



## IPv6

- **Motivation: Adressknappheit durch starkes Abwachsen des Internet (abgemildert durch verschiedene „kurzfristige“ Lösungsansätze)**
- **in wesentlichen Teilen seit 1998 standardisiert**
- **wichtigste Änderungen zu IPv4**
  - vergrößertes Adressfeld (128 Bit anstelle von 32 Bit)
  - Basis-Header, Optionen in verketteten Headern



## IPv6 - Adressen

- Länge der Adressen: 128 Bit
- Darstellung in 8 Gruppen von 4 x 4 Bit davon je 4 Bit in Hexadezimaldarstellung getrennt durch “:”
- Beispiel:                   4711:0000:0000:0000:0000:0005:EEC1:0008  
(entsprechend:       0100 0111 0001 0001: ... )
- **Eine** Folge von Nullen kann weggelassen werden und wird dann durch **::** gekennzeichnet

Bei dem Beispiel von oben ergibt sich damit: 4711::**0000**:EEC1:0008

- Nullen am Anfang einer Hexadezimalgruppe d.h. „**führende**“ Nullen können ebenfalls weggelassen werden:

Für das Beispiel von oben folgt damit:       4711::~~00~~5:EEC1:~~00~~8  
= 4711:: 5: EEC1: 8

- Die IPv6-Adressen bezeichnen (wie bei IPv4) einzelne Interfaces



## IPv6 - Adressen

- IPv6 verwendet Adressen-„Präfixe“, die durch ein „/“ abgetrennt werden  
z.B. IPv6-Adresse/ n  
n bezeichnet dann die ersten n Bit der Adresse, die für das Präfix relevant sind.  
**Dieser Bereich der Adresse kann vom „Besitzer“ der Adresse nicht verändert werden und zeigt in der Regel den für das Routing relevanten Bereich der Adresse.**
- **Provider** erhalten in der Regel deutlich größere Adressbereich und versorgen damit ihre Kunden mit kleineren (dem Bedarf angepassten) IPv6-Adressbereichen.
- **Beispiel:** Ein Provider erhält den Adressbereich 4711: A::/ 32. Eine größere Firma könnte von diesem Provider den Adressbereich 4711:A:1::/48 erhalten und eine kleine Firma (oder ein Privathaushalt) den Adressbereich 4711:A:B:C::/64.



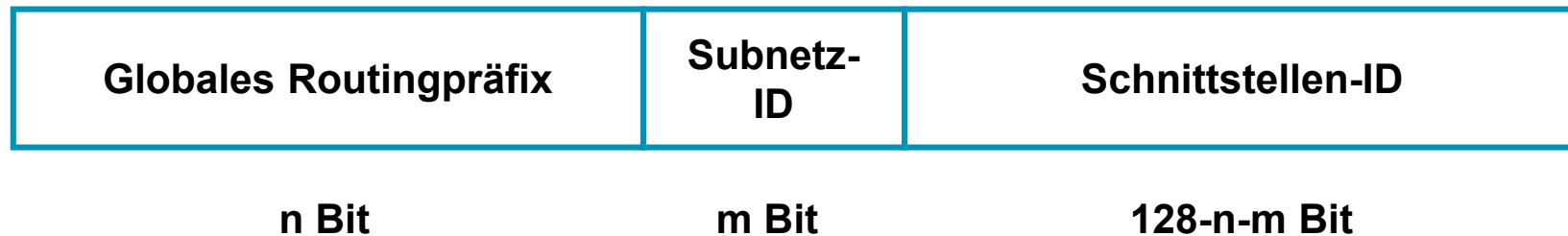
## IPv6 - Adresstypen

- **Unicast-Adressen**  
bezeichnen immer eine Schnittstelle
- **Multicast-Adressen**  
bezeichnen mehrere Schnittstellen
- **Anycast-Adressen**  
bezeichnen mehrere Schnittstellen, wobei die Daten nur an eine und zwar an die nächstgelegene Schnittstelle (d.h. die mit dem kürzesten Pfad) ausgeliefert werden
- **Es gibt bei IPv6 keine Broadcast Adressen!!!**



## Globale Unicast-Adressen (1)

- Allgemeine Struktur einer globalen Unicast-Adresse nach RFC 4291

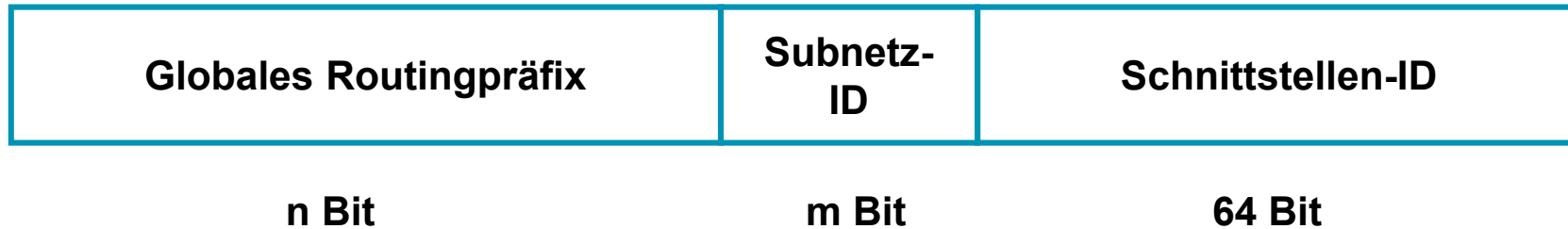


- Alle globalen Unicast Adressen, die nicht mit 000 beginnen, haben eine **Schnittstellen-ID mit einer Länge von 64 Bit**
- Global gültige Adressen, die zur Zeit vergeben werden, beginnen in der Regel mit 2001



## Globale Unicast-Adressen (2)

$n + m = 64 \text{ Bit !}$



- Das globale Routingpräfix kennzeichnet den Standort einer Firma oder Organisation
- Die Subnetz-ID kann zur Definition von Subnetzen in mehreren Ebenen verwendet werden
- Die Schnittstellen-ID bezeichnet eine Schnittstelle im Subnetz einer Organisation

### Eine globale Unicast-Adresse umfasst 3 Ebenen:

- Die ersten n Bit bezeichnen die **Öffentliche Topologie**
- Die nächsten m Bit bezeichnen die **Standort Topologie**
- Der letzte Bereich bezeichnet eine **Schnittstelle**



## Generierung der Schnittstellen-ID (1)

- Bei IPv4:

Zuweisung der Rechner IP-Adressen entweder manuell oder automatisiert  
z.B. mit dem Dynamic Host Configuration Protocol (DHCP)

### *Wie könnte eine eindeutige Schnittstellen-ID generiert werden ???*

- Generierung des lokalen Anteils der Adresse mit einem Zufallsverfahren (RFC 4941)
- Bei IPv6 hat die Schnittstellen-ID eine Länge von 64 Bit  
=> Ableitung der Schnittstellen-ID aus der MAC- Adresse möglich !

### Ableitung der Schnittstellen-ID aus der MAC Adresse nach EUI-64

- Abbildung von 48 Bit auf 64 Bit notwendig
- Einfügen von FF und FE (11111111 11111110) in der „Mitte“ d.h. nach dem 24. Bit
- Eine weltweiteindeutige Adresse ist durch eine „0“ im vorletzten Bit des ersten Byte markiert
- Markierung einer lokal gültigen Adresse durch eine „1“ im vorletzten Bit des ersten Byte

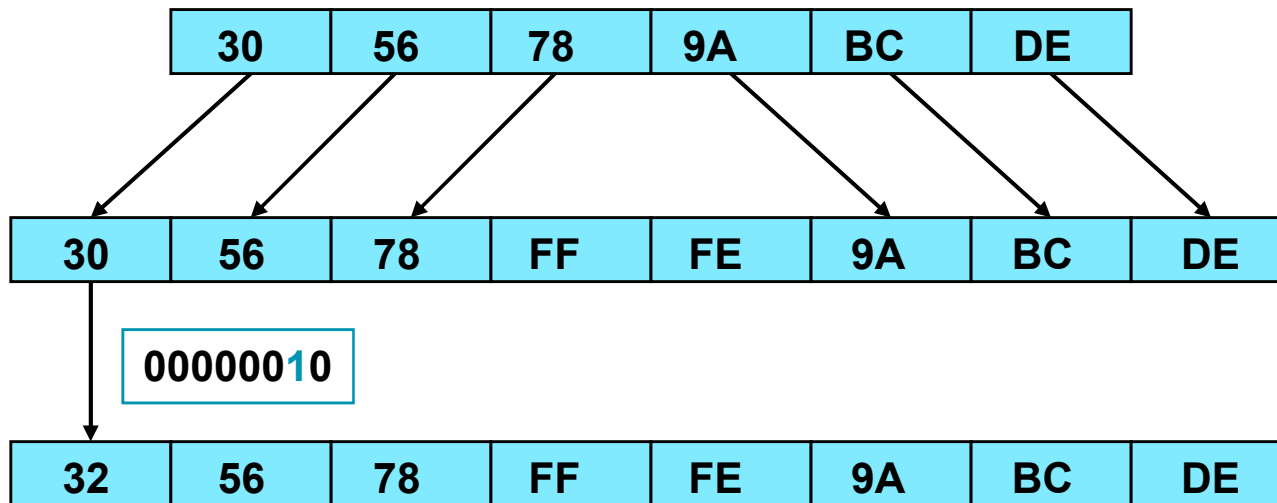




### Beispiel:

- Generierung einer Schnittstellen ID aus der MAC Adresse nach EUI-64

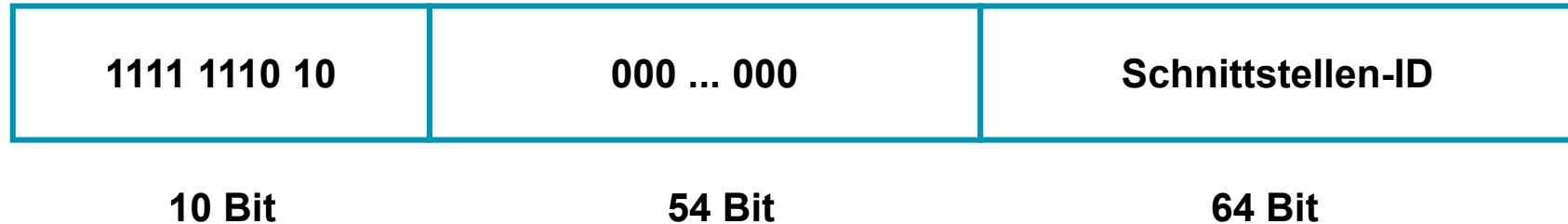
48 Bit → 64 Bit



Das vorletzte Bit im ersten Byte zeigt die „globale Eindeutigkeit“ einer MAC Adresse an. Es wird mit „OR“ 00000010 verknüpft und auf „lokale Gültigkeit“ gesetzt.



## Lokale Unicast-Adressen (1)

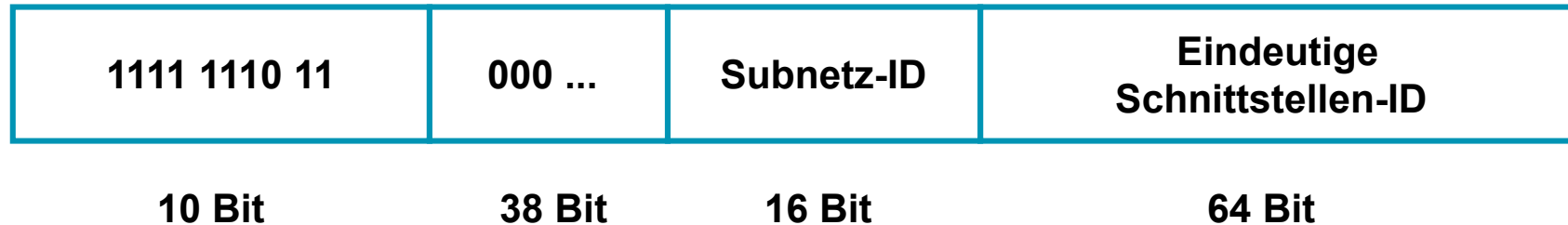


## Verbindungslokale Adressen (Link Local)

- **Beginnen mit FE80 und werden immer automatisch generiert und z.B. für Zwecke der Nachbarerkennung verwendet**
- **Gültigkeitsbereich ist die lokale Verbindung**
- **Entsprechen den automatisch auf Computern konfigurierten IPv4 Adressen**



## Lokale Unicast-Adressen (2)

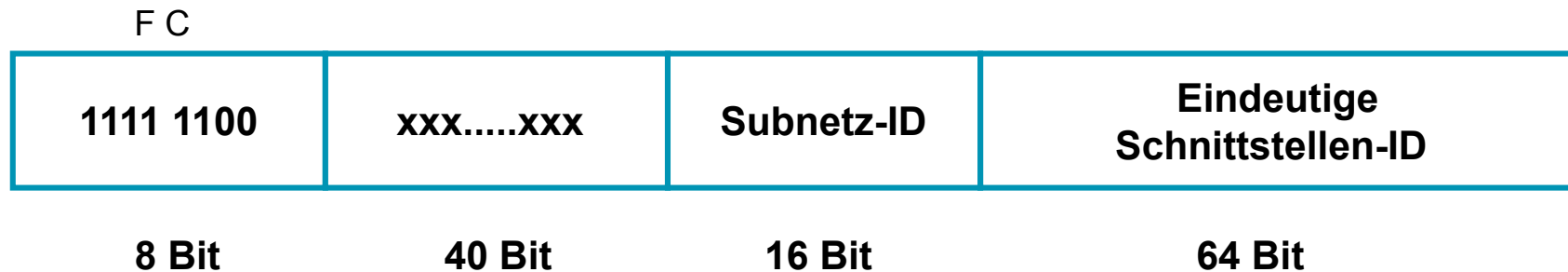


### Standortlokale Adressen (Site Local, veraltet und werden nicht mehr verwendet)

- Konnten in privaten Intranets verwendet werden
- Waren von anderen Standorten aus nicht erreichbar und wurden nicht ins Internet weitergeleitet
- Wurden nicht automatisch konfiguriert
- Entsprachen den privaten IPv4 Adressen



## Lokale Unicast-Adressen (3)



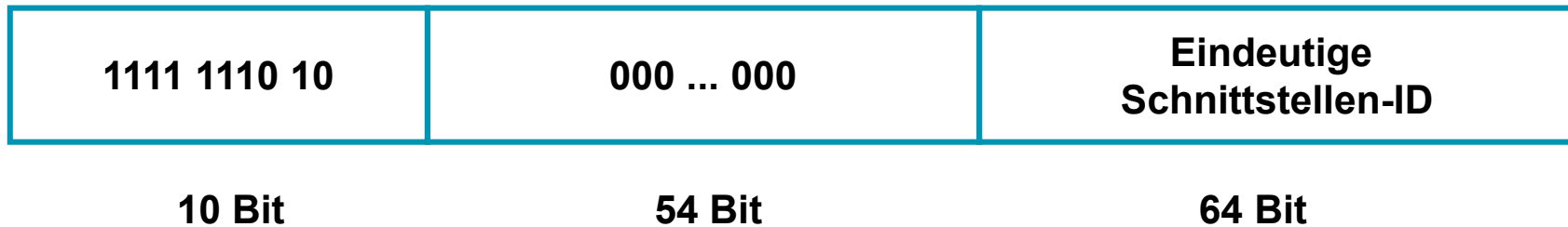
## Standortlokale Adressen (Unique Local)

- Mit 40 Bit langer zufällig erzeugter Side-ID xxx...xxx (dadurch ist die Side-ID mit hoher Wahrscheinlichkeit eindeutig!)
- Dadurch sind diese Adressen auch mit hoher Wahrscheinlichkeit eindeutig
- Können in privaten Intranets verwendet werden
- Entsprechen den privaten IPv4 Adressen



## Beispiel: Bildung einer lokal gültigen Adresse

- Z.B. mit Präfix für „Lokal Link“ Adresse: FE 80 (= 1111 1110 1000 0000 ),  
dann 48 Bit 00 .. 00 und  
64 Bit eindeutige Schnittstellen ID aus MAC-Adresse nach EUI-64





## Spezielle IPv6 Adressen (1)

### **Nicht spezifizierte Adresse (0:0:0:0:0:0:0:0 oder ::)**

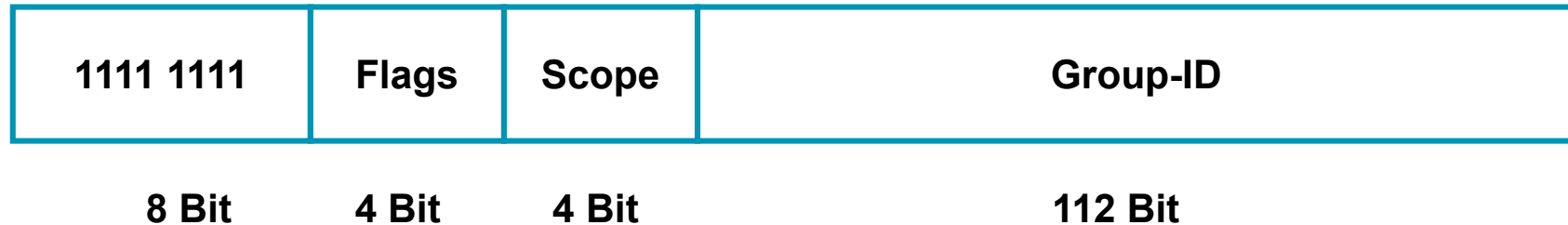
- **Zeigt an, dass keine Adresse vorhanden ist**
- **Wird niemals zugewiesen oder als Zieladresse verwendet**

### **Loopback-Adresse (0:0:0:0:0:0:0:1 oder ::1)**

- **Kennzeichnet eine Loopbackschnittstelle d.h. damit kann ein Knoten Pakete an sich selbst schicken**
- **Wird niemals von einem IPv6-Router weitergeleitet**
- **Entspricht 127.0.0.1**



## Multicast-Adressen



- **Beginnen immer mit FF**
- **Flags und Scope enthalten Informationen über den Geltungsbereich und Aufbau der Adresse (siehe RFC 4291 und RFC 3306)**
- **Group ID gibt die Multicast Gruppe an (siehe RFC 4291 und RFC 3306)**



## Beispiel aus dem Labor: Warum hat das Interface eth2 zwei IPv6 Adressen?

```
netlab@flanders: ~  
Datei Bearbeiten Reiter Hilfe  
netlab@flanders:~$ ip addr show  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536  
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
   inet 127.0.0.1/8 scope host lo  
       valid_lft forever  
   inet6 ::1/128 scope host  
       valid_lft forever  
2: eth2: <BROADCAST,MULTICAST> mtu 1500 state DOWN  
   link/ether 00:15:77:3e:d5:37 brd ff:ff:ff:ff:ff:ff  
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500  
   link/ether 00:1c:c0:a3:f9:87 brd ff:ff:ff:ff:ff:ff  
   inet 10.30.0.209/24 brd 10.30.0.255  
       valid_lft 86295sec  
   inet6 2001:638:408:100:9ce2:41a3:d4a9:b2b1/128 scope global  
       valid_lft 3495sec  
   inet6 fe80::21c:c0ff:fea3:f987/64 scope link  
       valid_lft forever
```

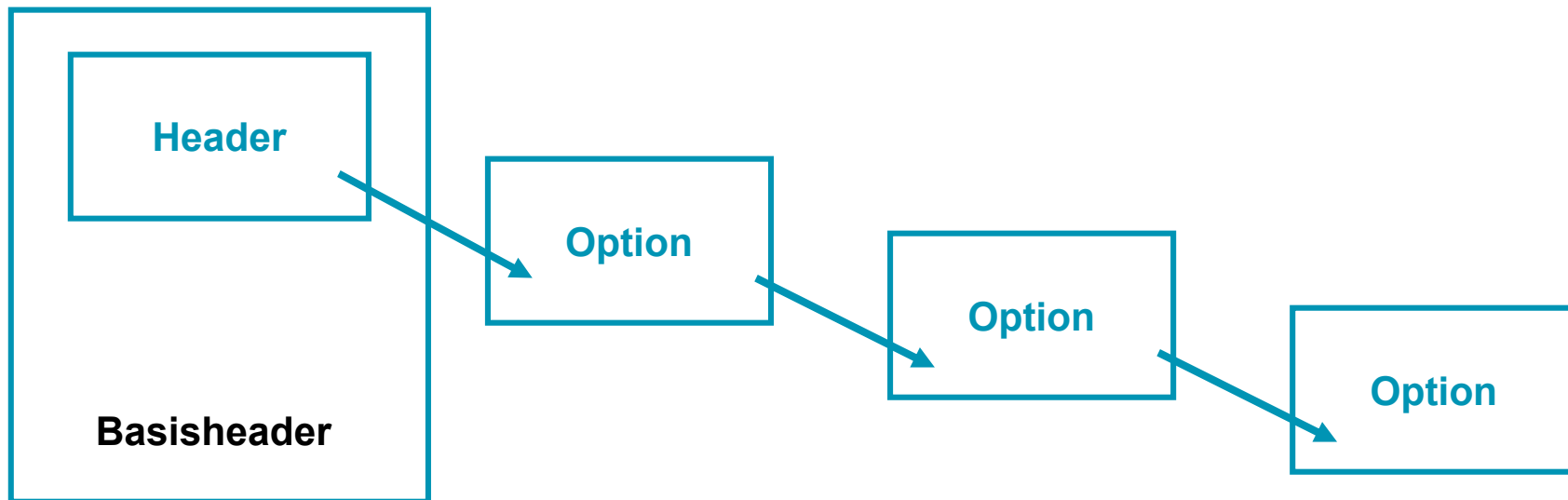




## Prinzipieller Aufbau des IPv6 Headers

- Es gibt einen Basisheader, der nur die wichtigsten und für den Transport notwendigen Daten enthält
- Die Optionen befinden sich in hierarchisch verketteten Headern, **die nicht von allen Netzkomponenten ausgewertet werden, sondern nur dort, wo die Information auch benötigt werden!!!**

⇒ Die Reihenfolge, in der die Header auftreten können, ist festgelegt!





## Aufbau des Basisheaders (1)



Version	Class	Flow-Label	
Payload Length		Next	Hop-Limit
Source Address (128 Bit)			
Destination Address (128 Bit)			



## Aufbau des Basisheaders bei IPv6 (2)

- **Der Basisheader hat eine Länge von 40 Byte und enthält nur die für den Transport einer Nachricht notwendigen Angaben**
- **Versionsnummer: bei IPv6 „6“**
- **Verkehrsklasse (Class): zur Kennung von verschiedenen Verkehrsklassen**
- **Flusskennung (Flow-Label): zur eindeutigen Kennzeichnung eines Datenstroms**
- **Nutzdatenlänge (Payload length): max. 65.535 Byte**
- **Nächster Header (Next): Typ des nächsten Headers bzw. falls kein weiterer Header folgt: Typ der Nutzdaten (also z.B. TCP)**
- **Teilstreckenlimit (Hop Limit): Erlaubte Anzahl von Netzknoten, wird in jedem Router um eins gesenkt**
- **Empfänger- und Senderadresse**



## Kennzeichnung verschiedener Headertypen im NEXT Feld

Nummer	Typ des Headers
0	Hop-by-Hop-Optionen
60	Ziel-Optionen
43	Header für Routing
44	Header für Fragmentierung
51	Header für Authentisierung
50	Header für Verschlüsselung
59	Kein weiterer Header
6	TCP
17	UDP



## Hierarchische Reihenfolge der Header bei IPv6

IPv6 Basisheader
Hop-by-Hop-Optionen (0)
Ziel-Optionen (60) für jeden Router
Header für Routing (43)
Header für Fragmentierung (44)
Header für Authentisierung (51)
Header für Verschlüsselung (50)
Ziel-Optionen (60) für das Endgerät
TCP/ UDP etc.



## Beispiel für ein minimales Paket bei IPv6

IPv6 Basisheader  
Next = 17 (d.h. UDP)

UDP- Header  
Daten



## Beispiel für ein komplexes Paket bei IPv6

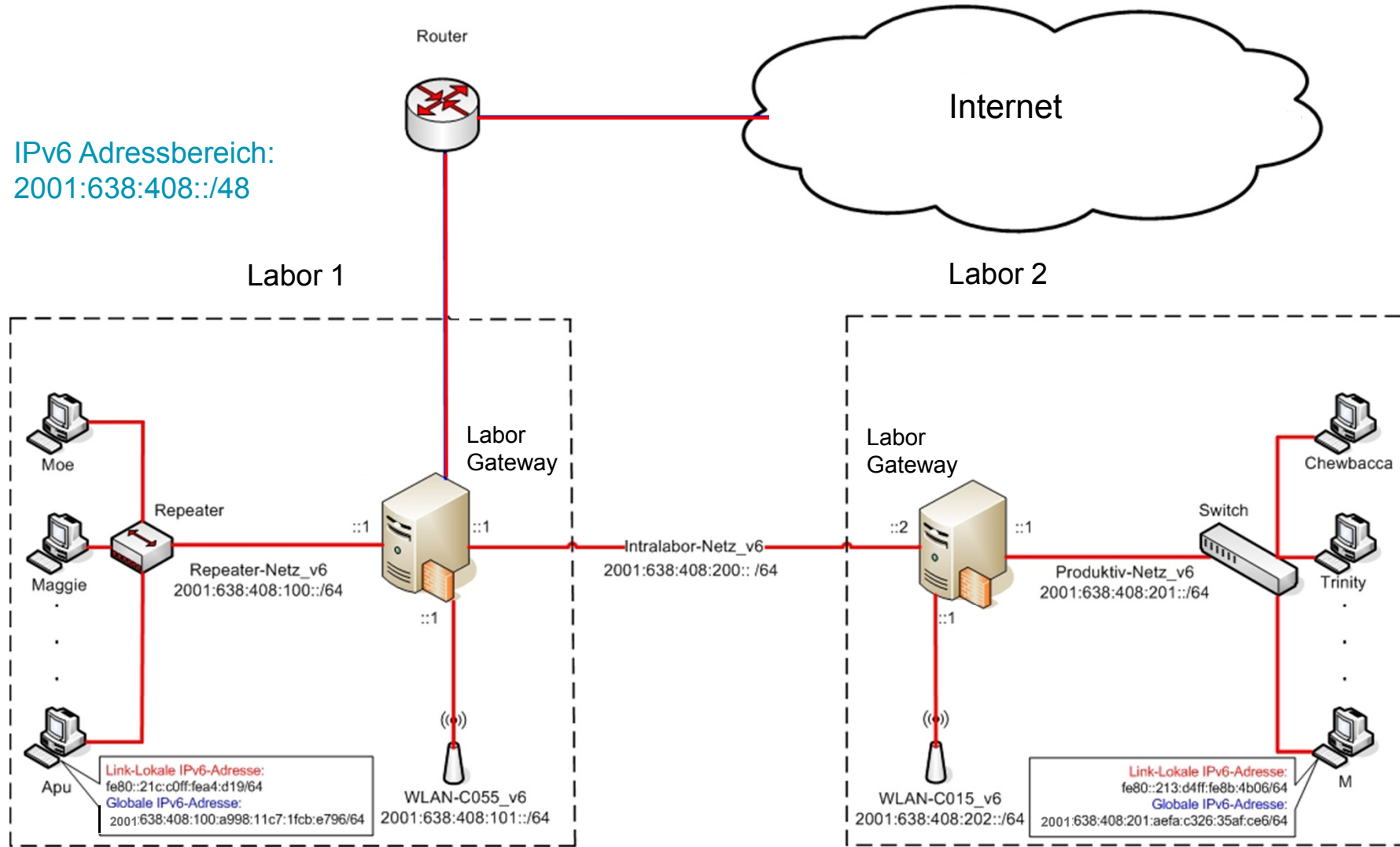
IPv6 Basisheader Next = 43 (d.h. Routing-Header)
Routing-Header Next = 44 (d.h. Fragmentierung)
Header für Fragmentierung Next = 51 (d.h. Authentisierung)
Header für Authentisierung Next = 50 (d.h. Verschlüsselung)
Header für Verschlüsselung Next = 6 (d.h. TCP)
TCP-Header, Daten



## Beispiel für eine IPv6 Infrastruktur

Stand: Januar 2011

IPv6 Adressbereich:  
2001:638:408::/48

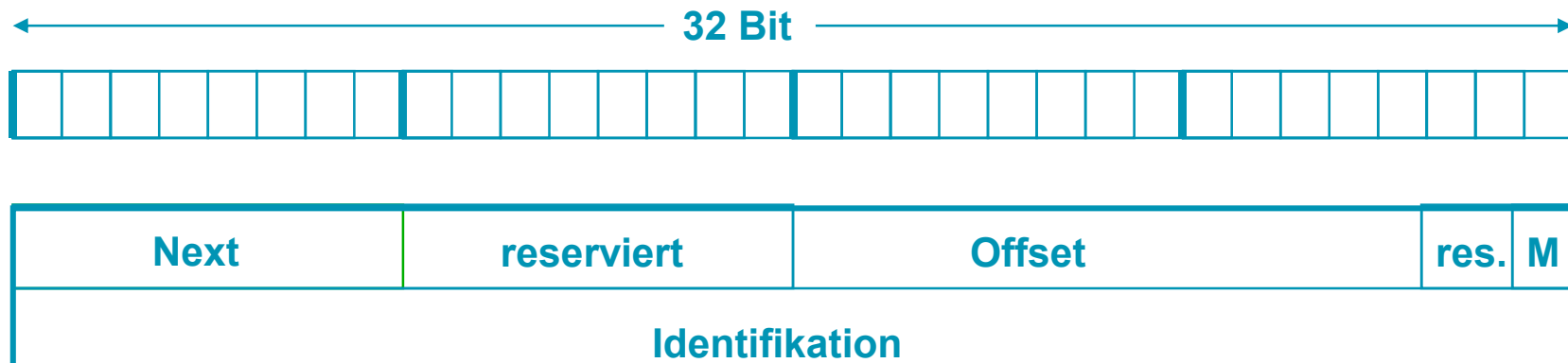






## Header zur Fragmentierung

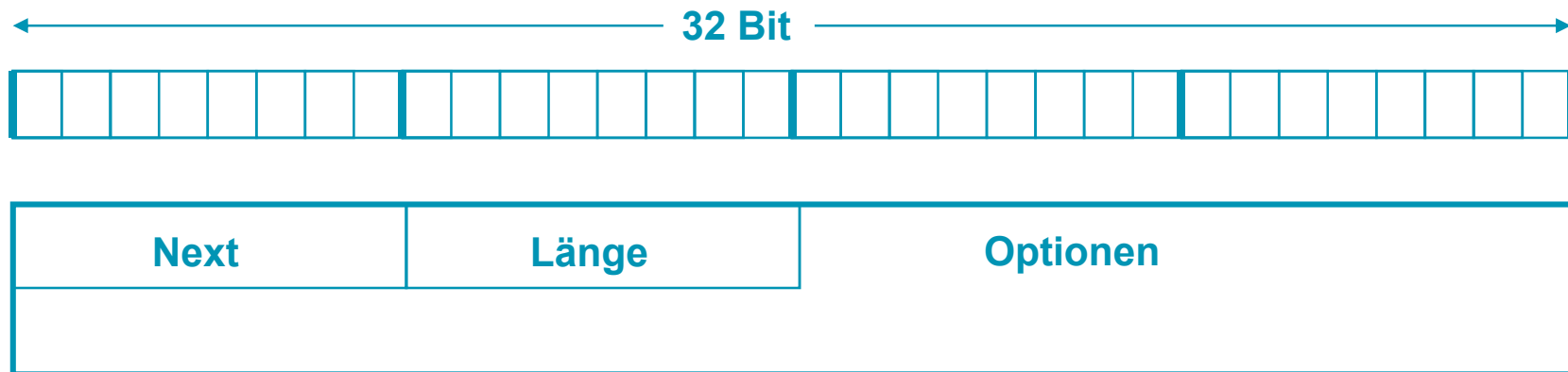
Bei IPv6 darf nur der Sender fragmentieren, Header unterstützt das Zusammenfügen des Datagramms beim Empfänger



- Nächster Header (Next): Typ des nächsten Headers
- reserviertes Feld mit 8 Bit
- Offset (13 Bit): Position des Fragments in 8 Byte
- Reserviertes Feld mit 2 Bit
- Flag: M = 1, es folgen weitere Fragmente, M = 0, letztes Fragment
- Identifikation: alle Fragmente haben gleichen Wert



## Hop-by-Hop Header/ Ziel-Optionen (1)



Der Hop-by-Hop Header wird bei jedem Übertragungsschritt ausgewertet, verschiedene weitere Optionen möglich

- **Felder im Header: Next, Länge und Optionsfelder**



## Hop-by-Hop Header (2)

- Jedes Optionsfeld besteht aus Optionstyp, Länge des Optionsparameterfeldes (in Byte) und Optionsparameter

