

Aufbau eines virtuellen privaten Netzes mit Peer-to-Peer-Technologie

Wolfgang Ginolas

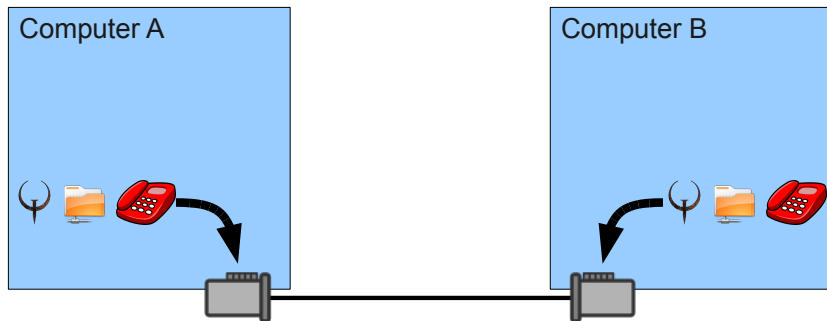
Fachhochschule Wedel

14. Oktober 2009

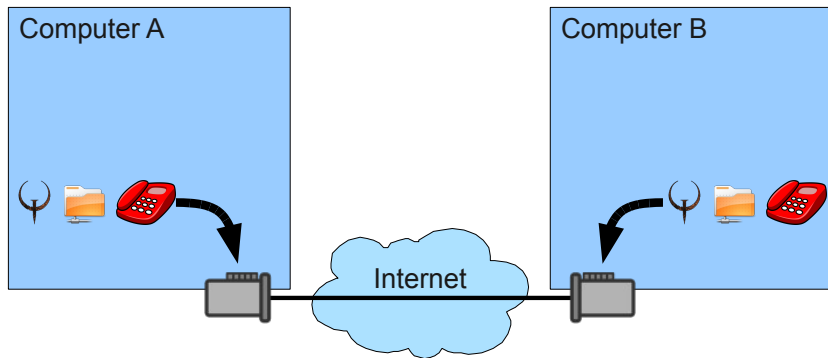
Inhalt

- 1 Einleitung
- 2 Probleme
- 3 Entwurf
- 4 Implementierung
- 5 Demo
- 6 Ergebnis

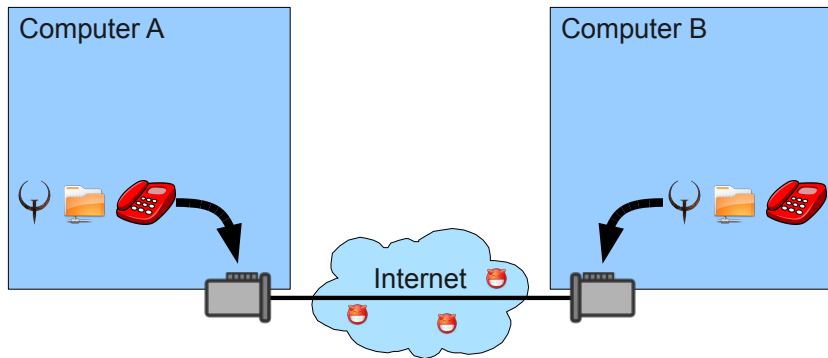
VPN - Aufbau



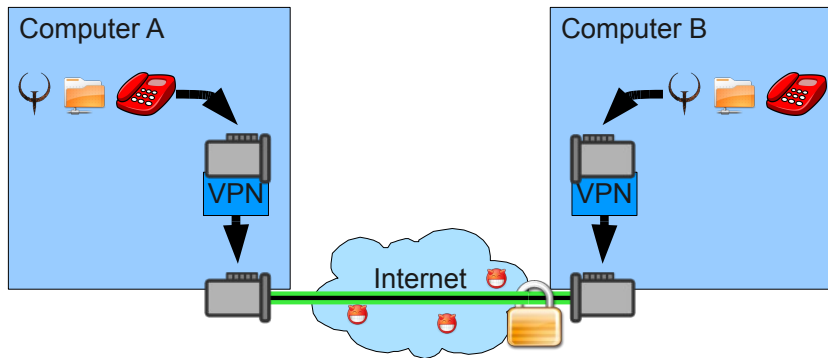
VPN - Aufbau



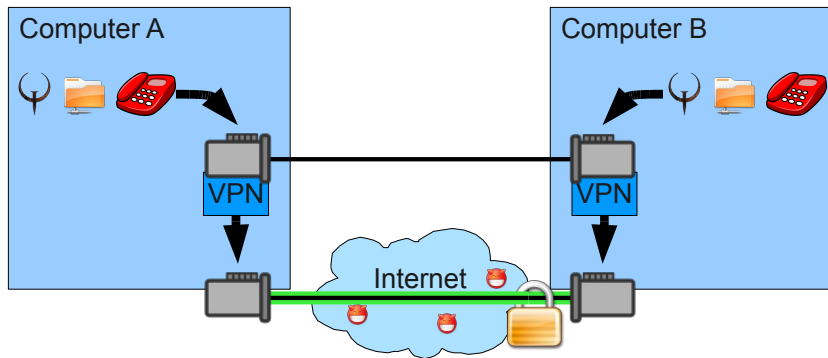
VPN - Aufbau



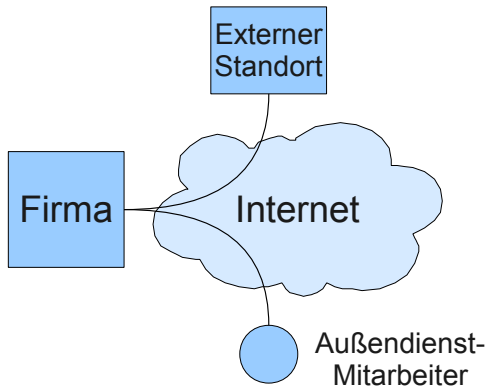
VPN - Aufbau



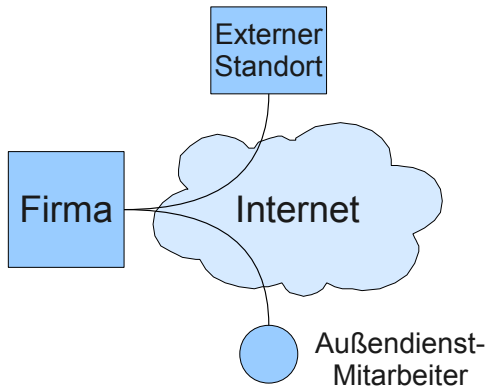
VPN - Aufbau



VPN - Firma



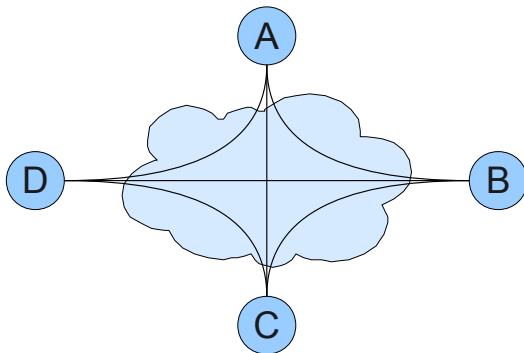
VPN - Firma



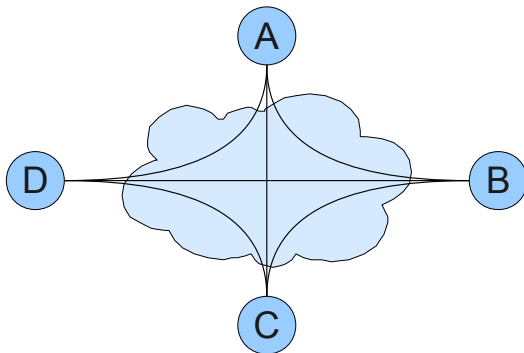
- Server

- Administrator

VPN - Privatpersonen



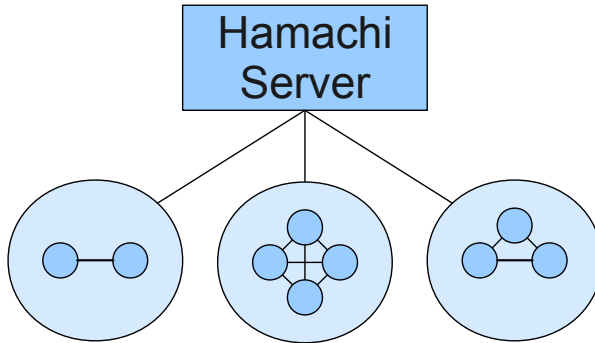
VPN - Privatpersonen



- Kein Server

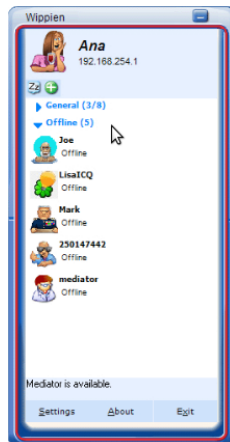
- Kein Administrator

Hamachi



Wippen

- Instant Messaging mit VPN
- Kein zentraler Server vom Hersteller
- Teilweise Open Source
 - Kernkomponenten sind Closed Source



Tinc

- Kein zentraler Server
- Open Source
- Nicht Benutzerfreundlich
 - Aufwendige Konfiguration
 - Keine grafische Oberfläche

The screenshot shows a terminal window titled 'wolfgang@kiwi: ~'. The window displays the manual page for 'tincd(8)'. The menu bar includes 'Datei', 'Bearbeiten', 'Ansicht', 'Terminal', 'Reiter', and 'Hilfe'. The title bar of the terminal window reads 'TINCD(8) BSD System Manager's Manual TINCD(8)'. The content of the manual page is as follows:

```

NAME
    tincd - tinc VPN daemon

SYNOPSIS
    tincd [-c CONF] [--config=DIR] [--no-detach] [--debug=LEVEL]
          [--kill=SIGNAL] [--net=NETNAME] [--generate-keys=BITS]
          [--nlock] [--logfile=FILE] [--pidfile=FILE] [--bypass-security]
          [--help] [--version]

DESCRIPTION
    This is the daemon of tinc, a secure virtual private network (VPN)
    project. When started, tincd will read it's configuration file to deter-
    mine what virtual subnets it has to serve and to what other tinc daemons
    it should connect. It will connect to the ethertap or tun/tap device and
    set up a socket for incoming connections. Optionally a script will be
    executed to further configure the virtual device. If that succeeds, it
    will detach from the controlling terminal and continue in the background,
    accepting and setting up connections to other tinc daemons that are part
    of the virtual private network. Under Windows (not Cygwin) tinc will
    install itself as a service, which will be restarted automatically after
    reboots.
  
```

At the bottom of the terminal window, it says 'Manual page tincd(8) line 1'.

Ziele dieser Arbeit

- Benutzerfreundlichkeit
- Sicherheit

Ziele dieser Arbeit

- Benutzerfreundlichkeit
 - Konfiguration einfach und verständlich
 - Server aufsetzen ist nicht nötig
 - Grafische Oberfläche
- Sicherheit

Ziele dieser Arbeit

- Benutzerfreundlichkeit
 - Konfiguration einfach und verständlich
 - Server aufsetzen ist nicht nötig
 - Grafische Oberfläche
- Sicherheit
 - Zugang nur für autorisierte Personen
 - Vertrauen darf nicht erzwungen werden
 - In Implementierung
 - In fremden Server

Ziele dieser Arbeit

- Benutzerfreundlichkeit
 - Konfiguration einfach und verständlich
 - Server aufsetzen ist nicht nötig
 - Grafische Oberfläche
- Sicherheit
 - Zugang nur für autorisierte Personen
 - Vertrauen darf nicht erzwungen werden
 - In Implementierung
 - In fremden Server
- Also:

Ziele dieser Arbeit

- Benutzerfreundlichkeit
 - Konfiguration einfach und verständlich
 - Server aufsetzen ist nicht nötig
 - Grafische Oberfläche
- Sicherheit
 - Zugang nur für autorisierte Personen
 - Vertrauen darf nicht erzwungen werden
 - In Implementierung
 - In fremden Server
- Also:
 - Quelloffen

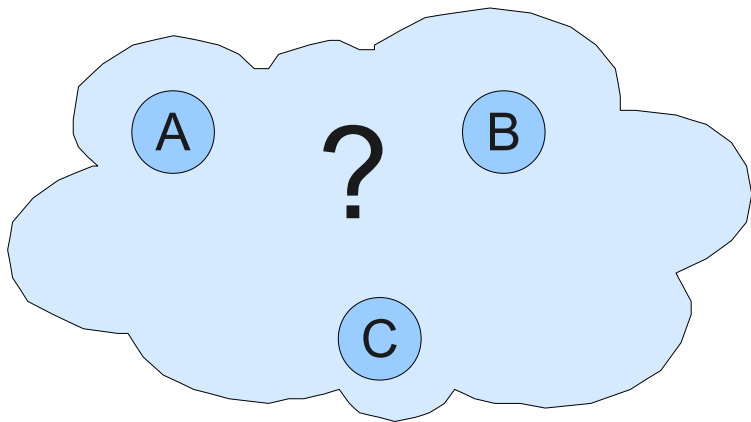
Ziele dieser Arbeit

- Benutzerfreundlichkeit
 - Konfiguration einfach und verständlich
 - **Server aufsetzen ist nicht nötig**
 - Grafische Oberfläche
- Sicherheit
 - Zugang nur für autorisierte Personen
 - Vertrauen darf nicht erzwungen werden
 - In Implementierung
 - **In fremden Server**
- Also:
 - Quelloffen
 - **(Möglichst) Dezentral**

Implementierung

- 1 Einleitung
- 2 Probleme**
- 3 Entwurf
- 4 Implementierung
- 5 Demo
- 6 Ergebnis

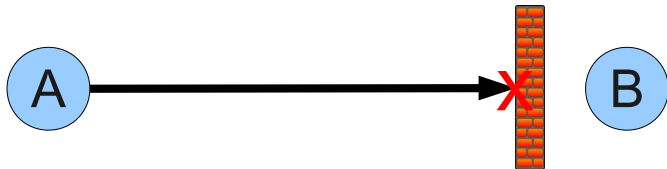
Bootstrapping



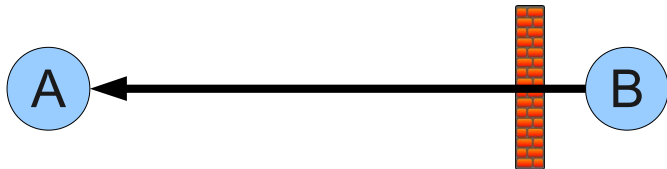
Network Address Translation



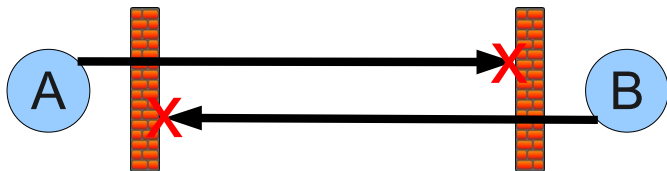
Network Address Translation



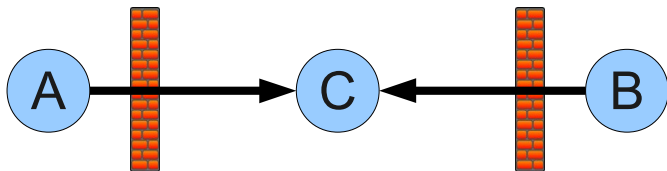
Network Address Translation



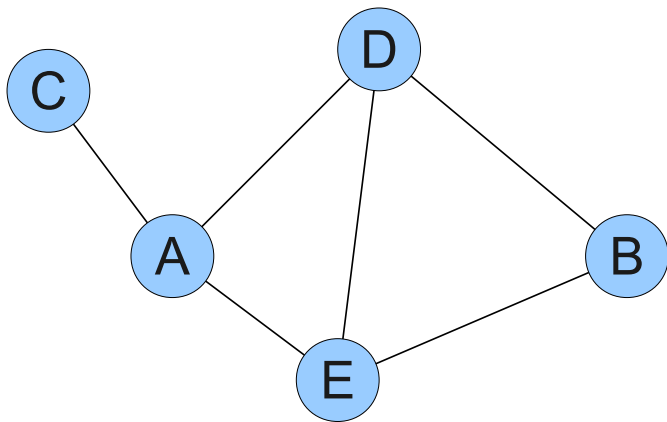
Network Address Translation



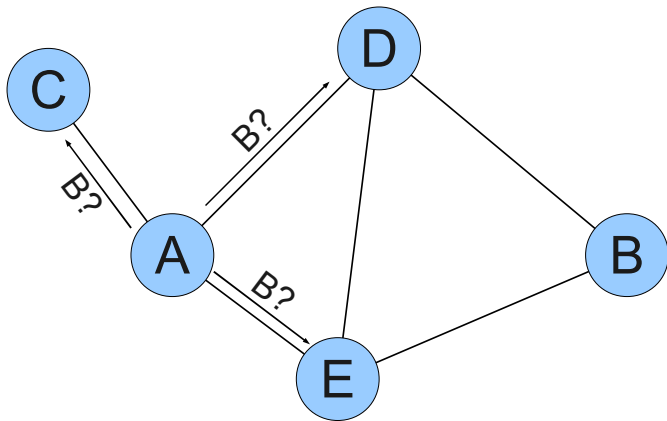
Network Address Translation



Routing



Routing



Sicherheit

Sicherheit

- Zwecklos: Sicherheit nach innen
 - (Virtuelle) LANs sind durch Designfehler unsicher (ARP-Spoofing)

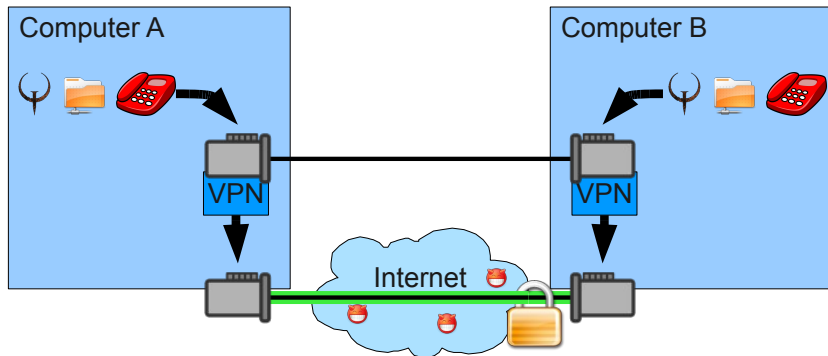
Sicherheit

- Zwecklos: Sicherheit nach innen
 - (Virtuelle) LANs sind durch Designfehler unsicher (ARP-Spoofing)
- Ziel: Sicherheit nach außen
 - Zugang zu dem virtuellen Netz kontrollieren
 - Problem: Dezentrale Kontrolle

Entwurf

- 1 Einleitung
- 2 Probleme
- 3 Entwurf**
- 4 Implementierung
- 5 Demo
- 6 Ergebnis

Schichten



Schichten

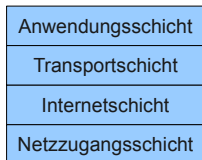
Knoten A

Knoten B

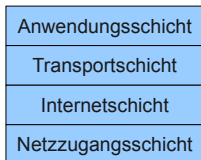
Knoten C

Schichten

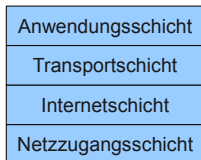
Knoten A



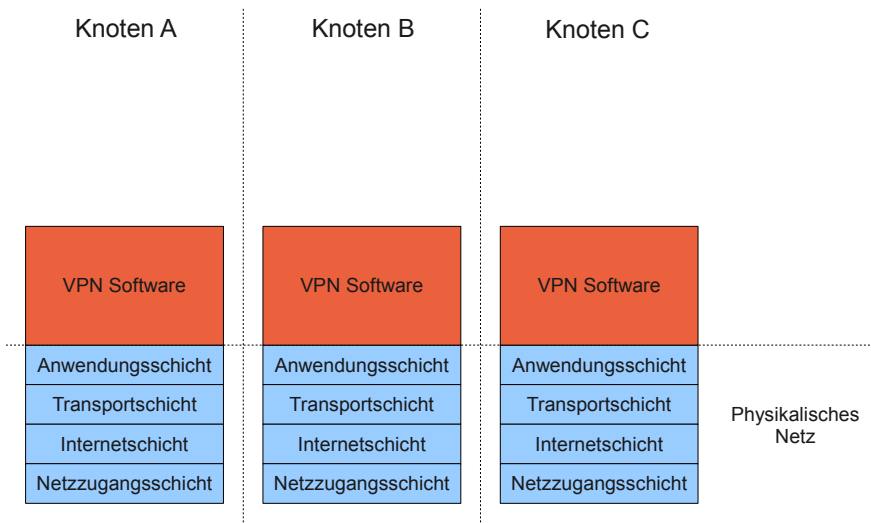
Knoten B



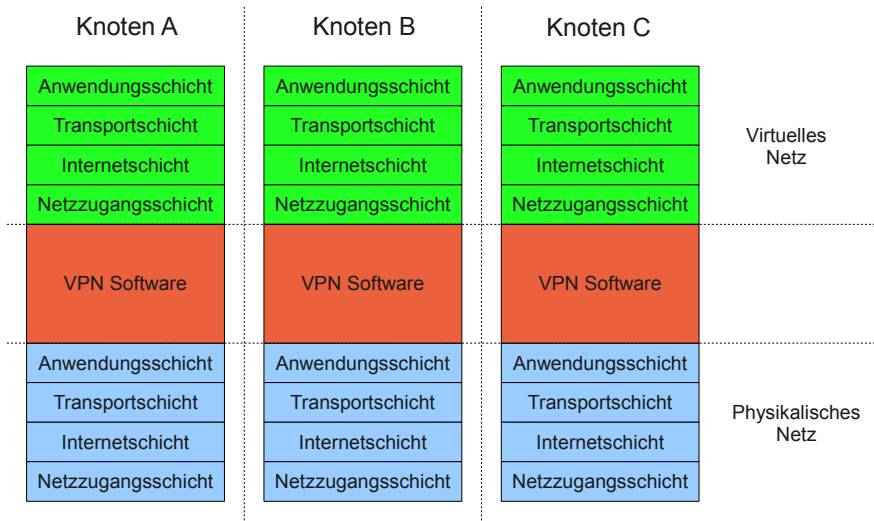
Knoten C

Physikalisches
Netz

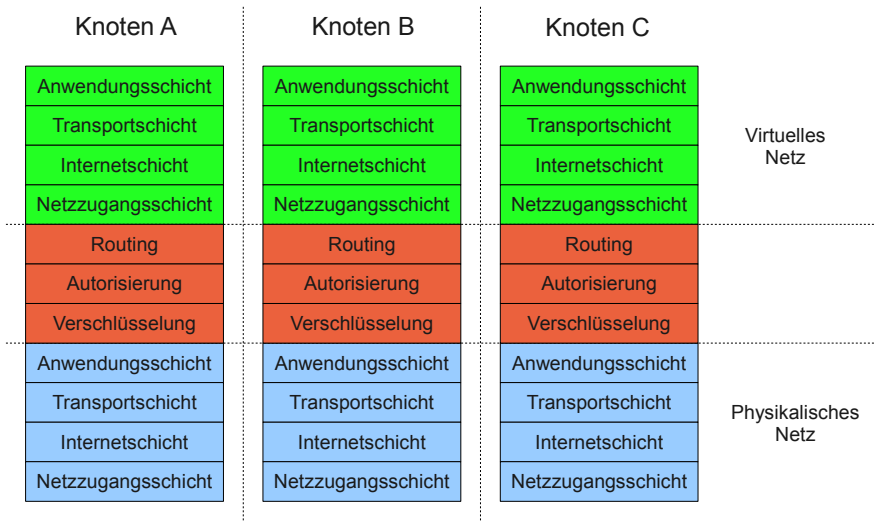
Schichten



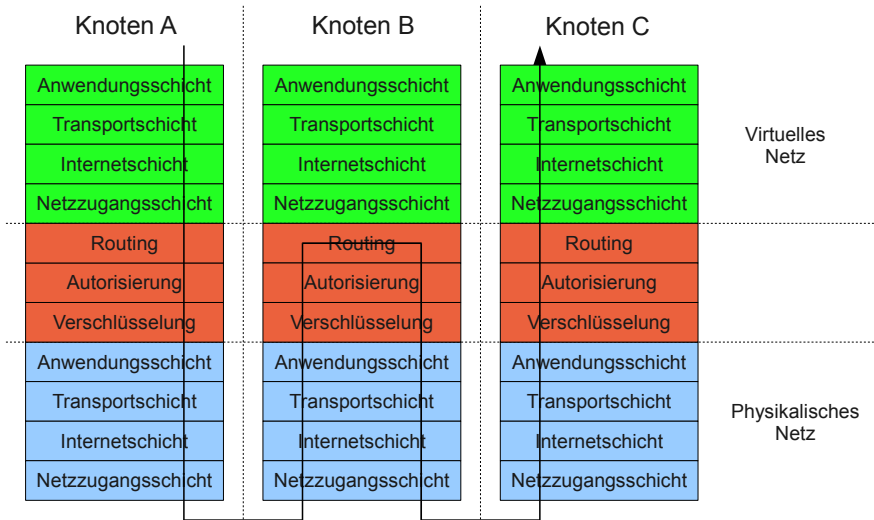
Schichten



Schichten



Schichten

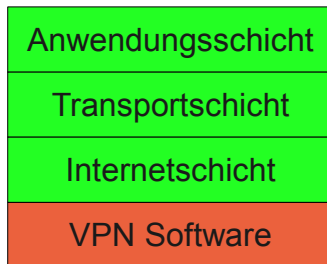


Virtueller Netzwerkadapter

- Zugriff

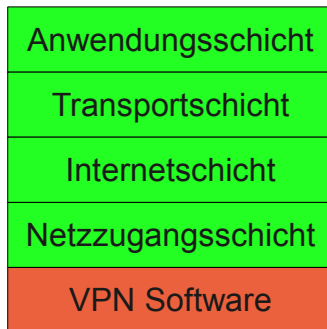
Virtueller Netzwerkadapter

- Zugriff
 - Internetschicht
 - ⇒ IP-Adressen



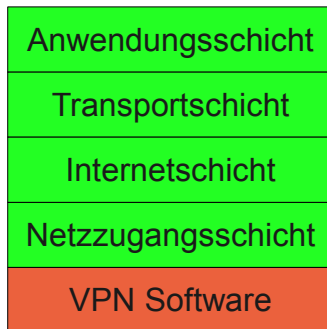
Virtueller Netzwerkadapter

- Zugriff
 - Internetschicht
 - ⇒ IP-Adressen
 - Netzzugangsschicht
 - ⇒ MAC-Adressen
 - Broadcasts



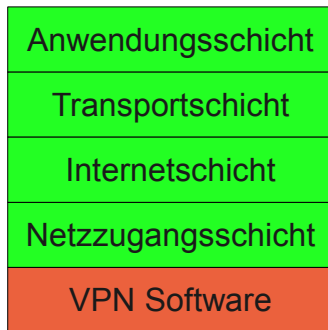
Virtueller Netzwerkadapter

- Zugriff
 - Internetschicht
 - ⇒ IP-Adressen
 - Netzzugangsschicht
 - ⇒ MAC-Adressen
 - Broadcasts
- TUN/TAP Schnittstelle



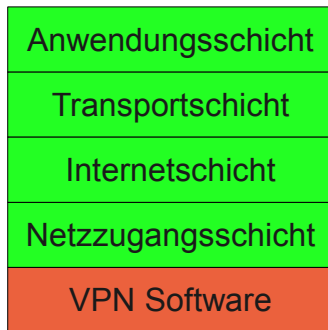
Virtueller Netzwerkadapter

- Zugriff
 - Internetschicht
 - ⇒ IP-Adressen
 - Netzzugangsschicht
 - ⇒ MAC-Adressen
 - Broadcasts
- TUN/TAP Schnittstelle
 - Verfügbar unter Unix (Linux, *BSD, MacOS)



Virtueller Netzwerkadapter

- Zugriff
 - Internetschicht
 - ⇒ IP-Adressen
 - Netzzugangsschicht
 - ⇒ MAC-Adressen
 - Broadcasts
- TUN/TAP Schnittstelle
 - Verfügbar unter Unix (Linux, *BSD, MacOS)
 - Mit extra Treiber auch unter Windows



Verteilte Datenbank

Verteilte Datenbank

- Verteilte Datenbank
 - Jeder Knoten kann Daten über sich veröffentlichen
 - Diese Daten werden verteilt und synchronisiert
 - ⇒ Jeder Knoten hat eine aktuelle Kopie aller Daten

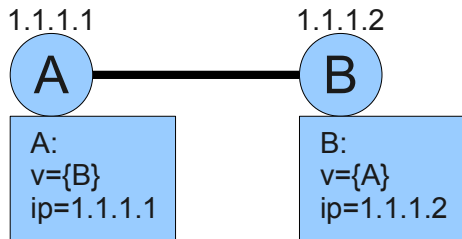
Verteilte Datenbank

- Verteilte Datenbank
 - Jeder Knoten kann Daten über sich veröffentlichen
 - Diese Daten werden verteilt und synchronisiert
 - ⇒ Jeder Knoten hat eine aktuelle Kopie aller Daten
- Jeder Knoten veröffentlicht:
 - Routinginformationen
 - Seine virtuelle MAC-Adresse
 - Liste der Nachbarn
 - Sonstiges
 - Seine virtuelle IP-Adresse
 - Seine physikalischen IP-Adressen
 - ...

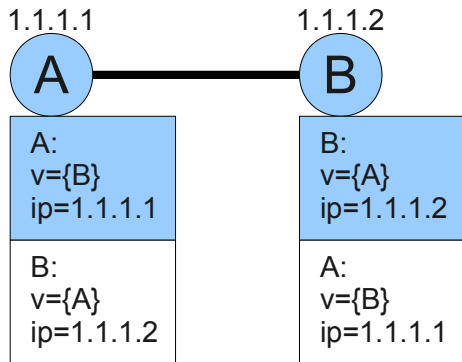
Verteilte Datenbank

- Verteilte Datenbank
 - Jeder Knoten kann Daten über sich veröffentlichen
 - Diese Daten werden verteilt und synchronisiert
 - ⇒ Jeder Knoten hat eine aktuelle Kopie aller Daten
- Jeder Knoten veröffentlicht:
 - Routinginformationen
 - Seine virtuelle MAC-Adresse
 - Liste der Nachbarn
 - Sonstiges
 - Seine virtuelle IP-Adresse
 - Seine physikalischen IP-Adressen
 - ...
- Prinzip
 - Jeder Knoten fragt seine Nachbarn regelmäßig: (polling)
Was weißt du über X

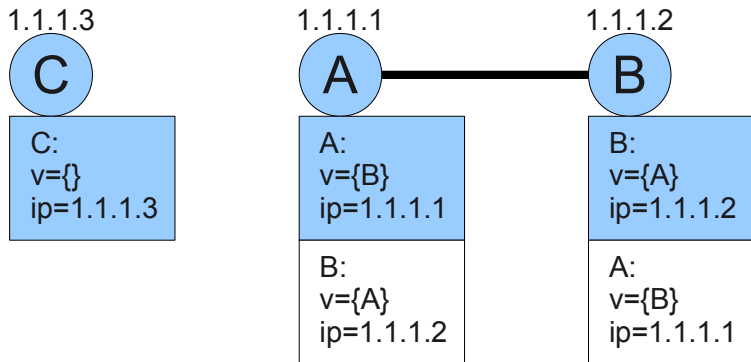
Verteilte Datenbank - Beispiel



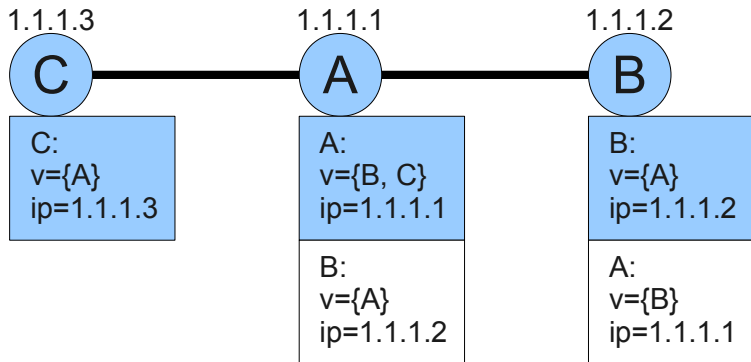
Verteilte Datenbank - Beispiel



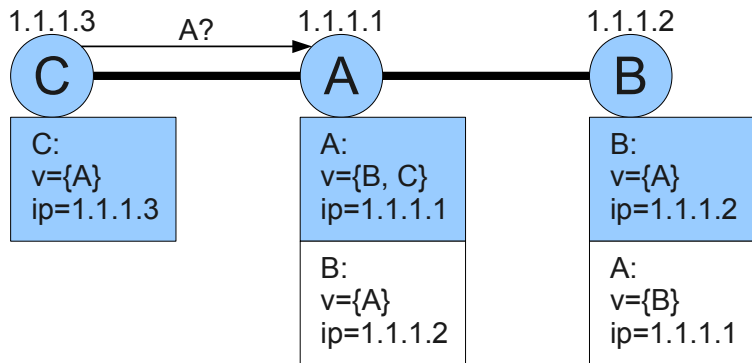
Verteilte Datenbank - Beispiel



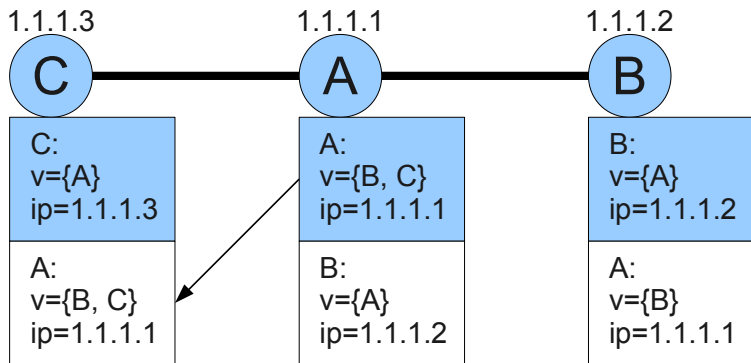
Verteilte Datenbank - Beispiel



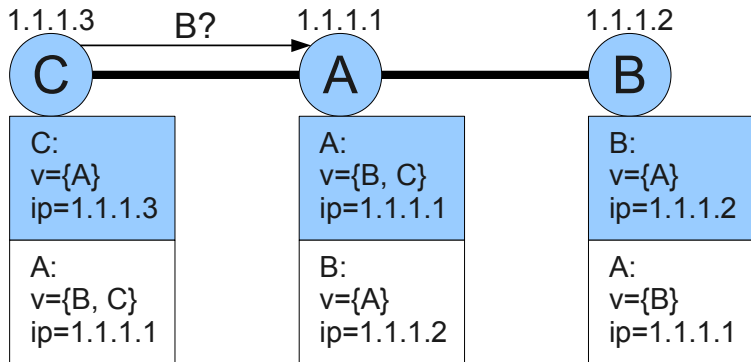
Verteilte Datenbank - Beispiel



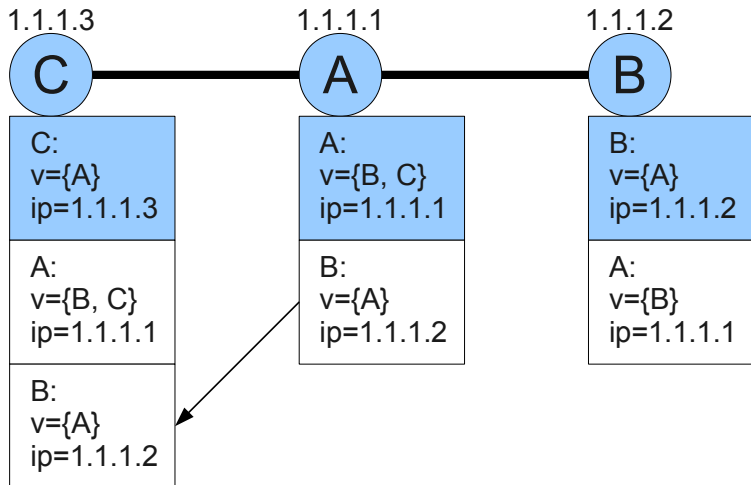
Verteilte Datenbank - Beispiel



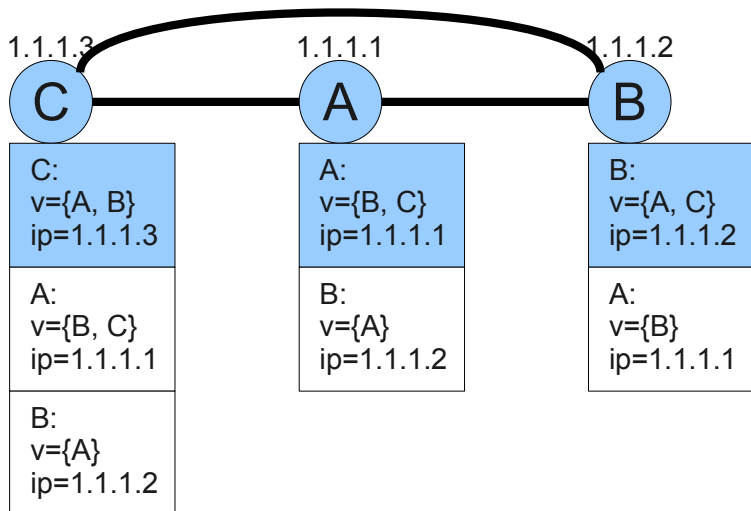
Verteilte Datenbank - Beispiel



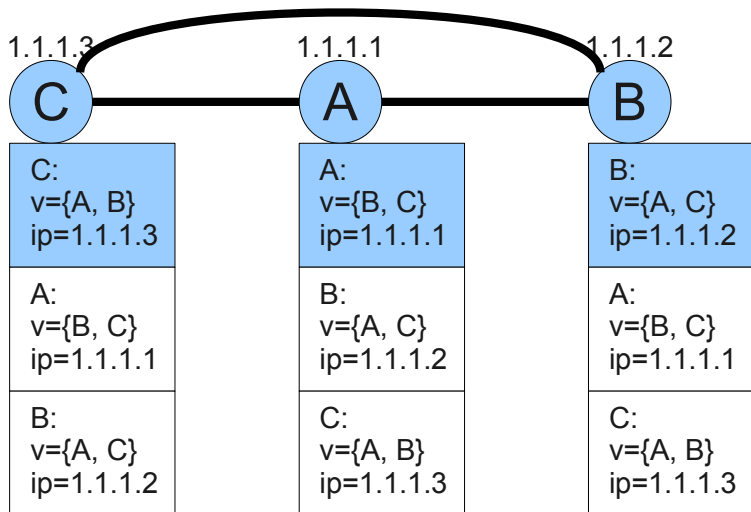
Verteilte Datenbank - Beispiel



Verteilte Datenbank - Beispiel



Verteilte Datenbank - Beispiel



Bootstrapping

Wie wird die erste Verbindung aufgebaut?

Bootstrapping

Wie wird die erste Verbindung aufgebaut?

- Lösung 1: Das Problem dem Benutzer überlassen

Bootstrapping

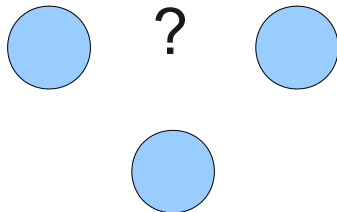
Wie wird die erste Verbindung aufgebaut?

- Lösung 1: Das Problem dem Benutzer überlassen
- Lösung 2: Einen BitTorrent-Tracker nutzen

Bootstrapping

Wie wird die erste Verbindung aufgebaut?

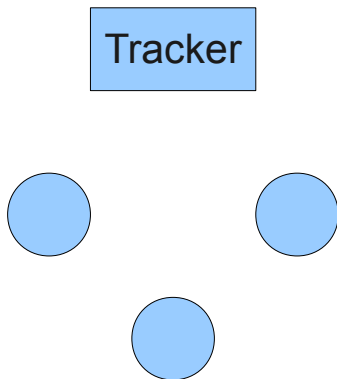
- Lösung 1: Das Problem dem Benutzer überlassen
- Lösung 2: Einen BitTorrent-Tracker nutzen



Bootstrapping

Wie wird die erste Verbindung aufgebaut?

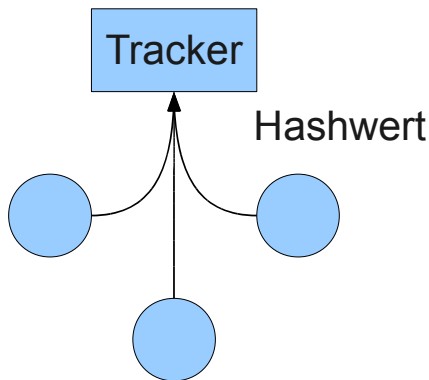
- Lösung 1: Das Problem dem Benutzer überlassen
- Lösung 2: Einen BitTorrent-Tracker nutzen



Bootstrapping

Wie wird die erste Verbindung aufgebaut?

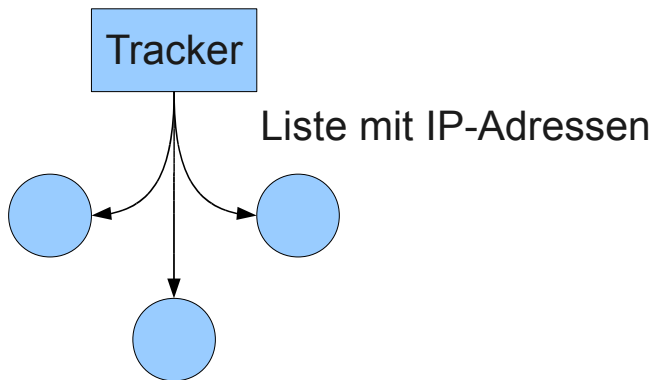
- Lösung 1: Das Problem dem Benutzer überlassen
- Lösung 2: Einen BitTorrent-Tracker nutzen



Bootstrapping

Wie wird die erste Verbindung aufgebaut?

- Lösung 1: Das Problem dem Benutzer überlassen
- Lösung 2: Einen BitTorrent-Tracker nutzen

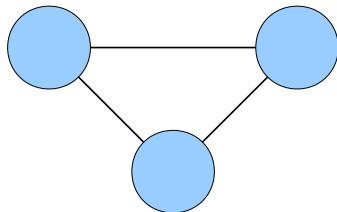
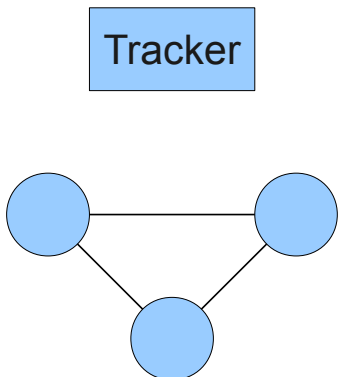


Bootstrapping

Wie wird die erste Verbindung aufgebaut?

- Lösung 1: Das Problem dem Benutzer überlassen
- Lösung 2: Einen BitTorrent-Tracker nutzen

Tracker



Routing

Routing

- Voraussetzungen
 - Jeder Knoten kennt den kompletten Verbindungsgraphen
 - Es wird versucht, alle möglichen Verbindungen aufzubauen
 - Die Distanz zweier Knoten im Verbindungsgraphen ist meist ≤ 2

Routing

- Voraussetzungen
 - Jeder Knoten kennt den kompletten Verbindungsgraphen
 - Es wird versucht, alle möglichen Verbindungen aufzubauen
 - Die Distanz zweier Knoten im Verbindungsgraphen ist meist ≤ 2
- Algorithmus
 - 1 Sende das Paket zu dem Nachbarn, der dem Ziel am nächsten ist
 - 2 Falls mehrere Möglichkeiten existieren:
 - ⇒ Wähle den Nachbarn, der zu mir die geringste Latenz hat

Sicherheit

Sicherheit

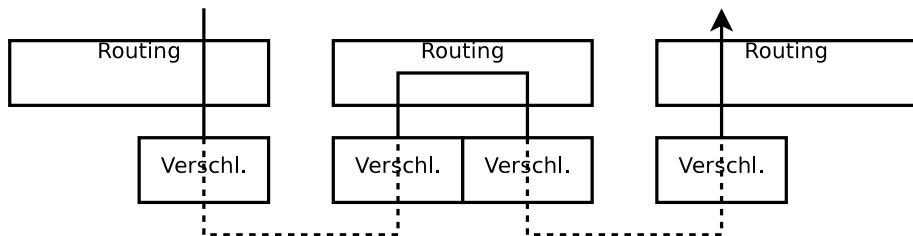
Ziel: Zugriff nur für autorisierte Personen

Sicherheit

Ziel: Zugriff nur für autorisierte Personen

- Verschlüsselungsschicht
⇒ Abhören verhindern
- Autorisierungsschicht
⇒ Zugang kontrollieren

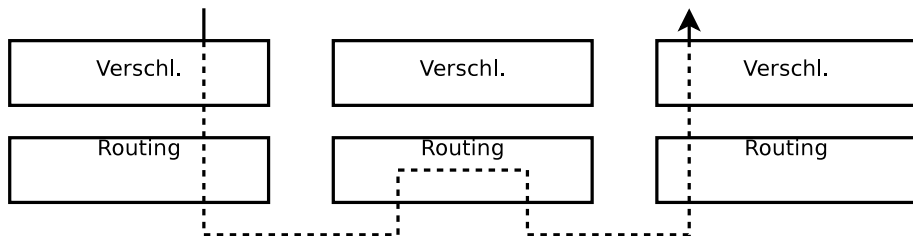
Verschlüsselungsschicht



Der Weg eines Paketes

—— unverschlüsselt - - - - verschlüsselt

Verschlüsselungsschicht



Der Weg eines Paketes

—— unverschlüsselt - - - - verschlüsselt

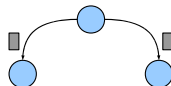
Autorisierung

Autorisierung

- Problem
 - Entscheidung, wer teilnehmen darf: Zentral
 - Überprüfung/Autorisierung: Dezentral

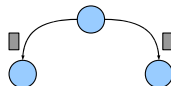
Autorisierung

- Problem
 - Entscheidung, wer teilnehmen darf: Zentral
 - Überprüfung/Autorisierung: Dezentral
- Lösung
 - Es werden Zugangsberechtigungen verteilt
 - Knoten können sich mit diesen gegenseitig autorisieren



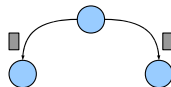
Autorisierung

- Problem
 - Entscheidung, wer teilnehmen darf: Zentral
 - Überprüfung/Autorisierung: Dezentral
- Lösung
 - Es werden Zugangsberechtigungen verteilt
 - Knoten können sich mit diesen gegenseitig autorisieren
- Schlüssel
 - Ein Schlüsselpaar pro Netzwerk
 - ⇒ Zum Signieren der Zugangsberechtigungen
 - Ein Schlüsselpaar pro Zugangsberechtigung
 - ⇒ Zum Autorisieren



Autorisierung

- Problem
 - Entscheidung, wer teilnehmen darf: Zentral
 - Überprüfung/Autorisierung: Dezentral
- Lösung
 - Es werden Zugangsberechtigungen verteilt
 - Knoten können sich mit diesen gegenseitig autorisieren
- Schlüssel
 - Ein Schlüsselpaar pro Netzwerk
 - ⇒ Zum Signieren der Zugangsberechtigungen
 - Ein Schlüsselpaar pro Zugangsberechtigung
 - ⇒ Zum Autorisieren
- Zusätzlich
 - Jeder Knoten hat eine Kennung, die er nicht ändern kann
 - Zugangsberechtigungen können ein Verfallsdatum haben
 - Der Netzwerkschlüssel kann weitergegeben werden



Zugangsberechtigungen

Zugangsberechtigungen

- Netzwerk-Einladung

I_N Informationen über das Netzwerk

P_N Öffentlicher Schlüssel des Netzwerkes

Sig_N Signatur über (I_N, P_N) mit Netzwerkschlüssel

S_N Geheimer Schlüssel des Netzwerkes

Zugangsberechtigungen

- Netzwerk-Einladung

 - I_N Informationen über das Netzwerk

 - P_N Öffentlicher Schlüssel des Netzwerkes

 - Sig_N Signatur über (I_N, P_N) mit Netzwerkschlüssel

 - S_N Geheimer Schlüssel des Netzwerkes

- Zugangs-Einladung/Zugangsberechtigung

 - I_A Informationen über den Knoten (z. B. Verfallsdatum)

 - P_A Öffentlicher Schlüssel des Zugangs

 - Sig_A Signatur über (I_A, P_A) mit Netzwerkschlüssel

 - S_A Geheimer Schlüssel des Zugangs

 - Kopie der Netzwerk-Einladung (I_N, P_N, Sig_N)

Zugangsberechtigungen

- Netzwerk-Einladung

 - I_N Informationen über das Netzwerk

 - P_N Öffentlicher Schlüssel des Netzwerkes

 - Sig_N Signatur über (I_N, P_N) mit Netzwerkschlüssel

 - S_N Geheimer Schlüssel des Netzwerkes

- Zugangs-Einladung/Zugangsberechtigung

 - I_A Informationen über den Knoten (z. B. Verfallsdatum)

 - P_A Öffentlicher Schlüssel des Zugangs

 - Sig_A Signatur über (I_A, P_A) mit Netzwerkschlüssel

 - S_A Geheimer Schlüssel des Zugangs

 - Kopie der Netzwerk-Einladung (I_N, P_N, Sig_N)

- Kennung

 - ID_N $\text{hash}(Sig_N)$

 - ID_A $\text{hash}(Sig_A)$

Protokoll

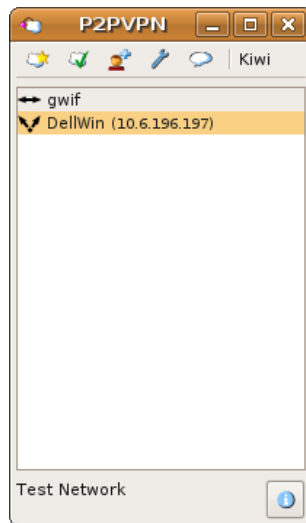
- ① X und Y setzen den Schlüssel der Verschlüsselungsschicht auf ID_N .
- ② $X \rightarrow Y : (I_{AX}, P_{AX}, Sig_{AX})$
- ③ $Y \rightarrow X : (I_{AY}, P_{AY}, Sig_{AY})$
- ④ X überprüft die Signatur und dessen Verfallsdatum von Y .
- ⑤ Y überprüft die Signatur und dessen Verfallsdatum von X .
- ⑥ $X \rightarrow Y : E_{S_{AX}}(R_X)$ mit ($R_X = \text{Zufallszahl}$)
- ⑦ $Y \rightarrow X : E_{S_{AY}}(R_Y)$ mit ($R_Y = \text{Zufallszahl}$)
- ⑧ X setzt den Schlüssel der Verschlüsselungsschicht auf $D_{P_{AY}}(E_{S_{AY}}(R_Y)) \oplus R_X$.
- ⑨ Y setzt den Schlüssel der Verschlüsselungsschicht auf $D_{P_{XY}}(E_{S_{AX}}(R_X)) \oplus R_Y$.

Implementierung

- 1 Einleitung
- 2 Probleme
- 3 Entwurf
- 4 Implementierung**
- 5 Demo
- 6 Ergebnis

Implementierung

- P2PVPN
 - Programmiersprache
 - Java
 - C für die Schnittstelle zum virtuellen Netz
 - Plattformen
 - Linux 32Bit
 - Windows 32Bit

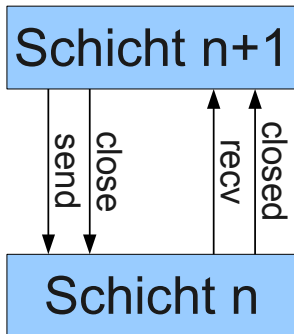


Schichten - Schnittstelle

Die Schnittstelle zwischen den Schichten ist Ereignisbasiert

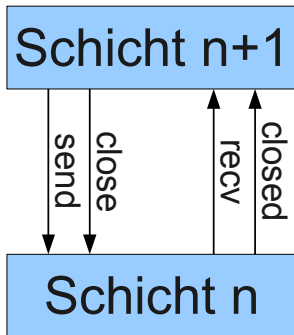
Schichten - Schnittstelle

Die Schnittstelle zwischen den Schichten ist Ereignisbasiert



Schichten - Schnittstelle

Die Schnittstelle zwischen den Schichten ist Ereignisbasiert

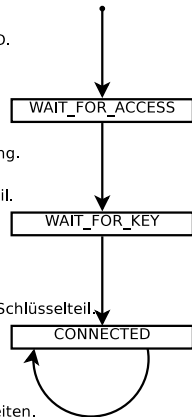


Setze Schlüssel auf Netzwerk ID.
Sende Zugangs-Einladung.

Paket wurde empfangen:
Paket ist eine Zugangs-Einladung.
Überprüfe Einladung.
Erzeuge und Sende Schlüsselteil.

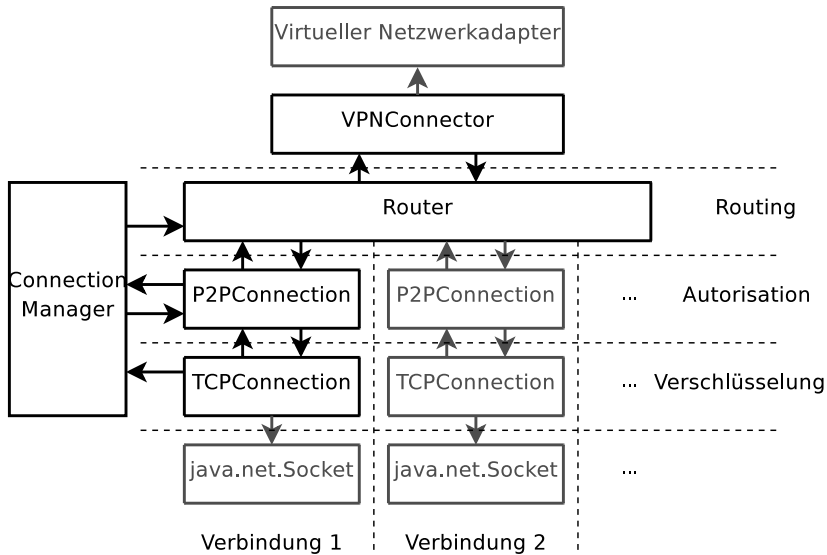
Paket wurde empfangen:
Paket ist ein Schlüsselteil.
Setze Schlüssel auf
mein Schlüsselteil XOR sein Schlüsselteil.

Paket wurde empfangen:
Paket enthält Daten.
Paket an obere Schicht weiterleiten.



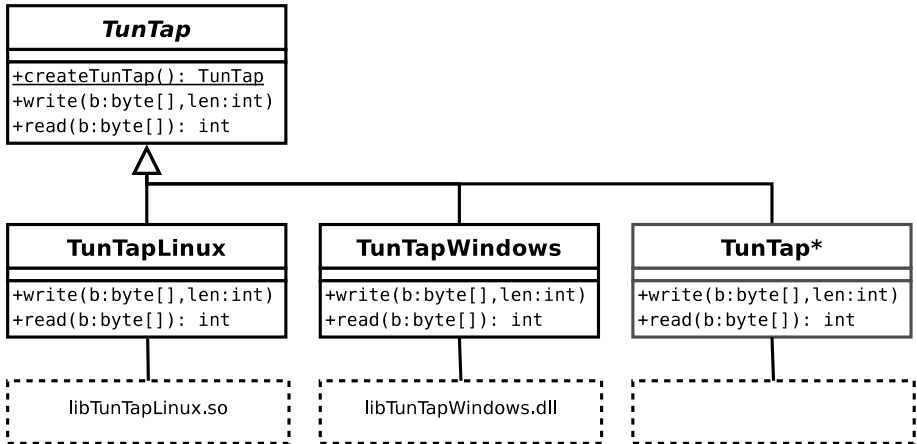
Schichten - Genauer

Schichten - Genauer



Virtueller Netzwerkadapter

Virtueller Netzwerkadapter



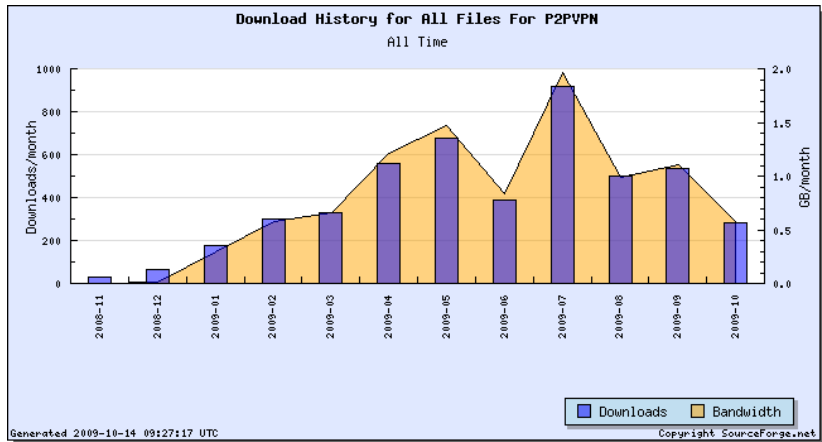
Demo

- 1 Einleitung
- 2 Probleme
- 3 Entwurf
- 4 Implementierung
- 5 Demo**
- 6 Ergebnis

Ergebnis

- 1 Einleitung
- 2 Probleme
- 3 Entwurf
- 4 Implementierung
- 5 Demo
- 6 Ergebnis**

Downloads



Fast 5000 Downloads

Hilfe

- Fehlerberichte

Hilfe

- Fehlerberichte
- Featurewünsche
 - UDP-Broadcast Workaround
 - Chat
 - Rechtsklick

Hilfe

- Fehlerberichte
- Featurewünsche
 - UDP-Broadcast Workaround
 - Chat
 - Rechtsklick
- Mitarbeit
 - Icons
 - Linux 64Bit, MIPS
 - Windows Installer

Anwendungen

Anwendungen

- LAN übers Internet

Anwendungen

- LAN übers Internet
- Sperrern umgehen

Anwendungen

- LAN übers Internet
- Sperrungen umgehen
- Sicherheit

Lob

- *„Very nice project (P2PVPN). Just want you thank you for your effort and keep it opensource ;)“*
- *„The software looks like it's coming along very nicely, I plan on using it in the near future. I commend you for the hard work, even though it is for your education, it looks like a very nice addition to the open source community. ;)“*
- *„First of all, I congrats you for such a fine piece of software. It's exactly what I had in mind to develop when I have learned enough programming myself.“*
- *„Great project you have going, in fact I was a bit saddened when I found your page tonight, as I just spent a bunch of time typing out a idea very similar to yours thinking it was unique.“*
- *„hyper-awesome“, „Sheer simplicity and elegancy – wow.“, „just sheer brilliance“*

Erreichte Ziele

- Benutzerfreundlichkeit
 - Konfiguration weitgehend automatisch
 - Autorisierung verständlich
 - Knoten finden durch BitTorrent-Tracker
 - Grafische Oberfläche
- Sicherheit

Erreichte Ziele

- Benutzerfreundlichkeit
 - Konfiguration weitgehend automatisch
 - Autorisierung verständlich
 - Knoten finden durch BitTorrent-Tracker
 - Grafische Oberfläche
- Sicherheit
 - Implementierung und Protokoll sind offen
 - Zugriff von außen nicht möglich
 - Knoten finden durch eigenen Server