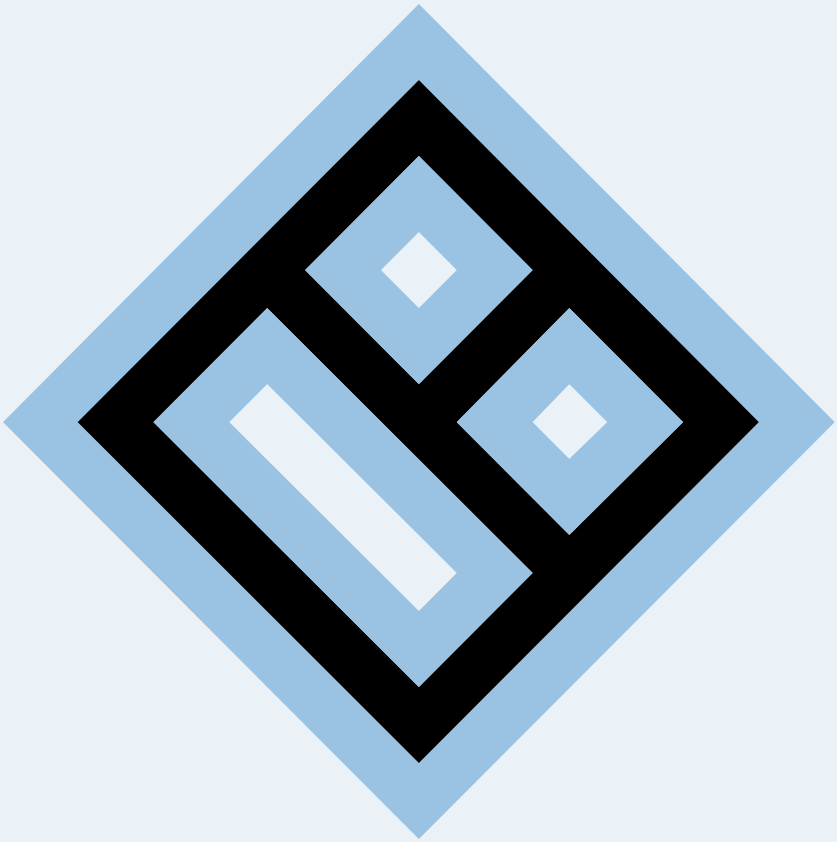
NEXTRAGEN

NEXT GENERATION TESTING

Network Address Translation (NAT) behindert die VoIP-Telefonie



Management Summary

Durch die begrenzte Verfügbarkeit von öffentlichen IP-Adressen werden heute in der Regel die privaten IP-Netze (Unternehmensnetze) mit den öffentlichen IP-Netzen (Provider-Netzen) mit Hilfe der Network Address Translation-Techniken verbunden. Diese arbeiten transparent und beeinflussen die zu übermittelnden Datenströme nicht. Will man jedoch mit Voice over IP (VoIP) über diese NAT-Komponenten kommunizieren, müssen die Endgeräte die hierfür notwendigen Zusatzfunktionen verstehen und entsprechend konfiguriert sein. Auch bei der Messung der Sprachgüte über die gesamte Übermittlungsstrecke zwischen dem Anrufer und dem angerufenen Telefondienstteilnehmer müssen die NAT-Funktionen ordnungsgemäß unterstützt werden. Funktioniert das sogenannte STUN-Verfahren nicht, funktioniert zwar die Signalisierung, aber die eigentlichen Sprachströme zwischen den Endgeräten bleiben im virtuellen Raum hängen.

Beim Network Address Translation (NAT) werden die IP-Adressen eines privaten Netzes mit Übersetzungstabellen den öffentlich registrierten IP-Adressen zugeordnet. Dabei bleiben die internen IP-Adressen jedoch völlig verborgen. Damit verändert NAT aktiv die Pakete zwischen dem internen privaten Netz und dem äußeren öffentlichen IP-Netz. Bei diesem Prozess werden durch die NAT-Dienste im IP-Header die Sender- und die Empfänger-IP-Adressen ausgetauscht.

Die einfachste Form der Adressumwandlung wird als statisches NAT bezeichnet. Bei der Adressübersetzung wird eine private IP-Adresse beim Übermitteln vom privaten in den öffentlichen Adressraum in eine öffentliche IP-Adresse umgewandelt. Beim Antwortpaket erfolgt die Wandlung in umgekehrter Reihenfolge.

Die Adressübersetzung erfolgt über feste Tabellen (Von-Zu-Tabellen) die vom Netzadministrator per Konfiguration angelegt werden. Empfängt die NAT-Komponente ein Datenpaket vom inneren Netz werden folgende Funktionen erbracht:

- Empfang eines Datenpakets vom privaten IP-Netz
- Überprüfung der privaten IP-Adresse anhand der Von-Tabelle.
- Selektion der öffentlichen IP-Adresse anhand der Zu-Tabelle
- Austausch der Adressen im IP-Header
- Neuberechnung der TTL-Werte und Prüfsummen im IP-Header
- Generieren des neuen IP-Headers
- Übermittlung des Datenpakets auf das öffentliche IP-Netz

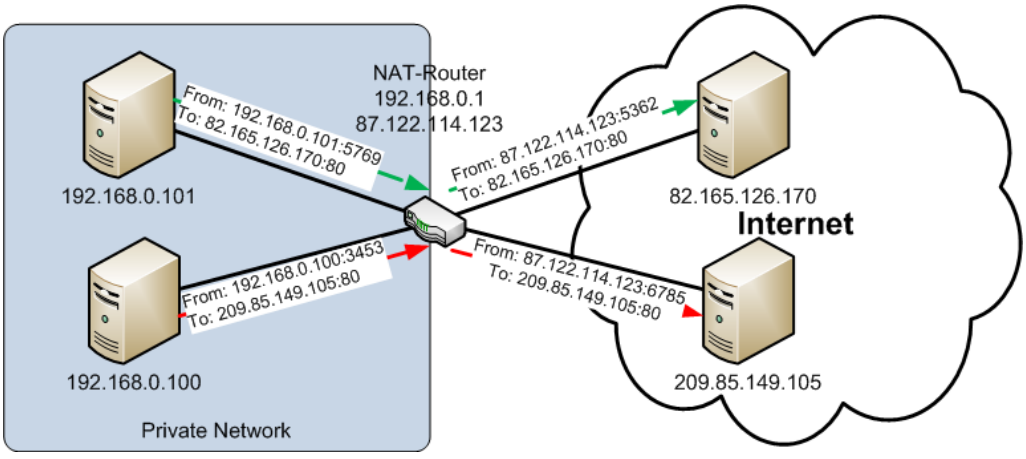


Abbildung 1: *Prinzip des Network Address Translations*

Diese Übersetzung der IP-Pakete wird bei jedem IP-Paket individuell vorgenommen und es wird nicht registriert, ob das IP-Paket zu einem IP-Datenstrom gehört. Aus diesem Grund spricht man auch von einer zustandslosen Adressübersetzung. Bei dieser Form der Adressumwandlung werden keine zusätzlichen Sicherheitsmaßnahmen getroffen. Es wird lediglich der IP-Header mit den privaten Adressen an die Bedingungen der öffentlichen IP-Adressen angepasst.

Diese Art der Adressumwandlung wird auch als 1:1-Adresswandlung bezeichnet. Da jeder privaten IP-Adresse eine öffentliche IP-Adresse gegenübersteht, werden keine Adressen eingespart.

Port Address Translation (PAT)

Der Port and Address Translation (PAT) Mechanismus bildet alle IP-Adressen eines privaten Netzes auf eine einzige öffentliche IP-Adresse ab. Auf diese Weise benötigt ein komplettes privates Netz nur eine einzige registrierte öffentliche IP-Adresse. Einige Hersteller bezeichnen die PAT-Funktion auch als verborgenes NAT.

Teilen sich in der Praxis zwei interne Rechner auf der Basis der privaten IP-Adressen eine externe IP-Adresse, kommt es unweigerlich zu einem Adresskonflikt. Kommunizieren beide interne Rechner gleichzeitig mit externen Kommunikationspartnern, muss die NAT-Komponente die Entscheidung treffen, an welchen internen Rechner das empfangene Paket weiterzuleiten ist. Da die Routing- bzw. Weiterleitungsentscheidung nur anhand der im IP-Header integrierten IP-Adressen erfolgt, ist dieses Problem nicht zu lösen.

Wie beim Adressmapping auf der Schicht 3 muss die NAT-Komponente nur beim Verbindungsaufbau eine entsprechende Mapping-Tabelle anlegen und ist somit in der Lage, die einzelnen Verbindungen den richtigen IP-Adressen zuzuordnen. Der NAT-Prozess durchsucht einfach die Mapping-Tabelle nach der Verbindung, zu der das betreffende Paket gehört. Bei einer Übereinstimmung wird die Adresse gewandelt und an den betreffenden IP-Rechner im internen Netz weitergeleitet. Soweit die Theorie. In der Praxis stellt sich dieser Prozess jedoch viel komplizierter dar. Beispielsweise kommunizieren zwei interne Rechner mit einer gemeinsamen externen IP-Adresse und übermitteln beide eine DNS-Anfrage zum DNS-Server, der vom ISP für das betreffende Unternehmen oder Privatperson betrieben wird. Der vom ISP betriebene DNS-Server befindet sich aus der Sicht der DNS-Clients im externen Netz. Damit durchlaufen sämtliche DNS-Anfragen immer den NAT-Prozess und es erfolgt immer eine Adresswandlung. Somit besteht die Adresstabelle aus folgenden Informationen:

Externes Netz	**internes Netz**
Externe IP-Adresse	
Sender-TCP/UDP-Port von DNS-Server für Session 1	interne IP-Adresse 1 (DNS-Client 1) und Empfänger
	Sender-TCP/UDP-Port von DNS-Client 1
Sender-TCP/UDP-Port von DNS-Server für Session 2	interne IP-Adresse 2 (DNS-Client 2)
	Sender-TCP/UDP-Port von DNS-Client 2

Die DNS-Clients übermitteln an den DNS-Server im öffentlichen Netz ihre DNS-Requests. Damit sind in den auf das öffentliche IP-Netz übermittelten Paketen folgende IP/TCP/UDP-Informationen enthalten: die gleiche IP-Source-Adresse, die gleiche IP-Destination-Adresse und die gleiche Destination-Port-Nummer (UDP-Port 53 für DNS-Anfragen). Nur durch die Source-Port-Nummern unterscheiden sich die DNS-Anfragen. Genau diese Informationen werden zur Identifizierung der internen Verbindungen herangezogen.

Die meisten Betriebssysteme starten die Zuweisung der Absenderports mit dem Wert 1025 und weisen die Source-Portnummern fortlaufend den einzelnen Verbindungen zu. Unter Umständen können die beiden IP-Sender die gleichen Source-Port-Nummern für die Kommunikation mit dem DNS-Server benutzen. In diesem Fall ist ein Konflikt unvermeidbar. Um auch diese statistische Möglichkeit einer vollkommenen Adressgleichung zu vermeiden, wandelt der PAT-Prozess nicht nur die IP-Adressen, sondern auch die Portnummern um. Damit wird sichergestellt, dass die internen IP-Komponenten immer mit einer individuellen Portnummer mit den externen IP-Ressourcen kommunizieren.

Durch die zusätzliche Wandlung der Source-Port-Adresse erhöht sich auch die Sicherheit der Gesamtlösung. Ein Hacker kann weder über die externe IP-Adresse noch über die Source-Port-Adresse auf die interne IP-Ressource schließen. Auch für die Sicherheit vom inneren Netz zum äußeren Netz lässt sich der PAT-Mechanismus einsetzen. Mit PAT lässt sich über einen internen Adresspool der Zugriff auf die externe IP-Adresse einschränken.

NAT/PAT erfordert Zusatzfunktionen in VoIP-Komponenten

Das Session Initiation Protocol (SIP) ist heute das Signalisierungsprotokoll der Wahl für den Aufbau von Voice over IP (VoIP) Telefonielösungen. Bei dem Übergang über Router/Firewalls mit integriertem Network Address Translation hat die SIP-basierte Kommunikation ihre liebe Mühe. Allzu oft scheitert der VoIP-Anruf oder bestimmte Funktionen des Unified Communications (UC) an der NAT-Realität.

Bei den VoIP-Protokollen werden auf Basis des SIP-Protokolls beim Verbindungsaufbau die Endpunktadressen im SIP Header – genauer gesagt im Header des Session Discription Protocols (SDP) - mitgeteilt. In einem solchen Fall handelt es sich um die IP- und Port-Adresse des die Verbindung aufbauenden Telefons. Durch die von NAT hervorgerufenen Adressänderungen auf der Ebene 4 stimmt natürlich die im SDP-Header verzeichnete Adresse nicht mehr und der RTP-Datenstrom kommt von der Gegenseite nicht mehr beim Absender an. Aus diesem Grund muss vom Endpunkt bereits während der Signalisierungsphase dem Empfänger die öffentliche Adresse im SDP-Header mitgeteilt werden. Zur Ermittlung der öffentlichen Adresse wird das Session Traversal Utilities for NAT (STUN) genutzt.

Der Mechanismus „Session Traversal Utilities for NAT“ (STUN) bezeichnet eine technische Möglichkeit zur transparenten Weiterleitung von VoIP-Strömen über NAT-Systeme hinweg. STUN sorgt für die korrekte Bestimmung der öffentlichen IP-Adressen der beteiligten VoIP-Endpunkte. Darüber hinaus bietet STUN einen Mechanismus zur Überprüfung der Verbindungen zwischen zwei Endpunkten und sorgt auf Basis eines Keep-Alive-Protokolls für die notwendigen Pflegemechanismen zur Aufrechterhaltung der NAT-Adresszuordnungen.

Die erste Version von STUN wurde im RFC 3489 beschrieben. Diese Version wird inzwischen als "klassisches STUN" bezeichnet. Die in der Praxis gemachten Erfahrungen erforderten eine völlige Überarbeitung des STUN-Konzepts. Das neue STUN (gemäß RFC 5389) stellt jetzt nur noch eine Teillösung in Zusammenarbeit mit anderen Translation-Mechanismen (beispielsweise SIP-OUTBOUND, TURN und ICE) dar.

Die Aufgabe eines „Standalone STUN Servers“ besteht in der Bereitstellung der korrekten Transportadressen anhand der STUN Binding-Funktion. Hierzu muss der STUN-Server die notwendigen Messages über das UDP-Protokoll austauschen.

Der STUN Standard legt fest:

- welche STUN-Methoden verwendet werden,
- welche Authentifizierungs- und Integritätsmechanismen genutzt werden,
- welche Mechanismen zur Unterscheidung der STUN-Nachrichten von anderen Messages notwendig sind.
- wie ein STUN-Client die IP-Adresse und den Port des STUN-Servers ermittelt.
- ob eine Rückwärtskompatibilität zu RFC 3489 erforderlich ist.
- welche optionalen Attribute (FINGERPRINT und ALTERNATE-SERVER) oder Erweiterungen erforderlich sind.

STUN bietet für VoIP-Endpunkte einen Mechanismus zur korrekten Bestimmung der IP-Adresse und des aktuell genutzten Ports am anderen Ende eines NAT-Gateways/Routers (Übergang zwischen privaten und öffentlichem IP-Adressbereich). Im Gegensatz zum klassischen TUN lassen sich mit dem neuen STUN auch optionale Attribute und eine Authentifizierung mit VoIP-Servern aushandeln.

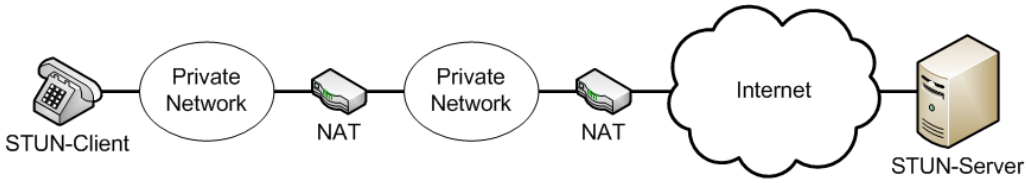


Abbildung 2: STUN-Konfiguration

In voran gegangener Abbildung gibt es zwei STUN-Agenten. Das VoIP-Telefon arbeitet in diesem Fall als Client und befindet sich im privaten Netzwerk 1. Dieses Netz ist mit dem privaten Netzwerk 2 über NAT 1 verbunden. Das private Netzwerk 2 verfügt über eine Verbindung zum öffentlichen Internet über NAT-2. Der eigentliche STUN-Server befindet sich im öffentlichen Internet.

STUN ist ein Client-Server-Protokoll. Es unterstützt zwei Arten von Transaktionen:

- Mit Hilfe der Request/Response-Transaktion werden zwischen STUN Client und dem STUN Server die Anfrage und Antworten übermittelt.
- Die Informationstransaktion wird von beiden Agents (Client oder Server) generiert und wird vom Empfänger nicht beantwortet.

Beide Transaktionsarten enthalten eine individuelle Transaktions-ID, die aus einer zufällig gewählten 96-Bit-Zahl besteht. Die Transaktions-ID dient der eindeutigen Zuordnung von Anfragen zu Antworten.

Alle STUN-Nachrichten basieren auf einem festen Header. Dieser definiert die spezifische Methode, die Klasse und die Transaktions-ID. Als Methode ist momentan nur die Binding-Methode definiert. Die Klasse zeigt an, ob es sich um eine Anfrage, ein positive Response, eine Fehlermeldung oder einen Hinweis handelt. Auf den festen STUN Header folgen ein oder mehrere Attribute und signalisieren zusätzliche Informationen.

Folgende STUN Methoden sind momentan festegelegt:

Wert	**Bedeutung**
000	Reserved
001	Binding
002	Reserved

Ein Binding erfolgt entweder auf eine Request/Response-Transaktionen oder in Indikator-Transaktionen. Wird das Binding in einer Request/Response-Transaktion genutzt, wird damit festgestellt, welche Adresszuordnung auf dem NAT-Gateway für den betreffenden STUN Client angelegt wurden. Mit Hilfe der Indikator-Transaktionen werden aktive Adressbindungen auf dem NAT-Gateway am Leben erhalten.

Mit einer Request/Response-Transaktion wird ein Binding-Request von einem STUN-Client an einen STUN-Server übermittelt. Empfängt ein STUN-Server einen Binding-Request, kann dieser bereits auf dem Weg zwischen STUN-Client und der STUN-Server über ein oder mehrere NAT Gateways übermittelt worden sein. Daher muss beim Transport durch ein NAT-Gateway der Binding-Request die darin enthaltenen Absenderdressen (die Source-IP-Adresse und den Source-Port) anpassen. Dies hat zur Folge, dass die Source-Adresse im Binding-Request beim Empfang auf dem STUN-Server die öffentliche IP-Adresse und die zugehörige Port-Adresse enthält. Diese Adresse wird auch als „reflexive Transportadresse“ bezeichnet.

Der STUN-Server kopiert die reflexive Transportadresse in das XOR-mapped-Adress-Attribut der Binding-Response und sendet diese als Binding-Response an den STUN-Client zurück. Der Binding-Response durchläuft auf dem Weg zwischen STUN Server und Client ebenfalls die auf dem Weg liegenden NAT-Gateways. Dabei wird jedoch nur die Adresse im IP-Header umgeschrieben. Die im XOR-mapped- Adress-Attribut enthaltenen öffentlichen IP-Adressen bleiben unverändert und der STUN Client erfährt dadurch seine öffentlichen Adressen.

STUN definiert eine Reihe optionaler Mechanismen. Hierzu gehören beispielsweise die Ermittlung von DNS-Adressen, eine Umleitungstechnik zu alternativen Servern, das Fingerabdruck-Attribut und zwei Authentifizierungs- und Integritätsfunktionen. Mit Hilfe der Authentifizierungsmechanismen werden die Benutzernamen, Kennworte und Nachrichten-Integritätswerte übermittelt. Hierzu wurden folgende Authentifizierungsmechanismen festgelegt:

- Ein langfristiger Credential-Mechanismus und
- Ein temporärer Credential-Mechanismus.

Beim langfristigen Credential-Mechanismus teilen sich Client und Server voreingestellte Benutzernamen und Passworte und tauschen die Geheimnisse mit Hilfe eines eine Challenge/ Response-Mechanismus aus. Der temporäre Credential-Mechanismus dient dem Austausch vor dem eigentlichen STUN-Prozess zwischen Client und Server, um die entsprechenden Benutzernamen und Passworte mit Hilfe eines Out-of-Band-Mechanismus auszutauschen. Dadurch werden beispielsweise bei ICE die Integrität der Anfragen und Antworten geschützt.

Übermittlung von STUN Messages über UDP

Da bei der Übermittlung von STUN Messages über UDP einzelne Nachrichten verloren gehen können, wird durch eine Sendewiederholung die Zuverlässigkeit von STUN Request/Response-Transaktionen erhöht. Die Sendewiederholung wird durch den STUN-Client initiiert. Verloren gegangene STUN Indicator-Transaktionen werden nicht erneut gesendet.

Ein Client leitet die Sendewiederholung für einen verloren gegangenen STUN-Request nach dem Retransmission Timeout (RTO) Intervall ein. Die RTO basiert gemäß RFC 2988 auf der Round-Trip-Zeit (RTT). Es gelten jedoch zwei Ausnahmen:

- Der Standardwert für RTO sollte konfigurierbar sein und größer als 500 ms sein.
- der RTO-Wert darf nicht auf die nächste Sekunde aufgerundet werden. Es wird eine Genauigkeit von 1 ms angestrebt.

Der RTO-Wert sollte vom Client nach Abschluss einer Transaktion zwischengespeichert werden. Dieser Wert dient anschließend als RTO-Startwert für die nächste Transaktion mit dem gleichen STUN-Server.

Nach dem Aussenden eines STUN Requests werden vom Client bis zu sieben Sendewiederholungen übermittelt. Erhält der Client nach der letzten Anfrage keine Antwort gilt die Transaktion als gescheitert. Die Pausendauer zwischen Sendewiederholungen bestimmt der Parameter „Rm“. RM arbeitet mit einem Standardwert = 16. Eine gescheiterte STUN-Transaktion wird durch eine ICMP-Fehlermeldung abgeschlossen.

Wird beispielsweise ein RTO von 500 ms angenommen, dann wird der erste Request direkt (bei 0 ms) ausgeschickt. Die Sendewiderholungen erfolgen bei 500 ms, 1500 ms, 3500 ms, 7500 ms, 15500 ms und 31500 ms. Erhält der Client nach 39500 ms keine Antwort wird sich die Transaktion mit einem Timeout abgebrochen.

Empfangen von STUN Nachrichten

Empfängt ein STUN-Agent eine STUN-Nachricht wird diese zuerst auf Regelkonformität überprüft. Dabei wird kontrolliert, ob die ersten zwei Bits der Message auf den Wert = 00 gesetzt sind, das Magic Cookie einen gültigen Wert aufweist, die Nachrichtenlänge sinnvoll ist und der Methodenwert auf eine unterstützte Methode verweist. Anschließend wird überprüft, ob die Nachrichtenklasse im spezifischen Kontext zulässig ist. Handelt es sich bei der Klasse um einen positive Reaktion auf einen ausgeschickten Request oder eine Fehlerreaktion, überprüft der Agent die Transaktions-ID gegenüber der zugehörigen Transaktion. Wird die Fingerprint Extension genutzt, prüft der Agent, ob das Fingerprint-Attribut mit dem zugehörigen richtigen Wert in der Message vorhanden ist. Bei Fehlern in der Nachricht wird die Message stillschweigend verworfen.

Anschließend werden vom STUN-Agent die erforderlichen Sicherheitsüberprüfungen gemäß dem spezifizierten Authentifizierungsmechanismus ausgeführt.

Direkt nach der erfolgreichen Authentifizierung untersucht der STUN-Agent die Nachricht auf unbekannte oder unerwartete Attribute. Unbekannte und unerwartete Attribute werden vom Agenten ignoriert. Die Verarbeitung von unbekannten Kontext-Attributen hängt von der jeweiligen Nachrichtenklasse des STUN Requests ab.

Enthält ein STUN Request eine oder mehrere unbekannte Kontext-Attribute antwortet der Server mit einer Fehlermeldung und dem Fehlercode = 420 (Unbekanntes Attributes) und kopiert das UNKNOWN-Attributes Attribut in die Antwort. Der STUN-Server unternimmt anschließend alle weiteren Prüfungen, die von der spezifizierten Methode oder der spezifischen Nutzung gefordert wird. Nach erfolgreichem Abschluss aller Prüfungen generiert der STUN-Server eine Erfolgsmeldung im jeweiligen STUN Response.

Empfängt ein STUN-Server über UDP einen Request, kann es sich um den ersten Request einer Transaktion oder eine Sendewiederholung handeln. Der STUN-Server muss auf Sendewiederholung wie folgt reagieren: Wenn der Client die Antwort auf die Sendewiederholung erhält (und somit nicht die Antwort auf die ursprüngliche Anfrage) befinden sich STUN-Client und STUN-Server im identischen Stadium. Dieser Kommunikationszustand entspricht exakt dem Fall, wenn die Antwort auf den Original-Request empfangen wird oder wenn sowohl die Antwort auf den ursprünglichen STUN Request und die Antwort auf die Sendewiederholung eingegangen ist.

STUN Requests

STUN Requests werden vom STUN-Client unter zwei Bedingungen erzeugt:

- die erste Anfrage vom STUN-Client zum STUN-Server (Identifiziert anhand der IP-Adresse und dem zugehörigen Port).
- Weitere vom STUN-Client initiierte Requests folgen nach Abschluss einer früher vollständig abgeschlossenen Transaktion.

Ein STUN Request als Reaktion auf 401 oder 438 Fehlermeldungen gelten nicht als „nachfolgende Requests“.

Erster Request

Hat der STUN-Client noch keine erfolgreiche Request/Response-Transaktion mit dem STUN Server abgeschlossen, dürfen im ersten STUN Request nicht die USERNAME-, MESSAGE-INTEGRITY-, REALM- und NONCE- Attribute genutzt werden. Dadurch wirkt der erste Request als bestünde zwischen STUN-Client und STUN-Server keine Authentifizierung oder würden die Nachrichtenintegritätsfunktionen genutzt.

Nachfolgender Request

Sobald eine Request/Response-Transaktion erfolgreich abgeschlossen wurde, verfügt der STUN-Client über die notwendige Informationen (Realm und Nonce) des STUN-Servers, und den für die Authentifizierung notwendigen Benutzernamen und Passwort. Der STUN-Client legt für die weitere Kommunikation mit dem STUN-Server den Benutzernamen, das Passwort, das Realm und die Nonce im Cache ab.

Generiert der STUN-Client eine weitere Anfrage, füllt dieser das Benutzernamen-, das Realm- und Nonce-Attribut mit den zwischengespeicherten Werten. Darüber hinaus enthält der STUN Request das MESSAGE-INTEGRITY Attribut entsprechend dem zwischengespeicherten Passwort.

Empfang eines STUN Requests

Nachdem der STUN-Server die grundlegende Prozesse des STUN Requests abgearbeitet hat, werden folgende Funktionen ausgeführt:

- Enthält die Message kein MESSAGE-INTEGRITY-Attribut, muss der STUN-Server eine Fehlermeldung mit dem Fehlercode 401 (Unauthorized) verschicken. Die STUN Response enthält in diesem Fall einen konkreten Realm-Wert. Der Standard empfiehlt, dass als Wert der dem Domain-Namen des Providers des betreffenden STUN-Servers entspricht. Die Antwort muss eine vom STUN-Server ausgewählte Nonce enthalten. Die STUN Response darf nicht das USERNAME- oder das MESSAGE-INTEGRITY Attribut enthalten.
- Enthält die Message das MESSAGE-INTEGRITY-Attribut, aber fehlen das USERNAME-, REALM- oder NONCE Attribut, muss der STUN-Server eine Fehlermeldung mit dem Fehlercode 400 (Bad Request) verschicken. Die Fehlermeldung darf nicht das USERNAME-, NONCE-, REALM oder MESSAGE-INTEGRITY Attribut enthalten.
- Ist die Nonce nicht mehr gültig, muss der STUN-Server mit einer Fehlermeldung und dem Fehlercode 438 (Stale Nonce) reagieren. Die Fehlermeldung muss nur das NONCE- und das REALM-Attribute enthalten.
- Ist der Benutzernamen im USERNAME-Attribut ungültig, muss der STUN-Server einer Fehlermeldung und dem Fehlercode 401 (Unauthorized) reagieren. Die Fehlermeldung enthält den entsprechenden Realm-Wert. Der Standard empfiehlt, dass als Wert der dem Domain-Namen des Providers des betreffenden STUN-Servers entspricht. Die Antwort muss eine vom STUN-Server ausgewählte Nonce enthalten. Die Response darf nicht das USERNAME- oder das MESSAGE-INTEGRITY Attribut enthalten.
- Das Passwort in Verbindung mit dem USERNAME-Attribut dient der Berechnung des Integritätswerts der jeweiligen Message. Entspricht der berechnete Wert nicht dem Wert des MESSAGE-INTEGRITY-Attributs, verwirft der STUN-Server den Request und erzeugt eine Fehlermeldung mit dem Fehlercode 401 (Unauthorized). In der Fehlermeldung dürfen nur das Realm- und das Nonce Attribute enthalten sein.

Nach Abschluss der Kontrollmechanismen bearbeitet der STUN-Server den betreffenden STUN Request weiter und generiert im jeweiligen Kontext die entsprechende Antwort.

Empfang einer Response

Empfängt der STUN-Client eine Fehlermeldung mit dem Fehlercode 401 (Unauthorized), muss dieser die Request/Response-Transaktion erneut starten. Der STUN Request muss einen USERNAME (entsprechend dem vom STUN-Client für den Realm der Fehlermeldung festgelegten Benutzernamen) enthalten. Der STUN Request enthält in diesem Fall den Realm der Fehlermeldung. Darüber hinaus enthält der STUN Request die Nonce der Fehlermeldung, das MESSAGE-INTEGRITY–Attribut (der Wert besteht aus dem Passwort in Verbindung mit dem Benutzernamen).

Enthält die Fehlermeldung den Fehlercode 438 (Stale Nonce), muss der STUN-Client bei der Sendewiederholung des STUN Request die neue Nonce nutzen.

Der STUN-Client durchsucht nach dem Empfang einer STUN Response diese auf das Vorhandensein des MESSAGE-INTEGRITY-Attributs ab. Wird dies gefunden, so überprüft der STUN-Client die Integrität der Nachricht. Entspricht der resultierende Wert dem Wert des MESSAGE-INTEGRITY-Attributs, gilt der Response als authentifiziert. Besteht keine Übereinstimmung oder enthält die STUN Response nicht das MESSAGE-INTEGRITY-Attribut wird die Response stillschweigend verworfen.

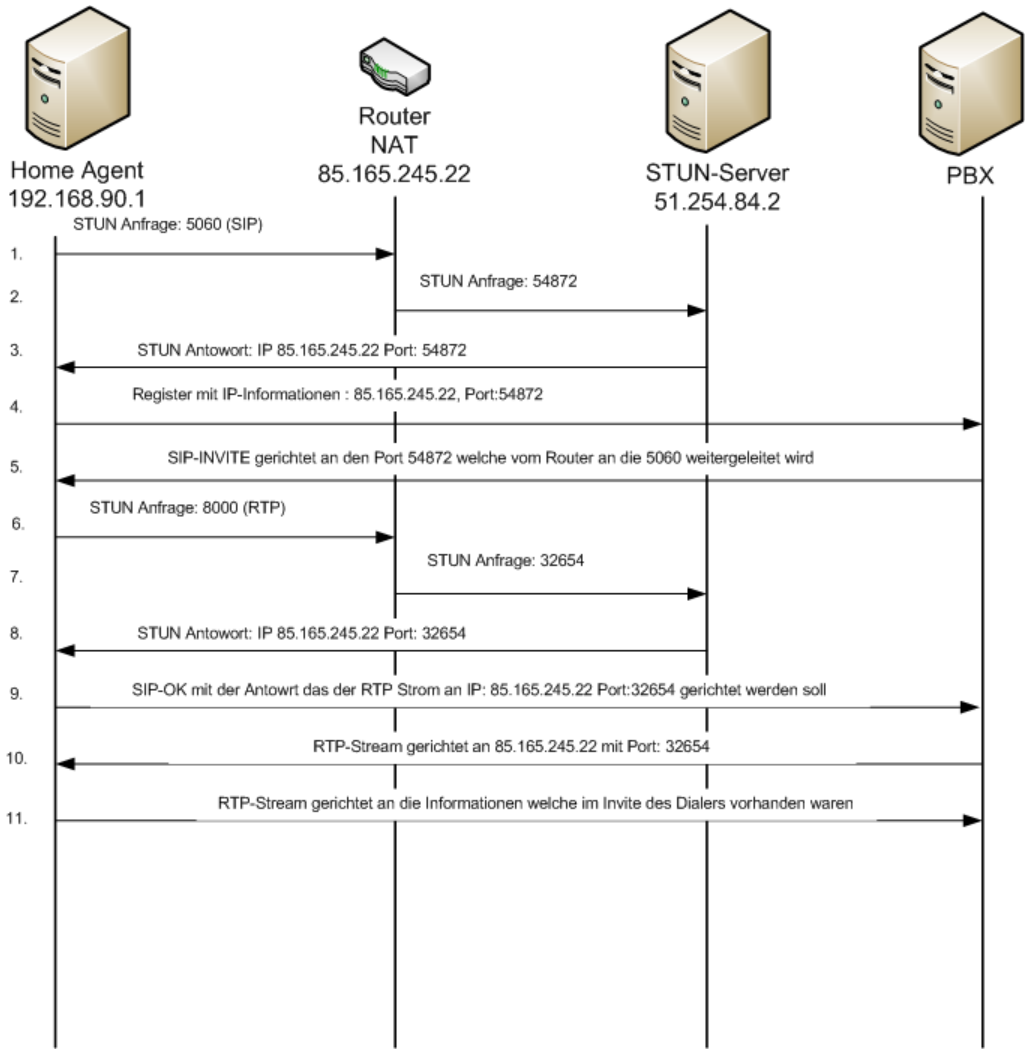


Abbildung 3: Funktionsablauf STUN

Die Abbildung 3 veranschaulicht den VoIP-Kommunikationsablauf über einen NAT-fähigen Router.

1. Die auf dem Homeagent installierte Softphone-Applikation (VoIP-Telefon) sendet vor der eigentlichen SIP-Signalisierung eine STUN-Anfrage an den STUN-Server. Im Detail passiert dabei folgendes:

a. Der VoIP-Client verschickt seine Anfrage über den SIP-Port 5060 an das Gateway bzw. den Router.

b. Der Router empfängt das Paket und beginnt mit der NAT-Prozedur. Dabei werden interne die IP-Adresse (Schicht 3) und der UDP Port (Schicht 4) auf die öffentliche Adresse des Routers geändert. Die Adresse entspricht der öffentlichen IP-Adresse und als Port-Adresse wird ein zufälliger Port gewählt.

2. Der Router verschickt die STUN Anfrage an den STUN-Server. Der STUN-Server erkennt öffentliche IP/Port-Adresse des Routers. Anschlie-

ßend schickt der STUN-Server die empfangene öffentliche IP/Port-Adresse an den Router zurück.

3. Der Router leitet das über Port 54872 empfangene Paket direkt an die IP-Adresse und die Port-Adresse 5060 des Homeagents weiter. Damit ist der Port 54872 auf dem Router für den Homeagent freigeschaltet und alle vom öffentlichen IP-Netz empfangenen Pakete werden direkt an Port 5060 des Homeagents weitergeleitet

4. Nach dem Empfang des STUN-Pakets kennt das Softphone des Homeagents seine öffentliche IP-Adresse (85.165.245.22) und den zugehörigen Port (54872) und verfügt über genügend Informationen um das Gespräch zu initiieren.

5. Die anschließend generierte SIP-Invite Message wird an den Dialer übermittelt IM SDP-Header wird dabei die folgende Information übermittelt: IP-Adresse: 85.165.245.22; Port-Adresse: 54872

6. Die SIP-Invite Message an den Telefonieserver zum Aufbau des Anrufs übermitteln.

7. Die Gegenstelle empfängt die SIP-Invite Message und beantwortet die SIP-Anfrage mit der SIP-Meldung 200 ok im SDP-Header signalisiert der Telefonserver seine IP-Adresse und seine Portnummer zum Aufbau der nachfolgenden RTP-Verbindung.

8. Nach dem Empfang der SIP-Meldung 200 ok durch das Softphone übermittelt dieses erneut eine STUN-Anfrage. In diesem Fall wird der Port 8000 abgefragt, da dieser der Eingangsport für den ankommenden RTP-Strom ist.

9. Der NAT-Router setzt die Adresse in eine öffentliche IP-Adresse um und leitet diese STUN-Anfrage an den STUN-Server weiter.

10. STUN antwortet mit der IP Adresse (85.165.245.22) und dem zugeordneten Port (32654)

11. Danach erfolgt das endgültige SIP-OK vom Softphone an den Telefonserver und die öffentliche IP-Adresse (Schicht 3) und der öffentliche Port (Schicht 4) wird im SDP-Header für den ankommenden RTP-Stream signalisiert.

12. Anschließend baut der Telefonservers zum Softphone über die in der SDP-Message signalisierte Adresse die zur Übermittlung der Sprachdaten notwendige RTP-Verbindung auf.

13. Ebenso verfährt das Softphone. Es baut eine Peer-to-Peer-Verbindung auf Basis des RTP-Protokolls zum Telefonserver auf.

14. Damit bestehen zwischen den beiden VoIP-Endgeräten zwei halbduplex Verbindungen über die die Sprachströme übermittelt werden.

Einsatz von TraceSim in einer NAT-Umgebung

Das von Nextragen entwickelte Messwerkzeug TraceSim wird zur Simulation von VoIP-Verbindungen genutzt. Damit diese auch über NAT Router zustande kommen können, muss in einer solchen NAT-Umgebung selbstverständlich die Konfigurationen der Simulationssoftware an die Netzbedingungen angepasst werden..

TraceSim wurde für den Einsatz in den unterschiedlichsten VoIP-Umgebungen optimiert und unterstütz daher alle notwendigen Mechanismen für den ordnungsgemäßen NAT-Betrieb in NAT-Umgebungen. Für die VoIP-Messungen in einer NAT-Umgebungeneiner NAT-Umgebung kommen dabei folgende Messmethoden zum Einsatz:

- Automatische Ermittlung der öffentlichen IP-Adresse mittels STUN oder
- Manuelles Festlegen der anzuwendenden öffentlichen IP-Adresse durch die entsprechende Konfiguration in TraceSim

Die öffentliche IP-Adresse wird benötigt, um diese im SDP-Header der entsprechenden SIP-Nachrichten korrekt zum Empfänger der SIP-Nachricht zu übermitteln. Bei der automatischen Ermittlung der öffentlichen IP-Adresse durch STUN muss hierzu ein STUN- Server konfiguriert werden. Dieser STUN- Server, welcher im Internet platziert ist, wird beim Start einer SIP-Verbindung kontaktiert und anschließend festgestellt, welche öffentliche IP-Adresse aktuell für die Übermittlung von SIP-Messages genutzt werden soll. Bei der Benutzung der STUN-Funktionalitäten, werden die IP-Adressen automatisch ermittelt und eine manuelle Konfiguration bleibt aus.

Die zu nutzende öffentliche IP-Adresse kann auch vom Anwender durch eine manuelle Eingabe in der TraceSim-Software konfiguriert werden. Dadurch bestimmt der Anwender selbst, welche öffentliche IP-Adresse im SDP-Header der SIP Message übermittelt wird. Dies hat den großen Vorteil, dass bestimmte Konfigurationen von IP-Adresse getestet werden können. Dies hat den Vorteil, dass der Anwender genau festlegen kann, welche IP Adresse benutzt wird. Gleichzeitig kann er feststellen, und dadurch feststellen kann ob das Netzwerk funktioniert und keine Fehler am STUN-Server auftreten.

Firmenprofil Nextragen

Die Nextragen GmbH ist auf die Entwicklung von VoIP/Video-Produkten und Software-Lösungen rund um die Themen „Monitoring, Analysing und Testing“ ausgerichtet. Die Sicherstellung der End2End Dienstequalität (QoS, QoE) für Next Generation Networks und Triple Play Dienste steht im Fokus des im Jahr 2009 gegründeten Unternehmens mit Sitz an der Flensburger Förde im Norden Deutschlands. Kunden aus dem Carrier-, Telekommunikations- und Enterprise-Segment setzen die Nextragen-Lösungen ein, um die Qualität und Zuverlässigkeit von VoIP- und Video-Anwendungen zu monitoren, zu analysieren und zu testen. Produkte, Lösungen und Dienstleistungen der Nextragen GmbH sind 100% „made in Germany“ und werden weltweit über zertifizierte Partner vertrieben.

Weitere Informationen erhalten Sie auf der Firmenwebsite unter www.nextragen.de.

NEXTRAGEN
NEXT GENERATION TESTING

Messkom Vertriebs GmbH
Awarenring 38
D-85419 Mauern

Tel: 0049 (0)8764 / 948 430
Fax: 0049 (0)8764 / 948 433

Email: info@messkom.de
www.messkom.de