ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Communication Systems

Security Overview

University of Freiburg
Computer Science
Computer Networks and Telematics
Prof. Christian Schindelhauer

CoNe
Freiburg

IIF
INSTITUT FÜR
INFORMATIK
FREIBURG

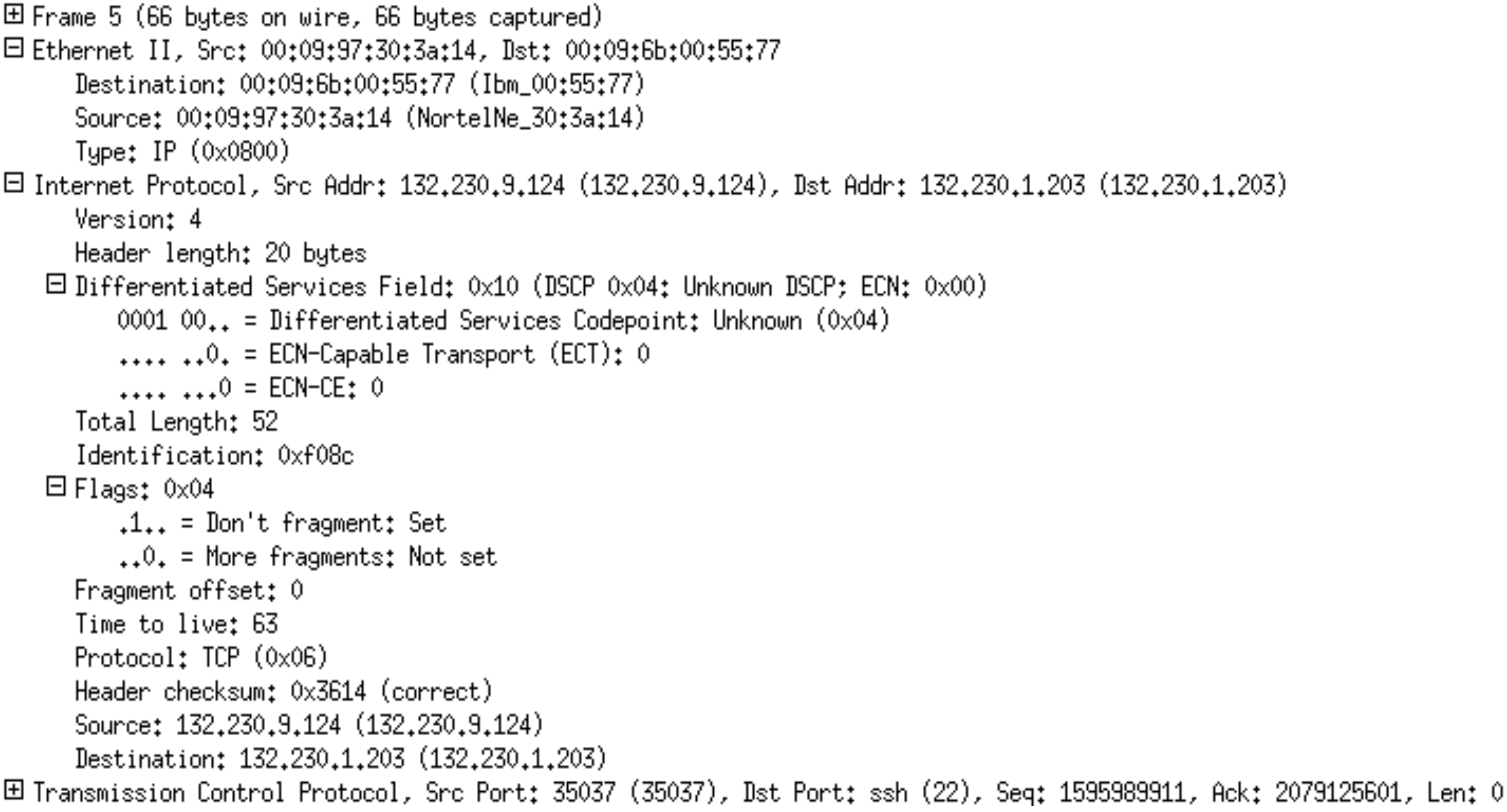
Organization

- I. Data and voice communication in IP networks
- **II. Security issues in networking**
- III. Digital telephony networks and voice over IP

Security in Computer Networks

- This lecture – broader introduction into problems of open networks, types and points of possible attacks
  - more than introduction is not possible
  - whole lectures may be held on that topic
- Security measures do not focus on a single network layer
- Different measures try to solve different problems that might occur
- There is no single measure, which will solve all security issues at once
- There will evolve new types of attacks and new types of counter measures

Packets Easy to Read Wireshark



```
⊞ Frame 5 (66 bytes on wire, 66 bytes captured)
⊟ Ethernet II, Src: 00:09:97:30:3a:14, Dst: 00:09:6b:00:55:77
      Destination: 00:09:6b:00:55:77 (Ibm_00:55:77)
      Source: 00:09:97:30:3a:14 (NortelNe_30:3a:14)
      Type: IP (0x0800)
⊟ Internet Protocol, Src Addr: 132.230.9.124 (132.230.9.124), Dst Addr: 132.230.1.203 (132.230.1.203)
      Version: 4
      Header length: 20 bytes
    ⊟ Differentiated Services Field: 0x10 (DSCP 0x04: Unknown DSCP; ECN: 0x00)
          0001 00.. = Differentiated Services Codepoint: Unknown (0x04)
          .... ..0. = ECN-Capable Transport (ECT): 0
          .... ...0 = ECN-CE: 0
      Total Length: 52
      Identification: 0xf08c
    ⊟ Flags: 0x04
          .1.. = Don't fragment: Set
          ..0. = More fragments: Not set
      Fragment offset: 0
      Time to live: 63
      Protocol: TCP (0x06)
      Header checksum: 0x3614 (correct)
      Source: 132.230.9.124 (132.230.9.124)
      Destination: 132.230.1.203 (132.230.1.203)
⊞ Transmission Control Protocol, Src Port: 35037 (35037), Dst Port: ssh (22), Seq: 1595989911, Ack: 2079125601, Len: 0
```

Network Insecurity

- IP packets are easily readable (if provided with the proper tools)
- e.g. **wireshark** can provide the user/network administrator
  - with a graphical user interface for interpreting packets
  - can grab all packets visible to a machine (promiscous mode in LANs like ethernets)
  - can sort out TCP streams (check which packets are part of a certain communication)
  - can interpret most of protocol packets
- You should be familiar with this tool (and others like tcpdump) from the last lectures

Network Insecurity

- Why packets are as easily readable?
- All communication has to follow standards – otherwise no communication would be possible (think of people talk in different languages with each other)
- Even not open protocols, like certain implementations of windows network service are interpretable – such the samba service is developed through trial-and-error and reverse engineering
- Thus: **no security by obscurity**!!
- In the beginning of "The Internet”
    - very few participants in networks
    - very few computers connected to each other
    - very few people with deep understanding of networking
    - not many network analyzation tools available (for free)

Network Insecurity

- Restricted computing power of connected machines (in the beginning of IP networking)
  - protocols should be very simple and should not impose high loads on the machine
  - encryption technologies were not common knowledge / restricted for export ("strategic technology”)
- And: simplicity of TCP/IP protocol suite helped the rapid growth of the Internet and fast adaptation for the different operating systems
- By now: the Internet is one of base technologies for information exchange and communication
- Wide range of businesses directly depend on this network (online shops, auctions, b2b, multiplayer games, advertisements, porn sites, web services, ... :-))

Network Insecurity

- Inner and intra firm communication moves from the classic communication media telephone and fax over to mail and similar technologies
  - Sending and reception of a wide range of digital objects
  - E.g. with the “melissa” virus (spectacular some years ago) you could observe employees entering their offices at eight and leaving them at half past nine (no mail and online communication was available – most MS operated networks)

- Production and development heavily depend on networks – most information between firms is directly interchanged between databases over the net
- At the moment: move of telecommunications into IP networks to avoid duplicated infrastructure and cut communication costs

Network Insecurity

- Networks could be attacked on all layers
- **Layer 1 and 2**
  - E.g. ARP spoofing in broadcast networks for man-in-the-middle attack, redirection of default gateway traffic over the attackers host (earlier exercise)
  - Rather simple within WLANs (unguided media with no distinct boundaries):
    - spamming with corrupt packets or simply noise (microwave oven) – frequency band is rendered unusable
    - breaking the weak WEP algorithm
  - “dialer” programs (mostly history by now) – redirection of Internet traffic over costly dial-in lines (attack is of course induced via web applications, trojan horses, ...)

Network Insecurity

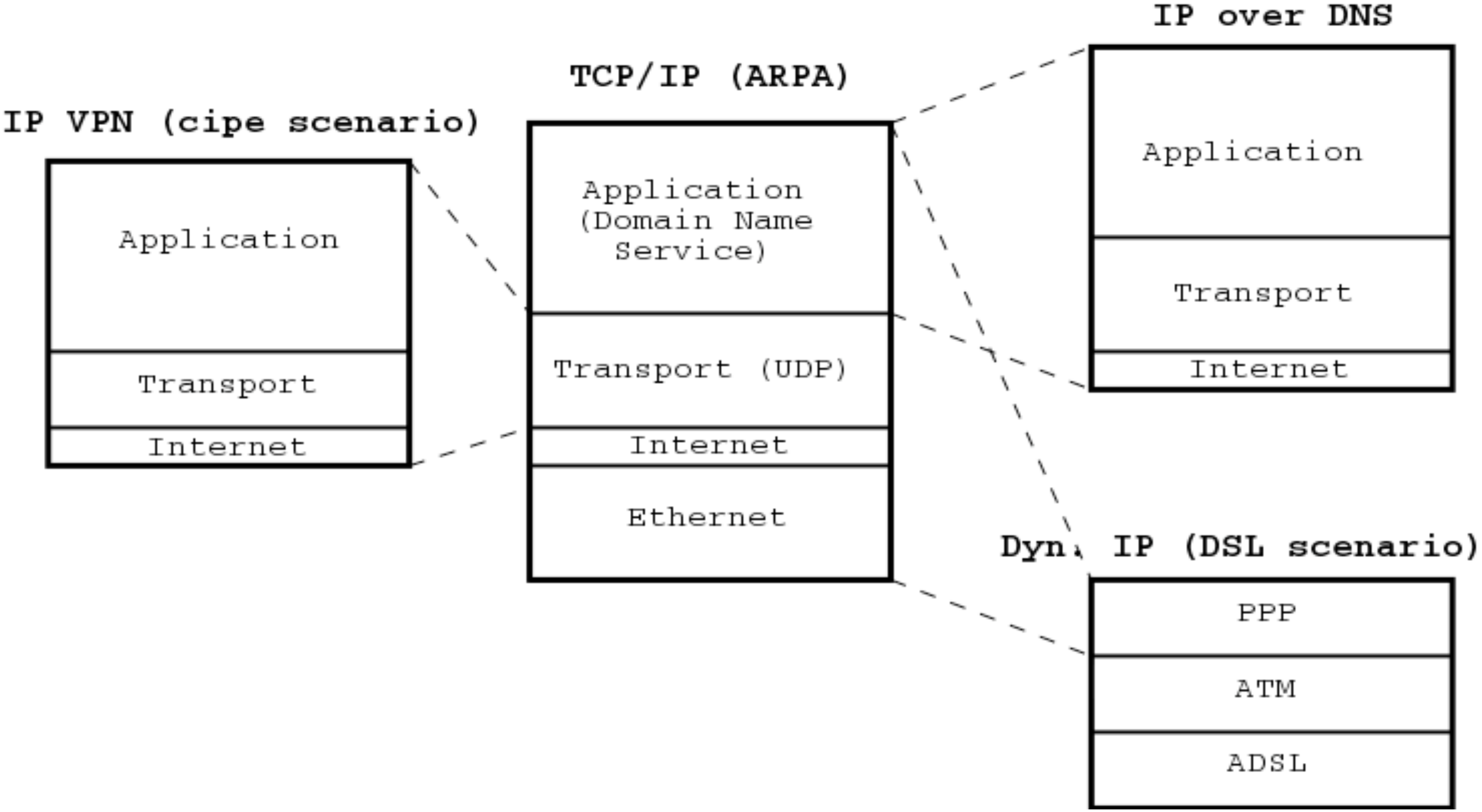
- **Layer 3**
  - IP spoofing – forging of IP addresses for good or malicious reasons for motivation of IPsec
  - Attacking router protocols, e.g. RIP (II) for redirecting traffic in LANs, ICMP redirects, ...
- **Layer 4**
  - very simple to send unsolicited UDP packets – connectionless service (such spoof protocols like SNMP, DHCP, DNS, ...)
  - take over open TCP connections – grab an open telnet, mail, http session to use an authenticated session to a remote host
  - TCP syn attacs (open as many TCP connections as possible from different hosts and leave them in open state without further communication – type of distributed denial of service DdoS)
  - dynamic routing protocols (drop in replacement for TCP or UDP) have their weaknesses too ...

Network Insecurity and Tunnels

- Special issue in networking and security are network tunnels
  - IPv6-in-IPv4 tunnels (earlier lecture)
  - Virtual Private Networks (VPN) can be set up with the help of tunneling (discussed in later lecture)
  - And special tunnels for servicing: Database producers providing tools to open a service tunnel over HTTP (because all other traffic is blocked)
  - Tunnels for cost optimizations (save real money) ... we will talk of now
- Nowadays, tunneling techniques are popular among users for defeating firewall restrictions to freely access the Internet
- Could be defined like: Tunneling, in the most general sense, means not to play by the rules of the layering concept, thus allowing to transfer data without restrictions between the layers

Network Insecurity and Tunnels

- Tunnels from the view of the protocol stack



Network Insecurity and Tunnels

- Legal issues
  - A tunnel in general can not be considered as illegal. However a tunnel may allow you to commit illegal activities (transfer data without permission, surf the web using a “free” account)
  - For this reason many administrators deal with this problem by introducing rules, that forbid the use of tunnels that fool their security systems. People who do not play by the rules will suffer certain penalties (e.g. get fired)
- All following explanations are based on the workings of the NSTX project team and their freely available sourcecode

IP-over-HTML / IP-over-WAP

- Motivation for HTTP tunnels
  - Firewall secured network with only HTTP connects allowed (via proxy, transparent proxying or alike)
  - Typical scenario found in companies or lecture room environments
  - Problem: Lecturer, guest or sales/service official would like to demonstrate some services which require open network access or simply work remote other than HTTP (getting mail via IMAP etc.)

- Motivation for WAP tunnel (obsolete here, but still of use in some other countries like Greece – depending on your mobile providers business model)
  - Originally there was some ad of a mobile provider (O2) for “flat Internet access for 5€” (really cool if true, but of course not – just WAP) – thus how to “enhance” the service for general IP

IP-over-HTML / IP-over-WAP

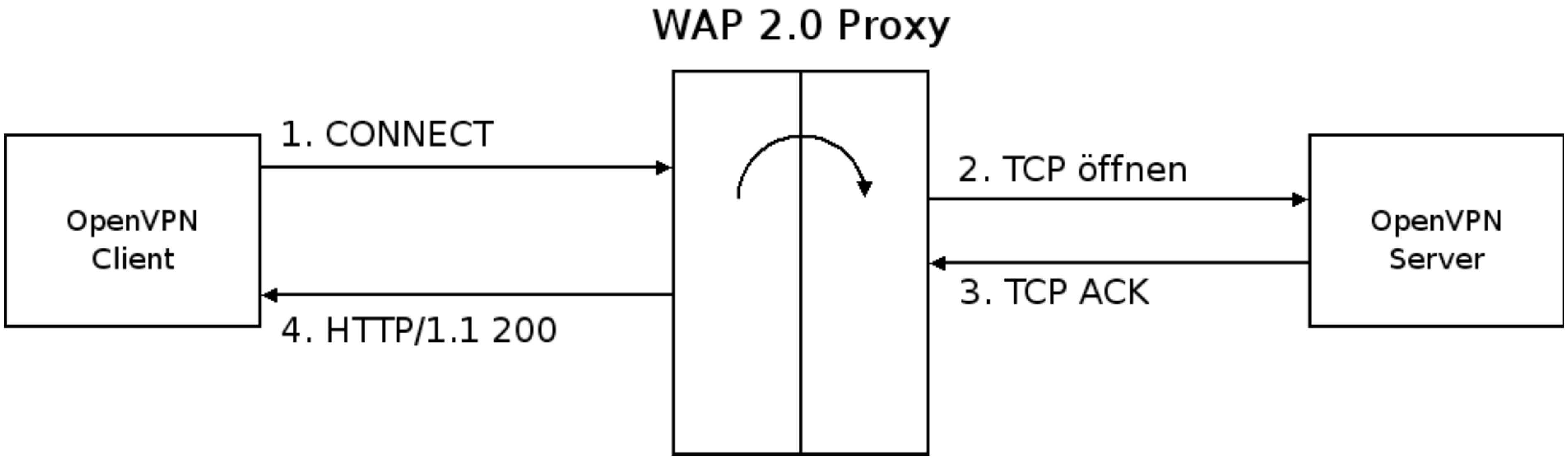
- No big difference between WAP2.0 and standard HTTP, you will find rather similar setup in normal private/secured networks
  - Private IP address range
  - Special gateway/proxy which restricts communication to just some ports
- Thus further on WAP tunnels, but mostly same applies for other setups
  - Special APN within the O2 network
  - GPRS, private IP network
  - Only ports 8080 (WAP2.0) and 9201 (WAP1.X) were

IP-over-WAP / OpenVPN

- What should the tunnel service offer?
  - IP packets need to be hooked on HTTP proxy/WAP conforming request packets, universal network interface for the own applications
  - Data compression, encryption, scalability
  - Platform independent
- Offered by Open Source OpenVPN project
  - Started as universal VPN in 1999, to be independent of IPsec (too complex for many setups, more flexible for tunneling, NAT etc.)
  - Uses the OpenSSL library
  - Offers normal network interface (based on TUN/TAP interface) and HTTP proxy support

Tunneling with OpenVPN

- For tunneling: You need some accessible endpoint in your IP network
- Setup of OpenVPN via HTTP proxies using the CONNECT-Option
  - CONNECT could be seen as a TCP redirect, required for SSL-connects (caching useless)
- Most important phase: initialization of the connection



Tunneling with OpenVPN

- OpenVPN with CONNECT tunneling needs a HTTP header like:
  - User-Agent:
    - Mozilla/1.22 (compatible; MSIE 5.01; PalmOS 3.0) EudoraWeb 2.1
    - Profile: http://wap.sonyericsson.com/UAprof/P800R102.xml
- Some special adaptions: Changes in tunnel restarting timeout
  - Required because proxy cuts the connection after the transfer of ~1,5MByte or a certain amount of time

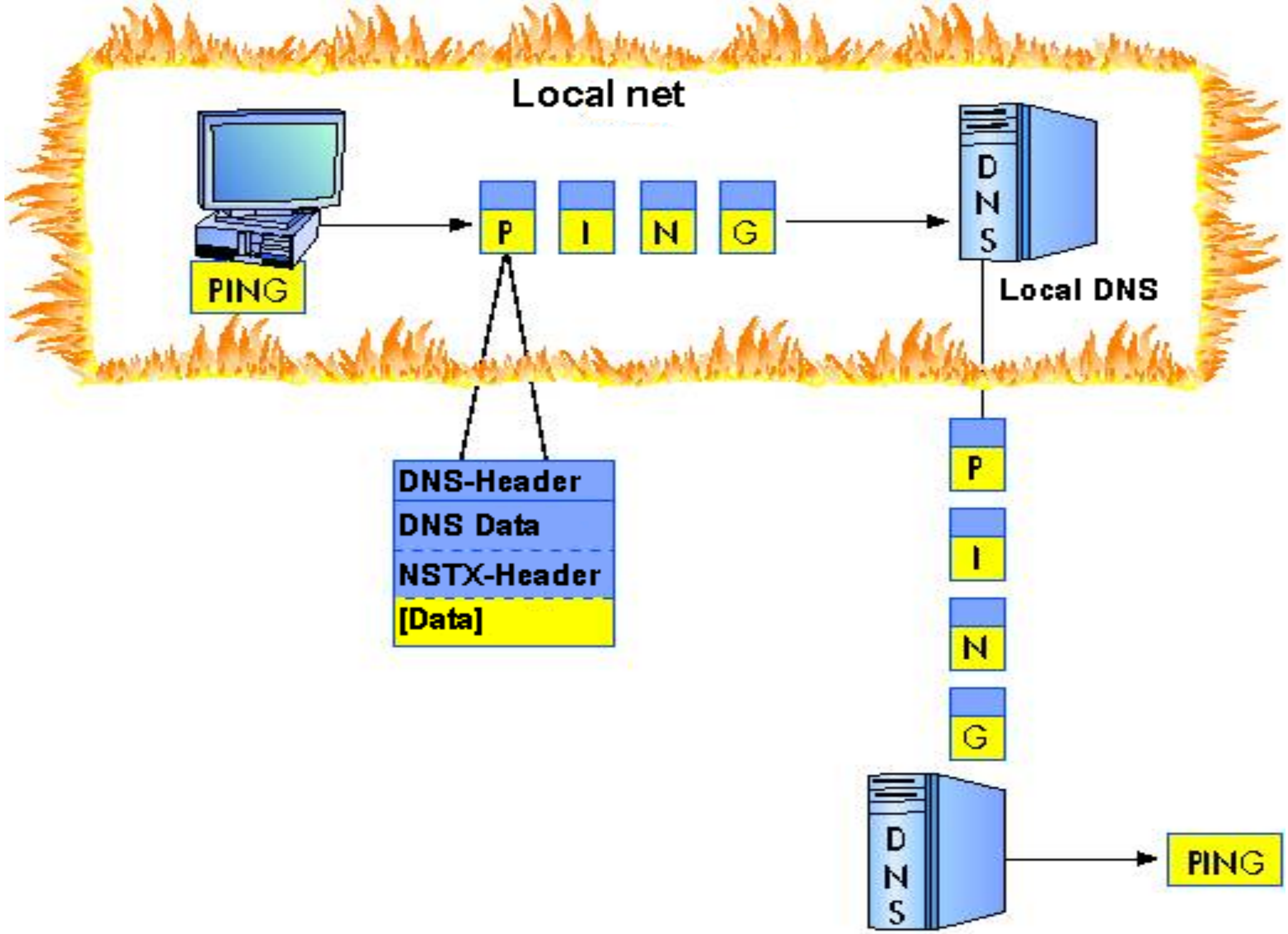
Other tunnels

- HTTP is not the only protocol typically open in restricted networks
- Original idea:
  - Microsoft offered some “service” for listing of available IP providers (which paid to be listed) in Windows2000 via toll-free number
  - Simple dial-in service with restricted network access
  - DNS resolutions worked for some reason
  - So clever people thought of using this service for more general Internet access (just for free)

DNS tunnel using NSTX

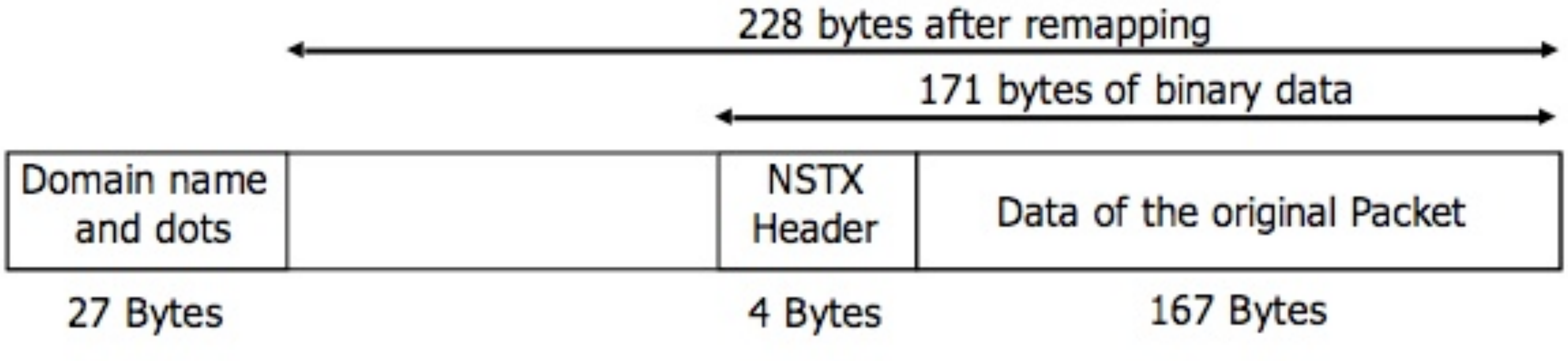
- By now typical scenario in access networks like WLANs (found in hotels, airports, our university, ...)
  - Setup: IP and DNS configuration via DHCP, later on setup of a secured/authenticated tunnel or disabling of traffic (after authentication in both cases)
  - In the VPN setups you get a new IP address but typically no new DNS configuration thus DNS has to handle requests in both states (restricted and open)
  - Try it: Ping some external service (like www.google.de) - you will see the name resolved to some IP but nothing more happens (no routing)
  - DNS server acts like a proxy between restricted local net and outside world

DNS tunnel using NSTX – main idea



NSTX Encoding

- Depending on the length of your NSTX domain (select a short one!) and minus NSTX header you get a payload of around 160Byte in every DNS request package
- IP packets are naturally bigger (think of fragmentation, some earlier lecture) – thus NSTX uses own fragment handling
- Example for „.gjas.de”



DNS tunnel using NSTX

- Other problem: data encoding – 8bit to some 6bit (think of the allowed characters in domains!) thus theoretical size of 255 Byte payload (max. length of a name) is smaller
- Discussed how data is passed from NSTX client to the NSTX server through DNS queries. Now what about the other direction
  - NSTX server uses DNS replies to send data to the client
  - TXT Resource Records can be used in a DNS reply
  - (TXT RR can contain any data => remapping is not needed => fragments can be 223 bytes long)
  - Problem: NSTX server has to wait for a DNS query to send a DNS reply (!)
  - polling techniques are needed in the NSTX client (send empty kind of keep-alive packets)

Conclusion

- Some issues of DNS tunnel
  - Overhead in DNS query : mainly influenced by fragmentation (10 fragments means ten times the size of the needed headers (Ethernet, IP, UDP & DNS) and the remapping
  - Overhead in DNS reply : is increased by the fact, that original hostname of the last query also has to be included
  - Delay: The number of hops the DNS query takes seems to be the most important factor in the performance of the tunnel
- Detection
  - Filter TXT-RR in DNS-replies, DNS server traffic analysis
  - Restrict the DNS bandwidth
  - Point clients at an internal caching name server, that is separated from the outside

ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Communication Systems

University of Freiburg
Computer Science
Computer Networks and Telematics
Prof. Christian Schindelhauer

CoNe Freiburg

IIF INSTITUT FÜR INFORMATIK FREIBURG