Comunicações por Computador
Trabalho Prático 1: Protocolos da Camada de Transporte

Ana Rita Peixoto, Leonardo Marreiros, and Luís Pinto

University of Minho, Department of Informatics, 4710-057 Braga, Portugal
e-mail: {a89612,a89537,a89506}@alunos.uminho.pt

Questão 1. Inclua no relatório uma tabela em que identifique, para cada comando executado, qual o protocolo de aplicação, o protocolo de transporte, porta de atendimento e overhead de transporte, como ilustrado no exemplo seguinte:

Comando Usado (Aplicação)	Protocolo de Aplicação	Protocolo de transporte	Porta de atendimento	Overhead de transporte em bytes
Ping	-	-	-	-
traceroute	TRACEROUTE	UDP	33446	8
telnet	TELNET	TCP	23	20
ftp	FTP	TCP	21	20
Tftp	TFTP	UDP	69	8
browser/http	HTTP	TCP	80	20
Nslookup	DNS	UDP	53	8
ssh	SSH	TCP	22	20

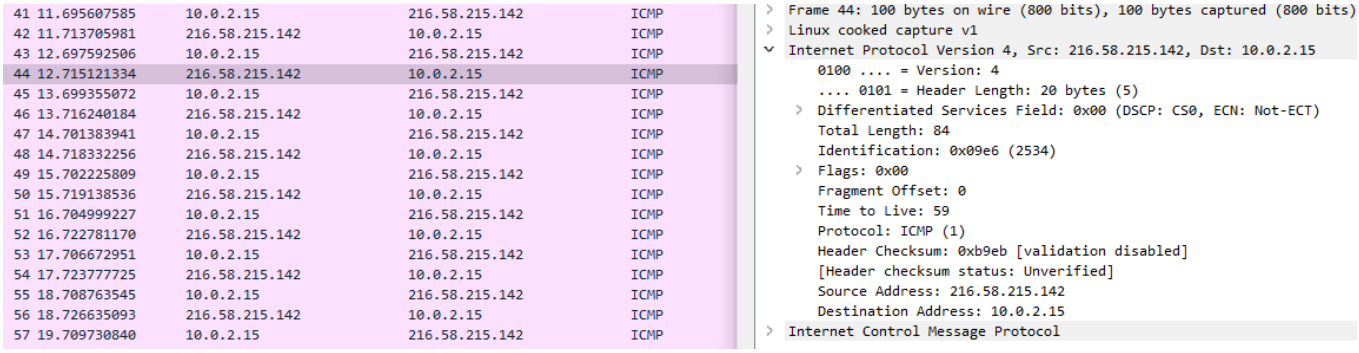


Fig. 1: Ping

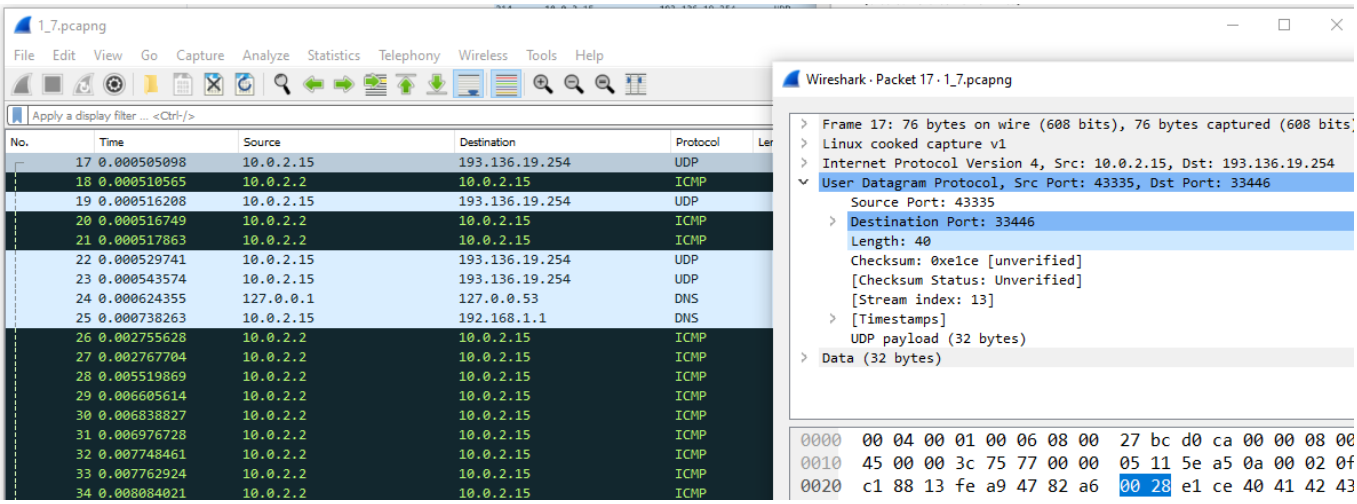


Fig. 2: TRACEROUTE

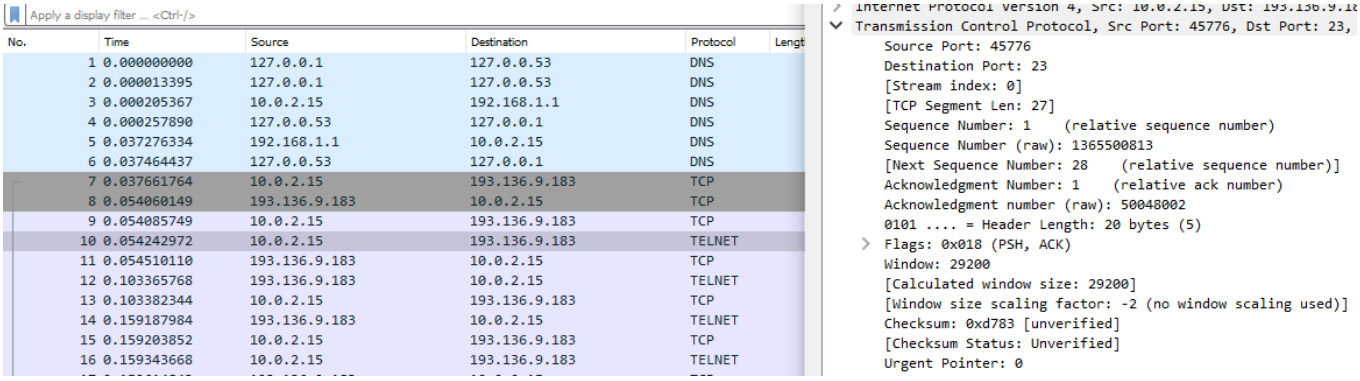


Fig. 3: TELNET

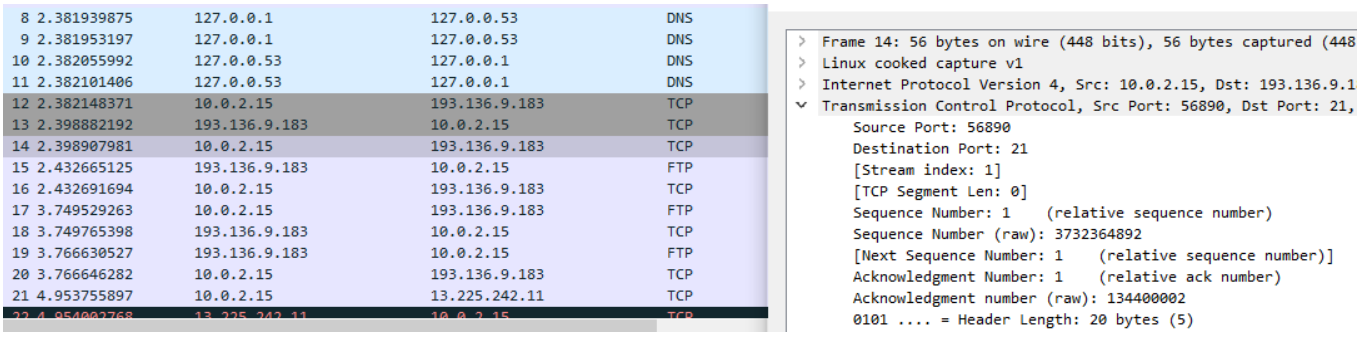


Fig. 4: FTP

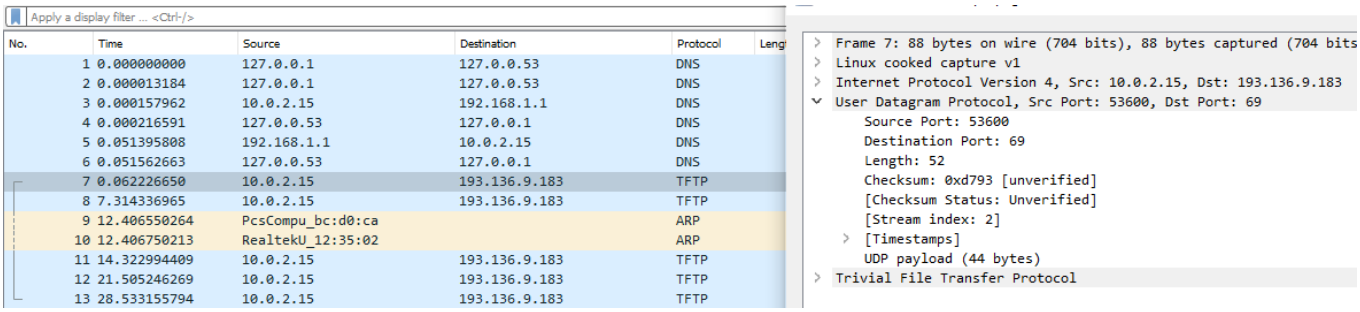


Fig. 5: TFTP

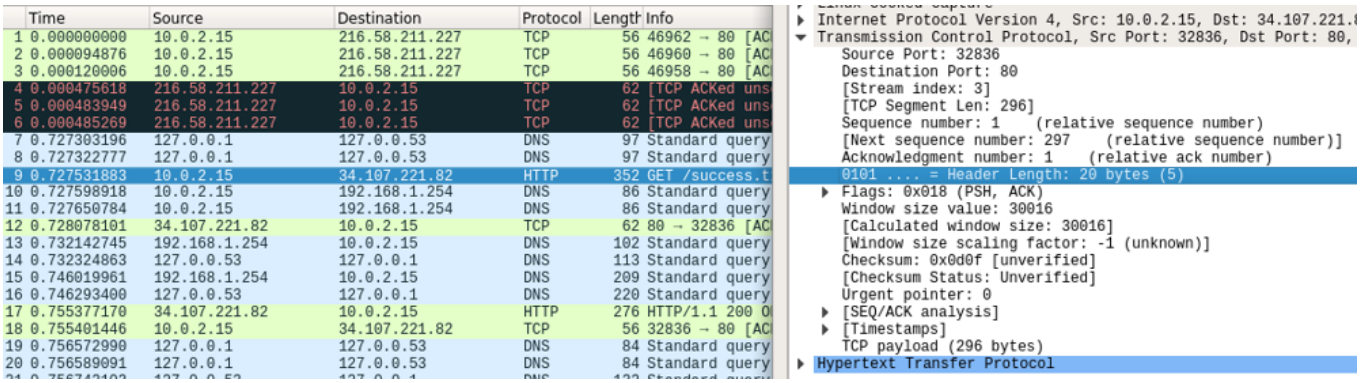


Fig. 6: HTTP

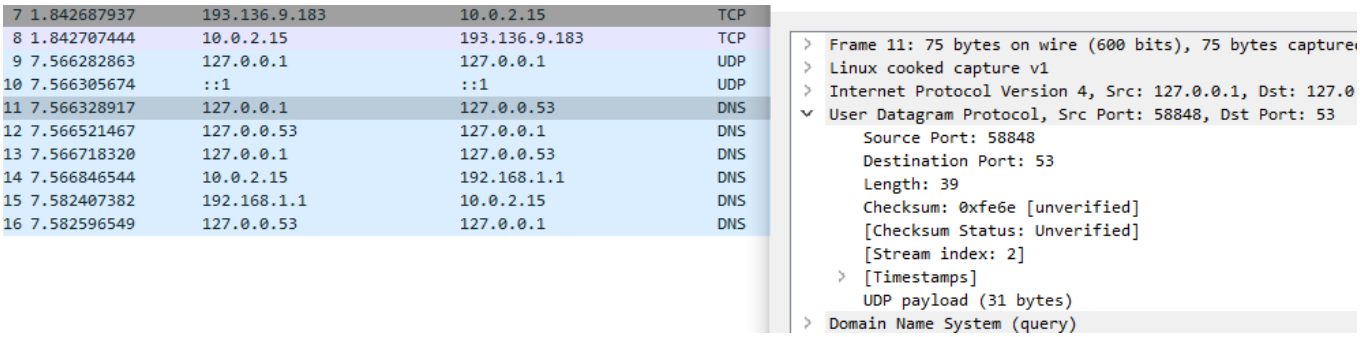


Fig. 7: NSLOOKUP

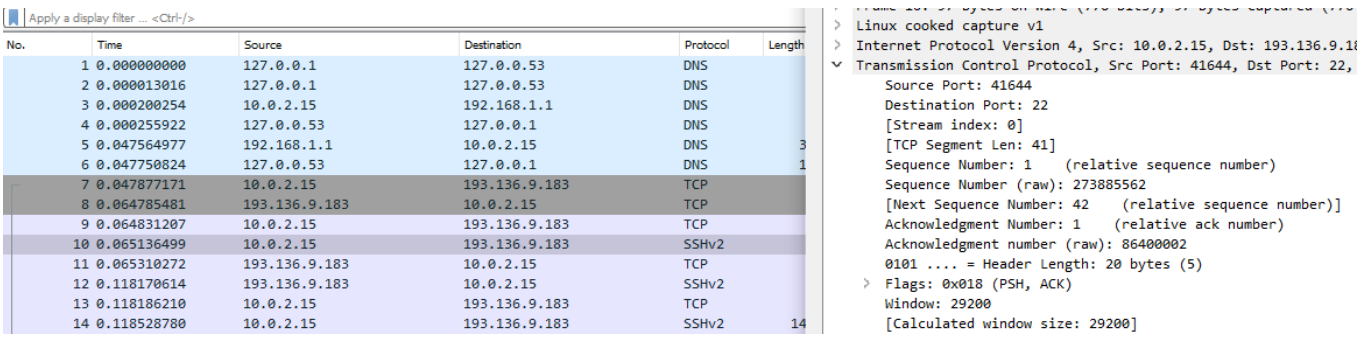


Fig. 8: SSH

Questão 2. Uma representação num diagrama temporal das transferências da file1 por FTP e TFTP respetivamente. Se for caso disso, identifique as fases de estabelecimento de conexão, transferência de dados e fim de conexão. Identifica também claramente os tipos de segmentos trocados e os números de sequência usados quer nos dados como nas confirmações. (*(Nota: a transferência por FTP envolve mais que uma conexão FTP, nomeadamente uma de controlo [ftp] e outra de dados [ftp-data]. Faça o diagrama apenas para a conexão de transferência de dados do ficheiro mais pequeno)*)

tcp.stream eq 3

No.	Time	Source	Destination	Protocol	Length	Info
718	2011.1109421…	10.1.1.1	10.4.4.1	TCP	74	20 → 54855 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2321818725 TSecr=0 WS=128
719	2011.1128244…	10.4.4.1	10.1.1.1	TCP	74	54855 → 20 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=329056797 TSecr=2321818725 WS=128
720	2011.1129929…	10.1.1.1	10.4.4.1	TCP	66	20 → 54855 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2321818730 TSecr=329056797
722	2011.1135096…	10.1.1.1	10.4.4.1	FTP-DA…	192	FTP Data: 126 bytes (PORT) (LIST)
723	2011.1135131…	10.1.1.1	10.4.4.1	TCP	66	20 → 54855 [FIN, ACK] Seq=127 Ack=1 Win=29312 Len=0 TSval=2321818730 TSecr=329056797
724	2011.1137546…	10.4.4.1	10.1.1.1	TCP	66	54855 → 20 [ACK] Seq=1 Ack=127 Win=29056 Len=0 TSval=329056799 TSecr=2321818730
725	2011.1137553…	10.4.4.1	10.1.1.1	TCP	66	54855 → 20 [FIN, ACK] Seq=1 Ack=128 Win=29056 Len=0 TSval=329056799 TSecr=2321818730
726	2011.1139559…	10.1.1.1	10.4.4.1	TCP	66	20 → 54855 [ACK] Seq=128 Ack=2 Win=29312 Len=0 TSval=2321818730 TSecr=329056799

Fig. 9: FTP

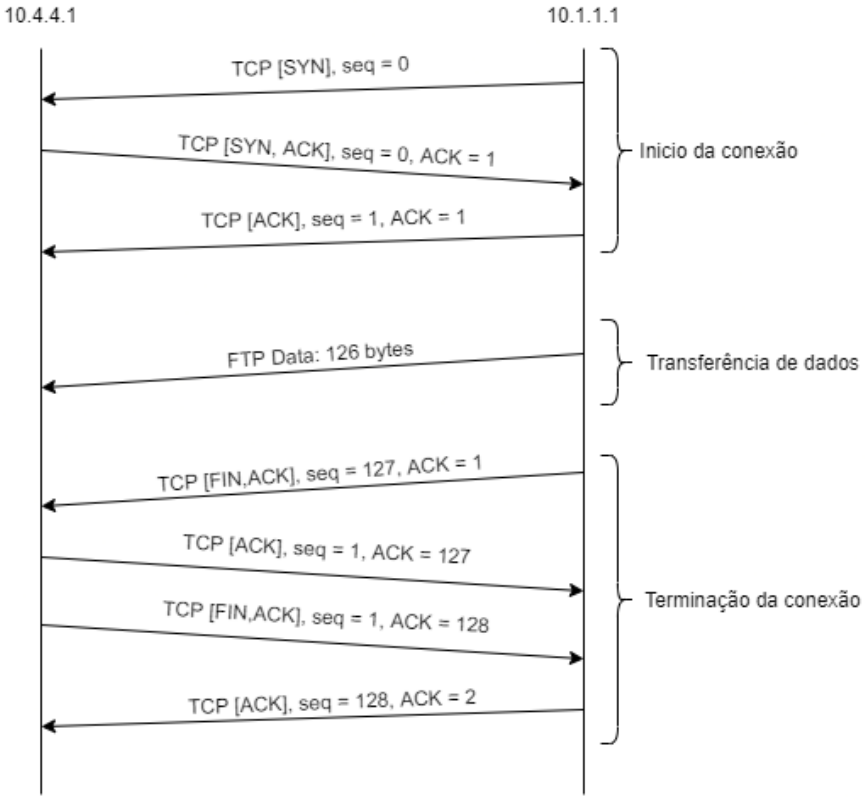


Fig. 10: Diagrama temporal das transferências do file1 por FTP

```
25 107.011916037 10.4.4.1    10.1.1.1    TFTP    56 Read Request, File: file1, Transfer type: octet
26 107.013641685 10.1.1.1    10.4.4.1    TFTP    46 Data Packet, Block: 1 (last)
27 107.013834105 10.4.4.1    10.1.1.1    TFTP    46 Acknowledgement, Block: 1
```

Fig. 11: TFTP

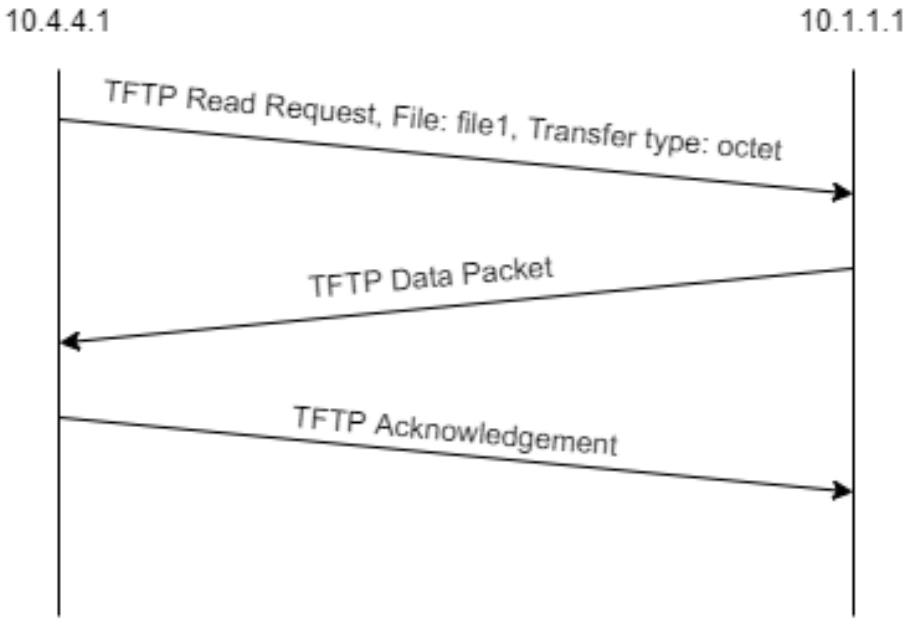


Fig. 12: Diagrama temporal das transferências do file1 por TFTP

Questão 3. Com base nas experiências realizadas, distinga e compare sucintamente as quatro aplicações de transferência de ficheiros que usou nos seguintes pontos (i) uso da camada de transporte; (ii) eficiência na transferência; (iii) complexidade; (iv) segurança;

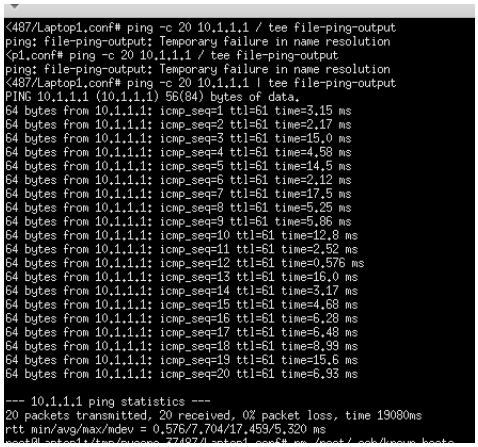
SFTP - Utiliza o protocolo TCP na camada de transporte.É um protocolo eficiente, na medida em que é possível transferir rapidamente vários ficheiros. Trata-se de uma versão segura do protocolo FTP, recorrendo a técnicas de encriptação e requer autenticação entre utilizador e servidor. Corre por cima do protocolo SSH, que fornece segurança e autenticação.

FTP - Utiliza o protocolo TCP na camada de transporte. Trata-se de um protocolo eficiente na transferência de grandes ficheiros, tanto para o uso do CPU do host como no uso da largura de banda da rede. Embora seja o protocolo de transferência de ficheiros mais comum em TCP/IP, é também o mais complexo e dificel de programar. Não é um protocolo seguro dado que não utiliza encriptação. Desta forma, os dados que os utilizadores usam para autenticação ficam vulneráveis e suscetiveis a ataques.

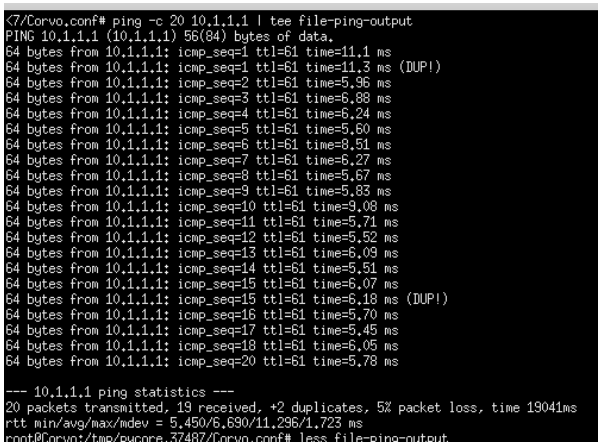
TFTP - Utiliza o protocolo UDP na camada de transporte. Desenhado para ser um mecanismo simples e leve para trocas de ficheiros, usa capacidade minima e tem menor overhead tornando-se assim mais eficiente. Em termos de complexidade, é uma alternativa mais simples do protocolo FTP e é igualmente inseguro. Não utiliza encriptação e permite que os dados da autenticação e os próprios dados transferidos circulem de forma clara pela rede.

HTTP - Utiliza o protocolo TCP na camada de transporte. É mais eficiente a transferir ficheiros de pequenas dimensões, ao contrário do protocolo FTP. Não utiliza métodos de encriptação nem há garantias de privacidade (qualquer pessoa pode ver/alterar o conteúdo). Trata-se de um protocolo pouco complexo.

Questão 4. As características das ligações de rede têm uma enorme influência nos níveis de Transporte e de Aplicação. Discuta, relacionando a resposta com as experiências realizadas, as influências das situações de perda ou duplicação de pacotes IP no desempenho global de Aplicações fiáveis (se possível, relacionando com alguns dos mecanismos de transporte envolvidos).



(a) Ping - Laptop1



(b) Ping - Corvo

Fig. 13: Resultado do comando ping ao Server1 em diferentes clientes

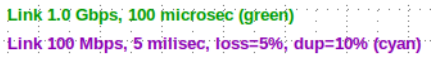


Fig. 14: Ligações

Observando as figuras Fig.13(a) e Fig.13(b) podemos verificar que, enquanto que no *Laptop1* houve 0% *packet loss* e nenhum pacote duplicado, no *Corvo* houve 5% *packet loss* e 2 pacotes duplicados (10%). A maior parte das vezes, isto acontece devido a problemas ao nível da ligação de rede, ora, como podemos verificar na topologia usada (Fig. 14), o *Laptop1* está conetado à rede *Backbone* por uma ligação de 1.0 Gbps enquanto que o *Corvo* está conetado por uma ligação de 100Mbps com 5%*loss* e 10%*dup* que vão exatamente ao encontro dos valores encontrados na Fig.13(b).

Quanto aos protocolos de transporte existem dois principais: TCP e UDP que tratam deste problema. O TCP é um protocolo orientado à conexão (*connection-oriented protocol*), envia *Delivery Acknowledgements*, re-transmissão no caso de erros, deteção fácil de erros... Os pacotes são enviados na ordem certa, usam um *handshake protocol* como *SYN, SYN-ACK, ACK*. Além disso, TCP é de confiança pois garante a entrega de informação no destino. No entanto, este protocolo atrasa a transmissão quando a rede está congestionada e a sua substituição por outro protocolo não é fácil.

Em contrapartida, o UDP é um protocolo sem conexão (*connectionless protocol*), é suportado em aplicações com uso intensivo de largura de banda que toleram a perda de pacotes, tem menos delay que o TCP, os pacotes não são enviados numa ordem fixa, cada pacote é independente, existe verificação de erros mas pacotes errados são descartados. A entrega de informação não é garantida no protocolo UDP. Como desvantagens, no UDP um pacote pode não ser entregue ou entregue fora de ordem e pacotes não são retransmitidos em caso de colisões.

Conclusão

Com este trabalho prático conseguimos consolidar os temas abordados nas aulas teóricas relativamente à camada de transporte. Em relação aos protocolos de transporte, foi possível ficar a conhecer mais acerca do funcionamento do TCP e UDP. Além disso, conseguimos aprofundar os conhecimentos e clarificar conceitos em relação aos serviços de transferência de ficheiros, tais como SFTP, FTP, TFTP e HTTP. Por fim, foi também explorada e testada a conectividade e caraterísticas das ligações.