**Programma laboratoriale di lavoro**

**Studio di protocolli Internet**

**Il protocollo UDP**

1. Generazione traffico UDP
   1. Ricerca comandi/applicativi specifici (peer-to-peer, client-server, per PC, per terminali mobili) per la generazione di traffico UDP

<https://play.google.com/store/apps/details?id=com.jca.udpsendreceive>

https://packetsender.com/

* 1. Valutazione delle problematiche incontrate nel loro effettivo funzionamento (presenza firewall, proxy, reti paritetiche e reti di dominio ecc.)

l’app deve essere presente su entrambi i dispositivi mobili, il programma windows deve avere accesso al firewall(disattivato)

1. Analisi dei pacchetti UDP
   1. Utilizzo di Wireshark per:
      * l’analisi di file .pcap forniti a lezione
      * la cattura e l’analisi del traffico generato secondo le metodologie sviluppate al punto precedente
   2. Compilazione di un questionario finalizzato all’individuazione ed all’analisi critica delle informazioni contenute nel datagramma UDP
   3. Calcolo del checksum
      * Verifica del checksum riscontrato nell’analisi di pacchetti (vedi punto precedente)
      * Sviluppo di un semplice software C# che analizzando i dati di un datagramma inserito in un file, ne calcoli e visualizzi automaticamente la checksum
2. Implementazione di un algoritmo di comunicazione e trasferimento dati basato sul protocollo UDP (es. trasferimento di immagini)

Esempio (PSEUDO-HEADER, HEADER, DATI)

**Pseudo-header** (from IP):

C0 A8 // source IP

00 80 //

C0 A8 // destination IP

00 8D //

00 11 // zero + protocol (UDP=17)

00 0C // UDP length (data included)

**Header** (from UDP)

82 35 // source port

82 35 // destination port

00 0C // UDP length

00 00 // checksum cleared

**Data (from UDP)**

43 69 // C i

61 6F // a o

**Trasmittente**

Formare le word a 16-bit e sommarle a 32 bit :

C0A8 + 0080 + C0A8 + 008D +

0011 + 000C + 8235 + 8235 +

000C + 0000 + 4369 + 616F =

= **0003 2BC8**

Sommiamo i carry accumulati nei 16 bit più significativi con il valore dei 16 bit meno significativi.

**0003 2BC8 = 2BCB**

Adesso calcoliamo il C1 della somma ottenuta:

~**2BCB** = **D434 (checksum calcolato)**

Si spedisce

C0A8 0080 C0A8 008D 0011 000C 8235 8235 000C **D434** 4369 616F

**Ricevente**

Calcoliamo la somma considerando anche il valore di checksum

C0A8 + 0080 + C0A8 + 008D +

0011 + 000C + 8235 + 8235 +

000C + **D434** + 4369 + 616F =

= **0002 FFFD**

Sommiamo i carry accumulati nei 16 bit più significativi con il valore dei 16 bit meno significativi.

FFFD + 0002 = **FFFF (OK)**

**Calcolo del Checksum** (con UDP)

Esempio per macchina a 32 bit. Segue il calcolo checksum secondo le specifiche espresse nel RFC 1071.

* I byte adiacenti sono presi due a due, così come compaiono in Wireshark, per formare elementi a 16 bit;
* Si esegue poi la loro somma a 16 bit usando 32 bit per il risultato;
* Nella somma bisogna considerare anche i riporti (carry) che si accumulano nei 16 bit più significativi dei 32 totali
* Del risultato a 32 bit ottenuto, si sommano i 16 bit meno significativi con i 16 bit più significativi (contenenti i riporti)
* Si effettua il complemento a 1 del risultato (C1);
* Il risultato viene trascritto nel campo checksum del datagramma UDP;
* Il ricevente esegue la somma come prima ma questa volta considera anche il valore di checksum presente
* Se il risultato è **FFFF** (cioè ~0) tutto è **OK**, altrimenti NO.