Internet Engineering Task Force (IETF)
Request for Comments : 8445
RFC rendue obsolète : 5245
Catégorie : Sur la voie de la normalisation
ISSN : 2070-1721

A. Keranen, Ericsson
C. Holmberg, Ericsson
J. Rosenberg, jdrosen.net
juillet 2018
Traduction Claude Brière de L'Isle

Établissement de connexité interactive (ICE) : protocole de traversée de traducteur d'adresse réseau (NAT)

Résumé
Le présent document décrit un protocole pour la traversée de traducteur d'adresse réseau (NAT, *Network Address Translator*) pour la communication fondée sur UDP. Ce protocole est appelé établissement de connexité interactive (ICE, *Interactive Connectivity Establishment*). ICE utilise le protocole d'utilitaires de traversée de session pour les NAT (STUN, *Session Traversal Utilities for NAT*) et son extension, "traversée de NAT au moyen d'un relais" (TURN, *Traversal Using Relay NAT*).

Le présent document rend obsolète la RFC 5245.

Statut de ce mémoire
Ceci est un document de l'Internet sur la voie de la normalisation.

Le présent document a été produit par l'équipe d'ingénierie de l'Internet (IETF). Il représente le consensus de la communauté de l'IETF. Il a subi une révision publique et sa publication a été approuvée par le groupe de pilotage de l'ingénierie de l'Internet (IESG). Tous les documents approuvés par l'IESG ne sont pas candidats à devenir une norme de l'Internet ; voir la Section 2 de la RFC5741.

Les informations sur le statut actuel du présent document, tout errata, et comment fournir des réactions sur lui peuvent être obtenues à http://www.rfc-editor.org/info/rfc8445

Table des matières

1. Introduction

Les protocoles qui établissent des sessions de communication entre des homologues impliquent normalement l'échange des adresses et accès IP pour les sources et collecteurs de données. Cependant, cela pose des défis quand ils doivent opérer à travers des traducteurs d'adresse réseau (NAT, *Network Address Translator*) [RFC3235]. Ces protocoles cherchent aussi à créer un flux de données directement entre les participants, de sorte qu'il n'y ait pas d'intermédiaire de couche d'application entre eux. Cela est fait pour réduire la latence des données, diminuer les pertes de paquet, et réduire les coûts de fonctionnement du déploiement de l'application. Cependant, il est difficile d'accomplir cela à travers les NAT. Un traitement complet des raisons de cela sort du domaine d'application de la présente spécification.

De nombreuses solutions ont été définies pour permettre à ces protocoles d'opérer à travers les NAT. Cela inclut les passerelles de couche application (ALG, *Application Layer Gateway*) le protocole de contrôle de boîtier de médiation [RFC3303], la spécification originale de simple traversée de NAT par UDP (STUN, *Simple Traversal of UDP Through NAT*) [RFC3489] (noter que la RFC 3489 a été rendue obsolète par la RFC 5389) et IP spécifique de domaine [RFC3102] [RFC3103] ainsi que les extensions de description de session nécessaires pour les faire fonctionner, comme l'attribut de protocole de description de session (SDP, *Session Description Protocol*) [RFC4566] pour le protocole de commande en temps réel (RTCP, *Real-Time Control Protocol*) [RFC3605]. Malheureusement, ces techniques ont toutes des avantages et des inconvénients qui les rendent chacune optimales dans certaines topologies de réseau, mais un mauvais choix dans d'autres. Le résultat est que les administrateurs et les mises en œuvre font des hypothèses sur les topologies des réseaux dans lesquels leurs solutions vont être déployées. Cela introduit de la complexité et de la fragilité dans le système.

La présente spécification définit l'établissement de connexité interactive (ICE, *Interactive Connectivity Establishment*) comme une technique pour la traversée de NAT pour les flux de données fondés sur UDP (bien que ICE ait été étendu pour traiter les protocoles de transport, comme TCP [RFC6544]). ICE fonctionne en échangeant des adresses et accès IP, qui sont alors essayés quant à leur connectivité par des vérifications de connexité d'homologue à homologue. Les adresses et accès IP sont échangés en utilisant des mécanismes spécifiques de l'usage de ICE (par exemple, dans un échange offre/réponse) et les vérifications de connexité sont effectuées en utilisant STUN [RFC5389]. ICE utilise aussi la traversée de NAT au moyen d'un relais (TURN, *Traversal Using Relay around NAT*) [RFC5766], une extension de STUN. Parce que ICE échange plusieurs adresses et accès IP pour chaque flux de supports, il permet aussi le choix d'adresse pour les hôtes multi-rattachements et double piles. Pour cette raison, la [RFC5245] déconseillait les solutions précédemment définies dans la [RFC4091] et la [RFC4092].

L'Appendice B donne des informations sur les fondements et les motivations concernant les décisions de conceptions de ICE.

2. Vue d'ensemble de ICE

Dans un déploiement normal de ICE, il y a deux points d'extrémité (agents ICE) qui veulent communiquer. Noter que ICE n'est pas destiné à la traversée de NAT pour le protocole de signalisation, qui est supposée être fournie via un autre mécanisme. ICE suppose que les agents sont capables d'établir une connexion de signalisation entre eux.

Initialement, les agents sont ignorants de leur propre topologie. En particulier, les agents peuvent ou non être derrière des NAT (ou plusieurs rangées de NAT). ICE permet aux agents de découvrir assez d'informations sur leurs topologies pour trouver éventuellement un ou plusieurs chemins par lesquels ils peuvent établir une session de données.

La Figure 1 montre un déploiement typique de ICE. Les agents sont marqués L et R. L et R sont tous deux derrière leurs propres NAT respectifs, bien qu'ils puissent ne pas le savoir. Le type de NAT et ses propriétés ne sont pas connus non plus. L et R sont capables de s'engager dans un processus d'échange de candidats, dont l'objet est d'établir une session de données entre L et R. Normalement, cet échange va se faire par un serveur de signalisation (par exemple, un mandataire SIP).

En plus des agents, un serveur de signalisation, et des NAT, ICE est normalement utilisé de concert avec des serveurs STUN ou TURN dans le réseau. Chaque agent peut avoir son propre serveur STUN ou TURN, ou ce peut être le même.

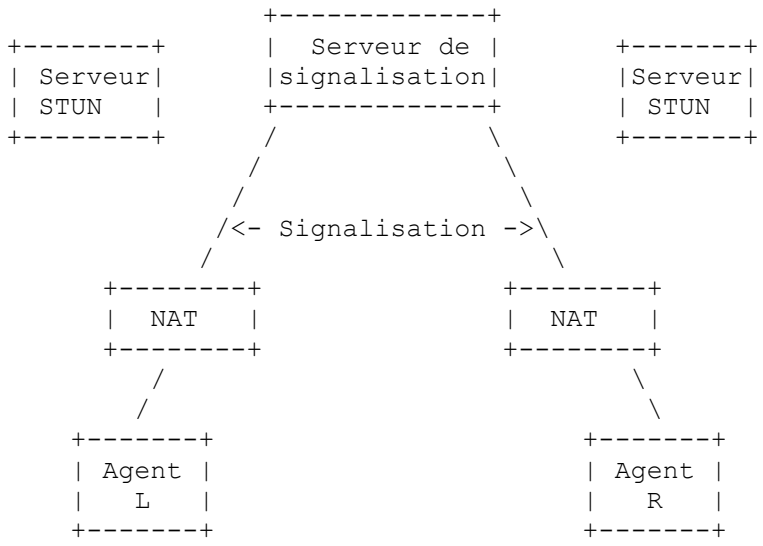


Figure 1 : Scénario de déploiement de ICE

L'idée de base de ICE est la suivante : chaque agent a diverses adresses de transport candidates (combinaison d'adresse et accès IP pour un protocole de transport particulier, qui est toujours UDP dans cette spécification) qu'il pourrait utiliser pour communiquer avec l'autre agent. Cela pourrait inclure :

- Une adresse de transport sur une interface réseau directement rattachée.
- Une adresse de transport traduite sur le côté public d'un NAT (une adresse "reflet de serveur").
- Une adresse de transport allouée d'un serveur TURN (une "adresse relayée").

Potentiellement, toute adresse de transport candidate de L peut être utilisée pour communiquer avec toute adresse de transport candidate de R. En pratique, cependant, de nombreuses combinaisons ne vont pas fontionner. Par exemple, si L et R sont tous deux derrière des NAT, leurs adresses d'interface directement rattachées ont peu de chances d'être capables de communiquer directement (c'est pourquoi ICE est nécessaire, après tout !). L'objet de ICE est de découvrir quelles paires d'adresses vont fonctionner. La façon dont ICE fait cela est d'essayer systématiquement toutes les paires possibles (dans un ordre soigneusement établi) jusqu'à ce qu'il en trouve une ou plusieurs qui fonctionnent.

2.1 Rassemblement des candidats

Pour exécuter ICE, un agent ICE identifie et rassemble une ou plusieurs adresses candidates. Un candidat a une adresse de transport -- une combinaison d'adresse et accès IP pour un protocole de transport particulier (dont seul UDP est spécifié ici). Il y a différents types de candidates ; certains sont déduits d'interfaces réseau physiques ou logiques, et d'autres sont découverts via STUN et TURN.

La première catégorie de candidates est celle avec une adresse de transport obtenue directement d'une interface locale. Une telle candidate est appelée "une candidate hôte". L'interface locale pourrait être Ethernet ou Wi-Fi, ou elle pourrait être obtenue par un mécanisme de tunnel, comme un réseau privé virtuel (VPN, *Virtual Private Network*) ou IP mobile (MIP, *Mobile IP*). Dans tous les cas, une telle interface réseau apparaît à l'agent comme une interface locale à partir de laquelle des accès (et donc des candidates) peuvent être alloués.

Ensuite, l'agent utilise STUN ou TURN pour obtenir des candidates supplémentaires. Il y a là deux nuances : les adresses traduites du côté public d'un NAT (les candidates reflets de serveur) et les adresses sur les serveurs TURN (candidates relayées). Quand les serveurs TURN sont utilisés, les deux types de candidates sont obtenus du serveur TURN. Si

seulement les serveurs STUN sont utilisés, seules les candidates reflets de serveur sont obtenues d'eux. La relation de ces candidates au candidat hôte est montrée à la Figure 2. Dans cette figure, les deux types de candidates sont découverts en utilisant TURN. Dans la figure, la notation X:x signifie l'adresse IP X et l'accès UDP x.

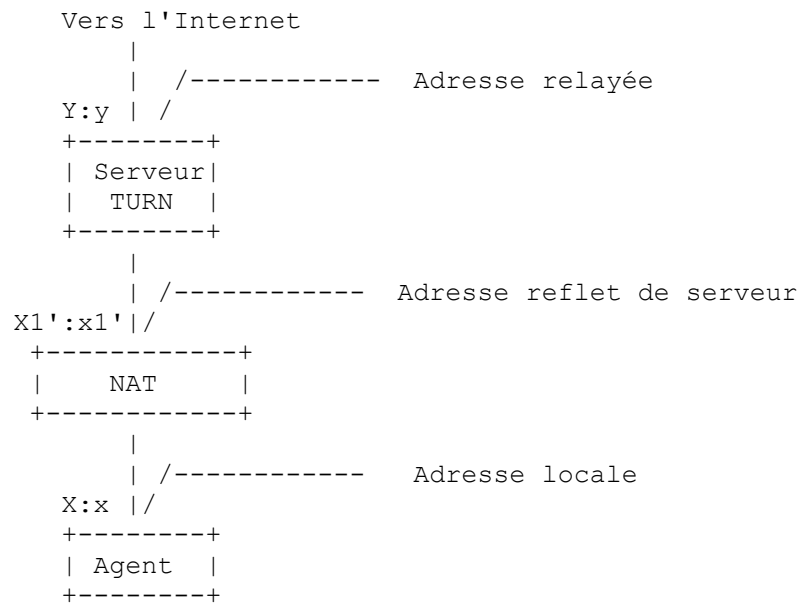


Figure 2 : Relations de candidats

Quand l'agent envoie une demande TURN Allocate à partir de l'adresse et accès IP X:x, le NAT (en supposant qu'il y en ait un) va créer un lien X1':x1', transposant cette candidate reflet de serveur en la candidate hôte X:x. Les paquets sortants envoyés de la candidate hôte vont être traduits par le NAT en la candidate reflet de serveur. Les paquets entrants envoyés à la candidate reflet de serveur vont être traduits par le NAT en la candidate hôte et transmis à l'agent. La candidate hôte associée à une candidate reflet de serveur donnée est la "base".

Note : "base" se réfère à l'adresse qu'un agent envoie pour une candidate particulière. Donc, à la limite, les candidates hôtes ont aussi une base, mais c'est la même que celle de la candidate hôte.

Quand il y a plusieurs NAT entre l'agent et le serveur TURN, la demande TURN va créer un lien sur chaque NAT, mais seule la candidate reflet de serveur la plus externe (celle qui est la plus proche du serveur TURN) va être découverte par l'agent. Si l'agent n'est pas derrière un NAT, alors la candidate de base va être la même que la candidate reflet de serveur, et la candidate reflet de serveur est redondante et va être éliminée.

La demande Allocate arrive alors au serveur TURN. Celui-ci alloue un accès à partir de son adresse IP locale Y, et génère une réponse Allocate, informant l'agent de cette candidate relayée. Le serveur TURN informe aussi l'agent de la candidate reflet de serveur, X1':x1', en copiant l'adresse de transport de source de la demande Allocate dans la réponse Allocate. Le serveur TURN agit comme un relais de paquets, transmettant le trafic entre L et R. Afin d'envoyer le trafic à L, R envoie le trafic au serveur TURN à Y:y, et le serveur TURN transmet cela à X1':x1', qui passe à travers le NAT où il est transposé en X:x et livré à L.

Quand seuls des serveurs STUN sont utilisés, l'agent envoie une demande Binding STUN [RFC5389] à son serveur STUN. Le serveur STUN va informer l'agent de la candidate reflet de serveur X1':x1' en copiant l'adresse de transport de source de la demande Binding dans la réponse Binding.

2.2 Vérifications de connexité

Une fois que L a rassemblé toutes ses candidates, il les ordonne par priorité de la plus haute à la plus basse et les envoie à R sur le canal de signalisation. Quand R reçoit les candidates de L, il effectue le même processus de rassemblement et répond avec sa propre liste de candidates. À la fin de ce processus, chaque agent ICE a une liste complète de ses candidates et de celles de son homologue. Il les apparie, résultant en paires candidates. Pour voir quelles paires fonctionnent, chaque agent programme une série de vérifications de connexité. Chaque vérification est une transaction de demande/réponse STUN que le client va effectuer sur une paire candidate particulière en envoyant une demande STUN de la candidate locale à la

candidate distante.

Le principe de base des vérifications de connexité est simple :
1. Tri des paires candidates par ordre de priorité.
2. Envoi des vérifications sur chaque paire candidate dans l'ordre de priorité.
3. Accuser réception des vérifications reçues de l'autre agent.

Avec les deux agents qui effectuent une vérification sur une paire candidate, le résultat est une prise de contact en 4 phases :

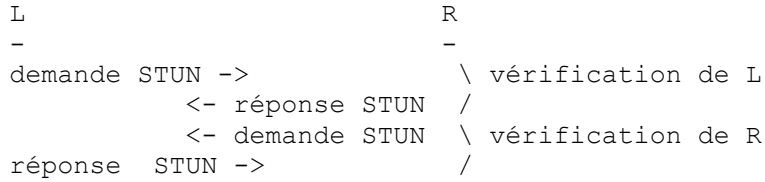


Figure 3 : Vérification de connexité de base

Il est important de noter que les demandes STUN sont envoyées de et à exactement les mêmes adresses et accès IP que celles qui vont être utilisées pour les données (par exemple, RTP, RTCP, ou autres protocolss). Par conséquent, les agents démultiplexent STUN et les données en utilisant le contenu des paquets plutôt que l'accès sur lequel ils sont reçus.

Parce que une demande STUN Binding est utilisée pour la vérification de connexité, la réponse STUN Binding va contenir l'adresse de transport de l'agent traduite du côté public de tout NAT entre l'agent et son homologue. Si cette adresse de transport est différente de celle des autres candidates que l'agent connaît déjà, elle représente une nouvelle candidate (candidate reflet de l'homologue) qui va alors être vérifiée par ICE juste comme toute autre candidate.

Parce que l'algorithme ci-dessus cherche toutes les paires candidates, si une paire qui fonctionne existe, l'algorithme va finalement la trouver, sans considération de l'ordre de tri des candidates. Afin de produire plus vite (et mieux) les résultats, les candidates sont triées dans l'ordre spécifié. La liste résultante des paires candidates triées est appelée la "liste de contrôle".

L'agent travaille sur la liste de contrôle en envoyant périodiquement une demande STUN pour la prochaine paire candidate sur la liste. C'est appelé une "vérification ordinaire". Quand une transaction STUN réussit, une ou plusieurs paires candidates vont devenir des "paires valides" et vont être ajoutées à la liste de paires candidates appelée la "liste valide".

Pour une optimisation, aussitôt que R obtient le message de vérification de L, R programme un message de vérification de connexité à envoyer à L sur la même paire candidate. Ceci est appelé une "vérification déclenchée", et cela accélère le traitement de découverte des paires valides.

À la fin de cette prise de contact, L et R savent tous deux qu'ils peuvent envoyer (et recevoir) des messages de bout en bout dans les deux directions.

En général, l'algorithme de priorité est conçu de telle sorte que les candidates d'un type similaire obtiennent des priorités similaires afin que plus de chemins directs (c'est-à-dire, des chemins sans relais de données ni NAT) soient préférés à des chemins indirects (chemins avec des relais de données ou des NAT). Dans ces lignes directrices, cependant, les agents ont une grande latitude quant à la façon de régler leurs algorithmes.

Un flux de données pourrait consister en de multiples composants (éléments d'un flux de données qui exigent leur propre ensemble de candidates, par exemple, RTP et RTCP).

2.3 Désignation de paires de candidats et conclusion de ICE

ICE alloue à un des agents ICE le rôle d'agent contrôleur, et à l'autre le rôle d'agent contrôlé. Pour chaque composant d'un flux de données, l'agent contrôleur désigne une paire valide (dans la liste valide) à utiliser pour les données. Le moment exact de la désignation est fondé sur la politique locale.

Quand il est désigné, l'agent contrôleur laisse les vérifications se continuer jusqu'à ce qu'au moins une paire valide pour chaque composant d'un flux de données soit trouvée, et il prend alors une paire valide et envoie une demande STUN sur

cette paire, en utilisant un attribut pour indiquer à l'homologue contrôlé qu'elle a été désignée. Cela est montré à la Figure 4.

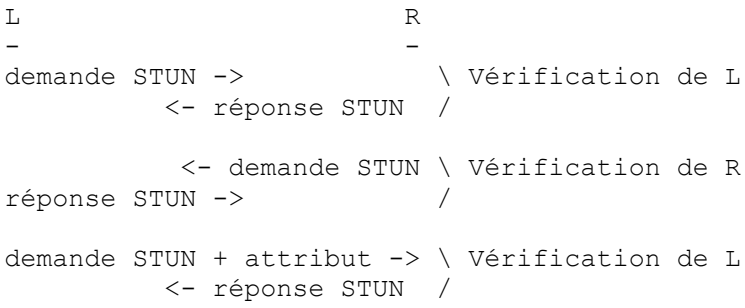


Figure 4 : Désignation

Une fois que l'agent contrôlé a reçu la demande STUN avec l'attribut, il va vérifier (sauf si la vérification a déjà été faite) la même paire. Si les transactions ci-dessus réussissent, les agents vont établir le fanion Désigné pour les paires et vont annuler toutes les futures vérifications pour ce composant du flux de données. Une fois qu'un agent a établi le fanion Désigné pour chaque composant d'un flux de données, les paires deviennent les paires choisies. Après cela, seules les paires choisies vont être utilisées pour envoyer et recevoir les données associées à ce flux de données.

2.4 Redémarrage de ICE

Une fois ICE terminé, il peut être redémarré à tout moment pour un ou tous les flux de données par l'un ou l'autre agent ICE. Ceci est fait en envoyant des informations de candidate mises à jour indiquant un redémarrage.

2.5 Mise en œuvre légère

Certains agents ICE vont toujours être connectés à l'Internet public et avoir une adresse IP publique à laquelle ils peuvent recevoir des paquets de tout correspondant. Pour rendre plus facile à ces appareils la prise en charge de ICE, ICE définit un type spécial de mise en œuvre appelée "lègère" (à l'opposé de la mise en œuvre complète normale). Les agents légers utilisent seulement des candidates hôtes et ne génèrent pas de vérifications de connexité ou n'utilisent pas d'automate à états, bien qu'ils doivent être capables de répondre aux vérifications de connexité.

3. Utilisation de ICE

Le présent document spécifie l'utilisation générique de ICE avec les protocoles qui fournissent les moyens d'échanger les informations de candidates entre agents ICE. Les détails spécifiques (c'est-à-dire, comment coder les informations de candidates et le processus réel d'échange de candidates) pour les différents protocoles qui utilisent ICE (appelés les "protocoles d'utilisation") sont décrits dans des documents d'usage séparés.

Un mécanisme qui permet aux agents d'échanger les informations de candidates est l'utilisation de la sémantique d'offre/réponse (qui se fonde sur la [RFC3264]) au titre du protocole SIP [RFC3261], [RFC8839].

La [RFC7825] définit un usage de ICE pour le protocole de flux en temps réel (RSTP, *Real-Time Streaming Protocol*). Noter cependant que l'usage de ICE se fonde sur la RFC 5245.

4. Terminologie

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119], [RFC8174] quand, et seulement quand ils apparaissent tout en majuscules, comme montré ci-dessus.

Les lecteurs doivent être familiarisés avec la terminologie définie dans la [RFC5389] et dans les exigences pour le comportement de NAT pour UDP [RFC4787].

La présente spécification utilise la terminologie supplémentaire suivante :

Adresse de transport : combinaison d'une adresse IP et de l'accès du protocole de transport (comme UDP ou TCP).

Adresse et accès relatifs : adresse de transport relative à une candidate, qui est utile pour les besoins de diagnostics et autres. Si une candidate est un reflet de serveur ou d'homologue, l'adresse et accès relatifs sont égaux à la base pour cette candidate reflet de serveur ou d'homologue. Si la candidate est relayée, l'adresse et accès relatifs sont égaux à l'adresse transposée dans la réponse Allocate qui a fourni au client cette candidate relayée. Si la candidate est une candidate hôte, l'adresse et accès relatifs sont identiques à la candidate hôte.

Agent contrôleur : agent ICE qui désigne une paire candidate. Dans toute session, il y a toujours un agent contrôleur et un agent contrôlé.

Agent contrôlé : l'agent ICE qui attend que l'agent contrôleur désigne une paire candidate.

Agent ICE, agent : un agent ICE (parfois simplement appelé un "agent") est la mise en œuvre de protocole impliquée dans l'échange de candidats ICE. Il y a deux agents impliqués dans un échange de candidates normal.

Base : adresse de transport qu'un agent ICE envoie pour une candidate particulière. Pour les candidates hôtes, reflets de serveur, et reflets d'homologue, la base est la même que la candidate hôte. Pour les candidates relayées, la base est la même que la candidate relayée (c'est-à-dire, l'adresse de transport utilisée pour l'envoi par le serveur TURN).

Candidate, informations de candidate : adresse de transport qui est un potentiel point de contact pour la réception de données. Les candidates ont aussi des propriétés -- leur type (reflet de serveur, relayé, ou hôte) leur priorité, fondation, et base.

Candidate distante : candidate qu'un agent ICE a reçu de son homologue.

Candidate hôte : candidate obtenue en se liant à un accès spécifique à partir d'une adresse IP sur l'hôte. Cela inclut les adresses IP sur des interfaces physiques et logiques, comme celles obtenues à travers des VPN.

Candidate locale : candidate qu'un agent ICE a obtenu et peut envoyer à son homologue.

Candidate reflet de serveur : candidate dont l'adresse et l'accès IP sont un lien alloué par un NAT pour un agent ICE après qu'il a envoyé un paquet à travers le NAT à un serveur, comme un serveur STUN.

Candidate reflet d'homologue : candidate dont l'adresse et l'accès IP sont un lien alloué par un NAT pour un agent ICE après qu'il a envoyé un paquet à travers le NAT à son homologue.

Candidate relayée : candidate obtenu d'un serveur relais, comme un serveur TURN.

Composant : un composant est un élément d'un flux de données. Un flux de données peut exiger plusieurs composants, dont chacun doit fonctionner pour que le flux de données fonctionne comme un tout. Pour les flux de données RTP/RTCP, sauf si RTP et RTCP sont multiplexés dans le même accès, il y a deux composants par flux de données -- un pour RTP, et un pour RTCP. Un composant a une paire candidate, qui ne peut pas être utilisée par les autres composants.

Désignation : processus de l'agent contrôleur qui indique à l'agent contrôlé quelles paires candidates vont utiliser les agents ICE pour envoyer et recevoir les données. Le processus de désignation défini dans la présente spécification était appelé "désignation régulière" dans la RFC 5245. Le processus de désignation qui était appelé "désignation agressive" dans la RFC 5245 a été déconseillé dans la présente spécification.

Destination/candidate par défaut : la destination par défaut pour un composant d'un flux de données est l'adresse de transport qui serait utilisée par un agent ICE qui n'a pas de capacité ICE. Une candidate par défaut pour un composant est celle dont l'adresse de transport correspond à la destination par défaut pour ce composant.

Données, flux de données, session de données : quand ICE est utilisé pour établir des sessions de données, les données sont transportées en utilisant un protocole. Les supports sont généralement transportés sur RTP, composés d'un flux de paquets RTP. Une session de données se réfère aux paquets de données qui sont échangés entre les homologues sur le chemin créé et vérifié avec ICE.

Échange de candidates ICE, échange de candidates : processus par lequel les agents ICE échangent les informations (par

exemple, des candidates et des mots de passe) qui sont nécessaires pour effectuer ICE. L'offre/réponse avec codage SDP [RFC3264] est un exemple de protocole qui peut être utilisé pour échanger les informations de candidates.

Ensemble de listes de contrôle : liste ordonnée de toutes les listes de contrôle. L'ordre est déterminé par chaque usage ICE.

Fanion Désigné : une fois que la désignation d'une paire candidate a réussi, la paire candidate devient désignée, et la valeur de son fanion Désigné est réglée à vrai.

Fondation : chaîne arbitraire utilisée dans l'algorithme de fixation pour grouper les candidates similaires. Elle est la même pour deux candidates qui ont le même type, adresse IP de base, protocole (UDP, TCP, etc.) et serveur STUN ou TURN. Si un d'eux est différent, alors la fondation va être différente.

Homologue initiateur, agent initiateur, initiateur: c'est un agent ICE qui initie le processus d'échange de candidats ICE.

Homologue répondant, agent répondant, répondant : un agent répondant est un agent ICE qui reçoit et répond au processus d'échange de candidates initié par l'agent initiateur.

Homologue : du point de vue d'un des agents ICE dans une session, son homologue est l'autre agent. Spécifiquement, du point de vue de l'agent initiateur, l'homologue est l'agent répondant. Du point de vue de l'agent répondant, l'homologue est l'agent initiateur.

Liste de contrôle : ensemble ordonné de paires candidates qu'un agent ICE va utiliser pour générer des vérifications.

Liste valide : ensemble ordonné de paires candidates pour un flux de données qui ont été validées par une transaction STUN réussie.

Mise en œuvre complète : mise en œuvre ICE qui effectue l'ensemble complet de fonctionnalités défini par la présente spécification.

Mise en œuvre légère : mise en œuvre ICE qui omet certaines fonctions, mettant seulement en œuvre autant qu'il est nécessaire pour qu'un homologue qui n'est pas une mise en œuvre légère tire partie de ICE. Les mises en œuvre légères ne tiennent aucun des automates à états et ne génèrent pas de vérifications de connexité.

Paire candidate : paire contenant une candidate locale et une candidate distante.

Paire choisie, paire candidate choisie : la paire candidate utilisée pour envoyer et recevoir les données pour un composant d'un flux de données est appelée la "paire choisie". Avant que les paires choisies aient été produites pour un flux de données, toute paire valide associée à un composant d'un flux de données peut être utilisée pour envoyer et recevoir des données pour le composant. Une fois qu'il y a des paires désignées pour chaque composant d'un flux de données, les paires désignées deviennent les paires choisies pour le flux de données. Les candidates associées aux paires choisies sont appelées les "candidates choisies".

Paire valide : paire candidate dont la candidata locala est égale à l'adresse transposée d'une réponse de vérification de connexité réussie et dont la candidate distante est égale à l'adresse de destination à laquelle la demande de vérification de connexité était envoyée.

Protocole utilisateur, usage de ICE : protocole qui utilise ICE pour la traversée de NAT. Une spécification d'usage définit les détails spécifiques du protocole de la façon dont les procédures définies ici sont appliquées à ce protocole.

Session ICE : une session ICE consiste en toutes les actions relatives à ICE en commençant par le rassemblement de candidates, suivi par les interactions (échange de candidates, vérifications de connexité, désignations, et maintiens en vie) entre les agents ICE jusqu'à ce que toutes les candidates soient libérées ou qu'un redémarrage de ICE soit déclenché.

Temporisateur Ta : temporisateur pour générer de nouvelles transactions STUN ou TURN.

Temporisateur de retransmission (RTO, *Retransmission Timeout*) : temporisateur de retransmission pour une transaction STUN ou TURN donnée.

Vérification, vérification de connexité, vérification STUN : demande STUN Binding pour les besoins de la vérification de connexité. Une vérification est envoyée de la base de la candidate locale à la candidate distante d'une paire candidate.

Vérification déclenchée : vérification de connexité générée en conséquence de la réception d'une vérification de connexité provenant de l'homologue.

Vérification ordinaire : vérification de connexité générée par un agent ICE par suite de l'expiration périodique d'un temporisateur, lui ordonnant d'envoyer une vérification.

5. Rassemblement et échange de candidats ICE

Au titre du traitement ICE, les agents, initiateur et répondant, rassemblent tous deux des candidates, donnent des priorités et éliminent les candidates redondantes, et échangent les informations de candidates avec l'homologue comme défini par le protocole d'utilisation (d'usage de ICE). Les spécificités du mécanisme de codage de candidate et la sémantique de l'échange d'informations de candidate sortent du domaine d'application de la présente spécification.

5.1 Mise en œuvre complète

5.1.1 Rassemblement des candidates

Un agent ICE rassemble les candidates quand il estime que cette communication est imminente. Un agent initiateur peut faire cela en se fondant sur une indication d'interface d'utilisateur ou sur une demande explicite d'initier une session. Chaque candidate a une adresse de transport. Elle a aussi un type et une base. Quatre types sont définis et rassemblés par la présente spécification -- candidates hôtes, candidates reflets de serveur, candidates reflets d'homologue, et candidates relayées. Les candidates reflets de serveur sont rassemblées en utilisant STUN ou TURN, et les candidates relayées sont obtenues par TURN. Les candidates reflets d'homologue sont obtenues dans les phases ultérieures de ICE, par suite des vérifications de connexité.

Le processus pour rassembler les candidates chez l'agent répondant est identique au processus chez l'agent initiateur. Il est RECOMMANDÉ que l'agent répondant commence ce processus immédiatement à réception des informations de candidate, avant d'alerter l'utilisateur de l'application associée à la session ICE.

5.1.1.1 Candidates hôtes

Les candidates hôtes sont obtenues en se liant aux accès sur une adresse IP rattachée à une interface (physique ou virtuelle, incluant des interfaces de VPN) sur l'hôte.

Pour chaque composant de chaque flux de données que l'agent ICE souhaite utiliser, l'agent DEVRAIT obtenir une candidate sur chaque adresse IP qu'a l'hôte, avec les exceptions mentionnées ci-dessous. L'agent obtient chaque candidate en se liant à un accès UDP sur l'adresse IP spécifique. Une candidate hôte (et bien sût toute candidate) est toujours associée à un composant spécifique pour lequel elle est candidate.

Chaque composant a un identifiant qui lui est alloué, appelé "identifiant de composant". Pour les flux de données RTP/RTCP, sauf si RTP et RTCP sont tous deux multiplexés dans le même accès UDP (multiplexage RTP/RTCP) le RTP lui-même a un identifiant de composant de 1, et RTCP a un identifiant de composant de 2. Dans le cas de multiplexage RTP/RTCP, un identifiant de composant de 1 est utilisé pour les deux.

Quand les candidates sont obtenues, sauf si l'agent sait avec certitude que le multiplexage RTP/RTCP va être utilisé (c'est-à-dire, l'agent sait que l'autre agent prend aussi en charge, et veut utiliser, le multiplexage RTP/RTCP) ou à moins que l'agent prenne seulement en charge le multiplexage RTP/RTCP, l'agent DOIT obtenir une candidate séparée pour RTCP. Si un agent a obtenu une candidate pour RTCP, et finit par utiliser le multiplexage RTP/ RTCP, l'agent n'a pas besoin d'effectuer de vérifications de connexité sur la candidate RTCP. L'absence d'un identifiant de composant 2 n'implique pas l'utilisation du multiplexage RTCP/RTP, car cela pourrait aussi signifier que RTCP n'est pas utilisé.

Si un agent utilise des candidates séparées pour RTP et RTCP, il va finir avec 2*K candidates hôtes si un agent a K adresses IP.

Noter que l'agent répondant, quand il obtient ses candidates, va normalement savoir si l'autre agent prend en charge le multiplexage RTP/RTCP, auquel cas il n'a pas besoin d'obtenir une candidate séparée pour RTCP. Cependant, l'absence d'un identifiant de composant de 2 n'implique pas l'utilisation du multiplexage RTCP/RTP, car cela pourrait aussi signifier

que RTCP n'est pas utilisé..

L'utilisation de plusieurs composants, autres que pour des flux RTP/RTCP, est déconseillée car elle augmente la complexité du traitement ICE. Si plusieurs composants sont nécessaires, les identifiants de composant DEVRAIENT commencer à 1 et augmenter de 1 pour chaque composant.

La base pour chaque candidate hôte est réglée à la candidatr elle-même.

Les candidates hôtes sont rassemblées à partir de toutes les adresses IP avec les exceptions suivantes :
- Les adresses provenant d'une interface de rebouclage NE DOIVENT PAS être incluses dans les adresses candidates.
- Les adresses IPv6 compatibles IPv4 déconseillées [RFC4291] et les adresses IPv6 d'envoi individuel de site local [RFC3879] NE DOIVENT PAS être incluses dans les adresses candidates.
- Les adresses IPv6 transposées en IPv4 NE DEVRAIENT PAS être incluses dans les adresses candidates sauf si l'application qui utilise ICE ne prend pas en charge IPv4 (c'est-à-dire, si elle est une application IPv6 seul [RFC4038]).
- Si on rassemble une ou plusieurs candidates hôtes qui correspondent à une adresse IPv6 générée en utilisant un mécanisme qui empêche le traçage de localisation [RFC7721], les candidates hôtes qui correspondent aux adresses IPv6 qui permettent le traçage de localisation, sont configurées sur la même interface, et font partie du même préfixe de réseau NE DOIVENT PAS être rassemblées. De même, quand des candidates hôtes correspondant à une adresse IPv6 générée en utilisant un mécanisme qui empêche le traçage de localisation sont rassemblées, alors des candidates hôtes correspondant à des adresses IPv6 de liaison locale [RFC4291] NE DOIVENT PAS être rassemblées.

La spécification de choix d'adresse IPv6 par défaut [RFC6724] spécifie que les adresses temporaires [RFC4941] sont à préférer aux adresses permanentes.

5.1.1.2 Candidates de reflet de serveur et relayées

Un agent ICE DEVRAIT rassembler des candidates reflets de serveur et relayées. Cependant, l'utilisation de serveurs STUN et TURN peut être inutile dans certains réseaux et l'utilisation de serveurs TURN peut être coûteuse, de sorte que certains déploiements peuvent choisir de ne pas les utiliser. Si un agent ne rassemble pas de candidates reflets de serveur ou relayées, il est RECOMMANDÉ que la fonction soit mise en œuvre et juste désactivée par configuration, afin qu'elle puisse être réactivée par configuration si les conditions changent à l'avenir.

L'agent apparie chaque candidate hôte avec les serveurs STUN ou TURN avec lesquels il est configuré ou qu'il a découvert par un moyen quelconque. Il est RECOMMANDÉ qu'un nom de domaine soit configuré, que les procédures du DNS de la [RFC5389] (en utilisant les enregistrements SRV avec le service "stun") soient utilisées pour découvrir le serveur STUN, et que les procédures du DNS de la [RFC5766] (en utilisant les enregistrements SRV avec le service "turn") soient utilisées pour découvrir le serveur TURN.

Quand plusieurs serveurs STUN ou TURN sont disponibles (ou quand ils sont appris par des enregistrements du DNS et que plusieurs résultats sont retournés) l'agent PEUT rassembler des candidates pour tous et DEVRAIT rassembler des candidates pour au moins un d'eux (un serveur STUN et un serveur TURN). Il fait ainsi en appariant les candidates hôtes avec les serveurs STUN ou TURN, et pour chaque paire, l'agent envoie une demande Binding ou Allocate au serveur à partir de la candidate hôte. Les demandes Binding à un serveur STUN ne sont pas authentifiées, et tout attribut ALTERNATE-SERVER dans une réponse est ignoré. Les agents DOIVENT prendre en charge le mode de rétro-compatibilité pour la demande Binding définie dans la [RFC5389]. Les demandes Allocate DEVRAIENT être authentifiées en utilisant un accréditif à long terme obtenu par le client par d'autres moyens.

Le processus de rassemblement est contrôlé en utilisant un temporisateur, Ta. Chaque fois que Ta expire, l'agent peut générer une autre nouvelle transaction STUN ou TURN. Cette transaction peut être soit un ré-essai d'une transaction antérieure qui avait échoué avec une erreur récupérable (comme un échec d'authentification) soit une transaction pour une nouvelle paire de candidate hôte et serveur STUN ou TURN. L'agent NE DEVRAIT PAS générer de transactions plus fréquemment que une fois par chaque expiration de Ta. Voir à la Section 14 des lignes directrices sur la façon de régler Ta et la temporisation de retransmission STUN, RTO.

L'agent va recevoir une réponse Binding ou Allocate. Une réponse Allocate de succès va donner à l'agent une candidate reflet de serveur (obtenue de l'adresse transposée) et une candidate relayée dans l'attribut XOR-RELAYED-ADDRESS. Si la demande Allocate est rejetée parce que le serveur manque de ressources pour la satisfaire, l'agent DEVRAIT envoyer à la place une demande Binding pour obtenir une candidate reflet de serveur. Une réponse Binding va donner à l'agent seulement une candidate reflet de serveur (aussi obtenue de l'adresse transposée). La base de la candidate reflet de serveur est la candidate hôte d'où la demande Allocate ou Binding a été envoyée. La base d'une candidate relayée est cette candidate elle-même. Si une candidate relayée est identique à une candidate hôte (ce qui peut arriver dans des cas rares) la

candidate relayée DOIT être éliminée.

Si un agent IPv6 seul est dans un réseau qui utilise des technologies NAT64 [RFC6146] et DNS64 [RFC6147], il peut aussi rassembler des candidates reflets de serveur IPv4 et/ou relayées de serveurs STUN ou TURN IPv4 seul. Les agents IPv6 seul DEVRAIENT aussi utiliser la découverte de préfixe IPv6 [RFC7050] pour découvrir le préfixe IPv6 utilisé par NAT64 (si il en est) et générer en conséquence des candidates reflets de serveur pour chaque interface IPv6 seul. Les candidates reflets de serveur NAT64 sont mises en priorité comme les candidates reflets de serveur IPv4.

5.1.1.3 Calcul des fondations

L'agent ICE alloue à chaque candidate une fondation. Deux candidates ont la même fondation quand tous les points suivants sont vrais :

- o Elles ont le même type (hôte, relayé, reflet de serveur, ou reflet d'homologue).
- o Leurs bases ont la même adresse IP (les accès peuvent être différents).
- o Pour les candidates reflets et relayées, les serveurs STUN ou TURN utilisés pour les obtenir ont la même adresse IP (l'adresse IP utilisée par l'agent pour contacter le serveur STUN ou TURN).
- o Elles ont été obtenues en utilisant le même protocole de transport (TCP, UDP).

De même, deux candidates ont des fondations différentes si leurs types sont différents, leurs bases ont des adresses IP différentes, les serveurs STUN ou TURN utilisés pour les obtenir ont des adresses IP différentes (les adresses IP utilisées par les agent pour contacter le serveur STUN ou TURN) ou leurs protocoles de transport sont différents.

5.1.1.4 Maintien en vie des candidates

Une fois que les candidates reflets de serveur et relayées sont allouées, elles DOIVENT être gardées en vie jusqu'à ce que le traitement ICE soit achevé, comme décrit au paragraphe 8.3. Pour les candidates reflets de serveur apprises par une demande Binding, les liens DOIVENT être maintenus en vie par des demandes Binding supplémentaires au serveur. Les rafraîchissements pour les allocations sont faites en utilisant la transaction Refresh, comme décrit dans la [RFC5766]. Les demandes Refresh vont aussi rafraîchir la candidate reflet de serveur.

Les candidates hôtes ne se périment pas, mais les adresses candidates peuvent changer ou disparaître pour un certain nombre de raisons. Un agent ICE DEVRAIT surveiller les interfaces qu'il utilise, invalider les candidates dont la base a disparu, et acquérir de nouvelles candidates comme approprié quand de nouvelles adresses IP (sur des interfaces nouvelles ou actuellement utilisées) apparaissent.

5.1.2 Priorité des candidates

Le processus de priorité résulte en l'allocation d'une priorité à chaque candidate. Chaque candidate pour un flux de données DOIT avoir une priorité unique qui DOIT être un entier positif entre 1 et (2**31 - 1). Cette priorité va être utilisée par ICE pour déterminer l'ordre des vérifications de connexité et la préférence relative pour les candidates. Les plus fortes valeurs de priorité donnent plus de priorité que les valeurs inférieures.

Un agent ICE DEVRAIT calculer cette priorité en utilisant la formule du paragraphe 5.1.2.1 et choisir ses paramètres en utilisant les lignes directrices du paragraphe 5.1.2.2. Si un agent choisit d'utiliser des formules différentes, ICE peut prendre plus longtemps pour converger car les agents ne seront pas coordonnés dans leurs vérifications.

Le processus d'attribution de priorités aux candidats est commun à l'agent initiateur et à l'agent répondant.

5.1.2.1 Formule recommandée

La formule recommandée combine une préférence pour le type de candidate (reflet de serveur, reflet d'homologue, relayée, et hôte) une préférence pour l'adresse IP pour laquelle la candidate a été obtenue, et un identifiant de composant utilisant la formule suivante :

priorité = (2^24)*(préférence de type) + (2^8)*(préférence locale) + (2^0)*(256 - identifiant de composant)

La préférence de type DOIT être un entier de 0 (plus faible préférence) à 126 (plus haute préférence) inclus, DOIT être identique pour toutes les candidates de même type, et DOIT être différente pour les candidates de types différents. La préférence de type pour les candidates reflets d'homologue DOIT être plus haute que celle des candidates reflets de serveur. Régler la valeur de 0 signifie que les candidats de ce type vont seulement être utilisés en dernier ressort. Noter que les candidates rassemblées sur la base des procédures du paragraphe 5.1.1 ne seront jamais des candidates reflets

d'homologue ; les candidates de ce type sont apprises des vérifications de connexité effectuées par ICE.

La préférence locale DOIT être un entier de 0 (plus faible préférence) à 65535 (plus forte préférence) inclus. Quand il y a seulement une adresse IP, cette valeur DEVRAIT être réglée à 65535. Si il y a plusieurs candidates pour un composant de flux de données particulier qui ont le même type, la préférence locale DOIT être unique pour chacune. Si un agent ICE est en double pile, la préférence locale DEVRAIT être réglée en accord avec les bonnes pratiques courantes décrites dans la [RFC8421].

L'identifiant de composant DOIT être un entier entre 1 et 256 inclus.

5.1.2.2 Lignes directrices pour le choix du type et des préférences locales

Les valeurs RECOMMANDÉES pour les préférences de type sont 126 pour les candidates hôtes, 110 pour les candidates reflets d'homologue, 100 pour les candidates reflets de serveur, et 0 pour les candidates relayées.

Si un agent ICE est multi-rattachements et a plusieurs adresses IP, les recommandations de la [RFC8421] DEVRAIENT être suivies. Si plusieurs serveurs TURN sont utilisés, les priorités locales pour les candidates obtenues des serveurs TURN sont choisies de la même façon que pour les candidates locales multi-rattachements : la valeur de préférence locale est utilisée pour indiquer une préférence parmi différents serveurs, mais la préférence DOIT être unique pour chacun.

Quand ils choisissent les préférences de type, les agents peuvent tenir compte de facteurs comme la latence, les pertes de paquets, le coût, la topologie du réseau, la sécurité, la confidentialité, et d'autres.

5.1.3 Élimination des candidates redondantes

Ensuite, les agents ICE (initiateur et répondant) éliminent les candidates redondantes. Deux candidates peuvent avoir la même adresse de transport mais des bases différentes, et elles ne vont pas être considérées comme redondantes. Fréquemment, une candidate reflet de serveur et une candidate hôte vont être redondantes quand l'agent n'est pas derrière un NAT. Une candidate est redondante si et seulement si son adresse de transport et sa base sont égales à celles d'une autre candidate. L'agent DEVRAIT éliminer la candidate redondante avec la plus faible priorité.

5.2. Procédures de mise en œuvre légère

Les mises en œuvre légères utilisent seulement des candidates hôtes. Pour chaque adresse IP, indépendante d'une famille d'adresses IP, il DOIT y avoir zéro ou une candidate. Avec la mise en œuvre légère, ICE ne peut pas être utilisé pour choisir dynamiquement parmi des candidates. Donc, inclure plus d'une candidate provenant d'une famille d'adresses IP particulière n'est PAS RECOMMANDÉ, car seule une vérification de connexité peut vraiment déterminer si il faut utiliser une adresse ou l'autre. Il est plutôt RECOMMANDÉ que les agents qui ont plusieurs adresses IP publiques fassent des mises en œuvre complètes de ICE pour assurer le meilleur usage de leurs adresses.

Chaque composant a un identifiant qui lui est alloué, appelé "identifiant de composant". Pour les flux de données RTP/RTCP, sauf si RTCP est multiplexé dans le même accès avec RTP, le RTP lui-même a un identifiant de composant de 1 et RTCP un identifiant de composant de 2. Si un agent utilise RTCP sans multiplexage, il DOIT obtenir des candidats pour lui. Cependant, l'absence d'un identifiant de composant de 2 n'implique pas par lui-même l'utilisation du multiplexage RTCP/RTP, car cela pourrait aussi signifier que RTCP n'est pas utilisé.

Une fondation est allouée à chaque candidat. La fondation DOIT être différente pour deux candidates allouées à partir d'adresses IP différentes ; autrement, elle DOIT être la même. Un simple entier qui s'incrémente pour chaque adresse IP va suffire. De plus, chaque candidate DOIT recevoir une priorité unique parmi toutes les candidates pour le même flux de données. Si la formule du paragraphe 5.1.2.1 est utilisée pour calculer la priorité, la valeur de préférence de type DEVRAIT être réglée à 126. Si un hôte est IPv4 seul, la valeur de préférence locale DEVRAIT être réglée à 65535. Si un hôte est IPv6 ou double pile, la valeur de préférence locale DEVRAIT être réglée à la valeur de préséance pour les adresses IP décrite dans la [RFC6724].

Ensuite, un agent choisit une candidate par défaut pour chaque composant de chaque flux de données. Si un hôte est IPv4 seul, il va seulement y avoir une candidate pour chaque composant de chaque flux de données ; donc, cette candidate est celle par défaut. Si un hôte est IPv6 seul, la candidate par défaut va normalement être une adresse IPv6 de portée mondiale. Les hôtes double pile DEVRAIENT permettre la configuration de si IPv4 ou IPv6 est utilisé pour la candidate par défaut, et la configuration doit être fondée sur celle que son administrateur estime avoir la meilleurs chance de succès dans l'environnement de réseau actuel.

Les procédures de ce paragraphe sont communes aux agents initiateur et répondant.

5.3 Échange des informations de candidate

Les agents ICE (initiateur et répondant ont besoin d'échanger les informations suivantes sur les candidats. Chaque usage ICE DOIT définir comment les informations sont échangées avec le protocole d'utilisation. Ce paragraphe décrit les informations qui ont besoin d'être échangées.

Candidates : une ou plusieurs candidates. Pour chaque candidate :
- Adresse : l'adresse IP et l'accès de protocole de transport de la candidate.
- Transport : le protocole de transport de la candidate. Cela PEUT être omis si le protocole utilisateur fonctionne seulement sur un seul protocole de transport.
- Fondation : séquence de jusqu'à 32 caractères.
- Identifiant de composant : l'identifiant de composant de la candidate. Il PEUT être omis si le protocole utilisateur n'utilise pas le concept de composants.
- Priorité : les 32 bits de la priorité de la candidate.
- Type : le type de la candidate.
- Adresse et accès relatifs : adresse IP et accès relatifs de la candidate. Ils PEUVENT être omis ou réglés à des valeurs invalides si l'agent ne veut pas les révéler, par exemple, pour des raisons de confidentialité.
- Paramètres d'extensibilité : le protocole utilisateur pourrait définir des moyens pour ajouter de nouveaux paramètres par candidate ICE à l'avenir.

Léger ou complet : si l'agent est un agent complet ou léger.

Valeur de régulation de vérification de connexité : la valeur de régulation pour les vérifications de connexité que l'agent souhaite utiliser. Elle PEUT être omise si l'agent souhaite utiliser une valeur par défaut définie.

Fragment de nom d'utilisateur et mot de passe : valeurs utilisées pour effectuer les vérifications de connexité. Les valeurs DOIVENT être imprévisibles, avec au moins 128 bits d'un résultat de générateur de nombres aléatoires utilisé pour générer le mot de passe, et au moins 24 bits de résultat pour générer le fragment de nom d'utilisateur.

Extensions : nouveaux attribut de flux de suports ou de niveau session (options ICE).

Si le protocole utilisateur est vulnérable aux discordances de ICE, et capable de les détecter (paragraphe 5.4) un moyen est nécessaire pour que l'agent détecteur porte cette information à son homologue. C'est un fanion booléen.

Le protocole utilisateur peut (ou non) avoir besoin de composer avec la rétro compatibilité avec de plus anciennes mises en œuvre qui ne prennent pas en charge ICE. Si un mécanisme de repli non ICE est pris en charge et est utilisé, alors vraisemblablement le protocole utilisateur fournit un moyen de convoyer la candidate par défaut (son adresse et accès IP) en plus des paramètres ICE.

Une fois qu'un agent a envoyé ses informations de candidates, il DOIT être prêt à recevoir des paquets STUN et de données sur chaque candidate. Comme discuté au paragraphe 12.1, les paquets de données peuvent être envoyés à une candidate avant qu'elle apparaisse comme la destination par défaut pour les données.

5.4 Discordance ICE

Certains boîtiers de médiation, comme des ALG, peuvent altérer les informations de signalisation d'une manière qui casse ICE (par exemple, en réécrivant les adresses IP dans SDP). Ceci est appelé une "discordance ICE". Si le protocole utilisateur est vulnérable à la discordance ICE, l'agent répondant doit être capable de le détecter et d'informer l'agent homologue ICE de la discordance ICE.

Chaque protocole utilisateur doit définir si le protocole utilisateur est vulnérable à la discordance ICE, comment la discordance ICE est détectée, et si des actions spécifiques doivent être prises quand une discordance ICE est détectée.

6. Traitement du candidat ICE

Une fois qu'un agent ICE a rassemblé ses candidates et échangé les candidates avec son homologue (Section 5) il va déterminer son propre rôle. De plus, les mises en œuvre complètes vont former des listes de contrôle et commencer à effectuer les vérifications de connexité avec l'homologue.

6.1 Procédures de mise en œuvre complète

6.1.1 Détermination des rôles

Pour chaque session, chaque agent ICE (initiateur et répondant) prend un rôle. Il y a deux rôles -- contrôleur et contrôlé. L'agent contrôleur est responsable du choix des paires candidates finales utilisées pour les communications. Les paragraphes qui suivent décrivent en détails les procédures réelles suivies par les agents contrôleur et contrôlé.

Les règles pour déterminer le rôle et l'impact sur le comportement sont les suivantes :

Les deux agents sont complets : l'agent initiateur qui a commencé le traitement ICE DOIT prendre le rôle de contrôleur, et l'autre DOIT prendre le rôle de contrôlé. Les deux agents vont former des listes de contrôle, lancer les automates à états ICE, et générer des vérifications de connexité. L'agent contrôleur va exécuter la logique du paragraphe 8.1 pour désigner des paires qui vont devenir (si les vérifications de connexité associées aux désignations réussissent) les paires choisies, et alors les deux agents terminent ICE comme décrit au paragraphe 8.1.2.

Un agent complet, un léger : l'agent complet DOIT prendre le rôle de contrôleur, et l'agent léger DOIT prendre le rôle de contrôlé. L'agent complet va former les listes de contrôle, lancer les automates à états ICE, et générer les vérifications de connexité. Cet agent va exécuter la logique du paragraphe 8.1 pour désigner les paires qui vont devenir (si les vérifications de connexité associées à la désignation réussissent) les paires choisies et utiliser la logique du paragraphe 8.1.2 pour terminer ICE. La mise en œuvre légère va juste écouter les vérifications de connexité, les recevoir et leur répondre, et ensuite conclure ICE comme décrit au paragraphe 8.2. Pour la mise en œuvre légère, l'état du traitement ICE pour chaque flux de données est considéré être Running, et l'état global de ICE est Running.

Deux agents légers : l'agent initiateur qui a commencé le traitement ICE DOIT prendre le rôle de contrôleur, et l'autre DOIT prendre le rôle de contrôlé. Dans ce cas, aucune vérification de connexité n'est jamais envoyée. Plutôt, une fois que les candidats sont échangés, chaque agent effectue le traitement décrit à la Section 8 sans vérifications de connexité. Il est possible que les deux agents croient qu'ils sont le contrôlé ou le contrôleur. Dans ce dernier cas, le conflit est résolu à la lumière des capacités de détection dans le protocole de signalisation qui permet l'échange de candidates. L'état du traitement ICE pour chaque flux de données est considéré comme étant Running, et l'état global de ICE est Running.

Une fois que les rôles sont déterminés pour une session, ils persistent pendant toute la durée de vie de la session. Les rôles peuvent être redéterminés au titre d'un redémarrage ICE (Section 9) mais un agent ICE NE DOIT PAS redéterminer le rôle au titre d'un redémarrage ICE sauf si un ou plusieurs des critères suivants sont satisfaits :

Complet devient léger : si l'agent contrôleur est complet, et passe à léger, les rôles DOIVENT être redéterminés si l'agent homologue est aussi complet.

Conflit de rôles : si le rédémarrage ICE cause un conflit de rôles, les rôles pourraient être redéterminés à cause des procédure de conflit de rôle du paragraphe 7.3.1.1.

Note : il y a certains scénarios de commande d'appel de tiers (3PCC, *Third Party Call Control*) [RFC3725] où un redémarrage ICE pourrait causer un conflit de rôle.

Note : les agents ont besoin de s'informer l'un l'autre de si ils sont complets ou légers avant que les rôles soient déterminés. Le mécanisme pour cela est spécifique du protocole de signalisation et sort du domaine de ce document.

Un agent DOIT accepter si l'homologue initie une redétermination des rôles même si les critères pour le faire ne sont pas remplis. Cela peut arriver si l'homologue est conforme à la RFC 5245.

6.1.2 Formation des listes de contrôle

Il y a une liste de contrôle pour chaque flux de données. Pour former une liste de contrôle, les agents ICE initiateur et répondant forment des paires de candidates, calculent des priorités de paires, ordonnent les paires par priorité, élaguent les paires, suppriment les paires de priorité inférieure, et règlent les états des listes de contrôle. Si des candidates sont ajoutées à une liste de contrôle (par exemple, du fait de la détection de candidates reflets d'homologue) l'agent va effectuer à nouveau ces étapes pour la liste de contrôle mise à jour.

6.1.2.1 État de liste de contrôle

Chaque liste de contrôle a un état, qui capture l'état des vérifications ICE pour le flux de données associé à la liste de contrôle. Les états sont :

Running *(en cours)* : la liste de contrôle n'est ni Completed *(terminé)* ni encore Failed *(échec)*. Les listes de contrôle sont initialement réglées à l'état Running.

Completed *(terminé)* : la liste de contrôle contient une paire désigné pour chaque composant du flux de données.

Failed *(échec)* : la liste de contrôle n'a pas une paire valide pour chaque composant du flux de données, et toutes les paires candidates dans la liste de contrôle sont dans l'état Failed ou Succeeded *(réussi)*. En d'autres termes, au moins un composant de la liste de contrôle a des paires candidates qui sont toutes dans l'état Failed, ce qui signifie que le composant a échoué, et que la liste de contrôle a échoué.

6.1.2.2 Formation de paires de candidates

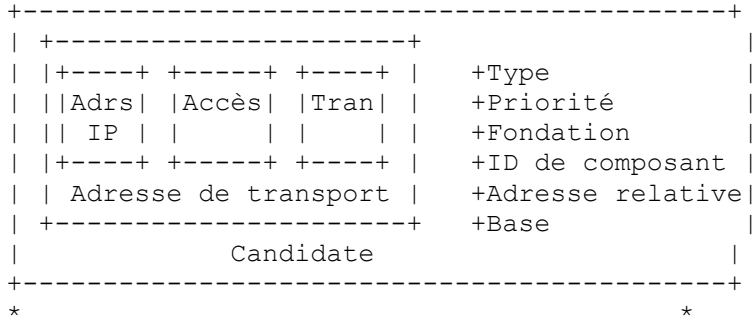
L'agent ICE apparie chaque candidate locale avec chaque candidate distante pour le même composant du même flux de données avec la même famille d'adresses IP. Il est possible que certaines des candidates locales ne soient pas appariées avec des candidates distantes, et que certaines des candidates distantes ne soient pas appariées avec des candidates locales. Cela peut arriver si un agent n'inclut pas de candidates pour tous les composants pour un flux de données. Si cela arrive, le nombre de composants pour ce flux de données est effectivement réduit et est considéré être égal au minimum sur les deux agents de l'identifiant de composant maximum fourni par chaque agent à travers tous les composants pour le flux de données.

Dans le cas de RTP, cela arriverait quand un agent fournit des candidates pour RTCP, et pas l'autre. Dans un autre exemple, l'agent initiateur peut multiplexer RTP et RTCP sur le même accès [RFC5761]. Cependant, comme l'agent initiateur ne sait pas si l'agent homologue peut effectuer un tel multiplexage, il inclut des candidates pour RTP et RTCP sur des accès séparés. Si l'agent homologue peut effectuer un tel multiplexage, il va inclure juste un seul composant pour chaque candidate -- pour le multiplex combiné RTP/RTCP. ICE va finir par agir comme si il y avait juste un seul composant pour cette candidate.

Avec IPv6, il est courant pour un hôte d'avoir plusieurs candidates hôtes pour chaque interface. Pour garder la quantité de paires candidates résultantes raisonnable et éviter des paires candidates qui ont peu de chances de fonctionner, des adresses IPv6 de liaison locale NE DOIVENT PAS être appariées avec d'autres adresses que de liaison locale.

Les paires candidates dont les candidates locales et distantes sont toutes deux les candidates par défaut pour un composant particulier sont appelées la "paire candidate par défaut " pour ce composant. C'est la paire qui va être utilisée pour transmettre les données si les deux agents n'avaient pas été à capacité ICE.

La Figure 5 montre les propriétés et les relations entre adresses de transport, candidates, paires candidates, et listes de contrôle.



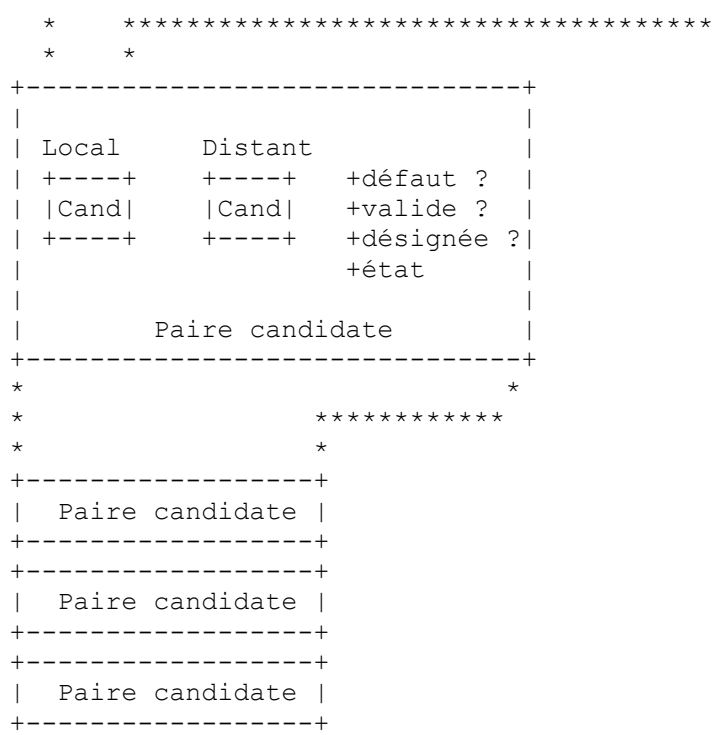


Figure 5 : Diagramme conceptuel d'une liste de contrôle

6.1.2.3 Calcul de la priorité et de l'ordre des paires

L'agent ICE calcule une priorité pour chaque paire candidate. Soit G la priorité pour la candidate fournie par l'agent contrôleur. Soit D la priorité pour la candidate fournie par l'agent contrôlé. La priorité pour une paire est calculée comme suit :

priorité de paire = 2^32*MIN(G,D) + 2*MAX(G,D) + (G>D?1:0)

L'agent trie chaque liste de contrôle en ordre décroissant de priorité de paire candidate. Si deux paires ont une priorité identique, l'ordre entre elles est arbitraire.

6.1.2.4 Élagage des paires

Cette liste de paires candidates triées est utilisée pour déterminer une séquence de vérifications de connexité qui vont être effectuées. Chaque vérification implique l'envoi d'une demande d'une candidate locale à une candidate distante. Comme un agent ICE ne peut pas envoyer directement des demandes à partir d'une candidate reflet (reflet de serveur ou reflet d'homologue) mais seulement à partir de sa base, l'agent passe ensuite par la liste de paires candidates triées. Pour chaque paire où la candidate locale est un reflet, la candidate DOIT être remplacée par sa base.

L'agent élague chaque liste de contrôle. C'est fait en retirant une paire candidate si elle est redondante avec une paire candidate de priorité supérieure dans la même liste de contrôle. Deux paires candidates sont redondantes si leurs candidates locales ont la même base et si leurs candidates distantes sont identiques. Le résultat est une séquence ordonnée de paires candidates, appelée la "liste de contrôle" pour ce flux de données.

6.1.2.5 Suppression des paires de priorité inférieure

Afin de limiter les attaques décrites au paragraphe 19.5.1, un agent ICE DOIT limiter le nombre total de vérifications de connexité que l'agent effectue sur toutes les listes de contrôle dans l'ensemble de listes de contrôle. Ceci est fait en limitant le nombre total de paires candidates dans l'ensemble de listes de contrôle. La limite par défaut de paires candidates pour l'ensemble de listes de contrôle est 100, mais la valeur DOIT être configurable. La limite est appliquée, au sein de chaque liste de contrôle, en éliminant les paires candidates de priorité inférieure jusqu'à ce que le nombre total de paires candidates dans l'ensemble de listes de contrôle soit inférieur à la valeur limite. L'élimination DEVRAIT être faite équitablement afin que le nombre de paires candidates dans chaque liste de contrôle soit réduit du même montant.

Il est RECOMMANDÉ qu'une valeur de limite inférieure à la valeur par défaut soit prise lorsque possible, et que la valeur

soit réglée au nombre maximum de paires candidates plausibles qui pourraient être créées dans une configuration réelle de déploiement. L'exigence d'une configuration est destinée à fournir un outil pour corriger cette valeur dans le champ si, une fois déployée, elle se trouvait être problématique.

6.1.2.6 Calcul de l'état de la paire candidate

Chaque paire candidate dans la liste de contrôle a une fondation (la combinaison de la fondation de la candidate locale et de celle de la candidate distante dans la paire) et un des états suivants :

Waiting *(en attente)* : une vérification n'a pas encore été envoyée pour cette paire, mais la paire n'est pas Frozen.

In-Progress *(en progrès)* : une vérification a été envoyée pour cette paire, mais la transaction est en cours.

Succeeded *(réussie)* : une vérification a été envoyée pour cette paire, et elle a produit un résultat réussi.

Failed *(échec)* : une vérification a été envoyée pour cette paire, et elle a échoué (aucune réponse n'a été reçue à la vérification, ou une réponse d'échec a été reçue).

Frozen *(gelée)* : une vérification n'a pas été envoyée pour cette paire, et elle ne peut pas être envoyée jusqu'à ce que la paire soit dégelée et passé dans l'état En attente.

Les paires passent entre les états comme le montre la Figure 6.

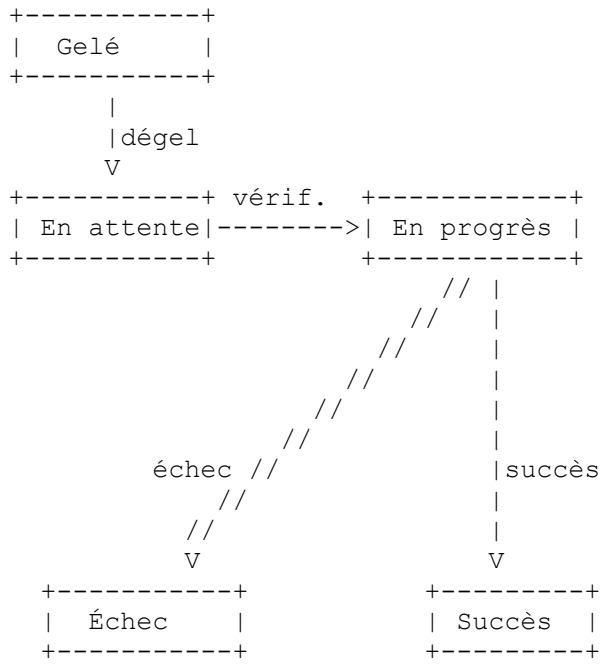


Figure 6 : Automate à états finis d'état de paire

Les états initiaux pour chaque paire dans une liste de contrôle sont calculés en effectuant les étapes suivantes :

1. Les listes de contrôle sont placées dans une liste ordonnée (l'ordre est déterminé par chaque usage de ICE) appelée un "ensemble de listes de contrôle".

2. L'agent ICE place initialement toutes les paires candidates dans l'état Gelé.

3. L'agent règle toutes les listes de contrôle dans l'ensemble de listes de contrôle à l'état En cours.

4. Pour chaque fondation, l'agent règle l'état de exactement une paire candidate à l'état En attente (en la dégelant). La paire candidate à dégeler est choisie en trouvant la première paire candidate (ordonnée par le plus faible identifiant de composant et ensuite la plus haute priorité si les identifiants de composant sont égaux) dans la première liste de contrôle (selon l'ordre de l'ensemble de listes de contrôle défini par l'usage) qui avait cette fondation.

Note : les procédures ci-dessus sont différentes de celles de la RFC 5245, où seules les paires candidates dans la première liste de contrôle étaient initialement placées dans l'état En attente. Maintenant, elles s'appliquent aux paires candidates dans la première liste de contrôle qui ont cette fondation, même si la liste de contrôle n'est pas la première dans l'ensemble de listes de contrôle.

Le tableau ci-dessous illustre un exemple.

Légende du tableau :
Chaque rangée (m1, m2,...) représente une liste de contrôle associée à un flux de données. m1 représente la première liste de contrôle dans l'ensemble de listes de contrôle.
Chaque colonne (f1, f2,...) représente une fondation. Chaque paire candidate dans une colonne partage la même fondation.
f-cp représente une paire candidate dans l'état Gelé.
w-cp représente une paire candidate dans l'état En attente.

1. L'agent met toutes les paires dans l'ensemble de listes de contrôle à l'état Gelé.

	f1	f2	f3	f4	f5
m1	f-cp	f-cp	f-cp		
m2	f-cp	f-cp	f-cp	f-cp	
m3	f-cp			f-cp	

2. Pour chaque fondation, la paire candidate avec le plus faible identifiant de composant est placé dans l'état En attente, jusqu'à ce qu'une paire candidate associée à la même fondation ait déjà été mise dans l'état En attente dans une des autres listes de contrôle examinées dans l'ensemble de listes de contrôle.

	f1	f2	f3	f4	f5
m1	w-cp	w-cp	w-cp		
m2	f-cp	f-cp	f-cp	w-cp	
m3	f-cp				w-cp

Tableau 1 : Exemple d'état de paire

Dans la première liste de contrôle (m1) la paire candidate pour chaque fondation est placée dans l'état En attente, car aucune paire pour la même fondation n'a encore été placée dans l'état En attente.

Dans la seconde liste de contrôle (m2) la paire candidate pour la fondation f4 est placée dans l'état En attente. La paire candidate pour les fondations f1, f2, et f3 est gardée dans l'état Gelé, car les paires candidates pour ces fondations ont déjà été placées dans l'état En attente (dans la liste de contrôle m1).

Dans la troisième liste de contrôle (m3) la paire candidate pour la fondation f5 est placée dans l'état En attente. La paire candidate pour la fondation f1 est gardée dans l'état Gelé, car une paire candidate pour cette fondation a déjà été placée dans l'état En attente (dans la liste de contrôle m1).

Une fois que chaque liste de contrôle a été traitée, une paire candidate pour chaque fondation dans l'ensemble de listes de contrôle a été placé dans l'état En attente.

6.1.3 État ICE

L'agent ICE a un état déterminé par l'état des listes de contrôle. L'état est Terminé si toutes les listes de contrôle sont achevées, Échec si toutes les listes de contrôle sont en échec, ou En cours *(Running)* autrement.

6.1.4 Vérifications de programmation

6.1.4.1 File d'attente à vérification déclenchée

Une fois que l'agent ICE a calculé les listes de contrôle et créé l'ensemble de listes de contrôle, comme décrit au paragraphe 6.1.2, l'agent va commencer à effectuer les vérifications de connexité (ordinaires et déclenchées). Pour les vérifications de connexité déclenchées, l'agent tient une file d'attente FIFO *(premier entré, premier sorti)* pour chaque liste de contrôle, appelée la "file d'attente de vérification déclenchée", qui contient les paires candidates pour lesquelles les vérifications sont à envoyer à la prochaine opportunité disponible. La file d'attente de vérification déclenchée est initialement vide.

6.1.4.2 Réalisation des vérifications de connexité

La génération de vérifications de connexité ordinaires et déclenchées est gouvernée par le temporisateur Ta. Aussitôt que les états initiaux pour les paires candidates dans l'ensemble de listes de contrôle a été établi, une vérification est effectuée pour une paire candidate dans la première liste de contrôle dans l'état En cours *(Running)* suivant les procédures de la Section 7. Après cela, chaque fois que Ta arrive à expiration, on prend la prochaine liste de contrôle dans l'état En cours dans l'ensemble de listes de contrôle, et une vérification est effectuée pour une candidate de cette liste de contrôle. Après le traitement de la dernière liste de contrôle dans l'état En cours dans l'ensemble de listes de contrôle, la première liste de contrôle est reprise, etc.

Chaque fois que Ta arrive à expiration, l'agent ICE va effectuer une vérification pour une paire candidate dans la liste de contrôle qui a été prise en effectuant les étapes suivantes :

1. Si la file d'attente de vérification déclenchée associée à la liste de contrôle contient une ou plusieurs paires candidates, l'agent supprime la paire supérieure de la file d'attente, effectue une vérification de connexité sur cette paire, met l'état de la paire candidate à En progrès *(In-Progress)* et interrompt les étapes suivantes.

2. Si il n'y a pas de paire candidate dans l'état En attente, et si il y a une ou plusieurs paires dans l'état Gelé, l'agent vérifie la fondation associée à chaque paire dans l'état Gelé. Pour une fondation donnée, si il n'y a pas de paire (dans aucune liste de contrôle de l'ensemble de listes de contrôle) dans l'état En attente ou En progrès, l'agent met la paire candidate dans l'état En attente et continue avec l'étape suivante.

3. Si il y a une ou plusieurs paires candidates dans l'état En attente, l'agent prend la paire candidate de plus haute priorité (si il y a plusieurs paires de même priorité, la paire avec le plus faible identifiant de composant est prise) dans l'état En attente, effectue une vérification de connexité sur cette paire, met l'état de la paire candidate à En progrès, et interrompt les étapes suivantes.

4. Si cette étape est atteinte, aucune vérification n'a pu être effectuée pour la liste de contrôle prise. Donc, sans attendre l'expiration du temporisateur Ta, choisir la prochaine liste de contrôle dans l'état En cours et retourner à l'étape n° 1. Si cela arrive pour toutes les listes de contrôle seules dans l'état En cours, ce qui signifie qu'il n'y a plus de paires candidates restantes pour lesquelles effectuer de vérifications de connexité, interrompre ces étapes.

Une fois que l'agent a pris une paire candidate pour laquelle une vérification de connexité est à effectuer, l'agent commence une vérification et envoie la demande Binding à partir de la base associée à la candidate locale de la paire à la candidate distante de la paire, comme décrit au paragraphe 7.2.4.

Sur la base de la politique locale, un agent PEUT choisir de terminer d'effectuer les vérifications de connexité pour une ou plusieurs listes de contrôle dans l'ensemble de listes de contrôle à tout moment. Cependant, seul l'agent contrôleur est admis à terminer ICE (Section 8).

Pour calculer l'intégrité du message pour la vérification, l'agent utilise le fragment de nom d'utilisateur distant et le mot de passe appris des informations de candidate obtenues de son homologue. Le fragment de nom d'utilisateur local est connu directement par l'agent pour sa propre candidate.

6.2 Procédures de mise en œuvre légère

Les mises en œuvre légères sautent la plupart des étapes de la Section 6 sauf pour vérifier la prise en charge de ICE par l'homologue et déterminer son rôle dans le traitement ICE.

Si la mise en œuvre légère est l'agent contrôleur (ce qui n'arrive que si l'agent de l'homologue ICE est aussi une mise en œuvre légère) elle choisit une paire candidate sur la base de celles de l'échange de candidates (pour IPv4, il n'y a jamais qu'une paire) et ensuite met à jour l'homologue avec les nouvelles informations de candidate reflétant ce choix, si nécessaire (ce n'est jamais nécessaire pour un hôte IPv4 seul).

7. Réalisation des vérifications de connexité

Cette section décrit comment les vérifications de connexité sont effectuées.

Un agent ICE DOIT être conforme à la [RFC5389]. Une mise en œuvre complète agit à la fois comme client STUN et

comme serveur STUN, tandis qu'une mise en œuvre légère agit seulement comme serveur STUN (car elle ne génère pas de vérification de connexité).

7.1 Extensions à STUN

ICE étend STUN avec des attributs : PRIORITY, USE-CANDIDATE, ICE-CONTROLLED, et ICE-CONTROLLING. Ces attributs sont formellement définis au paragraphe 16.1. Ce paragraphe décrit l'usage des attributs.

Les attributs sont seulement applicables aux vérifications de connexité ICE.

7.1.1 PRIORITY

L'attribut PRIORITY DOIT être inclus dans une demande Binding et être réglé à la valeur calculée par l'algorithme du paragraphe 5.1.2 pour la candidate locale, mais avec la préférence de type de candidate des candidates reflets d'homologue.

7.1.2 USE-CANDIDATE

L'agent contrôleur DOIT inclure l'attribut USE-CANDIDATE afin de désigner une paire candidate (paragraphe 8.1.1). L'agent contrôlé NE DOIT PAS inclure l'attribut USE-CANDIDATE dans une demande Binding.

7.1.3 ICE-CONTROLLED et ICE-CONTROLLING

L'agent contrôleur DOIT inclure l'attribut ICE-CONTROLLING dans une demande Binding. L'agent contrôlé DOIT inclure l'attribut ICE-CONTROLLED dans une demande Binding.

Le contenu de l'un et l'autre attribut est utilisé comme valeur de départage quand un conflit de rôle ICE se produit (paragraphe 7.3.1.1).

7.2 Procédures de client STUN

7.2.1 Création de permissions pour les candidates relayées

Si la vérification de connexité est envoyée en utilisant une candidate locale relayée, le client DOIT créer d'abord une permission si il n'en a pas déjà créé une précédemment. Il en aurait créé une précédemment si il avait dit au serveur TURN de créer une permission pour la candidate relayée en question sur l'adresse IP de la candidate distante. Pour créer la permission, l'agent ICE suit les procédures définies dans la [RFC5766]. La permission DOIT être créée sur l'adresse IP de la candidate distante. Il est RECOMMANDÉ que l'agent diffère la création d'un canal TURN jusqu'à la fin de ICE, auquel cas les permissions pour les vérifications de connexité sont normalement créées en utilisant une demande CreatePermission. Une fois établie, l'agent DOIT garder la permission active jusqu'à la fin de ICE.

7.2.2 Formation d'accréditifs

Une demande Binding de vérification de connexité DOIT utiliser le mécanisme d'accréditif STUN à court terme.

Le nom d'utilisateur pour l'accréditif est formé en enchaînant le fragment de nom d'utilisateur fourni par l'homologue avec le fragment de nom d'utilisateur de l'agent ICE qui envoie la demande, séparée par un caractère deux-points (":").

Le mot de passe est égal au mot de passe fourni par l'homologue.

Par exemple, considérons le cas où l'agent ICE L est l'agent initiateur et l'agent ICE R est l'agent répondant. L'agent L a inclus un fragment de nom d'utilisateur de LFRAG pour ses candidates et un mot de passe de LPASS. L'agent R a fourni un fragment de nom d'utilisateur de RFRAG et un mot de passe de RPASS. Une vérification de connexité de L à R utilise le nom d'utilisateur RFRAG:LFRAG et un mot de passe de RPASS. Une vérification de connexité de R à L utilise le nom d'utilisateur LFRAG:RFRAG et un mot de passe de LPASS. Les réponses utilisent les mêmes noms d'utilisateur et mots de passe que les demandes (noter que l'attribut USERNAME n'est pas présent dans la réponse).

7.2.3 Traitement Diffserv

Si l'agent utilise les marquage de codet de service différenciés (DSCP, *Differentiated Services Code Point*) [RFC2475] dans les paquets de données qu'il va envoyer, l'agent DEVRAIT appliquer les mêmes marquages aux demandes et réponses Binding qu'il va envoyer.

Si plusieurs marquages DSCP sont utilisés sur les paquets de données, l'agent DEVRAIT choisir l'un d'eux pour l'utiliser avec la vérification de connexité.

7.2.4 Envoi de la demande

Une vérification de connexité est générée par l'envoi d'une demande Binding à partir de la base associée à une candidate locale à une candidate distante. La [RFC5389] décrit comment les demandes Binding sont construites et générées.

La prise en charge de la rétro compatibilité avec la RFC 3489 NE DOIT PAS être supposée quand on effectue les vérifications de connexité. Le mécanisme FINGERPRINT DOIT être utilisé pour les vérifications de connexité.

7.2.5 Traitement de la réponse

Ce paragraphe définit des procédures supplémentaires pour le traitement des réponses Binding spécifiques des vérifications de connexité ICE.

Quand une réponse Binding est reçue, elle est corrélée à la demande Binding correspondante en utilisant l'identifiant de transaction [RFC5389], qui associe alors la réponse à la paire candidate pour laquelle la demande Binding a été envoyée. Après cela, la réponse est traitée en accord avec les procédures de conflit de rôle, d'échec, ou de succès, selon les procédures ci-dessous.

7.2.5.1 Conflit de rôle

Si la demande Binding génère une réponse d'erreur 487 (Conflit de rôle) (paragraphe 7.3.1.1) et si l'agent ICE a inclus un attribut ICE-CONTROLLED dans la demande, l'agent DOIT passer au rôle de contrôleur. Si l'agent a inclus un attribut ICE-CONTROLLING dans la demande, l'agent DOIT passer au rôle de contrôlé.

Une fois que l'agent est passé à son rôle, l'agent DOIT ajouter la paire candidate dont la vérification a généré la réponse d'erreur 487 à la file d'attente de vérification déclenchée associée à la liste de contrôle à laquelle appartient la paire, et régler l'état de la paire candidate à En attente. Quand la vérification de connexité déclenchée est ensuite effectuée, l'attribut ICE-CONTROLLING/ICE-CONTROLLED de la demande Binding va indiquer le nouveau rôle de l'agent. L'agent DOIT changer la valeur de départage.

Note : un changement de rôle exige d'un agent qu'il recalcule les priorités de paires (paragraphe 6.1.2.3) car les valeurs de priorité dépendent du rôle.

Note : un changement de rôle va aussi impacter si l'agent est responsable de la désignation des paires candidates, et de si l'agent est responsable de l'initiation de l'échange des informations de candidate mises à jour avec l'homologue une fois que ICE est terminé.

7.2.5.2 Échec

Ce paragraphe décrit les cas où l'état de la paire candidate est réglé à Échec.

Note : quand l'agent ICE règle l'état de la paire candidate à Échec par suite d'une erreur de vérification de connexité, l'agent ne change pas l'état des autres paires candidates avec la même fondation.

7.2.5.2.1 Adresses de transport non symétriques

L'agent ICE DOIT vérifier que les adresses de transport de source et destination dans la demande et réponse Binding sont symétriques. C'est-à-dire, l'adresse et accès IP de source de la réponse DOIVENT être égales à l'adresse et accès IP de destination à laquelle la demande Binding était envoyée, et l'adresse et accès IP de destination de la réponse DOIVENT être égales à l'adresse et accès IP de source d'où la demande Binding a été envoyée. Si les adresses ne sont pas symétriques, l'agent DOIT régler l'état de la paire candidate à Échec.

7.2.5.2.2 Erreur ICMP

Un agent ICE PEUT prendre en charge le traitement des erreurs ICMP pour les vérifications de connexité. Si l'agent prend en charge le traitement des erreurs ICMP, et si une demande Binding génère une erreur ICMP dure, l'agent DEVRAIT régler l'état de la paire candidate à Échec. Les mises en œuvre doivent savoir que les erreurs ICMP peuvent être utilisées comme une méthode d'attaque de déni de service (DoS, *Denial-of-Service*) quand on prend une décision sur comment et si on traite les erreurs ICMP.

7.2.5.2.3 Fin de temporisation

Si la transaction de demande Binding arrive en fin de temporisation, l'agent ICE DOIT régler l'état de la paire candidate à Échec.

7.2.5.2.4 Réponse STUN irrécupérable

Si la demande Binding génère une réponse d'erreur STUN qui est irrécupérable [RFC5389], l'agent ICE DEVRAIT régler l'état de la paire candidate à Échec.

7.2.5.3 Succès

Une vérification de connexité est considérée comme un succès si chacun des critères suivants est vrai :
- La demande Binding a généré une réponse de succès ; et
- Les adresses de transport de source et destination dans la demande et réponse Binding sont symétriques.

Si une vérification est considérée comme réussie, l'agent ICE effectue (dans l'ordre) les actions décrites dans les paragraphes qui suivent.

7.2.5.3.1 Découverte des candidates reflets d'homologue

L'agent ICE DOIT vérifier l'adresse transposée de la réponse STUN. Si l'adresse de transport ne correspond pas à une des candidates locales connues de l'agent, l'adresse transposée représente une nouvelle candidate : une candidate reflet de l'homologue. Comme les autres candidates, une candidate reflet de l'homologue a un type, une base, une priorité, et une fondation. Elles sont calculées comme suit :
- Le type est reflet d'homologue.
- La base est la candidate locale de la paire candidate d'où la demande Binding a été envoyée.
- La priorité est la valeur de l'attribut PRIORITY dans la demande Binding.
- La fondation est décrite au paragraphe 5.1.1.3.

La candidate reflet de l'homologue est alors ajoutée à la liste des candidates locales pour le flux de données. Le fragment de nom d'utilisateur et le mot de passe sont les mêmes que pour toutes les autres candidates locales pour ce flux de données.

L'agent ICE n'a pas besoin d'apparier la candidate reflet de l'homologue avec des candidates distantes, car une paire valide va être créée du fait des procédures du paragraphe 7.2.5.3.2. Si un agent souhaite apparier la candidate reflet de l'homologue avec des candidates distantes autres que celle de la paire valide qui va être générée, l'agent PEUT fournir des informations de candidate mises à jour à l'homologue qui incluent la candidate reflet de l'homologue. Cela va causer l'appariement de la candidate reflet de l'homologue avec toutes les autres candidates distantes.

7.2.5.3.2 Construction d'une paire valide

L'agent ICE construit une paire candidate dont la candidate locale est égale à l'adresse transposée de la réponse et dont la candidate distante est égale à l'adresse de destination à laquelle la demande a été envoyée. C'est appelé une "paire valide".

La paire valide pourrait être égale à la paire qui a généré la vérification de connexité, à une paire différente dans la liste de contrôle, ou à une paire qui n'est actuellement pas dans la liste de contrôle.

L'agent tient une liste séparée, appelée la "liste valide". Il y a une liste valide pour chaque liste de contrôle dans l'ensemble de listes de contrôle. La liste valide va contenir les paires valides. Initialement, chaque liste valide est vide.

Chaque paire valide dans la liste valide a un fanion, appélé le "fanion Désigné". Quand une paire valide est ajoutée à une liste valide, la valeur du fanion est réglée à "faux".

La paire valide va être ajoutée à une liste valide comme suit :

1. Si la paire valide est égale à la paire qui a généré la vérification, la paire est ajoutée à la liste valide associée à la liste de contrôle à laquelle la paire appartient ; ou

2. Si la paire valide est égale à une autre paire dans une liste de contrôle, cette paire est ajoutée à la liste valide associée à la liste de contrôle de cette paire. La paire qui a généré la vérification n'est pas ajoutée à une liste valide ; ou

3. Si la paire valide n'est dans aucune liste de contrôle, l'agent calcule la priorité pour la paire sur la base de la priorité de chaque candidate, en utilisant l'algorithme du paragraphe 6.1.2. La priorité de la candidate locale dépend de son type. Sauf si le type est reflet d'homologue, la priorité est égale à la priorité signalée pour cette candidate dans l'échange de candidates. Si le type est reflet d'homologue, elle est égale à l'attribut PRIORITY que l'agent a placé dans la demande Binding qui vient de s'achever. La priorité de la candidate distante est tirée des informations de candidate de l'homologue. Si la candidate n'apparaît pas là, alors la vérification a été déclenchée sur une nouvelle candidate distante. Dans ce cas, la priorité est prise comme la valeur de l'attribut PRIORITY dans la demande Binding qui a déclenché la vérification qui vient de s'achever. La paire est alors ajoutée à la liste valide.

Note : Il va arriver très souvent que la paire valide ne soit dans aucune liste de contrôle. On rappelle que la liste de contrôle a des paires dont les candidates locales ne sont jamais des reflets ; ces paires ont leurs candidates locales converties en la base des candidates reflets et ont été alors élaguées si elles étaient redondantes. Quand la réponse à la demande Binding arrive, l'adresse transposée va être un reflet si il y a un NAT entre les deux. Dans ce cas, la paire valide va avoir une candidate locale qui ne correspond à aucune des paires de la liste de contrôle.

7.2.5.3.3 Mise à jour des états d'une paire de candidats

L'agent ICE règle l'état de la paire candidate qui a généré la vérification et de la paire valide construite (qui peut être différente) à Succès.

L'agent DOIT régler l'état pour toutes les autres paires candidates gelées dans toutes les listes de contrôle avec la même fondation à En attente.

Note : dans une liste de contrôle donnée, les paires candidates avec la même fondation vont normalement avoir des valeurs d'identifiant de composant différentes.

7.2.5.3.4 Mise à jour du fanion Désigné

Si l'agent contrôleur envoie une demande Binding avec l'attribut USE-CANDIDATE établi, et si l'agent ICE reçoit une réponse de succès à la demande, l'agent règle le fanion Désigné de la paire à vrai. Si la demande échoue (paragraphe 7.2.5.2) l'agent DOIT supprimer la paire candidate de la liste valide, régler l'état de la paire candidate à Échec, et régler l'état de la liste de contrôle à Échoué.

Si l'agent contrôlé reçoit une réponse de succès à une demande Binding envoyée par l'agent, et si cette demande Binding a été déclenchée par une demande Binding reçue avec l'attribut USE-CANDIDATE établi (paragraphe 7.3.1.4) l'agent règle le fanion Désigné de la paire à vrai. Si la demande déclenchée échoue, l'agent DOIT supprimer la paire candidate de la liste valide, régler l'état de la paire candidate à Échec, et régler l'état de la liste de contrôle à Échoué.

Une fois que le fanion Désigné est établi pour un composant d'un flux de données, il termine le traitement ICE pour ce composant (Section 8).

7.2.5.4 Mise à jour de l'état de liste de contrôle

Sans considération de si une vérification de connexité a réussi ou échoué, l'achèvement de la vérification peut exiger de mettre à jour l'état de la liste de contrôle. Pour chaque liste de contrôle dans l'ensemble de listes de contrôle, si toutes les paires candidates sont dans l'état Échec ou Succès, et si il n'y a pas une paire valide dans la liste valide pour chaque composant du flux de données associé à la liste de contrôle, l'état de la liste de contrôle est réglé à Échec. Si il y a une paire valide pour chaque composant dans la liste valide, l'état de la liste de contrôle est réglé à Succès.

7.3 Procédures de serveur STUN

Un agent ICE (léger ou complet) DOIT être prêt à recevoir des demandes Binding sur la base de chaque candidate qu'il a incluse dans son plus récent échange de candidates.

L'agent DOIT utiliser le mécanisme d'accréditif à court terme (c'est-à-dire, l'attribut MESSAGE-INTEGRITY) pour authentifier la demande et effectuer la vérification de l'intégrité d'un message. De même, le mécanisme d'accréditif à court terme DOIT être utilisé pour la réponse. L'agent DOIT considérer que le nom d'utilisateur est valide si il consiste en deux valeurs séparées par deux-points, où la première valeur est égale au fragment de nom d'utilisateur généré par l'agent dans un échange de candidates pour une session en progrès. Il est possible (et en fait très probable) que l'agent initiateur va recevoir une demande Binding avant de recevoir les candidates de son homologue. Si cela arrive, l'agent DOIT immédiatement générer une réponse (incluant le calcul d' l'adresse transposée comme décrit au paragraphe 7.3.1.2). L'agent a des informations suffisantes à ce point pour générer la réponse ; le mot de passe de l'homologue n'est pas nécessaire. Une fois la réponse reçue, il DOIT poursuivre avec les étapes restantes, à savoir celles des paragraphes 7.3.1.3, 7.3.1.4, et 7.3.1.5 pour les mises en œuvre complètes. Dans les cas où plusieurs demandes STUN sont reçues avant la réponse, cela peut causer la mise en file d'attente de plusieurs paires dans la file d'attente de vérifications déclenchées.

Un agent NE DOIT PAS utiliser le mécanisme ALTERNATE-SERVER et NE DOIT PAS prendre en charge les mécanismes de rétro compatibilité définis dans la RFC 5389 (pour travailler avec le protocole de la RFC 3489). Il DOIT utiliser le mécanisme FINGERPRINT.

Si l'agent utilise les marquages DSCP [RFC2475] dans ses paquets de données, il DEVRAIT appliquer les mêmes marquages aux réponses Binding. La même chose s'appliquerait à tous marquages de couche 2 que le point d'extrémité pourrait appliquer aux paquets de données.

7.3.1 Procédures supplémentaires pour mise en œuvre complète

Ce paragraphe définit les procédures supplémentaires de serveur applicables aux mises en œuvre complètes, quand elles acceptent la demande Binding.

7.3.1.1 Détection et réparation des conflits de rôle

Dans certains usages de ICE (comme 3PCC) les deux agents ICE peuvent finir par choisir le même rôle, résultant en un conflit de rôle. Ce paragraphe décrit un mécanisme pour détecter et réparer les conflits de rôle. Le document d'usage DOIT spécifier si ce mécanisme est nécessaire.

Un agent DOIT examiner si la demande Binding comporte l'attribut ICE-CONTROLLING ou ICE-CONTROLLED. Il DOIT suivre ces procédures :

- Si l'agent est dans le rôle de contrôleur, et si l'attribut ICE-CONTROLLING est présent dans la demande :
  * Si la valeur de départage de l'agent est supérieure ou égale au contenu de l'attribut ICE-CONTROLLING, l'agent génère une réponse d'erreur Binding et inclut un attribut ERROR-CODE de valeur 487 (Conflit de rôle) mais conserve son rôle.
  * Si la valeur de départage de l'agent est inférieure au contenu de l'attribut ICE-CONTROLLING, l'agent passe au rôle de contrôlé.

- Si l'agent est dans le rôle contrôlé, et si l'attribut ICE-CONTROLLED est présent dans la demande :
  * Si la valeur de départage de l'agent est supérieure ou égale au contenu de l'attribut ICE-CONTROLLING, l'agent passe au rôle de contrôleur.
  * Si la valeur de départage de l'agent est inférieure au contenu de l'attribut ICE-CONTROLLING, l'agent génère une réponse d'erreur Binding et inclut un attribut ERROR-CODE de valeur 487 (Conflit de rôle) mais conserve son rôle.

- Si l'agent est dans le rôle contrôlé, et si l'attribut ICE-CONTROLLING était présent dans la demande, ou si l'agent était dans le rôle de contrôleur et si l'attribut ICE-CONTROLLED était présent dans la demande, il n'y a pas de conflit.

Un changement des rôles va exiger d'un agent qu'il recalcule les priorités des paires (paragraphe 6.1.2.3) car ces priorités sont fonction du rôle. Le changement de rôle va aussi impacter si l'agent est responsable du choix des paires désignées et d'initier l'échange avec les informations de candidates à la conclusion de ICE.

Les paragraphes restants de paragraphe 7.3.1 sont suivis si l'agent a généré une réponse de succès à la demande Binding, même si l'agent a changé de rôle.

7.3.1.2 Calcul des adresses transposées

Pour les demandes reçues sur une candidate relayée, l'adresse de transport de source utilisée pour le traitement STUN (à

savoir, la génération de l'attribut XOR-MAPPED-ADDRESS) est l'adresse de transport telle que vue par le serveur TURN. Cette adresse de transport de source va être présente dans l'attribut XOR-PEER-ADDRESS d'un message Indication de données, si la demande Binding a été livrée par une indication de données. Si la demande Binding a été livrée par un message ChannelData, l'adresse de transport de source est celle qui était liée au canal.

7.3.1.3 Apprendre les candidates reflet d'homologue

Si l'adresse de transport de source de la demande ne correspond pas à une candidate distante existante, elle représente une nouvelle candidate distante reflet d'homologue. Cette candidate est construite comme suit :

- Le type est reflet d'homologue.
- La priorité est la valeur de l'attribut PRIORITY dans la demande Binding.
- La fondation est une valeur arbitraire, différente de la fondation de tous les autres candidates distantes. Si un échange de candidates ultérieur contient cette candidate reflet de l'homologue, il va signaler la fondation réelle pour la candidate.
- L'identifiant de composant est celui de la candidate locale à laquelle la demande était envoyée.

Cette candidate est ajoutée à la liste des candidates distantes. Cependant, l'agent ICE n'apparie pas cette candidate à une candidate locale.

7.3.1.4 Vérifications déclenchées

Ensuite, l'agent construit une paire dont la candidate locale a l'adresse de transport (comme vue par l'agent) sur laquelle la demande STUN a été reçue et dont la candidate distante est égale à l'adresse de transport de source d'où vient la demande (qui peut être la candidate distante reflet de l'homologue qui vient d'être apprise). La candidate locale va être soit une candidate hôte (dans les cas où la demande n'a pas été reçue à travers un relais) soit une candidate relayée (pour les cas où elle est reçue à travers un relais). La candidate locale ne peut jamais être une candidate reflet de serveur. Comme les deux candidates sont connues de l'agent, il peut obtenir leurs priorités et calculer la priorité de la paire candidate. Cette paire est alors cherchée dans la liste de contrôle. Il peut y avoir un des résultats suivants :

- Quand la paire est déjà dans la liste de contrôle :
  - Si l'état de cette paire est Succès, rien d'autre n'est fait.
  - Si l'état de cette paire est En-progrès, l'agent annule la transaction En-Progrès. Annulation signifie que l'agent ne va pas retransmettre les demandes Binding associées à la transaction de vérification de connexité, ne va pas traiter l'absence de réponse comme un échec, mais va attendre la durée de la temporisation de la transaction pour une réponse. De plus, l'agent DOIT mettre en file d'attente la paire dans la liste de contrôle déclenchée associée à la liste de contrôle, et régler l'état de la paire à En attente, afin de déclencher une nouvelle vérification de connexité de la paire. La création d'une nouvelle vérification de connexité permet de valider les paires En-Progrès aussitôt que possible, sans avoir à attendre des retransmissions des demandes Binding associées à la transaction originale de vérification de connexité.
  - Si l'état de cette paire est En attente, Gelé, ou Échec, l'agent DOIT mettre en file d'attente la paire dans la liste de contrôle déclenchée associée à la liste de contrôle (si elle n'y est pas déjà présente) et régler l'état de la paire à En attente, afin de déclencher une nouvelle vérification de connexité de la paire. Noter qu'un changement d'état de la paire de Échec à En attente pourrait aussi déclencher un changement d'état de la liste de contrôle associée.

  Ces étapes sont faites pour faciliter un achèvement rapide de ICE quand les deux agents sont derrière un NAT.

- Si la paire n'est pas déjà sur la liste de contrôle :
  - La paire est insérée dans la liste de contrôle sur la base de sa priorité.
  - Son état est réglé à En attente.
  - La paire est mise en file d'attente dans la file d'attente de vérifications déclenchées.

Quand une vérification déclenchée doit être envoyée, elle est construite et traitée comme décrit au paragraphe 7.2.4. Ces procédures exigent que l'agent connaisse l'adresse de transport, le fragment de nom d'utilisateur, et le mot de passe pour l'homologue. Le fragment de nom d'utilisateur pour la candidate distante est égal à la partie après les deux-points du USERNAME dans la demande Binding qui vient d'être reçue. En utilisant ce fragment de nom d'utilisateur, l'agent peut vérifier les candidates reçues de son homologue (il peut y en avoir plus d'une en cas de fourchement) et trouver ce fragment de nom d'utilisateur. Le mot de passe correspondant est alors pris.

7.3.1.5 Mise à jour du fanion Désigné

Si l'agent contrôlé reçoit une demande Binding avec l'attribut USE-CANDIDATE établi, et si l'agent ICE accepte la demande, l'action suivante est fondée sur l'état de la paire calculé au paragraphe 7.3.1.4 :

- Si l'état de cette paire est Succès, cela signifie que la vérification précédemment envoyée par cette paire a produit une réponse réussie et a généré une paire valide (paragraphe 7.2.5.3.2). L'agent règle la valeur du fanion Désigné de la paire valide à vrai.

- Si la demande Binding reçue a déclenché une nouvelle vérification à mettre en file d'attente dans la file d'attente de vérifications déclenchées (paragraphe 7.3.1.4) une fois que la vérification est envoyée et si elle génère une réponse de succès, et génère une paire valide, l'agent règle le fanion Désigné de la paire à vrai. Si la demande échoue (paragraphe 7.2.5.2) l'agent DOIT supprimer la paire candidate de la liste valide, régler l'état de la paire candidate à Échoué, et régler l'état de la liste de contrôle à Échoué.

Si l'agent contrôlé n'accepte pas la demande provenant de l'agent contrôleur, l'agent contrôlé DOIT rejeter la demande de désignation avec un code de réponse d'erreur approprié (par exemple, 400) [RFC5389].

Une fois le fanion Désigné établi pour un composant d'un flux de données, cela termine le traitement ICE pour ce composant. Voir la Section 8.

7.3.2 Procédures supplémentaires pour mises en œuvre légères

Si l'agent contrôlé reçoit une demande Binding avec l'attribut USE-CANDIDATE établi, et si l'agent ICE accepte la demande, l'agent construit une paire candidate dont la candidate locale a l'adresse de transport sur laquelle la demande a été reçue, et dont la candidate distante est égale à l'adresse de transport de source de la demande qui a été reçue. Cette paire candidate reçoit une priorité arbitraire et est placée dans la liste valide de la liste de contrôle associée. L'agent règle le fanion Désigné pour cette paire à vrai.

Une fois de fanion Désigné établi pour un composant d'un flux de données, cela conclut le traitement ICE pour ce composant. Voir la Section 8.

8. Conclusion du traitement ICE

Cette Section décrit comment un agent ICE termine ICE.

8.1 Procédures pour les mises en œuvre complètes

Terminer ICE implique de désigner les paires par l'agent contrôleur et de mettre à jour les automates à états.

8.1.1 Désignation des paires

Avant les désignations, l'agent contrôleur laisse les vérifications de connexité se continuer jusqu'à ce qu'un critère d'arrêt soit satisfait. Après cela, sur la base d'un critère d'évaluation, l'agent contrôleur prend une paire parmi les paires valides dans la liste valide pour la désignation.

Une fois que l'agent contrôleur a pris une paire valide pour la désignation, il répète la vérification de connexité qui a produit cette paire valide (en mettant en file d'attente la paire qui a généré la vérification dans la file d'attente de vérification déclenchée) cette fois avec l'attribut USE-CANDIDATE (paragraphe 7.2.5.3.4). Les procédures pour l'agent contrôlé sont décrites au paragraphe 7.3.1.5.

Finalement, si les désignations réussissent, l'agent contrôleur et l'agent contrôlé vont tous deux avoir une seule paire désignée dans la liste valide pour chaque composant du flux de données. Une fois qu'un agent ICE a réglé l'état de la liste de contrôle à Terminé (quand il y a une paire désignée pour chaque composant du flux de données) cette paire devient la paire choisie pour cet agent et elle est utilisée pour envoyer et recevoir les données pour ce composant du flux de données.

Si un agent n'est pas capable de produire des paires choisies pour chaque composant d'un flux de données, l'agent DOIT prendre les actions appropriées pour informer l'autre agent, par exemple, en supprimant le flux. Les actions exactes sortent du domaine d'application de cette spécification.

Les critères pour arrêter les vérifications de connexité et pour prendre une paire pour la désigner sortent du domaine d'application de cette spécification. Ils sont une matière d'optimisation locale. La seule exigence est que l'agent DOIT

finalement prendre une et seulement une paire candidate et générer une vérification pour cette paire avec l'attribut USE-CANDIDATE établi.

Une fois que l'agent contrôleur a réussi à désigner une paire candidate (paragraphe 7.2.5.3.4) l'agent NE DOIT PAS désigner une autre paire pour le même composant du flux de données au sein de la session ICE. Le faire exige un redémarrage de ICE.

Un agent contrôleur qui ne prend pas en charge la présente spécification (c'est-à-dire, il est mis en œuvre conformément à la RFC 5245) pourrait désigner plus d'une paire candidate. Cela était appelé une "désignation agressive" dans la RFC 5245. Si plus d'une paire candidate est désignée par l'agent contrôleur, et si l'agent contrôlé accepte plusieurs demandes de désignation, les agents DOIVENT produire les paires choisies et utiliser les paires de plus haute priorité.

L'usage de l'option ICE "ice2" (Section 10) par les points d'extrémité qui prennent en charge la présente spécification est supposé empêcher les agents contrôleurs qui sont mis en œuvre conformément à la RFC 5245 d'utiliser la désignation agressive.

Note : dans la RFC 5245, l'usage de la "désignation agressive" permettait aux agents de désigner continuellement des paires, avant qu'une paire soit finalement choisie, afin de permettre d'envoyer des données sur ces paires. Dans la présente spécification, les données peuvent toujours être envoyées sur toute paire valide, sans désignation. Donc, il n'y a plus besoin de désignation agressive.

8.1.2 Mise à jour de la liste de contrôle et des états ICE

Pour un agent contrôleur comme pour un agent contrôlé, quand une paire candidate pour un composant d'un flux de données est désignée, cela pourrait impacter d'autres paires dans la liste de contrôle associée au flux de données. Cela pourrait aussi impacter l'état de la liste de contrôle :

- Une fois qu'une paire candidate pour un composant d'un flux de données a été désignée, et que l'état de la liste de contrôle associée au flux de données est En cours, l'agent ICE DOIT supprimer toutes les paires candidates pour le même composant de la liste de contrôle et de la file d'attente de vérifications déclenchées. Si l'état d'une paire est En-progrès, l'agent annule la transaction en progrès. Annulation signifie que l'agent ne va pas retransmettre les demandes Binding associées à la transaction de vérification de connexité, ne va pas traiter l'absence de réponse comme un échec, mais va attendre une réponse pendant la durée de la temporisation de transaction.

- Une fois que les paires candidates pour chaque composant d'un flux de données ont été désignées, et que l'état de la liste de contrôle associée au flux de données est Running, l'agent ICE règle l'état de la liste de contrôle à Terminé.

- Une fois qu'une paire candidate pour un composant d'un flux de données a été désignée, un agent DOIT continuer de répondre à toute demande Binding qu'il pourrait encore recevoir pour la paire désignée et pour toutes paires candidates restantes dans la liste de contrôle associée au flux de données. Comme défini au paragraphe 7.3.1.4, quand l'état d'une paire est Succès, un agent ne va plus générer de vérifications déclenchées quand il reçoit une demande Binding pour la paire.

Une fois que l'état de chaque liste de contrôle dans l'ensemble de listes de contrôle est Terminé, l'agent règle l'état de la session ICE à Terminé.

Si l'état d'une liste de contrôle est Échec, ICE n'a pas été capable de terminer avec succès le processus pour le flux de données associé à la liste de contrôle. Le comportement correct dépend de l'état des listes de contrôle dans 'ensemble de listes de contrôle. Si l'agent contrôleur veut continuer la session sans le flux de données associé à la liste de contrôle défaillante, et si il y a encore une ou plusieurs listes de contrôle en mode En cours ou Terminé, l'agent peut laisser le processus ICE se continuer. L'agent DOIT prendre les actions appropriées pour supprimer le flux de données en échec. Si l'agent contrôleur ne veut pas continuer la session et DOIT la terminer, l'état de la session ICE est réglé à Échec.

Si l'état de chaque liste de contrôle dans l'ensemble de listes de contrôle est Échec, l'état de la session ICE est réglé à Échec. Sauf si l'agent contrôleur veut continuer la session sans le flux de données, il DOIT terminer la session.

8.2 Procédures pour les mises en œuvre légères

Quand ICE se termine, un agent ICE léger peut libérer les candidates hôtes qui n'ont pas été utilisées par ICE, comme décrit

au paragraphe 8.3.

Si l'homologue est un agent complet, une fois que l'agent léger accepte une demande de désignation pour une paire candidate, l'agent léger considère la paire comme désignée. Une fois qu'il y a des paires désignées pour chaque composant d'un flux de données, les paires deviennent les paires choisies pour les composants du flux de données. Une fois que l'agent léger a produit des paires choisies pour tous les composants de tous les flux de données, l'état de la session ICE est réglé à Terminé.

Si l'homologue est un agent léger, l'agent apparie les candidates locales avec les candidates distantes qui sont du même flux de données et ont le même composant, protocole de transport, et famille d'adresse IP. Pour chaque composant de chaque flux de données, si il y a seulement une paire candidate, cette paire est ajoutée à la liste valide. Si il y a plus d'une paire, il est RECOMMANDÉ qu'un agent suive les procédures de la [RFC6724] pour choisir une paire et l'ajoute à la liste valide.

Si tous les composants pour tous les flux de données ont une paire, l'état du traitement ICE est Terminé. Autrement, l'agent contrôleur DOIT envoyer une liste de candidates mise à jour pour réconcilier les différents agents qui choisissent les différentes paires candidates. Le traitement ICE est achevé après et seulement après l'achèvement de la mise à jour de l'échange de candidats.

8.3 Libération des candidates

8.3.1 Procédures pour les mises en œuvre complètes

Les règles de ce paragraphe décrivent quand il est sûr pour un agent de cesser d'envoyer ou recevoir des vérifications sur une candidate qui n'est pas devenue candidate choisie (c'est-à-dire, n'est pas associée à une paire choisie) et quand libérer cette candidate.

Une fois qu'une liste de contrôle a atteint l'état Terminé, l'agent DEVRAIT attendre trois secondes de plus, et ensuite il peut cesser de répondre aux vérifications ou de générer des vérifications déclenchées sur toutes les candidates locales autres que celles qui sont devenues candidates choisies. Une fois que toutes les sessions ICE ont cessé d'utiliser une certaine candidate locale (une candidate peut être utilisée par plusieurs sessions ICE, par exemple, dans des scénarios de fourchement) l'agent peut libérer cette candidate. Le délai de trois secondes traite les cas où la désignation agressive est utilisée, et où les paires choisies peuvent changer rapidement après que ICE s'est terminé.

La libération des candidates reflets de serveur n'est jamais explicite ; elle survient par manque de maintien en vie.

8.3.2 Procédures pour les mises en œuvre complètes

Une mise en œuvre légère peut libérer les candidates qui ne sont pas devenues des candidates choisies aussitôt que le traitement ICE a atteint l'état Terminé pour toutes les sessions ICE qui utilisent ces candidates.

9. Redémarrages de ICE

Un agent ICE PEUT redémarrer ICE pour des flux de données existants. Un redémarrage ICE cause la purge de tous les états précédents du flux de données, sauf les rôles des agents. La seule différence entre un redémarrage ICE et une nouvelle session de données est que durant le redémarrage, les données peuvent continuer d'être envoyées en utilisant les sessions de données existantes, et qu'une nouvelle session de données exige toujours de déterminer les rôles.

Les actions suivantes peuvent être accomplies seulement en utilisant un redémarrage ICE (l'agent DOIT utiliser les redémarrages ICE pour le faire) :
- o Changer les destinations du flux de données.
- o Changer d'une mise en œuvre légère à une mise en œuvre complète.
- o Changer d'une mise en œuvre complète à une mise en œuvre légère.

Pour redémarrer ICE, un agent DOIT changer le mot de passe et le fragment de nom d'utilisateur pour le ou les flux de données à redémarrer.

Quand ICE est redémarré, l'ensemble de candidates pour la nouvelle session ICE pourrait inclure certaines des candidates utilisées dans la session ICE actuelle, ou aucune, ou toutes.

Comme décrit au paragraphe 6.1.1, les agents NE DOIVENT PAS redéterminer les rôles au titre d'un redémarrage ICE, sauf si certains critères qui exigent que les rôles soient redéterminés sont satisfaits.

10. Option ICE

Cette Section définit une nouvelle option ICE, "ice2". Quand un agent ICE inclut "ice2" dans un échange de candidates, l'option ICE indique qu'il est conforme à la présente spécification. Par exemple, l'agent ne va pas utiliser la procédure de désignation agressive définie dans la RFC 5245. De plus, il va assurer qu'un homologue conforme à la RFC 5245 n'utilise pas non plus la désignation agressive, comme exigé par la Section 14 de la RFC 5245 pour les homologues qui reçoivent des options ICE in connues.

Un agent conforme à la présente spécification DOIT informer l'homologue de la conformité à l'utilisation de l'option "ice2".

Note : le codage de l'option "ice2", et le ou les messages utilisés pour la porter à l'homologue, sont spécifiques du protocole. Le codage pour SDP [RFC4566] est défini dans la [RFC8839].

11. Maintiens en vie

Tous les points d'extrémité DOIVENT envoyer des maintiens en vie pour chaque session de données. Ces maintiens en vie servent à garder en vie les liens de NAT pour la session de données. Les maintiens en vie DEVRAIENT être envoyés en utilisant un format qui est pris en charge par son homologue. Les points d'extrémité ICE permettent des maintiens en vie fondés sur STUN pour les flux UDP, et à ce titre, les maintiens en vie STUN DOIVENT être utilisés quand un agent ICE est une mise en œuvre ICE complète et communique avec un homologue qui prend en charge ICE (léger ou complet).

Un agent DOIT envoyer un maintien en vie sur chaque paire candidate qui est utilisée pour envoyer des données si aucun paquet n'a été envoyé sur cette paire dans les Tr dernières secondes. Les agents DEVRAIVENT utiliser une valeur de Tr de 15 secondes. Les agents PEUVENT utiliser une plus grande valeur mais NE DOIVENT PAS utiliser une valeur inférieure à 15 secondes.

Une fois que les paires choisies ont été produites pour un flux de données, les maintiens en vie sont seulement envoyés sur ces paires.

Un agent DOIT arrêter d'envoyer des maintiens en vie sur un flux de données si le flux de données est supprimé. Si la session ICE est terminée, un agent DOIT arrêter d'envoyer des maintiens en vie sur tous les flux de données.

Un agent PEUT utiliser une autre valeur pour Tr, par exemple, sur la base de la configuration ou des caractéristiques de réseau/NAT. Par exemple, si un agent a une façon dynamique de découvrir les durées de vie de lien des NAT intervenants, il peut utiliser cette valeur pour déterminer Tr. Les administrateurs qui déploient ICE dans des environnements de réseautage plus contrôlés DEVRAIENT régler Tr à la plus longue durée possible dans leur environnement.

Quand STUN est utilisé pour les maintiens en vie, une indication de lien STUN est utilisée [RFC5389]. L'indication NE DOIT PAS utiliser de mécanisme d'authentification. Elle DEVRAIT contenir l'attribut FINGERPRINT pour aider au démultiplexage, mais elle NE DEVRAIT PAS contenir d'autre attribut. Elle est utilisée seulement pour garder en vie les liens de NAT. L'indication de lien est envoyée en utilisant les mêmes candidates locales et distantes qui sont utilisées pour les données. Bien que les indications de lien soient utilisées pour les maintiens en vie, un agent DOIT être prêt à recevoir aussi une vérification de connexité. Si une vérification de connexité est reçue, une réponse est générée comme discuté dans la [RFC5389], mais cela n'a pas d'autre impact sur le traitemenr ICE.

Les agents DOIVENT par défaut utiliser les maintiens en vie STUN. Des usages individuels de ICE et des extensions ICE PEUVENT spécifier des maintiens en vie spécifiques de l'usage/extension.

12. Traitement des données

12.1 Envoi des données

Un agent ICE PEUT envoyer des données sur toute paire valide avant que des paires choisies aient été produites pour le flux de données.

Une fois que des paires choisies ont été produites pour un flux de données, un agent DOIT envoyer des données seulement sur ces paires.

Un agent envoie des données à partir de la base de la candidate locale à la candidate distante. Dans le cas d'une candidate relayée locale, les données sont transmises à travers la base (située dans le serveur TURN) en utilisant les procédures définies dans la [RFC5766].

Si la candidate locale est une candidate relayée, il est RECOMMANDÉ qu'un agent crée un canal sur le serveur TURN vers la candidate distante. Cela est fait en utilisant les procédures de création de canal définies dans la Section 11 de la [RFC5766].

La paire choisie pour un composant d'un flux de données est :
- vide si l'état de la liste de contrôle pour ce flux de données est En cours, et si il n'y a pas de précédente paire choisie pour ce composant du fait d'un redémarrage ICE ;
- égale à la précédente paire choisie pour un composant d'un flux de données si l'état de la liste de contrôle pour ce flux de données est En cours, et si il y a une paire choisie précédente pour ce composant du fait d'un redémarrage ICE.

Sauf si un agent est capable de produire une paire choisie pour chaque composant associé à un flux de données, l'agent NE DOIT PAS continuer d'envoyer des données pour un composant associé à ce flux de données.

12.1.1 Procédures pour mises en œuvre légères

Une mise en œuvre légère NE DOIT PAS envoyer de données jusqu'à ce qu'elle ait une liste valide qui contient une paire candidate pour chaque composant de ce flux de données. Une fois que c'est arrivé, l'agent ICE PEUT commencer à envoyer des paquets de données. Pour faire cela, il envoie des données à la paire candidate distante (en réglant l'adresse et l'accès de destination du paquet égaux à ceux de la candidate distante) et il va l'envoyer à partir de la base associée à la paire candidate utilisée pour envoyer les données. Dans le cas d'une candidate relayée, les données sont envoyées de l'agent et transmises à travers la base (située dans le serveur TURN) en utilisant les procédures définies dans la [RFC5766].

12.2 Réception des données

Même quand les agents ICE ont seulement la permission d'envoyer des données en utilisant des paires candidates valides (et, une fois que les paires choisies ont été produites, seulement sur les paires choisies) les mises en œuvre de ICE DEVRAIENT par défaut être prêtes à recevoir les données sur toute candidate fournie dans le plus récent échange de candidates avec l'homologue. Les usages de ICE PEUVENT définir des règles différentes, par exemple, en définissant que les données ne seront pas envoyées tant que les paires choisies n'ont pas été produites pour un flux de données.

Quand un agent reçoit un paquet RTP avec une nouvelle adresse IP de source ou destination pour un flux de données RTP/RTCP particulier, il est RECOMMANDÉ que l'agent réajuste ses mémoires-tampons de gigue.

Le paragraphe 8.2 de la [RFC3550] décrit un algorithme pour détecter les collisions et boucles de source de synchronisation (SSRC, *Synchronization Source*). Ces algorithmes se fondent, en partie, sur la vue de différentes adresses de transport de source avec la même SSRC. Cependant, quand ICE est utilisé, de tels changements vont parfois se produire lorsque le flux de données change entre les candidates. Un agent va être capable de déterminer qu'un flux de données provient du même homologue par suite de l'échange STUN qui traite la transmission des données du support. Donc, si il y a un changement de l'adresse de transport de source, mais si les paquets de données de support viennent du même agent d'homologue, cela NE DOIT PAS être traité comme une collision de SSRC.

13. Considérations d'extensibilité

La présente spécification fait des choix très spécifiques sur la façon dont les agents ICE dans une session se coordonnent

pour arriver à l'ensemble de paires candidates qui sont choisies pour les données. Il est prévu que de futures spécifications voudront changer ces algorithmes, que ce soient de simples changements comme de durée de temporisateur ou de plus grands changements comme une refonte de l'algorithme de priorité. Quand un tel changement est fait, assurer l'interopérabilité entre les deux agents d'une session est critique.

D'abord, ICE fournit le concept d'option ICE. Chaque extension ou changement à ICE est associé à une option ICE. Quand un agent prend en charge une telle extension ou changement, il fournit l'option ICE à l'agent d'homologue au titre de l'échange de candidates.

Une des complications de la réalisation de l'interopérabilité est que ICE s'appuie sur un algorithme distribué qui fonctionne sur les deux agents pour converger sur un accord sur l'ensemble de paires candidates. Si les deux agents ont des algorithmes différents, il peut être difficile de garantir la convergence sur les mêmes paires candidates. La procédure de désignation décrite à la Section 8 élimine un peu le besoin d'une étroite coordination en délégant complètement l'algorithme de choix à l'agent contrôleur, et ICE va converger parfaitement même quand les deux agents utilisent des algorithmes de priorité de paire différents. Une des clés d'une telle convergence est la vérification déclenchée, qui assure que la paire désignée est validée par les deux agents.

ICE est aussi extensible aux autres flux de données au delà de RTP et pour les protocoles de transport au-delà de UDP. Les extensions à ICE pour des flux de données non RTP doivent spécifier combien de composants elles utilisent et leur allouer des identifiants de composant, en commençant à 1 pour le plus important identifiant de composant. Les spécifications pour de nouveaux protocoles de transport DOIVENT définir comment les diverses étapes, s'il en est, du traitement ICE diffèrent de UDP.

14. Réglage de Ta et du RTO

14.1 Généralités

Durant la phase de rassemblement ICE (paragraphe 5.1.1) et pendant que ICE effectue les vérifications de connexité (Section 7) un agent ICE déclenche les transactions STUN et TURN. Ces transactions sont régulées à un rythme indiqué par Ta, et l'intervalle de retransmission pour chaque transaction est calculé sur la base du temporisateur de retransmission pour les transactions STUN (RTO) [RFC5389].

Cette section décrit comment les valeurs de Ta et de RTO sont calculées durant la phase de rassemblement ICE et pendant que ICE effectue les vérifications de connexité.

Note : Précédemment, dans la RFC 5245, différentes formules étaient définies pour calculer Ta et RTO, selon que ICE était ou non utilisé pour un flux de données en temps réel (par exemple, RTP).

Les formules ci-dessous résultent en un comportement par lequel un agent va envoyer son premier paquet pour chaque vérification de connexité avant d'effectuer une retransmission. Cela peut être vu dans les formules pour le RTO (qui représente l'intervalle de retransmission). Ces formules s'adaptent avec N, le nombre de vérifications à effectuer. Par suite, ICE maintient un rythme assez constant, mais devient plus sensible à la perte de paquets. La perte du premier paquet pour toute vérification de connexité va probablement causer un plus long temps pour valider cette paire, et à la place, une vérification de priorité inférieure (mais pour laquelle il n'y a pas de perte de paquet) a plus de chances de se terminer avant. Il en résulte des performances sous optimales de ICE, qui choisit des paires de priorité inférieures plutôt que des paires de haute priorité.

14.2 Ta

Les agents ICE DEVRAIENT utiliser une valeur de Ta par défaut de 50 ms, mais PEUVENT utiliser une autre valeur sur la base des caractéristiques des données associées.

Si un agent veut utiliser une valeur de Ta autre que la valeur par défaut, l'agent DOIT indiquer la valeur proposée à son homologue durant l'établissement de la session ICE. Les deux agents DOIVENT utiliser la plus haute valeur proposée. Si un agent ne propose pas de valeur, la valeur par défaut est utilisée pour cet agent quand on compare les valeurs.

Sans considération de la valeur de Ta choisie pour chaque agent, la combinaison de toutes les transactions provenant de tous les agents (si une certaine mise en œuvre a plusieurs agents concurremment) NE DOIT PAS être envoyée plus souvent qu'une fois toutes les 5 ms (comme si il y avait une valeur Ta globale pour réguler tous les agents). Voir à l'appendice B.1

les fondements de l'utilisation d'une valeur de 5 ms pour ICE.

Note : l'appendice C montre des exemples de la bande passante requise, en utilisant différentes valeurs de Ta.

14.3 RTO

Durant la phase de rassemblement de ICE, les agents ICE DEVRAIENT calculer la valeur de RTO en utilisant la formule suivante :

RTO = MAX (500 ms, Ta * (Nombre de candidates))

Nombre de candidates : nombre de candidates reflets de serveur et relais

Pour les vérifications de connexité, les agents DEVRAIENT calculer la valeur de RTO en utilisant la formule suivante :

RTO = MAX (500 ms, Ta * N * (Nombre En attente + Nombre En-Progrès))

N : nombre total de vérifications de connexité à effectuer.

Nombre En attente : nombre de vérifications dans la liste de contrôle réglées à l'état En attente.

Nombre En-Progrès : nombre de vérifications dans la liste de contrôle réglées à l'état En-Progrès.

Noter que le RTO va être différent pour chaque transaction car le nombre de vérifications dans les états En attente et En-Progrès change.

Les agents PEUVENT calculer la valeur de RTO en utilisant d'autres mécanismes que ceux décrits ci-dessus. Les agents NE DOIVENT PAS utiliser une valeur de RTO inférieure à 500 ms.

15. Exemples

La présente section montre deux exemples de ICE : une qui utilise des adresses IPv4 et une qui utilise des adresses IPv6.

Pour faciliter la compréhension, les adresses de transport sont données en utilisant des variables qui ont des mnémoniques. Le format du nom est entité-type-numéroséquence : "entité" se réfère à l'entité dont l'adresse IP sur laquelle est l'adresse de transport et est un de "L", "R", "STUN", ou "NAT". Le type est soit "PUB" pour les adresses de transport qui sont publiques, soit "PRIV" pour les adresses de transport qui sont privées [RFC1918]. Finalement, numéroséquence est un numéro de séquence qui est différent pour chaque adresse de transport du même type sur une entité particulière. Chaque variable a une adresse et accès IP, noté par varname.IP et varname.PORT, respectivement, où varname est le nom de la variable.

Dans le flux d'appels lui-même, les messages STUN sont annotés avec plusieurs attributs. L'attribut "S=" indique l'adresse de transport de source du message. L'attribut "D=" indique l'adresse de transport de destination du message. L'attribut "MA=" est utilisé dans les messages de lien de réponse STUN et se réfère à l'adresse transposée. "USE-CAND" implique la présence de l'attribut USE-CANDIDATE.

Les exemples de flux d'appels omettent les opérations d'authentification STUN et se concentrent sur un seul flux de données entre deux mises en œuvre complètes.

15.1 Exemple avec des adresses IPv4

L'exemple ci-dessous utilise la topologie de la Figure 7.

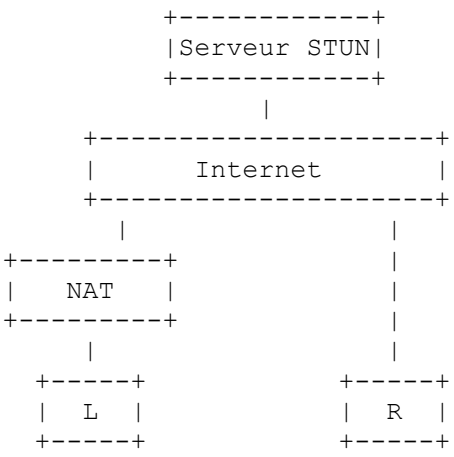
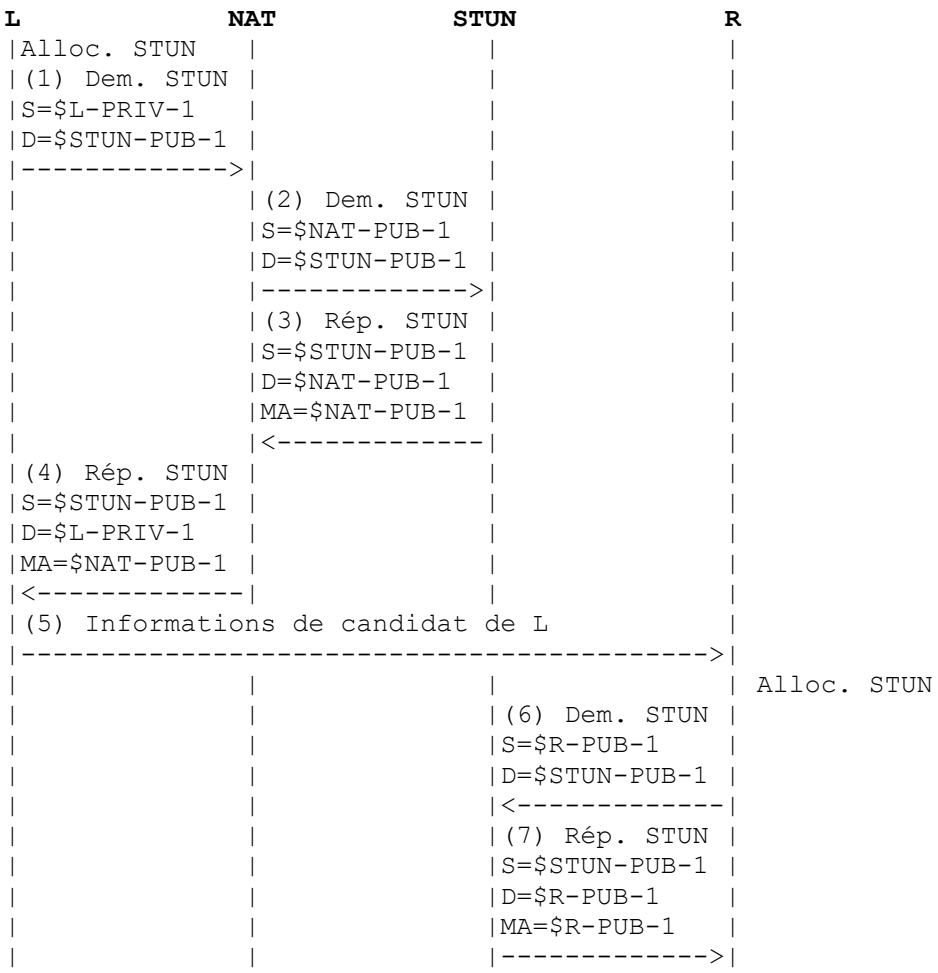
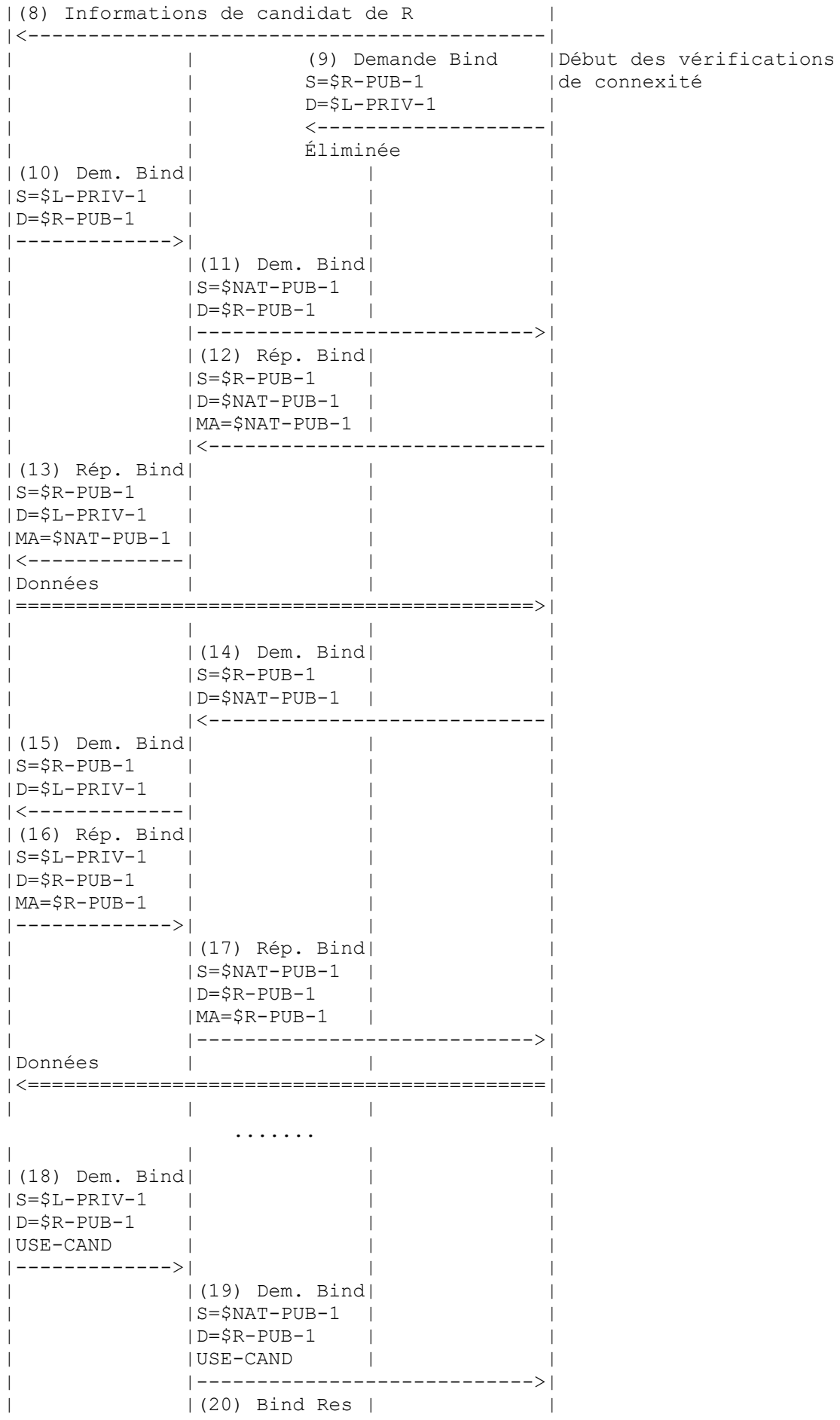


Figure 7 : Exemple de topologie

Dans l'exemple, les agents ICE L et R sont des mises en œuvre complètes de ICE. Les deux agents ont une seule adresse IPv4, et tous deux sont configurés avec le même serveur STUN. Le NAT a une propriété de transposition indépendante du point d'extrémité et une propriété de filtrage qui dépend de l'adresse. Les adresses IP des agents ICE, du serveur STUN, et du NAT sont montrées ci-dessous :

Entité	**Adresse IP**	**Mnémonique du nom**
Agent ICE L :	10.0.1.1	L-PRIV-1
Agent ICE R :	192.0.2.1	R-PUB-1
Serveur STUN :	192.0.2.2	STUN-PUB-1
NAT (public) :	192.0.2.3	NAT-PUB-1




(8) Informations de candidat de R
(9) Demande Bind
S=$R-PUB-1
D=$L-PRIV-1
Éliminée
Début des vérifications de connexité
(10) Dem. Bind
S=$L-PRIV-1
D=$R-PUB-1
(11) Dem. Bind
S=$NAT-PUB-1
D=$R-PUB-1
(12) Rép. Bind
S=$R-PUB-1
D=$NAT-PUB-1
MA=$NAT-PUB-1
(13) Rép. Bind
S=$R-PUB-1
D=$L-PRIV-1
MA=$NAT-PUB-1
Données
Données
(14) Dem. Bind
S=$R-PUB-1
D=$NAT-PUB-1
(15) Dem. Bind
S=$R-PUB-1
D=$L-PRIV-1
(16) Rép. Bind
S=$L-PRIV-1
D=$R-PUB-1
MA=$R-PUB-1
(17) Rép. Bind
S=$NAT-PUB-1
D=$R-PUB-1
MA=$R-PUB-1
Données
....
(18) Dem. Bind
S=$L-PRIV-1
D=$R-PUB-1
USE-CAND
(19) Dem. Bind
S=$NAT-PUB-1
D=$R-PUB-1
USE-CAND
(20) Bind Res

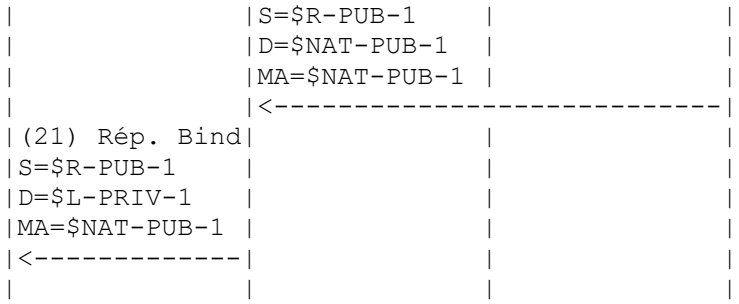


Figure 8 : Exemple de flux

Messages 1 à 4 : l'agent L rassemble une candidate hôte à partir de son adresse IP locale, et il en envoie une demande Binding STUN au serveur STUN. La demande crée un lien de NAT. L'adresse IP publique du NAT devient la candidate reflet de serveur de l'agent L.

Message 5 : l'agent L envoie ses informations de candidate locale à l'agent R, en utilisant le protocole de signalisation associé à l'usage ICE.

Messages 6 à 7 : l'agent R rassemble une candidate hôte à partir de son adresse IP locale, et en envoie une demande Binding STUN au serveur STUN. Comme l'agent R n'est pas derrière un NAT, la candidate reflet de serveur de R va être identique à la candidate hôte.

Message 8 : l'agent R envoie ses informations de candidate locale à l'agent L, en utilisant le protocole de signalisation associé à l'usage ICE.

Comme les deux agents sont des mises en œuvre complètes de ICE, l'agent initiateur (agent L) devient l'agent contrôleur.

Les agents L et R apparient tous deux les candidates. Les deux agents ont initialement deux paires. Cependant,l' agent L va élaguer la paire contenant sa candidate reflet de serveur, résultant en juste une (L1). Chez l'agent L, cette paire a une candidate locale de $L_PRIV_1 et une candidate distante de $R_PUB_1. Chez l'agent R, il y a deux paires. La paire de plus haute priorité (R1) a une candidate locale de $R_PUB_1 et une candidate distante de $L_PRIV_1, et la seconde paire (R2) a une candidate locale de $R_PUB_1 et une candidate distante de $NAT_PUB_1. Les paires sont montrées ci-dessous (les numéros de paires sont seulement pour la référence) :

Entité	**Paire locale**	**Paire distante**	**N ° de paire**
Agent ICE L :	L_PRIV_1	R_PUB_1	L1
Agent ICE R :	R_PUB_1	L_PRIV_1	R1
	R_PUB_1	NAT_PUB_1	R2

Message 9 : l'agent R initie une vérification de connexité pour la paire n° 2. Comme la candidate distante de la paire est l'adresse privée de l'agent L, la vérification ne va pas réussir, car la demande ne peut pas être acheminée de R à L, et va être éliminée par le réseau.

Messages 10 à 13 : l'agent L initie une vérification de connexité pour la paire L1. La vérification réussit, et L crée une nouvelle paire (L2). La candidate locale de la nouvelle paire est $NAT_PUB_1, et la candidate distante est $R_PUB_1. La paire (L2) est ajoutée à la liste valide de l'agent L. L'agent L peut maintenant envoyer et recevoir des données sur la paire (L2) si il le souhaite.

Entité	**Paire locale**	**Paire distante**	**N ° de paire**	**Valide**
Agent ICE L :	L_PRIV_1	R_PUB_1	L1	
	NAT_PUB_1	R_PUB_1	L2	X
Agent ICE R :	R_PUB_1	L_PRIV_1	R1	
	R_PUB_1	NAT_PUB_1	R2	

Messages 14 à 17 : quand l'agent R reçoit la demande Binding de l'agent L (message 11) il initie une vérification de connexité déclenchée. La paire correspond à une des paires existantes de l'agent R (R2). La vérification réussit, et la paire (R2) est ajoutée à la liste valide de l'agent R. L'agent R peut maintenant envoyer et recevoir des données sur la paire (R2) si il le souhaite.

Entité	Paire locale	Paire distante	N ° de paire	Valide
Agent ICE L :	L_PRIV_1	R_PUB_1	L1	
	NAT_PUB_1	R_PUB_1	L2	X
Agent ICE R :	R_PUB_1	L_PRIV_1	R1	
	R_PUB_1	NAT_PUB_1	R2	X

Messages 18 à 21 : à un moment donné, l'agent contrôleur (agent L) décide de désigner une paire (L2) dans la liste valide. Il effectue une vérification de connexité sur la paire (L2) et inclut l'attribut USE-CANDIDATE dans la demande Binding. Comme la vérification réussit, l'agent L règle la valeur du fanion Désigné de la paire (L2) à "vrai", et l'agent R règle la valeur du fanion Désigné de la paire correspondante (R2) à "vrai". Comme il n'y a plus de composant associé au flux, les paires désignées deviennent les paires choisies. Par conséquent, le traitement pour ce flux passe à l'état Terminé. Le traitement ICE passe aussi à l'état Terminé.

15.2 Exemple avec des adresses IPv6

L'exemple ci-dessous utilise la topologie montrée par la Figure 9.

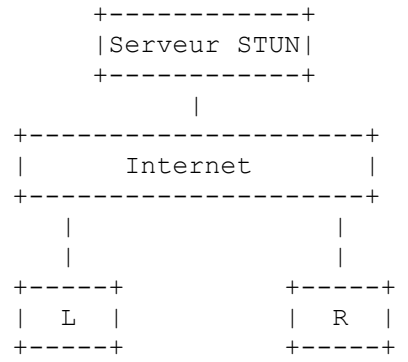
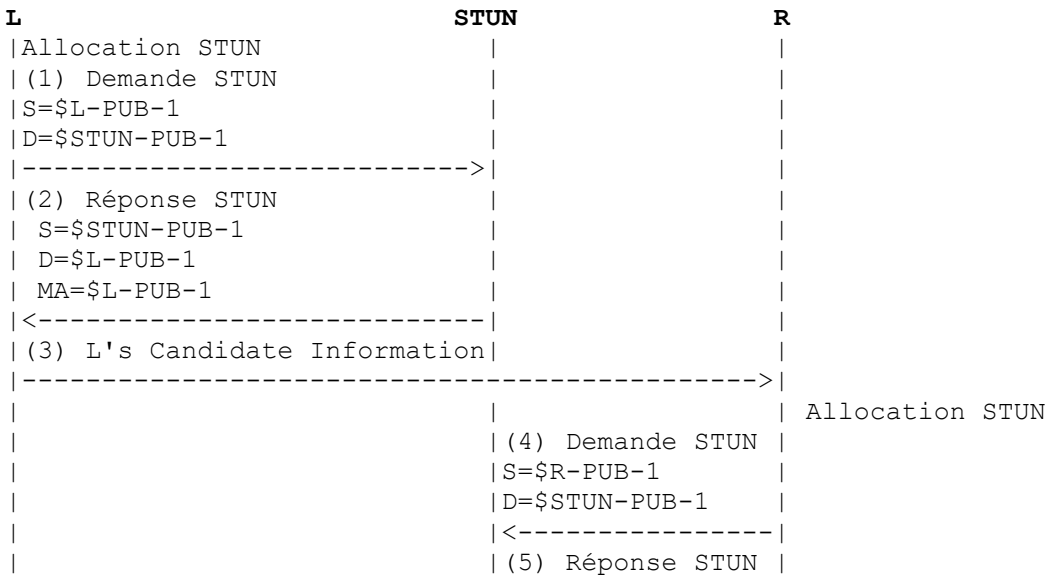


Figure 9 : Exemple de topologie

Dans l'exemple, les agents ICE L et R sont des mises en œuvre complètes de ICE. Les deux agents ont une seule adresse IPv6, et tous deux sont configurés avec le même serveur STUN. Les adresses IP des agents ICE et du serveur STUN sont montrées ci-dessous :

Entité	Adresse IP	Mnémonique du nom
Agent ICE L :	2001:db8::3	L-PUB-1
Agent ICE R :	2001:db8::5	R-PUB-1
Serveur STUN :	2001:db8::9	STUN-PUB-1



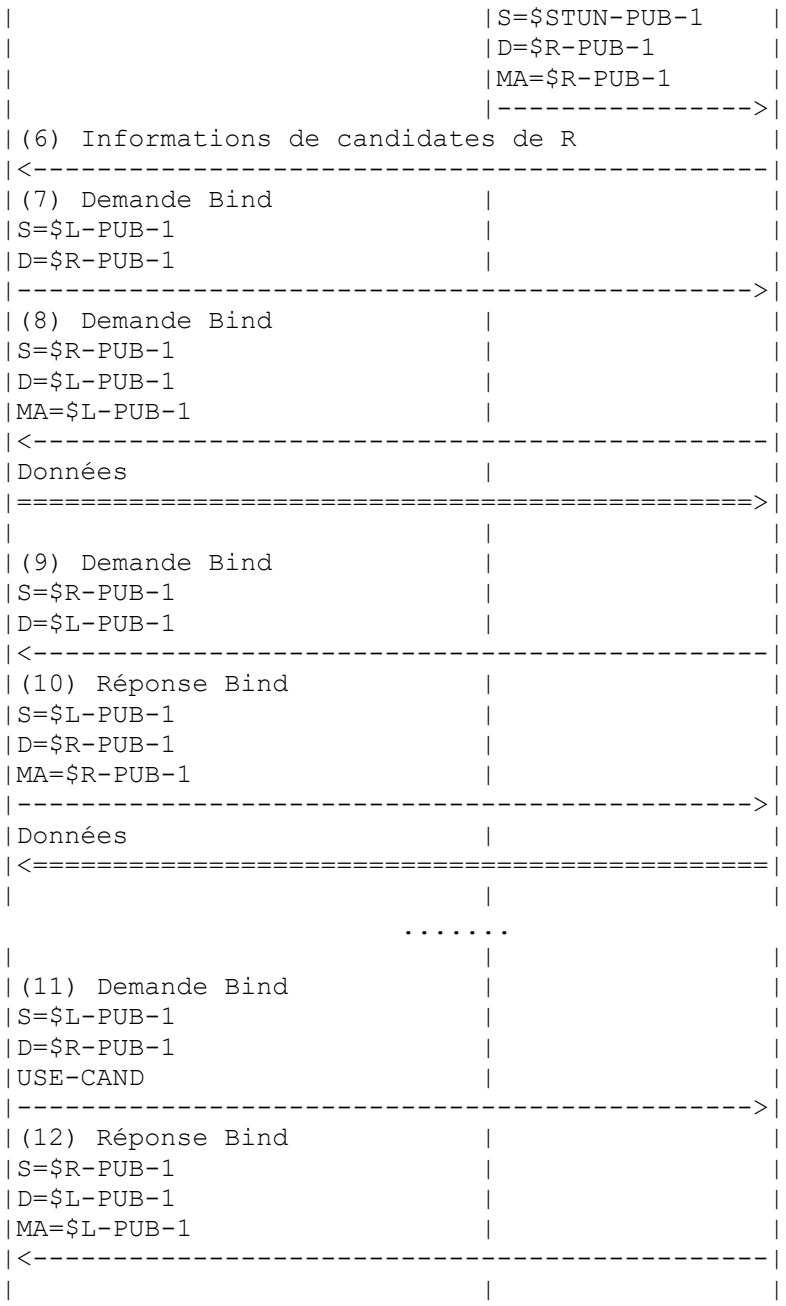


Figure 10 : Exemple de flux

Messages 1-2 : l'agent L rassemble une candidate hôte à partir de son adresse IP locale, et en envoie une demande Binding STUN au serveur STUN. Comme l'agent L n'est pas derrière un NAT, la candidate reflet de serveur de L va être identique à la candidate hôte.

Message 3 : l'agent L envoie ses informations de candidate locale à l'agent R, en utilisant le protocole de signalisation associé à l'usage ICE.

Messages 4-5 : l'agent R rassemble une candidate hôte à partir de son adresse IP locale, et en envoie une demande Binding STUN au serveur STUN. Comme l'agent R n'est pas derrière un NAT, la candidate reflet de serveur de R va être identique au candidat hôte.

Message 6 : l'agent R envoie ses informations de candidate locale à l'agent L, en utilisant le protocole de signalisation associé à l'usage ICE.
Comme les deux agents sont des mises en œuvre complètes de ICE, l'agent initiateur (agent L) devient l'agent contrôleur.

Les agents L et R apparient tous deux les candidates. Les deux agents ont initialement une paire chacun. Chez l'agent L, la paire (L1) a une candidate locale de $L_PUB_1 et une candidate distante de $R_PUB_1. Chez l'agent R, la paire (R1) a une candidate locale de $R_PUB_1 et une candidate distante de $L_PUB_1. Les paires sont montrées ci-dessous (les numéros de paires sont seulement pour référence) :

Entité	Paire locale	Paire distante	N ° de paire
Agent ICE L :	L_PUB_1	R_PUB_1	L1
Agent ICE R :	R_PUB_1	L_PUB_1	R1

Messages 7-8 : l'agent L initie une vérification de connexité pour la paire L1. La vérification réussit, et la paire (L1) est ajoutée à la liste valide de l'agent L. L'agent L peut maintenant envoyer et recevoir des données sur la paire (L1) si il le souhaite.

Entité	Paire locale	Paire distante	N ° de paire	Valide
Agent ICE L :	L_PUB_1	R_PUB_1	L1	X
Agent ICE R :	R_PUB_1	L_PUB_1	R1	

Messages 9-10 : quand l'agent R reçoit la demande Binding de l'agent L (message 7) il va initier une vérification de connexité déclenchée. La paire correspond à la paire existante de l'agent R (R1). La vérification réussit, et la paire (R1) est ajoutée à la liste valide de l'agent R. L'agent R peut maintenant envoyer et recevoir des données sur la paire (R1) si il le souhaite.

Entité	Paire locale	Paire distante	N ° de paire	Valide
Agent ICE L :	L_PUB_1	R_PUB_1	L1	X
Agent ICE R :	R_PUB_1	L_PUB_1	R1	X

Messages 11-12 : à un moment donné, l'agent contrôleur (agent L) décide de désigner une paire (L1) dans la liste valide. Il effectue une vérification de connexité sur la paire (L1) et inclut l'attribut USE-CANDIDATE dans la demande Binding. Comme la vérification réussit, l'agent L règle la valeur du fanion Désigné de la paire (L1) à "vrai", et l'agent R règle la valeur du fanion Désigné de la paire correspondante (R1) à "vrai".

Comme il n'y a plus de composant associé au flux, les paires désignées deviennent les paires choisie. Par conséquent, le traitement de ce flux passe à l'état Terminé. Le processus ICE passe aussi à l'état Terminé.

16. Extensions à STUN

16.1 Attributs

La présente spécification définit quatre attributs STUN : PRIORITY, USE-CANDIDATE, ICE-CONTROLLED, et ICE-CONTROLLING.

L'attribut PRIORITY indique la priorité à associer à une candidate reflet de l'homologue, si il va en être découvert une par cette vérification. C'est un entier non signé de 32 bits et il a une valeur d'attribut de 0x0024.

L'attribut USE-CANDIDATE indique que la paire candidate résultant de cette vérification va être utilisée pour la transmission de données. L'attribut n'a pas de contenu (le champ Longueur de l'attribut est zéro) ; il sert de fanion. Il a une valeur d'attribut de 0x0025.

L'attribut ICE-CONTROLLED est présent dans une demande Binding. L'attribut indique que le client estime qu'il est actuellement dans le rôle de contrôlé. Le contenu de l'attribut est un entier non signé de 64 bits dans l'ordre des octets du réseau, et il contient un nombre aléatoire. Le nombre est utilisé pour résoudre les conflits de rôle, et il est appelée la "valeur de départage". Un agent ICE DOIT utiliser le même nombre pour toutes les demandes Binding, pour tous les flux, au sein d'une session ICE, sauf si il a reçu une réponse 487, et dans ce cas, il DOIT changer le nombre (paragraphe 7.2.5.1). L'agent PEUT changer le nombre quand un redémarrage ICE se produit.

L'attribut ICE-CONTROLLING est présent dans une demande Binding. L'attribut indique que le client estime qu'il est actuellement dans le rôle de contrôleur. Le contenu de l'attribut est un entier non signé de 64 bits dans l'ordre des octets du réseau, et contient un nombre aléatoire. Comme pour l'attribut ICE-CONTROLLED, le nombre est utilisé pour résoudre les conflits de rôle. Un agent DOIT utiliser le même nombre pour toutes les demandes Binding, pour tous les flux, au sein

d'une session ICE, sauf si il a reçu une réponse 487, et dans ce cas, il DOIT changer le nombre (paragraphe 7.2.5.1). L'agent PEUT changer le nombre quand un redémarrage ICE se produit.

16.2 Nouveaux codes de réponse d'erreur

La présente spécification définit un seul code de réponse d'erreur :

487 (Conflit de rôle) : la demande Binding contenait un attribut soit ICE-CONTROLLING, soit ICE-CONTROLLED, indiquant un rôle ICE qui est en conflit avec le serveur. Le serveur distant a comparé les valeurs de départage du client et du serveur et a déterminé que le client doit changer de rôle.

17. Considérations de fonctionnement

Cette Section discute les questions relevant des opérateurs de réseaux où ICE va être utilisé par les points d'extrémité.

17.1 Types de NAT et de pare-feu

ICE a été conçu pour fonctionner avec les équipements de NAT et pare-feu existants. Par conséquent, il n'est pas nécessaire de remplacer ou reconfigurer les équipements de NAT et pare-feu existants pour faciliter le déploiement de ICE. Bien sûr, ICE a été développé pour être déployé dans les environnements où l'opérateur de voix sur IP (VoIP, *Voice over IP*) n'a pas de contrôle sur l'infrastructure de réseau IP, y compris les pare-feu et NAT.

Ceci dit, ICE fonctionne mieux dans des environnements où les appareils de NAT ont un "comportement" conforme, satisfaisant les recommandations définies dans les [RFC4787] et [RFC5382]. Dans les réseaux avec des NAT au comportement conforme, ICE va fonctionner sans avoir besoin d'un serveur TURN, améliorant donc la qualité vocale, diminuant les temps d'établissement d'appel, et réduisant la demande de bande passante sur les opérateurs de réseau.

17.2 Exigences de bande passaznte

Le déploiement de ICE peut avoir plusieurs interactions avec la capacité réseau disponible que les opérateurs doivent prendre en considération.

17.2.1 Plannification de capacité de serveur STUN et TURN

Tout d'abord, ICE utilise les serveurs TURN et STUN, qui vont normalement être situés dans des centres de données. Les serveurs STUN exigent relativement peu de bande passante. Pour chaque composant de chaque flux de données, il va y avoir une ou plusieurs transactions STUN provenant de chaque client au serveur STUN. Dans un déploiement de base de VoIP IPv4 seulement vocal, il va y avoir quatre transactions par appel (une pour RTP et une pour RTCP, pour l'appelant et pour l'appelé). Chaque transaction est une seule demande et une seule réponse, la première de 20 octets, et la dernière, de 28.

Par conséquent, si un système a N utilisateurs, et si chacun fait quatre appels à l'heure chargée, cela va exiger N*1,7 bit/s. Pour un million d'utilisateurs, cela fait 1,7 M bit/s, un très petit nombre (relativement parlant).

Le trafic TURN est plus substantiel. Le serveur TURN va voir un volume de trafic égal au volume STUN (bien sûr, si des serveurs TURN sont déployés, il n'y a pas besoin d'un serveur STUN séparé) en plus du trafic pour les données réelles. La quantité d'appels exigeant TURN pour le relais de données dépend fortement des topologies de réseau, et peut et va varier au fil du temps. Dans un réseau avec 100 % de NAT au comportement conforme, c'est exactement zéro.

Les considérations de planification ci-dessus deviennent plus significatives dans des scénarios multimédia (par exemple, conférences audio et vidéo) et quand le nombre des participants dans une session croît.

17.2.2 Vérifications de rassemblement et de connexité

Le processus de rassemblement des candidates et d'effectuer les vérifications de connexité peut être gourmand en bande passante. ICE a été conçu pour réguler ces deux processus. Les phases de rassemblement et de vérification de connexité

sont destinées à générer du trafic en gros avec la même bande passante que ce que le trafic de données lui-même va consommer une fois que le processus ICE se termine. Cela a été fait pour assurer que si un réseau est conçu pour prendre en charge un trafic de communication d'un certain type (voix, vidéo, ou juste texte) il va avoir une capacité suffisante pour prendre en charge les vérifications ICE pour ces données. Une fois que ICE s'est terminé, les maintiens en vie ICE ultérieurs vont ensuite causer un accroissement marginal de l'utilisation de la bande passante totale ; cependant, cela va normalement être un accroissement extrêmement faible.

L'encombrement dû aux phases de rassemblement et de vérification s'est révélé être un problème dans des déploiements qui n'utilisent pas la régulation. Normalement, les liaisons d'accès deviennent encombrées lorsque les points d'extrémité inondent le réseau avec des vérifications aussi rapidement qu'ils peuvent les envoyer. Par conséquent, les opérateurs de réseau doivent s'assurer que leurs mises en œuvre de ICE prennent en charge la caractéristique de régulation. Bien que cette régulation augmente les temps d'établissement d'appels, cela rend le réseau favorable à ICE et plus facile à déployer.

17.2.3 Maintiens en vie

Les maintiens en vie STUN (sous la forme des indications STUN Binding) sont envoyés au milieu d'une session de données. Cependant, elles sont envoyées seulement en l'absence de trafic réel de données. Dans les déploiements avec des supports continus et sans utiliser la détection d'activité vocale (VAD, *Voice Activity Detection*) ou les déploiements où la VAD est utilisée avec un court intervalle de bruit de confort (maximum de 1 seconde) les maintiens en vie ne sont jamais utilisés et il n'y a pas d'augmentation de l'usage de la bande passante. Quand la VAD est utilisée sans bruit de confort, les maintiens en vie vont être envoyés durant les périodes de silence. Cela implique un seul paquet toutes les 15 à 20 secondes, beaucoup moins que le paquet toutes les 20 à 30 ms qui est envoyé quand il y a de la voix. Donc, les maintiens en vie n'ont pas de réel impact sur la planification de capacité.

17.3 ICE et ICE léger

Les déploiements qui utilisent un mélange de ICE et ICE léger inter-opèrent les uns avec les autres. Ils ont été explicitement conçus pour le faire.

Cependant, ICE léger peut seulement être déployé dans des cas d'utilisation limités. Ces cas, et les précautions qu'ils impliquent, sont documentés dans l'Appendice A.

17.4 Gestion de réparations et de performances

ICE utilise des vérifications de connexité de bout en bout et place beaucoup du traitement dans les points d'extrémité. Cela lance un défit à l'opérateur de réseau -- comment peut on réparer les déploiements de ICE ? Comment peut on savoir quelles sont les performances de ICE ?

ICE a des caractéristiques incorporées pour aider à traiter ces problèmes. Les serveurs de signalisation, normalement déployés dans des centres de données de l'opérateur du réseau, vont voir le contenu des échanges de candidates qui portent les paramètres de ICE. Ces paramètres incluent le type de chaque candidate (hôte, reflet de serveur, ou relayé) ainsi que leurs adresses respectives. Une fois que le traitement ICE s'est achevé, un échange de candidates mises à jour a lieu, signalant l'adresse choisie (et son type). Cette signalisation mise à jour est effectuée exactement pour l'édification des équipements de réseau (comme un outil de diagnostic attaché à la signalisation) sur les résultats du traitement de ICE.

Par conséquent, grâce aux enregistrements générés par un serveur de signalisation, un opérateur de réseau peut observer quels types de candidates sont utilisés pour chaque appel et quelles adresses ont été choisies par ICE. C'est la principale information qui aide à évaluer les performances de ICE.

17.5 Configuration de point d'extrémité

ICE s'appuie sur plusieurs éléments de données configurés dans les points d'extrémité. Ces données de configuration incluent des temporisateurs, des accréditifs pour les serveurs TURN, et des noms d'hôte pour les serveurs STUN et TURN. ICE lui-même ne fournit pas de mécanisme pour cette configuration. À la place, on suppose que ces informations sont attachées à tout mécanisme utilisé pour configurer tous les autres paramètres dans le point d'extrémité. Pour les téléphones SIP, des solutions standard telles que le cadre de configuration [RFC6080] ont été définies.

18. Considérations de l'IAB

L'IAB a étudié le problème de l'auto correction d'adressage unilatérale (UNSAF, *Unilateral Self-Address Fixing*) qui est le processus général par lequel un agent ICE tente de déterminer son adresse dans un autre domaine de l'autre côté d'un NAT à travers un mécanisme collaboratif de reflet de protocole [RFC3424]. ICE est un exemple de protocole qui effectue ce type de fonction. On remarque que le processus pour ICE n'est pas unilatéral, mais bilatéral, et la différence a un impact significatif sur les questions soulevées par l'IAB. Bien sûr, ICE peut être considéré comme un protocole d'auto correction bilatéral (B-SAF, *Bilateral Self-Address Fixing*) plutôt qu'un protocole UNSAF. Néanmoins, l'IAB a exigé que tous les protocoles développés à cette fin documentent un ensemble spécifique de considérations. La présente section répond à ces exigences.

18.1 Définition du problème

D'après la RFC 3424, toute proposition d'UNSAF doit fournir une définition précise du problème spécifique, de portée limitée, qui va être résolu par la proposition d'UNSAF. Une réparation à court terme ne devrait pas être généralisée pour résoudre d'autres problèmes. De telles généralisations conduisent à la dépendance et l'usage prolongés de la supposée réparation à court terme -- signifiant qu'il n'est pas approprié de la dire "à court terme".

Les problèmes spécifique que résout ICE sont :
Fournir un moyen pour que deux homologues déterminent l'ensemble d'adresses de transport qui peut être utilisé pour la communication.
Fournir un moyen pour qu'un agent détermine une adresse accessible par un autre homologue avec lequel il souhaite communiquer.

18.2 Stratégie de sortie

D'après la RFC 3424, toute proposition d'UNSAF doit fournir la description d'une stratégie de sortie / plan de transition. Les meilleures réparations à court terme sont celles qui vont naturellement voir de moins en moins d'utilisation à mesure que la technologie appropriée est déployée.

ICE lui-même ne va pas être facilement déphasé. Cependant, il est utile même dans un Internet mondialement connecté, pour servir de moyen pour détecter si une défaillance de routeur a temporairement perturbé la connexité, par exemple. ICE aide aussi à empêcher certaines attaques contre la sécurité qui n'ont rien à voir avec les NAT. Cependant, ce que fait ICE est d'aider à se débarasser d'autres mécanismes UNSAF. ICE prend effectivement parmi ces mécanismes, donnant la priorité à ceux qui sont meilleurs et déclassant ceux qui sont pires. Comme les NAT commencent à se dissiper avec l'introduction de IPv6, les reflets de serveur et les candidats relayés (deux formes d'adresses UNSAF) ne sont simplement jamais utilisés, parce que une connexité de plus haute priorité existe avec les candidates d'hôte natives. Donc, les serveurs sont utilisés de moins en moins et peuvent éventuellement être supprimés quand leur usage arrive à zéro.

Bien sûr, ICE peut aider à la transition de IPv4 à IPv6. Il peut être utilisé pour déterminer si on utilise IPv6 ou IPv4 quand deux hôtes double piles communiquent avec SIP (IPv6 va être utilisé). Il peut aussi permettre à un réseau avec les deux connexités 6à4 et v6 natif de déterminer quelle adresse utiliser quand il communique avec un homologue.

18.3 Fragilité introduite par ICE

D'après la RFC 3424, toute proposition d'UNSAF doit fournir une discussion des questions spécifiques qui peuvent rendre les systèmes plus "fragiles". Par exemple, des approches qui impliquent d'utiliser des données à plusieurs couches de réseau créent plus de dépendances, augmentent les défis de débogage, et rendent la transition plus difficile.

ICE supprime en fait la fragilité des mécanismes d'UNSAF existants. En particulier, le STUN classique (comme décrit dans la [RFC3489]) a plusieurs points de fragilité. Une d'elles est le processus de découverte qui exige qu'un agent ICE essaye de classer le type de NAT derrière lequel il est. Ce processus est enclin à l'erreur. Avec ICE, ce processus de découverte n'est simplement pas utilisé. Plutôt que d'affirmer unilatéralement la validité de l'adresse, sa validité est déterminée dynamiquement en mesurant la connexité à un homologue. Le processus de détermination de la connexité est très robuste.

Un autre point de fragilité dans le STUN classique et dans tout autre mécanisme unilatéral est sa dépendance absolue à un serveur supplémentaire. ICE utilise un serveur pour allouer des adresses unilatérales, mais il permet aux agents de se connecter directement si possible. Donc, dans certains cas, la défaillance d'un serveur STUN va quand même permettre à

un appel de progresser quand ICE est utilisé.

Un autre point de fragilité dans le STUN classique est qu'il suppose que le serveur STUN est sur l'Internet public. Il est intéressant de noter qu'avec ICE, cela n'est pas nécessaire. Il peut y avoir une multitude de serveurs STUN dans divers domaines d'adresses. ICE va découvrir celui qui a fourni une adresse utilisable.

Le point de fragilité le plus troublant dans le STUN classique est qu'il ne fonctionne pas dans toutes les topologies de réseau. Dans les cas où il y a un NAT partagé entre chaque agent et le serveur STUN, le STUN traditionnel ne peut pas fonctionner. Avec ICE, cette restriction est supprimée.

Le STUN classique introduit aussi des considérations de sécurité. Heureusement, ces considérations de sécurité sont aussi atténuées par ICE.

Par conséquent, ICE sert à réparer la fragilité introduite dans le STUN classique, et n'introduit pas de fragilité supplémentaire dans le système.

L'inconvénient de ces améliorations est que ICE augmente le temps d'établissement de session.

18.4 Exigences pour une solution à long terme

D'après la RFC 3424, toute proposition d'UNSAF doit fournir l'identification des exigences pour un plus long terme, des solutions techniques valables, contribuer au processus de découverte de la bonne solution à plus long terme.

Les conclusions de la RFC 3489 restent inchangées. Cependant, on estime que ICE aide en fait parce qu'il peut faire partie de la solution à long terme.

18.5 Problèmes avec les boîtes de NAPT existantes

D'après la RFC 3424, toute proposition d'UNSAF doit fournir une discussion de l'impact des questions pratiques notées avec les NA[P]T existants, déployés et les rapports de l'expérience.

Un certain nombre de boîtes de NAT sont maintenant déployées dans le marché qui essayent de fournir une fonction "générique" d'ALG. Ces ALG génériques chassent les adresses IP, sous forme de texte ou binaire dans un paquet, et les réécrivent si elles correspondent à un lien. Cela interfère avec le STUN classique. Cependant, la mise à jour de STUN [RFC5389] utilise un codage qui cache ces adresses binaires aux ALG génériques.

Les boîtes existantes de NAPT ont des délais d'expiration non déterministes et normalement courts pour les liens fondés sur UDP. Cela exige que les mises en œuvre envoient des maintiens en vie périodiques pour conserver ces liens. ICE utilise une valeur par défaut de 15 s, qui est une estimation très prudente. Éventuellement, au fil du temps, lorsque les boites de NAT deviendront conformes au comportement de la [RFC4787], ce maintien en vie minimum va devenir déterministe et bien connu, et les temporisateurs ICE pourront être ajustés. Avoir un moyen de découvrir et contrôler l'intervalle minimum de maintien en vie serait quand même bien mieux.

19. Considérations de sécurité

19.1 Confidentialité de l'adresse IP

Le processus de sondage à la recherche de candidates révèle les adresses de source du client et de son homologue à tout attaquant qui écoute sur le réseau, et le processus d'échange des candidates révèle les adresses à tout attaquant qui est capable de voir la négociation. Certaines adresses, comme les adresses reflet de serveur rassemblées par l'interface locale d'utilisateurs de VPN, peuvent être des informations sensibles. Si ces attaques potentielles ne peuvent pas être atténuées, les usages de ICE peuvent définir des mécanismes pour contrôler quelles adresses sont révélées dans la négociation et/ou le processus de sondage. Les mises en œuvre individuelles peuvent aussi avoir des règles spécifiques pour contrôler quelles adresses sont révélées. Par exemple, la [RFC8829] donne des informations supplémentaires sur les aspects de confidentialité de la révélation des adresses IP via ICE pour les applications WebRTC. Il est RECOMMANDÉ aux mises en œuvre de ICE où de telles questions peuvent apparaître de fournir une interface programmable ou d'utilisateur qui assure le contrôle sur les interfaces réseau qui sont utilisées pour générer les candidates.

Sur la base des types de candidates fournie par l'homologue, et des résultats des essais de connexité effectués sur ces

candidates, l'homologue pourrait être capable de déterminer les caractéristiques du réseau local, par exemple, si des rythmes horaires différents sont apparents à l'homologue. Dans cette limite, l'homologue pourrait être capable de sonder le réseau local.

Il y a plusieurs types d'attaques possibles dans un système ICE. Les paragraphes qui suivent considèrent ces attaques et leurs contre-mesures.

19.2 Attaques contre les vérifications de connexité

Un attaquant pourrait tenter de perturber les vérifications de connexité STUN. En fin de compte, toutes ces attaques visent à tromper un agent ICE en l'amenant à penser quelque chose d'incorrect sur les résultats des vérifications de connexité. Selon le type d'attaque, l'attaquant doit avoir des capacités différentes. Dans certains cas, l'attaquant a besoin d'être sur le chemin des vérifications de connexité. Dans d'autres cas, l'attaquant n'a pas besoin d'être sur le chemin, pour autant qu'il soit capable de générer des vérifications de connexité STUN. Alors que les attaques sur les vérifications de connexité sont normalement effectuées par des entités de réseau, si un attaquant est capable de contrôler un point d'extrémité, il pourrait être capable de déclencher des attaques de vérification de connexité. Les conclusions possiblement fausses qu'un attaquant peut essayer et causer sont :

Faux invalide : un attaquant peut tromper une paire d'agents pour qu'ils pensent qu'une paire candidate est invalide, quand elle ne l'est pas. Cela peut être utilisé pour amener un agent à préférer une candidate différente (comme une injectée par l'attaquant) ou pour perturber un appel en forçant toutes les candidates à échouer.

Faux valide : un attaquant peut tromper une paire d'agents pour qu'ils pensent qu'une paire candidate est valide, quand elle ne l'est pas. Cela peut amener un agent à poursuivre une session mais n'être pas capable de recevoir de données.

Fausse candidate reflet d'homologue : un attaquant peut amener un agent à découvrir une nouvelle candidate reflet de l'homologue quand elle n'est pas attendue. Cela peut être utilisé pour rediriger le flux de données sur une cible de DoS ou sur l'attaquant, pour l'espionnage ou autre.

Faux valide sur fausse candidate : un attaquant a déjà convaincu un agent qu'il y a une candidate avec une adresse qui n'achemine en fait pas sur cet agent (par exemple, en injectant une fausse candidate reflet de l'homologue ou une fausse candidate reflet de serveur). L'attaquant lance alors un attaque qui force les agents à croire que cette candidate est valide.

Si un attaquant peut causer une fausse candidate reflet de l'homologue ou une fausse valide sur une fausse candidate, il peut lancer une des attaques décrites dans la [RFC5389].

Pour forcer le résultat faux invalide, l'attaquant doit attendre l'envoi de la vérification de connexité à partir d'un des agents. Quand elle est envoyée, l'attaquant doit injecter une réponse fabriquée avec une réponse d'erreur irrécupérable (comme un code 400) ou éliminer la réponse afin qu'elle n'atteigne jamais l'agent. Cependant, comme la candidate est, en fait, valide, la demande originale peut atteindre l'agent homologue et résulter en une réponse de succès. L'attaquant doit forcer ce paquet ou sa réponse à être éliminé à travers une attaque de DoS, une perturbation de couche 2 du réseau, ou une autre technique. Si il ne fait pas cela, la réponse de succès va aussi atteindre l'origine, l'alertant d'une possible attaque. La capacité de l'attaquant à générer une fausse réponse est atténuée par le mécanisme d'accréditif à court terme de STUN. Afin que cette réponse soit traitée, l'attaquant a besoin du mot de passe. Si la signalisation d'échange de candidates est sécurisée, l'attaquant n'aura pas le mot de passe, et sa réponse sera éliminée.

Des erreurs dures ICMP falsifiées (de type 3, codes 2 à 4) peuvent aussi être utilisées pour créer des résultats faux invalides. Si un agent ICE met en œuvre une réponse à ces erreurs ICMP, l'attaquant est capable de générer un message ICMP qui est livré à l'agent envoyeur de la vérification de connexité. La validation du message d'erreur ICMP par l'agent est sa seule défense. Pour le type 3 code=4, l'en-tête IP externe ne fournit pas de validation, sauf si la vérification de connexité a été envoyée avec DF=0. Pour les codes 2 ou 3, qui sont générés par l'hôte, l'adresse est supposée être une des adresse IP candidate d'hôte, reflet, ou relais de l'agent distant. Le message ICMP inclut l'en-tête IP et l'en-tête UDP du message qui déclenche l'erreur. Ces champs doivent aussi être validés. L'accès de destination IP et UDP doit correspondre à l'adresse et accès de la candidate ciblée ou à l'adresse de base de la candidate. L'adresse IP et accès de source peuvent être toute candidate pour la même adresse de base de l'agent qui envoie la vérification de connexité. Donc, tout attaquant qui a accès à l'échange des candidates va avoir les informations nécessaires. Donc, la validation est une faible défense, et l'envoi d'attaques en détournement d'ICMP est aussi possible pour des attaquants hors chemin à partir d'un nœud dans un réseau sans validation d'adresse de source.

Forcer le résultat faux valide fonctionne d'une façon similaire. L'attaquant doit attendre la demande Binding provenant de chaque agent et injecte une fausse réponse de succès. Là encore, du fait du mécanisme d'accréditif à court terme de STUN , pour que l'attaquant injecte une réponse valide de succès, il a besoin du mot de passe. Autrement, l'attaquant peut acheminer à l'agent (par exemple, en utilisant un mécanisme de tunnelage) une réponse valide de succès, qui va normalement être éliminée ou rejetée par le réseau.

Forcer le résultat de fausse candidate reflet de l'homologue peut être fait avec de demandes ou réponses fabriquées, ou avec des répétitions. On considère d'abord le cas de fausses demandes et réponses. Cela exige que l'attaquant envoie une demande Binding à un agent avec une adresse IP et accès de source pour la fausse candidate. De plus, l'attaquant doit attendre une demande Binding provenant de l'autre agent et générer une fausse réponse avec un attribut XOR-MAPPED-ADDRESS contenant la fausse candidate. Comme les autres attaques décrites ici, cette attaque est atténuée par les mécanismes d'intégrité de message de STUN et les échanges de candidates sécurisés.

Forcer le résultat de fausse candidate reflet de l'homologue avec des répétitions de paquet est différent. L'attaquant attend jusqu'à ce qu'un des agents envoie une vérification. Il intercepte cette demande et la répète vers l'autre agent avec une adresse IP de source falsifiée. Il a aussi besoin d'empêcher la demande originale d'atteindre l'agent distant, en lançant une attaque de DoS pour causer l'élimination du paquet ou en forçant son élimination en utilisant des mécanismes de couche 2. Le paquet répété est reçu chez l'autre agent, et accepté, car la vérification d'intégrité réussit (la vérification d'intégrité ne peut pas couvrir et ne couvre pas l'adresse et accès IP de source). Il lui est alors répondu. Cette réponse va contenir un attribut XOR-MAPPED-ADDRESS avec la fausse candidate, et elle va être envoyée à cette fausse candidate. L'attaquant a alors besoin de la recevoir et de la relayer à l'origine.

L'autre agent va alors initier une vérification de connexité sur cette fausse candidate. Cette validation doit réussir. Cela exige que l'attaquant force un faux valide sur une fausse candidate. L'injection de fausses demandes ou réponses pour atteindre ce but est empêchée en utilisant les mécanismes d'intégrité de STUN et de l'échange de candidates. Donc, cette attaque ne peut être lancée qu'à travers des répétitions. Pour faire cela, l'attaquant doit intercepter la vérification sur cettc fausse candidate et la répéter sur l'autre agent. Ensuite, il doit aussi intercepter la réponse et la répéter.

Cette attaque est très difficile à lancer sauf si l'attaquant est identifié par la fausse candidate. C'est parce que elle exige que l'attaquant intercepte et répète les paquets envoyés par deux hôtes différents. Si les deux agents sont dans des réseaux différents (par exemple, à travers l'Internet public) cette attaque peut être difficile à coordonner, car elle doit se produire contre deux différents points d'extrémité sur des parties différentes du réseau en même temps.

Si l'attaquant lui-même est identifié par la fausse candidate, l'attaque est plus facile à coordonner. Cependant, si le chemin des données est sécurisé (par exemple, en utilisant le protocole sûr de transport en temps réel (SRTP, *Secure Real-time Transport Protocol*) [RFC3711], l'attaquant ne va pas être capable de traiter les paquets de données, mais va seulement être capable de les éliminer, désactivant effectivement le flux de données. Cependant, cette attaque exige que l'agent élimine les paquets afin d'empêcher la vérification de connexité d'atteindre la cible. Dans ce cas, si le but est d'interrompre le flux de données, il est bien plus facile de juste le perturber avec le même mécanisme, plutôt que d'attaquer ICE.

19.3 Attaques contre le rassemblement d'adresses de reflet du serveur

Les points d'extrémité ICE utilisent les demandes Binding de STUN pour rassembler les candidates reflets de serveur provenant d'un serveur STUN. Ces demandes ne sont en aucune façon authentifiées. Par conséquent, il y a de nombreuses techniques qu'un attaquant peut employer pour fournir au client une fausse candidate reflet de serveur :

- Un attaquant peut compromettre le DNS, causant le retour par les interrogations au DNS d'une adresse de serveur STUN félon. Ce serveur peut fournir au client de fausses candidates reflets de serveur. Cette attaque est atténuée par la sécurité du DNS, bien que DNSSEC ne soit pas exigé pour la traiter.

- Un attaquant qui peut observer les messages STUN (comme un attaquant sur un segment de réseau partagé, comme une Wi-Fi) peut injecter une fausse réponse valide qui va être acceptée par le client.

- Un attaquant peut compromettre un serveur STUN et faire qu'il envoie des réponses avec des adresses transposées incorrectes.

Une fausse adresse transposée apprise par ces attaques va être utilisée comme candidate reflet de serveur dans l'établissement de la session ICE. Pour que cette candidate soit réellement utilisée pour les données, l'attaquant a aussi besoin d'attaquer les vérifications de connexité, et en particulier, forcer un faux valide sur une fausse candidate. Cette attaque est très difficile à lancer si la fausse adresse identifie une quatrième partie (ni l'initiateur, ni le répondant, ni

l'attaquant) car cela exige d'attaquer les vérifications générées par chaque agent ICE dans la session et est empêchée par SRTP si il identifie l'attaquant lui-même.

Si l'attaquant choisit de ne pas attaquer les vérifications de connexité, le pire qu'il peut faire est d'empêcher la candidate reflet de serveur d'être utilisée. Cependant, si l'agent d'homologue a au moins une candidate accessible par l'agent attaqué, les vérifications de connexité STUN elles-mêmes vont fournir une candidate reflet de l'homologue qui peut être utilisée pour l'échange de données. Les candidates reflet de l'homologue sont généralement préférées aux candidates reflets de serveur. À ce titre, une attaque seulement sur le rassemblement d'adresses STUN ne va pas avoir d'impact du tout sur une session.

19.4 Attaques contre le rassemblement de candidats relayés

Un attaquant pourrait tenter de perturber le rassemblement des candidates relayées, forçant le client à croire qu'il a une fausse candidate relayée. Les échanges avec le serveur TURN sont authentifiés en utilisant un accréditif à long terme. Par conséquent, l'injection de fausses réponses ou demandes ne va pas fonctionner. De plus, à la différence des demandes Binding, les demandes Allocate ne sont pas susceptibles d'attaques en répétition avec des adresses et accès IP de source modifiés, car les adresses et accès IP de source ne sont pas utilisés pour fournir au client ses candidates relayées.

Même si un attaquant a fait croire au client une fausse candidate relayée, les vérifications de connexité font qu'une telle candidate ne sera utilisée que si elles réussissent. Donc, un attaquant a besoin de lancer une fausse valide sur une fausse candidate, conformément à ce qui est expliqué ci-dessus, ce qui est une attaque très difficile à coordonner.

19.5 Attaques de l'intérieur

En plus des attaques où l'attaquant est un tiers qui essaye d'insérer des informations ou messages STUN de fausses candidates, il y a des attaques possibles avec ICE quand l'attaquant est un participant authentifié et valide dans l'échange ICE.

19.5.1 Attaque d'amplification STUN

L'attaque d'amplification STUN est similaire à l'attaque du "marteau vocal", où l'attaquant amène les autres agents à diriger les paquets vocaux sur la cible de l'attaque. Cependant, au lieu de paquets vocaux dirigés sur la cible, les vérifications de connexité STUN sont dirigées sur la cible. L'attaquant envoie un grand nombre de candidates, disons, 50. L'agent qui répond reçoit les informations de candidates et commence ses vérifications, qui sont dirigées sur la cible, et par conséquent, ne génèrent jamais de réponse. Dans le cas de WebRTC, l'utilisateur ne pourrait même pas savoir que cette attaque est en cours, car elle pourrait être déclenchée en arrière plan par le code JavaScript malveillant que l'utilisateur est allé chercher. Celui qui répond va commencer une nouvelle vérification de connexité toutes les Ta ms (disons, Ta=50 ms). Cependant, les temporisateurs de retransmission sont réglés à un grand nombre à cause du grand nombre de candidates. Par conséquent, les paquets vont être envoyés à un intervalle de un toutes les Ta millisecondes et ensuite avec des intervalles croissants. Donc, STUN ne va pas envoyer de paquets à un rythme plus rapide que l'envoi des données, et les paquets STUN ne persistent que brièvement, jusqu'à ce que ICE soit défaillant pour la session. Néanmoins, c'est un mécanisme d'amplification.

Il est impossible d'éliminer l'amplification, mais le volume peut être réduit par diverses heuristiques. Les agents ICE DEVRAIENT limiter le nombre total de vérifications de connexité qu'ils effectuent à 100. De plus, les agents PEUVENT limiter le nombre de candidates qu'ils veulent accepter.

Fréquemment, les protocoles qui souhaitent éviter ces sortes d'attaques forcent l'initiateur à attendre une réponse avant d'envoyer le prochain message. Cependant, dans le cas de ICE, ce n'est pas possible. Il n'est pas possible de différencier les deux cas suivants :

- il n'y a pas eu de réponse parce que l'initiateur est utilisé pour lancer une attaque de DoS contre une cible innocente qui ne va pas répondre ;
- il n'y a pas eu de réponse parce que l'adresse et l'accès IP sont injoignables par l'initiateur.

Dans le second cas, une autre vérification va être envoyé à la prochaine occasion, alors que dans le premier cas, aucune autre vérification ne va être envoyée.

20. Considérations relatives à l'IANA

La spécification originale de ICE enregistrait quatre attributs STUN et une nouvelle réponse d'erreur STUN. Les attributs et la réponse d'erreur STUN sont reproduits ici. De plus, la présente spécification enregistre une nouvelle option ICE.

20.1 Attributs STUN

L'IANA a enregistré quatre attributs STUN :
0x0024 PRIORITY
0x0025 USE-CANDIDATE
0x8029 ICE-CONTROLLED
0x802A ICE-CONTROLLING

20.2 Réponses d'erreur STUN

L'IANA a enregistré le code de réponse d'erreur STUN suivant :

487 : Conflit de rôle. Le client affirme un rôle ICE (contrôleur ou contrôlé) en conflit avec le rôle du serveur.

20.3 Options ICE

L'IANA a enregistré l'option ICE suivante dans le sous registre "Options ICE" du registre "Interactive Connectivity Establishment (ICE)", suivant les procédures définies dans la [RFC6336].

Nom d'option ICE : ice2
Contact : IESG. iesg@ietf.org
Contrôleur des changements : IESG
Description : l'option ICE indique que l'agent ICE qui utilise l'option ICE est mis en œuvre conformément à la RFC 8445.
Référence : RFC 8445

21. Changements par rapport à la RFC 5245

L'objet de cette mise à jour de la spécification ICE est de :

- Préciser les procédures de la RFC 5245.

- Faire des changements techniques, dus à la découverte de fautes dans la RFC 5245 et des retours de la communauté qui a mis en œuvre et déployé des applications de ICE fondées sur la RFC 5245.

- Rendre les procédures indépendantes du protocole de signalisation, en supprimant les procédures SIP et SDP. Les procédures spécifiques d'un protocole de signalisation vont être définies dans des documents d'usage séparés. La [RFC8839] définit l'usage ICE avec SIP et SDP.

Les changements techniques suivants ont été faits :
- Suppression de la désignation agressive.
- Modification des procédures de calcul des états de paire candidate et de programmation des vérifications de connexité.
- Modification des procédures de calcul de Ta et RTO.
- Suppression de définitions de liste de contrôle active et de liste de contrôle gelée.
- Ajout de l'option "ice2" ICE.
- Modification des considérations IPv6.
- Suppression de l'usage avec no-op pour les maintiens en vie, et des maintiens en vie avec des homologues non ICE.

22. Références

22.1 Références normatives

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, DOI 10.17487/RFC2119, mars 1997. *(MàJ par* RFC8174)

[RFC4941] T. Narten et autres, "Extensions de confidentialité pour l'auto configuration d'adresse sans état dans IPv6", septembre 2007, DOI 10.17487/RFC4941. (*D.S. ; remplace* RFC3041 ; r*emplacée par* RFC8981*)*

[RFC5389] J. Rosenberg et autres, "Utilitaires de traversée de session pour les NAT (STUN)", octobre 2008, DOI 10.17487/RFC5389. *(P.S. ; remplace* RFC3489 *; remplacée par RFC*8489)

[RFC5766] R. Mahy, P. Matthews, J. Rosenberg, "Traversée de NAT au moyen d'un relais (TURN) : Extensions de relais aux utilitaires de traversée de session pour les NAT (STUN)", avril 2010, DOI 10.17487/RFC5766. *(P. S. ; MàJ par* RFC8155 *; Remplacée par* RFC8656)

[RFC6336] M. Westerlund, C. Perkins, "Registre de l'IANA pour les options d'établissement de la connexité interactive (ICE)", juillet 2011, DOI 10.17487/RFC6336. *(MàJ la* RFC5245, *remplacée par* RFC8839 ; *P.S.)*

[RFC6724] D. Thaler, R. Draves, A. Matsumoto, T. Chown, "Choix de l'adresse par défaut pour IPv6", septembre 2012, DOI 10.17487/RFC6724. (*Remplace la* RFC3484) *(P.S.)*

[RFC8174] B. Leiba, "Ambiguïté des mots clés en majuscules ou minuscules dans la RFC2119", DOI 10.17487/RFC8174, mai 2017. BCP14. *(MàJ RFC*2119)

22.2 Références pour information

[RFC1918] Y. Rekhter et autres, "Allocation d'adresse pour les internets privés", BCP 5, février 1996, DOI 10.17487/RFC1918.

[RFC2475] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang et W. Weiss, "Architecture pour services différenciés", décembre 1998, DOI 10.17487/RFC2475. *(MàJ par* RFC3260)

[RFC3102] M. Borella et autres, "IP spécifique de domaine : le cadre", octobre 2001, DOI 10.17487/RFC3102. *(Exp.)*

[RFC3103] M. Borella et autres, "IP spécifique de domaine : Spécification du protocole", octobre 2001, DOI 10.17487/RFC3103. *(Expérimentale)*

[RFC3235] D. Senie, "Traducteurs d'adresse réseau (NAT) : lignes directrices pour une conception d'application facile à comprendre", janvier 2002, DOI 10.17487/RFC3235. *(Information)*

[RFC3261] J. Rosenberg et autres, "SIP : Protocole d'initialisation de session", juin 2002, DOI 10.17487/RFC3261. (*Mise à jour par* 3265, 3853, 4320, 4916, 5393, 6665, 8217, 8760)

[RFC3264] J. Rosenberg et H. Schulzrinne, "Modèle d'offre/réponse avec le protocole de description de session (SDP)", juin 2002, DOI 10.17487/RFC3264. *(P.S. ; MàJ par* RFC8843, 9143)

[RFC3303] P. Srisuresh et autres, "Architecture et cadre de communication par boîtier de médiation", août 2002, DOI 10.17487/RFC3303. *(Information)*

[RFC3424] L. Daigle, éd., IAB, "Considérations de l'IAB sur l'auto correction d'adressage unilatérale (UNSAF) à travers la traduction d'adresse réseau", novembre 2002, DOI 10.17487/RFC3424. *(Information)*

[RFC3489] J. Rosenberg et autres, "STUN - Simple traversée par le protocole de datagramme d'utilisateur (UDP) des traducteurs d'adresse réseau (NAT)", mars 2003, DOI 10.17487/RFC3489. (*Obsolète, voir* RFC5389) *(P.S.)*

[RFC3550] H. Schulzrinne, S. Casner, R. Frederick et V. Jacobson, "RTP : un protocole de transport pour les applications

en temps réel", STD 64, DOI 10.17487/RFC3550, juillet 2003. *(MàJ par* RFC7164, RFC7160, RFC8083, RFC8108, RFC8860)

[RFC3605] C. Huitema, "Attribut du protocole de contrôle en temps réel (RTCP) dans le protocole de description de session (SDP)", octobre 2003, DOI 10.17487/RFC3605. *(P.S.)*

[RFC3711] M. Baugher et autres, "Protocole de transport sécurisé en temps réel (SRTP)", mars 2004, DOI 10.17487/RFC3711. *(P.S. ; MàJ par* RFC9335*)*

[RFC3725] J. Rosenberg et autres, "Bonnes pratiques actuelles pour la commande d'appel de tiers (3pcc) dans le protocole d'initialisation de session (SIP)", avril 2004, DOI 10.17487/RFC3725. (BCP0085)

[RFC3879] C. Huitema, B. Carpenter, "Les adresses IPv6 de site local en envoi individuel sont déconseillées", septembre 2004, DOI 10.17487/RFC3879. *(P.S.)*

[RFC4038] M-K. Shin et autres, "Aspects de l'application de la transition vers IPv6", mars 2005, DOI 10.17487/RFC4038. *(Information)*

[RFC4091] G. Camarillo, J. Rosenberg, "Sémantique des types d'adresse de réseau de remplacement (ANAT) pour le cadre de groupage du protocole de description de session (SDP)", juin 2005, DOI 10.17487/RFC4091. *(P.S.)*

[RFC4092] G. Camarillo, J. Rosenberg, "Usage de la sémantique des types d'adresse de réseau de remplacement (ANAT) du protocole de description de session (SDP) dans le protocole d'initialisation de session (SIP)", juin 2005, DOI 10.17487/RFC4092. *(P.S.)*

[RFC4103] G. Hellstrom, P. Jones, "Charge utile RTP pour conversation textuelle", juin 2005, DOI 10.17487/RFC4103. *(P.S. ;Remplace* RFC2793 *, MàJ par* RFC9071*)*

[RFC4291] R. Hinden, S. Deering, "Architecture d'adressage IP version 6", février 2006, DOI 10.17487/RFC4291. (*MàJ par* 5952 et 6052, 8064) *(D.S.)*

[RFC4566] M. Handley, V. Jacobson et C. Perkins, "SDP : Protocole de description de session", juillet 2006, DOI 10.17487/RFC4566. *(P.S. ; remplacée par RFC8866)*

[RFC4787] F. Audet, éd., C. Jennings, "Exigences sur le comportement des traducteurs d'adresse réseau (NAT) pour UDP en envoi individuel", janvier 2007, DOI 10.17487/RFC4787. (BCP0127) *(MàJ par* RFC7857)

[RFC5245] J. Rosenberg,"Établissement de connexité interactive (ICE) : Protocole pour la traversée de traducteur d'adresse réseau (NAT) pour les protocoles d'offre/réponse", avril 2010, DOI 10.17487/RFC5245. (*P. S. ;remplace* RFC4091, 4092 *; remplacée par* 8445)

[RFC5382] S. Guha et autres, "Exigences sur le comportement des NAT pour TCP", octobre 2008, DOI 10.17487/RFC5382. (BCP0142) *(MàJ par* RFC7857)

[RFC5761] C. Perkins, M. Westerlund, "Multiplexage de paquets de données et de contrôle RTP sur un seul accès", avril 2010, DOI 10.17487/RFC5761. (*MàJ* RFC3550, RFC3551). *(P. S., MàJ par* RFC8035)

[RFC6080] D. Petrie, S. Channabasappa, éd.. "Cadre pour la livraison de profil d'agent d'utilisateur du protocole d'initialisation de session", mars 2011, DOI 10.17487/RFC6080. *(P. S.)*

[RFC6146] M. Bagnulo, P. Matthews, I. van Beijnum, "NAT64 à états pleins : Traduction d'adresse et protocole réseau de clients IPv6 en serveurs IPv4", avril 2011, DOI 10.17487/RFC6146. *(P.S.)*

[RFC6147] M. Bagnulo et autres, "DNS64 : Extensions au DNS pour la traduction d'adresse réseau de clients IPv6 en serveurs IPv4", avril 2011, DOI 10.17487/RFC6147. *(P.S.)*

[RFC6298] V. Paxson, M. Allman, J. Chu, M. Sargent, "Calcul du temporisateur de retransmission de TCP", DOI 10.17487/RFC6298, juin 2011. *(Remplace la* RFC2988) *(MàJ la* RFC1122) *(P.S.)*

[RFC6544] J. Rosenberg et autres, "Candidats TCP avec établissement de connexité interactive (ICE)", mars 2012, DOI 10.17487/RFC6544. *(P.S.)*

[RFC6928] J. Chu et autres, "Augmentation de la fenêtre initiale de TCP", avril 2013, DOI 10.17487/RFC6928. *(Exp.)*

[RFC7050] T. Savolainen, J. Korhonen, D. Wing, "Découverte du préfixe IPv6 utilisé pour la synthèse d'adresse IPv6", novembre 2013, DOI 10.17487/RFC7050. *(P.S. ; MàJ par* RFC8880*)*

[RFC7721] A. Cooper, F. Gent, D. Thaler, "Considérations de sécurité et de confidentialité pour les mécanismes de génération d'adresse IPv6", mars 2016, DOI 10.17487/RFC7721. *(Information)*

[RFC7825] J. Goldberg, et autres, "Mécanisme de traversée de NAT pour support contrôlé par RTSP", décembre 2015, DOI 10.17487/RFC7825. *(P.S.)*

[RFC8421] P. Martinsen, T. Reddy, P. Patil, "Lignes directrices pour l'établissement de connexité interactive (ICE) multi rattachements et double piles IPv4/IPv6", juillet 2018, DOI 10.17487/RFC8421. BCP 217.

[RFC8829] J.Uberti, G. Shieb, "Exigences pour le traitement d'adresse IP WebRTC", janvier 2021. *(P.S.)* (DOI : 10.17487/RFC8829) *(Remplacée par* RFC9429*)*

[RFC8839] M. Petit-Hugenin, et autres, "Procédure d'offre/réponse du protocole de description de session pour ICE", janvier 2021. *(P.S. ; remplace* RFC5245*, et* RFC 6336*)* (DOI : 10.17487/RFC8839)

Appendice A. Mises en œuvre légères et complètes

ICE permet deux types de mises en œuvre. Une mise en œuvre complète prend en charge les rôles de contrôleur et de contrôlé dans une session et peut aussi effectuer le rassemblement d'adresses. À l'opposé, une mise en œuvre légère est une mise en œuvre minimaliste qui fait peu mais répond aux vérifications STUN, et elle prend seulement en charge le rôle de contrôlé dans une session.

Parce que ICE exige que les deux points d'extrémité le prennent en charge afin de bénéficier à l'un et l'autre des points d'extrémité, un déploiement incrémentaire de ICE dans un réseau est plus compliqué. De nombreuses sessions impliquent un point d'extrémité qui n'est pas lui-même derrière un NAT et personne qui se soucie de la traversée de NAT. Un cas très courant est d'avoir un point d'extrémité qui exige une traversée de NAT (comme un téléphone VoIP ou un téléphone logique) qui passe un appel à un de ces appareils. Même si le téléphone supporte une mise en œuvre ICE complète, ICE ne va pas être utilisé du tout si l'autre appareil ne le prend pas en charge. La mise en œuvre légère permet un point d'entrée à faible coût pour ces appareils. Une fois qu'elles prennent en charge la mise en œuvre légère, les mises en œuvre complètes peuvent se connecter à elles et bénéficier de tous les avantages de ICE.

Par conséquent, une mise en œuvre légère n'est appropriée que pour les appareils qui vont *toujours* être connectés à l'Internet public et avoir une adresse IP publique à laquelle ils peuvent recevoir des paquets de tout correspondant. ICE ne va pas fonctionner quand une mise en œuvre légère est placée derrière un NAT.

ICE permet à une mise en œuvre légère d'avoir une seule candidate hôte IPv4 et plusieurs adresses IPv6. Dans ce cas, les paires candidates sont choisies par l'agent contrôleur en utilisant un algorithme statique, comme celui de la RFC 6724, qui est recommandé par la présente spécification. Cependant, les mécanismes statiques pour le choix d'adresses sont toujours enclins à l'erreur, car ils ne peuvent jamais refléter la topologie réelle ou fournir des garanties réelles de connexité. Ils sont toujours heuristiques. Par conséquent, si un agent ICE met en œuvre ICE juste pour choisir entre ses adresses IPv4 et IPv6, et si aucune de ses adresses IP n'est derrière un NAT, l'usage de ICE complet est toujours RECOMMANDÉ afin de fournir la forme la plus robuste de choix d'adresses possible.

Il est important de noter que la mise en œuvre légère a été ajoutée à cette spécification pour fournir un appui à la mise en œuvre complète. Même pour les appareils qui sont toujours connectés à l'Internet public avec juste une seule adresse IPv4, une mise en œuvre complète est préférable si elle est réalisable. Les mises en œuvre complètes obtiennent aussi l'avantage de sécurité de ICE sans relation avec la traversée de NAT. Finalement, il est souvent le cas qu'un appareil qui se trouve lui-même avec une adresse publique aujourd'hui va être demain placé dans un réseau où il va être derrière un NAT. Il est difficile de savoir de façon définitive, sur la durée de vie d'un appareil ou produit, si il va toujours être utilisé sur l'Internet public. Une mise en œuvre complète donne l'assurance que la communication va toujours fonctionner.

Appendice B. Motivations de la conception

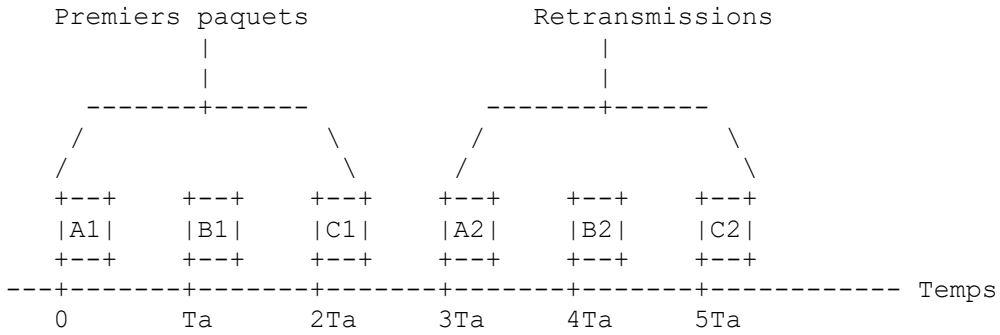
ICE contient un certain nombre de comportements normatifs qui peuvent eux-mêmes être simples mais dériver de pensées compliquées ou non évidentes ou de cas d'utilisation qui méritent plus de discussion. Comme ces motivations de conception ne sont pas nécessaires pour comprendre les besoins de la mise en œuvre, elles sont discutées ici. Cet Appendice n'est pas normatif.

B.1 Régulation des transactions STUN

Les transactions STUN utilisées pour rassembler les candidates et pour vérifier la connexité sont régulées à un taux approximatif d'une nouvelle transaction toutes les Ta millisecondes. Chaque transaction a, à son tour, un temporisateur de retransmission RTO qui est aussi fonction de Ta. Pourquoi ces transactions sont elles régulées, et pourquoi des formules sont elles utilisées ?

L'envoi de ces demandes STUN va souvent avoir pour effet de créer des liens sur les appareils de NAT entre le client et les serveurs STUN. L'expérience a montré que de nombreux appareils de NAT ont des limites supérieures sur le taux auquel ils vont créer de nouveaux liens. Les discussions dans le groupe de travail ICE de l'IETF durant l'élaboration de la présente spécification ont conclu que une fois toutes les 5 ms est bien accepté. C'est pourquoi Ta a une limite inférieure de 5 ms. De plus, la transmission de ces paquets sur le réseau utilise la bande passante et doit être limitée en débit par l'agent ICE. Les déploiements fondés sur des versions antérieures de projets de la [RFC5245] tendaient à surcharger les liaisons d'accès avec des contraintes de débit et avaient de mauvaises performances globales, en plus d'un impact négatif sur le réseau. Par conséquent, la régulation assure que l'appareil de NAT ne se trouve pas surchargé et que le trafic reste à un débit raisonnable.

La définition d'un débit "raisonnable" est que STUN NE DOIT PAS utiliser plus de bande passante que ce que RTP lui-même va utiliser, une fois que les données commencent à s'écouler. La formule pour Ta est conçue de telle sorte que, si un paquet STUN est envoyé toutes les Ta secondes, il va consommer la même quantité de bande passante que les paquets RTP, totalisés sur tous les flux de données. Bien sûr, STUN a des retransmissions, et le désir est de les réguler aussi. Pour cette raison, RTO est réglé de telle façon que la première retransmission sur la première transaction se produise juste quand se produit la première demande STUN sur la dernière transaction. C'est ce que montre la figure ci-dessous :



Dans cette figure, il y a trois transactions qui vont être envoyées (par exemple, dans le cas du rassemblement de candidates, il y a trois paires de candidates hôte/serveur STUN). Ce sont les transactions A, B, et C. Le temporisateur de retransmission est réglé de telle façon que la première retransmission sur la première transaction (paquet A2) soit envoyée à l'instant 3Ta.

Les retransmissions suivantes, après la première vont se produire encore moins fréquemment que Ta millisecondes les unes des autres, parce que STUN utilise un retard à croissance exponentielle de ses retransmissions.

Ce mécanisme d'une régulation globale minimum de l'intervalle de 5 ms n'est pas généralement applicable aux protocoles de transport, mais il est applicable à ICE sur la base du raisonnement suivant.

- On commence par les règles suivantes qui vont être généralement applicables aux protocoles de transport :

  1. Soit MaxBytes le nombre maximum d'octets permis en instance dans le réseau au démarrage, qui DEVRAIT être 14600, comme défini dans la Section 2 de la [RFC6928].

  2. Soit HTO le temporisateur de transaction, qui DEVRAIT être 2*RTT si RTT est connu ou 500 ms autrement. Ceci est fondé sur le RTO pour les messages STUN d'après la [RFC5389] et le RTO initial de TCP, qui est 1 s dans la

[RFC6298].

3. Soit MinPacing l'intervalle minimum de régulation entre transactions, qui est 5 ms (voir ci-dessus).

- On observe que les agents ne connaissent normalement pas le RTT pour les transactions ICE (les vérifications de connexité en particulier) ce qui signifie que HTO va presque toujours être 500 ms.

- On observe qu'un MinPacing de 5 ms et un HTO de 500 ms donnent au plus 100 paquets/HTO, qui pour une vérification ICE typique de moins de 120 octets signifie un maximum de 12 000 octets en instance dans le réseau, ce qui est moins que le maximum exprimé par la règle 1.

- Donc, pour ICE, l'ensemble de règles se réduit à juste la règle MinPacing, qui est équivalente à avoir une valeur globale de Ta.

B.2 Candidats avec plusieurs bases

Le paragraphe 5.1.3 parle d'éliminer les candidates qui ont la même adresse de transport et base. Cependant, les candidates avec la même adresse de transport mais une base différente ne sont pas redondantes. Quand un agent ICE peut-il avoir deux candidates qui ont la même adresse IP et accès mais des bases différentes ? Considérons la topologie de la Figure 11 :

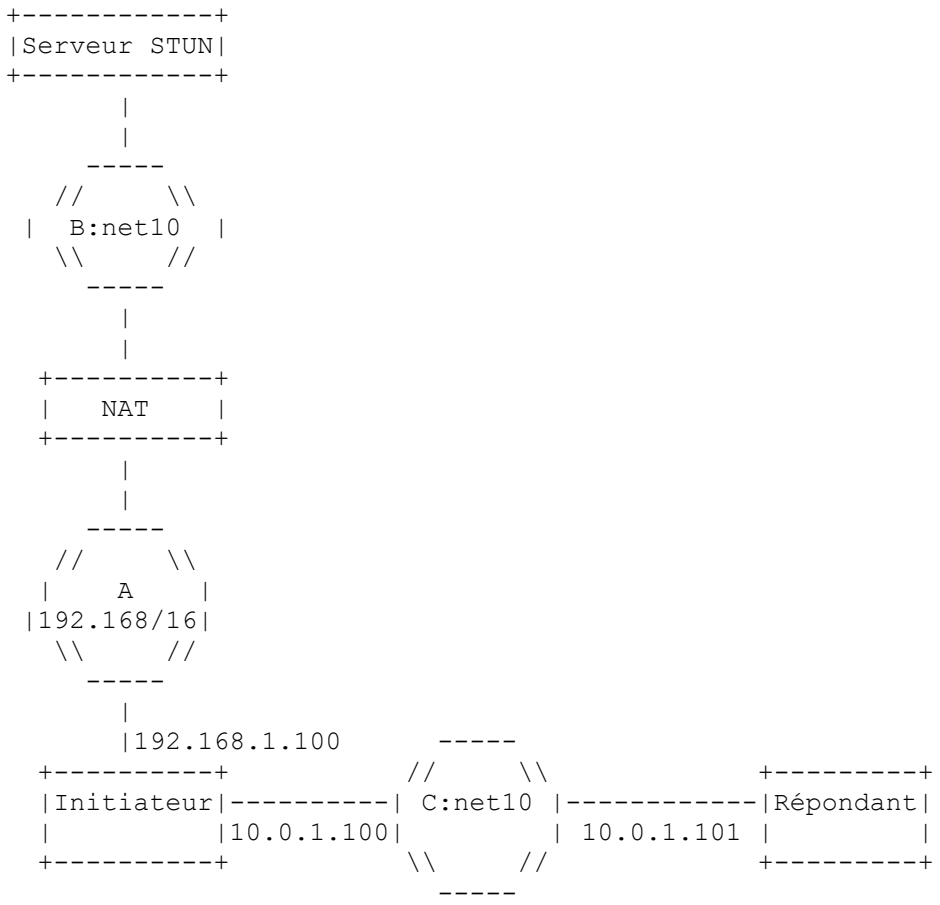


Figure 11 : Candidats identiques avec des bases différentes

Dans ce cas, l'agent initiateur est multi-rattachements. Il a une adresse IP, 10.0.1.100, sur le réseau C, qui est un réseau privé net 10. L'agent répondant est sur ce même réseau. L'agent initiateur est aussi connecté au réseau A, qui est 192.168/16, et a une adresse IP de 192.168.1.100. Il y a un NAT sur ce réseau, ouvrant sur le réseau B, qui est un autre réseau privé net 10, mais il n'est pas connecté au réseau C. Il y a un serveur STUN sur le réseau B.

L'agent initiateur obtient une candidate hôte sur son adresse IP au réseau C (10.0.1.100:2498) et une candidate hôte sur son adresse IP au réseau A (192.168.1.100:3344). Il effectue une interrogation STUN à son serveur STUN configuré à 192.168.1.100:3344. Cette interrogation passe à travers le NAT, qui alloue le lien 10.0.1.100:2498. Le serveur STUN

reflète cela dans la réponse Binding STUN. Maintenant, l'agent initiateur a obtenu une candidate reflet de serveur avec une adresse de transport identique à la candidate hôte (10.0.1.100:2498). Cependant, la candidate reflet de serveur a une base de 192.168.1.100:3344, et la candidate hôte a une base de 10.0.1.100:2498.

B.3 Objet des attributs Related-Address et Related-Port

L'attribut candidat contient deux valeurs qui ne sont pas du tout utilisées par ICE lui-même -- Adresse relative et Accès relatif. Pourquoi sont elles présentes ?

Il y a deux raisons à leur inclusion. La première est le diagnostic. Il est très utile de savoir la relation entre les différents types de candidates. En l'incluant, un agent ICE peut savoir quelle candidate relayée est associée à quelle candidate reflet, qui à son tour est associée à une candidate hôte spécifique. Quand les vérifications réussissent pour une candidate mais pas pour d'autres, cela fournit des diagnostics utiles sur ce qui se passe dans le réseau.

La seconde raison a à voir avec les mécanismes de qualité de service (QS) hors chemin. Quand ICE est utilisé dans des environnements comme PacketCable 2.0, des mandataires vont, en plus d'effectuer les opérations SIP normales, inspecter le SDP dans les messages SIP et extraire l'adresse IP et l'accès pour le trafic de données. Ils peuvent alors interagir, à travers les serveurs de politique, avec les routeurs d'accès dans le réseau, pour établir la QS garantie pour les flux de données. Cette QS est fournie en classant le trafic RTP sur la base d'un quintuplet et ensuite de lui fournir un taux garanti, ou de marquer son DSCP de façon appropriée. Quand un NAT résidentiel est présent, et qu'une candidate relayée est choisie pour les données, cette candidate relayée va être une adresse de transport sur un serveur TURN réel. Cette adresse ne dit rien de l'adresse de transport réelle dans le routeur d'accès qui pourrait être utilisé pour classer les paquets pour le traitement de QS. C'est plutôt la candidate reflet de serveur sur le serveur TURN qui est nécessaire. En portant la traduction en SDP, le mandataire peut utiliser cette adresse de transport pour demander la QS au routeur d'accès.

B.4 Importance du nom d'utilisateur STUN

ICE exige l'usage de la protection de l'intégrité de message avec STUN en utilisant sa fonction d'accréditif à court terme. L'accréditif de court terme réel est formé par l'échange de fragments de nom d'utilisateur dans l'échange de candidates. Le besoin de ce mécanisme va au delà de juste la sécurité ; il est en fait exigé en premier lieu pour le fonctionnement correct de ICE.

Considérons les agents ICE L, R, et Z. L et R sont dans une entreprise privée 1, qui utilise 10.0.0.0/8. Z est dans l'entreprise privée 2, qui utilise aussi 10.0.0.0/8. Il se trouve que R et Z ont tous deux une adresse IP 10.0.1.1. L envoi des candidates à Z. Z répond à L avec ses candidates hôtes. Dans ce cas, ces candidates sont 10.0.1.1:8866 et 10.0.1.1:8877. Il se trouve que R est dans une session en même temps et utilise aussi 10.0.1.1:8866 et 10.0.1.1:8877 comme candidates hôtes. Cela signifie que R est prêt à accepter des messages STUN sur ces accès, tout comme l'est Z. L va envoyer une demande STUN à 10.0.1.1:8866 et une autre à 10.0.1.1:8877. Cependant, cela ne va pas à Z comme attendu. À la place, cela va à R ! Si R leur répond, L va croire qu'il a la connexité avec Z, quand en fait il a la connexité avec un utilisateur complètement différent, R. Pour corriger cela, on utilise le mécanisme d'accréditif à court terme de STUN. Les fragments de nom d'utilisateur sont suffisamment aléatoires ; donc il est très improbable que R utilise les mêmes valeurs que Z. Par conséquent, R va rejeter la demande STUN car les accréditifs sont invalides. Par nature, les fragments de nom d'utilisateur STUN fournissent une forme d'identifiant d'hôte temporaire, liée à une session particulière établie au titre de l'échange de candidates.

Une conséquence malheureuse de la non unicité des adresses IP est que, dans l'exemple ci-dessus, R pourrait ne pas même être un agent ICE. Ce pourrait être n'importe quel hôte, et l'accès auquel le paquet STUN est dirigé pourrait être n'importe quel accès éphémère sur cet hôte. Si il y a une application qui écoute des paquets sur cette prise, et qu'elle n'est pas prête à traiter des paquets mal formés pour tout protocole utilisé, le fonctionnement de cette application pourrait être affecté. Heureusement, comme les accès échangés sont éphémères et généralement tirés d'une gamme dynamique ou enregistrée, il y a de bonnes chances pour que l'accès ne soit pas utilisé pour faire fonctionner un serveur sur l'hôte R, mais soit plutôt le côté agent d'un protocole. Cela diminue la probabilité de toucher un accès alloué, du fait de la nature temporaire de l'usage des accès dans cette gamme. Cependant, la possibilité d'un problème existe bien, et les déployeurs de réseau doivent y être prêts. Noter que ce n'est pas un problème spécifique de ICE ; des paquets errants peuvent arriver à un accès à tout moment pour tout type de protocole, en particulier ceux de l'Internet public. À ce titre, cette exigence est juste de réaffirmer une directive générale de conception pour les applications de l'Internet -- être prêt pour des paquets inconnus sur tout accès.

B.5 Formule de priorité de la paire candidate

La priorité pour une paire candidate a une forme impaire. C'est :

priorité de paire = 2^32*MIN(G,D) + 2*MAX(G,D) + (G>D?1:0)

Pourquoi cela ? Quand les paires candidates sont triées sur la base de cette valeur, le tri résultant a la propriété MAX/MIN. Cela signifie que les paires sont d'abord triées sur la base de la valeur décroissante du minimum des deux priorités. Pour des paires qui ont la même valeur de priorité minimum, la priorité maximum est utilisée pour les trier. Si les priorités maximum et minimum sont les mêmes, la priorité de l'agent contrôleur est utilisée comme départage dans la dernière partie de l'expression. Le facteur 2*32 est utilisé car la priorité d'une seule candidate est toujours moins que 2*32, résultant en la priorité de paire comme "enchaînement" des deux priorités composantes. Cela crée le tri MAX/MIN. MAX/MIN assure que, pour un agent ICE particulier, une candidate de priorité inférieure n'est jamais utilisée tant que toutes les candidates de priorité supérieure n'ont pas été essayées.

B.6 Pourquoi les maintiens en vie sont nécessaires

Une fois que les données commencent à s'écouler sur une paire candidate, il est encore nécessaire de garder en vie les liens aux NAT intermédiaires pour la durée de la session. Normalement, les paquets du flux de données eux-mêmes (par exemple, RTP) satisfont cet objectif. Cependant, plusieurs cas méritent une discussion plus approfondie. D'abord, dans certains usages de RTP, comme SIP, le flux de données peut être "mis en garde". Cela est accompli en utilisant les attributs SDP "sendonly" ou "inactive", comme défini dans la [RFC3264]. La RFC 3264 invite les mises en œuvre à cesser la transmission des données dans ces cas. Cependant, le faire peut causer la péremption des liens de NAT, et les données ne vont pas être capables de revenir de la mise en garde.

Ensuite, certains formats de charge utile RTP, comme le format de charge utile pour la conversation de texte [RFC4103], peuvent envoyer des paquets de façon si peu fréquente que l'intervalle excède les temporisations de lien de NAT.

Troisièmement, si la suppression de silence est utilisée, de longues périodes de silence peuvent causer la cessation de la transmission de données pendant suffisamment longtemps pour que les liens de NAT arrivent en fin de temporisation.

Pour ces raisons, on ne peut pas s'appuyer sur les paquets de données eux-mêmes. ICE définit un simple maintien en vie périodique qui utilise les indications STUN Binding. Cela rend ses exigences de bande passante très prévisibles et donc utilisables pour les réservations de QS.

B.7 Pourquoi préférer des candidates reflet d'homologue

Le paragraphe 5.1.2 décrit les procédures pour calculer la priorité d'une candidate sur la base de son type et ses préférences locales. Ce paragraphe exige que la préférence de type pour les candidates reflets d'homologue soit toujours supérieure à celle du reflet de serveur. Pourquoi cela ? La raison a à voir avec les considérations sur la sécurité de la Section 19. Il est bien plus facile à un attaquant de causer l'utilisation par un agent ICE d'une fausse candidate reflet de serveur que d'une fausse candidate reflet de l'homologue. Par conséquent, les attaques contre le rassemblement d'adresses avec des demandes Binding sont déjouées par ICE en préférant les candidates reflets d'homologue.

B.8 Pourquoi des indications Binding sont utilisées pour les maintiens en vie

Les maintiens en vie de données sont décrits à la Section 11. Ces maintiens en vie utilisent STUN quand les deux points d'extrémité sont à capacité ICE. Cependant, plutôt que d'utiliser une transaction de demande Binding (qui génère une réponse) les maintiens en vie utilisent une indication. Pourquoi cela ?

La principale raison a à voir avec les mécanismes de qualité de service du réseau. Une fois que les données commencent à s'écouler, les éléments du réseau vont supposer que le flux de données a une structure très régulière, utilisant des paquets périodiques à des intervalles fixes, avec la possibilité de gigue. Si un agent ICE envoie des paquets de données, et ensuite reçoit une demande Binding, il va devoir générer un paquet de réponse avec ses paquets de données. Cela va augmenter les exigences réelles de bande passante pour le quintuplet qui porte les paquets de données et introduit de la gigue dans la livraison de ces paquets. L'analyse a montré que c'est un souci dans certains réseaux d'accès de couche 2 qui utilisent des programmation très serrées des paquets pour les données.

De plus, l'utilisation d'une indication Binding permet de désactiver la vérification d'intégrité, ce qui peut résulter en de meilleures performances. Ceci est utile pour les points d'extrémité à grande échelle, comme les passerelles du réseau téléphonique public commuté (RTPC) et les contrôleurs de bordure de session (SBC, *Session Border Controller*).

B.9 Choix d'une préférence de type de candidate

Un critère pour choisir les valeurs de type et de préférence locale est l'utilisation d'un intermédiaire de données, comme un serveur TURN, un service de tunnel comme un serveur de VPN, ou un NAT. Avec un intermédiaire de données, si les données sont envoyées à cette candidate, elles vont d'abord transiter par l'intermédiaire de données avant d'être reçues. Un type de candidate qui implique un intermédiaire de données est la candidate relayée. Un autre type est la candidate hôte, qui est obtenue d'une interface de VPN. Quand les données transitent par un intermédiaire de données, elles peuvent avoir un effet positif ou négatif sur la latence entre transmission et réception. Elles peuvent ou non augmenter les pertes de paquet, parce que cela peut prendre des bonds de routeur supplémentaires. Cela peut augmenter le coût de fourniture du service, car les données vont être acheminées dans et hors d'un intermédiaire de données que fait fonctionner un fournisseur de services. Si ces problèmes sont importants, la préférence de type pour les candidates relayées doit être choisie avec soin.

Un autre critère pour choisir des préférences est la famille d'adresses IP. ICE fonctionne avec IPv4 et IPv6. Il fournit un mécanisme de transition qui permet à des hôtes double piles de préférer la connexité sur IPv6 mais de revenir à IPv4 en cas de déconnexion des réseaux v6. Une mise en œuvre DEVRAIT suivre les lignes directrices de la [RFC8421] pour éviter des délais excessifs de la phase de vérification de connexité si il existe des interruptions de chemins.

Un autre critère pour choisir des préférences est la connaissance de la topologie. Cela est avantageux pour les candidates qui utilisent des intermédiaires. Dans ces cas, si un agent ICE a une connaissance préconfigurée ou dynamiquement découverte de la proximité topologique des intermédiaires, il peut l'utiliser pour allouer de plus hautes préférences locales aux candidates obtenues d'intermédiaires plus proches.

Un autre critère pour choisir des préférences pourrait être la sécurité ou la confidentialité. Si un utilisateur est un télécommuteur, et donc est connecté à un réseau d'entreprise et à un réseau de rattachement local, l'utilisateur peut préférer que son trafic vocal soit acheminé sur le VPN ou tunnel similaire afin de le garder sur le réseau d'entreprise quand il communique au sein de l'entreprise mais peut utiliser le réseau local quand il communique avec des utilisateurs hors de l'entreprise. Dans ce cas, une adresse de VPN va avoir une plus forte préférence locale que toute autre adresse.

Appendice C. Bande passante pour les vérifications de connexité

Les tableaux ci-dessous montrent, pour IPv4 et IPv6, la bande passante requise pour effectuer les vérifications de connexité, en utilisant différentes valeurs de Ta (données en ms) et différentes tailles de ufrag (données en octets).

Les résultats ont été fournis par Jusin Uberti (Google) le 11 avril 2016.

Version IP : IPv4
Longueur de paquet (octets) : 108 + ufrag

ms	4	8	12	16
500	1,86 k	1,98 k	2,11 k	2,24 k
200	4,64 k	4,96 k	5,28 k	5,6 k
100	9,28 k	9,92 k	10,6 k	11,2 k
50	18,6 k	19,8 k	21,1 k	22,4 k
20	46,4 k	49,6 k	52,8 k	56,0 k
10	92,8 k	99,2 k	105 k	112 k
5	185 k	198 k	211 k	224 k
2	464 k	496 k	528 k	560 k
1	928 k	992 k	1,06 M	1,12 M

Version IP : IPv6
Longueur de paquet (octets) : 128 + ufrag

ms	4	8	12	16
500	2,18 k	2,3 k	2,43 k	2,56 k
200	5,44 k	5,76 k	6,08 k	6,4 k
100	10,9 k	11,5 k	12,2 k	12,8 k
50	21,8 k	23,0 k	24,3 k	25,6 k
20	54,4 k	57,6 k	60,8 k	64,0 k
10	108 k	115 k	121 k	128 k

5	217 k	230 k	243 k	256 k
2	544 k	576 k	608 k	640 k
1	1,09 M	1,15 M	1,22 M	1,28 M

Figure 12 : Bande passante pour les vérifications de connexité

Remerciements

La plupart du texte de ce document vient de la spécification ICE originale, la RFC 5245. Les auteurs tiennent à remercier tous ceux qui ont contribué à ce document. Pour les contributions supplémentaires à cette révision de la spécification, nous tenons à remercier Emil Ivov, Paul Kyzivat, Pal-Erik Martinsen, Simon Perrault, Eric Rescorla, Thomas Stach, Peter Thatcher, Martin Thomson, Justin Uberti, Suhas Nandakumar, Taylor Brandstetter, Peter Saint-Andre, Harald Alvestrand, et Roman Shpount. Ben Campbell a fait la révision AD, Stephen Farrell a fait la révision sec-dir, Stewart Bryant a fait la révision gen-art, Qin We a fait la révision ops-dir, et Magnus Westerlund a fait la révision tsv-art.

Adresse des auteurs

Ari Keranen
Ericsson
Hirsalantie 11
02420 Jorvas
Finland
mél: ari.keranen@ericsson.com

Christer Holmberg
Ericsson
Hirsalantie 11
02420 Jorvas
Finland
mél : christer.holmberg@ericsson.com

Jonathan Rosenberg
jdrosen.net
Monmouth, NJ
United States of America
mél : jdrosen@jdrosen.net
URI : http://www.jdrosen.net