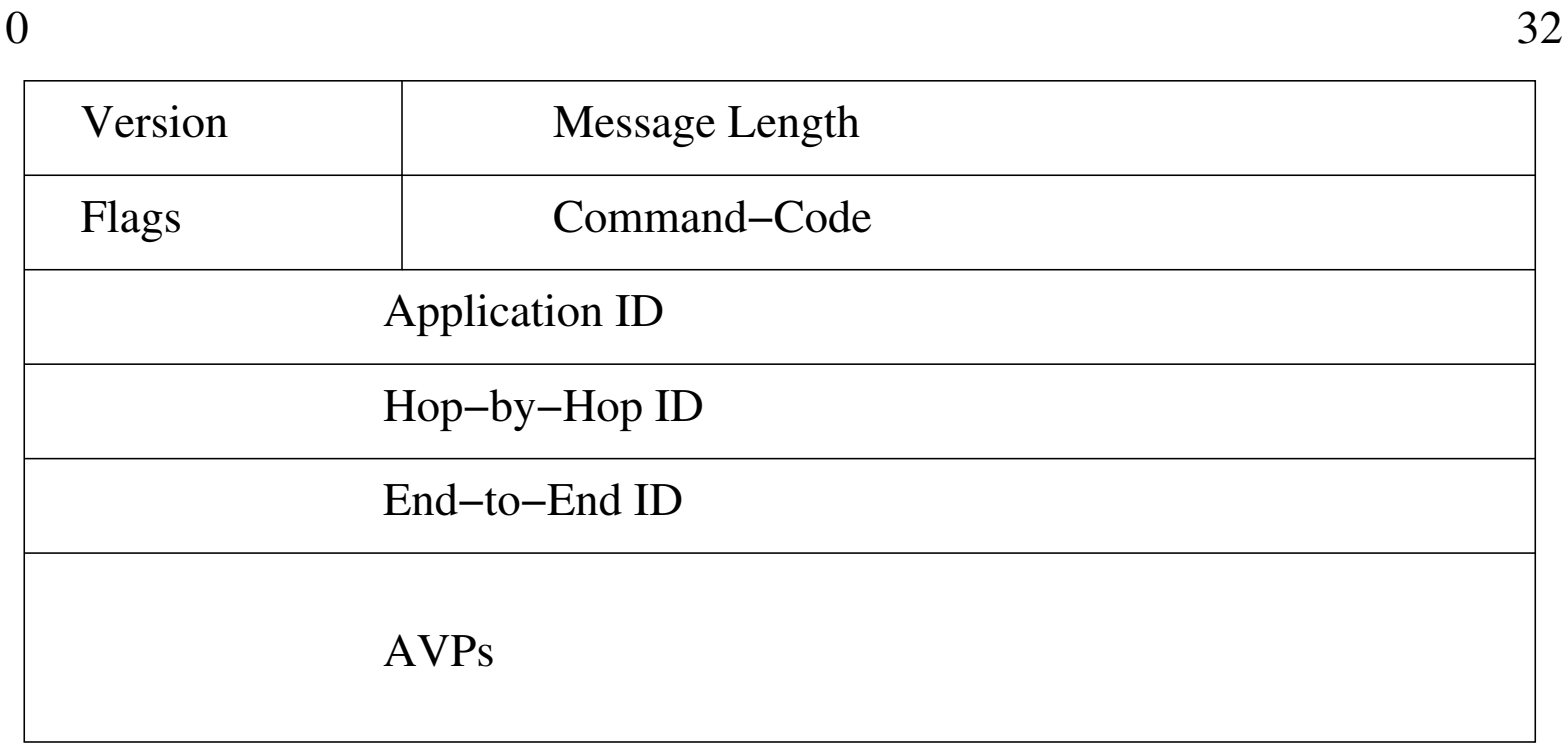
DIAMETER Base Protocol (RFC3588)

DIAMETER ist eine (nicht rückwärtskompatible) Fortentwicklung des RADIUS Protokolls (Remote Authentication Dial In User Service, RFC2865). Die wichtigsten Unterschiede sind:

- Es benutzt einen zuverlässigen Transport (z.B. TCP)
- DNS SRV und NAPTR zum Auffinden eines Servers
- Unterstützt verschiedenen Nachrichtentypen auf einem Transport
- leicht erweiterbar
- bidirektional (Push - Pull Dienste)
- Länge von Attributen ist auf 16kB begrenzt, nicht 253B

DIAMETER Rahmen



Felder im DIAMETER Rahmen

- **Version**: 1 für Diameter nach RFC3588
- **Message Length**: Länge in Byte inklusive der Header
- **Flags**: 8Bit, von denen 0-3 benutzt werden:
  0. 1 für Request, 0 für Response
  1. Nachricht kann von Proxies weitergeleitet werden.
  2. Error Message, darf nicht in Requests gesetzt werden.
  3. Erneute Übertragung nach einem Link Fehler
- **Command-Code**: Typ der Nachricht, die Werte werden durch die IANA verwaltet.

Felder im DIAMETER Rahmen

- **Application-ID**: Identifiziert die Anwendung, muß mit einem Application-ID Attribut übereinstimmen.
- **Hop-by-Hop ID**: 32Bit, dienen dazu, Request und Response einander zuzuordnen, muß in einem gewissen Zeitfenster eindeutig sein (auch nach Neustart).
- **End-to-End ID**: 32Bit, um Doubletten erkennen zu können, muß im Zeitfenster von 4 Minuten eindeutig sein (auch nach Neustart).
- **AVPs**: Folge von Attribut-Wert Paaren

DIAMETER AVP

AVP Code	
Flags	AVP Length
Vendor ID	
Data	

Felder im DIAMETER AVP

- **AVP Code**: Identifiziert (zusammen mit der Vendor-ID) den Typ des Attributes
- **Flags**: Nur Bits 0-2 werden benutzt:
  0. Vendor-ID ist im Rahmen
  1. Attribut muß vorhanden sein (bei dieser Nachricht)
  2. Attribut ist verschlüsselt
- **AVP Length**: Länge des Attributes in Bytes vom AVP-Code bis zum Ende der Daten
- **Vendor-ID**: Namensraum, in dem der AVP-Code interpretiert wird.
- **Data**: Daten, die je nach AVP-Code interpretiert werden.

XML Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<presence
  xmlns="urn:ietf:params:xml:ns:pidf"
  entity="sip:test@10.1.195.159">
  <tuple id="udupwd">
    <status>
      <basic>busy</basic>
    </status>
  </tuple>
</presence>
```

XPath, `http://www.w3.org/TR/xpath`

Sprache, um Teile eines XML Dokumentes zu adressieren:

- `text()`: Zugriff auf Textknoten
- `[Name="Wert"]`: Test auf Textknoten, z.B.: `status[basic="busy"]`
- `.../@Name`: Zugriff auf Wert eines Attributes
- `[@Name="Wert"]`: Test auf Wert eines Attributes
- `.../Name1/Name2`: Abstieg im Dokumentbaum
- `.../Name1[n]`: *n*-te Instanz von Name1

Beispielimplementation in Python

```
from xml.xpath import Evaluate
from xml.dom import minidom

path = '/presence[@entity="sip:test@10.1.195.159"]\
//status/basic'

doc = minidom.parse("test.xml").documentElement
x = Evaluate(path, doc)
print x[0].toxml()

# ergibt:
# <basic>busy</basic>
```

XML Configuration Access Protocol (XCAP, RFC4825)

XCAP dient dazu, in hierarchisch strukturierten Daten (XML Dateien), Einträge einzufügen, zu verändern oder zu löschen.

- XCAP basiert auf HTTP, die URL ist ein XPath Ausdruck, der ein Teildokument referenziert.
- `GET` liest den referenzierten Teil.
- `PUT` fügt den Nachrichtenkörper an der angegebenen Stelle in das Dokument ein.
- `DELETE` löscht den referenzierten Teilbaum des Dokumentes.

XDMS (vgl. OMA-ERELD-XDM-V1_0_1-20061128-A.pdf)

Ein XML Document Management Server dient dazu, ein XML Dokument in einer SIP/IMS Infrastruktur zugreifbar zu machen.

- `XCAP` dient dazu, Teile eines Dokumentes zu bearbeiten
- `SUBSCRIBE` wird benutzt, damit ein Kunde automatisch über Änderungen von Teilen des XML Dokumentes informiert wird.
- `NOTIFY` wird vom XDMS gesendet, wenn sich ein Teil des Dokumentes geändert hat und der Kunde darauf subscribiert ist.