Robert Fehrmann
Proseminar Technische Informatik
Institut für Informatik, Betreuer: Matthias Wählisch

You are Skyping - But How Does it Work!?

Freie Universität Berlin

You are Skyping - But How Does it Work!?

- Probleme im Internet
- Voice over IP mittels Skype
- Network Address Translator
- Firewalls
- Protokolle zur Überwindung von NATs und Firewalls
- Verbindungsaufbau von Skype-Endgeräten hinter NATs
- Bewertung & Ausblick
- Quellen

Probleme im Internet

- Internet wächst -> mehr IP-Adressen werden benötigt
  - Derzeit: IP-Protokoll in Version 4 (IPv4) mit 32-Bit-Adressen
    - 4.294.967.296
  - langfristige Lösung: Einführung von IPv6 mit 128-Bit-Adressen
    - Vergrößerung um Faktor 2^96
  - „Kurzfirstige" Lösung: Network Adress Translator (NAT)

➔ Adressierungsproblem von Computern hinter NATs

- Schutz vor Angriffen
  - Firewall einrichten
    - Zwischen zwei Netzwerken
    - Prüft Datenverkehr

➔ Datenverkehr zwischen Kommunikationspartnern ungewollt geblockt

Fazit: Ende-zu-Ende-Kommunikation erschwert

Ziel von Skype: transparenter Verbindungsaufbau zwischen Endgeräten

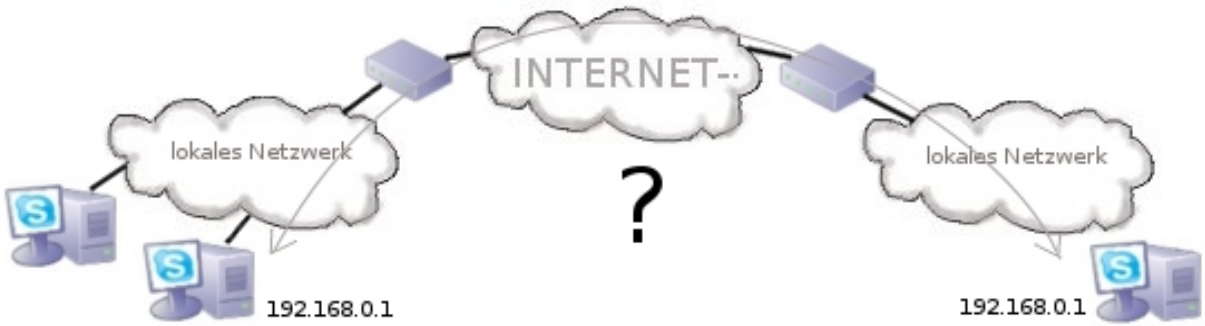
Voice over IP mittels Skype

- Übertragung von Sprache in IP-basierten Netzwerken
- Proprietäres Protokoll für Audio-, Video- und Text-Chat
- Overlay-Peer-to-Peer-Netzwerk
  - Einfache Knoten, Super-Knoten
  - Dezentrales Netzwerk, mit zentralem Login-Server
- Besonderheit:
  - Problemlose Überwindung von NATs und Firewalls
  - Hoher Verbreitungsgrad
  - Kostenlos für Privatanwender
  - Günstige Gespräche ins Ausland
  - Ortsunabhängig

Network Address Translator

- Vergabe von öffentlichen IP-Adressen minimieren
  - NATs bekommen öffentliche IP-Adressen
  - Endgeräte hinter NATs bekommen private IP-Adressen

- Lokale IP-Adressen auf öffentliche IP-Adressen des NATs abbilden

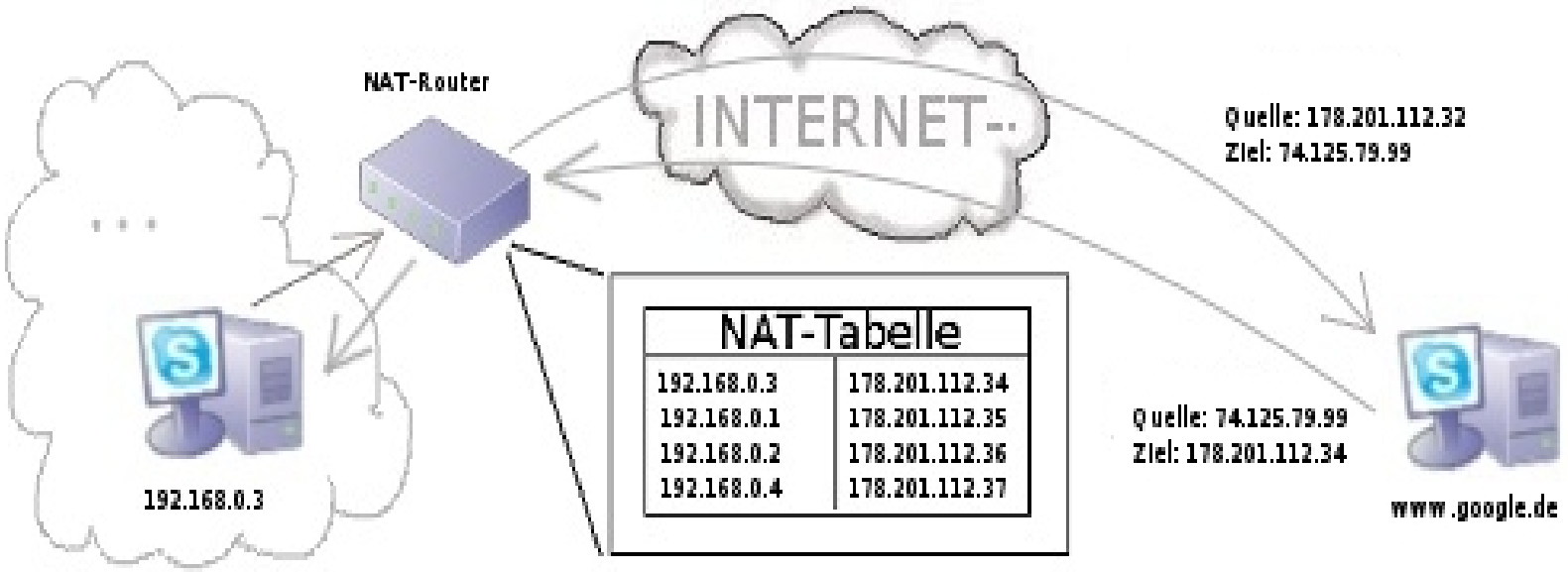


- Es existieren verschiedene Abbildungsstrategien
  - Full Cone NAT
    - Statisches NAT
  - Restricted Cone NAT
  - Post Restricted Cone NAT } dynamische NATs
  - Symmetric NAT

Dynamischer NAT

- Anzahl der IP-Adressen im privaten Netzwerk < verfügbaren IP-Adressen des NATs

- Beispiel:



- Masquerading ist eine besondere Form des dynamischen NATs
  - Es wird zusätzlich die Portnummer der Quelladresse mit Portnummer von NAT getauscht

Probleme mit NAT

- Zur Überwindung von NATs wird Hilfe eines Dritten benötigt
  - Muss IP-Adressen und Ports von Kommunikationspartnern kennen
  - Muss im Internet erreichbar sein
- Alle Verbindung verlaufen über NAT
  - Großer Overhead
  - Ende-zu-Ende-Paradigma wird verletzt
- NAT-Tabelle
  - Eingehende Verbindung bei Masquerading unmöglich, wenn Eintrag in NAT-Tabelle nicht vorhanden
  - Alte Einträge der Tabelle müssen gelöscht bzw. verdrängt werden
  - Neue Einträge müssen Platz in der Tabelle finden
  - Einträge zu aktiven Verbindungen dürfen nicht gelöscht werden

Firewalls

- Überwacht ein- bzw. ausgehende Verbindungen
- Verhindert Paketweiterleitung nach festen Regeln
  - Paketfilter
  - Stateful Inspection
  - Application Layer Firewall
  - Inhaltsfilter
  - Sperrt TCP/UDP-Ports
    - z.B. erlaubt nur Web-Zugriff (TCP Port 80)
- Probleme:
  - Kommunikation auf „beliebigen" Ports erschwert
  - „Verstecken" der Applikationsdaten

STUN

- Simple Traversal of User Datagram Protocol (UDP) Through NATs
- Protokoll zur Erkennung und Überwindung von NATs & Firewalls
- STUN-Server im Internet wird benötigt
- Kommunikation zwischen STUN-Client und STUN-Server
  - Binding Request
  - Shared Secret Request
- Antworten des Servers enthalten IP-Adresse und Port des Clients
  - Client vergleicht seine IP-Adresse und Port mit Informationen aus Antwort
  - Rückschlüsse auf NAT-Art möglich
- Nachteil: kann keine symmetrischen NATs überwinden

TURN

- Traversal using Relay NAT
- Überwindung von symmetrischen NATs fokussiert
- Sehr ähnlich zu STUN
  - Server im Internet wird benötigt
  - Anfragen von Clients und Antworten von Servern gleich aufgebaut
- Unterschiede:
  - Erweiterung der Datentypen, die verschickt werden können, bspw. RTT
  - Symmetrische NATs werden überwunden
  - TCP-Verbindungen sind möglich

Verbindungsaufbau

Viel Theorie bisher, aber wie funktioniert Skype nun ?

- Start der Software
- Anmelden beim Login-Server
  - Verbindungen erst über UDP, bei Fehlschlag über TCP/IP auf Port 80,
- Während Login-Prozesse Ermittlung von NATs/Firewalls
- Keine NATs/Firewalls vorhanden -> direkte Verbindung zwischen Endgeräten
- NATs/Firewalls vorhanden -> Umgehung mittels eines dedizierten Skype-Servers

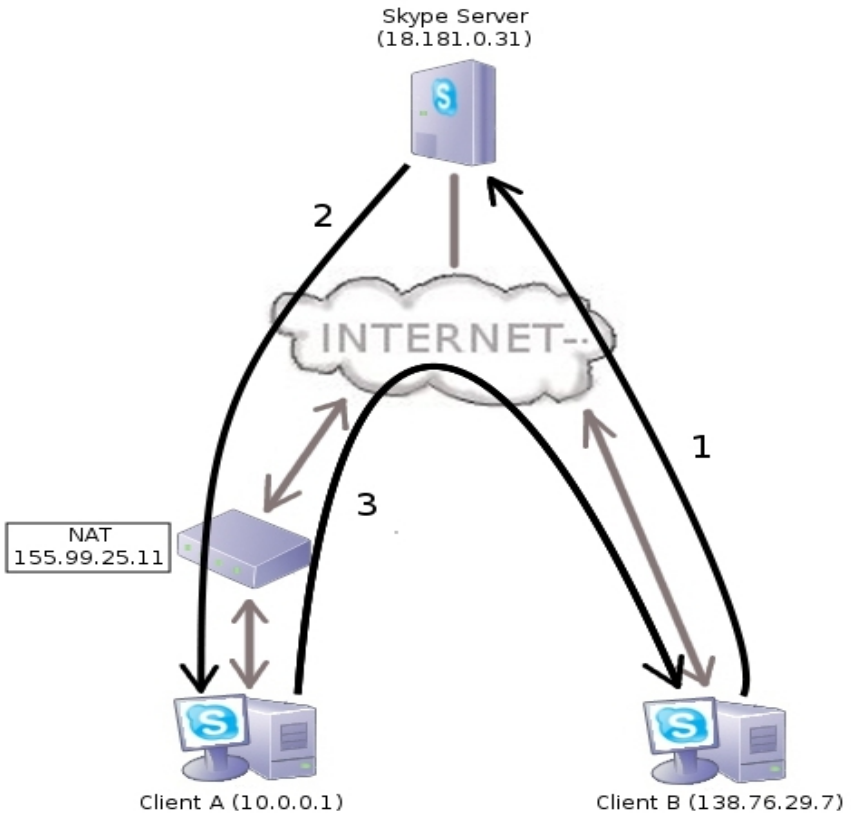
Verbindungsaufbau 1

- Skype Client A hinter NAT
- Skype Client B im öffentlichen Netzwerk

- Direkte Verbindung nicht möglich
  - B kennt A´s IP-Adresse nicht

Lösung:

- Beide Clients melden sich bei einem Super-Knoten an
- B hinterlässt eine Kommunikations-anfrage mit IP-Adresse
- A nutzt die hinterlassene IP-Adresse um Verbindung aufzubauen



Verbindungsaufbau 2

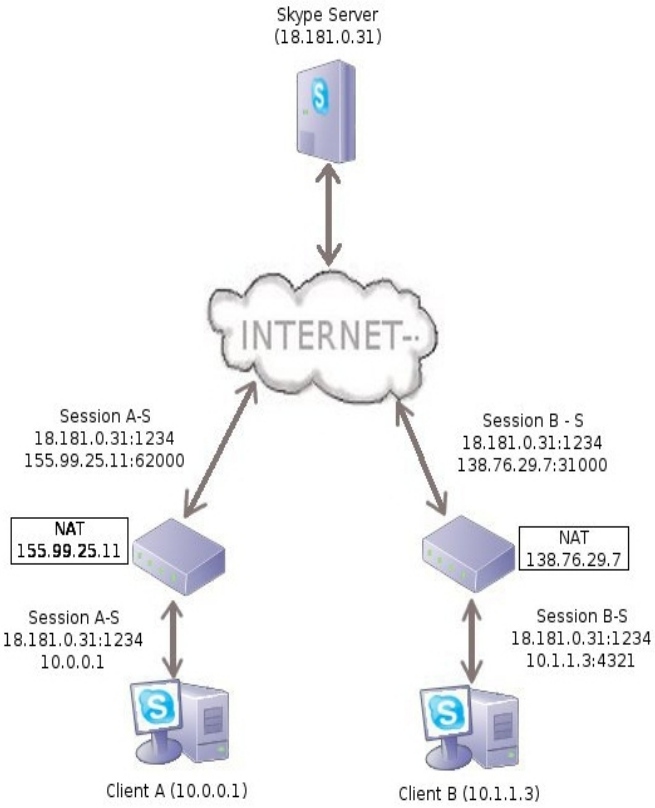
- Beide Clients hinter verschiedenen NATs

Lösung:

- Relaying
  - TURN stellt eine Implementierung von Relaying
- Beide Clients richten Verbindung zum selben Super-Knoten ein
- Client A hinterlässt Kommunikationsanfrage, Client B akzeptiert
- Alle Daten werden über Server geleitet

Nachteil:

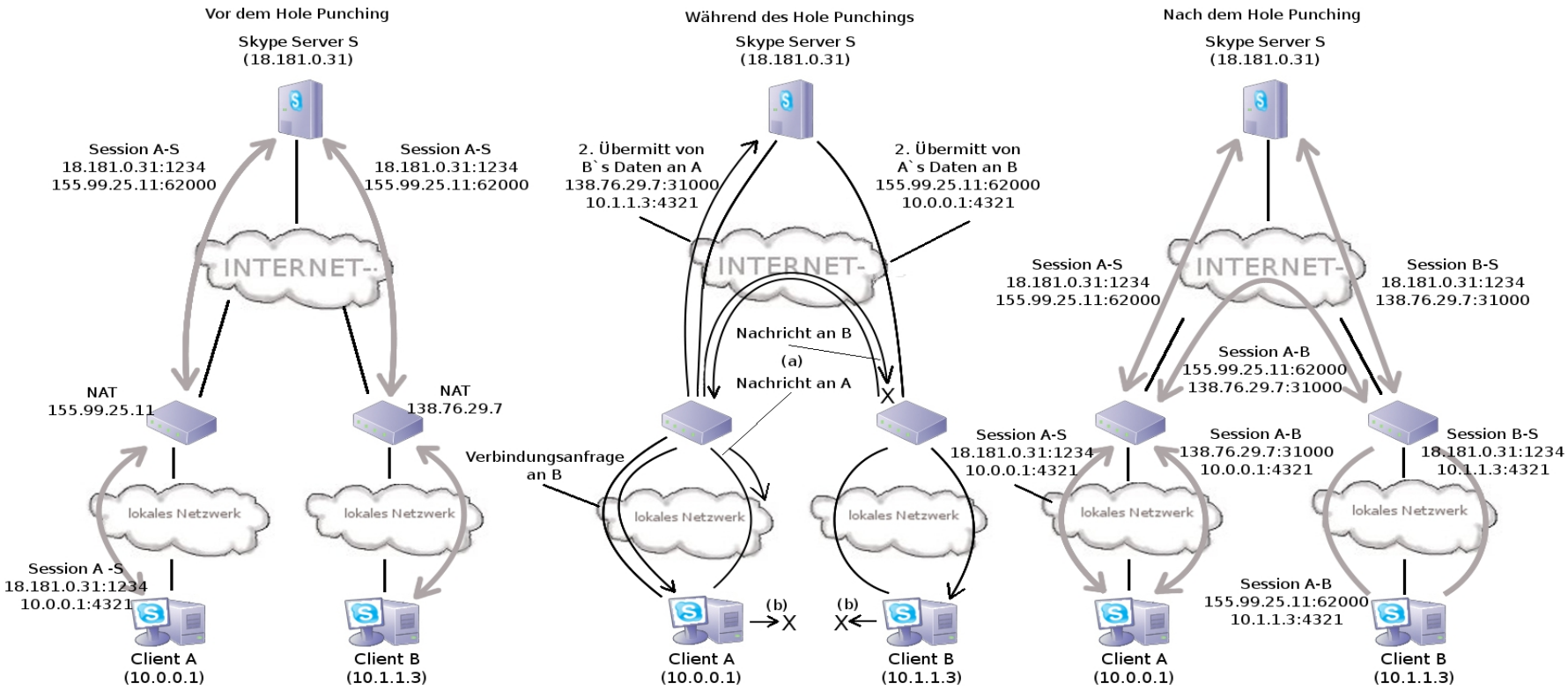
- Sehr ineffizient



Verbindungsaufbau 3

- Beide Clients hinter verschiedenen NATs
- Clients kennen gegenseitige IP-Adressen und Ports nicht
- UDP Hole Punching
  - Beide Clients melden sich beim Server an
  - Hinterlegen IP-Adressen und Ports
  - Ein Client sendet an öffentliche IP-Adresse und Port von Client B
  - Eingehende Verbindungen zu Client A von B ausgehend nun möglich
  - Client B sendet Nachricht an öffentliche IP-Adresse und Port von A
  - Eingehende Verbindungen zu Client B von A ausgehend nun möglich
  - „Löcher" entstanden
  - Direkte Verbindung möglich

Verbindungsaufbau 3



Verbindungen durch Firewalls

- Sind UDP Verbindungen gesperrt, kann TCP verwendet werden
  - Skype lauscht auf UDP und TCP-Ports
  - Auch Port 80 (HTTP) und Port 443 (HTTPS)
  - Für TCP existiert ebenfalls Hole Punching
- Vorteile von Firewalls
  - Schutz vor Schadsoftware
  - Kontrolle von Datenverkehr
- Nachteile von Firewalls
  - Blocken nach festen Parametern
  - Benötigt viele Ressourcen des Rechners
  - Gewollte Kommunikation erschwert

Bewertung & Ausblick

- Wachstum des Internets brachten Lösungen der IP-Adressknappheit
- Network Address Translator finden Verwendung
  - Aufweichung des Ende-zu-Ende-Paradigmas -> Kommunikationsprobleme im Internet
- Skype schafft diese Probleme mit Hilfe einer besonderen Architektur und STUN/TURN zu lösen
- Exakte Bewertung schwer aufgrund proprietären Protokoll
- Gegenüberstellung peer-to-peer-basierten Skype und Client/Server-basierter Konkurenzsoftware
- Sicherheitsaspekte von Peer-to-Peer-Netzwerken

Danke!

Quellen

- http://www.skype.de.
- S. A. Baset and H. G. Schulzrinne. An analysis of the skype peer-to-peer internet telephony protocol. In INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, pages 1–11, 2006.
- Sören Brunk. Internet-telefonie und instant messaging: Sip und skype. Seminararbeit, August 2007.
- C. Jennings F. Audet, Ed. Network address translation (nat) behavioral requirements for unicast udp. In RFC 4787. January 2007.
- C. Huitema J. Rosenberg, R. Mahy. Traversal using relay nat (turn). In Internet-Draft. September 2005.
- C. Huitema R. Mahy J. Rosenberg, J. Weinberger. Stun - simple traversal of user datagram protocol (udp) through network address translators (nats). In RFC 3489. March 2003.
- B. Ford D. Kegel P. Srisuresh, Kazeon Systems. State of peer-to-peer (p2p) communication across network address translators (nats). In RFC 5128. March 2008.
- M. Holdrege P. Srisuresh. Ip network address translator (nat) terminology and considerations.
In RFC 2663. August 1999.