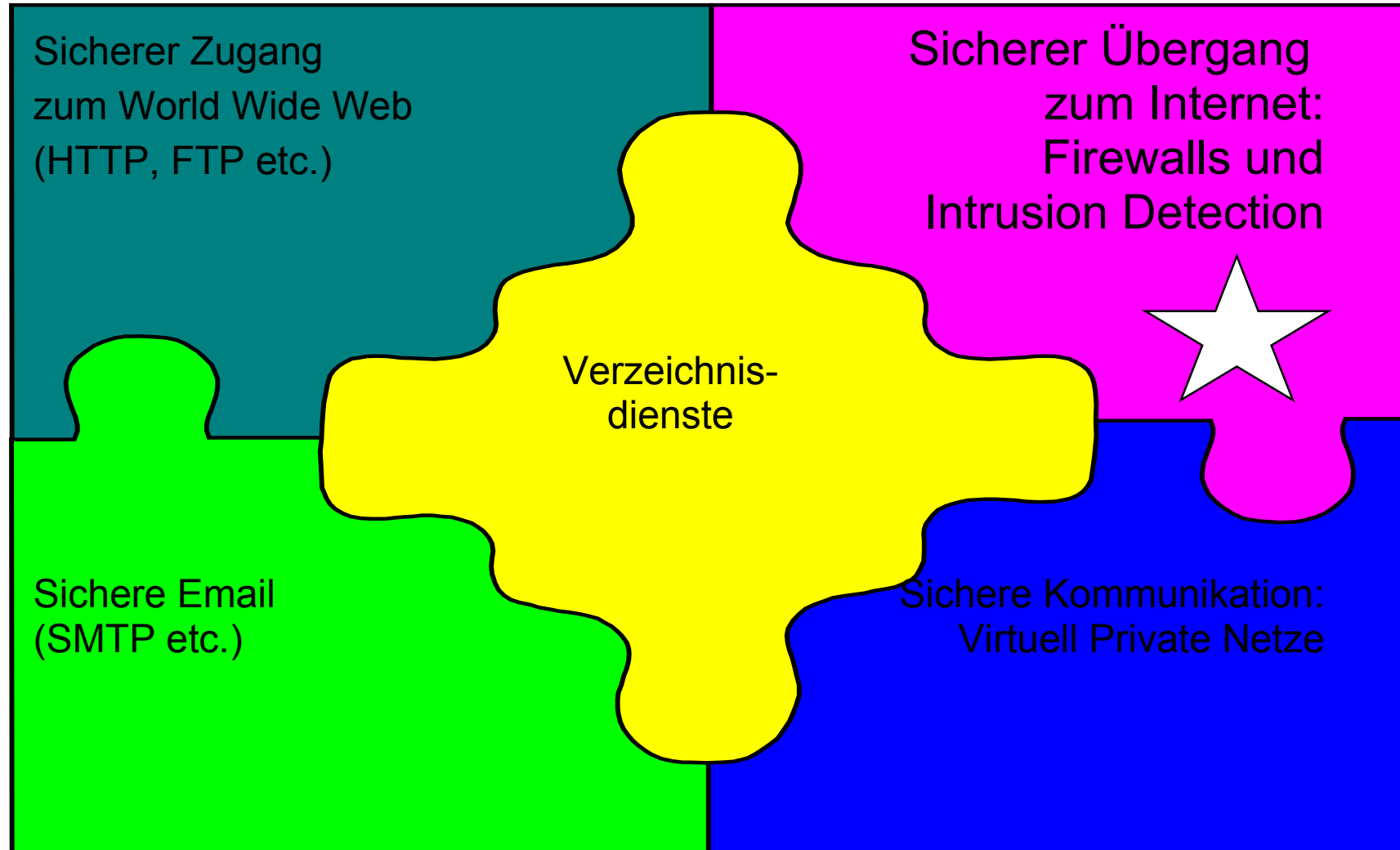


Sicherheitsdienste für große Firmen

=> Teil 2: Firewalls

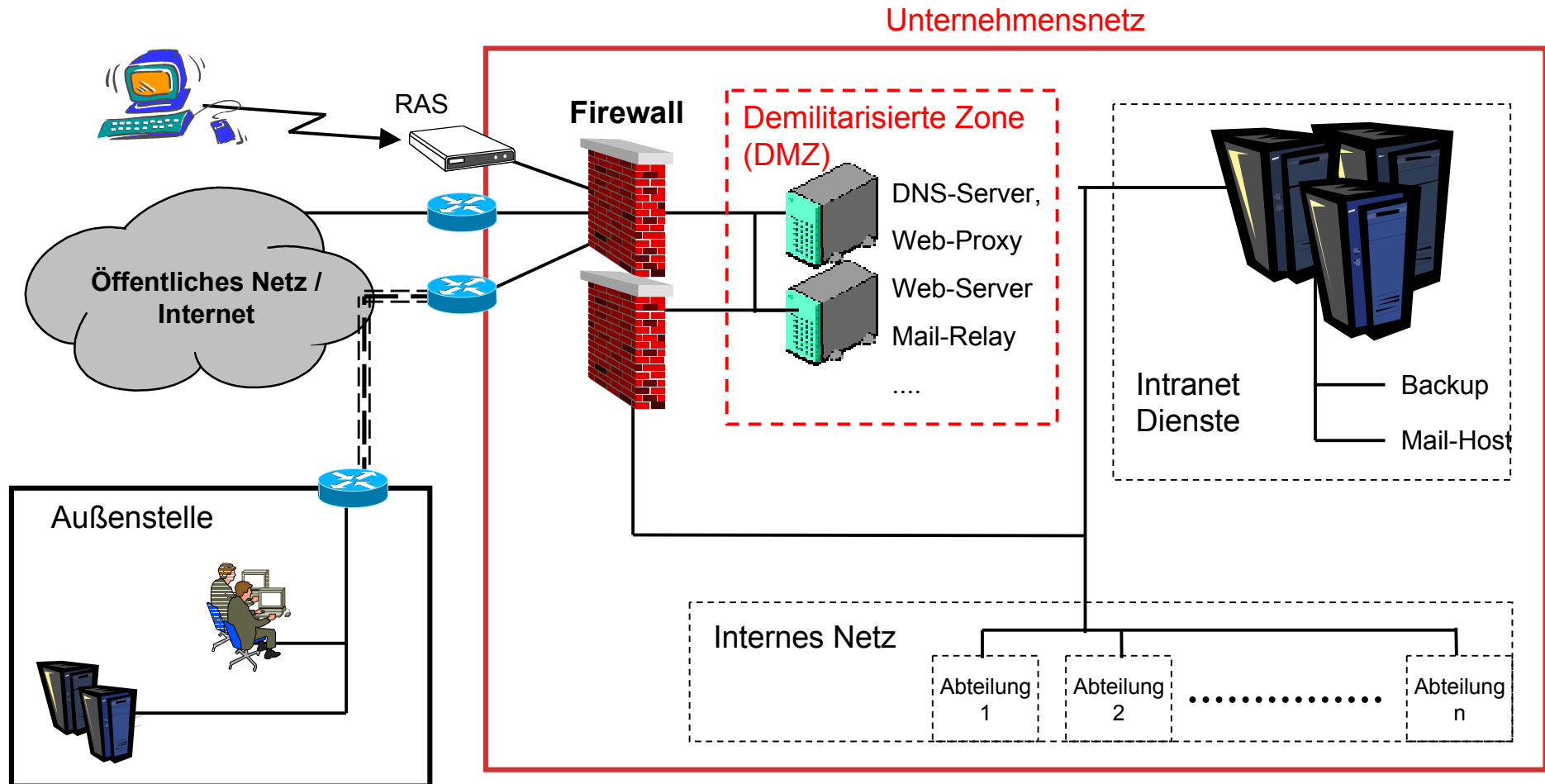


Firewalls

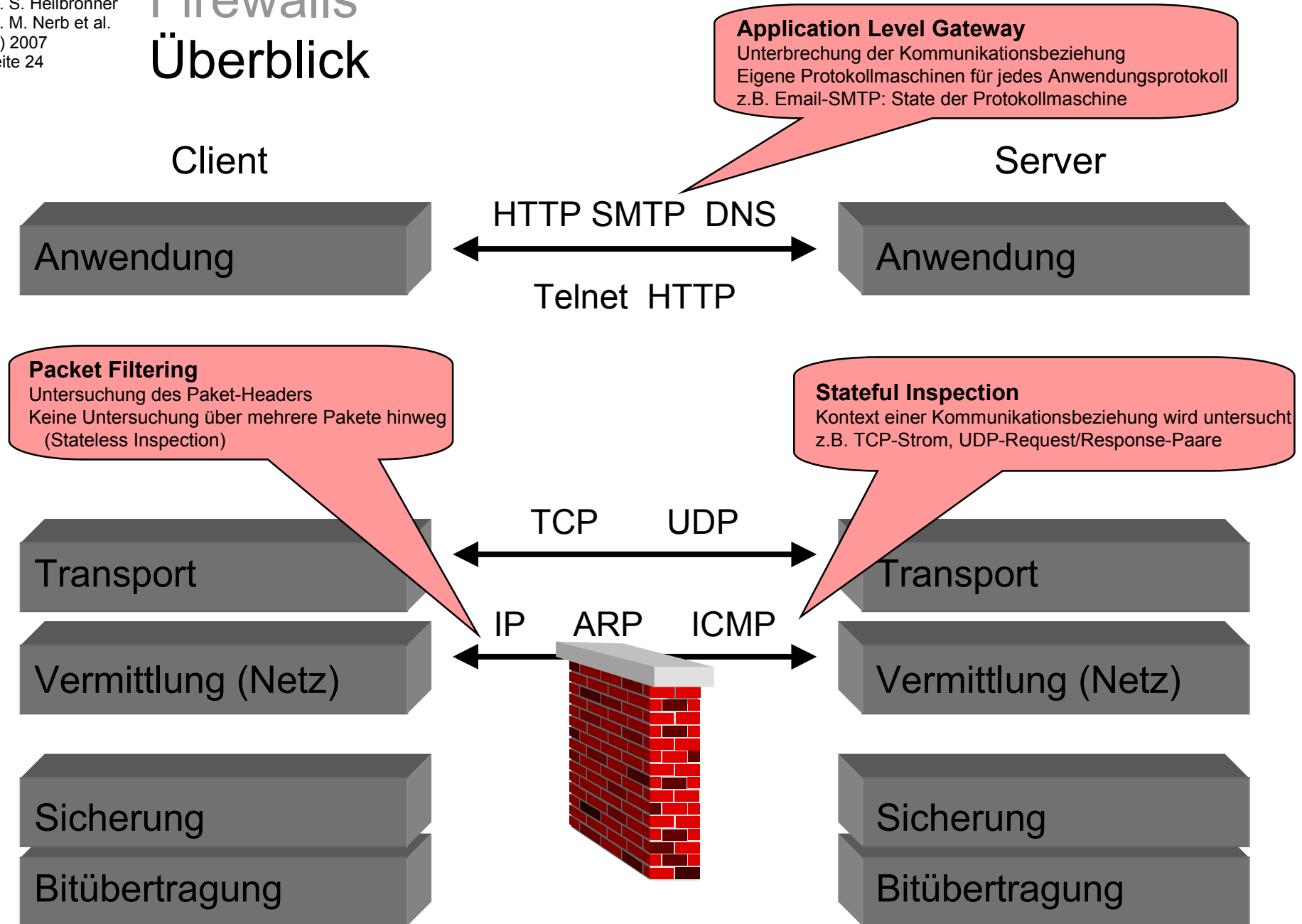
Einsatzzweck

- „*A Firewall helps you to keep **unauthorized** users from accessing your network **resources**.* „
 - Zugriffsrechteverwaltung für Kommunikationsbeziehungen (*Access Control Policy*)
- Grundprinzip:
 - Alles ist (zunächst) prinzipiell gesperrt.
 - Kommunikationsbeziehungen werden einzeln erlaubt.
- => ALLE Bereiche des Netzzugangs werden tangiert!
- Festlegung der Konfiguration in großen IT-Infrastrukturen
 - Iterativer Prozeß in Abstimmung mit vielen Beteiligten
 - Unterliegt ständigem „Change Management“
 - Umgehung durch Tunnelling vermeiden

Internet-Übergang Architektur

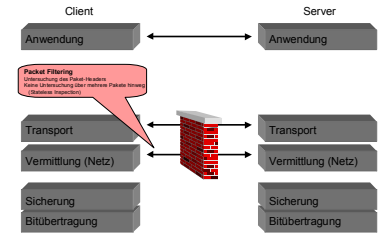


Firewalls Überblick



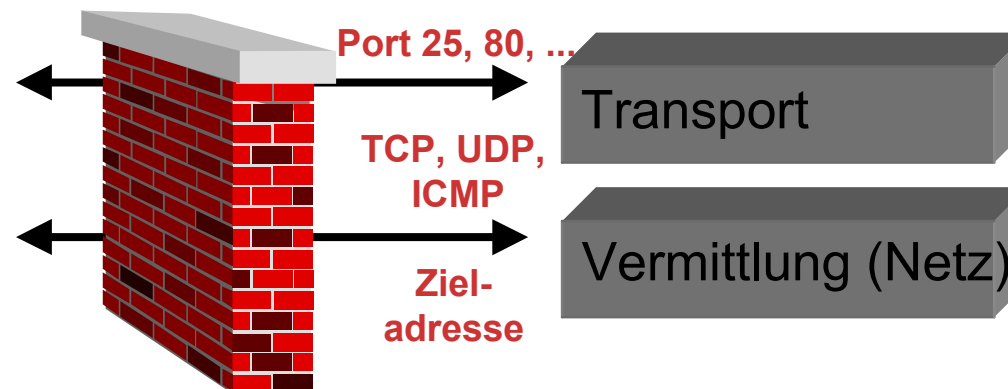
Firewalls

Packet Filtering

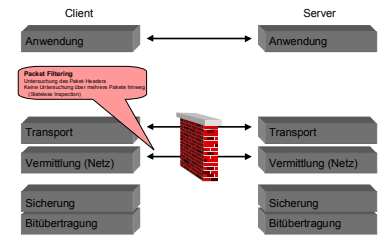


■ Filterung erfolgt nach Bitmustern im Paket-Header, z.B.

- IP-Absenderadresse, IP-Zieladresse
- Protokolltyp: TCP / UDP / ICMP
- Portnummern als Indiz für Dienst, z.B.
 - Port 80/TCP und 44s/TCP für HTTP /HTTPS
 - Port 25/TCP für SMTP (Email)
 - Port 22/TCP für SSH
 - Port 53/UDP für DNS



Packet Filtering Bewertung



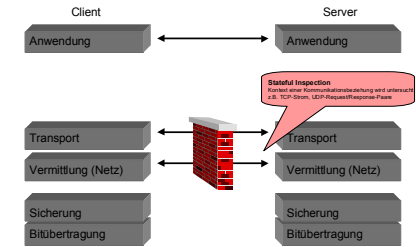
■ Vorteile

- transparent, keine spezielle Anpassung im Netzwerk nötig
- flexibel, jedes gängige Client/Server-Protokoll wird unterstützt
- geringe Kosten
- hoher Durchsatz, in Routern hoch-performant implementierbar

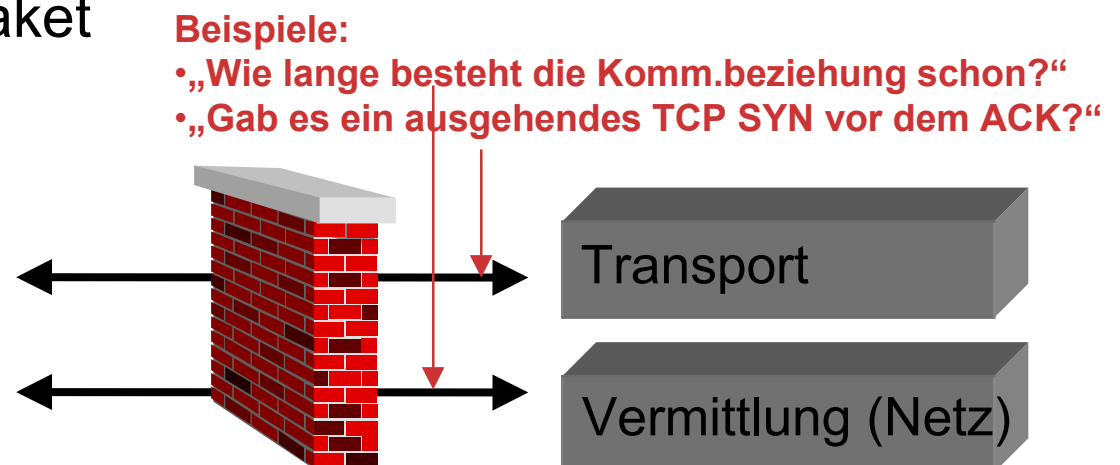
■ Nachteile

- Regelsätze starr und schwer zu verwalten
- unzureichende Authentifizierung (IP-Adresse nicht verifizierbar)
- Gefälschte Information in Anwendungsprotokollen (z.B. Mail-Header) können in das interne Netz gelangen.

Stateful Inspection Überblick



- Auch: „Stateful Inspection, Smart Filtering, Adaptive Screening“
- Zustände der „Verbindungen“ werden analysiert, z.B.
 - Verbindungsauf- und abbau
 - Dauer der Verbindung
- Verwendung zusätzlich zum Packet Filtering
- Dynamische Reaktion des Filters wird realisiert, z.B.:
 - Datenpakete werden nur für etablierte Verbindung akzeptiert.
 - Ausgehendes UDP-Paket öffnet ein Zeitfenster für nachfolgende Antwortpakete.
- Beste „einfache“ Lösung



**Aufbau
 TCP-Verb.**

The screenshot shows the Wireshark interface with a captured network packet. The packet list pane at the top shows a SYN packet (No. 1) from leia.muclab.de to 212.184.6.57 on port 80. The packet details pane below shows the structure of the packet:

- Frame 1 (74 on wire, 74 captured)
- Ethernet II
- Internet Protocol
 - Version: 4
 - Header length: 20 bytes
 - Differentiated Services Field: 0x00 (DSCP 0x00; Default; ECN: 0x00)
 - Total Length: 60
 - Identification: 0x30a6
 - Flags: 0x04
 - Fragment offset: 0
 - Time to live: 64
 - Protocol: TCP (0x06)
 - Header checksum: 0x2ba4 (correct)
 - Source: leia.muclab.de (62.157.196.227)
 - Destination: 212.184.6.57 (212.184.6.57)
- Transmission Control Protocol, Src Port: 4571 (4571), Dst Port: http (80), Seq: 4180447360, Ack: 0
 - Source port: 4571 (4571)
 - Destination port: http (80)
 - Sequence number: 4180447360
 - Header length: 40 bytes
 - Flags: 0x0002 (SYN)
 - Window size: 32120

The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII:

```

0010  00 3c 30 a6 40 00 40 06 2b a4 3e 9d c4 e3 d4 b8  .<.@.@.+. ....
0020  06 39 11 db 00 50 f9 2c 90 80 00 00 00 00 a0 02  .9..LP. ....
0030  7d 78 26 5d 00 00 02 04 05 b4 04 02 08 0a 00 f5  }x&]. ....
0040  28 f2 00 00 00 00 01 03 03 00                    (. ....
    
```

The filter bar at the bottom shows the filter: `Destination Port (tcp.dstport)`.

**Aufbau
 TCP-Verb.
 HTTP-
 Anfrage**

The screenshot shows the Wireshark interface with a packet capture of an HTTP GET request. The packet list pane at the top shows several packets, with packet 4 selected. The packet details pane below shows the structure of the selected packet, including Ethernet II, Internet Protocol, Transmission Control Protocol (TCP), and Hypertext Transfer Protocol (HTTP). The HTTP pane shows the request line and various headers. The packet bytes pane at the bottom shows the raw data of the selected packet in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	leia.muclab.de	212.184.6.57	TCP	4571 > http [SYN] Seq=4180447360 Ack=0 Win=32120 Len=0
2	0.046438	212.184.6.57	leia.muclab.de	TCP	http > 4571 [SYN, ACK] Seq=3393461787 Ack=4180447361 W:
3	0.046496	leia.muclab.de	212.184.6.57	TCP	4571 > http [ACK] Seq=4180447361 Ack=3393461788 Win=32:
4	0.048765	leia.muclab.de	212.184.6.57	HTTP	GET / HTTP/1.0
5	0.156148	212.184.6.57	leia.muclab.de	TCP	http > 4571 [ACK] Seq=3393461788 Ack=4180447787 Win=10:
6	0.321248	212.184.6.57	leia.muclab.de	HTTP	HTTP/1.1 200 OK
7	0.321289	leia.muclab.de	212.184.6.57	TCP	4571 > http [ACK] Seq=4180447787 Ack=3393462796 Win=32:

Frame 4 (492 on wire, 492 captured)

- Ethernet II
- Internet Protocol
- Transmission Control Protocol, Src Port: 4571 (4571), Dst Port: http (80), Seq: 4180447361, Ack: 3393461788
 - Source port: 4571 (4571)
 - Destination port: http (80)
 - Sequence number: 4180447361
 - Next sequence number: 4180447787
 - Acknowledgement number: 3393461788
 - Header length: 32 bytes
 - Flags: 0x0018 (PSH, ACK)
 - Window size: 32120
 - Checksum: 0xa905 (correct)
 - Options: (12 bytes)
- Hypertext Transfer Protocol
 - GET / HTTP/1.0\r\n
 - User-Agent: Mozilla/5.0 (X11; U; Linux 2.2.18 i686; en-US; rv:0.8.1+) Gecko/20010422\r\n
 - Accept: /*/*\r\n
 - Accept-Language: en, de; q=0.500\r\n
 - Accept-Encoding: gzip,deflate,compress,identity\r\n
 - Accept-Charset: ISO-8859-1, utf-8; q=0.667, *; q=0.667\r\n
 - Via: 1.1 leia.muclab.de:3128 (Squid/2.3.STABLE4-hno.CVS)\r\n

0040 87 df 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 30 ..GET / HTTP/1.0
 0050 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f ..User-Agent: Mo
 0060 7a 69 6c 6c 61 2f 35 2e 30 20 28 58 31 31 3b 20 zilla/5.0 (X11;
 0070 55 3b 20 4c 69 6e 75 78 20 32 2e 32 2e 31 38 20 U; Linux 2.2.18
 0080 69 36 38 36 3b 20 65 6e 2d 55 53 3b 20 72 76 3a i686; en-US; rv:

**Aufbau
 TCP-Verb.
 HTTP-
 Anfrage
 Antwort**

The screenshot shows the Wireshark interface with a packet capture of an HTTP response. The packet list pane at the top shows several packets, with packet 6 selected. The packet details pane shows the structure of the selected packet, including Ethernet II, Internet Protocol, Transmission Control Protocol (TCP), and Hypertext Transfer Protocol (HTTP). The HTTP pane shows the response status '200 OK' and various headers like 'Server: Netscape-Enterprise/4.0', 'Date: Mon, 30 Apr 2001 14:38:36 GMT', and 'Content-type: text/html'. The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII, with the ASCII column displaying the text of the response headers.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	leia.muclab.de	212.184.6.57	TCP	4571 > http [SYN] Seq=4180447360 Ack=0 Win=32120 Len=0
2	0.046438	212.184.6.57	leia.muclab.de	TCP	http > 4571 [SYN, ACK] Seq=3393461787 Ack=4180447361 W...
3	0.046496	leia.muclab.de	212.184.6.57	TCP	4571 > http [ACK] Seq=4180447361 Ack=3393461788 Win=32...
4	0.048765	leia.muclab.de	212.184.6.57	HTTP	GET / HTTP/1.0
5	0.156148	212.184.6.57	leia.muclab.de	TCP	http > 4571 [ACK] Seq=3393461788 Ack=4180447787 Win=10...
6	0.321248	212.184.6.57	leia.muclab.de	HTTP	HTTP/1.1 200 OK
7	0.321289	leia.muclab.de	212.184.6.57	TCP	4571 > http [ACK] Seq=4180447787 Ack=3393462796 Win=32...

Frame 6 (1074 on wire, 1074 captured)

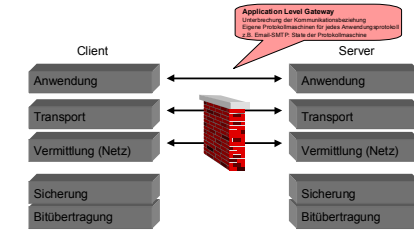
- Ethernet II
- Internet Protocol
- Transmission Control Protocol, Src Port: http (80), Dst Port: 4571 (4571), Seq: 3393461788, Ack: 4180447787
 - Source port: http (80)
 - Destination port: 4571 (4571)
 - Sequence number: 3393461788
 - Next sequence number: 3393462796
 - Acknowledgement number: 4180447787
 - Header length: 32 bytes
 - Flags: 0x0018 (PSH, ACK)
 - Window size: 10136
 - Checksum: 0xc5c7 (correct)
 - Options: (12 bytes)
- Hypertext Transfer Protocol
 - HTTP/1.1 200 OK\r\n
 - Server: Netscape-Enterprise/4.0\r\n
 - Date: Mon, 30 Apr 2001 14:38:36 GMT\r\n
 - Content-type: text/html\r\n
 - Connection: close\r\n
 - \r\n
 - Data (875 bytes)

0040 28 f6 48 54 54 50 2f 31 2e 31 20 32 30 30 20 4f (.HTTP/1.1 200 0
 0050 4b 0d 0a 53 65 72 76 65 72 3a 20 4e 65 74 73 63 k..Server: Netsc
 0060 61 70 65 2d 45 6e 74 65 72 70 72 69 73 65 2f 34 ape-Enterprise/4
 0070 2e 30 0d 0a 44 61 74 65 3a 20 4d 6f 6e 2c 20 33 .0..Date: Mon, 3
 0080 30 20 41 70 72 20 32 30 30 31 20 31 34 3a 33 38 0 Apr 20 01 14:38

Filter: / Reset Text (text)

Application Level Gateways

Überblick



- Eigene Protokollinstanz für jedes Anwendungsprotokoll
- Typische Manipulationen und Prüfungen
 - Entscheidung, ob Protokollschritte ausgeführt bzw. Daten übertragen werden.
 - Ist dieser Ablauf (Schritte/Inhalte) inhaltlich zulässig?
 - „Wird Email mit diesem Inhalt von diesem Mail-Relay akzeptiert?“
 - Einhaltung des Protokolls
 - Hat sich die andere Protokollinstanz „korrekt“ verhalten?
 - Port-Umsetzung
 - Anonymisierung des Verkehrs

Vergleich

Paketfilter

- i.a. nur Daten der Vermittlungs-/Transport-Protokolle werden geprüft und gefiltert
- hohe Performanz, da nicht bis Anwendungsebene geprüft wird
- Regeln werden statisch definiert
- niedrigeres Sicherheitsniveau

Application Level Gateway

- Anwendungsdaten und -Protokolle werden geprüft und gefiltert
- geringere Performanz, da aufwendige und tiefgreifende Prüfung und Filterung
- Regelwerk kann dynamisch und flexibel angepasst werden
- hohes Sicherheitsniveau

Rückblick auf Firewalls

- Trennung und Filterung des internen / externen Netzverkehrs

- Vorteile heutiger Firewalls (bzw. deren Implementierung)
 - einfache Regelung des Netzwerkverkehrs
 - Unterstützung und Prüfung aller wichtigen Protokolle: IP, UDP, TCP, Anwendungsprotokolle
 - Verbergen der internen Netzstruktur (durch „NAT“)

- Nachteile von Firewalls in großen Netzen
 - Regelwerk schnell unübersichtlich
 - häufige Konfigurationsänderungen notwendig
 - Komplexes Change Management
 - bei grossem Netzwerk potentieller Engpaß