Terminal Emulation (TELNET)

Prof. Dr. Norbert Pohlmann
Fachbereich Informatik
Verteilte Systeme und Informationssicherheit

Fachhochschule Gelsenkirchen

Inhalt

- Ziele und Einordnung
- Übersicht
- Telnet Protokoll
- Protokollmitschnitt
- Zusammenfassung

© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Inhalt

- **Ziele und Einordnung**
  - Übersicht
  - Telnet Protokoll
  - Protokollmitschnitt
  - Zusammenfassung

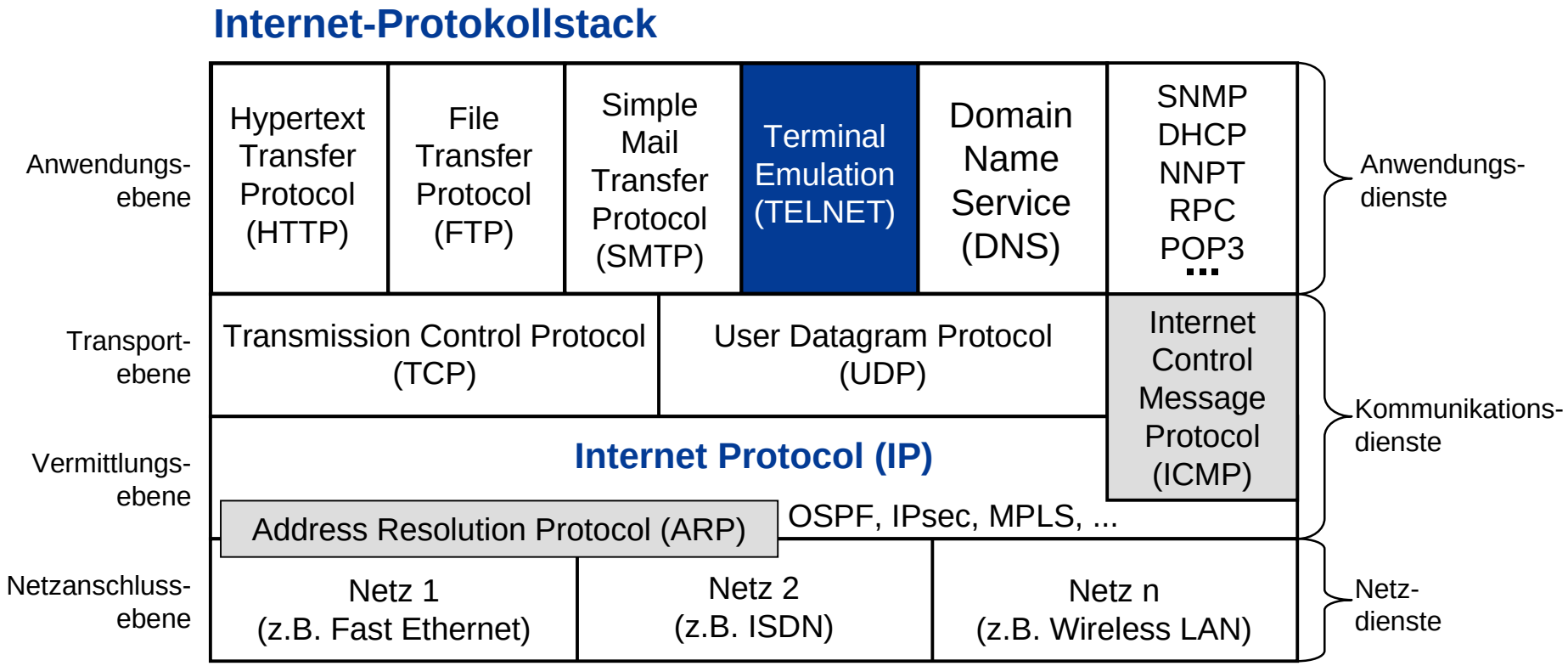
© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Terminal Emulation (TELNET)
→ Ziele

- Gutes Verständnis für das TELNET Protokoll.
- Erlangen der Kenntnisse über die Aufgaben, Prinzipien und Mechanismen des TELNET Protokolls.
- Gewinnen von praktischen Erfahrungen über das TELNET Protokoll mit Hilfe von Protokollanalysen.

© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Die Anwendungsebene
→ Terminal Emulation (TELNET) - Einordnung



© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Inhalt

- Ziele und Einordnung
- **Übersicht**
- Telnet Protokoll
- Protokollmitschnitt
- Zusammenfassung

© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Terminal Emulation (TELNET)
→ Standards und Literatur

RFC 854	Telnet Protocol Specification
RFC 855	Telnet Option Specification
RFC 856	Telnet Binary Transmission
RFC 857	Telnet Echo Option
RFC 858	Telnet Suppress Go Ahead Option
RFC 859	Telnet Status Option
RFC 860	Telnet Timing Mark Option
RFC 861	Telnet Extended Options - List Option
RFC 884	Telnet Terminal Type Option
...	

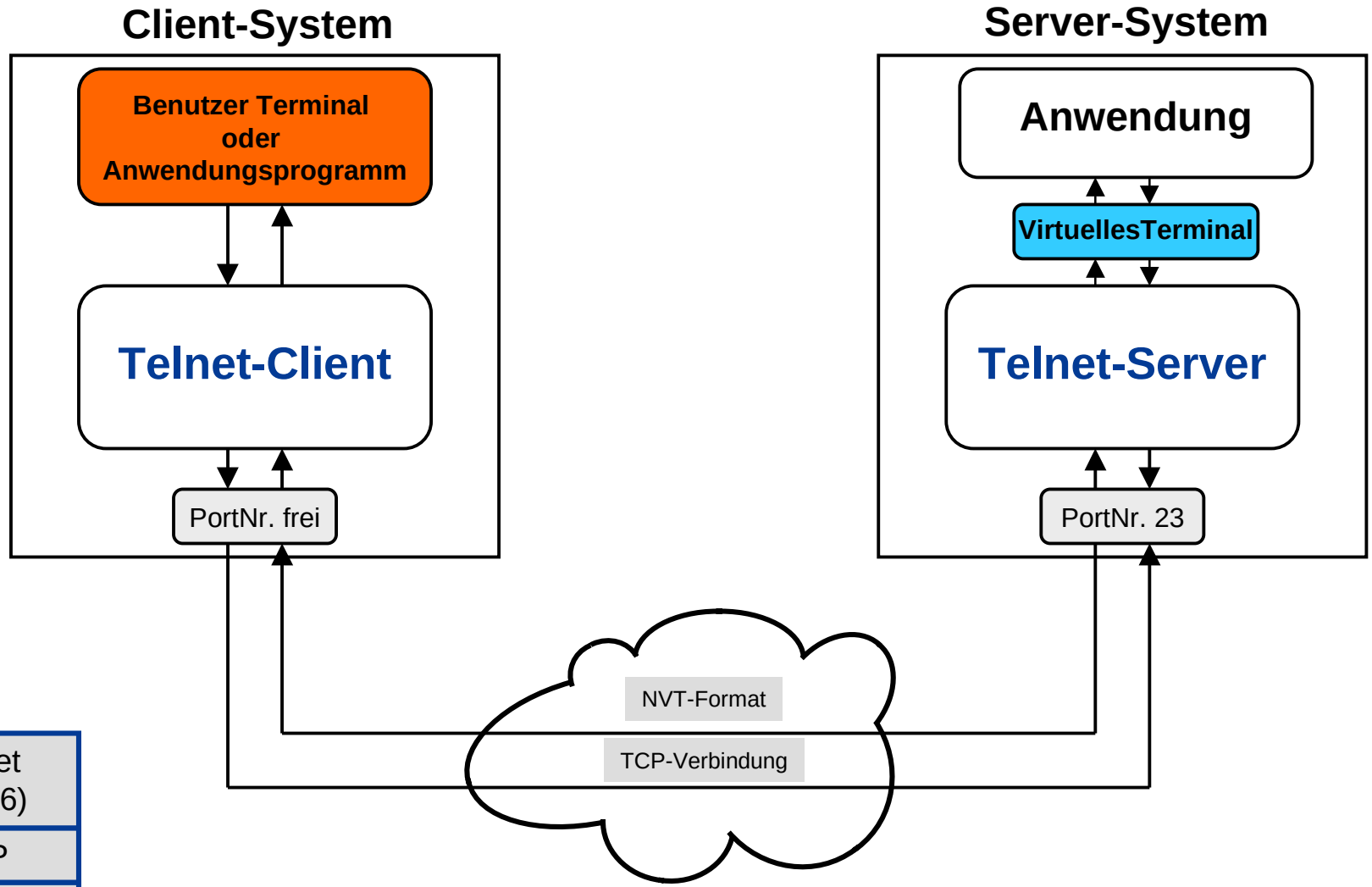
© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

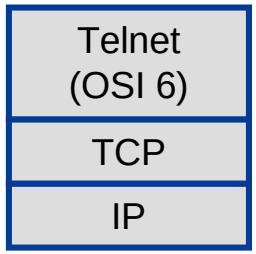
Telnet (Terminal Emulation)

- Telnet erlaubt dem Benutzer (kann auch ein Anwendungsprogramm sein), eine TCP-Verbindung zu einem entfernten Server-System herzustellen.
- Dabei werden Eingabedaten vom Telnet-Client (Terminal) direkt zum entfernten Rechnersystem gesandt und in Gegenrichtung Ausgaben vom entfernten Server-System zurück an den Telnet-Client geleitet, so als sei dieses **ein lokales Terminal des entfernten Server-Systems**.
- Dazu ist es erforderlich, dass das Betriebssystem des Server-Systems eine als „virtuelles Terminal“ bezeichnete Schnittstelle unterstützt.
- Das virtuelle Terminal gestattet, von einem Programm aus, Zeichen einzuschleusen als ob sie von einem realen Terminal kämen, und umgekehrt für ein Terminal bestimmte Ausgaben zu übernehmen.

© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Telnet (Terminal Emulation)
→ Client-Server Beziehung





© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Dienste von Telnet

- **Telnet** setzt auf dem gesicherten Transport Service von TCP auf.
- Dazu wird typischerweise eine Verbindung zu **Port 23** aufgebaut.
- Mit **Telnet** kann aber auch eine Verbindungsherstellung zu einem anderen Port durchgeführt werden.
- Außerdem bietet **Telnet** den Remote Login-Dienst.
- Bei **Telnet** wirken drei Funktionsgruppen zusammen:
  - Network Virtuell Terminal (NVT)
  - Telnet-Kommandos
  - Optionen

© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Inhalt

- Ziele und Einordnung
- Übersicht
- **Telnet Protokoll**
- Protokollmitschnitt
- Zusammenfassung

© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Telnet-Protokoll

- Telnet arbeitet **nicht** kommandoorientiert, wie die anderen Protokolle (FTP, SMTP, HTTP).
- Im Prinzip ist **Telnet** nur ein bidirektionaler Austausch von ASCII-Zeichen zwischen Client und Server beliebiger Plattformen über TCP.
- Zu verwendende Zeichensätze, vor allem Steuercodes, sind standardisiert - NVT (Network Virtual Terminal).
- Bei **Telnet** werden die Einstellungen und Optionen über Steuercodes geregelt.
- **Telnet** ist ein symmetrisches Protokoll.

© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

ASCII-Übertragung

- TCP überträgt bei **Telnet** die Daten Byte-weise (8-Bit)
- **Telnet** stellt Anwendungen jedoch nur 7-Bit zur Verfügung **(NVT - Code 0-127)**
- Codes 128-255 sind für Steuercodes reserviert (bisher belegt: 240-254)
- Jedem Steuercode geht der Code 255 (0xFF) **(IAC-Zeichen - Interpret As Command)** voraus

© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Telnet (Terminal Emulation)
→ Steuercodes

Code	Bedeutung
242 - Data Mark	*Out-Of-Band* Signal
244 - Interrupt	Ctrl-C
246 - Are You There	Ist Client bzw. Server noch da?
251 - WILL (0xFB)	Anfrage / Bestätigung für eine Option
252 - WON`T (0xFC)	Option wird abgelehnt
253 - DO (0xFD)	Aufforderung f. Gegens. eine Option zu nutzen
254 - DON`T (0xFE)	Ablehnung, dass Gegenseite Option nutzen darf
240 - SE (0xF0)	End of Subnegotiation
250 - SB (0xFA)	Begin of Subnegotiation

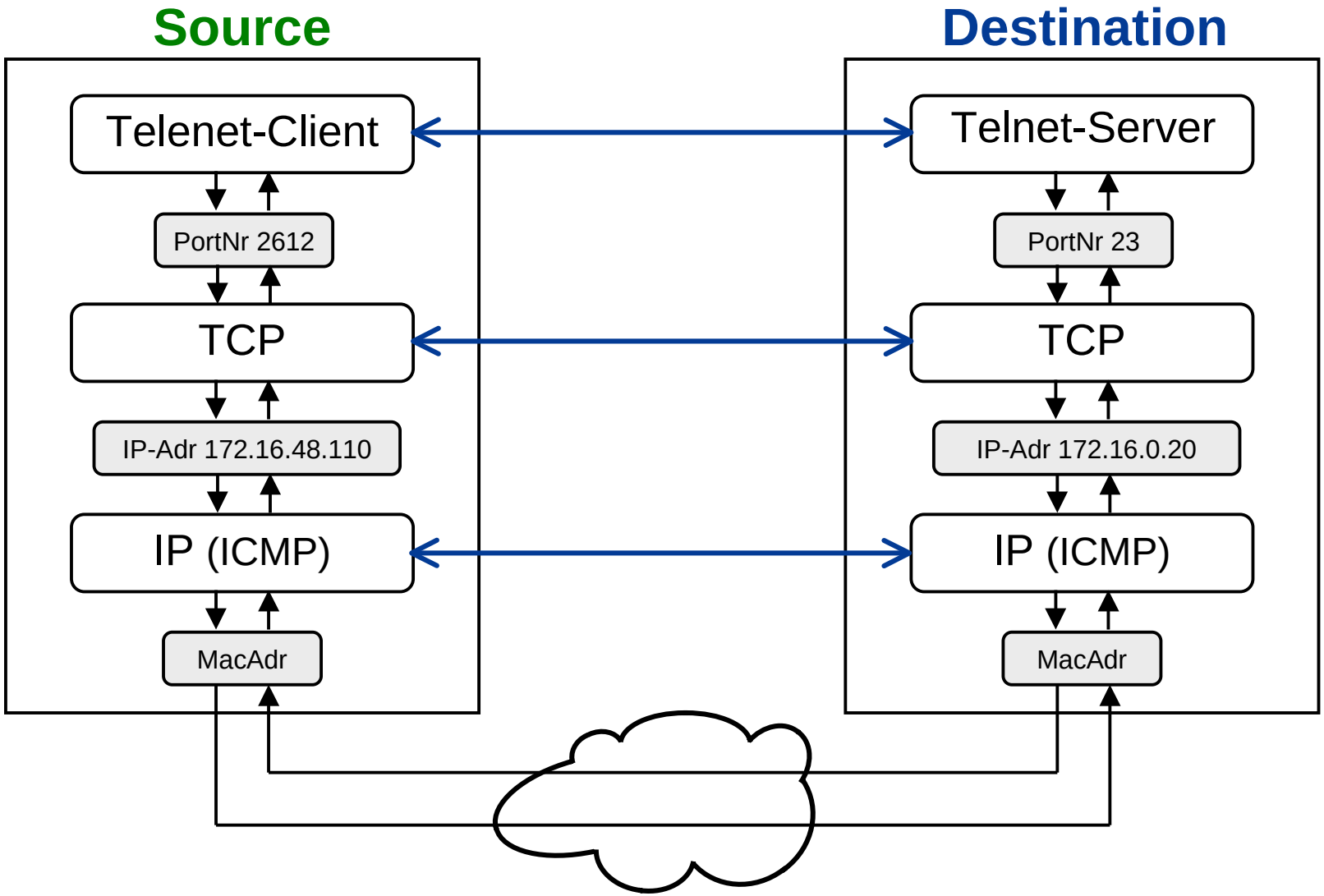
© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Inhalt

- Ziele und Einordnung
- Übersicht
- Telnet Protokoll
- **Protokollmitschnitt**
- Zusammenfassung

© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Telnet (Terminal Emulation)
→ Beispiel



© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Telnet (Terminal Emulation)
→ Beispiel

No.	Time	Source	Destination	Protocol	Info
1	0.000000	172.16.48.110	172.16.0.20	TCP	2612 > telnet [SYN] Seq=468624411 Ack=0 Win=64240 Len=0
2	0.001453	172.16.0.20	172.16.48.110	TCP	telnet > 2612 [SYN, ACK] Seq=716121853 Ack=468624412 Win=16368 Len=
3	0.001476	172.16.48.110	172.16.0.20	TCP	2612 > telnet [ACK] Seq=468624412 Ack=716121854 Win=64240 Len=0
4	0.058864	172.16.0.20	172.16.48.110	TCP	21621 > auth [SYN] Seq=683832601 Ack=0 Win=512 Len=0
5	0.058917	172.16.48.110	172.16.0.20	TCP	auth > 21621 [RST, ACK] Seq=0 Ack=683832602 Win=0 Len=0
6	0.119423	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
7	0.119675	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
8	0.121105	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
9	0.121137	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
10	0.122321	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
11	0.122354	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
12	0.141304	172.16.0.20	172.16.48.110	TCP	telnet > 2612 [ACK] Seq=716121881 Ack=468624436 Win=16368 Len=0
13	0.141357	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
14	0.148839	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
15	0.149048	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
16	0.161231	172.16.0.20	172.16.48.110	TCP	telnet > 2612 [ACK] Seq=716121893 Ack=468624455 Win=16365 Len=0
17	0.161254	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
18	0.173637	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
19	0.173827	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
20	0.191407	172.16.0.20	172.16.48.110	TCP	telnet > 2612 [ACK] Seq=716121952 Ack=468624467 Win=16365 Len=0
21	0.191469	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
22	0.194798	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
23	0.381495	172.16.48.110	172.16.0.20	TCP	2612 > telnet [ACK] Seq=468624470 Ack=716121966 Win=64128 Len=0
24	1.540922	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
25	1.542830	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
26	1.693983	172.16.48.110	172.16.0.20	TCP	2612 > telnet [ACK] Seq=468624471 Ack=716121967 Win=64127 Len=0
27	1.775526	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
28	1.777027	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
29	1.912724	172.16.48.110	172.16.0.20	TCP	2612 > telnet [ACK] Seq=468624472 Ack=716121968 Win=64126 Len=0

TCP Verbindungsaufbau vom Client zum Server

© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Telnet (Terminal Emulation)
→ Beispiel

No.	Time	Source	Destination	Protocol	Info
30	2.043137	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
31	2.044543	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
32	2.131477	172.16.48.110	172.16.0.20	TCP	2612 > telnet [ACK] Seq=468624473 Ack=716121969 Win=64125 Len=0
33	2.284790	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
34	2.286280	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
35	2.459584	172.16.48.110	172.16.0.20	TCP	2612 > telnet [ACK] Seq=468624474 Ack=716121970 Win=64124 Len=0
36	2.468374	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
37	2.469893	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
38	2.678340	172.16.48.110	172.16.0.20	TCP	2612 > telnet [ACK] Seq=468624475 Ack=716121971 Win=64123 Len=0
39	2.790170	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
40	2.791642	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
41	3.006478	172.16.48.110	172.16.0.20	TCP	2612 > telnet [ACK] Seq=468624476 Ack=716121972 Win=64122 Len=0
42	3.024536	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
43	3.026083	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
44	3.225203	172.16.48.110	172.16.0.20	TCP	2612 > telnet [ACK] Seq=468624477 Ack=716121973 Win=64121 Len=0
45	3.468904	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
46	3.470402	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
47	3.662697	172.16.48.110	172.16.0.20	TCP	2612 > telnet [ACK] Seq=468624478 Ack=716121975 Win=64119 Len=0
48	3.663747	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
49	3.881435	172.16.48.110	172.16.0.20	TCP	2612 > telnet [ACK] Seq=468624478 Ack=716121985 Win=64109 Len=0
50	4.610728	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
51	4.631189	172.16.0.20	172.16.48.110	TCP	telnet > 2612 [ACK] Seq=716121985 Ack=468624479 Win=16368 Len=0
52	4.824572	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
53	4.841309	172.16.0.20	172.16.48.110	TCP	telnet > 2612 [ACK] Seq=716121985 Ack=468624480 Win=16368 Len=0
54	5.070198	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
55	5.091324	172.16.0.20	172.16.48.110	TCP	telnet > 2612 [ACK] Seq=716121985 Ack=468624481 Win=16368 Len=0
56	5.432697	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
57	5.451243	172.16.0.20	172.16.48.110	TCP	telnet > 2612 [ACK] Seq=716121985 Ack=468624482 Win=16368 Len=0
58	5.818940	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...

© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Telnet (Terminal Emulation)
→ Beispiel

No.	Time	Source	Destination	Protocol	Info
59	5.820411	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
60	5.959553	172.16.48.110	172.16.0.20	TCP	2612 > telnet [ACK] Seq=468624483 Ack=716121987 Win=64107 Len=0
61	6.060218	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
62	6.178290	172.16.48.110	172.16.0.20	TCP	2612 > telnet [ACK] Seq=468624483 Ack=716122009 Win=64085 Len=0
63	6.179855	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
64	6.397041	172.16.48.110	172.16.0.20	TCP	2612 > telnet [ACK] Seq=468624483 Ack=716122140 Win=63954 Len=0
65	7.283228	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
66	7.490773	172.16.48.110	172.16.0.20	TCP	2612 > telnet [ACK] Seq=468624483 Ack=716122169 Win=63925 Len=0
67	8.640316	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
68	8.642451	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
69	8.803254	172.16.48.110	172.16.0.20	TCP	2612 > telnet [ACK] Seq=468624484 Ack=716122170 Win=63924 Len=0
70	8.985947	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
71	8.987791	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
72	9.131374	172.16.48.110	172.16.0.20	TCP	2612 > telnet [ACK] Seq=468624485 Ack=716122171 Win=63923 Len=0
73	9.231843	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
74	9.233662	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
75	9.350123	172.16.48.110	172.16.0.20	TCP	2612 > telnet [ACK] Seq=468624486 Ack=716122172 Win=63922 Len=0
76	9.503643	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
77	9.505555	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
78	9.678228	172.16.48.110	172.16.0.20	TCP	2612 > telnet [ACK] Seq=468624487 Ack=716122173 Win=63921 Len=0
79	10.193237	172.16.48.110	172.16.0.20	TELNET	Telnet Data ...
80	10.195276	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
81	10.202814	172.16.0.20	172.16.48.110	TELNET	Telnet Data ...
82	10.202878	172.16.48.110	172.16.0.20	TCP	2612 > telnet [ACK] Seq=468624488 Ack=716122183 Win=63911 Len=0
83	10.202896	172.16.0.20	172.16.48.110	TCP	telnet > 2612 [FIN, ACK] Seq=716122183 Ack=468624488 Win=16368 Len=0
84	10.202915	172.16.48.110	172.16.0.20	TCP	2612 > telnet [ACK] Seq=468624488 Ack=716122184 Win=63911 Len=0
85	10.203552	172.16.48.110	172.16.0.20	TCP	2612 > telnet [FIN, ACK] Seq=468624488 Ack=716122184 Win=63911 Len=0
86	10.204500	172.16.0.20	172.16.48.110	TCP	telnet > 2612 [ACK] Seq=716122184 Ack=468624489 Win=16367 Len=0

TCP Verbindungsab

© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Telnet (Terminal Emulation)
→ Beispiel: Aushandlung der Optionen (1/2)

Frame 6 (66 bytes on wire, 66 bytes captured) ← (Server)
Telnet

Command: Do Terminal Type	**FF FD 18**
Command: Do Terminal Speed	**FF FD 20**
Command: Do X Display Location	**FF FD 23**
Command: Do New Environment Option	**FF FD 27**

Frame 7 (60 bytes on wire, 60 bytes captured) → (Client)
Telnet

Command: Will Terminal Type	**FF FB 18**
Command: Will Negotiate About Window Size	**FF FB 1F**

Frame 8 (60 bytes on wire, 60 bytes captured) ← (Server)
Telnet

Command: Do Negotiate About Window Size

Frame 9 (63 bytes on wire, 63 bytes captured) → (Client)
Telnet

Command: Won't Terminal Speed	**FF FC 20**
Command: Won't X Display Location	**FF FC 23**
Command: Will New Environment Option	**FF FB 27**

© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Telnet (Terminal Emulation)
→ Beispiel: Aushandlung der Optionen (2/2)

Frame 10 (66 bytes on wire, 66 bytes captured) ← (Server)
Telnet
  Suboption Begin: New Environment Option
    Option data
  Command: Suboption End
  Suboption Begin: Terminal Type
    Send your Terminal Type
  Command: Suboption End

Frame 11 (63 bytes on wire, 63 bytes captured) → (Client)
Telnet
  Suboption Begin: Negotiate About Window Size
    Width: 80
    Height: 25
  Command: Suboption End

Frame 13 (70 bytes on wire, 70 bytes captured) → (Client)
Telnet
  Suboption Begin: New Environment Option
    Option data
  Command: Suboption End
  Suboption Begin: Terminal Type
    Here's my Terminal Type
    Value: ANSI
  Command: Suboption End

Frame 14 (66 bytes on wire, 66 bytes captured) ← (Server)
Telnet
  Command: Will Suppress Go Ahead
  Command: Do Echo
  Command: Will Status
  Command: Do Remote Flow Control

Frame 15 (57 bytes on wire, 57 bytes captured) → (Client)
Telnet
  Command: Do Suppress Go Ahead

© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Telnet (Terminal Emulation)
→ Beispiel

Frame 17 (63 bytes on wire, 63 bytes captured) → (Client)
Telnet
Command: Will Echo
Command: Don't Status
Command: Won't Remote Flow Control

Frame 18 (113 bytes on wire, 113 bytes captured) ← (Server)
Telnet
Command: Don't Echo
Command: Will Echo
Data: \r\n
Data: Linux 2.0.33 (mail.informatik.fh-ge.de) (ttyp0)\r\n
Data: \r\n

Frame 19 (57 bytes on wire, 57 bytes captured) → (Client)
Telnet
Command: Won't Echo

Frame 21 (57 bytes on wire, 57 bytes captured) → (Client)
Telnet
Command: Do Echo

Frame 22 (68 bytes on wire, 68 bytes captured) ← (Server)
Telnet
Data: \r\n
Data: mail login:

© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Telnet (Terminal Emulation)
→ Beispiel: Eingabe LoginID

Frame 24 (55 bytes on wire, 55 bytes captured) → (Client)
Telnet
  Data: s

Frame 25 (60 bytes on wire, 60 bytes captured) ← (Server)
Telnet
  Data: s

Frame 27 (55 bytes on wire, 55 bytes captured) → (Client)
Telnet
  Data: t

Frame 28 (60 bytes on wire, 60 bytes captured) ← (Server)
Telnet
  Data: t

Frame 30 (55 bytes on wire, 55 bytes captured) → (Client)
Telnet
  Data: u

Frame 31 (60 bytes on wire, 60 bytes captured) ← (Server)
Telnet
  Data: u

Frame 33 (55 bytes on wire, 55 bytes captured) → (Client)
Telnet
  Data: d

Frame 34 (60 bytes on wire, 60 bytes captured) ← (Server)
Telnet
  Data: d

Frame 36 (55 bytes on wire, 55 bytes captured) → (Client)
Telnet
  Data: e

Frame 37 (60 bytes on wire, 60 bytes captured) ← (Server)
Telnet
  Data: e

Frame 39 (55 bytes on wire, 55 bytes captured) → (Client)
Telnet
  Data: n

Frame 40 (60 bytes on wire, 60 bytes captured) ← (Server)
Telnet
  Data: n

Frame 42 (55 bytes on wire, 55 bytes captured) → (Client)
Telnet
  Data: t

Frame 43 (60 bytes on wire, 60 bytes captured) ← (Server)
Telnet
  Data: t

Frame 45 (55 bytes on wire, 55 bytes captured) → (Client)
Telnet
  Data: \r

Frame 46 (60 bytes on wire, 60 bytes captured) ← (Server)
Telnet
  Data: \r\n

LoginID = student

© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Telnet (Terminal Emulation)
→ Beispiel: Eingabe Password

Frame 48 (64 bytes on wire, 64 bytes captured) ← (Server)
Telnet
Data: Password:

Frame 50 (55 bytes on wire, 55 bytes captured) → (Client)
Telnet
Data: m

Frame 52 (55 bytes on wire, 55 bytes captured) → (Client)
Telnet
Data: o

Frame 54 (55 bytes on wire, 55 bytes captured) → (Client)
Telnet
Data: n

Frame 56 (55 bytes on wire, 55 bytes captured) → (Client)
Telnet
Data: d

Frame 58 (55 bytes on wire, 55 bytes captured) → (Client)
Telnet
Data: \r

Frame 59 (60 bytes on wire, 60 bytes captured) ← (Server)
Telnet
Data: \r\n

Password = mond

© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Telnet (Terminal Emulation)
→ Beispiel

Frame 61 (76 bytes on wire, 76 bytes captured) ← (Server)
Telnet
Data: Have a lot of fun...\r\n

Frame 63 (185 bytes on wire, 185 bytes captured) ← (Server)
Telnet
Data: 2 failures since last login. Last was 11:56:12 on ttyp0.\r\n
Data: Last login: Wed Dec 17 14:24:47 on ttyp3 from 172.16.49.154.\r\n
Data: No mail.\r\n

Frame 65 (83 bytes on wire, 83 bytes captured) ← (Server)
Telnet
Data: student@mail:/home/student >

Frame 67 (55 bytes on wire, 55 bytes captured) → (Client)
Telnet
Data: e
Frame 68 (60 bytes on wire, 60 bytes captured) ← (Server)
Telnet
Data: e

Frame 70 (55 bytes on wire, 55 bytes captured) → (Client)
Telnet
Data: x
Frame 71 (60 bytes on wire, 60 bytes captured) ← (Server)
Telnet
Data: x

© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Telnet (Terminal Emulation)
→ Beispiel

Frame 73 (55 bytes on wire, 55 bytes captured) → (Client)
Telnet
Data: i

Frame 74 (60 bytes on wire, 60 bytes captured) ← (Server)
Telnet
Data: i

Frame 76 (55 bytes on wire, 55 bytes captured) → (Client)
Telnet
Data: t

Frame 77 (60 bytes on wire, 60 bytes captured) ← (Server)
Telnet
Data: t

Frame 79 (55 bytes on wire, 55 bytes captured) → (Client)
Telnet
Data: \r

Frame 80 (60 bytes on wire, 60 bytes captured) ← (Server)
Telnet
Data: \r\n

Frame 81 (62 bytes on wire, 62 bytes captured) ← (Server)
Telnet
Data: logout\r\n

Eingabe = exit

© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Inhalt

- Ziele und Einordnung
- Übersicht
- Telnet Protokoll
- Protokollmitschnitt
- **Zusammenfassung**

© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Telnet (Terminal Emulation)
→ Zusammenfassung

- Das virtuelle Terminalprotokoll ermöglicht es dem Benutzer, sich von seinem Rechner aus an einem entfernten Rechner anzumelden und dort zu arbeiten.
- **Telnet** setzt auf dem gesicherten Transport Service von TCP auf.
- Dazu wird typischerweise eine Verbindung zu **Port 23** aufgebaut.
- Mit **Telnet** kann aber auch eine Verbindungsherstellung zu einem anderen Port durchgeführt werden.

© Prof. Dr. Norbert Pohlmann, Fachhochschule Gelsenkirchen

Terminal Emulation (TELNET)

Vielen Dank für Ihre Aufmerksamkeit

Fragen ?

norbert.pohlmann@informatik.fh-gelsenkirchen.de

Fachhochschule Gelsenkirchen