Praktikum Teil II

Realisierung einer LAN-LAN Kopplung auf Basis eines OpenVPN-Tunnels mit Remote Access

(Gruppe 2 $\leftrightarrow$ Gruppe 1)

Planung & Dokumentation Gruppe 2

Andreas Lemke	andreas_lemke@gmx.de
Torsten Eymann	dr@shnuggle.de
Jan-Nicolas Schulze	jan-nicolas.schulze@FernUni-Hagen.de
Christof Ullerich	mail_to_christof@web.de

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Ziel

Das Ziel wird durch die folgende Aufgabenstellung beschrieben:

- LAN-LAN Kopplung auf Basis von OpenVPN
- dynamische Namensauflösung
- Verteilung der IP Adressen über statisches Routing
- gegenseitige Erreichbarkeit der Clients über Ping

Im Rahmen der Bearbeitung wurde das Ziel u.a. durch Rücksprache mit dem Betreuer sowie die Diskussion innerhalb der Arbeitsgruppe weiter konkretisiert.

2 Planung

2.1 Ressourcen

Die Aufgabenverteilung gestaltete sich wie folgt

Andreas Lemke	Dokumentation
Torsten Eymann	Dokumentation
Jan-Nicolas Schulze	Konfiguration des OpenVPN Servers
Christof Ullerich	Präsentation

Der Gruppe 2 standen nachfolgend aufgelistete Server zur Verfügung, welche für die gruppeninternen Tests genutzt wurden.

	Server / Hostname	**Server Betriebssystem**
A. Lemke	vpn1599.homeip.net	Ubuntu Server 8.04 LTS
T. Eymann	shnuggle.dyndns.org	Debian Etch 4.0
N. Schulze	meph.homeip.net	Debian Etch 4.0

Die dynamische Namensauflösung erfolgt durch einen entsprechenden Dienstleister - hier `dyndns.org` und `homeip.net`.

OpenVPN-Server Gruppe 2

Beim weiter unten beschriebenen LAN-LAN Zusammenschluss übernimmt der Server `meph.homeip.net` die Funktion des Gruppenservers. Konkret handelt es sich hierbei um einen Router mit entsprechender Firewall sowie Port-Forwarding mit einem Laptop als Basis. Die Clients der Gruppe arbeiten teils unter Windows und teils unter Linux.

2.2 Kommunikation / Dokumentation

Die gruppeninterne Kommunikation erfolgte im Wesentlichen über eMail. Andere Kanäle (Skype und ICQ) waren nach Absprache möglich. Es wurde ein zentrales Wiki eingerichtet, in welchem begleitend alle Informationen gesammelt wurden und das somit als Basis für diese Dokumentation diente. Das Wiki ist unter
`http://vpn1599.homeip.net/wiki/doku.php?id=doku_hilfe` zu erreichen.

2.3 Bearbeitungsbasis / Erstkonfiguration

Als Arbeitsgrundlage wurden die entsprechenden HowTo's auf der OpenVPN-Website gewählt. `http://www.openvpn.net/index.php/documentaion/howto.html` Die Erstkonfiguration erfolgte auf Basis der dort angegebenen Client / Server config.

2.4 Betriebsmodus

In der Gruppe wurden zwei prinzipielle Wege diskutiert, die Aufgabenstellung zu bearbeiten und das Ziel zu erreichen. Diese ergaben sich durch die möglichen Betriebsmodi von OpenVPN, sowie durch unterschiedliche Interpretationen der Aufgabenstellung.

1. Bridging (TAP-Device) auf Layer 2 arbeitend
2. Routing (TUN-Device) auf Layer 3 arbeitend

Die Entscheidung fiel für die Routing-Variante mittels TUN-Device, da eine Layer 3 basierte Lösung zur Bearbeitung der Aufgabenstellung ausreichend war.

2.5 Authentifikation

Die Authentifizierung wurde auf Basis signierter Zertifikate realisiert. Grund hierfür ist der durch Zertifikate garantierte hohe Sicherheitsstandard. Die Zertifikate werden vom Serverbetreiber (CA) ausgestellt. Er bestimmt somit auch über die zugelassenen Client-Partner.

Die Erstellung der Zertifikate erfolgte mittels `openssl`, dessen Handhabung mit den im openVPN-Paket mitgelieferten shell-Scripten vereinfacht wird. Zuerst muss eine CA (Certificate authority) erzeugt werden, die alle anschliessend erzeugten Server- und Clientzertifikate signiert. Dieser CA müssen alle beteiligten Kommunikationspartner vertrauen. Die dabei entstehenden Dateien werden wie folgt verteilt:

Dateiname	benötigt von	Zweck	geheim
ca.crt	Server und Client	Root CA Zertifikat	nein
ca.key	Schlüssel-Server	Root CA Key	ja
dh_1024.pem	Server	Diffie-Hellmann Parameter	nein
server.crt	Server	Server Zertifikat	nein
server.key	Server	Server Key	ja
client[n].crt*	Client[n]	Client[n] Zertifikat	nein
client[n].key	Client[n]	Client[n] Key	ja
* client[n] steht für den common name der Zertifikate			

2.6 Protokoll / Port

Es wurde das UDP Protokoll gewählt, da von der Verwendung TCP über TCP abgeraten wird. Grund sind eventuelle Timing-Probleme und ein erhöhtes Aufkommen an Kontrollflussnachrichten.

Als Port wurde der Standardport 1194 gewählt. Alle drei Server werden via Router / Firewall über eine entsprechende NAT an das Internet angebunden.

3 Dokumentation

3.1 Gruppeninterne Tests Nr. 1-3

Als erster Test wurde eine Client-Konfiguration mit `meph.homeip.net` als Server aufgebaut. Die Ausstellung der Zertifikate sowie die Konfiguration des Servers und der Clients erfolgte durch den Server-Betreiber. Dieser Testaufbau wurde analog mit den anderen Servern wiederholt. Die Tests verliefen durchgehend erfolgreich.

3.2 Gruppeninterner Test Nr. 4

In einem weiteren Versuch wurden zwei OpenVPN Server miteinander verbunden. Als Server fungierte dabei `meph.homeip.net`, die Rolle des Clients wurde von `shnuggle.dyndns.org` übernommen. In diesem Zusammenhang wurden diverse Konfigurationsmöglichkeiten erprobt.

Im Mittelpunkt stand die Erprobung zweier Szenarien, welche bei einer gruppeninternen Diskussion der Aufgabenstellung erarbeitet wurden

- Das **erste Szenario** setzt die Kopplung mehrerer Rechner in einem VPN-Netz um und leistet die entsprechend der Aufgabenstellung geforderte Erreichbarbeit der Clients untereinander mittels Ping.
- Im **zweiten Szenario** werden auf Basis des ersten Szenarios die Clients hinter den VPN-Gateways zu einem Gesamtnetz verbunden.

3.3 Gruppenübergreifender Test - Szenario 1

Nach erfolgreichen gruppeninternen Tests der oben beschriebenen Szenarien, wurde die Abstimmung hinsichtlich der Konfigurations-Parameter mit Gruppe 1 vorgenommen.

Das Ziel des gruppenübergreifenden Tests war es, den in Szenario 1 beschriebenen und intern getesteten Aufbau zusammen mit Gruppe 1 zu realisieren. Dem lag folgende Konfiguration zu Grunde:

Device	TUN
Authentifikation	Zertifikate, erstellt durch den Server Betreiber Gruppe 2
Server Rolle	Server Gruppe 2
Client Rolle	Server Gruppe 1

Die IP Adressen wurden den Clients statisch mit dem Parameter **ifconfig-push** im jeweiligen Client-Config-File im client-config-dir (→siehe Konfiguration) zugewiesen. Die Unterteilung in zwei Subnetze stellt die unabhängige Erweiterung der beiden VPN-Netze durch weitere Clients sicher. Durch die dezentrale Verwaltung der beiden VPN-Netze kann der Abstimmungsbedarf gesenkt werden.

	Netz 10.8.2.0/30			
Server Config Gr1	**Netzwerkadresse**	**Server Endpunkt**	**Client Endpunkt**	**BC Adresse**
Gruppe 1 Client	10.8.2.88	10.8.2.89	10.8.2.90	10.8.2.91
A. Lemke	10.8.2.8	10.8.2.9	10.8.2.10	10.8.2.11
C. Ullerich	10.8.2.28	10.8.2.29	10.8.2.30	10.8.2.31
N. Schulze	10.8.2.48	10.8.2.49	10.8.2.50	10.8.2.51
T. Eymann	10.8.2.68	10.8.2.69	10.8.2.70	10.8.2.71
Server Config GR 2	**Netz 10.8.1.0/30**			
M. Arnskötter	10.8.1.8	10.8.1.9	10.8.1.10	10.8.1.11
A. Wolf	10.8.1.28	10.8.1.29	10.8.1.30	10.8.1.31
K. Gustat	10.8.1.48	10.8.1.49	10.8.1.50	10.8.1.51
S. Krapp	10.8.1.68	10.8.1.69	10.8.1.70	10.8.1.71

Der Netzplan, der als Basis des Aufbaus verwendet wurde, liegt als Anlage 1 bei, wobei die Adressen der 192.168.x.x Netze erst im Szenario 2 eine Bedeutung erlangen. (siehe Abschnitt 3.4)

Die nachfolgend beschriebene Konfiguration wird durch das in Anlage 2 angegebene Schema verdeutlicht. Das Pingen fand somit auf Basis der oben gelisteten VPN-IPs statt.

Server Gruppe 1

Der Server der Gruppe 1 soll das Netz der Gruppe 2 routen. Der Gateway ist ihm als Client des Servers von Gruppe 2 bekannt.

```
route 10.8.2.0 255.255.255.0
```

Der Server der Gruppe 1 teilt seinen Clients eine statische Route ins Netz der Gruppe 2 mit.

```
push "route 10.8.2.0 255.255.255.0"
```

Die Clients können durch den Server explizite Konfigurationen erhalten:

```
client-config-dir ccd
```

Die Clients sollen sich untereinander sehen können:

```
client-to-client
```

Server Gruppe 2

Der Server wird über die `server.conf` konfiguriert. Der folgende Ausschnitt enthält alle routingspezifischen Einträge:

```
(1)    route 10.8.1.0 255.255.255.0
(2)    client-to-client
(3)    client-config-dir ccd
```

Der Server soll Pakete in das Netz der Gruppe 1 routen `(1)`.
Die Clients des Servers, zu denen auch der Server der Gruppe 1 zählt, sollen sich untereinander sehen können. Das regelt die Anweisung `(2)`.
Über `(3)` wird dem Server der Ort der Konfigurations-Dateien für die Clients - hier `/ccd` - mitgeteilt, in dem sich für jeden der Clients eine separate Konfigurationsdatei befinden kann.

Client-Config-Files für Gruppe 2

Die Client-Config Files sind recht kurz gehalten. Für die Clients der Gruppe zwei sehen sie folgendermaßen aus:

```
(1)    push "route 10.8.1.0 255.255.255.0"
(2)    ifconfig-push 10.8.2.10 10.8.2.9
```

Über die `push` Anweisung werden die Clients aufgefordert, eine Route in das Netz der Gruppe 1 in ihre Routing-Tabelle aufzunehmen. Die `ifconfig-push` Anweisung `(2)` weist dem Client die IP Adresse für sein TUN Interface zu.

Spezielle Client-Config-File für Gruppe 1

In der Config-Datei für den Server der Gruppe 1, der ja ebenfalls ein Client des Servers der Gruppe 2 ist, gibt es eine geringfügige Änderung:

```
(1)    iroute 10.8.1.0 255.255.255.0
(2)    ifconfig-push 10.8.2.90 10.8.2.89
```

Über den `iroute` Befehl `(1)` wird der Server der Gruppe 2 angewiesen Pakete für das Netz der Gruppe 1 über diesen Client zu routen.

3.4 Konfiguration Szenario 2

Im Folgenden wird der theoretische Aufbau entsprechend Szenario 2 beschrieben. Er soll die Erreichbarkeit der hinter dem VPN-Verbund liegenden Netze (Gruppe 1 192.168.1.0/24 sowie Gruppe 2 192.168.2.0/24) ermöglichen. (siehe Netzplan Anlage 1)

Diese Konfiguration wird nur theoretisch abgehandelt, da sie bereits erfolgreich gruppenintern getestet wurde und einen wesentlich höheren Abstimmungs-/Verwaltungs- und Testaufwand bedeuten würden. Auf einen Live-Test mit der Gruppe 1 wird daher verzichtet.

- Der Server bekommt die Anweisung, dass er die beiden hinteren 192.168.x.x Netze routen soll.
- push route für das hintere Netz der jeweils anderen Gruppe
- die `iroute`-Direktiven änder sich entsprechend auf die jeweiligen eth0 Adressen

Konfiguration Server Gruppe 2

Der Server soll in das Netz der Gruppe 1 routen. Es werden die einzelnen IP Adressen geroutet, da das verwendete Netz schon auf dem Server der Gruppe 1 lokal verwendet wird.

```
route 192.168.1.11 255.255.255.255
route 192.168.1.12 255.255.255.255
route 192.168.1.13 255.255.255.255
route 192.168.1.14 255.255.255.255
```

Der Server soll außerdem das eigene (hintere) Netz der Gruppe 2 routen.

```
route 192.168.2.0 255.255.255.0
```

Die Clients der Gruppe 2 müssen informiert werden, dass der Server die Route in das Netz 192.168.2.0/24 kennt. (Das gesamte Netz 192.168.2.0/24 soll über den Server laufen.

```
push "route 192.168.2.0 255.255.255.0"
```

Client-Config-Files für Gruppe 2

```
iroute 192.168.2.x 255.255.255.255   //eth0 Interface angegeben
```

Die Client-Config-Files sind u.a. notwendig, damit der Server der Gruppe 1 diese Route nicht erneut bekommt, denn er kennt sie schon. Für die o.g. IP Adresse gilt: $x \in \{11, 12, 13, 14\}$, entsprechend der Clients von Gruppe 2. (siehe Abbildung 3)

```
push "route 192.168.1.11 255.255.255.255"
push "route 192.168.1.12 255.255.255.255"
push "route 192.168.1.13 255.255.255.255"
push "route 192.168.1.14 255.255.255.255"
```

Spezielle client-config für Gruppe 1

Es muss festgelegt werden, wie der Server routen soll. In dieser Konfiguration kann der Server der Gruppe 2 die hinteren Interfaces der Gruppe 1 über diesen Client erreichen.

```
iroute 192.168.1.11 255.255.255.255
iroute 192.168.1.12 255.255.255.255
iroute 192.168.1.13 255.255.255.255
iroute 192.168.1.14 255.255.255.255
```

Server Gruppe 1

Der Server soll die Netze der Gruppe 1 und 2 routen. Bei Konflikten mit dem lokalen Netz können auch die einzelnen IP Adressen (analog zum Server der Gruppe 1) geroutet werden.

```
route 192.168.2.0 255.255.255.0
route 192.168.1.0 255.255.255.0
```

Die Clients müssen informiert werden, dass der Server die hinteren Netze der Gruppe 1 und 2 routet. (erzeugen der entsprechen Routen bei den Clients)

```
push "route 192.168.1.0 255.255.255.0"
push "route 192.168.2.0 255.255.255.0"
```

Client-Config-Files

Dem Server wird mitgeteilt, an welchen der Clients die eingehenden Pakete zu schicken sind.

```
iroute 192.168.1.x 255.255.255.255
```

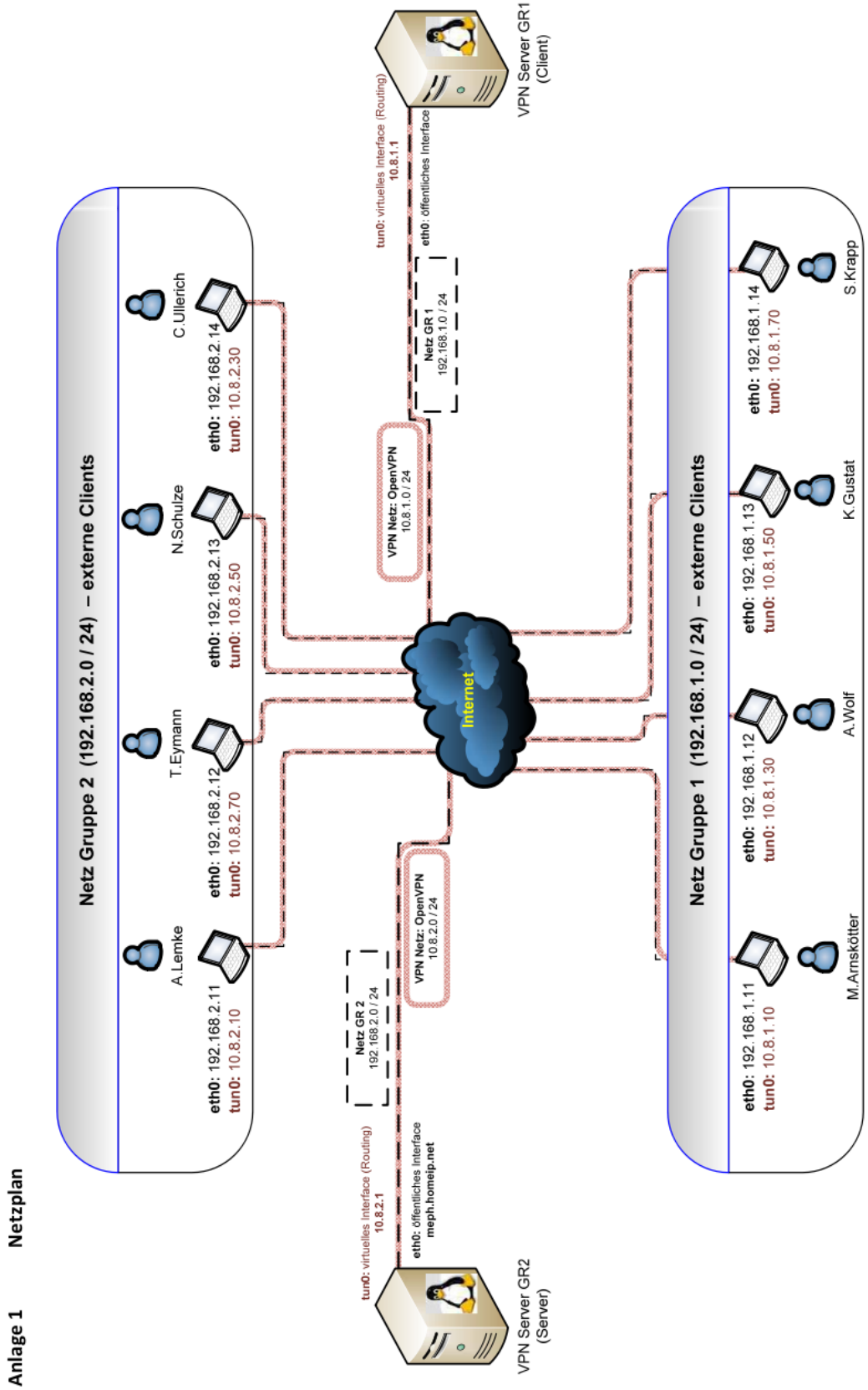

Abbildung 1: Netzplan

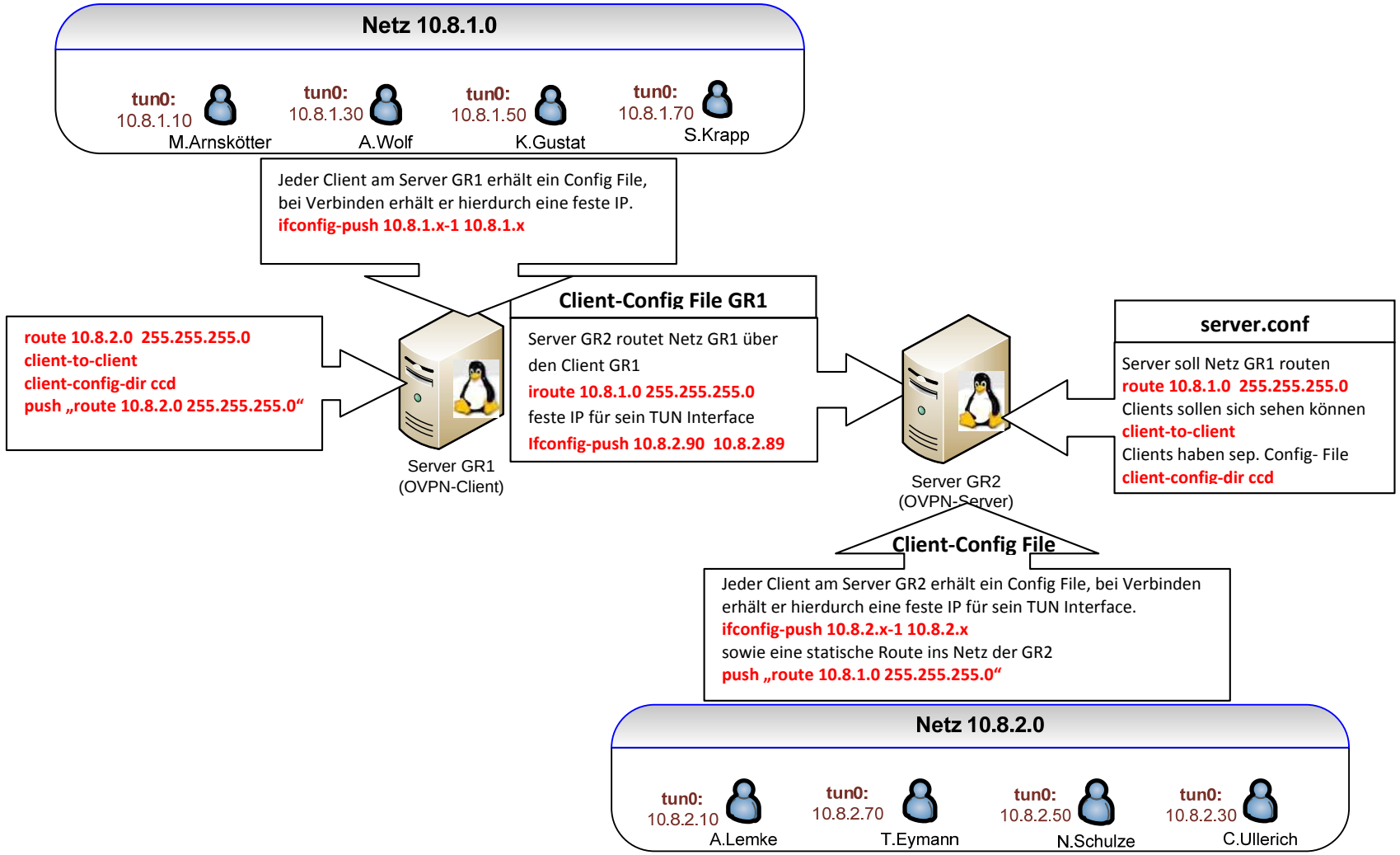

Abbildung 2: Konfiguration Szenario 1 / Routing zum gruppenübergreifenden Test

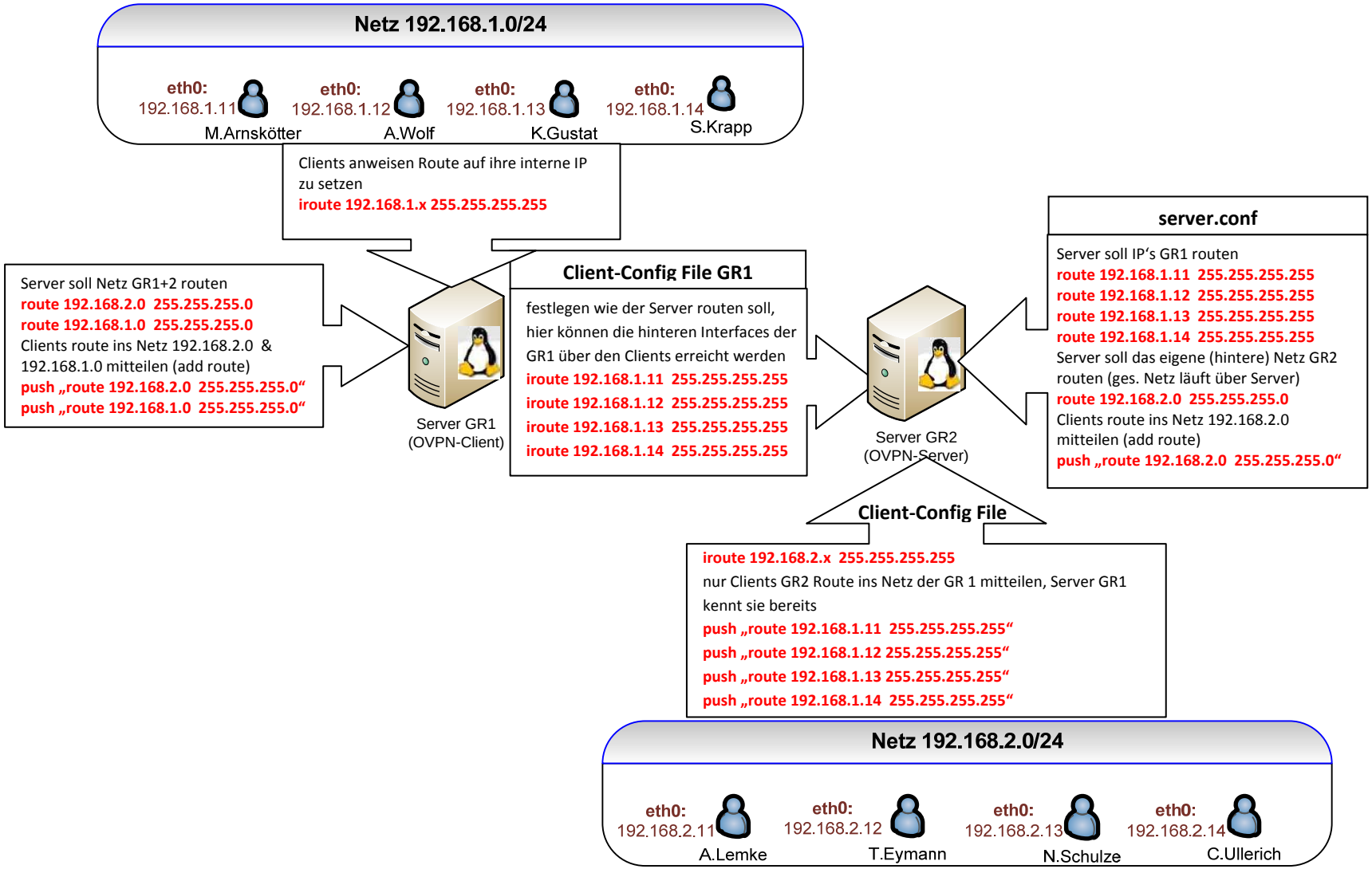

Abbildung 3: Konfiguration Szenario 2 / Routing zum gruppenübergreifenden Test