SPENGERGASSE
ausbildung mit zukunft

CCNA Exploration
Network Fundamentals

ARP
Address Resolution Protocol

ARP: Address resolution protocol

1. Eigenschaften
   - ARP-Cache
   - Aufbau
2. Ablauf
   - Beispiel
   - Flussschema
3. ARP-Arten
4. Sicherheit
   - Man-In-The-Middle-Attacke
5. Fazit

ARP: Eigenschaften

- Protokoll der Vermittlungsschicht (Network Layer)
- Schafft die Zuordnung zwischen MAC- & IP-Adresse
  - Nötig, da IP-Adressierung in Transportschicht erfolgt, MAC-Adressierung aber in der Sicherungsschicht
- Kommunikation:
  - ARP-Request: Broadcast
  - ARP-Reply: Unicast

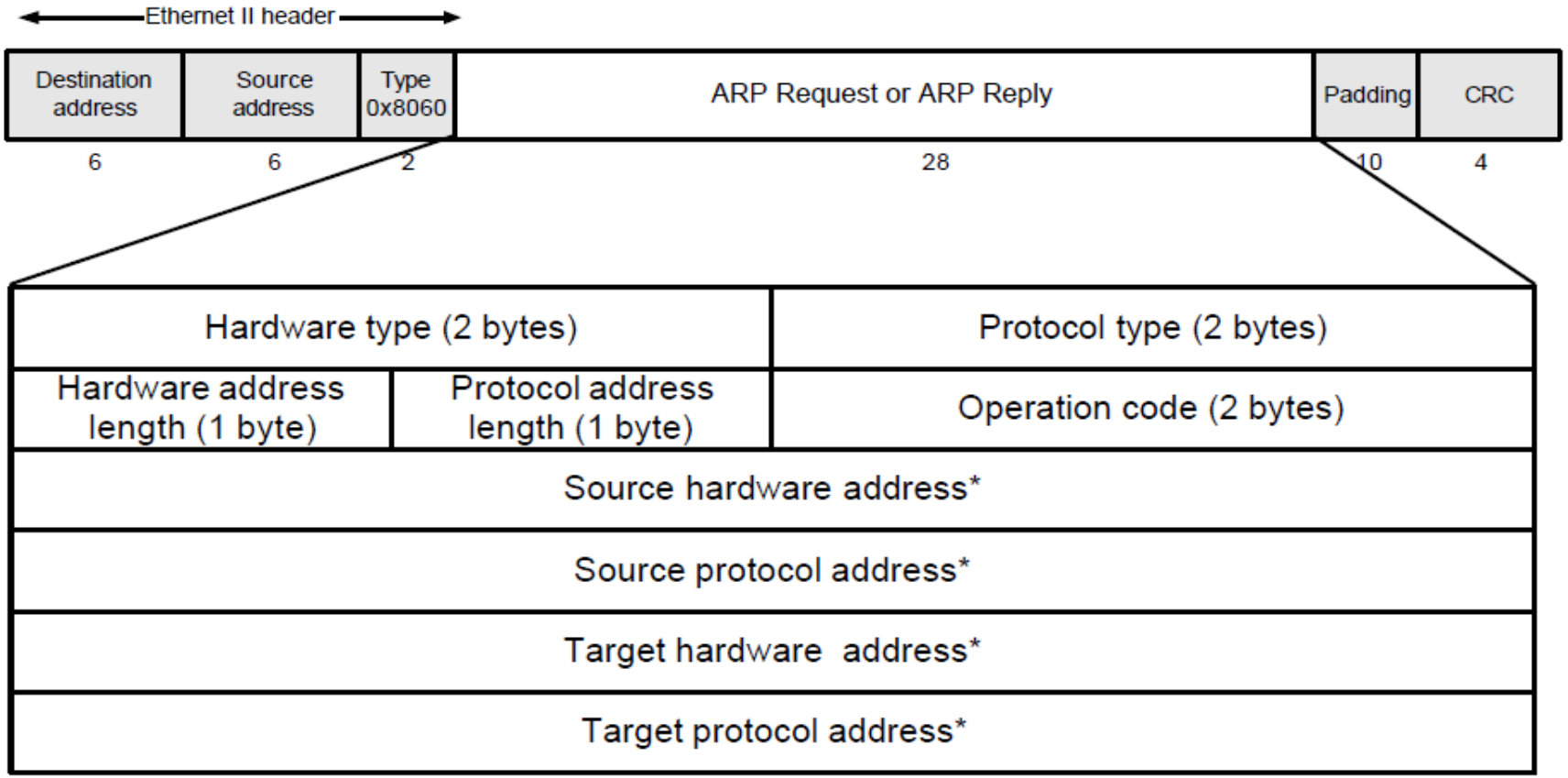
ARP Cache

- Zwischenspeicher für Adresszuordnung
  - Verringert unnötige Anfragen (Overhead)
- Enthält statische und dynamische Einträge
- Aktualisierung der dynamischen Einträge nach Timeout

Einordnung der Modelle

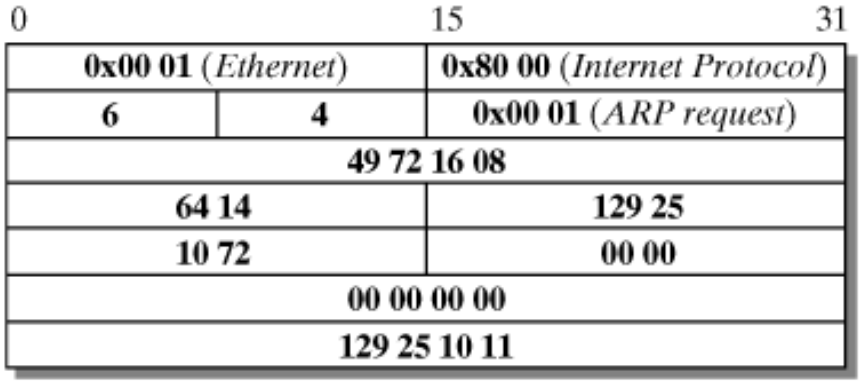
<table>
<tr><th>ISO-OSI</th><th>TCP/IP</th><th colspan="4">Protokolle</th></tr>
<tr><td>Application</td><td rowspan="3">Application</td><td colspan="4">HTTP, DNS, SMTP, …</td></tr>
<tr><td>Presentation</td><td colspan="4">EBCDIC, ASN.1, ASCII, …</td></tr>
<tr><td>Session</td><td colspan="4">RPC, NetBIOS, …</td></tr>
<tr><td>Transport</td><td>Transport</td><td colspan="2">TCP</td><td colspan="2">UDP</td></tr>
<tr><td rowspan="2">Network</td><td rowspan="2">Network</td><td colspan="4">IP</td></tr>
<tr><td colspan="4">ARP, RARP, …</td></tr>
<tr><td>Data Link</td><td rowspan="2">Link</td><td colspan="2">Ethernet</td><td colspan="2">…</td></tr>
<tr><td>Physical</td><td>X.25</td><td>RS-232</td><td>V.90</td><td>…</td></tr>
</table>

ARP Packet Format

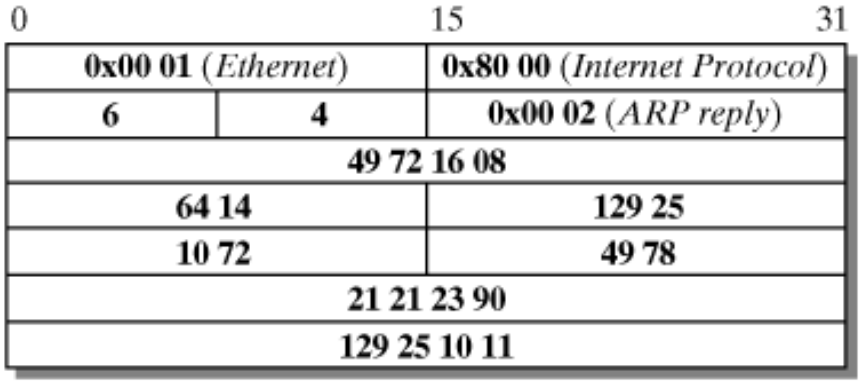


* Note: The length of the address fields is determined by the corresponding address length fields

ARP request to FF:FF:FF:FF:FF:FF



ARP reply to 49:72:16:08:64:14



ARP auf der Kommandozeile

- Tabelle ansehen:

  ```
  arp -a
  ```

  oder genauer mit:

  ```
  arp -a -v
  ```

- Statischen Eintrag hinzufügen:

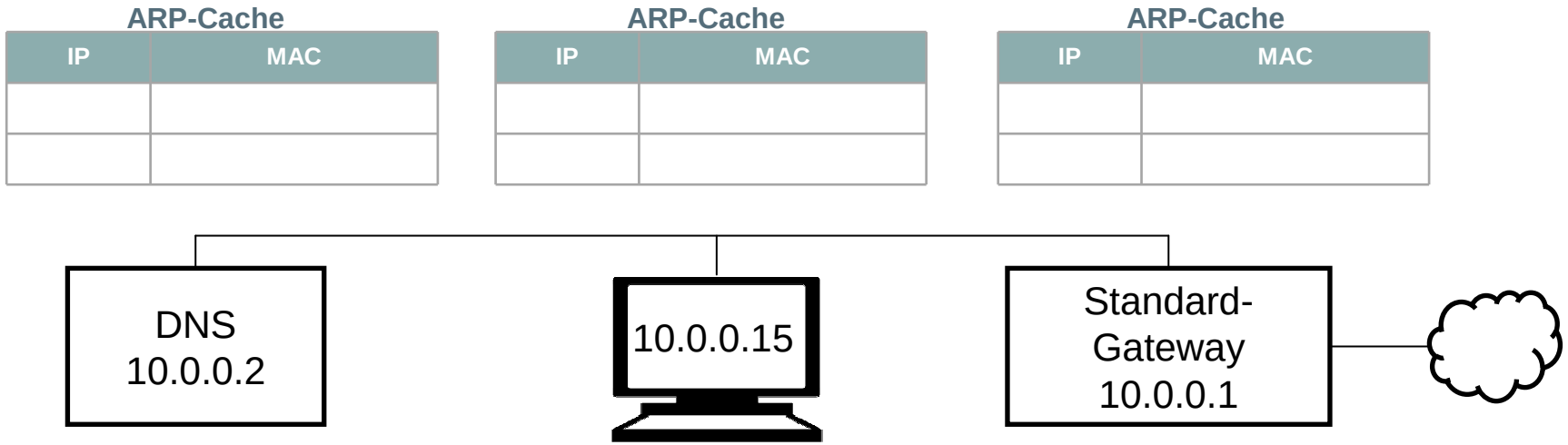
  ```
  arp -s 192.168.1.2  00-A3-FE-79-D8-CE
  ```

- Eintrag löschen:

  ```
  arp -d 192.168.1.2
  ```

Ablauf

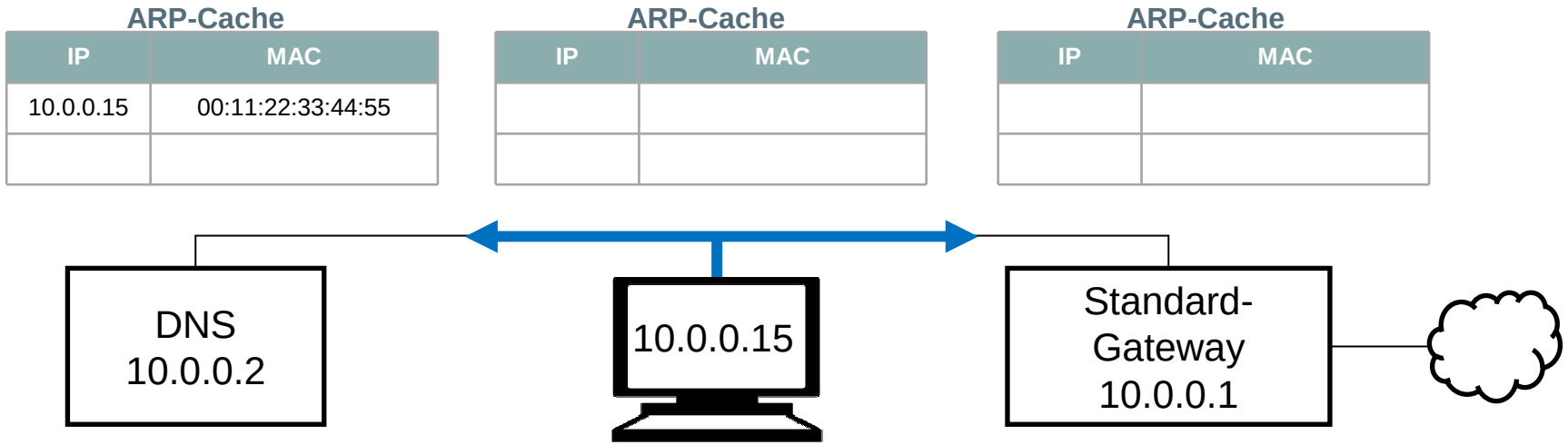
- ARP-Cache im Subnetz zu Beginn leer



- Rechner will Verbindung zu Facebook aufbauen, ohne IP- und MAC-Adresse zu kennen, kennt aber IP des DNS-Servers & IP des Standardgateways

Ablauf

- ARP-Cache im Subnetz zu Beginn leer
- ARP-Request senden



- Rechner will Verbindung zu Facebook aufbauen, ohne IP- und MAC-Adresse zu kennen, kennt aber IP des DNS-Servers & IP des Standardgateways
- sendet Broadcast-Request (*ff:ff:ff:ff:ff:ff*): *„Who has 10.0.0.2? Tell 10.0.0.15"*

Ablauf

- ARP-Cache im Subnetz zu Beginn leer
- ARP-Request senden, Unicast-Reply erhalten, ARP-Cache schreiben

ARP-Cache

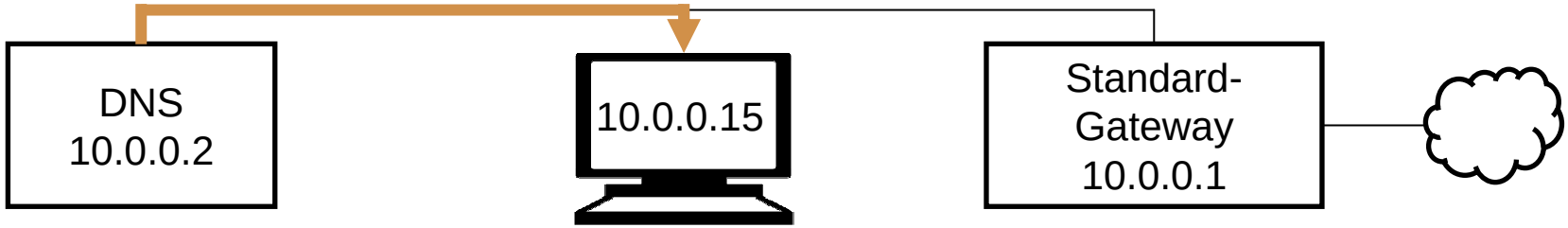
IP	MAC
10.0.0.15	00:11:22:33:44:55

ARP-Cache

IP	MAC
10.0.0.2	22:44:66:22:44:66

ARP-Cache

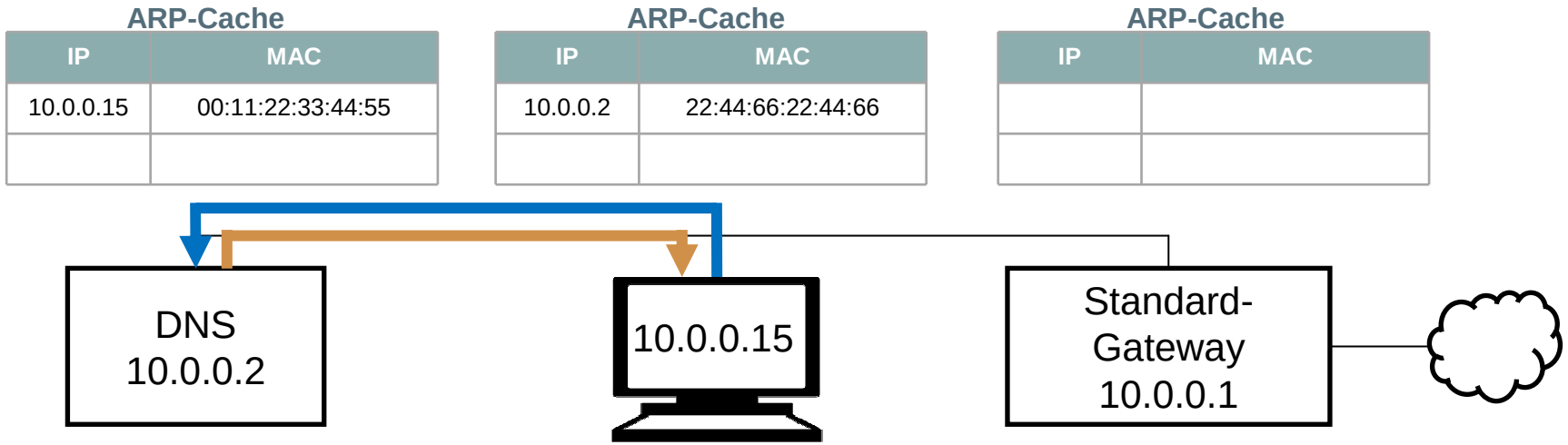
IP	MAC



- Rechner will Verbindung zu Facebook aufbauen, ohne IP- und MAC-Adresse zu kennen, kennt aber IP des DNS-Servers & IP des Standardgateways
- sendet Broadcast-Request (*ff:ff:ff:ff:ff:ff*): *„Who has 10.0.0.2? Tell 10.0.0.15"*
- erhält Unicast-Reply: *„10.0.0.2 is at 22:44:66:22:44:66"* → Cache schreiben

Ablauf

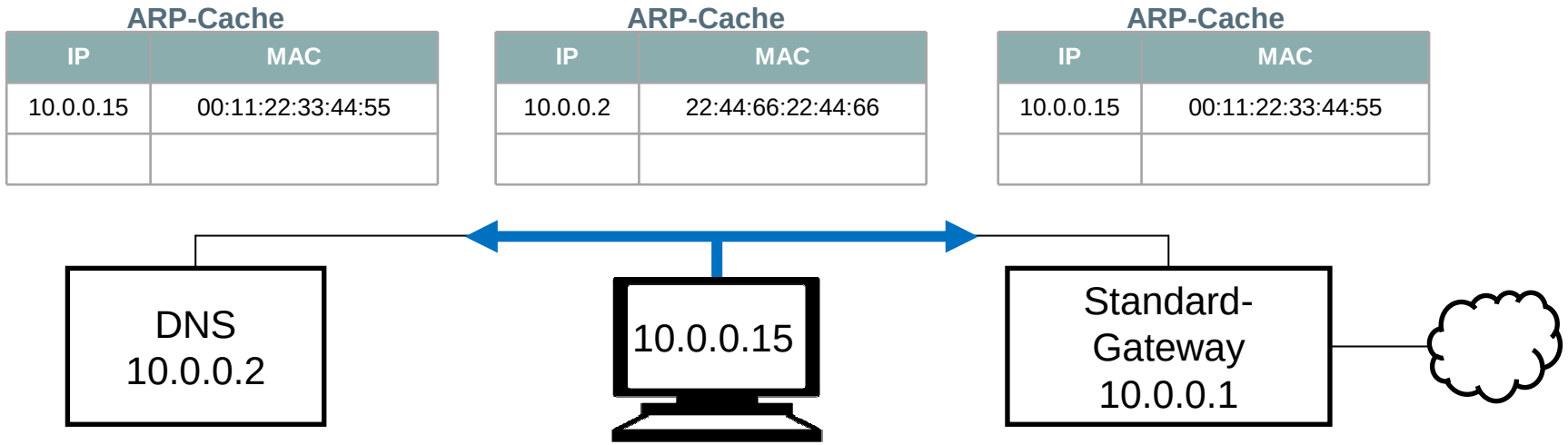
- ARP-Cache im Subnetz zu Beginn leer
- ARP-Request senden, Unicast-Reply erhalten, ARP-Cache schreiben



- Rechner will Verbindung zu Facebook aufbauen, ohne IP- und MAC-Adresse zu kennen, kennt aber IP des DNS-Servers & IP des Standardgateways
- sendet Broadcast-Request (*ff:ff:ff:ff:ff:ff*): *„Who has 10.0.0.2? Tell 10.0.0.15"*
- erhält Unicast-Reply: *„10.0.0.2 is at 22:44:66:22:44:66"* → Cache schreiben
- stellt Request an DNS-Host, welche IP „*www.facebook.com*" hat & erhält Reply

Ablauf

- ARP-Cache im Subnetz zu Beginn leer
- ARP-Request senden, Unicast-Reply erhalten, ARP-Cache schreiben



- Rechner will Verbindung zu Facebook aufbauen, ohne IP- und MAC-Adresse zu kennen, kennt aber IP des DNS-Servers & IP des Standardgateways
- sendet Broadcast-Request (*ff:ff:ff:ff:ff:ff*): *„Who has 10.0.0.2? Tell 10.0.0.15"*
- erhält Unicast-Reply: *„10.0.0.2 is at 22:44:66:22:44:66"* → Cache schreiben
- stellt Request an DNS-Host, welche IP „*www.facebook.com*" hat & erhält Reply
- sendet Broadcast-Request (*ff:ff:ff:ff:ff:ff*): *„Who has 10.0.0.1? Tell 10.0.0.15"*

Ablauf

- ARP-Cache im Subnetz zu Beginn leer
- ARP-Request senden, Unicast-Reply erhalten, ARP-Cache schreiben

ARP-Cache

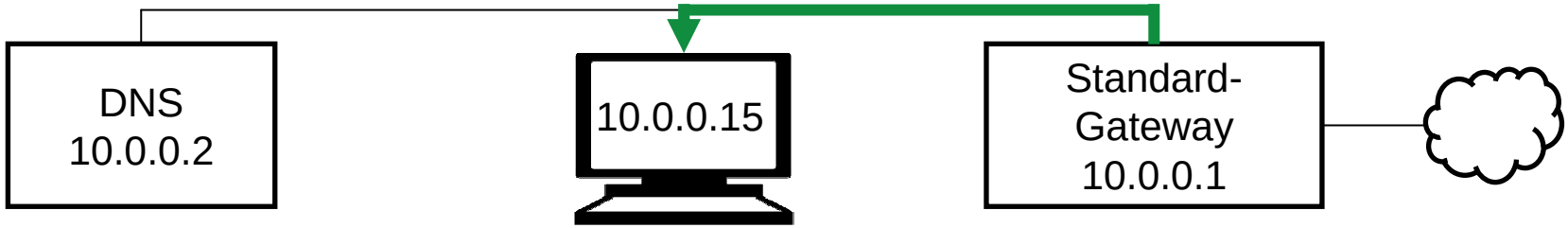
IP	MAC
10.0.0.15	00:11:22:33:44:55

ARP-Cache

IP	MAC
10.0.0.2	22:44:66:22:44:66
10.0.0.1	44:88:22:22:88:44

ARP-Cache

IP	MAC
10.0.0.15	00:11:22:33:44:55



- Rechner will Verbindung zu Facebook aufbauen, ohne IP- und MAC-Adresse zu kennen, kennt aber IP des DNS-Servers & IP des Standardgateways
- sendet Broadcast-Request (*ff:ff:ff:ff:ff:ff*): *„Who has 10.0.0.2? Tell 10.0.0.15"*
- erhält Unicast-Reply: *„10.0.0.2 is at 22:44:66:22:44:66"* → Cache schreiben
- stellt Request an DNS-Host, welche IP „*www.facebook.com*" hat & erhält Reply
- sendet Broadcast-Request (*ff:ff:ff:ff:ff:ff*): *„Who has 10.0.0.1? Tell 10.0.0.15"*
- erhält Unicast-Reply: *„10.0.0.1 is at 44:88:22:22:88:44"* → Cache schreiben

Ablauf

- ARP-Cache im Subnetz zu Beginn leer
- ARP-Request senden, Unicast-Reply erhalten, ARP-Cache schreiben

ARP-Cache

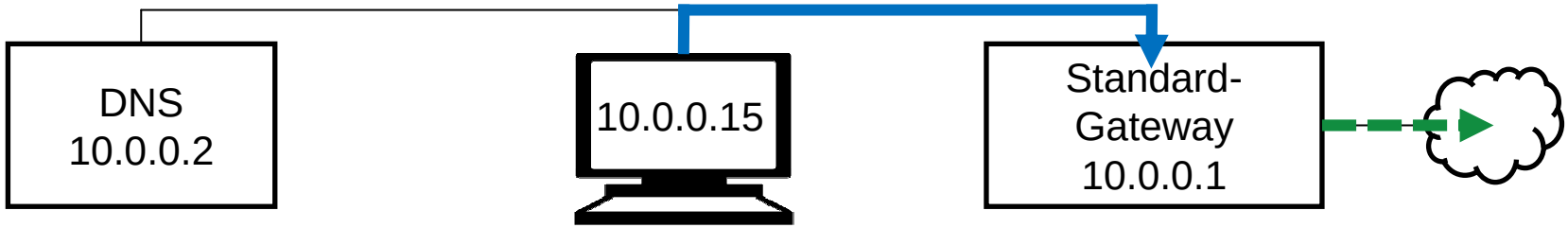
IP	MAC
10.0.0.15	00:11:22:33:44:55

ARP-Cache

IP	MAC
10.0.0.2	22:44:66:22:44:66
10.0.0.1	44:88:22:22:88:44

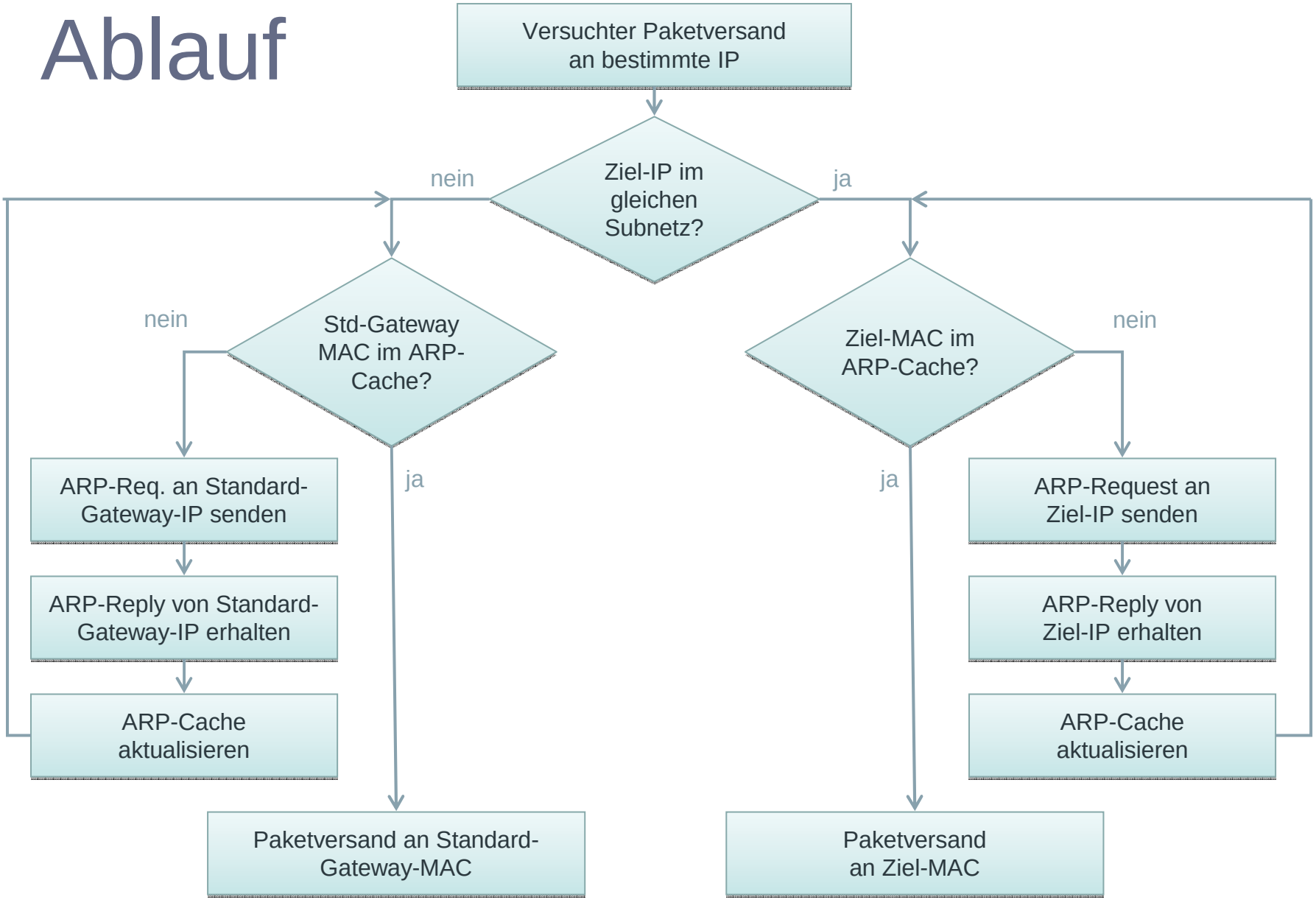
ARP-Cache

IP	MAC
10.0.0.15	00:11:22:33:44:55



- Rechner will Verbindung zu Facebook aufbauen, ohne IP- und MAC-Adresse zu kennen, kennt aber IP des DNS-Servers & IP des Standardgateways
- sendet Broadcast-Request (*ff:ff:ff:ff:ff:ff*): *„Who has 10.0.0.2? Tell 10.0.0.15"*
- erhält Unicast-Reply: *„10.0.0.2 is at 22:44:66:22:44:66"* → Cache schreiben
- stellt Request an DNS-Host, welche IP „*www.facebook.com*" hat & erhält Reply
- sendet Broadcast-Request (*ff:ff:ff:ff:ff:ff*): *„Who has 10.0.0.1? Tell 10.0.0.15"*
- erhält Unicast-Reply: *„10.0.0.1 is at 44:88:22:22:88:44"* → Cache schreiben
- sendet Paket mit IP „*173.252.110.27*" an MAC-Adresse des Standard-Gateways „*44:88:22:22:44:88*" → Standardgateway vermittelt weiter …

Ablauf


Versuchter Paketversand an bestimmte IP
Ziel-IP im gleichen Subnetz?
nein
ja
Std-Gateway MAC im ARP-Cache?
nein
ja
ARP-Req. an Standard-Gateway-IP senden
ARP-Reply von Standard-Gateway-IP erhalten
ARP-Cache aktualisieren
Paketversand an Standard-Gateway-MAC
Ziel-MAC im ARP-Cache?
nein
ja
ARP-Request an Ziel-IP senden
ARP-Reply von Ziel-IP erhalten
ARP-Cache aktualisieren
Paketversand an Ziel-MAC

ARP-Arten

	Reverse-ARP	Gratuitous ARP	Proxy ARP	S-ARP
Prinzip	– Zuordnung IP zu MAC – Server mit Datenbank benötigt – RFC 903	– ARP-Request auf eigene IP – zum Test auf IP-Doppelung	– Verbindung von Subnetzen über Proxy-Server – identische MAC-Adressen für verschiedene IPs	– Asymmetrische Verschlüsselung mit Generieren von Schlüssenpaar für jeden Host
Anwendung	– für Systeme ohne Cache – Hauptsächlich von DHCP abgelöst	– bei Tausch des Netzwerkinterfaces – Im Zuge von DHCP – beim Booten	– für alte Rechner, die kein Subnetting können	– bei erhöhtem Sicherheits-bedarf

ARP: Probleme & Sicherheit

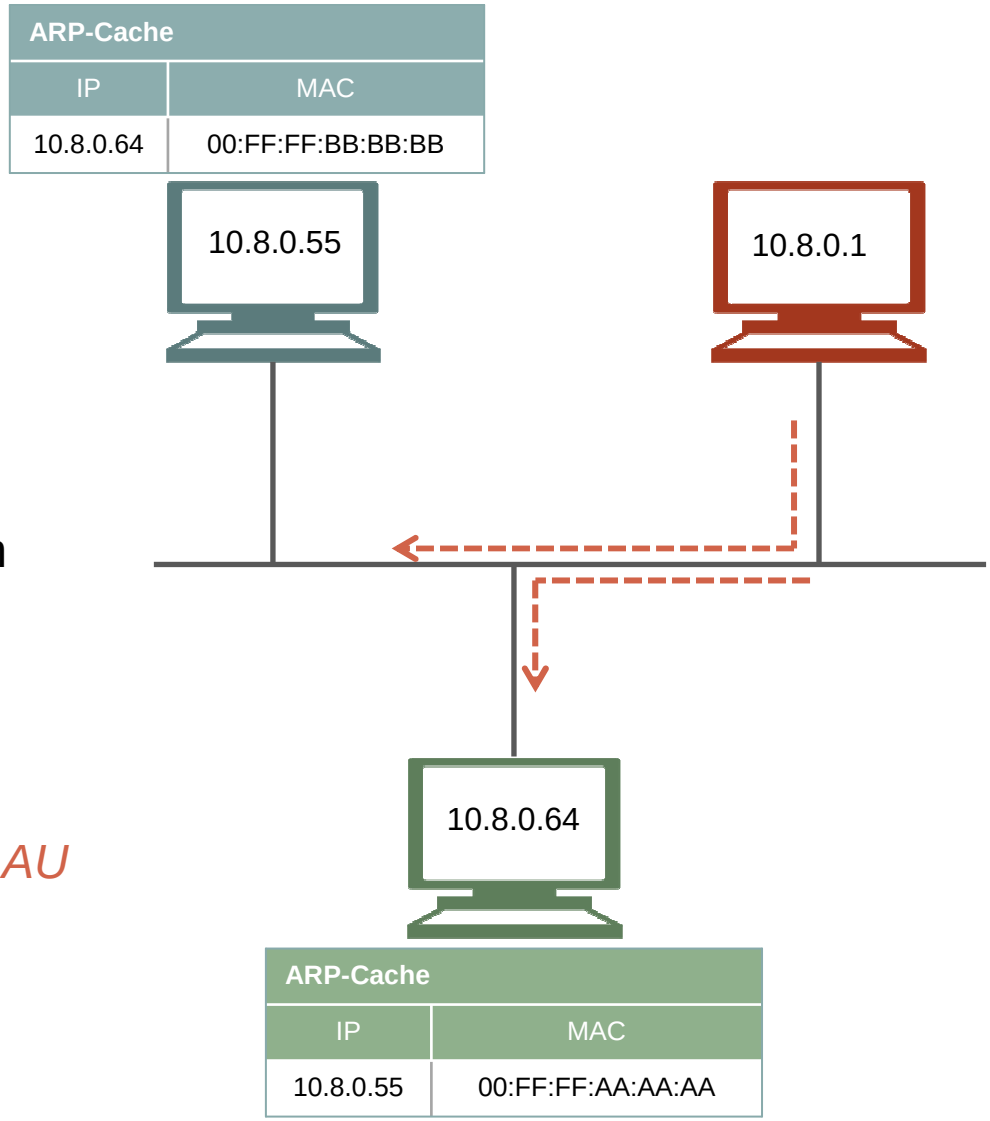
- Keine Überprüfung der korrekten Zuordnung möglich
  - Manipulation des ARP-Caches möglich
- Während Host offline ist und ARP-Cache des Clients noch nicht aktualisiert, treten Fehler auf
- Fehler sind nicht leicht zu erkennen, da ARP im Hintergrund abläuft

Man-In-The-Middle-Attacke

ARP-Cache der Rechner sind korrekt belegt

ROT will Verbindung zwischen **BLAU** und **GRÜN** kapern, um Daten mitzulesen

ROT schickt sekündlich gefälschte ARP-Replies an BLAU und GRÜN → Tragen diese in ihren ARP-Cache ein



Fazit

- Im Hintergrund ablaufendes, bewährtes Protokoll für IPv4
- Problematik Sicherheit
- Wird durch Neighbor-Discovery-Protocol in IPv6 abgelöst (http://tools.ietf.org/html/rfc4861)