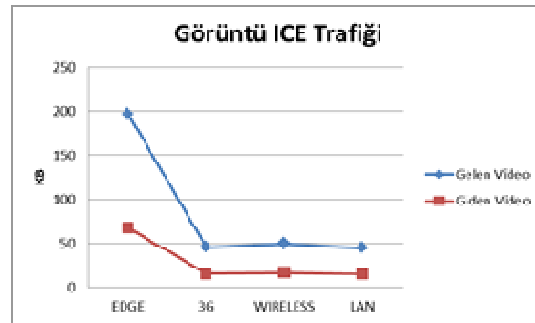
### III. SAYFA DÜZENİ VE BİÇİM

WebRTC ilk etapta kişisel bilgisayarlar arasında iletişimin kurulmasını sağlamakla birlikte 2014 yılının ilk çeyreğinden itibaren mobil dünyada da yer almayı hedeflemektedir. Ancak, mevcut kütüphanenin ihtiyaç duyduğu işlemci gücü ve ağ bağlantı gereksinimlerine bağlı olarak özellikle görüntülü görüşmeler için kodelerde (VP serisi) ve medya aktarım hata kontrol mekanizmaları konusunda geliştirmelere ihtiyaç bulunmaktadır. Medyanın yönlendirilmesinde (relay) kullanılan TURN sunucularının hız ve kapasiteleri; UDP paketlerinin gecikmesine ve kayıp olarak değerlendirilmesine neden olabilmektedir. Kayıp paketlere çözüm olarak sunulan hata kontrol sistemleri ise ağı ve işlemcinin meşgul olması, ses/görüntüde gürültü ve cihazın piline olumsuz yansımaları olmaktadır. Çift yönlü iletişimin (çağrı başlatma ve gelen çağrının bildirilmesi) gereği olarak bağlantının kurulması ve açık bırakılması enerji tüketimine olumsuz etkileri bulunmaktadır. Çağrı bildirimlerinin alınmasında Longpolling yönteminin kullanılması durumunda mesajların içerdiği başlık bilgileri ek veri trafiğine neden olmakla birlikte belirli aralıklarla bağlantı kurulumuna ihtiyaç bulunmaktadır. Mobil cihazlar için işletim sistemlerine özel etkin çözüm olarak PUSH mekanizmaları (Apple Push Notification (APNS) ve Android Push Notification (C2DM)) kullanılabilir. PUSH teknolojisinin kullanımında sunucuların güvenilir olması ve bilgilendirme mesajlarını birkaç saniyede cihaza

ulaştırması beklenmektedir. ICE Connectivity Check (STUN) mesajları NAT'lar üzerinde açılan portların sürekliliğinin sağlanması amacı ile atılmaktadır. Chrome'un 32.0.1700.76 nolu versiyonunda sesli ve görüntülü iletişim için (ses ve görüntü ayrı port) saniye de yaklaşık 2240 bytelık bir trafiğe neden olmaktadır. Günde 60 dakikalık görüşmenin yapıldığı bir ortamda sadece ICE STUN mesajları için ayda yaklaşık 230 MB'lık trafik ortaya çıkmaktadır. Trafik gerek enerji tüketimi gerekse de veri hattına sahip kullanıcılar için yüksek maliyetlere neden olacağı açıktır. 3G/4G veri planına sahip istemci uygulamaları için bu konuda araştırma ve geliştirme çalışmalarına ihtiyaç bulunmaktadır. Uygulanabilecek çözümler arasında, bağlantının sürdürüldüğü ortamda ICE Connectivity Check mesajlarına olan ihtiyacın giderilmesine yönelik çalışmalar bulunmaktadır. 3G/4G hizmetini vermekte olan operatörler, ağlarında çalışacak WebRTC uygulamaları için NAT cihazlarında ve istemci WebRTC kütüphanelerinde yapılacak düzenlemeler ile STUN paketlerinin gönderilmesi engellenebilir.



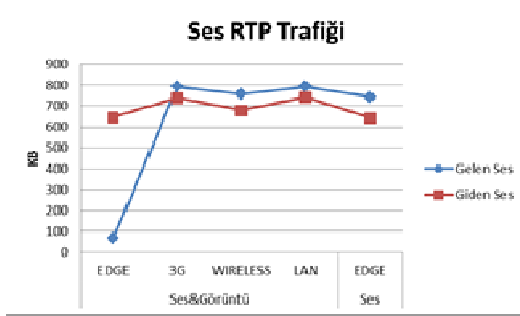
Şekil 3. Ses iletişimine ait ICE trafiği



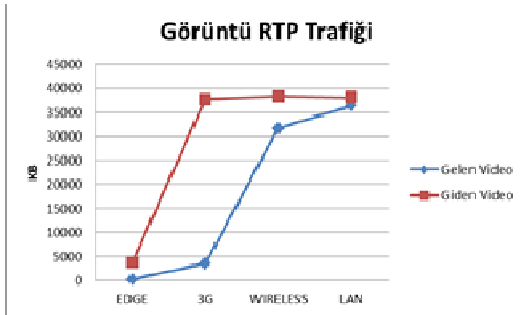
Şekil 4. Görüntü iletişimine ait ICE trafiği

WebRTC istemcilerinin farklı ağ bağlantılarındaki durumu her biri için yapılan 5 farklı sesli ve görüntülü 180 saniyelik görüşme ile ölçülmüş ve aritmetik ortalamaları alınmıştır. Görüşme bir kablosuz bağlantı ile ağına bağlı masaüstü bilgisayar ile farklı ağlara bağlı bir dizüstü bilgisayar arasında gerçekleştirilmiştir. EDGE ve 3G bağlantısı bir telefonun internet paylaşımı ile sağlanmıştır. İletişimdeki kopmalar 180 saniye şeklinde normalleştirilmiştir. İletişimde ses kodeği olarak ISAC

16000 ve görüntü kodeği olarak VP8 kullanılmıştır. Medya yerel TURN sunucusu üzerinden geçirilmiş ve ölçümler TURN sunucusu üzerinde alınmıştır. Gelen ICE paketlerine (her biri 128 byte) karşılık cevap ICE paketleri (her biri 44 byte) gönderilmektedir. ICE paketleri bağlantı kontrolü için kullanılmakta ve bağlantının zayıf olduğu durumlarda normalin 4 katı bir ICE trafiğine neden olduğu Şekil 3 ve Şekil 4'te gösterildiği üzere gözlemlenmiştir. EDGE ağı ile yapılan sesli ve görüntülü testlerde bağlantının ICE paketlerine cevapların zamanında alınamaması nedeni ile ortalama 63 saniye sonrasında koptuğu gözlemlenmiştir.



Şekil 5. Ses iletişimine ait medya trafiği



Şekil 6. Görüntü iletişimine ait medya trafiği

Şekil 5 ve 6'da gösterildiği üzere EDGE ağına bağlı istemci sesli ve görüntü iletişim kurmaya çalıştığında normalin çok altında ses ve trafiğini TURN sunucusuna iletebilmiştir. Bu da karşı tarafta iletişimin anlaşılamamasına neden olmuştur. Şekil 5'te gösterildiği üzere sadece ses için yapılan EDGE bağlantısında medyanın normal şekilde aktarıldığı gözlemlenmiştir. Şekil 6'ta yer alan 3G bağlantısında görüntülü ve zaman zaman görüntüde yaşanan donmalar ile birlikte sesli ve görüntülü görüşme yapılabilmektedir. Wireless (kablolu) ve LAN (yerel ağ) bağlantılarında ise oldukça başarılı iletişim kurulabilmektedir. 180 saniyelik bir görüntülü görüşmede yaklaşık olarak 36MB'lık bir medya trafiğine ihtiyaç duyulmaktadır. Görüldüğü üzere mobil cihazlarda EDGE ağlarında sesli, diğer ağlarda ise sesli ve görüntülü iletişim mobil cihazlar için mümkün görülmektedir.

Ancak, mobil cihazların işlemci gücü ve pil tüketim durumu çeşitli kodlar için incelenmelidir.

#### IV. BİLGİLENDİRME

Bu bildiride sunulan çalışma, TÜBİTAK tarafından desteklenen ve Netaş bünyesinde geliştirilen 1130066 nolu proje kapsamında yapılan çalışmalarının bir sonucu olarak ortaya çıkmıştır. İbrahim Açıkgöz'e test ortamının hazırlığındaki desteğinden dolayı teşekkür ederiz.

#### KAYNAKÇA

- [1] A. Bergkvist, D. Burnett, C. Jennings, and A. Narayanan, "WebRTC 1.0: Real-time Communication Between Browsers," *WebRTC 1.0: Real-time Communication Between Browsers*, 10-Sep-2013. [Online]. Available: <http://www.w3.org/TR/webrtc/>.
- [2] S. Loreto and S. P. Romano, "Real-Time Communications in the Web: Issues, Achievements, and Ongoing Standardization Efforts," *Internet Comput. IEEE*, vol. 16, no. 5, pp. 68–73, 2012.
- [3] Disruptive Analysis, "WebRTC Market Status & Forecasts," *WebRTC Market Status & Forecasts*. [Online]. Available: <http://disruptive-analysis.com/webrtc.htm>.
- [4] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman, *The Secure Real-time Transport Protocol (SRTP)*. IETF, 2004.
- [5] J. Muranyi and I. Kotuliak, "Identity management in WebRTC domains," in *Emerging eLearning Technologies and Applications (ICETA), 2013 IEEE 11th International Conference on*, 2013, pp. 289–293.
- [6] L. Lopez Fernandez, M. P. Diaz, R. Benitez Mejias, F. J. Lopez, and J. A. Santos, "Catalysing the success of WebRTC for the provision of advanced multimedia real-time communication services," in *Intelligence in Next Generation Networks (ICIN), 2013 17th International Conference on*, 2013, pp. 23–30.
- [7] V. Singh, A. Abello Lozano, and J. Ott, "Performance Analysis of Receive-Side Real-Time Congestion Control for WebRTC," in *Packet Video Workshop (PV), 2013 20th International*, 2013, pp. 1–8.
- [8] Nokia Siemens Networks Smart Labs, "Understanding smartphone behavior in the network," 2011.
- [9] "Fast Dormancy Best Practises Version 1.0," GSM Association, Jul. 2011.
- [10] F. Andreasen, M. Baugher, and D. Wing, *Session Description Protocol (SDP) Security Descriptions for Media Streams*. IETF, 2006.
- [11] E. Rescorla and N. Modadugu, *Datagram Transport Layer Security*. IETF, 2006.
- [12] J. Ott, S. Wenger, N. Sato, C. Burnmeister, and J. Rey, *Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)*. IETF, 2006.
- [13] J. Gruen, M. Goriunov, and T. Herfet, "Interactive RTP services with predictable reliability," in *Consumer Electronics, Berlin (ICCE-Berlin), 2013. ICCEBerlin 2013. IEEE Third International Conference on*, 2013, pp. 371–375.
- [14] J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols," RFC Editor, Fremont, CA, USA, 5245, Apr. 2010.
- [15] N. W. Group, J. Rosenberg, R. Mahy, P. Matthews, and D. Wing, "RFC 5389 - Session Traversal Utilities for NAT (STUN)," IETF, Oct. 2008.