Datanet
Obligatorisk opgave 3: IP og ICMP

René Hardi Hansen
Michael Falcke Nilou
Anders Bjerg Pedersen
Hold 1

26. september 2007

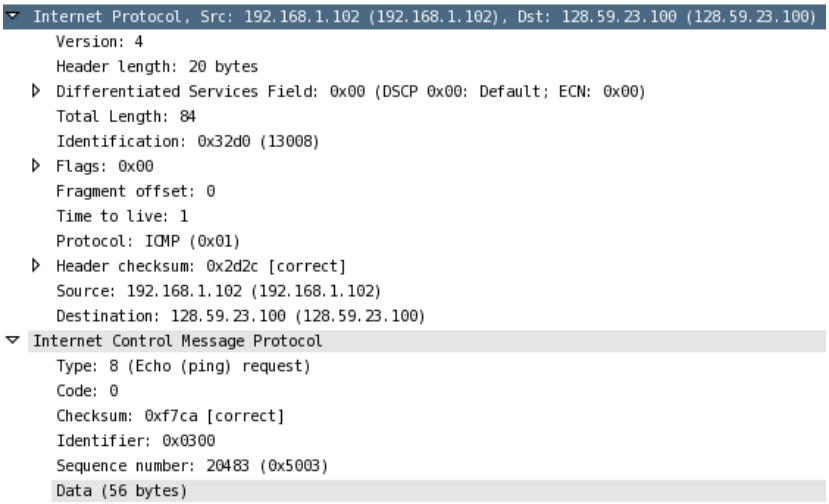
Indledning

Denne opgave går ud på at analysere IP-protokollen ved at betragte IP-datagrammer transmitteret under udførelse af programmet Traceroute. Endvidere ser vi på programmet ping, på IP-fragmentering og på ICMP-protokollen.

Wireshark Lab: IP

2. A look at the captured trace

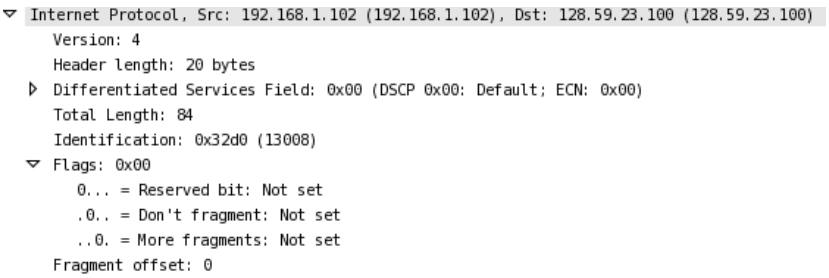
Q1: Vi anvender tracet, der er udgivet sammen med opgaven, og finder klientens IP-adresse i TCP-pakke nr. 8, der er den første pakke, der sendes ud fra traceroute-programmet (se screenshot nedenfor):
`Src: 192.168.1.102 (192.168.1.102).`



Q2: Værdien i protocol-feltet er: (se samme screenshot):
`Protocol: ICMP (0x01).`

Q3: Der er 20 bytes i IP-headeren. Der er 64 bytes data. Disse tal kan findes ved at trække længden af headeren fra den totale længde af IP-datagrammet, eller ved at tælle antallet af bytes i ICMP-protokollen i den pågældende pakke.

Q4: IP-datagrammet er ikke fragmenteret. Offset er 0, og flaget for flere fragmenter er ikke sat (se nedenstående screenshot).



Q5: Ved at traversere pakkerne i Wireshark ser vi, at følgende felter i IP datagrammet altid ændrer værdi: `Time to live: TTL`, `Header checksum` samt `Identification`.

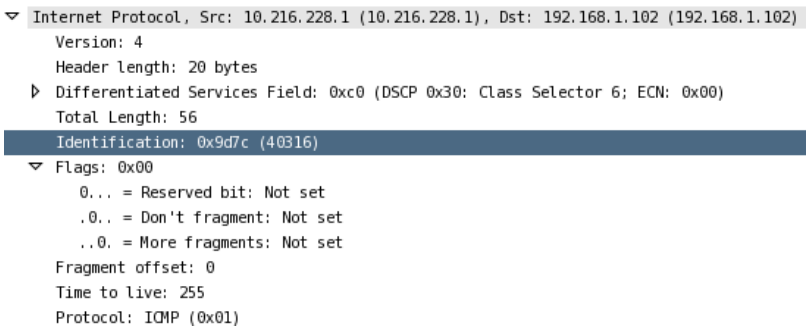
Q6: I besvarelsen af dette spørgsmål, begrænser vi os til at betragte IP headeren. Ved at traversere pakkerne i Wireshark ser vi, at følgende felter i IP datagrammet er konstante: `Version: 4`, `Header length: 20`, `Differentiated services field`, `Protocol: ICMP (0x01)`.

Følgende felter i IP-datagrammet skal være konstante (så længe vi anvender IPv4 og laver traceroute):
`Version: 4` (dog kan man med f.eks. tunnelling undgå, at denne skal holdes konstant), `Protocol: ICMP` (alt andet ville ikke være en del af traceroute-programmet).

Følgende felter i IP datagrammet skal variere:
`Time to live: TTL` og `Header checksum`. TTL varierer, fordi specifikationen af traceroute (RFC 2151) anvender TTL-værdien til at finde navnet på den n'te router. Traceroute starter med at udsende 3 pakker med TTL=1. Disse pakker medfører, at den nærmeste router (1 hop) svarer tilbage med en type 11 code 0 warning message: TTL expired, samt navn og IP-adresse på routeren. Dernæst øger traceroute TTL-værdien med 1, indtil den søgte host svarer tilbage med en ICMP type 3 code 3 (destination port unreachable). Denne meddelelse skyldes, at traceroute forsøger at aflevere et UDP-segment til en ugyldig port hos den søgte host.

Q7: Vi ser, at pakkens identifikationsnummer tælles op med 1, for hver pakke der afsendes, indtil pakkerne fragmenteres, da bevares det samme identifikationsnummer på samtlige fragmenter. `Identification number`.

Q8: Identifikationsnummeret i IP-delen af svaret fra routeren er:
`Identification: 0x9d7c (40316)`. TTL er 255.



Q9: For ICMP-delen af svaret, vil returneringen altid rumme en TTL på 1, da routeren ikke trækker yderligere fra i TTL, da den allerede har nået minimum, men

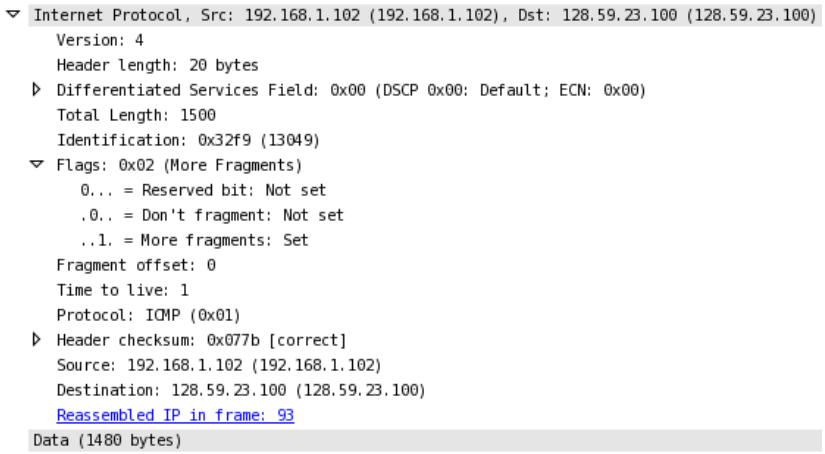
blot returnerer en ICMP type 11 code 0 warning message: TTL expired. Identification ændrer sig hver gang, da der er tale om nye pakker, som (hvis de skulle fragmenteres) skulle være mulige at skelne fra hinanden.

3. Fragmentation

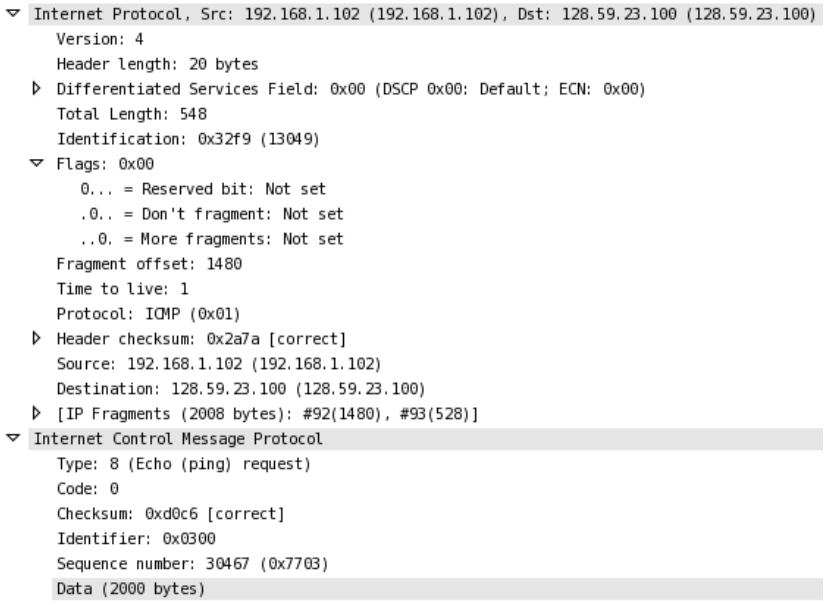
Q10: Ja, IP-datagrammet består af 2 fragmenter. Pakke nr 92 og 93 i tracet.

Q11: Se nedenstående screenshot. Ja, IP-datagrammet består af 2 fragmenter (pakke nr. 92 og 93 i tracet). I headeren til datagrammet er flaget for flere fragmenter sat. Når flaget for flere fragmenter er sat samtidig med at fragment offset = 0 så er der tale om det første fragment.

Vi tager længden af IP-segmentet minus headeren for begge pakker: $(1500-20)+(548-20) = 2008 bytes$. Dette er det samme tal, som Wireshark i den anden pakke præsenterer i bunden, hvor programmet tilbyder at samle IPv4-pakken for os.



Q12: Der er henvist til et fragment offset > 0. Heraf ses, at der er tale om et fragment. Der er ikke flere fragmenter, da flaget for flere fragmenter ikke er sat.



Q13: Følgende felter forandrer indhold fra 1. til 2. fragment: `Total Length`, `Flags:..0. = More fragments`, `Fragment offset`, og `Header checksum`. De interessante felter er her: `Flags` og `Fragment offset`. `Total length` behøver ikke nødvendigvis at ændre sig, og checksummen vil altid forandres ved forandring af pakken.

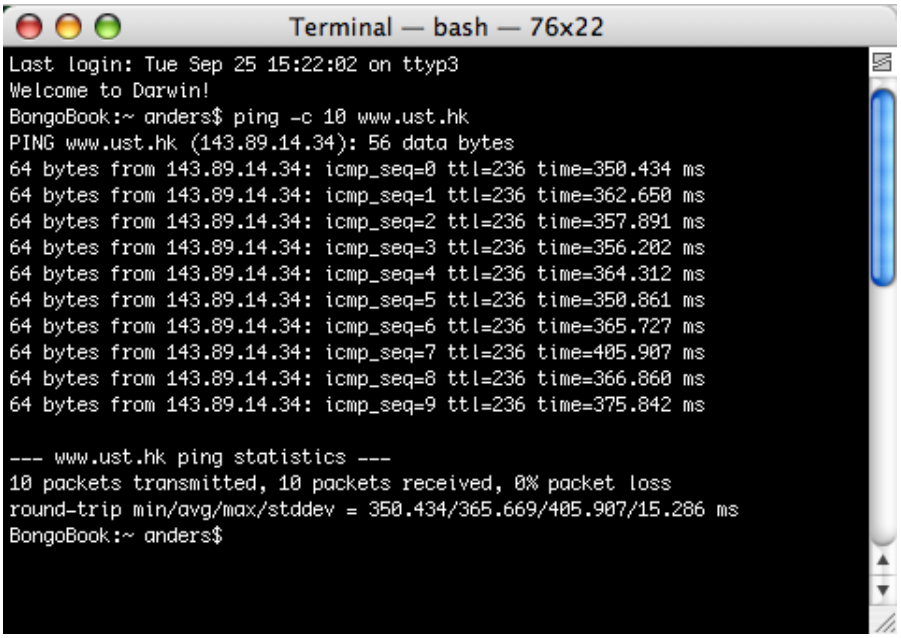
Q14: Der bliver dannet 3 fragmenter af det oprindelige datagram: Frame 216, 217 og 218.

Q15: Der henvises til besvarelsen af Q13. Endvidere ses, at `Total Length` sættes til maximum (1500 bytes), så længe der kan dannes fragmenter af denne længde. I den sidste pakke tilpasses `Total Length` til kun at indeholde den resterende datamængde.

Wireshark Lab: ICMP

1. ICMP and Ping

Screenshot af ping-programmet kørt i en UNIX-terminal:



BEMÆRK: Opgaverne er besvaret ud fra Wireshark-tracet og *ikke* ud fra ovenstående kørsel af ping-programmet i UNIX.

Q1: Source og Destination kan aflæses af IP-datagrammet (f.eks. fra pakke 3):
`Internet Protocol, Src: 192.168.1.101 (192.168.1.101)`,
`Dst: 143.89.14.34 (143.89.14.34)`.

Q2: ICMP er en protokol, der kun benyttes i netværkslaget (men rent principielt ligger lige over IP). Der er ikke tale om kommunikation mellem processer, og derfor er der ikke brug for portnumre på hverken source eller destination.

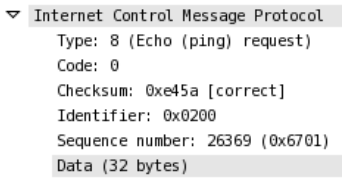
Q3:

ICMP Type: 8 (Echo (ping) request)
ICMP Code: 0

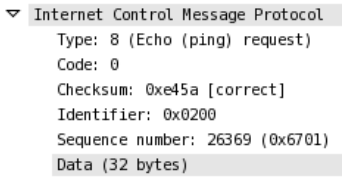
Ud over disse felter findes felterne: `Checksum`, `Identifier`, `Sequence Number` og `Data`.

ICMP Checksum:	2 bytes
ICMP Identifier:	2 bytes
ICMP Sequence Number:	2 bytes

Hvert hexadecimaltal repræsenteres af 4 bits. 4 hex-tal efter hvert felt må så betyde 2 bytes per felt.

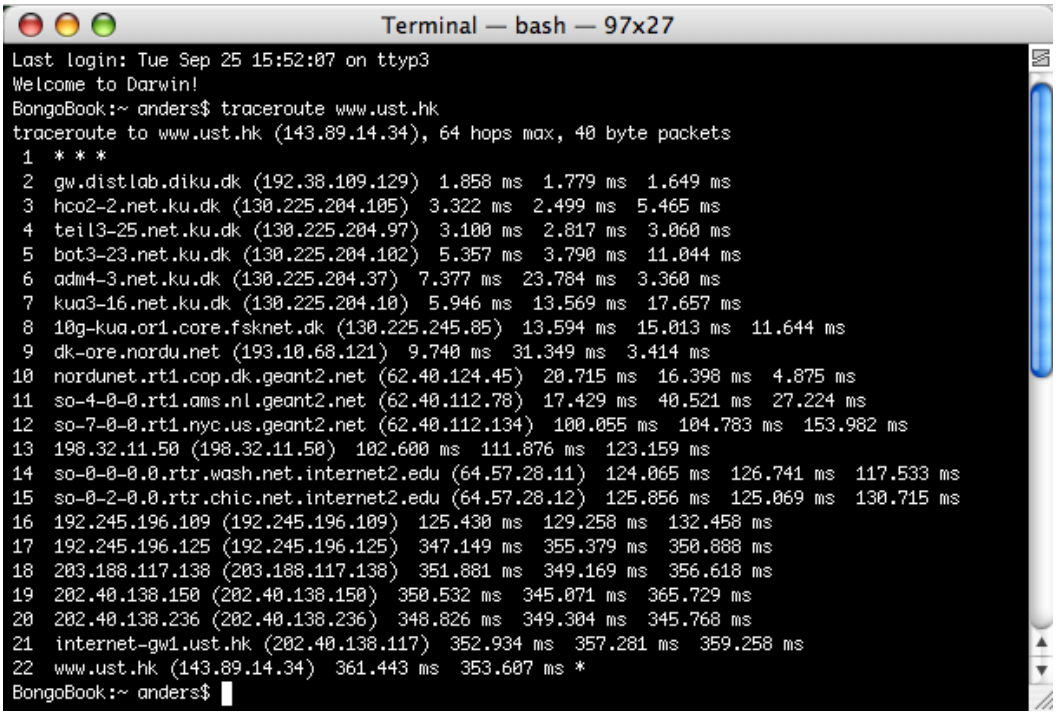


Q4: Reply-pakkerne indeholder nøjagtig det samme, bortset fra, at ICMP code er 0 i stedet for 8. Se derfor Q3 for feltstørrelser og andet.



2. ICMP and Traceroute

Screenshot af traceroute-programmet kørt i en UNIX-terminal mod www.ust.hk:



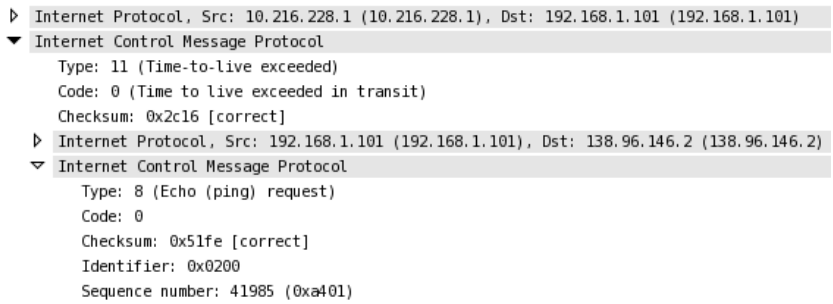
BEMÆRK: Opgaverne er primært besvaret ud fra Wireshark-tracet og *ikke* ud fra ovenstående kørsel af traceroute-programmet i UNIX.

Q5: Source og Destination får vi igen fra IP-protokollen:
`Internet Protocol, Src: 192.168.1.101 (192.168.1.101),`
`Dst: 143.89.14.34 (143.89.14.34).`

Q6: Nej, protokolnummeret er nu 11 for UDP, men der returneres stadig et ICMP-segment. På tracet fra ovenstående screenshot kommer der skiftevis UDP- og ICMP-pakker.

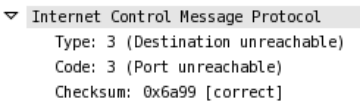
Q7: Payload-størrelsen på vores ping-requests er 32 bytes, hvorimod den i vores traceroute-requests er 64 bytes. Desuden er indholdet af payload også forskelligt (jævnfør at størrelsen er forskellig...).

Q8: ICMP-fejlpakken indeholder det IP-datagram (inkl. ICMP-segmentet), som fejlen udsprang fra. Dermed kan modtageren se, hvilken pakke der gik galt. Desuden har fejlpakken `Type: 11 (Time-to-live exceeded)` og `Code: 0 (Time to live exceeded in transit)`.



Q9: Traceroute sender altid 3 ping-requests til hver router, derfor de sidste 3 pakker. Disse 3 pakker er nået frem til den host, der skulle traceroutes til, derfor svarer destinationen med et almindeligt ping-reply på de 3 pakker. Dette betyder, at de sidste 3 pakker (når de når frem til destinationen) stadig har et tilstrækkelig højt TTL-nummer til, at serveren blot kan svare på dem uden at gå i TTL Expires-mode.

Dette gælder dog kun, fordi tracet er kørt på en Windows-maskine. På vores eget trace på UNIX giver de sidste pakker en `Type: 3 (Destination unreachable)`, `Code: 3 (Port unreachable)` retur. Dette skyldes, at UNIX sender UDP-datagrammer til en fiktiv port på serveren (se screenshot nedenfor).



Q10: Der ses et spring i responstiden fra hop nummer 9 til nummer 10 fra gennemsnitligt 24ms til 97ms. Det er her, forbindelsen hopper over atlanten fra New York (att-gw.nyc.opentransit.net) til Frankrig (P4-0.PASCR1.Pastourelle.opentransit.net).

Konklusion

Vi har i denne opgave set nærmere på, hvordan IP-protokollen bruges til at sende data over Internettet, både som hele og fragmenterede segmenter. Vi har også undersøgt, hvordan ICMP-protokollen i samspil med IP-protokollen kan bruges til at fejlfinde f.eks. tabte pakker eller dårlige modtageradresser og -porte. Til sidst har vi set, hvordan forskellene mellem forskellige platformes brug af Traceroute-programmet mht. UDP- eller ICMP-pakker.