Publication V

Jouni Mäenpää, Veera Andersson, Ari Keränen and Gonzalo Camarillo. Impact of Network Address Translator Traversal on Delays in Peer-to-Peer Session Initiation Protocol. In *2010 IEEE Global Telecommunications Conference (GLOBECOM 2010)*, Miami, USA, pp. 1-6, December 2010.

Impact of Network Address Translator Traversal on Delays in Peer-to-Peer Session Initiation Protocol

Jouni Mäenpää, Veera Andersson, Gonzalo Camarillo, and Ari Keränen
Ericsson Finland
{jouni.maenpaa, veera.andersson, gonzalo.camarillo, ari.keranen}@ericsson.com

***Abstract*—Peer-to-Peer Session Initiation Protocol (P2PSIP) is a distributed communication system being standardized in the Internet Engineering Task Force (IETF). Since it uses the peer-to-peer paradigm, P2PSIP faces the problems created by Network Address Translators (NATs); even peers located behind NATs need to be able to not only initiate connections to other peers but also accept connections initiated by other peers. In this paper, we study the impact of standardized NAT traversal solutions, namely Session Traversal Utilities for NAT (STUN), Traversal Using Relays around NAT (TURN), and Interactive Connectivity Establishment (ICE), on delays in P2PSIP overlay networks. These delays are studied from the viewpoint of wireless and wired nodes acting as clients in a P2PSIP overlay network running in the PlanetLab. The delays are also compared to those of the traditional client/server Session Initiation Protocol (SIP).**

I. Introduction

Peer-to-Peer Session Initiation Protocol (P2PSIP) is a new decentralized communication system being standardized in the Internet Engineering Task Force (IETF). P2PSIP uses the Chord Distributed Hash Table (DHT) to organize the participating peers in an overlay network. The Session Initiation Protocol (SIP) [1] is used to enable real-time communication between the nodes. SIP uses the P2PSIP overlay as a distributed database to map SIP address-of-record values to node identifiers that can be used to reach users participating in the system. The overlay network replaces the centralized proxy-registrar servers of traditional client/server SIP.

A significant challenge for any Peer-to-Peer (P2P) network is that many of the nodes wishing to participate in the system are not publicly reachable but located behind Network Address Translators (NATs). To enable P2P communication in environments with NATs, P2PSIP uses the standardized IETF NAT traversal solutions, namely Interactive Connectivity Establishment (ICE) [2], Traversal Using Relays around NAT (TURN) [3], and Session Traversal Utilities for NAT (STUN) [4].

In this paper, we study the impact of ICE, STUN, and TURN on call setup and connection establishment delays of P2PSIP and centralized client/server SIP. We study these delays both from the viewpoint of wireless and wired endpoints. The paper is structured as follows Section II introduces our P2PSIP prototype. Section III presents related work. Section IV describes the experiment setup. Section V presents the results. Section VI concludes the paper.

II. P2PSIP Prototype

Our P2PSIP prototype is implemented in the Java programming language. There are two versions, one for Java Standard Edition (J2SE) and one for Java Micro Edition (J2ME). The first runs on PCs and the latter on mobile phones. The prototype uses Peer-to-Peer Protocol (P2PP) [5] as the protocol between the peers in the overlay. The REsource LOcation And Discovery (RELOAD) peer protocol [6], which is currently being standardized in the IETF, is based on P2PP (the P2PP proposal was merged with RELOAD). The prototype uses SIP for call control. Both P2PP and SIP connections run over UDP. We chose UDP as the transport protocol due to the problems associated with TCP NAT traversal [7]. SIP uses the P2PSIP overlay as a lookup mechanism to map SIP address-of-record values to contact Uniform Resource Identifiers (URIs) and to establish direct connections between SIP user agents across NATs. The prototype uses the Chord DHT [8] to organize the overlay. Chord was chosen since the P2PSIP working group of the IETF specifies it as mandatory to implement [6]. The prototype uses STUN, TURN, and ICE for NAT traversal.

Nodes running the P2PSIP prototype can act as either peers or clients. Peers are part of the Chord ring and provide storage and routing services to other nodes. Clients do not participate in the ring and thus do not provide these services. Instead, clients access them by connecting to a peer. Peer and client modes are supported by both versions of the prototype. In addition to the P2PSIP prototype, we also used a SIP client in the experiments. Our SIP client uses the same SIP, ICE, STUN, and TURN stacks as the P2PSIP prototype.

A. NAT Traversal Using ICE, STUN, and TURN

The prototype uses ICE as the technique for NAT traversal for UDP-based streams, namely for P2PP, SIP, and Real Time Protocol (RTP). ICE uses the STUN protocol and its extension, TURN. STUN can be used by a host to determine the IP address and port allocated to it by a NAT, to test connectivity between two hosts, and as a keep-alive protocol to maintain NAT bindings. TURN is used in situations when two hosts are not able to communicate without the help of a relay. TURN allows a host to control the relay and to exchange packets with its peers using the relay.

To start ICE, a host needs to know a STUN or TURN server and to have a signaling path with the target host. The host first gathers a set of addresses, called candidates, that

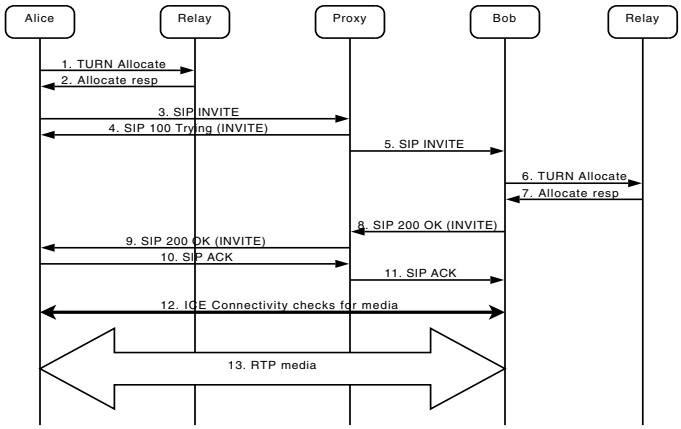

Fig. 1. SIP call setup

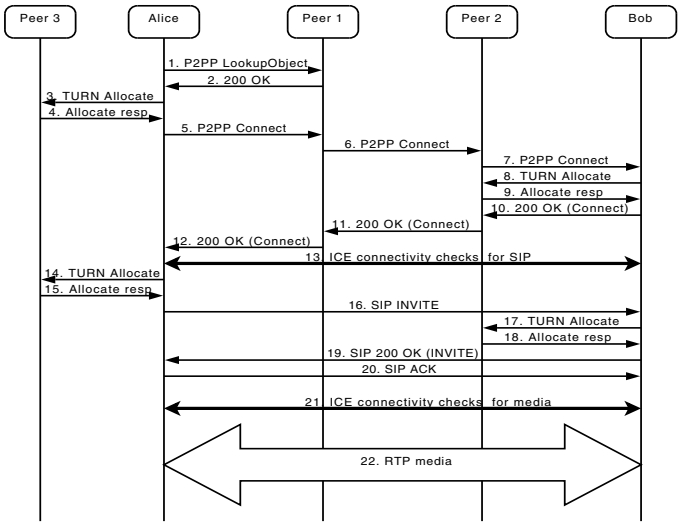

Fig. 2. P2PSIP call setup

can potentially be used to contact it. In addition to local interface addresses (host candidates), the host uses STUN to learn the address assigned to it by a NAT (server reflexive candidate) and TURN for allocating an address on a TURN server (relayed candidate). The candidates are exchanged over the signaling path. The host forms pairs using the local and remote candidates and arranges them by priority. Next, the host runs connectivity checks by sending STUN requests on each pair and by answering received checks. If a check succeeds on a pair, the pair is considered valid. The checks finish when a stopping criteria, whose selection ICE leaves as local optimization, is met. At that point, the initiator of ICE nominates one of the valid pairs for sending and receiving data.

B. Organizing Peers as STUN and TURN Servers

Every peer running our prototype can act as a STUN server. In complicated NAT topologies, a node may need more than one STUN server. Therefore, the node groups other nodes by the peer reflexive addresses it discovers through them to maximize the chances of achieving a direct connection [6]. One node from each group is selected as a STUN server. This increases the probability that there is a STUN server on the other side of each NAT behind which the node is located.

When a peer joins the overlay, it examines the reflexive candidates it learns while forming connections. If the peer has a public IP address and if all the reflexive addresses equal to the address of its local interface, the peer registers itself as a TURN server in the overlay. TURN server registration and discovery is done follows: peers capable of acting as TURN servers use a formula derived from the birthday paradox to determine how many pointers they need to store in the overlay to ensure that any peer can find a TURN server with a high probability using a bounded number of lookups. The formula takes as an input estimates of the total network size [9] and the worst case TURN server density.

C. Call Establishment

The call setup flow we use for client/server SIP is illustrated in Figure 1. In steps 1-2, the caller, Alice, performs a TURN Allocate transaction with her TURN server and possibly additional STUN transactions with STUN servers to obtain relayed and server reflexive candidates. In steps 3-4, Alice sends a SIP INVITE request to Bob, who is the callee, via the SIP proxy. The purpose of the INVITE request is to set up a VoIP call with one RTP audio stream. The INVITE request carries Alice's ICE candidates within the Session Description Protocol (SDP) [10] body of the request. On receiving the INVITE, Bob gathers his own ICE candidates in steps 6-7. The reason candidates are not gathered in advance but only when the need arises is to avoid the extra cost of keeping them alive while they are not being used. The candidates are returned to Alice via the proxy in steps 8-9 within a SIP 200 OK response. In steps 10-11, the response is acknowledged. In step 12, ICE connectivity checks are performed. In step 13, RTP media starts flowing on the connection ICE established.

Figure 2 shows the call setup flow of P2PSIP. In the figure, Peer 3 acts as a TURN server for Alice. Bob's TURN server is Peer 2. Both Alice and Bob are clients in the overlay. Alice uses Peer 1 as the peer through which her client accesses the overlay, whereas Bob uses Peer 2. In steps 1-2, Alice performs a P2PP lookup to discover Bob's contact information. The lookup may go through multiple hops in the overlay. In steps 3-4, Alice discovers her ICE candidates. In steps 5-7, Alice sends them in a P2PP Connect requests to Bob. The Connect request is sent via the overlay. The purpose of the Connect transaction is to set up a direct channel for SIP. In steps 8-9, Bob discovers his ICE candidates. They are returned to Alice in steps 10-12 in a Connect response. In step 13, connectivity checks are executed for SIP. In steps 14-15, Alice gathers new candidates for RTP. In step 16, they are sent to Bob in an INVITE request. The INVITE is sent on the connection negotiated using ICE. In steps 17-18, Bob gathers his candidates for RTP, which are returned to Alice in step 19. In step 20, the SIP response is acknowledged. ICE checks for RTP are performed in step 21. In step 22, RTP packets start being exchanged.

III. Related Work

To the best of our knowledge, existing work on P2PSIP performance has focused entirely on NAT-free environments. The performance of OpenDHT as a lookup service for SIP is studied in [11], [12]. In [13], the performance of a distributed SIP system is studied through simulations and theoretical calculations. In [14], we compare the delays of SIP and P2PSIP. All of the work above assumes publicly reachable endpoints and thus, STUN, TURN, and ICE were not used. In [15], we study the performance of ICE-based NAT traversal in P2P environments with Host Identity Protocol (HIP).

IV. Experiments

We carried out experiments in mobile and fixed access networks. The mobile phone models we used were Sony Ericsson W910 and K850. The phones were connected to the Internet through Third Generation (3G) High Speed Downlink Packet Access (HSDPA) connections with 2048 kbit/s downlink and 384 kbit/s uplink bandwidth. The experiments were run in the network of the Finnish operator DNA in conditions in which the phones reported the maximum signal strength. The phones were located in the same access network cell in the Helsinki region. The PCs were Dell Latitude D630 laptops. They were located in the Helsinki region, one in a corporate network and one in a home network connected to the Internet through a 8 Mbit/s Asymmetric Digital Subscriber Line (ADSL) connection. Although the access networks had different bandwidths, the impact on the results should be negligible since bandwidth consumption was low in the experiments.

The P2PSIP experiments were carried out in a 1000-peer overlay consisting of PlanetLab [16] nodes. We studied the delays of calls established between nodes acting as clients in the overlay. Only publicly reachable nodes were allowed to act as peers; nodes behind NATs acted as clients. Both mobile phones and PCs were used as clients. The bootstrap peer of the overlay, the caller, and the callee were located in Helsinki. TURN servers were discovered from the overlay, meaning that they were located in PlanetLab sites. The average uptime of peers was 8 hours. Since the network size was 1000, the resulting mean interarrival and departure time of users is 28.8s based on Little's law. The arrival and departure of peers was modeled as a Poisson process (i.e., the interarrival and departure times follow the exponential distribution). We have studied the impact of lower and higher interarrival times on P2PSIP lookup delays in [14]. The main observation was that, as expected, the lookup delay grows logarithmically as the churn rate increases. The sizes of Chord's finger and successor tables were set to 10 and the size of the predecessor list to 5 peers. The Chord stabilization interval was set to 60s based on the results in [17]. TURN servers were discovered using the algorithm described in Section II-B. During the experiments, we varied the node identifiers of the caller and the callee, meaning that the nodes did not always obtain the same positions on the Chord ring. This was done to eliminate the impact of proximity on the ring on our results.

TABLE I
Traffic model and parameters

Parameter	Value
Peer interarrival time	28.8s
Network size (N)	1000 peers
Busy hour call attemps	2.21 calls/user
% of calls to buddies	66.6
Average size of buddy list	22
Finger pointers	10
Successors	10
Chord stabilization interval	60s
ICE keepalive interval (Tr)	15s
Time between ICE checks (Ta), RTP	20ms
Time between checks (Ta), non-RTP	500ms
Deadline, highest priority pair	2s
Maximum ICE check execution time	10s
Audio codec	G.729

In the SIP experiments, we measured the call setup delays of traditional client/server SIP. The SIP proxy, and TURN servers, caller, and callee were all located in Helsinki. Both mobile phones and PCs were used as SIP clients. In both P2PSIP and SIP experiments, by call setup delay, we refer to the delay between initiating the call and finishing the ICE connectivity checks for RTP.

A. Traffic Model and Parameters

During the measurements, background P2PSIP lookup traffic consisted of VoIP and presence related lookups. We used the same traffic model as in our previous work [14].

As specified in [2], the decision when to stop ICE connectivity checks is a matter of local optimization. In the experiments, we used a stopping criteria consisting of two timeouts: a soft and a hard deadline [15]. The hard deadline is the maximum time that checks will be running. The soft deadline is a time after which the checks are stopped if a path that does not use relays has been found. The hard deadline was set to 10s because of three reasons. First, in person-to-person communication, a user is not likely to tolerate a delay longer than 10s. Second, a 10s delay ensures that even with a relatively long Round Trip Time (RTT) and some packet loss, there is a good chance of finding a path if one exists. Third, this value is also used by other implementations [18]. The soft deadline was set to 2s. This is because the International Telecommunication Union (ITU) recommendation for average post-selection delay on local connections is 3s [19]. A 2s delay ensures that there is still 1s left for signaling before and after ICE (e.g., for performing candidate exchange over SIP).

The audio codec used by the clients was G.729. Audio packets were sent every 20ms. RTP packet size was 32 bytes. The choice of audio codec affects ICE since ICE tries to pace STUN transactions at the same rate as media. For this, ICE uses a timer T_a to control how often nodes can generate STUN or TURN transactions during the connectivity checks and the candidate gathering phase. Based on the default values and formulas in [2], for non-RTP sessions (i.e., SIP and P2PP), T_a was set to 500ms, for RTP T_a was set to 20ms, and STUN keepalives were sent if no packet had been sent on a connec-

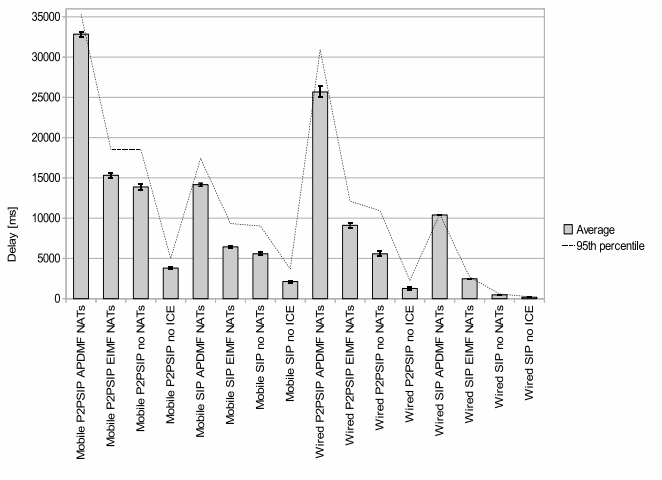

Fig. 3. Call setup delays

tion for 15s. In the experiments, the callee always accepted the call immediately after receiving the SIP INVITE request. The traffic related parameters are summarized in Table I.

We studied four different access network and SIP architecture combinations: mobile P2PSIP, wired P2PSIP, mobile client/server SIP, and wired client/server SIP. For each combination, we studied four different scenarios: (1) both nodes are publicly reachable and do not use ICE (no ICE), (2) both nodes are publicly reachable and use ICE (no NATs), (3) both nodes are behind NATs with Endpoint Independent Mapping and Filtering behavior (EIMF NATs) [20], and (4) both nodes are behind NATs with Address and Port Dependent Mapping and Filtering behavior (APDMF NATs). The difference between the "EIMF NATs" and "APDMF NATs" scenarios is that in the latter, a TURN server needs to relay all the traffic between the nodes. ICE needs to be used even in scenario (2) since a given host does not know a priori whether the remote host is publicly reachable (even a public IP address does not guarantee this). The types of NATs used in six major mobile operator networks are studied in [21] for TCP. Three of the studied operator networks used NATs, however, only one of the NATs supported endpoint independent mapping. Address and port-dependent filtering was used either due to a NAT or a firewall by all operators except one, which did not filter incoming connections at all.

V. Results

Figure 3 shows the average and 95th percentile call setup delays for all the scenarios. For each scenario, 150 calls were set up. The error bars in the figure represent 95% confidence intervals. From the figure, we can observe that regardless of the type of the SIP architecture (P2PSIP or SIP) and the type of the access network (mobile or wired), the lowest call setup delay is achieved when ICE has been disabled and the nodes are publicly reachable, as can be expected. The differences between delays are always significant (statistically at the 95% confidence level) within each of the four access network/SIP architecture combinations. As expected, the highest delays occur when both nodes are located behind APDMF NATs. For P2PSIP clients having a wired connection to the Internet, we can see that when ICE is not used and both nodes are publicly reachable, the average call setup delay is 1.27s. If ICE is enabled but the nodes are still publicly reachable, the delay grows to 5.57s. Thus, enabling ICE increases the delay by a factor of 4.4. If the nodes are located behind EIMF NATs, the delay becomes 9.11s. If the NATs are APDMF NATs, the delay is 25.67s.

The most dramatic increase in call setup delay is experienced in the wired SIP case; when the nodes are publicly reachable and do not use ICE, the average delay is less than 2% of the delay occurring when the nodes are behind APDMF NATs.

It is interesting to compare the average and 95th percentile call setup delays shown in Figure 3 to ITU recommendations. ITU E.721 [19] recommends an average delay of no more than 3.0, 5.0, or 8.0s for local, toll, and international calls, respectively and sets the 95th percentiles at 6.0, 8.0, and 11.0s. If considering all of the calls in our experiments as international calls, we can conclude that in the scenarios involving ICE, the delays of mobile P2PSIP are never acceptable (the lowest average delay is 13.9s and the lowest 95th percentile delay is 18.5s). The delays of wired P2PSIP are only acceptable in the "no NATs" scenario. The delays of mobile and wired SIP are acceptable in scenarios other than "APDMF NATs".

The components of call setup delay are shown in Figures 4-7 for each of the scenarios. In Figure 4, the delays of wired P2PSIP are depicted. In the figure, when ICE is disabled, the call setup delay consists only of P2PP lookup and SIP INVITE transactions. When ICE is enabled, the delay grows because of five new components: candidate gathering for SIP, exchange of candidates in the Connect transaction, ICE checks for SIP, candidate gathering for media, and ICE checks for media. In Figure 4, the INVITE transaction delay is considerably lower for the scenario in which ICE is disabled since in the other scenarios candidate gathering at the called party increases the delay. Also, although their average hop counts are the same (5.2 hops), the delay of the Connect transaction is higher than that of the Lookup transaction because the called party must gather candidates before it can send a reply to the Connect request. We can also observe from Figure 4 that it takes longer to finish the connectivity checks for SIP than for media. As an example, ICE checks for SIP take 6.9 times longer than for media in the "no NATs" scenario. This is explained by the different pacing of connectivity checks that ICE uses for SIP and RTP. In the case of SIP, STUN and TURN transactions are paced at a rate of 500ms. For RTP, they are paced at the same rate as media, that is, every 20ms in our case. This allows the checks to finish earlier for RTP. ICE paces checks at a lower rate for non-RTP streams due to concerns over bandwidth consumption. However, since in our case the checks are paced aggressively for RTP in any case, an easy optimization would be to pace them as aggressively for SIP as well.

From Figures 4-7, we can also see the impact of the soft

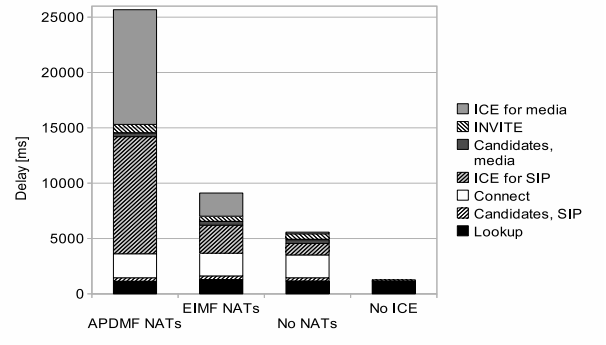

Fig. 4. Components of call setup delay, wired P2PSIP

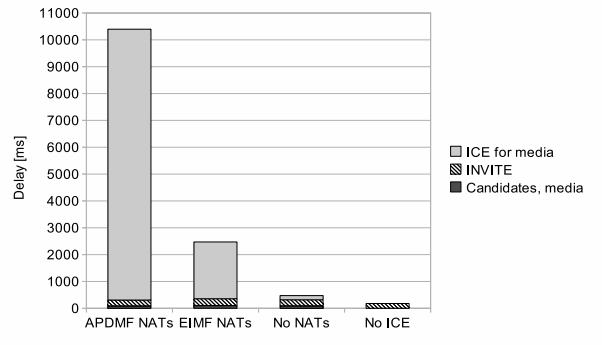

Fig. 5. Components of call setup delay, wired client/server SIP

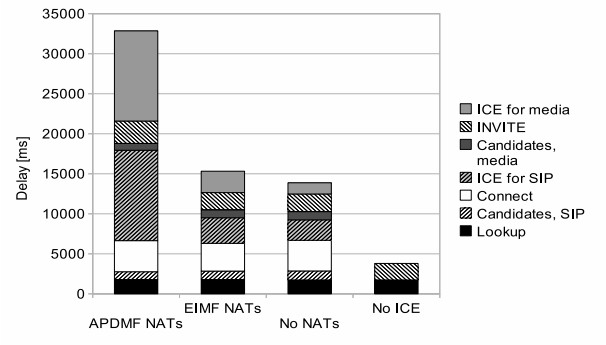

Fig. 6. Components of call setup delay, mobile P2PSIP

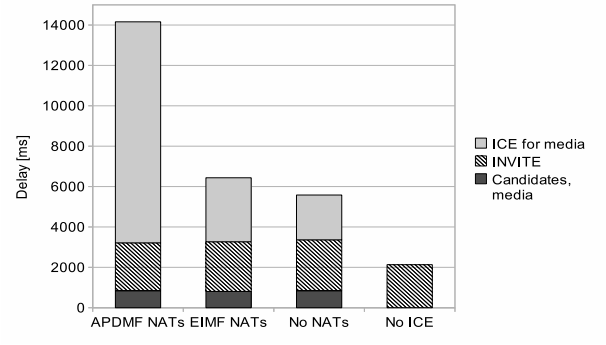

Fig. 7. Components of call setup delay, mobile client/server SIP

and hard ICE deadlines. When both nodes are located behind any type of a NAT, the highest priority pair (i.e., the host candidates) will not work. If the nodes have found a pair that does not use relays when the 2s soft deadline is reached (finding a path with no relays is possible as long as both of the NATs are not APDMF NATs), they stop the checks and select that pair. Thus, in the "EIMF NATs" scenario, the duration of ICE checks is always at least 2s. If the NATs are APDMF NATs, the paths not using a relay will not work. Therefore, the endpoints have to wait until the 10s hard deadline passes before stopping the checks. The result is that the ICE delay is always at least 10s when a relay is needed. As expected, the ICE check delays are clearly the dominant components of the P2PSIP call setup delay when both nodes have APDMF NATs.

Figure 5 depicts the call setup delays for wired client/server SIP. Intuitively, one would except the INVITE transaction delay to be higher than for wired P2PSIP (see Figure 4). This is because when client/server SIP is used, SIP transactions are routed via the proxy server, whereas in the P2PSIP case, they are exchanged directly between the endpoints. The INVITE transaction delay is higher for SIP only in the "no ICE" scenario. This is caused by the location of TURN servers. In client/server SIP, the endpoints use a provisioned TURN server address. Typically, a geographically close TURN server is provisioned. However, in P2PSIP, the TURN server discovery process returns a random TURN server that in our global P2PSIP overlay can be located in any PlanetLab site. The higher RTT with the TURN server is visible in the INVITE transaction and candidate gathering delays of P2PSIP. Also, we can observe that the INVITE transaction delay is especially high in the "APDMF NATs" scenario for P2PSIP. This is because in P2PSIP, the messages of the INVITE transaction are sent on an ICE-negotiated connection, unlike in client/server SIP, in which they are sent via the proxy. If the ICE-negotiated path uses a relay, as it does when two APDMF NATs are involved, there is extra delay since all SIP messages must be sent via the relay. Clearly, P2PSIP would benefit from a service discovery mechanism that locates a geographically close TURN server.

Figure 6 shows the P2PSIP call setup delays between mobile phones. The delays are 3.0, 2.5, 1.7, or 1.3 times higher for mobile nodes in the "no ICE", "no NATs", "EIMF NATs", and "APDMF NATs" scenarios than for wired nodes, respectively. By looking at the components of the call setup delays, we can observe that the higher costs of communicating over the radio interface in the caller's and callee's wireless access networks cause all of the components to be higher. The individual components of the call setup delay are 1.5-9.3 times higher in the mobile case than in the wired case.

Figure 7 shows the call setup delay between two mobile client/server SIP clients. Compared to wired SIP, the call setup delay is 11.8, 11.8, 2.6, or 1.4 times higher for mobile SIP in the "no ICE", "no NATs", "EIMF NATs", and "APDMF NATs" scenarios, respectively. Candidate gathering takes 8.0-9.7 times longer for mobile SIP. The INVITE transaction delay is 9.6-11.8 times higher for the mobile case.

Surprisingly, in Figure 7, the average ICE check delay for media (2.2s) is higher than the 2s soft deadline even in the "no NATs" scenario. In the same scenario for mobile P2PSIP, the delay is 1.4s. The ICE check delay is higher for mobile SIP also in the "EIMF NATs" scenario. No similar difference exists between ICE delays of wired P2PSIP and SIP. By comparing the standard deviations of ICE check delays of mobile P2PSIP and mobile SIP (not shown for brevity), one can see that the standard deviation is much higher for SIP. The SIP and P2PSIP call setup flows are different because for SIP, the first time when data is sent frequently over the radio occurs when ICE checks for media start. In the case of P2PSIP, messages have already been exchanged frequently at that point. In 3G Wideband Code Division Multiple Access (WCDMA), a dedicated channel is allocated when the terminal transmits data [22]. This allows maximum throughput and minimum delay. However, if the terminal does not send data for several seconds, the dedicated channel is released. Because of this, in the mobile SIP scenario, the terminal may not always have a dedicated channel allocated when it starts the ICE checks, which results in higher delays and standard deviation.

It is also interesting to study the connection setup delays. The average delay of setting up a new P2PP or SIP connection consists of candidate gathering, Connect transaction, and ICE check delays. P2PP connections are established when a node joins the overlay or as a result of DHT maintenance operations. For wired P2PSIP, the connection establishment delays are 3.4, 4.9, and 13.1s in the "no NATs", "EIMF NATs", and "APDMF NATs" scenarios, respectively. This represents a considerable increase compared to the "no ICE" scenario, in which the connection setup delay does neither include candidate exchange across the overlay nor connectivity checks.

VI. Conclusions and Future Work

In this paper, we studied the impact of ICE, STUN, and TURN on call setup delays of P2PSIP and client/server SIP. As expected, the presence of NATs has a considerable impact on these delays. ITU E.721 [19] recommends an average delay of no more than 8.0s for international calls and sets the 95th percentile at 11.0s. If considering the calls in our experiments as international calls, we can conclude that in the scenarios involving ICE, the delays of mobile P2PSIP are never acceptable (the lowest average delay is 13.9s and the lowest 95th percentile delay is 18.5s). The delays of wired P2PSIP are only acceptable when no NATs are involved. The delays of mobile and wired SIP are acceptable in all other scenarios except those in which relays are used.

Based on our results, some optimizations can be suggested for P2PSIP implementations and deployments. An important factor increasing P2PSIP call setup delay is that ICE is run twice. Eliminating one of these negotiations can reduce the delays significantly especially when relays are needed. In [23], we propose how a Host Identity Protocol based architecture can be applied to P2PSIP to achieve this goal. We plan to evaluate the performance of this architecture in future work. Another optimization is to pace ICE connectivity checks as aggressively for SIP as they are paced for RTP. P2PSIP would also benefit from a service discovery mechanism locating a TURN server that is geographically close to the user. We plan to study how to reduce ICE check delays in future work.

References

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261, IETF, 2002.

[2] J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols," IETF, Internet Draft – work in progress, February 2010.

[3] J. Rosenberg, R. Mahy, and P. Matthews, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)," IETF, Internet Draft – work in progress, Feb 2010.

[4] J. Rosenberg, R. Mahy, P. Matthews, and D. Wing, "Session Traversal Utilities for NAT (STUN)," RFC 5389, IETF, 2009.

[5] S. Baset, H. Schulzrinne, and M. Matuszewski, "Peer-to-Peer Protocol (P2PP)," IETF, Internet Draft – work in progress, November 2007.

[6] C. Jennings, B. Lowekamp, E. Rescorla, S. Baset, and H. Schulzrinne, "REsource LOcation And Discovery (RELOAD) Base Protocol," IETF, Internet Draft – work in progress, February 2010.

[7] S. Guha and P. Francis, "Characterization and measurement of TCP traversal through NATs and firewalls," in *IMC '05: Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*. Berkeley, CA, USA: USENIX Association, 2005.

[8] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for internet applications," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 17–32, 2003.

[9] J. Mäenpää and G. Camarillo, "Estimating Operating Conditions in a Session Initiation Protocol Overlay Network," in *Proc. of IEEE IPDPS*, Atlanta, USA, April 2010.

[10] M. Handley, V. Jacobson, and C. Perkins, "SDP: Session Description Protocol," RFC 4566, IETF, 2006.

[11] B. Meyer and M. Portmann, "Practical Performance Evaluation of Peer-to-Peer Internet Telephony Using SIP," in *Proc. of IEEE CITWORKSHOPS*, Washington, DC, USA, 2008, pp. 204–209.

[12] M. Matuszewski and E. Kokkonen, "Mobile P2PSIP: Peer-to-Peer SIP Communication in Mobile Communities," in *Proc. of IEEE CCNC*, Las Vegas, Nevada, USA, 2008, pp. 1159–1165.

[13] E. Harjula, J. Ala-Kurikka, D. Howie, and M. Ylianttila, "Analysis of Peer-to-Peer SIP in a Distributed Mobile Middleware System," in *Proc. of IEEE GLOBECOM*, San Francisco, CA, USA, 2006.

[14] J. Mäenpää and G. Camarillo, "Analysis of Delays in a Peer-to-Peer Session Initiation Protocol Overlay Network," in *Proc. of IEEE CCNC*, Las Vegas, USA, January 2010.

[15] A. Keränen, "Host Identity Protocol-based Network Address Translator Traversal in Peer-to-Peer Environments," Master's thesis, Helsinki University of Technology, Sep 2008.

[16] L. Peterson, A. Bavier, M. E. Fiuczynski, and S. Muir, "Experiences building PlanetLab," in *Proc. of the 7th symposium on Operating systems design and implementation (OSDI '06)*. Berkeley, CA, USA: USENIX Association, 2006, pp. 351–366.

[17] J. Mäenpää and G. Camarillo, "Study on Maintenance Operations in a Peer-to-Peer Session Initiation Protocol Overlay Network," in *Proc. of IEEE IPDPS*, 2009.

[18] Microsoft, "Interactive Connectivity Establishment (ICE) Extensions," Microsoft Corporation, Technical specification v20100218, Feb 2010.

[19] ITU-T, "Network Grade of Service Parameters and Target Values for Circuit-Switched Services in the Evolving ISDN," International Telecommunication Union, Geneva, Recommendation E.721, May 1999.

[20] F. Audet and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP," RFC 4787, IETF, Jan 2007.

[21] L. Mäkinen and J. K. Nurminen, "Measurements on the Feasibility of TCP NAT Traversal in Cellular Networks," in *Next Generation Internet Networks*. IEEE, April 2008, pp. 261–267.

[22] H. Haverinen, J. Siren, and P. Eronen, "Energy Consumption of Always-On Applications in WCDMA Networks." in *VTC Spring*. IEEE, 2007.

[23] A. Keränen, G. Camarillo, and J. Mäenpää, "Host Identity Protocol-Based Overlay Networking Environment (HIP BONE) Instance Specification for REsource LOcation And Discovery (RELOAD)," IETF, Internet Draft – work in progress, January 2010.