

UNIVERSITATEA BABEȘ-BOLYAI CLUJ-NAPOCA
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
SPECIALIZAREA INFORMATICĂ

LUCRARE DE LICENȚĂ

Distribuția streamurilor audio-video în sesiuni WebRTC

Conducător științific
Lect. dr. Sterca Adrian-Ioan

Absolvent
Haja Florin-Gabriel

2021

ABSTRACT

Această lucrare este un studiu în profunzime al protocoalelor de comunicare folosite de proiectul WebRTC și al modului în care conținutul audio-video poate fi distribuit în cadrul a mai multor clienți. Odată ce oamenii folosesc Internetul în scop productiv din ce în ce mai mult, este crucială asigurarea faptului că informația este transmisă pe cât de fluent posibil. Deoarece unele conexiuni la Internet au o lățime de bandă limitată, este mai dificilă gestionarea sesiunilor de apel video cu mulți utilizatori.

Putem opta ori pentru un server media care să gestioneze livrarea datelor pentru noi, ori pentru a conecta fiecare client cu toți ceilalți, într-o manieră a topologiei mesh (engl. plasă). De exemplu, în cadrul participării la un curs, se poate observa un scenariu de broadcasting, în care profesorul furnizează conținutul, iar studenții doar îl observă, așadar datele sunt transmise într-o singură direcție, iar topologia mesh nu are un beneficiu real. În locul servirii tuturor studenților deodată, putem servi câțiva, aceștia trimițând repetat streamul mai departe, până când toți pot să vizioneze ceea ce predă profesorul. Se poate observa o topologie arborescentă. Nu mai este nevoie de un server media, însemnând costuri de mentenanță mai mici, cu un compromis din partea clienților al lățimii de bandă folosite.

Cuprins

1	Introducere	1
2	Familiarizarea cu WebRTC	2
2.1	Istoria	2
2.2	API-uri JavaScript	2
2.2.1	Capturarea conținutului media	2
2.2.2	Inițializarea unei conexiuni peer	3
2.2.3	Adăugarea stream track-urilor	4
2.2.4	Obținerea candidaților ICE	4
2.3	Signaling	5
2.4	NAT traversal	5
3	Topologii des întâlnite în rețelistică	6
3.1	Sectiune	6
3.1.1	Subsectiune	6
3.2	Sectiune	7
3.2.1	Subsectiune	7
4	Contribuția personală	8
5	Concluzii și posibile îmbunătățiri	10
	Bibliografie	11

Capitolul 1

Introducere

Introducere: obiectivele lucrării și descrierea succintă a capitolelor, prezentarea temei, prezentarea contribuției proprii, respectiv a rezultatelor originale și menționarea (dacă este cazul) a sesiunii de comunicări unde a fost prezentată sau a revistei unde a fost publicată.

Capitolul 2

Familiarizarea cu WebRTC

Acest capitol are ca scop introducerea în principiile de bază și în componentele proiectului WebRTC, cu menirea de a putea înțelege conceptele următoare. Acoperă API-urile comune, precum și principiile de rețelistică folosite pentru comunicare calitativă, cu latență mică.

2.1 Istoria

WebRTC a fost dezvoltat ca un standard de către World Wide Web Consortium (W3C) și de către Internet Engineering Task Force (IETF), ca o colecție open-source de standarde și protocoale. Tehnologiile necesare, precum codec-uri și algoritmi de anulare a ecoului, au fost dezvoltate de către o companie suedeză numită Global IP Solutions (GIPS), care a fost mai târziu cumpărată de Google în mai 2010 [EAST18].

2.2 API-uri JavaScript

WebRTC este furnizat cu fiecare browser web popular, precum și cu framework-urile derivate (ex. Electron). Este metoda principală de a dezvolta aplicații noi de conferințe video. Expune mai multe API-uri JavaScript ușor de folosit, fiecare responsabil de setul său specific de funcții, precum: cameră și microfon, conținutul ecranului și conexiuni de la egal la egal.

2.2.1 Capturarea conținutului media

Pentru a face folosi streaming-ul audio-video cu un scop, o sursă de conținut este necesară. Media Devices este un nume comun pentru camerele și microfoanele conectate. În JavaScript, aceste device-uri sunt accesibile prin intermediul interfeței `navigator.mediaDevices`, prin care toate dispozitivele conectate pot fi

enumerate, urmărite pentru schimbări, precum și deschise pentru a obține o instanță de `MediaStream` [Weba].

Acest API este folosit prin apelul funcției `mediaDevices.getUserMedia()`, care returnează un `promise` al cărei funcție de `resolve` conține un parametru pentru `MediaStream`-ul asociat dispozitivului. Aceasta cere furnizarea ca parametru un obiect de tip `MediaStreamConstraints`, prin care constrângeri asupra sursei audio și video se pot specifica, precum și, opțional, identitatea peer, singura care are acces la stream, unde conținutul este protejat ca și cum regulile CORS cross-origin ar fi în aplicare.

O altă opțiune este de a partaja conținutul ecranului. Aceasta este posibilă prin intermediul funcției `mediaDevices.getDisplayMedia()`, foarte similară cu `getUserMedia`, care de asemenea returnează un `promise` ce rezolvă un `MediaStream`. Totuși, constrângerile video sunt diferite: alegerea între multiple opțiuni de a afișa cursorul (mereu, doar când este în mișcare, sau niciodată) și de a afișa zona capturată (întregul ecran, doar o fereastră, sau o filă din browser).

Înregistrarea conținutului unui `MediaStream` este posibilă prin API-ul `MediaRecorder`, dar în această teză, ne vom concentra doar pe streaming.

Vizionarea streamului o chestiune de a crea un element HTML `video`, setarea obiectului sursă ca fiind `MediaStream`-ul și de a furniza o funcție care să redea streamul când metadatele sunt încărcate, ca event handler în proprietatea `onloadedmetadata` din elementul `video`.

2.2.2 Inițializarea unei conexiuni peer

Conexiunile peer sunt o parte a specificațiilor WebRTC care ajută la stabilirea unei conexiuni între două aplicații pe două calculatoare diferite pentru a comunica printr-un protocol peer-to-peer. Acestea pot transmite audio, video, sau date bine (cât timp clienții suportă API-ul `RTCDataChannel`) [Webb].

Fiecare conexiune peer este gestionată de un obiect `RTCPeerConnection`, care este instanțiată prin constructorul său specificat ce primește ca parametru un `RTCConfiguration`, care definește modul în care conexiunea peer este pregătită și care ar trebui să conțină informații despre serverele ICE folosite.

Pentru inițierea comunicării, este prioritară crearea unei cereri sau unui răspuns SDP, în funcție dacă este vorba despre peer-ul ce apelează, sau despre peer-ul ce răspunde. Apelantul trebuie să trimită obiectul SDP pe care l-a creat peer-ului de la distanță printr-un canal de signaling, diferit de cel prin care se vor transmite datele. Procedura numită signaling nu are o definiție standard.

Inițierea unui `RTCPeerConnection` din apelant cere crearea unui descriptor local, obiect de tip `RTCSessionDescription` prin metoda `createOffer()` din

instanța de `RTCPeerConnection`. Va fi setat ca fiind descriptor local prin `setLocalDescription()` și va fi trimis la apelat prin canalul de signaling. Apelatul, când primește descriptorul sesiunii, va apela `setRemoteDescription()` și va crea un răspuns la cererea primită cu `createAnswer()` care de asemenea va fi trimis prin canalul de signaling. Inițiatorul apelului va primi răspunsul și îl va seta ca fiind descriptorul remote.

2.2.3 Adăugarea stream track-urilor

După crearea unei instanțe de `RTCPeerConnection`, track-urile de stream trebuie adăugate. Revenind la dispozitivele media și la obiectele sale `MediaStream`, track-urile sunt conținute într-o listă accesibilă din funcția `getTracks()`. Acestea pot fi adăugate la conexiunea peer cu `addTrack`, care cere instanța streamului [Webc].

Peer-ul care răspunde nu conține o instanță proprie de `MediaStream` care să conțină track-urile, așadar trebuie să creeze una și să adauge track-urile remote la ea. Un handler de evenimente numit `ontrack` trebuie adăugat. Gestionează câte un track odată. În acest caz, handler-ul va adăuga la instanța nou creată de `MediaStream`, care va fi setată ca obiectul sursă al elementului `video`.

2.2.4 Obținerea candidaților ICE

Schimbul de informație de conectivitate este obligatoriu înainte ca doi peers pot comunica. Condițiile rețelei pot fi variabile în funcție de un număr de factori (ex. ascunderea adreselor sale IP prin NAT), așadar un intermediar este folosit pentru a descoperi candidații pentru conectarea la peer. Acesta este un serviciu extern și se numește ICE (Interactive Connectivity Establishment) și se bazează pe un server STUN sau pe un server TURN. Indirect, serverele STUN (Session Traversal Utilities for NAT) sunt folosite în cele mai multe aplicații WebRTC, deoarece principiul lor de funcționare este mai simplu. Acestea sunt specificate în câmpul `iceServers` al obiectului de tip `RTCConfiguration` folosit la crearea obiectului `RTCPeerConnection`.

Fiecare instanță de `RTCPeerConnection` conține un handler de evenimente pentru noii candidați, numit `onicecandidate`. Este recomandat să se adauge unul care apelează `addIceCandidate()` și trimite noii candidați la celălalt peer, care, la rândul său, îi va memora.

2.3 Signaling

Nu există un protocol standard definit pentru signaling (engl. „semnalizare”). Orice mecanism RPC care funcționează pe aplicații web poate fi aplicat, indiferent dacă este un API Web bazat pe HTTP (ex. REST sau GraphQL) sau pe WebSockets, care folosește HTTP pentru inițializare.

In the classic VoIP world, the dominant signaling protocol was SIP (Session Initiation Protocol), which brought a variety of features meant for

2.4 NAT traversal

Capitolul 3

Topologii des întâlnite în rețelistică

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

3.1 Sectiune

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

3.1.1 Subsectiune

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

3.2 Sectiune

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

3.2.1 Subsectiune

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Capitolul 4

Contribuția personală

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu

fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Capitolul 5

Concluzii și posibile îmbunătățiri

Concluzii ...

Bibliografie

- [EAST18] Naktal Edan, Ali Al-Sherbaz, and Scott Turner. Design and implement a hybrid webrtc signalling mechanism for unidirectional & bi-directional video conferencing. *International Journal of Electrical and Computer Engineering*, 8:390–399, 02 2018.
- [Weba] Getting started with media devices. <https://webrtc.org/getting-started/media-devices>. Online; accessed 13 April 2021.
- [Webb] Getting started with peer connections. <https://webrtc.org/getting-started/peer-connections>. Online; accessed 13 April 2021.
- [Webc] Getting started with remote streams. <https://webrtc.org/getting-started/remote-streams>. Online; accessed 13 April 2021.