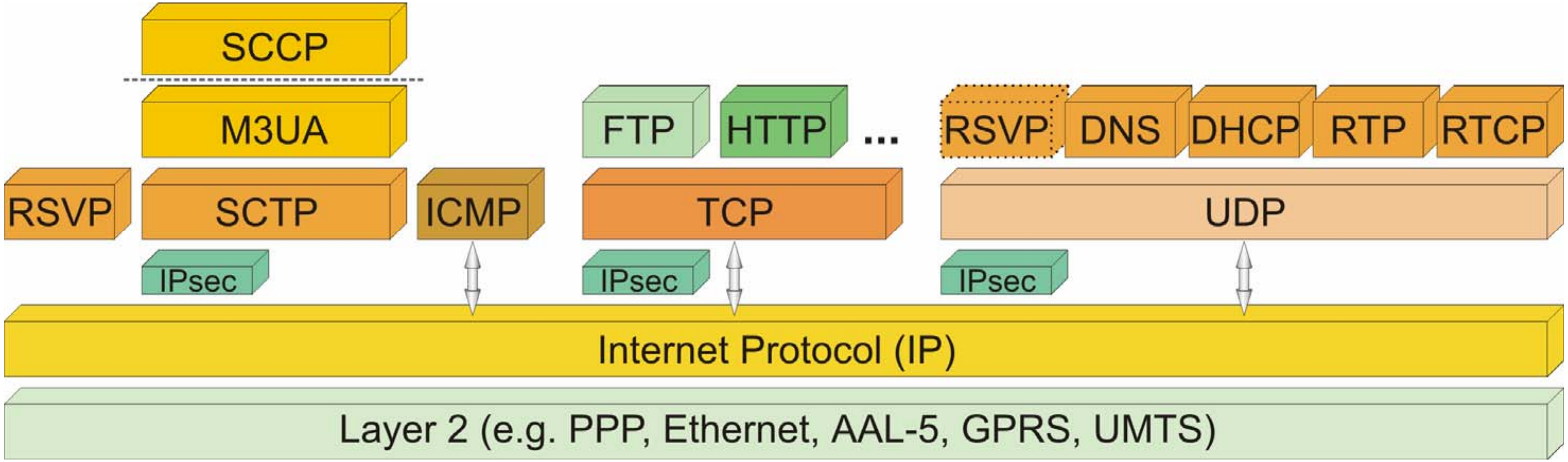
Introducing the IP-Protocol Stack



Introducing the IP-Protocol Stack

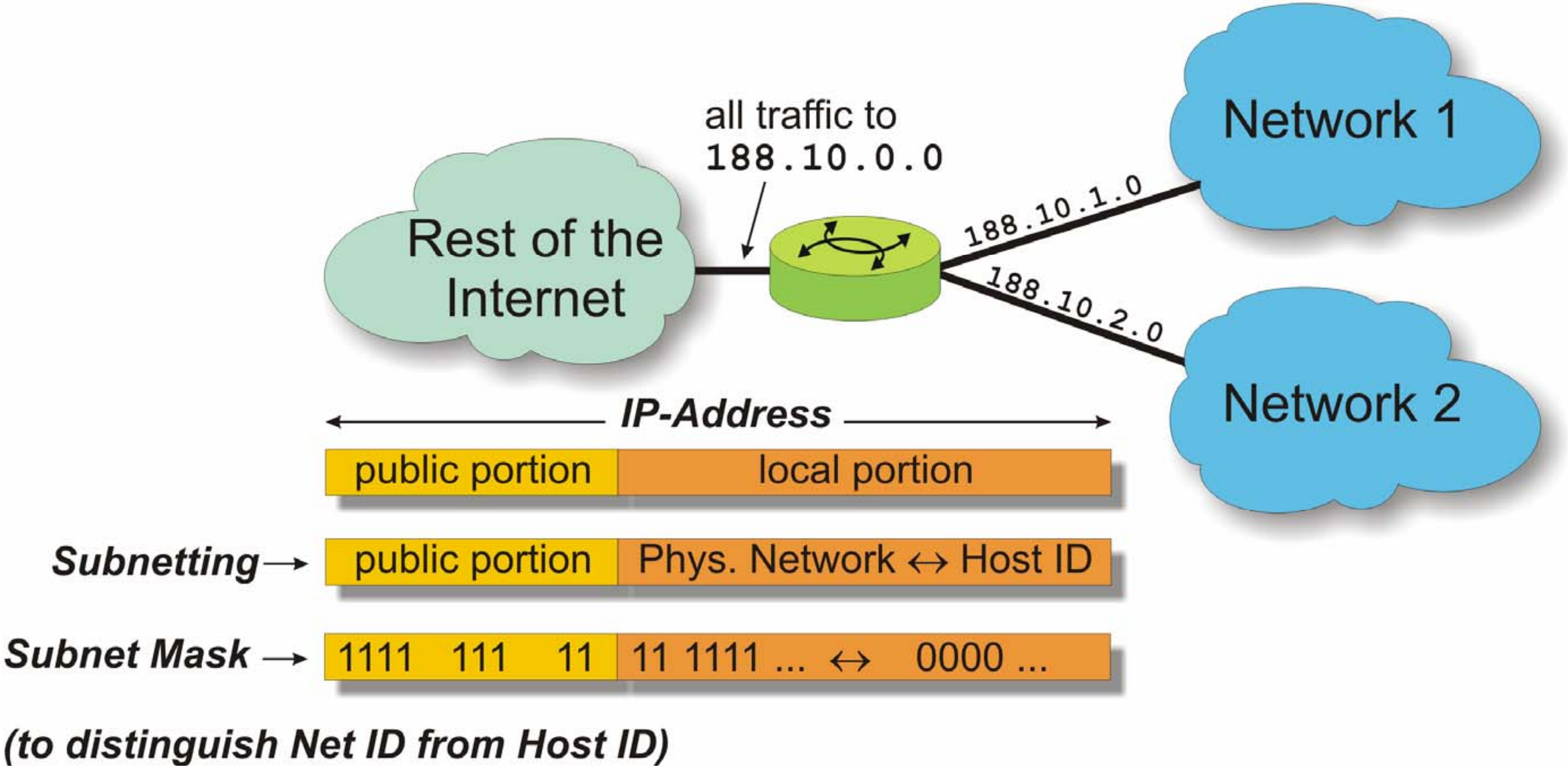
A simplified view of the IP-protocol stack is provided in the figure. The Internet Protocol (IP) itself can be located on top of almost any available layer 2-protocol like for example:

- ⇒ PPP (⬄ Point-to-Point Protocol / RFC 1661)
- ⇒ Ethernet (⬄ IEEE 802.3)
- ⇒ AAL-5 (⬄ ATM / ITU-T I.363.5 (6), Q.2110 (4)
- ⇒ or even on mobile bearers like GPRS or UMTS (⬄ 3GPP recommendations).

This flexibility of IP makes it the preferred network layer solution of the starting 21st century.

However, the term IP-protocol stack rather relates to IP and the higher layers which make use of the IP's networking capabilities. In the following sub-clauses we will take a more detailed look at the IP itself and the protocols on top of it.

Subnet-Addressing



	public portion	local portion
Subnetting →	public portion	Phys. Network ↔ Host ID
Subnet Mask →	1111 111 11	11 1111 ... ↔ 0000 ...

(to distinguish Net ID from Host ID)

Subnet-Addressing

The original classification of IP-addresses into Class A, B and C addresses provides unfortunately only a limited flexibility for sharing e.g. a single Class B address range between two or more different networks. However, there is obviously various enterprises which require an Class B address range since they can't be suited with a single Class C address range. However, these enterprises frequently do not require all the 65534 Host ID's that one Class B Net ID offers.

It is obvious that a means needed to be standardized to allow two or more networks to share a single Class B Net ID. The specific problem in that respect is routing over the internet. This problem can be resolved by one router (see figure) which interconnects both networks towards the internet. From the point of view of the internet there is only one network (188.10.0.0).

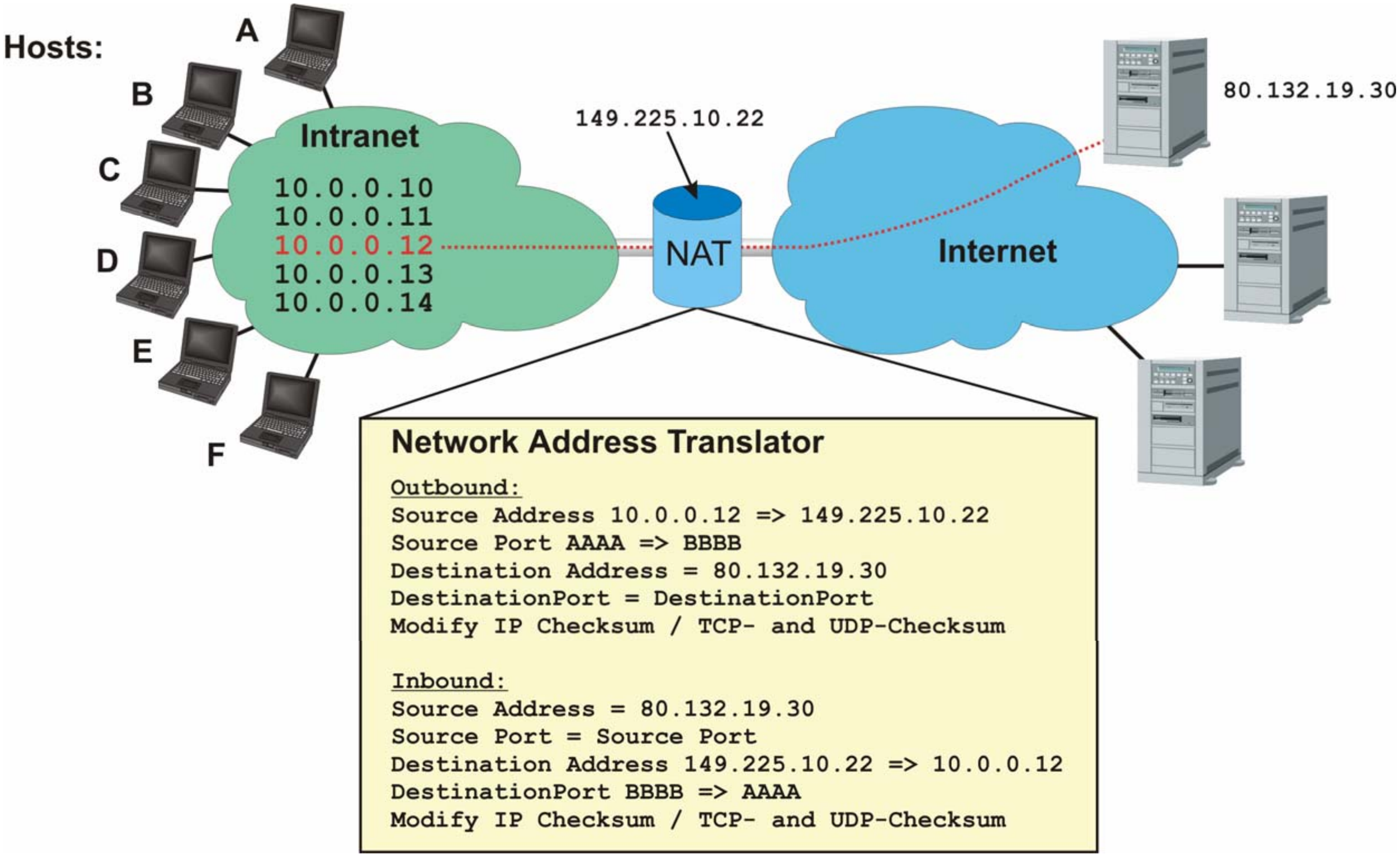
It is the router which will evaluate the following N bits of the local portion to determine to which physical network (either 188.10.1.0 or 188.10.2.0) the IP-frame needs to be sent.

Subnetting allows a large organization (e.g. a holding company) to provide distinctive fractions of their IP-address space to their different facilities. In that respect, subnetting is not restricted to a specific number of bits for the identification of the physical network or the Host ID.

If subnet-addressing is used a so called subnet mask identifies which bits of the IP-address relate to the Net ID (all '1') and which ones relate to the Host ID (all '0')

[RFC 950]

Principles of Network Address Translation



Network Address Translator

<u>Outbound:</u>

```
Source Address 10.0.0.12 => 149.225.10.22
Source Port AAAA => BBBB
Destination Address = 80.132.19.30
DestinationPort = DestinationPort
Modify IP Checksum / TCP- and UDP-Checksum
```

<u>Inbound:</u>

```
Source Address = 80.132.19.30
Source Port = Source Port
Destination Address 149.225.10.22 => 10.0.0.12
DestinationPort BBBB => AAAA
Modify IP Checksum / TCP- and UDP-Checksum
```

Principles of Network Address Translation

In general, Network Address Translation will use the same IP-address for multiple intranet hosts. In that respect, the following characteristics of NAT is very important:
NAT requires that any transaction is initiated by a host within the intranet. This also applies for TCP-connections. In that respect, NAT also provides a firewall function: No external application can initiate a communication process to an internal user.
How does NAT work? Let us examine our example in the figure:

- **Outbound Traffic**
  - ⇒ Whenever a host within the intranet needs to communicate with an external host, the intranet host will build a UDP/IP-frame or TCP/IP-frame with its own private IP-address and source port (e.g. 10.0.0.12 / Source Port = AAAA) and the destination IP-address and destination port number of the external host (e.g. 80.132.19.30 / DestinationPort).
  - ⇒ This UDP/IP-frame or TCP/IP-frame will be routed to the NAT-router.
  - ⇒ The NAT-router will perform the following functions:
    1. Replace the private source IP-address of the internal host by its own public IP-address 149.225.10.22.
    2. Replace the source port number AAAA by another source port number BBBB.
    3. Adjust the checksums for TCP or UDP and for the IP-header according to the performed changes.

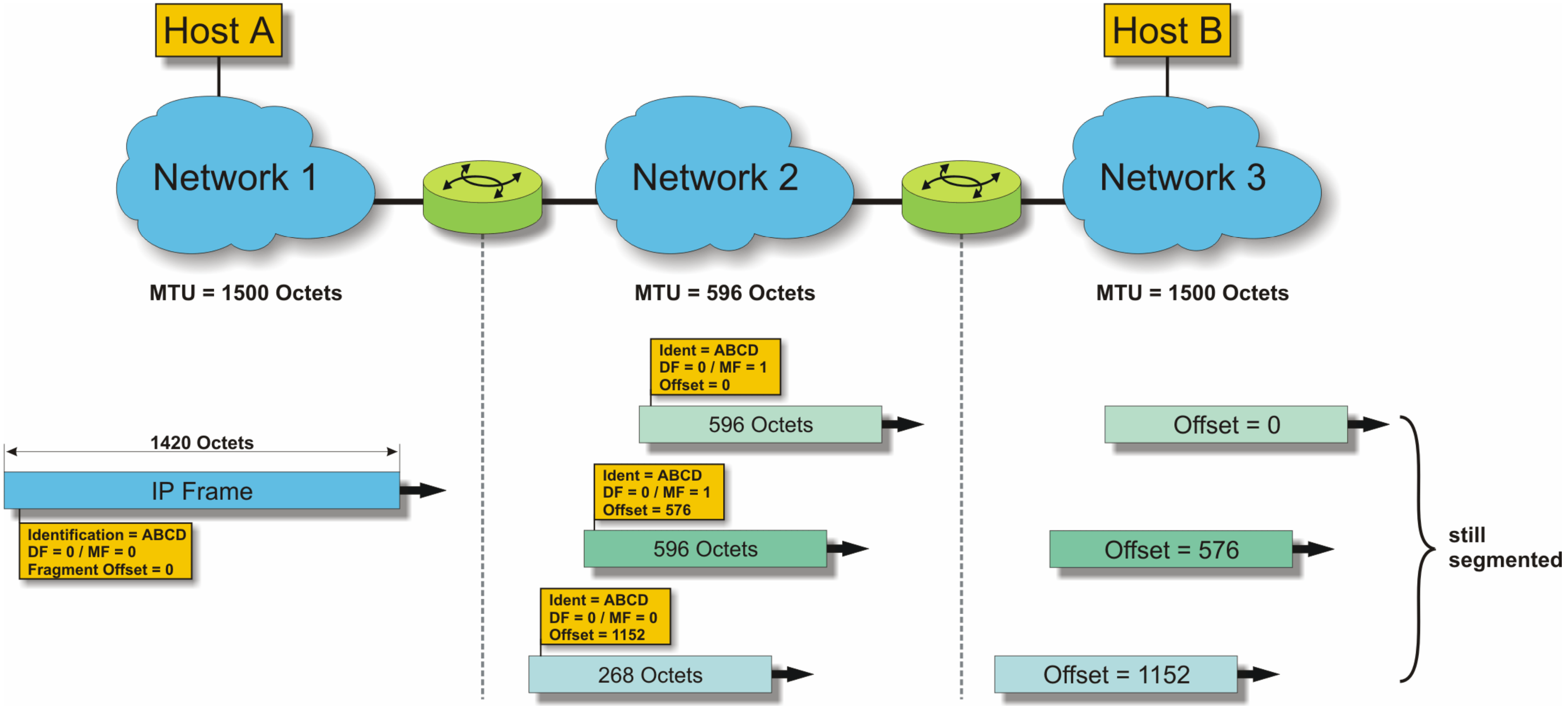
In addition, the NAT-router will enter a new pair of tuples, consisting of private source IP-address 10.0.0.12 / original source port AAAA and public IP-address 149.225.10.22 / new source port number BBBB) into its database to be able to map an incoming UDP/IP- or TCP/IP-frame to the original request.

- **Inbound Traffic**
  - ⇒ When the external host responds to the request of the internal host, this response will obviously be destined to the public IP-address 149.225.10.22 and destination port BBBB of the NAT-router. The NAT-router will look up its database to match the tuple 149.225.10.22 / BBBB to the requesting intranet host which is identified by the related tuple 10.0.0.12 / AAAA.
  - ⇒ Consequentially, the NAT-router will replace the destination IP-address 149.225.10.22 by the private IP-address 10.0.0.12 and the destination port BBBB by AAAA.. Obviously, the checksums for TCP or UDP and for the IP-header need to be changed accordingly.
- **Conclusion:**

By using NAT and private IP-addressing, app. 64,000 users can operate on a single public IP-address. They are discriminated through the 16 bit long port number of which the values up to $1{,}024_{dec}$ are the already assigned well known port numbers which cannot be used for NAT.

[RFC 2633 / RFC 2766]

Fragmentation Control in IP



Fragmentation Control in IP

When two IP-modules communicate through the internet, an unknown number of networks lies between the two hosts. These networks most likely will not provide identical layer 2 frame sizes (⇔ referred to as MTU (Maximum Transmit Unit). Therefore fragmentation of an IP-frame may become necessary at an intermediate router.
In our example, network 1 and 3 support an MTU of 1500 octets but the intermediate network 2 only supports 596 octets. Accordingly, the two routers need to fragment the outgoing IP-frames (with a length of 1420 octets) into smaller slices which fit the MTU of the intermediate network.

Note that segmentation may only occur in units of 8 octets. That is, a router may not segment a 1500 octet long IP-frame into 5 units of 300 octets each, because 300 divided by 8 has no integer result (a possible proceeding in this case would be to segment into 5 units of 296 octets each plus a 6th segment with only 20 octets.

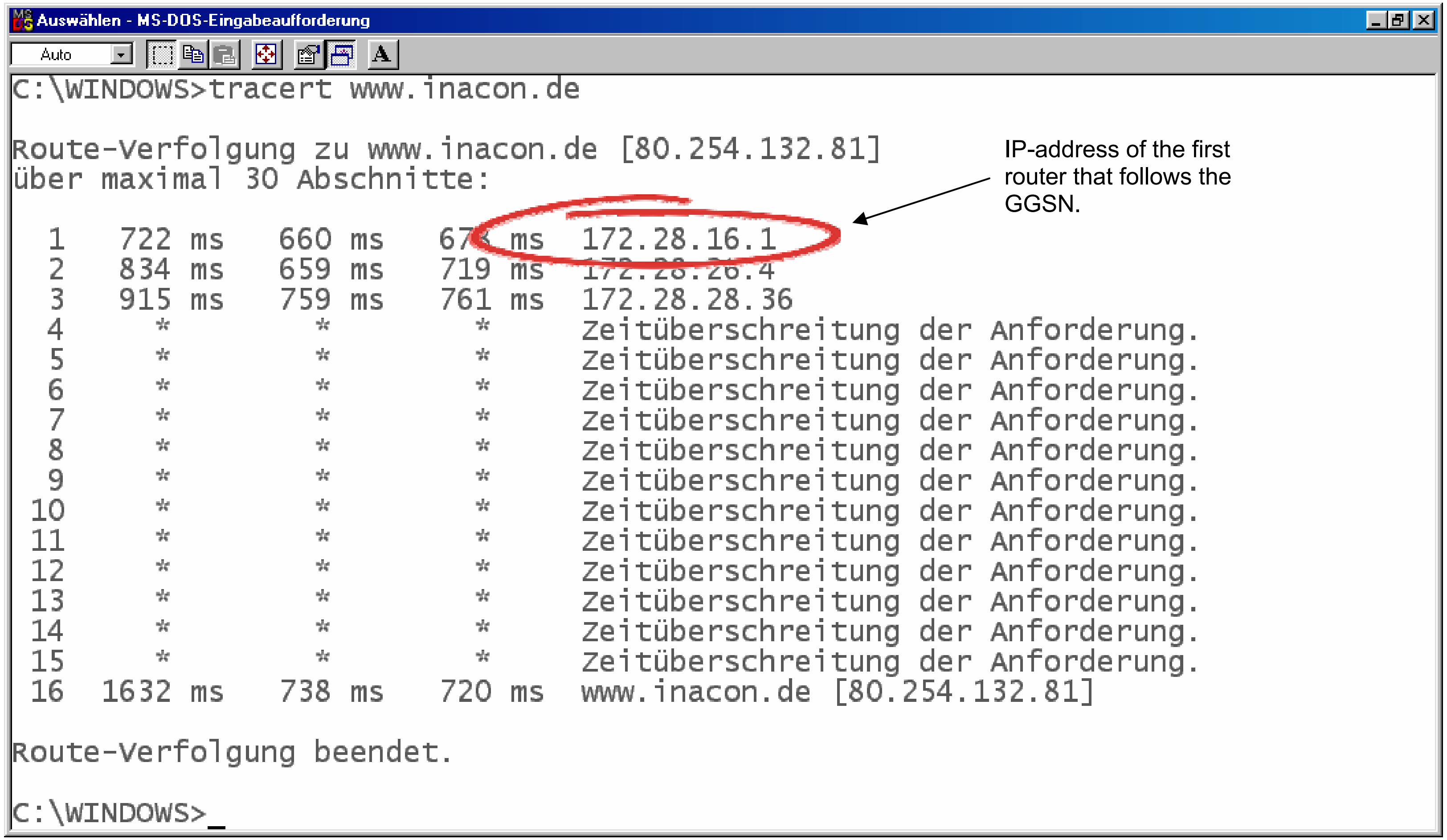
In our example, the 1420 octet long IP-frame is segmented into two segments of 576 octets each plus a last segment of 248 octets. The figure illustrates the setting of DF, MF, *Identification* and *Fragment Offset*. Please note that the actual length of the three IP-frames is 20 octets longer because of their individual IP-headers.
The resulting three IP-frames will be using the same setting of *Identification* but by means of the *Fragment Offset* and the setting of the MF-bit in the different segments the receiver is able to correctly reassemble the original IP-frame.

If an IP-frame needed to be segmented on its way through the internet, re-segmentation will only occur at the final destination. If one or more segments of an IP-frame need to be segmented another time, this occurs the same way as illustrated before.

Note: The default MTU on the internet is 596 octets. This MTU has to be supported by all routers.

[RFC 815]

Use Trace Route to Determine the IP-Address of the 1st Router



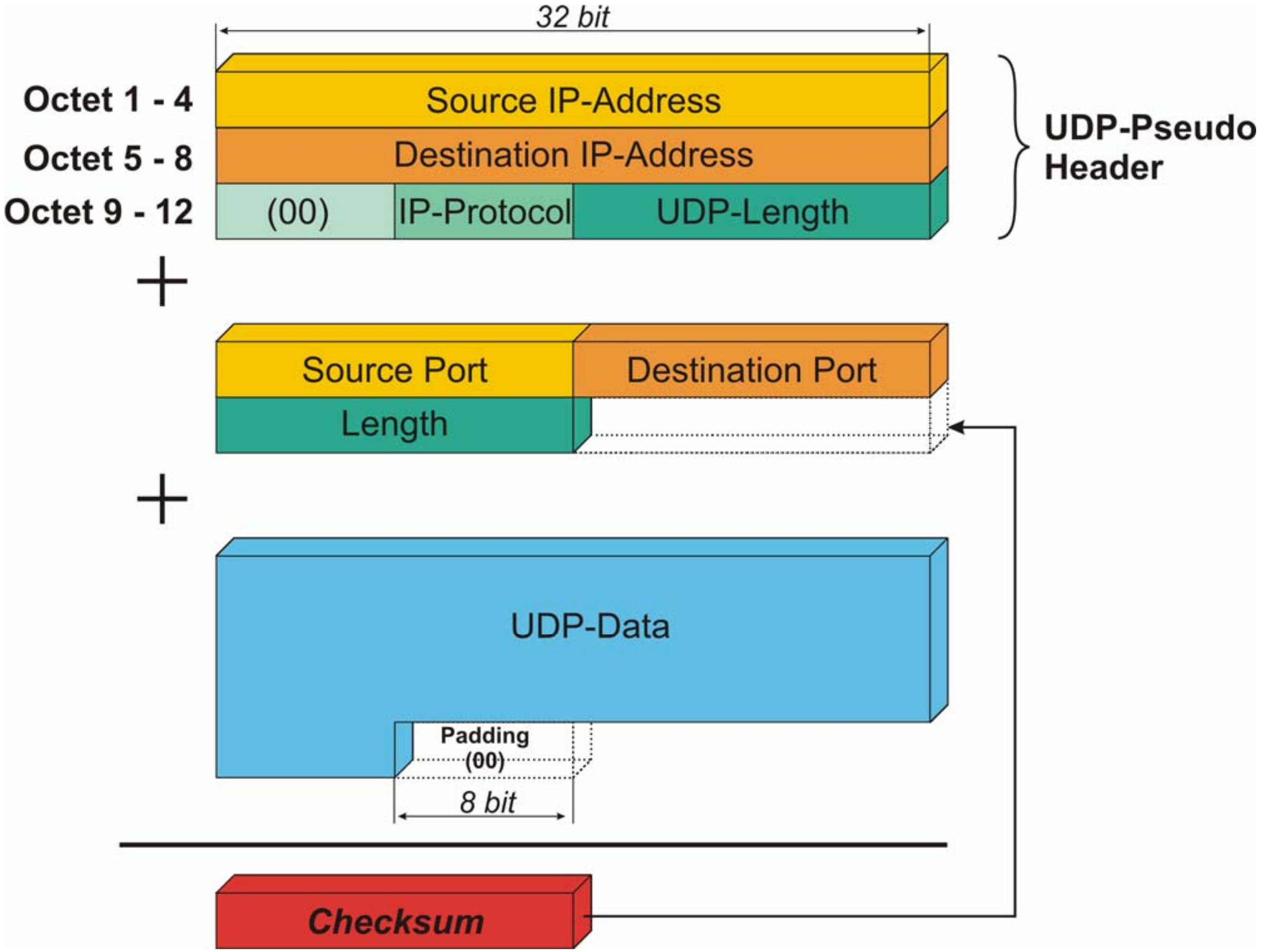
```
C:\WINDOWS>tracert www.inacon.de

Route-Verfolgung zu www.inacon.de [80.254.132.81]
über maximal 30 Abschnitte:

  1    722 ms   660 ms   67[illegible] ms  172.28.16.1
  2    834 ms   659 ms   719 ms  172.28.26.4
  3    915 ms   759 ms   761 ms  172.28.28.36
  4     *        *        *     Zeitüberschreitung der Anforderung.
  5     *        *        *     Zeitüberschreitung der Anforderung.
  6     *        *        *     Zeitüberschreitung der Anforderung.
  7     *        *        *     Zeitüberschreitung der Anforderung.
  8     *        *        *     Zeitüberschreitung der Anforderung.
  9     *        *        *     Zeitüberschreitung der Anforderung.
 10     *        *        *     Zeitüberschreitung der Anforderung.
 11     *        *        *     Zeitüberschreitung der Anforderung.
 12     *        *        *     Zeitüberschreitung der Anforderung.
 13     *        *        *     Zeitüberschreitung der Anforderung.
 14     *        *        *     Zeitüberschreitung der Anforderung.
 15     *        *        *     Zeitüberschreitung der Anforderung.
 16   1632 ms   738 ms   720 ms  www.inacon.de [80.254.132.81]

Route-Verfolgung beendet.

C:\WINDOWS>_
```

IP-address of the first router that follows the GGSN.

Use Trace Route to Determine the IP-Address of the 1st Router

Intentionally left blank

UDP-Pseudo Header and UDP-Checksum



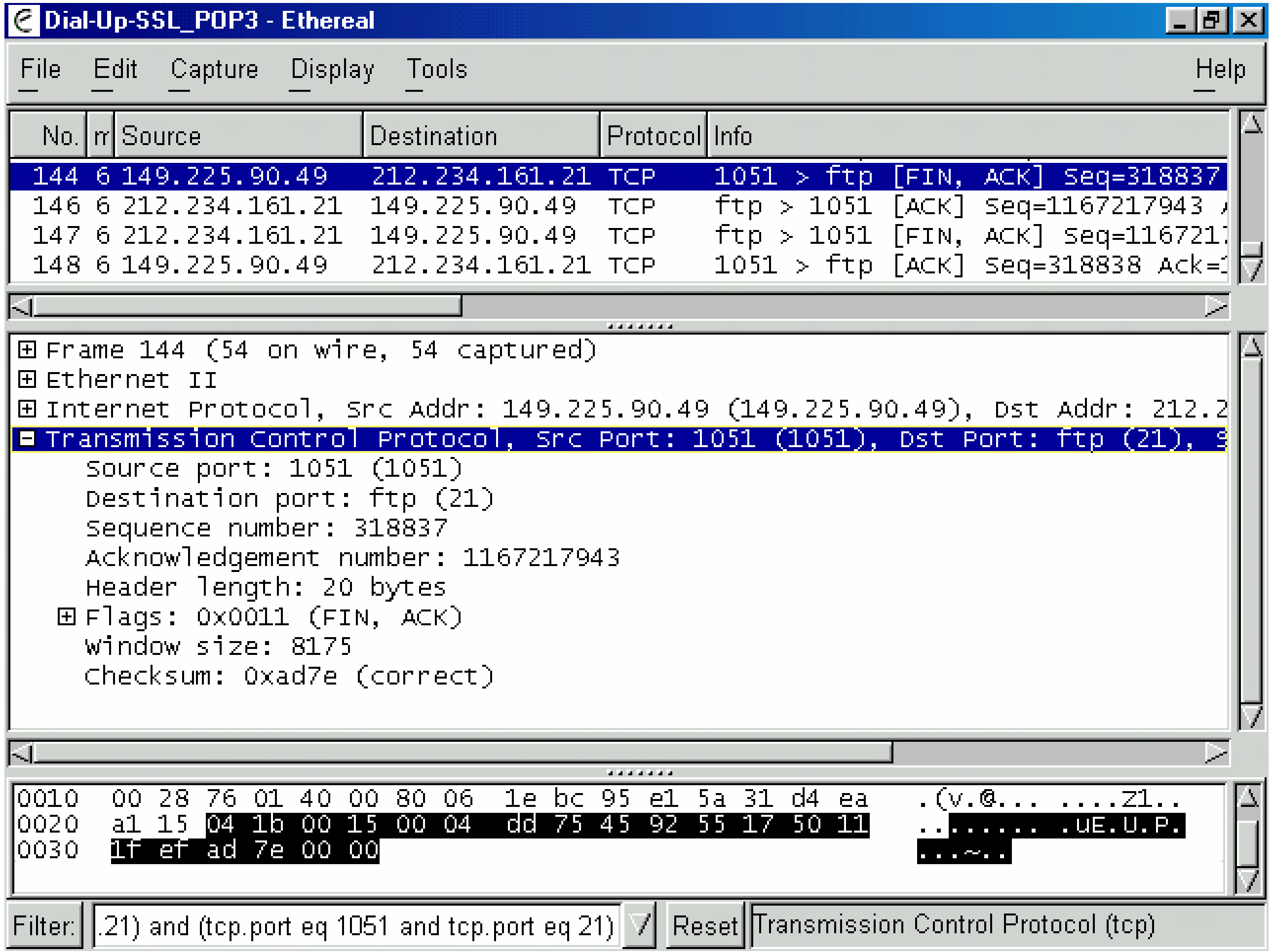
UDP-Pseudo Header and UDP-Checksum

As the figure illustrates, the UDP-checksum does not only protect the UDP-header and the UDP-data part. It rather also considers part of the IP-header. This is achieved through the definition of a 12 octet long UDP-pseudo header which consists of:

- ⇒ The source and destination IP-addresses (2 x 32 bit)
- ⇒ One octet which is fixed coded to ‘0’
- ⇒ The IP-header Protocol field (8 bit)
- ⇒ The length of the UDP-frame (UDP-header + UDP-data) (16 bit)

This UDP-pseudo header is binary added to the “real” UDP header and the UDP-data portion. The result is fed into the checksum field within the UDP-header. Please note that the final padding octet (‘0’) is only required and appended, if the data field contains an odd number of octets (the checksum algorithm is tailored to 16 bit words.

(1) Example for TCP Connection Release



```
No. rr Source          Destination     Protocol Info
144 6 149.225.90.49   212.234.161.21  TCP      1051 > ftp [FIN, ACK] Seq=318837
146 6 212.234.161.21  149.225.90.49   TCP      ftp > 1051 [ACK] Seq=1167217943 A
147 6 212.234.161.21  149.225.90.49   TCP      ftp > 1051 [FIN, ACK] Seq=116721
148 6 149.225.90.49   212.234.161.21  TCP      1051 > ftp [ACK] Seq=318838 Ack=1
```

```
⊞ Frame 144 (54 on wire, 54 captured)
⊞ Ethernet II
⊞ Internet Protocol, Src Addr: 149.225.90.49 (149.225.90.49), Dst Addr: 212.2
⊟ Transmission Control Protocol, Src Port: 1051 (1051), Dst Port: ftp (21), S
     Source port: 1051 (1051)
     Destination port: ftp (21)
     Sequence number: 318837
     Acknowledgement number: 1167217943
     Header length: 20 bytes
   ⊞ Flags: 0x0011 (FIN, ACK)
     Window size: 8175
     Checksum: 0xad7e (correct)
```

```
0010  00 28 76 01 40 00 80 06  1e bc 95 e1 5a 31 d4 ea   .(v.@... ....Z1..
0020  a1 15 04 1b 00 15 00 04  dd 75 45 92 55 17 50 11   ........ .uE.U.P.
0030  1f ef ad 7e 00 00                                   ...~..
```

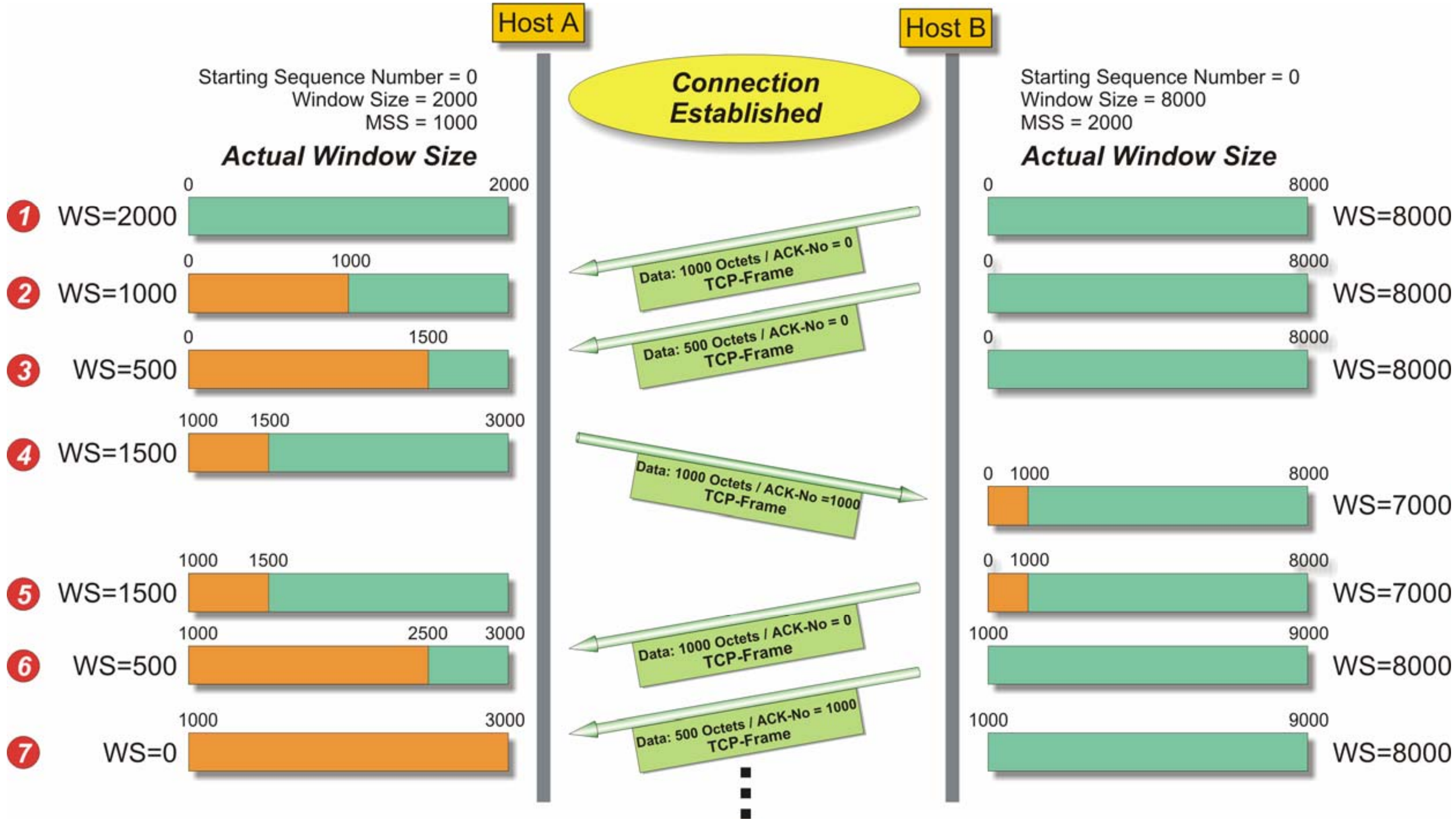
Filter: .21) and (tcp.port eq 1051 and tcp.port eq 21)

Transmission Control Protocol (tcp)

(1) Example for TCP Connection Release

- **The figure illustrates the first of four TCP-frames which are required for connection release.**
  - ⇒ With this first frame, host A indicates to host B a “half-close”.
  - ⇒ Both flags <FIN> and <ACK> are set in this frame.
  - ⇒ No options are included

Sliding Windows in TCP

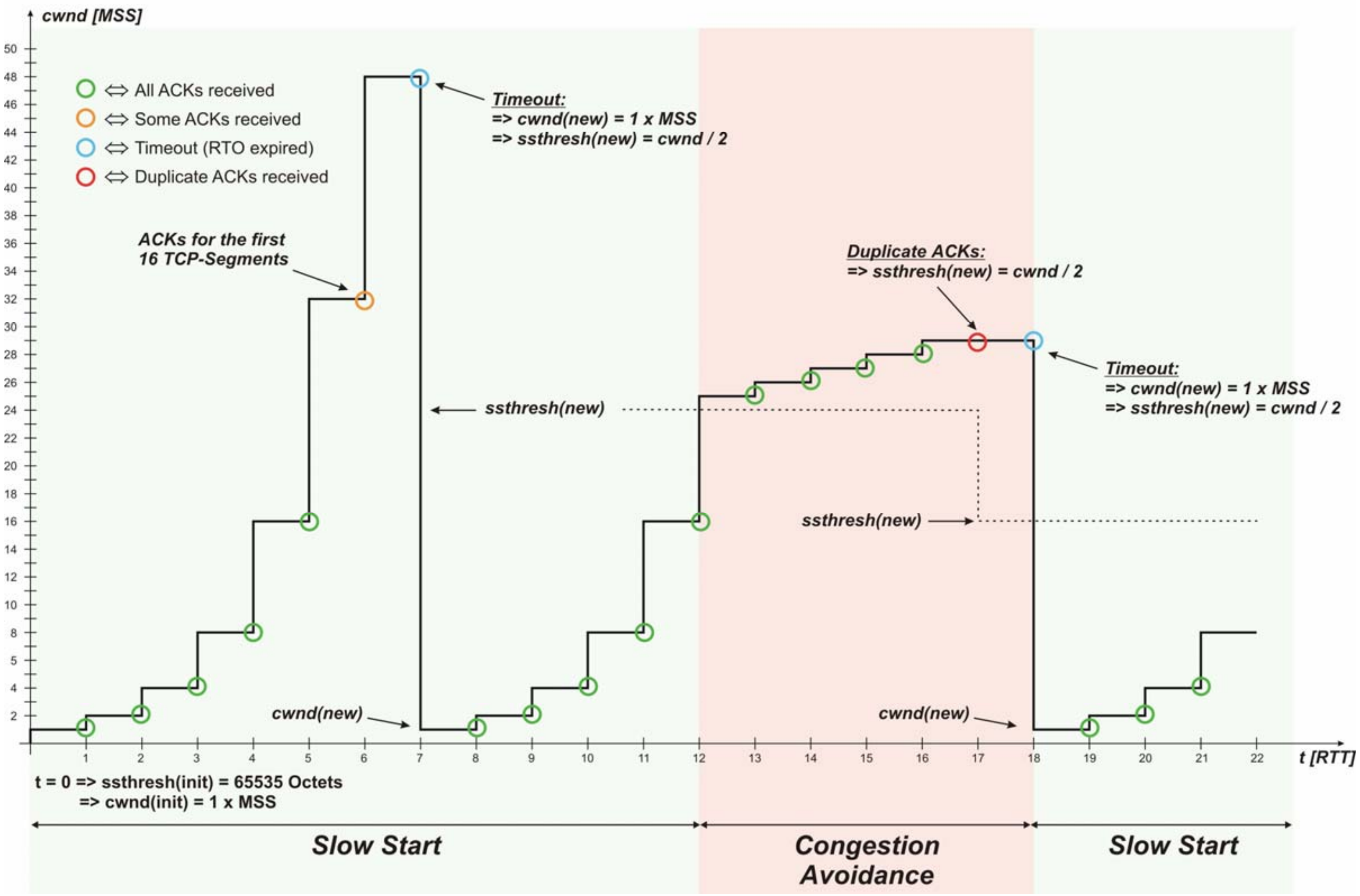


Sliding Windows in TCP

For acknowledged data transmission, TCP deploys a sliding window mechanism which is based on the sequence and acknowledgement numbers as well as the window size as indicated in the TCP-header. The figure shall illustrate the sliding window mechanism in operation. The parameter settings for the starting sequence numbers etc. are provided in the figure. Step 1, 2, 3, … relate to the red index numbers on the left hand side:

- **Step 1**
  Both peers have initialized their window sizes according to the information received from their peer. Host B sends one TCP-frame (1000 Octets) to host A.
- **Step 2**
  Host B has received the first TCP-frame which decreases its window size to 1000 octets (from sequence number 1000 – 1999). In the same instant, host B sends another TCP-frame (500 octets) to host A. Note that the acknowledgement number in both frames from node B is identical, because it points to the next sequence number that host B expects from host A to receive.
- **Step 3**
  Having received the second TCP-frame, the window size of host A is reduced to 500 octets (from sequence number 1500 – 1999).
- **Step 4**
  Host A has finally processed part of the data (sequence number 0 – 999) which it has previously received from host B. Consequentially, the acknowledgement number points to octet number 1000. Note that host A's window size is accordingly increased to 1500 octets (from 1500 to 3000). In addition, host A has data to send to host B (1000 octets). Having received this frame from host A, host B's window size is reduced to 7000 octets.
- **Step 5**
  Host B is again sending data to host A (1000 octets). The acknowledgement number is still '0' because host B has not yet processed the previously received frame.
- **Step 6**
  Having received the previous frame from host B, the window size of host A is reduced to 500 octets. At the same time, host B has processed the data from step 4 and acknowledges all octets up to number 999 (⇔ acknowledgement number = 1000) by sending another frame to host A. Note that this frame includes only 500 octets of data, since the window size of host A is exhausted.
- **Step 7**
  The window size of host A is completely exhausted and an <ACK>-frame will be required to resume transmission.

Slow Start and Congestion Avoidance in Operation



Slow Start and Congestion Avoidance in Operation

Slow start and congestion avoidance operate together as follows:

⇒ Initially, the sender operates in slow start mode: It will only send a single TCP-segment to its peer, waiting for the acknowledgement. When this acknowledgement is received, cwnd is incremented to 2 segments.

⇒ Accordingly, the sender sends these two segments again waiting for the acknowledgement. If the acknowledgement for these two segments is received (possibly as a single acknowledgement for both segments), cwnd is incremented to 4 segments. For every acked TCP-segment, cwnd is incremented by 1 segment.

Slow start therefore provides for an exponential opening of the transmit window.

⇒ Eventually, the number of injected TCP-segments will congest the transmission line, resulting in timeouts. If this occurs, half of the current value of cwnd shall be stored in ssthresh and cwnd is reduced to 1 MSS.

A timeout is an indication for the sending TCP that the capacity of the internet is exceeded and that an intermediate router has discarded the TCP-segments for which the RTO expired.

⇒ What we see is another time slow start in operation, at least as long as cwnd ≤ ssthresh. As soon as cwnd > ssthresh, TCP will change to congestion avoidance operation mode.

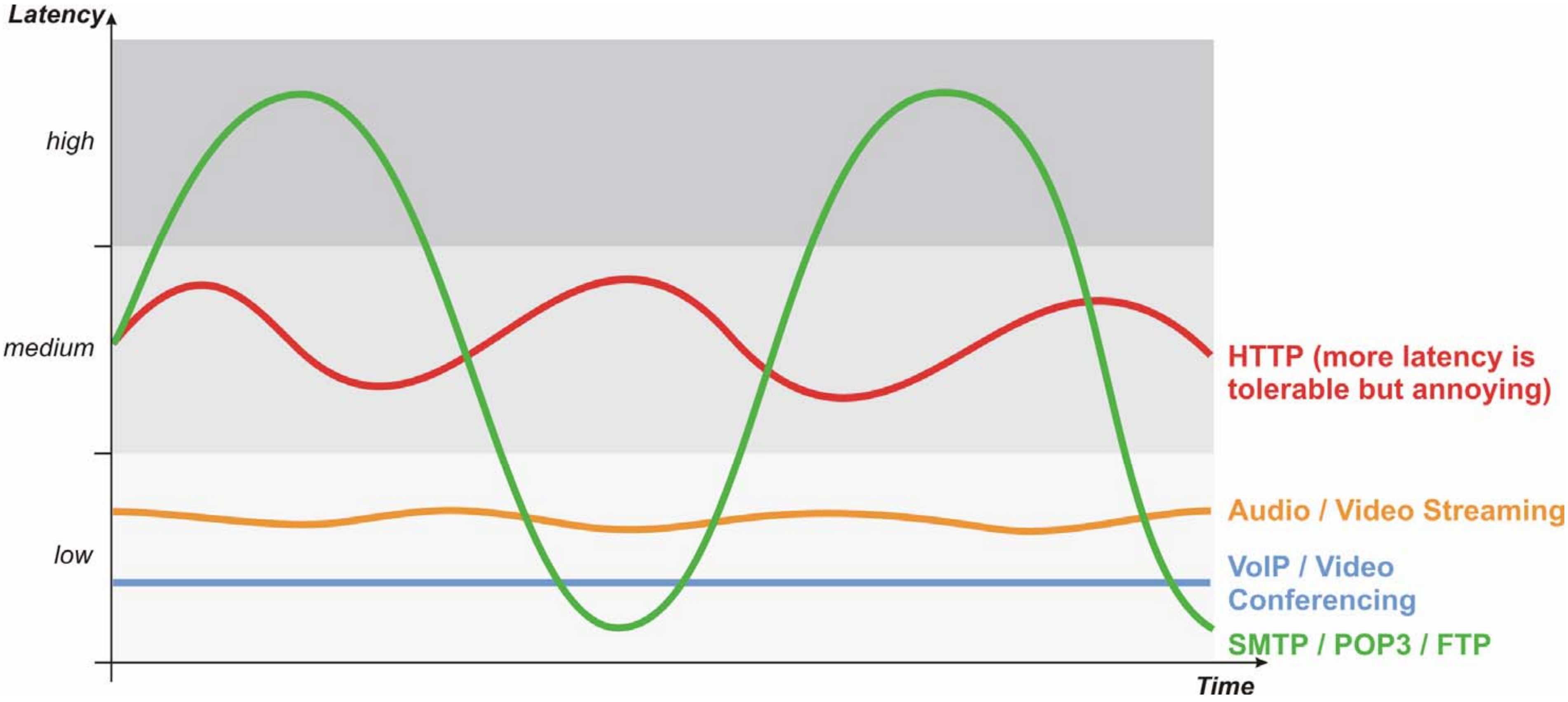
⇒ In congestion avoidance operation mode, the value of cwnd is still incremented but only by a maximum of 1 segment per roundtrip time. As a matter of fact, the increase of cwnd is controlled by the following formula: $cwnd_{new} = cwnd_{old} + (MSS^2 / cwnd_{old}.)$ x (No of received Acks).

⇒ When duplicate acknowledgements are received (⇔ less than 3), TCP shall adjust the value for ssthresh but not for cwnd.

The reception of duplicate acknowledgements is an indication for the sending TCP that TCP-segments have been received out of order by the peer. Still, the transmission line appears to be still open. Therefore, cwnd shall not be reduced but ssthresh is adjusted.

[RFC 2001]

Latency Requirements



Latency Requirements

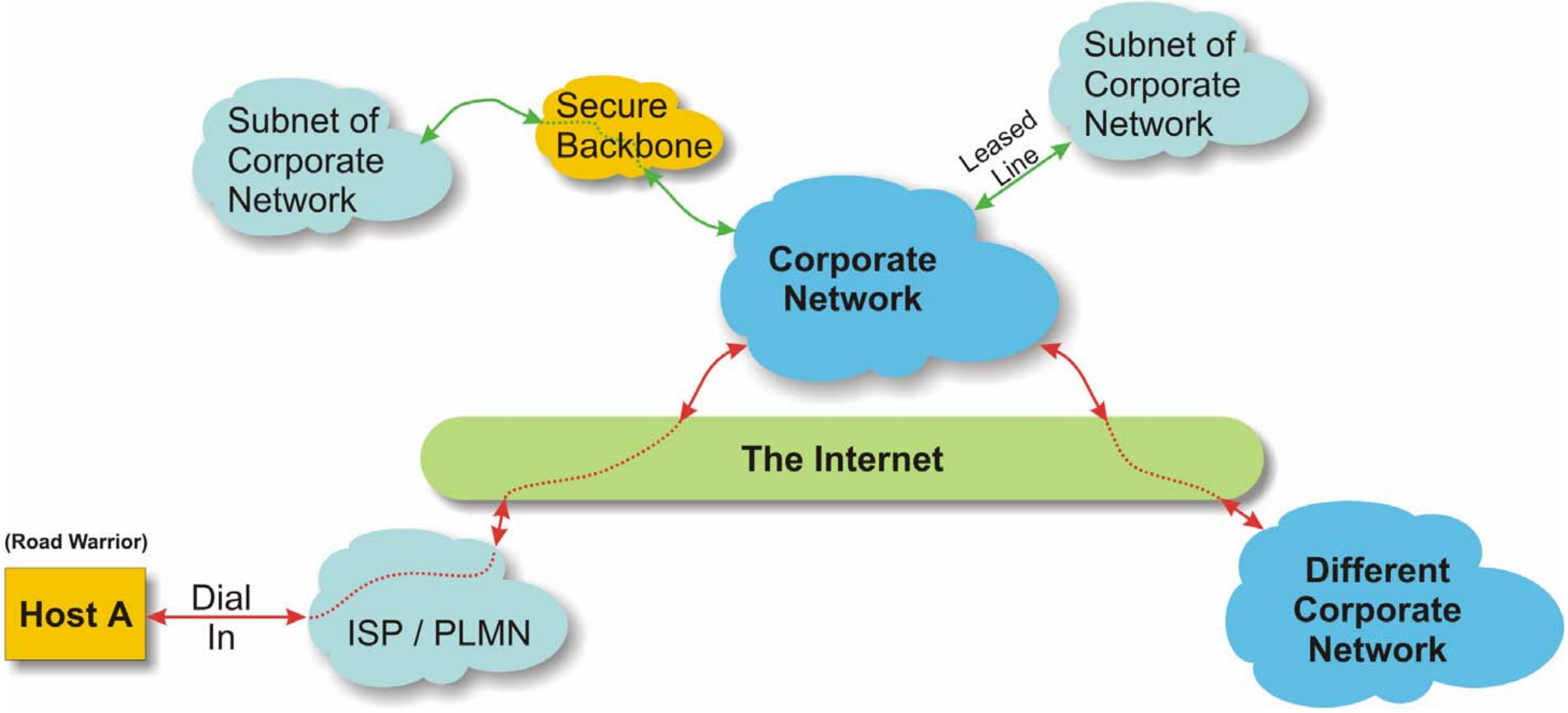
The term “latency” represents the time delay for transferring a data packet from peer A to peer B. The different GPRS-applications come with different requirements regarding latency. The figure illustrates the latency requirements of different application protocols over time.

⇒ For FTP it is for instance less important whether the latency is stable or varies essentially.

⇒ In the other extreme, applications like VoIP or video conferencing require very low and stable delay times.

⇒ HTTP as a standard application is less demanding than e.g. VoIP. However, the user will have a very bad experience when the latency is highly variable.

Typical Network Configurations and Access Types



Subnet of Corporate Network

Secure Backbone

Leased Line

Subnet of Corporate Network

Corporate Network

The Internet

(Road Warrior)

Host A

Dial In

ISP / PLMN

Different Corporate Network

Security Analysis of Typical Network Configurations

The figure illustrates typical network configurations for corporate and private internet traffic. The corporate network in the center has to communicate to various peers which are located outside the borders of the physical central corporate network. We will start with a consideration of the situation starting with the uncritical interconnections:

Subnet ⇐ SECURE BACKBONE ⇒ Central Corporate

Large corporations usually deploy satellite facilities around the world. Each of these satellite facilities will be equipped with their own island network, frequently referred to as a subnet (which has nothing to do with subnet addressing). These subnets usually need to exchange more or less confidential information with the central corporate network. On the upper left side a typical configuration is illustrated: The corporation signs a service contract with some backbone provider which guarantees security on their network. The connection is considered to be safe and is therefore shown as a green line.

Subnet ⇐ LEASED LINE ⇒ Central Corporate

The same security can be obtained through renting a leased line between each satellite network and central corporate.

“Road Warrior” ⇐ DIAL UP / INTERNET ⇒ Central Corporate

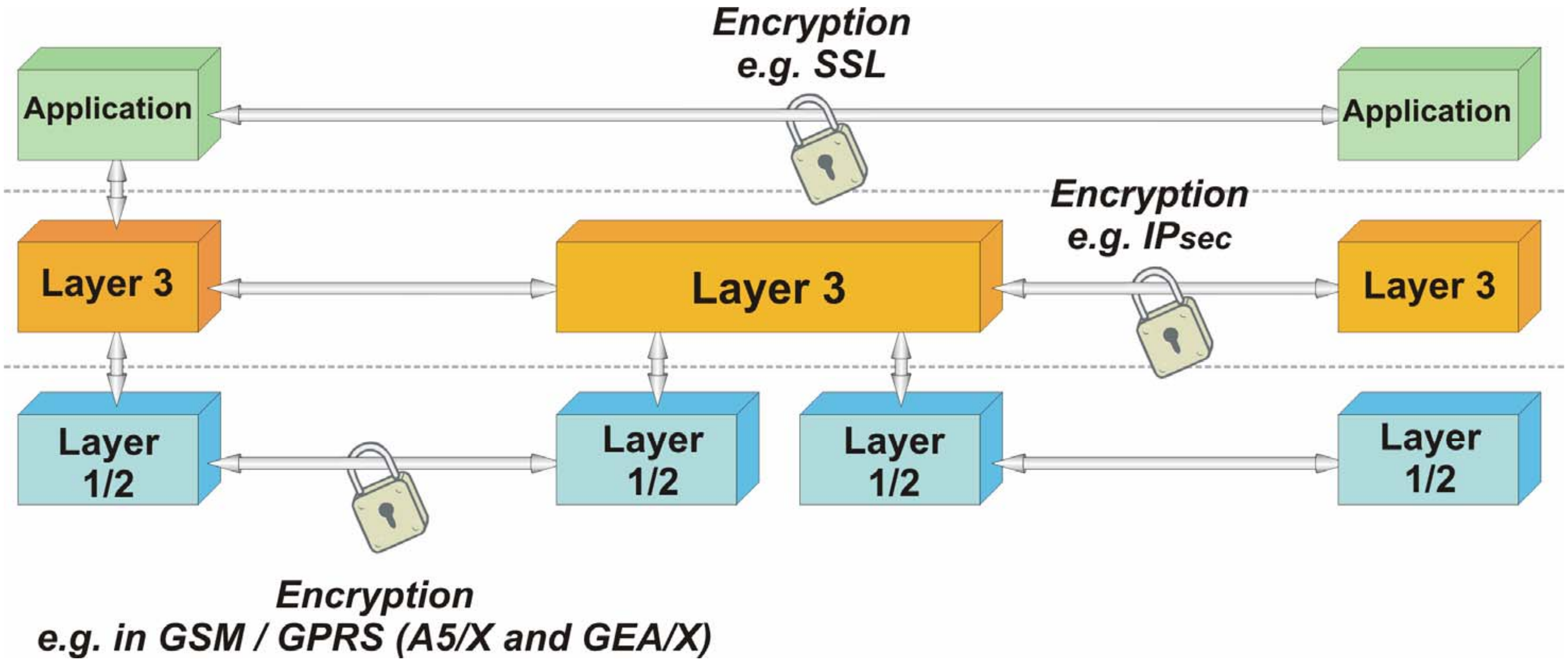
A major concern are the increasing numbers of the so called “Road Warriors”. Road Warriors are usually members of the staff who are working onsite or at home. These users will access the information in the corporate network through some sort of dial-up service which may be offered as a mobile or wired access. This type of network interconnection is achieved through the internet and is therefore highly vulnerable against any kind of security hazards.

Other Corporate Networks ⇐ INTERNET ⇒ Central Corporate

To complete our consideration, corporations usually need to exchange confidential information among each other. It is commercially unfeasible to establish leased lines between these networks or to establish a backbone connection between them. Therefore, this communication also needs to be looked at as highly dangerous.

Leased lines and secure backbones offer excellent security means and are safe against eavesdropping, provided they are properly operated and set up. However, neither leased line nor secure backbones can address the increasing demand for security for the communication of road warriors to the corporate network or between corporate networks.

Alternatives for Network Security



Alternatives for Network Security

When it comes to more security for network communication, there are obviously different options to achieve this goal:

Encryption and Authentication on Layer 1 / 2

Many technologies provide for encryption and authentication on the bearer channels. A very good example is the authentication and encryption technologies of GSM and GPRS. However, since bearer channels are only intermediate, these technologies do not offer end-to-end protection plus there is still a good chance for intruders to tap the traffic within the intermediate nodes.

Despite these disadvantages GPRS encryption should not be disregarded since it adds another level of security for the security sensitive transmission over the air interface.

Encryption and Authentication on the Network Layer

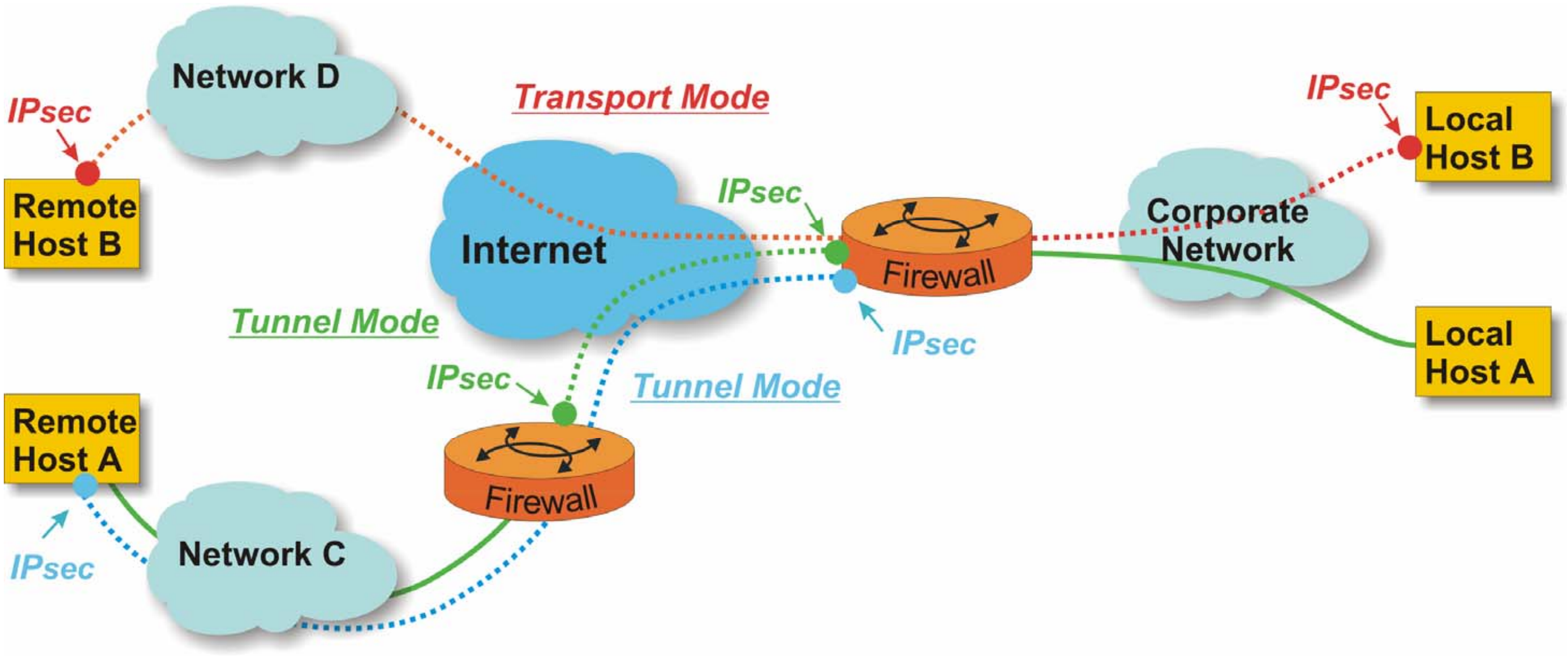
End-to-end authentication and encryption is possible through network layer implementations. The best known example for network layer authentication and encryption is IPsec. The advantage is that intermediate network nodes and the application hard- and software can remain unaware of these changes. IPsec can also be deployed between two networks (⇔ VPN-implementation) which makes it completely transparent to the user.

VPN-solutions will establish a secure IPsec tunnel between two networks that have to communicate through unsecured networks like the internet. In that respect, VPN-solutions have a large cost saving potential compared to leased lines or backbone networks.

Encryption and Authentication on higher layers

Although the effort is modest, there is no means to establish VPN's between any two peers who need to exchange sensitive and confidential information. Consider for instance e-commerce applications where a single user connects to a site. For these applications, it makes more sense to deploy authentication and encryption on the application layer. A typical example is the Secure Socket Layer (SSL) which was originally developed by NETSCAPE Inc. and which is based on digital signatures to ensure confidence.

VPN with IPsec in Tunnel Mode and Transport Mode



VPN with IPsec in Tunnel Mode and Transport Mode

If IPsec and VPN-technology is deployed, the two operation modes transport and tunnel mode need to be distinguished:

VPN with IPsec in Tunnel Mode

The standard mode of VPN-operation is the tunnel mode. In tunnel mode, two network operators have negotiated a service level agreement (SLA) and have exchanged relevant security information. Whenever needed or permanently, an IPsec tunnel is established between the two networks. The end users who communicate between the two networks remain unaware of the security mode and of any details related to security.
Another implementation of tunnel mode is indicated through the blue dotted line: Remote Host A has established an IPsec tunnel to the security gateway of the corporate network. This implementation is almost end-to-end as the communication through the blue link is secured also on its way through network C.

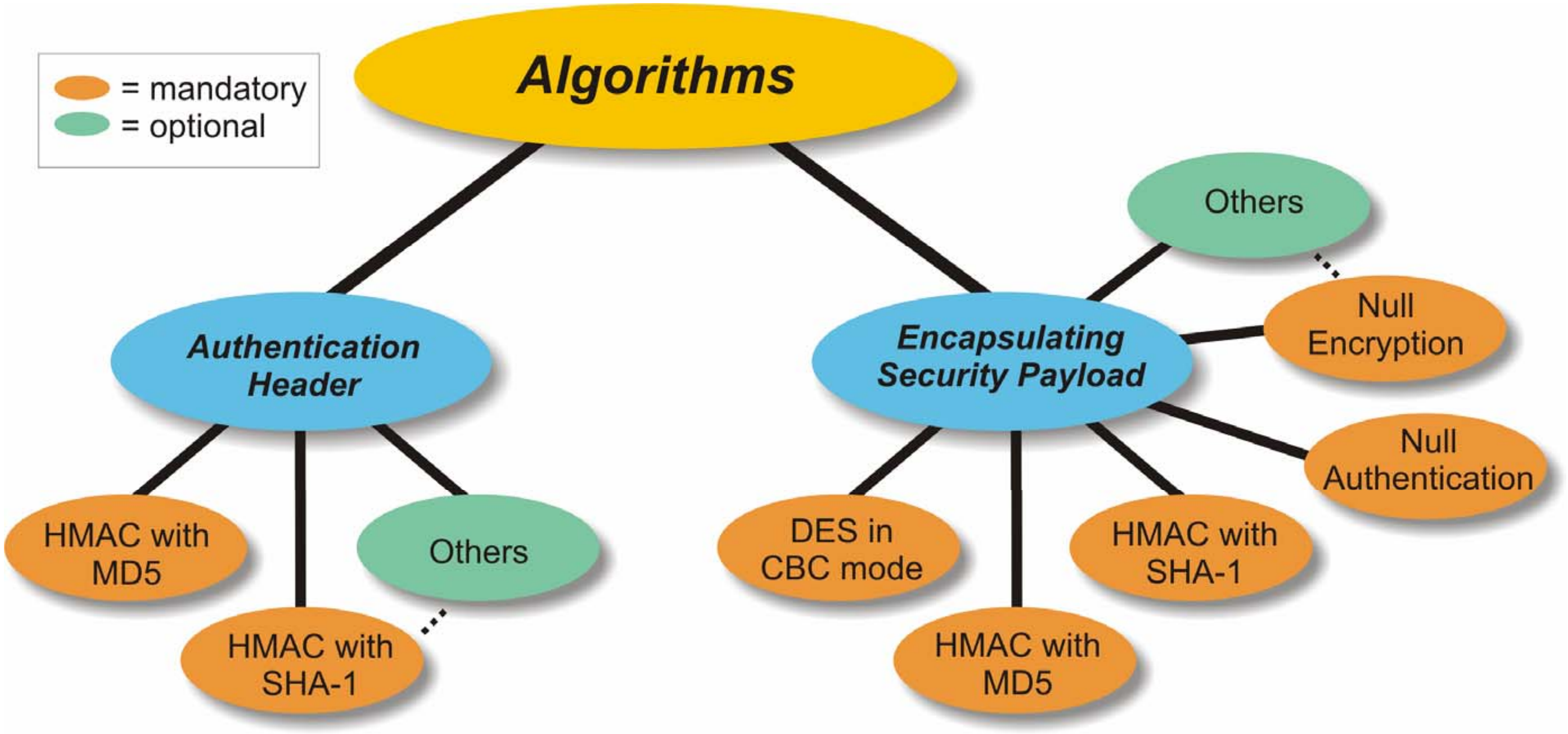
The tunnel mode is very appealing for PLMN operators offering GPRS. For certain subscribers (to be identified through their IMSI), the PLMN-operator offers an IPsec-tunnel to the corporate network of these subscribers. Obviously, there can be as many tunnels to as many corporate networks as necessary.

VPN with IPsec in Transport Mode

In transport mode we really have no VPN at all. As a matter of fact, in transport mode there needs to be an IPsec “tunnel” established between any two hosts on two different networks (in our example it is Remote Host B and Local Host B).

[RFC 2401 / RFC 2402 / RFC 2406]

Algorithms for IPsec



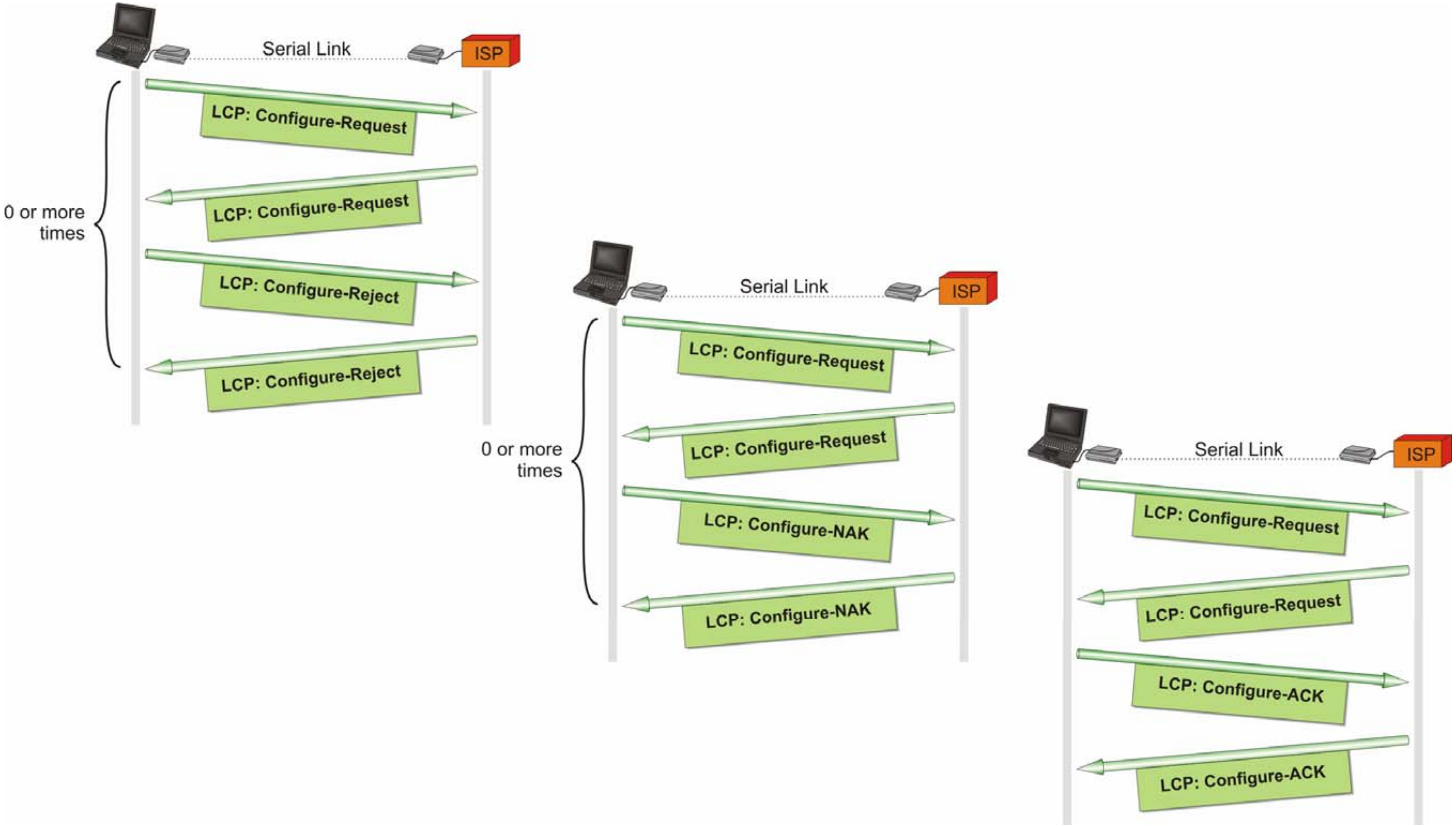
Algorithms for IPsec

The IETF does not restrict an organization to deploy specific algorithms with IPsec. However, the IPsec mandates that certain algorithms shall be supported by *every* IPsec-implementation.

⇒ The authentication algorithms are so called Hash-algorithms which provide a fixed length hash value.
⇒ The encryption algorithms require a secret key as ciphering key and operate on data block sizes of different lengths.

[HMAC with MD 5 ⇒ RFC 2403, MD 5 ⇒ RFC 1321, …]

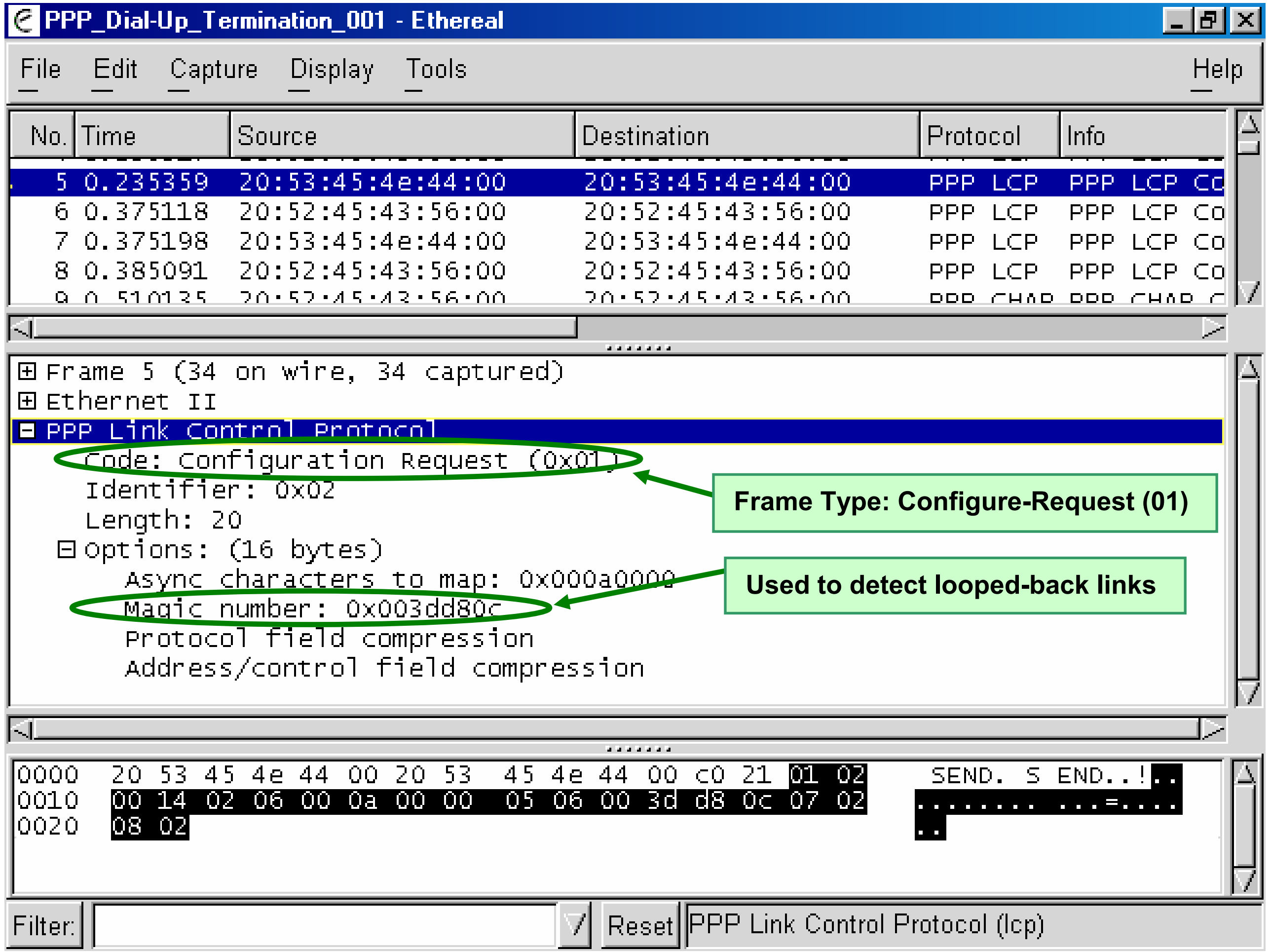
Link Establishment Phase



Link Establishment Phase

The link establishment phase consists of the discussion of various link control parameters like maximum size of the information field (MRU), authentication protocol to be used.

(1) Example for Dial-Up Network Access using the PPP


PPP_Dial-Up_Termination_001 - Ethereal
File
Edit
Capture
Display
Tools
Help
Filter:
Reset
PPP Link Control Protocol (lcp)

```
No. Time      Source             Destination        Protocol  Info
  5 0.235359  20:53:45:4e:44:00  20:53:45:4e:44:00  PPP LCP   PPP LCP Co
  6 0.375118  20:52:45:43:56:00  20:52:45:43:56:00  PPP LCP   PPP LCP Co
  7 0.375198  20:53:45:4e:44:00  20:53:45:4e:44:00  PPP LCP   PPP LCP Co
  8 0.385091  20:52:45:43:56:00  20:52:45:43:56:00  PPP LCP   PPP LCP Co
```

```
⊞ Frame 5 (34 on wire, 34 captured)
⊞ Ethernet II
⊟ PPP Link Control Protocol
    Code: Configuration Request (0x01)
    Identifier: 0x02
    Length: 20
  ⊟ Options: (16 bytes)
      Async characters to map: 0x000a0000
      Magic number: 0x003dd80c
      Protocol field compression
      Address/control field compression
```

```
0000  20 53 45 4e 44 00 20 53  45 4e 44 00 c0 21 01 02    SEND. S END..!..
0010  00 14 02 06 00 0a 00 00  05 06 00 3d d8 0c 07 02   ........ ...=....
0020  08 02                                              ..
```

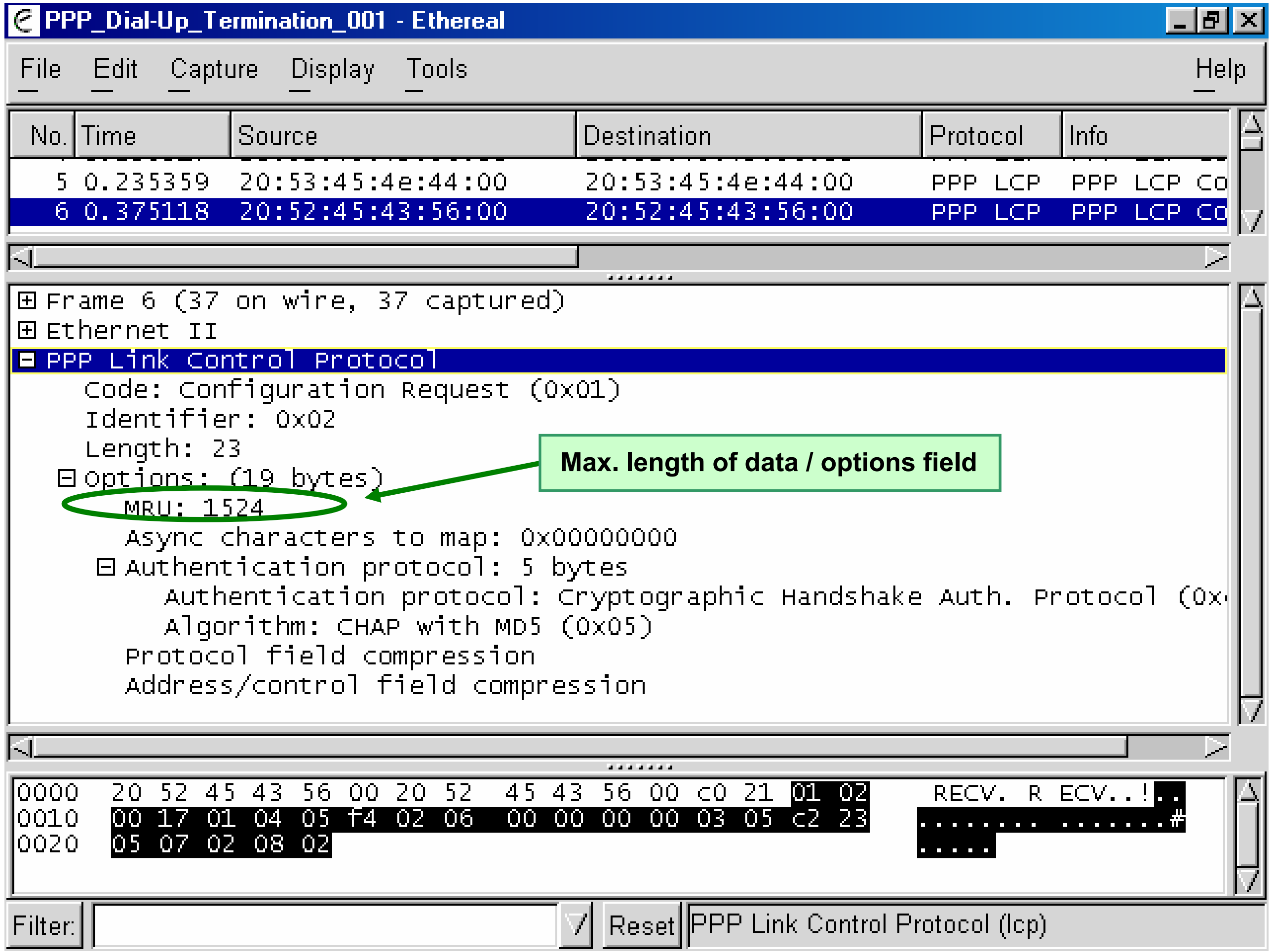
Frame Type: Configure-Request (01)

Used to detect looped-back links

(1) Example for Dial-Up Network Access using the PPP

Intentionally left blank

(2) Example for Dial-Up Network Access using the PPP


PPP_Dial-Up_Termination_001 - Ethereal
File
Edit
Capture
Display
Tools
Help
Filter:
Reset
PPP Link Control Protocol (lcp)

```
No. Time       Source             Destination        Protocol  Info
  5 0.235359   20:53:45:4e:44:00  20:53:45:4e:44:00  PPP LCP   PPP LCP Co
  6 0.375118   20:52:45:43:56:00  20:52:45:43:56:00  PPP LCP   PPP LCP Co
```

```
⊞ Frame 6 (37 on wire, 37 captured)
⊞ Ethernet II
⊟ PPP Link Control Protocol
     Code: Configuration Request (0x01)
     Identifier: 0x02
     Length: 23
   ⊟ Options: (19 bytes)
       MRU: 1524
       Async characters to map: 0x00000000
     ⊟ Authentication protocol: 5 bytes
         Authentication protocol: Cryptographic Handshake Auth. Protocol (0x
         Algorithm: CHAP with MD5 (0x05)
       Protocol field compression
       Address/control field compression
```

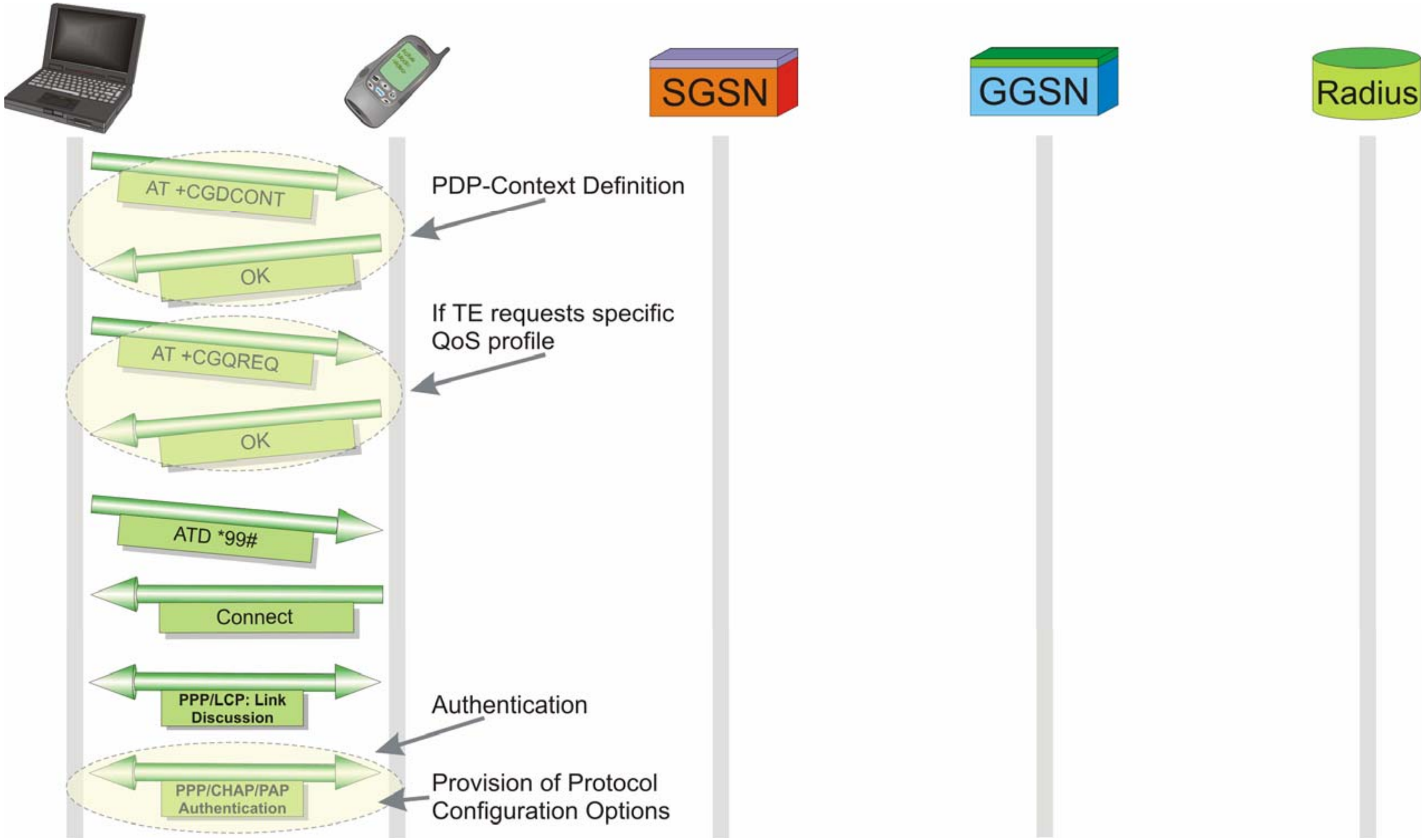
```
0000  20 52 45 43 56 00 20 52  45 43 56 00 c0 21 01 02    RECV. R ECV..!..
0010  00 17 01 04 05 f4 02 06  00 00 00 00 03 05 c2 23   ........ .......#
0020  05 07 02 08 02                                     .....
```

Max. length of data / options field

(2) Example for Dial-Up Network Access using the PPP

Intentionally left blank

(1) The Mobile Originating PDP-Context Activation Procedure



PDP-Context Definition

If TE requests specific QoS profile

Authentication

Provision of Protocol Configuration Options

(1) The Mobile Originating PDP-Context Activation Procedure

Initial Conditions

The mobile station is of any class A, B or C and has already established a GMM-context (⇔ GPRS-attached).

Applicability of this Procedure

⇒ This procedure is applicable for all mobile originating PDP-context activation with PDP-type IP, in particular with dynamic IP-address allocation

Description

⇒ The terminal equipment (e.g. laptop) first needs to define the PDP-context to be established through the transfer of the AT-command +CGDCONT = *[cid (⇔ PDP-context identifier) (m), PDP_type (m) (⇔ X.25, IP, OSPIH, PPP), APN (o), PDP-address (o), d_comp (o) (⇔ data compression V.42bis y/n), h_comp (o)(⇔ header compression RFC1144 y/n)]*. The mobile station shall confirm the reception of this AT-command.

⇒ Optionally, the terminal equipment may specify a specific QoS-profile to be requested for this PDP-context. In this case, the terminal equipment shall send another AT-command +CGQREQ = [cid (⇔ PDP-context identifier) (m), precedence (o), delay (o), reliability (o), peak (o), mean (o)] to the mobile station. The mobile station shall confirm the reception of this AT-command.

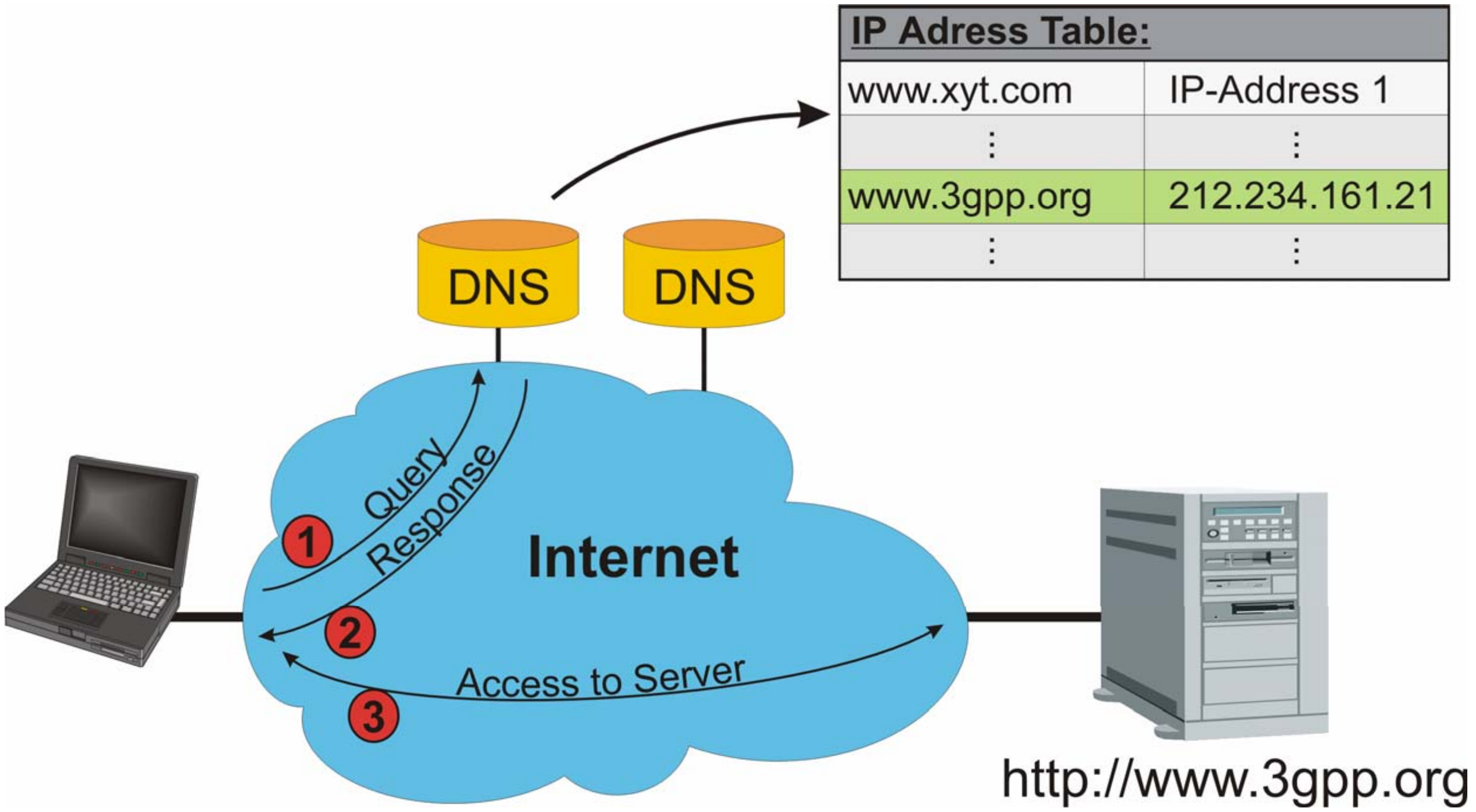
⇒ To initiate the PDP-context activation procedure, the terminal equipment will send another AT-command: ATD [*(GPRS Service Code = 99)(*called address)(*Layer 2 protocol)(*PDP-context identifier)#] to the mobile station. The mobile station shall confirm the reception of this AT-command and start the PDP-context activation procedure.

Note: This dial command usually defaults to ATD *99# because the parameters “called party address”, “Layer 2 protocol” and “PDP-context identifier” are most likely omitted. If one of these parameters shall be included then the respective number of “*” shall be added before this parameter. Example: If the PDP-context identifier shall be included then the command will be ATD *99***1# for the PDP-context = 1.

⇒ After the dial command, the PPP (Point-to-Point Protocol) will establish a layer 2 connection between terminal equipment and mobile station. This is done through the PPP LCP (Link Control Protocol). In addition, terminal equipment and mobile station shall negotiate a PPP authentication protocol (either CHAP or PAP with preference on CHAP), if user authentication to the PDN and / or between the terminal equipment and the mobile station is required.

[2GTS 03.60 / 2GTS 07.07 / 2GTS 07.60 / 2GTS 09.61 / RFC 1661 / RFC 1334]

Access to Applications ⇒ The Domain Name System (DNS)



IP Adress Table:

www.xyt.com	IP-Address 1
⋮	⋮
www.3gpp.org	212.234.161.21
⋮	⋮

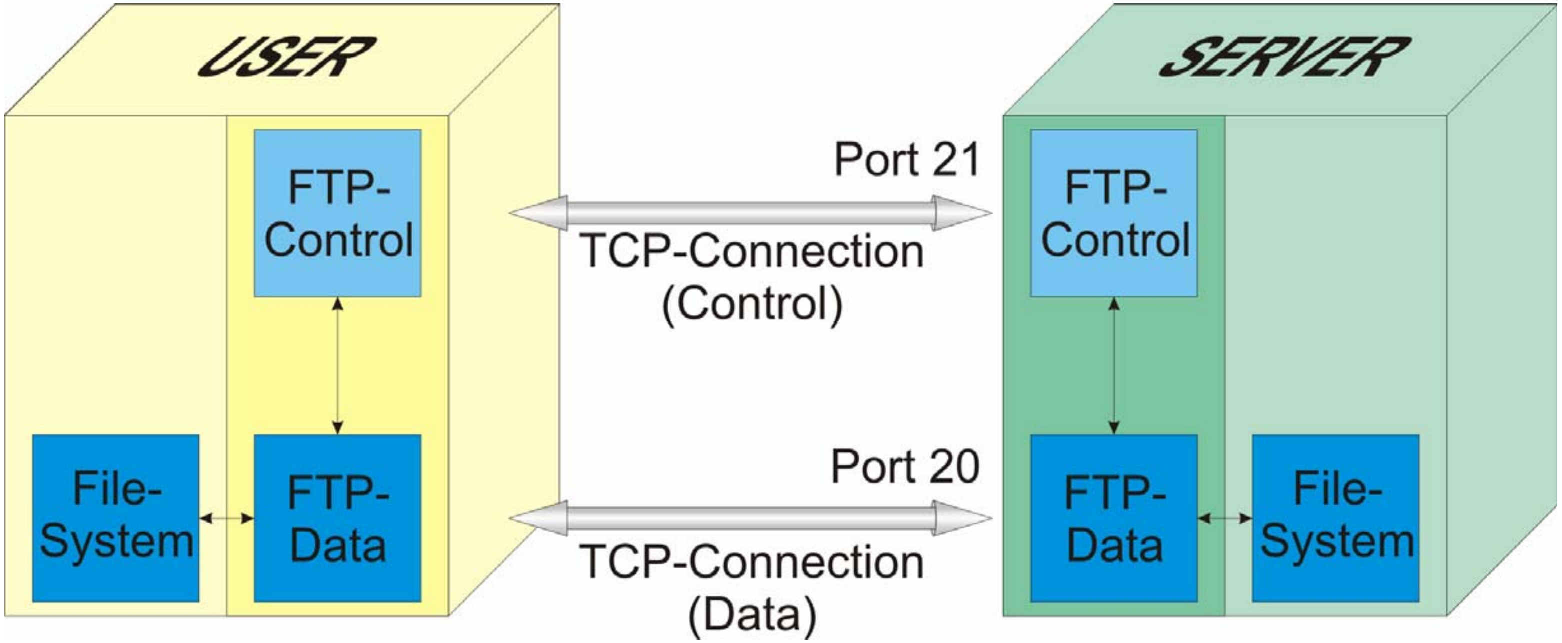
Access to Applications ⇒ The Domain Name System (DNS)

The Domain Name System is used to replace mnemonic Uniform Resource Identifiers (⇔ URI) against IP-addresses.

Note: For dial-up connections, the so called primary and secondary DNS-server's IP-address is conveyed to the user host upon network access. Another option which is frequently used for GPRS is to manually configure the mnemonic IP-addresses of the primary and secondary DNS-server upon installation.

[RFC 1101]

The File Transfer Protocol (FTP)



The File Transfer Protocol (FTP)

The File Transfer Protocol (FTP) is used between clients and servers to share file systems among them. This general description opens the possibility for a wide range of applications. For example, remote users may use the FTP to access “their” directory on the company server. FTP is more complex than HTTP, especially because two different ports are necessary for FTP.

- **The FTP-Control Port 21**
  FTP uses a control connection to exchange control information about the requested actions. Examples for such actions are the change of a directory, the request to download a specified file (⇔ RETR) or the request to upload a specific file to the server (⇔ STOR). TCP control connections are usually left open for a limited time (e.g. 300 s) before there closing is invoked by the server, if there is no activity.
- **The FTP-Data Port 20**
  The actual transfer of data occurs from or to port 20. Data transfer may relate to the transfer of files or to listing the content of a directory on the client terminal. Note that the data connection on port 20 is only opened if data shall be exchanged. The data connection will in turn be closed when the data transfer is completed.

[RFC 959]