LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Rechnernetze und verteilte Systeme

Übungsblatt 5

Koenig.Noah@campus.lmu.de

Aktuelle Produktivnetze (H)

(a) Nennen Sie einen Vor- und Nachteil von Ringstrukturen in Produktivnetzen.

Topologien: https://de.wikipedia.org/wiki/Topologie_(Rechnernetz)

Vorteile:

- Kommunikation auch bei Ausfall eines Links / Knoten möglich
- Höherer Durchsatz, wenn beide Richtungen parallel genutzt werden

Nachteile:

- Zusätzliche Verbindung teuer → Kosten / Nutzen abwägen
- Beispiel Oettingenstraße: nur über eine Leitung verbunden

Aktuelle Produktivnetze

(b) Diskutieren Sie die Verwendung von Kupfer- gegenüber Glasfaserkabeln innerhalb von Gebäuden.

Glasfaser:
- Lichtwellenleiter (~300 000 km/s)
- Höherer Durchsatz
- Minimaler Biegeradius (keine Verlegung in engen Ecken, sonst kann das Kabel brechen)

Kupferkabel:
- ~200 000 km/s
- Auch Stromübertragung (z.B. Power over Ethernet)
- Ausreichend über kleine Distanzen (z.B. in Gebäuden)
- Günstiger

Aktuelle Produktivnetze

(c) Diskutieren Sie Vor- und Nachteile von modularen Netztechniksystemen.

Vorteile:

- Redundanz (= zusätzliches Vorhandensein von Ressourcen, wenn diese bei einem störungsfreien Betrieb nicht benötigt werden)
- Unterbrechungsfrei austauschbar

Nachteile:

- Platzbedarf
- Kosten
- Komplexität
- Fehleranfälligkeit

Transportschicht in Pseudocode (H)

Auf Übungsblatt 3 haben wir die Signatur der Funktion `sendto` behandelt. Diese implementiert einen Teil eines Protokolls des Internet-Modells der Schicht N. Sie verschickt `bytes` an einen in `addr_tuple` spezifizierten Empfänger.

Die Implementierung sei durch folgenden Pseudocode gegeben.

```
def sendto(bytes, (address, dst_port)):
    address_bytes = hton(address)
    destination_port = hton(dst_port)
    source_port = hton(random())
    length = hton(len(bytes) + 8)
    checksum = build_checksum(
        length, destination_port, source_port)

    PDU = length + destination_port + source_port + checksum + bytes
    route_packet(address_bytes, PDU)

    return
```

Transportschicht in Pseudocode

host to network byte order

Die Funktion `hton` konvertiert hier jeden beliebigen Typ zu einem für die Übertragung geeigneten Byte-Format („big-endian"). Gehen Sie wieder von dem folgenden Beispielaufruf aus.

```
msg = bytes([0x48, 0x65, 0x6c, 0x6c, 0x6f])
recipient = ("192.168.1.135", 6243)
sendto(msg, recipient)
```

(a) Welche Aussage können Sie über die (N)-PDU treffen?

Implementierung gegeben: (N)-PCI wird generiert und mit der (N)-SDU zur (N)-PDU

Transportschicht in Pseudocode

(b) Welche Informationen sind in der (N)-PCI enthalten? Stimmen diese mit der (N)-ICI überein?

(N)-PCI enthält:

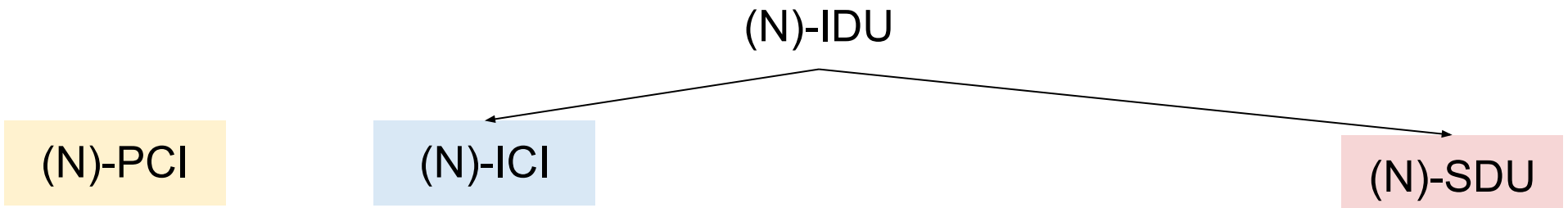
- Länge in Bytes (= Länge des Inhalts + 8)
- Destination Port aus (N)-ICI
- zufälligen Source-Port
- Checksum / Prüfsumme

Die (N)-PCI enthält einen Teil der (N)-ICI, stimmt aber nicht überein.

(c) Was entspricht $(N-1)$-ICI, -ID bzw. -IDU?

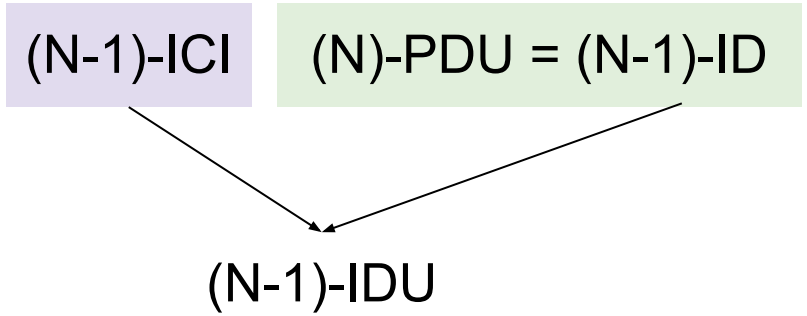
- address_bytes entspricht (N-1)-ICI
- (N)-PDU entspricht (N-1)-ID
- Beides zusammen entspricht (N-1)-IDU

Transportschicht in Pseudocode



(PCI und ICI)

```
PDU = length + destination_port + source_port + checksum + bytes
route_packet(address_bytes, PDU)
```



Transportschicht in Pseudocode

(d) Handelt es sich um ein verbindungsorientiertes oder -loses Protokoll?
Vorgriff: Welches Protokoll wird hier skizziert? Mögliche Protokolle lernen Sie in der nächsten Vorlesung am 24.05. kennen.

- keine Hinweise auf Verbindung(-saufbau)
  → verbindungsloses Protokoll

- möglicherweise UDP, aber:
  - UDP Header haben 8 Byte, während der Code die Headerlänge offen lässt
  - Checksum (hier mit Inhaltlänge, Ziel- und Quellport) enthält bei UDP noch den eigentlichen Inhalt und Informationen der darunterliegenden Schicht (einem Pseudo-IP-Header)