

Retele de calculatoare – Informatica anul 3 (2019-2020)

Note de Laborator
Retele de calculatoare

Specializare: Informatica anul 3

Contact:
retelecdsd@gmail.com
<http://www.cdsd.ro>

Comunicatii de
Date si
Sisteme
Distribuite



<http://www.cdsd.ro>

Motto: "Gandeste inainte sa vorbesti. Citeste inainte sa gandesti."

Frances Ann Lebowitz

Laborator 5

1. Obiective:

- PDU - Verificarea integritatii folosind CRC-32 - Aplicatii Java / Python pentru calculul sumei de verificare.
- Adresarea IP: Subnetting - calculul adreselor IP ale subretelelor, domeniilor de adrese IP alocate hosturilor, adreselor IP de broadcast ale subretelelor; Aplicatii Java / Python.
- Studiu de caz: Planificarea unei retele (utilizatori/servicii/hosturi) - Aplicatie **Riverbed Modeler Academic Edition** – mediu de simulare a retelelor de calculatoare (**Varianta: OMNeT++ Network Simulation Framework** <http://www.omnetpp.org/>)
- Aplicatii de retea in Python

2. Consideratii teoretice (**Partea practica – pag.9; Tema – pag.21**)

2.1. PDU - Verificarea integritatii folosind CRC-32

- **PDU** – Protocol Data Unit
- **CRC** = Cyclic Redundance Code – contine restul sumei de verificare ciclica a redundantei (suma ciclica de control) calculat polinomial prin CRC-32. Secventa CRC (FCS – Frame Check Sequence) este dată de restul împărțirii acestui polinom la un polinom generator primitiv standard.
- Standardul IEEE 802 foloseste următorul **polinom generator primitiv**:

$$\text{CRC32} = X^{32} + X^{26} + X^{25} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1.$$

Observatie (recapitulare!):

Fie $f(x) = f_n x^n + f_{n-1} x^{n-1} + \dots + f_1 x + f_0$, cu f_n si f_0 nenuli. Considerand coeficientii polinomului in Z_2 (pot fi doar 0 sau 1!), un polinom $f(x)$ este primitiv daca este:

- irreductibil: nu exista nici un polinom de grad nenul si mai mic de n care sa fie divizor al lui $f(x)$
- daca α este o radacina a lui $f(x)$ ($f(\alpha) = 0$) atunci cel mai mic intreg p pentru care $\alpha^i = \alpha^{i+p}$ este $p = 2^n - 1 = 2^{\text{grad}(f)} - 1$.

Notatii: **SLE** = Statia de Lucru Emitatoare; **SLD** = Statia de Lucru Destinatie

SLD primește frame-ul, face propriul calcul CRC-32 si compara valoarea calculata cu cea aflata in campul CRC din pachet (valoarea calculata la emisie de SLE), concluzionand daca frame-ul a sosit intact. In caz de alterare, informatia este distrusa bit cu bit, cerandu-se retransmiterea la nivelul protocoalelor de pe nivelurile OSI superioare.

2.2. Exemplu (<https://crccalc.com/>)

| Message (ASCII) | CRC-32 (Hexa) |
|-------------------|---------------|
| Computer Networks | 0xB07330AE |
| Computer networks | 0x490552F8 |

Retele de calculatoare – Informatica anul 3 (2019-2020)

2.3. CRC-32 ([2 CRC Error detection.pdf](#); [3 CRC exercises.pdf](#))

Valorile CRC-32 reprezinta o modalitate foarte eficienta de verificare a integritatii datelor.

http://en.wikipedia.org/wiki/Cyclic_redundancy_check

<http://www34.brinkster.com/dizzyk/crc32.asp>

<http://www.createwindow.com/programming/crc32/crcfile.htm>

<http://webnet77.com/cgi-bin/helpers/crc.pl>

<http://www.softpedia.com/get/Others/Miscellaneous/CRC32-Calculator.shtml>

<http://www.wikiera.net/EthernetCRC-readytouseexample.html>

http://www.wireshark.org/docs/wsug_html_chunked/ChAdvChecksums.html

Exemplu (Anexa 3 – pag.33): Verificarea integritatii unui fisier. Pentru a detecta daca un fisier a fost corupt in cadrul procesului de transmisie, emitatorul calculeaza o semnatura digitala pe care o trimite impreuna cu fisierul. Destinatarul calculeaza si el semnatura fisierului, si daca aceasta se potriveste cu valoarea calculata de emitator, atunci fisierul nu a suferit modificari. CRC inseamna Cyclic Redundancy Check iar 32 indica faptul ca suma de verificare este pe 32 de biti. Probabilitatea ca doua fisiere sa aiba aceiasi suma de verificare este de 1 la $2^{32} = 4.294967E+09$. Din acest motiv CRC-32 poate fi folosit si pentru a identifica fisiere duplicat. CRC-ul se bazeaza pe operatia de impartire intr-un inel comutativ, numit inelul polinoamelor peste intregii modulo 2, adica polinoamele care au coeficientii doar de un bit. Orice sir de biti poate fi vazut ca fiind coeficientii unui astfel de polinom, iar pentru a afla codul CRC asociat se imparte polinomul cu un alt polinom, iar coeficientii polinomului rest sunt chiar bitii ce reprezinta codul CRC.

Exercitiu (recapitulare):

```
INITIAL VALUE = 01100101010000110010000100000000 = 0x65432100
x8 + x2 + x1 + 1 = 100000111
0100100100000110010000100000000
100000111
100011000110010000100000000
100000111
11111110010000100000000
100000111
111101110000100000000
100000111
1111000000010000000
100000111
1110011100010000000
100000111
1100100100100000000
100000111
10010101010000000
100000111
101101100000000
100000111
110101100000
100000111
101010110000
100000111
1010001000
100000111
CHECKSUM = 10000110 = 0x86
```

Algoritm:

1. Se genereaza tabela CRC-32 folosind un polinom specific.
2. Se foloseste o functie de calcul CRC care executa un XOR intre un octet din datele care trebuie verificate si valoarea curenta CRC, obtinandu-se astfel o noua valoare CRC.
3. Algoritmul CRC citeste primul octet din datele ce urmeaza sa fie verificate si apeleaza functia CRC care intoarce valoarea CRC pentru octetul respectiv. Se apeleaza apoi functia CRC cu urmatorul octet si valoarea CRC calculata anterior. Dupa al doilea apel, valoarea CRC reprezinta suma de verificare pentru primii doi octeti. Se apeleaza functia CRC pana cand toti octetii din fisier au fost procesati. Ultima valoare pentru CRC este suma de verificare pentru datele initiale.

Retele de calculatoare – Informatica anul 3 (2019-2020)

Metoda “hardware” de calcul pentru CRC foloseste operatii pe biti. Ca o alternativa la calculul sumei de verificare bit cu bit, se poate utiliza o tabela cu valori precalculate care permite efectuarea a 8 operatii pe biti simultan. Pentru CRC-32, tabela este constituita din 256 de elemente de tip Double Word (4 octeti). Un DWORD poate fi reprezentat ca un intreg pe 32 de biti cu sau fara semn (in implementarea Java se poate folosi tipul int).

- Un anumit CRC este definit prin polinomul folosit pentru calcularea lui. Pentru a calcula un CRC pe n biti este necesar un polinom de gradul n de forma $x^n + \dots + 1$. Acesta este reprezentat pe $n+1$ biti, dar cum termenul x^n este implicit se iau in considerare doar ultimii n biti. De obicei coeficientii polinomului se reprezinta hexazecimal, astfel pentru standardul CRC-16, $x^{16} + x^{15} + x^2 + 1$, va fi reprezentat in hexazecimal ca 0x8005.
- Cel mai intalnit CRC este CRC-32, fiind folosit printre altele de Ethernet, FDDI (Fiber Distributed Data Interface), PKZIP, WinZip si PNG (Portable Network Graphics). Polinomul lui poate fi scris ca fiind 0x04C11DB7. Mai jos este lista cu cele mai folosite CRC-uri si polinoamele lor asociate:

| | |
|------------------------|---|
| CRC-8 | $x^8 + x^2 + x + 1$ |
| CRC-12 | $x^{12} + x^{11} + x^3 + x^2 + x + 1$ |
| CRC-CCITT | $x^{16} + x^{12} + x^5 + 1$ |
| CRC-16 (IBM) | $x^{16} + x^{15} + x^2 + 1$ |
| CRC-32 (802.3) | $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ |
| CRC32c (Castagnoli) | $x^{32} + x^{28} + x^{27} + x^{26} + x^{25} + x^{23} + x^{22} + x^{20} + x^{19} + x^{18} + x^{14} + x^{13} + x^{11} + x^{10} + x^9 + x^8 + x^6 + 1$ |

- CRC-ul este util in detectarea erorilor, dar nu si pentru verificarea integritatii datelor deoarece cunoscandu-se polinomul folosit datele pot fi intentionat modificate astfel incat CRC-ul sa ramana neschimbat !!!!!!!!!!!!!. Pentru integritate: MD5...MD6; SHA

CRC-32 - Implementare

1. XOR intre primul octet de date si octetul mai putin semnificativ al valorii CRC pentru a obtine un index.
2. Deplasare la dreapta fara semn cu 8 biti a valorii CRC.
3. XOR intre valoarea CRC si valoarea TabelaCRC[index]
4. Se repeta pasii de la 1 la 3 pentru toti octetii

2.4. Clase de adrese IPv4

2.4.1. Adresa IP este exprimata pe 32 biți (în versiunea IPv4) / 128 biti in versiunea IPv6 si conține suficiente informații pentru a identifica în mod unic o rețea sau o gazdă aparținând unei rețele. Toate dispozitivele si protocoalele de pe **nivelul rețea - modelul OSI** / **nivelul Internet – modelul TCP/IP** folosesc adrese IP. Formatul adresei corespunde notatiei cu punct zecimal. Aceasta notatie imparte o adresa IP in 4 campuri de cate 8 biti fiecare, si specifica valoarea fiecarui camp ca un numar zecimal. Campurile sunt separate prin puncte. **Formate:** in zecimal (cu punct zecimal); binar.

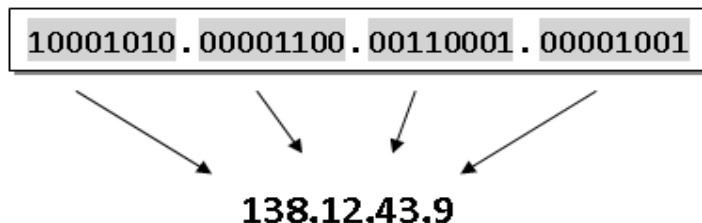


Fig 4.1 : Conversia unei adrese IP in formatul zecimal cu punct

Rețele de calculatoare – Informatica anul 3 (2019-2020)

| Notatie cu punct | Notatie Binara |
|------------------|-------------------------------------|
| 217.145.43.12 | 11011001 10010001 00101011 00001100 |
| 192.168.0.1 | 11000000 10101000 00000000 00000001 |
| 10.2.1.123 | 00001010 00000010 00000001 01111011 |

Fig 4.2 : Exemple de adrese IP

- adresă IP conține un câmp care identifică rețeaua sau subrețeaua la care este atașată gazda identificată de către respectiva adresă și un câmp care identifică în mod unic fiecare gazdă din rețea. Numărul de biți folosiți pentru identificarea rețelei (**câmpul NetId**) și numărul de biți folosiți pentru identificarea host-ului (**câmpul HostId**) **depind de clasa adresei (pentru classfull), si nu depind de clasa, pentru classless.**
- Există cinci clase de adrese: A, B, C, D și E, dintre care numai primele trei sunt disponibile pentru adresarea rețelelor standard. Adresele de clasă D sunt rezervate pentru adresarea la nivel de grupuri (tip multicast) în timp ce adresele de clasă E sunt rezervate pentru scopuri experimentale (nefolosite pentru a adresa gazde sau grupuri multicast).

Următorul tabel rezumă caracteristicile fiecărei clase de adrese:

| Clasa | Primul Octet | Intervalul de adrese | NetId (octeți) | HostId (octeți) | Nr. de rețele | Nr. de host-uri /rețea |
|-------|--------------|---------------------------|----------------|-----------------|----------------------|------------------------|
| A | 0xxxxxxx | 0.0.0.0-127.255.255.255 | 1 | 3 | 128 (2 rezervate) | 16.777.214 |
| B | 10xxxxxx | 128.0.0.0-191.255.255.255 | 2 | 2 | 16.384 | 65.534 |
| C | 110xxxxx | 192.0.0.0-223.255.255.255 | 3 | 1 | 2.097.152 | 254 |
| D | 1110xxxx | 224.0.0.0-239.255.255.255 | - | - | - | - |
| E | 1111xxxx | 240.0.0.0-255.255.255.0 | - | - | - | - |

Fig 4.3 Caracteristicile claselor de adrese IP v4

2.4.2. Adrese IP speciale

Orice adresă al cărei prim octet este 127 este o adresă de buclă, folosită la diagnosticare și testare. Un mesaj transmis unei adrese IP al cărei prim octet are valoarea 127 este returnat expeditorului. 255 este un octet ce indică o difuzare în rețeaua locală (broadcast) – expedierea multiplă a unui mesaj. Primul octet nu poate avea o valoare mai mare decât 223. Celelalte adrese sunt rezervate pentru anunțuri în rețele și pentru scopuri experimentale. Ultimul octet al unui identificator de gazdă nu poate fi 0 sau 255.

Adrese private: 10.0.0.0–10.255.255.255 10.0.0.0 /8

172.16.0.0–172.31.255.255 172.16.0.0 /12

192.168.0.0–192.168.255.255 192.168.0.0 /16

Obs: **169.254.0.0 to 169.254.255.255*** (65,536 possible IP addresses) **169.254.0.0 /16**

* The range, 169.254.x.x, is a special IP address range that is reserved for devices to self-assign an address. You generally do not manually assign addresses in this range.

2.4.3. Masca de subrețea

Este o machetă de biți ce definește porțiunea din adresa IP ce reprezintă NetID. În plus, masca de subrețea permite subalocarea adresei de rețea (*subnetting*). Subalocarea înseamnă împărțirea unei adrese IP în sensul folosirii acesteia în mai multe rețele mai mici. Formatul măștii de subrețea corespunde unei **zone continue de 1 (NetID)** și unei **zone continue de 0 (HostID)**, cu reprezentarea în zecimal, pe câmpuri de 8 biți. **Exemplu: 255.255.224.0 scrisă și /19.** Posibile valori ale unui octet al măștii de subrețea:

Rețele de calculatoare – Informatica anul 3 (2019-2020)

| Binar | Zecimal |
|-----------|---------|
| 0000.0000 | 0 |
| 1000.0000 | 128 |
| 1100.0000 | 192 |
| 1110.0000 | 224 |
| 1111.0000 | 240 |
| 1111.1000 | 248 |
| 1111.1100 | 252 |
| 1111.1110 | 254 |
| 1111.1111 | 255 |

Măștile de rețea predefinite, corespunzătoare claselor de adrese A,B si C sunt prezentate în tabelul următor.

| Clasă A | | Clasă B | | Clasă C | |
|-----------|----|-------------|-----|---------------|-----|
| 255.0.0.0 | /8 | 255.255.0.0 | /16 | 255.255.255.0 | /24 |



2.4.3.1. Folosirea măștii de subrețea

Plecând de la o adresă IP ce aparține uneia din clasele A, B si C, exprimată binar, se realizează un **“si” logic la nivel de bit cu masca de subrețea folosită, exprimată tot binar. Alegerea măștii de subrețea se face din considerentul obținerii numărului dorit de subrețele, pentru fiecare din acestea permitându-se un număr de hosturi cerut de datele de proiectare.**

Exemplu (AND pentru bitii de pe aceeași poziție):

Adr. Host: 205.101.55.91 in format binar 11001101 01100101 00110111 01011011
 mască: 255.255.255.224 in format binar 11111111 11111111 11111111 11100000

205.101.55.62 ← 11001101 01100101 00110111 01000000
 Rezultă că adresa 205.101.55.91 aparține subrețelei 205.101.55.62

Retele de calculatoare – Informatica anul 3 (2019-2020)

Pentru cei trei biti „împrumutati” (nrbi=3) din partea de HOST ID putem construi $2^3=8$ subrețele, cu următoarele valori posibile: 000–111.

Cele 8 subrețele desemnate de masca de subrețea 255.255.255.224 **vor fi asociate următoarelor domenii de valori ale celui de-al 4-lea octet din adresa IP:**

| | | |
|-------------------|-----------------------|-----------|
| Subrețea 0 | 000_00001 – 000_11110 | 1 – 30 |
| Subrețea 1 | 001_00001 – 001_11110 | 33 – 62 |
| Subrețea 2 | 010_00001 – 010_11110 | 65 – 94 |
| Subrețea 3 | 001_00001 – 001_11110 | 97 – 156 |
| Subrețea 4 | 100_00001 – 100_11110 | 129 – 158 |
| Subrețea 5 | 101_00001 – 101_11110 | 161 – 190 |
| Subrețea 6 | 110_00001 – 110_11110 | 193 – 222 |
| Subrețea 7 | 111_00001 – 111_11110 | 225 – 254 |

| | | |
|------------|---|---|
| AND | 0 | 1 |
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| | | |

Concluzie: Folosind formatul 224 pentru masca de subrețea, practic, împrumutăm 3 biti din câmpul *hostid* în scopul realizării subalocării adresei IP (de clasă C) pentru un număr de 8 subrețele cerute. Pentru fiecare din subrețele prezentate, avem un număr de $2^5-2 = 30$ de adrese IP pentru hosturi.

IP ANDbit SM => Network Address; IP ORbit Not(SM) => Broadcast Address

2.4.3.2. Evaluarea “gradului” de utilizare a unei adrese de clasă C

Adresă clasa C: *Netid* (24 biti) + *Hostid* (8 biti) = NT hosturi.

Mască de subrețea /26 \Rightarrow NS (Număr subrețele disponibile) = $2^2 = 4$ și NH (Număr hosturi pe fiecare subrețea) = $2^6 - 2 = 62$.

$$\text{Grad de utilizare} = \frac{\text{NS} * \text{NH}}{\text{NT}}$$

2.4.3.3. Metode Java pentru lucrul cu clase de adrese (exemple)

2.4.3.3.1. Metoda care întoarce clasa unei adrese IP

```
public static char getClass(InetAddress ia) {
    byte[] address = ia.getAddress();

    int firstByte = address[0];
    if ((firstByte & 0x80) == 0) return 'A';
    else if ((firstByte & 0xC0) == 0x80) return 'B';
    else if ((firstByte & 0xE0) == 0xC0) return 'C';
    else if ((firstByte & 0xF0) == 0xE0) return 'D';
    else if ((firstByte & 0xF8) == 0xF0) return 'E';
    else return 'F';
}
```

2.4.3.3.2. Metoda pentru conversia unei adrese IP salvată sub forma unui tablou de bytes în format cu punct zecimal.

```
int unsignedByte = signedByte < 0 ? signedByte + 256 : signedByte;
```

2.4.3.4. Operatori pe biti în limbajul Java

| | | | | | | | | | | | |
|-------------------|---|---|--------------|---|---|---------------|---|---|---------------|---|---|
| &(AND) | 0 | 1 | (OR) | 0 | 1 | ^(XOR) | 0 | 1 | ~(NOT) | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | | | |

Retele de calculatoare – Informatica anul 3 (2019-2020)

Operatorii de deplasare (<< , >> si >>>) se definesc astfel:

- $a \ll b = a * 2^b$
- $a \gg b = a / 2^b$.
- $a \ggg b$ este egala cu valoarea lui a deplasata la dreapta cu b pozitii prin completarea la stanga cu zerouri

2.5. Riverbed Modeler Academic Edition

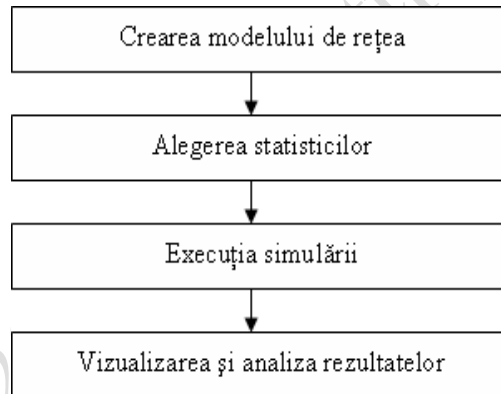
Ovservatie: (Lab_1 + Lab_2 + Lab_3 + Lab_4 si toate celelalte materiale, prezente la <http://www.cdsd.ro> ...F.F.F.Importante)

Riverbed Modeler Academic Edition (versiune actuala a Opnet-ului - **Optimized Network Application and Network Performance**) – mediu de simulare a rețelilor de calculatoare - furnizează software de management pentru aplicații și rețele, care oferă soluții pentru:

- Planificarea capacității rețelilor,
- Modelare și simulare pentru rețele și aplicații
- Managementul configurării rețelilor
- Managementul performanțelor aplicațiilor

Varianta “programare” C++: [OMNeT++ Network Simulation Framework](http://www.omnetpp.org/), <http://www.omnetpp.org/>

Riverbed oferă o versiune academică (**Modeler Academic Edition**) - include modele standard pentru protocoale și echipamentele disponibile în tehnologia IT (disponibile, după instalare, în subdirectoare). Etapele de lucru avute în vedere:



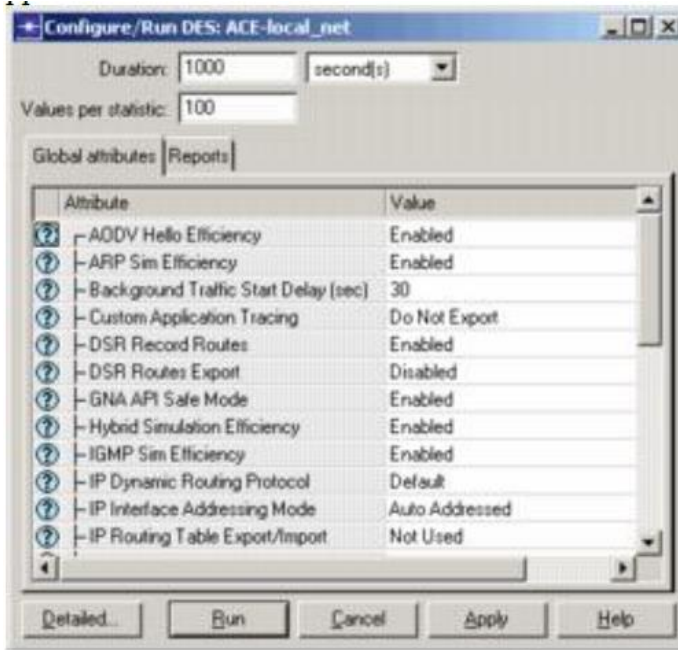
Etapele de lucru pentru Modeler Academic Edition pentru simularea și analiza unei rețele

Obs: O statistica este o caracteristica numerica a unui esantion (Anexa 3, Lab_03)

- **Statistica** este stiinta colectarii, clasificarii, prezentarii, interpretarii datelor numerice si a folosirii acestora pentru a formula concluzii si a lua decizii.
- **Statistica descriptiva** (Descriptive Statistics) se ocupa cu colectarea, clasificarea si prezentarea datelor numerice.
- **Statistica inferentiala** (Inferential Statistics) se ocupa cu interpretarea datelor oferite de statistica descriptiva si cu folosirea acestora pentru a formula concluzii si lua decizii.

Configure/Run DES Dialog Box (Simple) The Configure/Run DES dialog box lets you configure and run a discrete event simulation for the current scenario. The simple version of the dialog box, (shown in the following figure), which appears when the DES configuration mode is set to “simple”, presents a reduced set of controls to simplify configuration and execution of **discrete event simulations**. Only single simulation runs are supported.

Rețele de calculatoare – Informatica anul 3 (2019-2020)



The simple Configure/Run DES dialog box has two pages of controls. These controls are organized by type and can be selected by clicking the corresponding tab. The following table lists the controls in this dialog box.

| Element | Description |
|------------------------|---|
| Basic controls | Duration field—Sets the duration of the simulation. Specify units with the pull-down menu following this field. This value sets the “duration” simulation preference. |
| | Values per statistic field—Sets the maximum number of values collected for each statistic. This value sets the “num_collect_values” simulation preference. |
| Global Attributes page | Use this page to define the values of global simulation attributes. |
| | This page is similar to the Global Attributes page—Used to define the values of global simulation attributes for the simulation. seen in Detailed mode, except that you cannot set multiple values for an attribute or automatically reset the default value. |
| Reports page | Use this page to select Statistic reports and Service Level Agreement (SLA) reports for the simulation. Reports are predefined sets of statistic probes. |
| | This page is identical to the Configure/Run DES Dialog Box (Detailed)—Report Controls seen in Detailed mode. |
| Dialog box controls | Detailed... button—Switches temporarily to detailed mode and the detailed Configure/Run DES dialog box, as described in Configure/Run DES Dialog Box (Detailed) . (This button does not change the des.configuration_mode preference.) |
| | Run button—Saves the current settings, closes the dialog box, and runs the simulation. Running a simulation from here opens the Simulation Execution Dialog Box . |
| | Cancel button—Closes the dialog box without saving any changed settings. |
| | Apply button—Saves the current settings and keeps the dialog box open. |
| | Help button—Opens a help file for the dialog box. |

3. Partea practica (tema – pag.22 !!!!!)

RECAPITULARE: [2 CRC Error detection.pdf](#); [3 CRC exercises.pdf](#)

3.1. Program Java ce permite calculul CRC-32 pentru un sir de caractere introdus de la tastatura (se poate folosi fisierul [6 Lab 05 CRC table.pdf](#)). **Challenge:** Aplicatie Python

Anexa 4: The Programming Process !

Indicatii:

```
import java.io.*;

public class CRC32 {
    int crc32val;

    // Tabela CRC-32
    static int tabel_crc32[] = {
        0x00000000, 0x77073096, 0xee0e612c, 0x990951ba, 0x076dc419,
        0x706af48f, 0xe963a535, 0x9e6495a3, 0x0edb8832, 0x79dcb8a4,
        0xe0d5e91e, 0x97d2d988, 0x09b64c2b, 0x7eb17cbd, 0xe7b82d07,
        0x90bf1d91, 0x1db71064, 0x6ab020f2, 0xf3b97148, 0x84be41de,
        0x1dad47d, 0x6ddde4eb, 0xf4d4b551, 0x83d385c7, 0x136c9856,
        0x646ba8c0, 0xfd62f97a, 0x8a65c9ec, 0x14015c4f, 0x63066cd9,
        0xfa0f3d63, 0x8d080df5, 0x3b6e20c8, 0x4c69105e, 0xd56041e4,
        0xa2677172, 0x3c03e4d1, 0x4b04d447, 0xd20d85fd, 0xa50ab56b,
        0x32b5a8fa, 0x42b2986c, 0xdbb9c9d6, 0xacb9cf940, 0x32d86ce3,
        0x45df5c75, 0xdcd60dcf, 0xabd13d59, 0x26d930ac, 0x51de003a,
        0xc8d75180, 0xbfd06116, 0x21b4f4b5, 0x56b3c423, 0xcfba9599,
        0xb8bda50f, 0x2802b89e, 0x5f058808, 0xc60cd9b2, 0xb10be924,
        0x2f6f7c87, 0x58684c11, 0xc1611dab, 0xb6662d3d, 0x76dc4190,
        0x01db7106, 0x98d220bc, 0xefd5102a, 0x71b18589, 0x06b6b51f,
        0x9fbfe4a5, 0xe8b8d433, 0x7807c9a2, 0x0f00f934, 0x9609a88e,
        0xe10e9818, 0x7f6a0dbb, 0x086d3d2d, 0x91646c97, 0xe6635c01,
        0x6b6b51f4, 0x1c6c6162, 0x856530d8, 0xf262004e, 0x6c0695ed,
        0x1b01a57b, 0x8208f4c1, 0xf50fc457, 0x65b0d9c6, 0x12b7e950,
        0x8bbbeb8ea, 0xfcb9887c, 0x62dd1ddf, 0x15da2d49, 0x8cd37cf3,
        0xfbd44c65, 0x4db26158, 0x3ab551ce, 0xa3bc0074, 0xd4bb30e2,
        0x4adfa541, 0x3dd895d7, 0xa4d1c46d, 0xd3d6f4fb, 0x4369e96a,
        0x346ed9fc, 0xad678846, 0xda60b8d0, 0x44042d73, 0x33031de5,
        0xaa0a4c5f, 0xdd0d7cc9, 0x5005713c, 0x270241aa, 0xbe0b1010,
        0xc90c2086, 0x5768b525, 0x206f85b3, 0xb966d409, 0xce61e49f,
        0x5edef90e, 0x29d9c998, 0xb0d09822, 0xc7d7a8b4, 0x59b33d17,
        0x2eb40d81, 0xb7bd5c3b, 0xc0ba6cad, 0xedb88320, 0x9abfb3b6,
        0x03b6e20c, 0x74b1d29a, 0xeada54739, 0x9dd277af, 0x04db2615,
        0x73dc1683, 0xe3630b12, 0x94643b84, 0x0d6d6a3e, 0x7a6a5aa8,
        0xe40ecf0b, 0x9309ff9d, 0x0a00ae27, 0x7d079eb1, 0xf00f9344,
        0x8708a3d2, 0x1e01f268, 0x6906c2fe, 0xf762575d, 0x806567cb,
        0x196c3671, 0x6e6b06e7, 0xfed41b76, 0x89d32be0, 0x10da7a5a,
```

Retele de calculatoare – Informatica anul 3 (2019-2020)

```
0x67dd4acc, 0xf9b9df6f, 0x8eb9eff9, 0x17b7be43, 0x60b08ed5,
0xd6d6a3e8, 0xa1d1937e, 0x38d8c2c4, 0x4fdff252, 0xd1bb67f1,
0xa6bc5767, 0x3fb506dd, 0x48b2364b, 0xd80d2bda, 0xaf0a1b4c,
0x36034af6, 0x41047a60, 0xdf60efc3, 0xa867df55, 0x316e8eef,
0x4669be79, 0xcb61b38c, 0xbc66831a, 0x256fd2a0, 0x5268e236,
0xcc0c7795, 0xbb0b4703, 0x220216b9, 0x5505262f, 0xc5ba3bbe,
0xb2bd0b28, 0x2bb45a92, 0x5cb36a04, 0xc2d7ffa7, 0xb5d0cf31,
0x2cd99e8b, 0x5bdeae1d, 0x9b64c2b0, 0xec63f226, 0x756aa39c,
0x026d930a, 0x9c0906a9, 0xeb0e363f, 0x72076785, 0x05005713,
0x95bf4a82, 0xe2b87a14, 0x7bb12bae, 0x0cb61b38, 0x92d28e9b,
0xe5d5be0d, 0x7cdcefb7, 0x0bdbdf21, 0x86d3d2d4, 0xf1d4e242,
0x68ddb3f8, 0x1fda836e, 0x81be16cd, 0xf6b9265b, 0x6fb077e1,
0x18b74777, 0x88085ae6, 0xff0f6a70, 0x66063bca, 0x11010b5c,
0x8f659eff, 0xf862ae69, 0x616bffd3, 0x166ccf45, 0xa00ae278,
0xd70dd2ee, 0x4e048354, 0x3903b3c2, 0xa7672661, 0xd06016f7,
0x4969474d, 0x3e6e77db, 0xaed16a4a, 0xd9d65adc, 0x40df0b66,
0x37d83bf0, 0xa9bcae53, 0xdeb9ec5, 0x47b2cf7f, 0x30b5ffe9,
0xbdbdf21c, 0xcabac28a, 0x53b39330, 0x24b4a3a6, 0xbad03605,
0xcdd70693, 0x54de5729, 0x23d967bf, 0xb3667a2e, 0xc4614ab8,
0x5d681b02, 0x2a6f2b94, 0xb40bbe37, 0xc30c8ea1, 0x5a05df1b,
0x2d02ef8d
};

//constructor
public CRC32() {
    crc32val = 0;
}

// metoda de calcul a CRC-32
public static int crc32(byte[] s){
    CRC32 crc = new CRC32();
    crc.calcul(s,0,s.length);
    return crc.getValoareCRC();
}

public void calcul(byte[] s, int index, int dim) {
    for (int i=index; i<dim; i++) {
        crc32val =
            tabel_crc32[(crc32val ^ s[i]) & 0xff] ^ (crc32val >>> 8);
    }
}

public int getValoareCRC() {
    return crc32val;
}

public static void main(String args[]) {
    try{
        CRC32 crc = new CRC32();
    }
}
```

Retele de calculatoare – Informatica anul 3 (2019-2020)

```
BufferedReader br = new BufferedReader(  
    new InputStreamReader(System.in));  
  
System.out.print("Introduceti textul=");  
  
String s = br.readLine();  
byte [] sir = s.getBytes();  
  
String formatare =  
    Long.toHexString(((long)crc.crc32(sir))&0xffffffffL);  
System.out.println("CRC32: " + formatare);  
  
}catch(IOException ioe){  
    System.out.println(ioe.getMessage());}  
}  
}
```

3.2. Program pentru calculul sumei de verificare pentru un fisier. Se utilizeaza clasa [java.util.zip.CRC32](#); Challenge: Aplicatie Python

Indicatii:

```
import java.io.*;  
import java.util.zip.CheckedInputStream;  
import java.util.zip.CRC32;  
  
public class CRC32Fisier{  
  
    public static void main(String [] args){  
  
        try {  
  
            CRC32 crc32 = new CRC32();  
  
            CheckedInputStream cis = new CheckedInputStream(  
                new FileInputStream("test.txt"), crc32);  
  
            byte[] tempBuf = new byte[128];  
            while (cis.read(tempBuf) >= 0) {  
  
                }  
  
            long checksum = cis.getChecksum().getValue();  
            System.out.println("Valoare CRC-32 (long) :"+checksum);  
  
            String formatare = Long.toHexString(checksum&0xffffffffL);  
            System.out.println("Valoare CRC-32 (hex) :"+formatare);  
  
        }catch(FileNotFoundException fnfe){  
            System.err.println(fnfe.getMessage());  
            System.exit(-1);  
        }catch (IOException ioe) {  
            System.err.println(ioe.getMessage());  
            System.exit(-1);  
        }  
    }  
}
```

Retele de calculatoare – Informatica anul 3 (2019-2020)

3.3. Realizati un program care sa ofere utilizatorului urmatoarele optiuni:

- salvarea unui text intr-un fisier impreuna cu suma de verificare CRC-32 corespunzatoare, verificarea integritatii unui fisier pe baza valorii CRC-32. **Challenge:** Aplicatie Python

Anexa 4: The Programming Process!

Indicatii:

```
import java.io.*;
import java.util.zip.CRC32;

public class SumaVerificare {
    public static void main(String args[]) throws IOException {

        FileWriter fw = new FileWriter("out.txt");
        BufferedWriter bw = new BufferedWriter(fw);

        BufferedReader br = new BufferedReader(
            new InputStreamReader(System.in));

        CRC32 checksum = new CRC32();

        String s = br.readLine();
        //actualizeaza suma de verificare
        while (!(s.equals("Exit"))){
            bw.write(s);
            bw.newLine();
            checksum.update(s.getBytes());
            s = br.readLine();
        }

        // scrie suma de verificare

        bw.write(Long.toString(checksum.getValue()));
        bw.newLine();

        bw.close();
        br.close();
    }
}
```

Metodele clasei CRC32

| | |
|---|--|
| <code>public void update(byte[] b)</code> | Actualizeaza suma de verificare cu octetii din tabloul b |
| <code>public long getValue()</code> | intoarce valoarea CRC-32 |
| <code>public void reset()</code> | reseteaza valoarea CRC-32 |

[`CheckedInputStream`](#)([`InputStream`](#) in, [`Checksum`](#) cksum)

Creaza un flux de intrare folosind suma de verificare cksum specified cksum.

[`CheckedOutputStream`](#)([`OutputStream`](#) out, [`Checksum`](#) cksum)

Creaza un flux de iesire folosind suma de verificare cksum specified cksum.

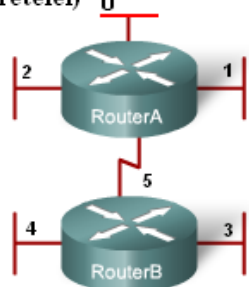
[`java.util.zip`](#) – pachet care contine clase pentru calcularea sumelor de verificare(CRC32, Adler32)

Rețele de calculatoare – Informatica anul 3 (2019-2020)

3.4. Exerciții Subnetting (Subalocarea unei adrese IP)

3.4.1. O organizație a primit adresa de rețea **192.168.1.0/24** și are nevoie de șase subrețele. Cea mai mare dintre aceste subrețele trebuie să conțină 25 de host-uri. Definiți masca de subrețea corespunzătoare și completați **Tabelul de adrese IP** (determinați adresele subrețelelor, adresele hosturilor, adresele de broadcast pentru toate subrețelele obținute) și **Tabelul de adresare IP** (adresele interfețelor ruterele (alocate descrescător, începând cu ultima valoare!), adresele gateway-urilor rețelelor conectate la rutere, a (n+3)-a adresa a hostului corespunzătoare apartenenței la rețeaua n)

| | | | |
|---|---|--|---|
| Adresa IP inițială: | - | 192.168.1.0 (/24) 255.255.255.0 | Address: 11000000.10101000.00000001.00000000 Mask: 11111111.11111111.11111111.00000000 |
| Se obțin 8 Subrețele; | 0 | 192.168.1.0 (/27) 255.255.255.224 | Address: 11000000.10101000.00000001.00000000 Mask: 11111111.11111111.11111111.11100000 |
| Se alocă SR0 - SR5 (6 SR-uri); | 1 | 192.168.1.32 (/27) 255.255.255.224 | Address: 11000000.10101000.00000001.00100000 Mask: 11111111.11111111.11111111.11100000 |
| SR6; SR7 (rezerva - dezvoltarea ulterioară a rețelei) | 2 | 192.168.1.64 (/27) 255.255.255.224 | Address: 11000000.10101000.00000001.01000000 Mask: 11111111.11111111.11111111.11100000 |
| | 3 | 192.168.1.96 (/27) 255.255.255.224 | Address: 11000000.10101000.00000001.01100000 Mask: 11111111.11111111.11111111.11100000 |
| | 4 | 192.168.1.128 (/27) 255.255.255.224 | Address: 11000000.10101000.00000001.10000000 Mask: 11111111.11111111.11111111.11100000 |
| | 5 | 192.168.1.160 (/27) 255.255.255.224 | Address: 11000000.10101000.00000001.10100000 Mask: 11111111.11111111.11111111.11100000 |
| | 6 | 192.168.1.192 (/27) 255.255.255.224 | Address: 11000000.10101000.00000001.11000000 Mask: 11111111.11111111.11111111.11100000 |
| | 7 | 192.168.1.224 (/27) 255.255.255.224 | Address: 11000000.10101000.00000001.11100000 Mask: 11111111.11111111.11111111.11100000 |



Tabelul de adrese IP:

| Nr SR | Adresa SR | Interval adrese hosturi | Adresa broadcast SR |
|-------|-----------|-------------------------|---------------------|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Tabelul de adresare IP:

| | Rețea 0 | Rețea 1 | Rețea 2 | Rețea 3 | Rețea 4 | Rețea 5 |
|---------------------------------------|---------|---------|---------|---------|---------|---------|
| Router A (interfață) | | | | | | |
| Router B (interfață) | | | | | | |
| Gateway | | | | | | |
| Adresa hostului (n+3), cu n=nr. rețea | | | | | | |

Retele de calculatoare – Informatica anul 3 (2019-2020)

3.4.2. O organizatie a primit adresa de retea **179.27.0.0/18** si are nevoie de 12 subretele. Cea mai mare dintre aceste subretele trebuie sa contina 102 host-uri. Definiti masca de subretea corespunzatoare ramanerii in rezerva a unor subretele cu un numar cat mai mare de hosturi si determinati adresele subretelelor precum si adresele gazdelor din subretea 2. Calculati adresele de broadcast pentru fiecare din subretelele obtinute.

3.4.3. O organizatie a primit adresa de retea **132.137.0.0/16** si trebuie sa defineasca subretele care sa suporte fiecare maxim 60 de gazde. Definiti masca de subretea corespunzatoare si determinati adresele subretelelor precum si adresele gazdelor din subretea 3. Calculati adresele de broadcast pentru fiecare subretea.

3.4.4. Se da adresa de retea **187.123.151.0/24**. Cati biti trebuie imprumutati din campul *hostid* pentru a permite definirea unor subretele care sa suporte minim 20 de gazde? Care este numarul maxim de gazde care poate fi suportat de fiecare subretea? Care este numarul maxim de subretele care poate fi definit? Determinati subretelele si exprimati adresele IP in format binar si in formatul cu punct zecimal. Care este domeniul de adrese IP folosit de subretea 6? Care este adresa de broadcast pentru subretea 6?

3.5. Realizati o aplicatie Java pentru subnetting. Adresa IP si masca de subretea sunt preluate de la tastatura. Programul trebuie sa afiseze:

- Clasa din care face parte adresa.
- Numarul de biti pentru campul *hostid*, respectiv campul *netid*.
- Adresa subretelei din care face parte adresa IP.
- Masca de subretea pentru clasa de adrese din care face parte adresa IP.
- Numarul de subretele posibile in retea specificata.
- Numarul de gazde pentru fiecare subretea in parte.
- Adresele host-urilor pentru fiecare subretea in parte, de exemplu:
Subretea 125.13.67.96 , Adrese Hosturi 125.13.67.97 - 125.13.67.126, Adresa Broadcast Subretea
- Adresa de broadcast a retelei
- Gradul de utilizare al adresei IP
- Programul trebuie sa salveze toate datele intr-un fisier.

Challenge: Aplicatie Python

Indicatii:

Aplicatie care testeaza validitatea unei adresa IP si afiseaza clasa din care face parte si adresa in format binar.

```
public class TestIP{

    public static void testareAdresa(String strIP){

        boolean err = false;
        String output = null;
        int adresa=0;

        String [] sIP = strIP.split("\\.");
```


Retele de calculatoare – Informatica anul 3 (2019-2020)

```
if(sIP.length!=4) {
    err= true;
    output = "Adresa IP incorecta: o adresa IP are 4 campuri numerice!\n";
    return;
}

int [] intIP= new int[4];
for(int i=0; i<4; i++){
    intIP[i]= Integer.parseInt(sIP[i]);
    if((intIP[i]<0||intIP[i]>255)) {
        err= true;
        output = "Adresa IP incorecta:" +
            + "Campurile unei adrese IP pot lua valori numai intre 0 si 255!\n";
        return;
    }
}

for(int i=3; i>=0; i--)
    adresa= adresa|(intIP[i] << (8*(3-i)));

System.out.println("Adresa IP este de clasa " + getClass(intIP[0]));
System.out.println("In format binar " + toBits(adresa));
System.out.println("Eroare=" + err);
System.out.println("Output=" + output);
}

public static char getClass(int firstByte) {
    if ((firstByte & 0x80) == 0) return 'A';
    else if ((firstByte & 0xC0) == 0x80) return 'B';
    else if ((firstByte & 0xE0) == 0xC0) return 'C';
    else if ((firstByte & 0xF0) == 0xE0) return 'D';
    else if ((firstByte & 0xF8) == 0xF0) return 'E';
    else return 'F';
}

//Citeste un bit
public static int getBit(int nr, int b){
    if(b>31) return 0;
    int x=1;
    x=x<<b;
    if((nr&x)!=0) return 1;

    else return 0;
}

public static String toBits(int n){
    String ret="";
    for(int i=31;i>=0; i--){
        ret+=getBit(n,i);
    }
}
```

Retele de calculatoare – Informatica anul 3 (2019-2020)

```
}  
    return ret;  
}  
  
public static void main(String [] args) throws Exception{  
  
    String strIP = "192.168.0.1";  
    testareAdresa(strIP);  
}  
}
```

3.6. Planificarea (ca etapa a proiectarii) unei rețele cu un număr diferit de utilizatori, hosturi, servicii. Se va parcurge integral materialul [Network Design Modeler.pdf](#) (Indicatie: <https://www.youtube.com/watch?v=e8XBDZLtKrE>)

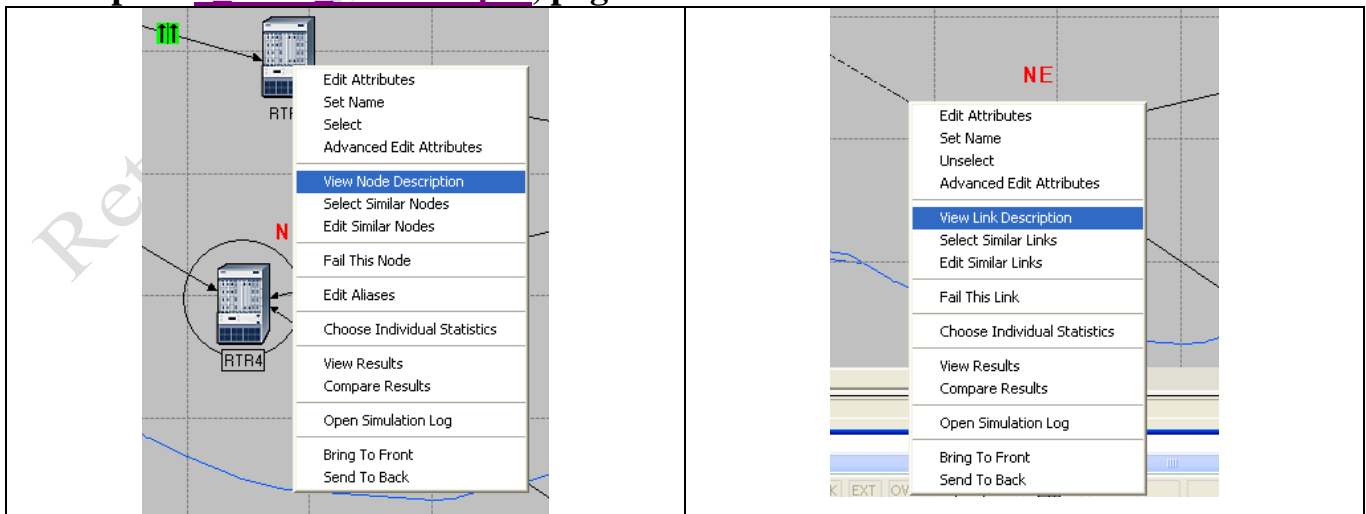
- **Cursantii sunt incurajati** sa foloseasca materialul de mai jos intr-un mod constructiv, astfel incat sa evalueze caracteristicile legaturilor fizice si a dispozitivelor de retea folosite (*click dreapta, view link/node description*), a modelelor de retea precum si elementele de baza privind simularea **sistemelor cu evenimente discrete** analizate; rezultatele obtinute grafic vor fi analizate si interpretate in contextul cerintelor prezentate in sectiunea *Lab Report*.

Obiectiv: Parcurgerea etapelor de baza privind planificarea unei rețele, tinand cont de utilizatori, servicii, hosturi, localizarea hosturilor. **Indicatie:** [Network Design Modeler.pdf](#)

ATENTIE:

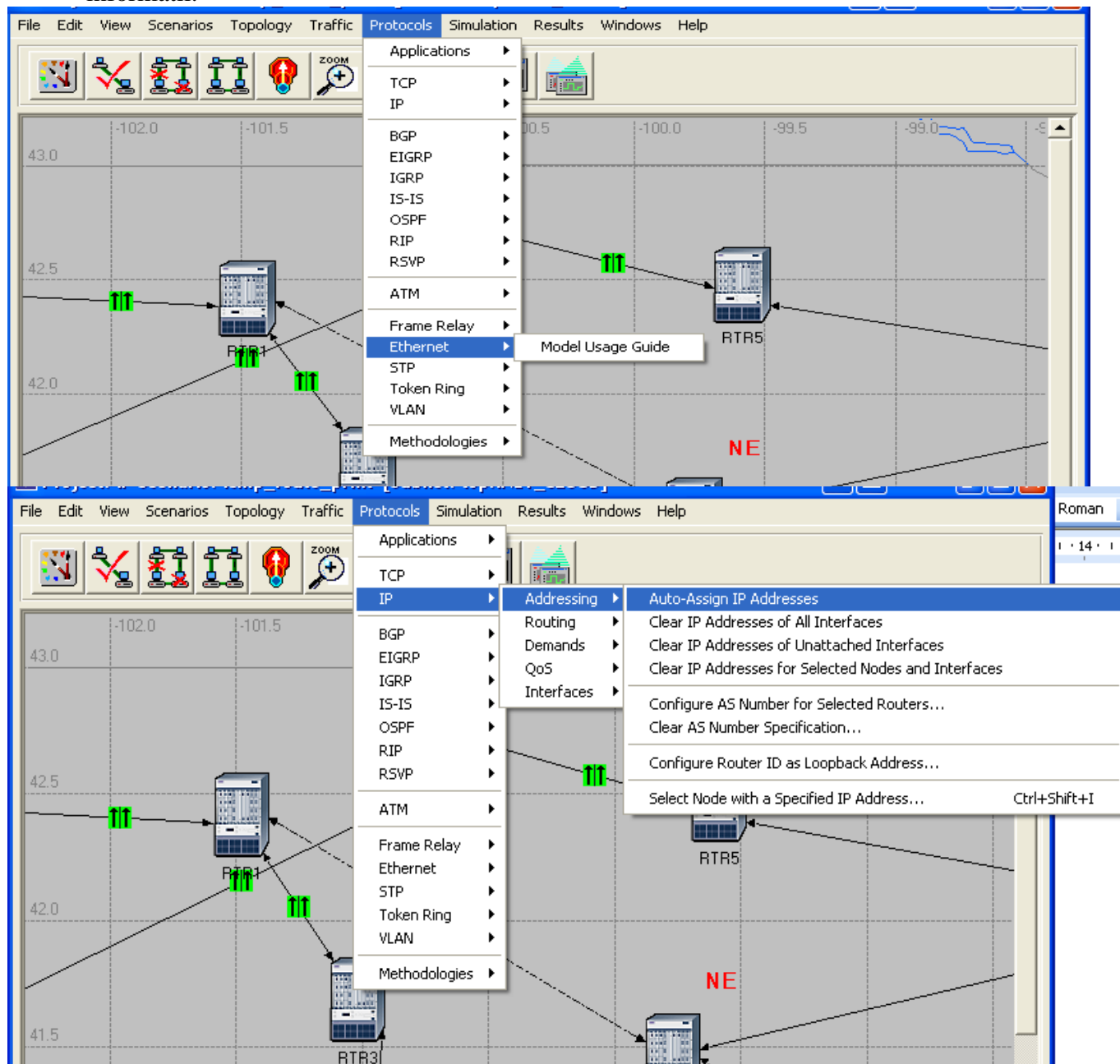
- **Cursantii sunt incurajati** sa foloseasca materialul de mai sus intr-un mod constructiv, astfel incat sa evalueze caracteristicile legaturilor fizice si a dispozitivelor de retea folosite (*click dreapta, view link/node description*), a modelelor de retea precum si elementele de baza privind simularea **sistemelor cu evenimente discrete** analizate; rezultatele obtinute grafic vor fi analizate si interpretate in contextul cerintelor prezentate in sectiunea *Lab Report*.

Exemplu: [4 Intro Modeler.pdf](#), pag.7

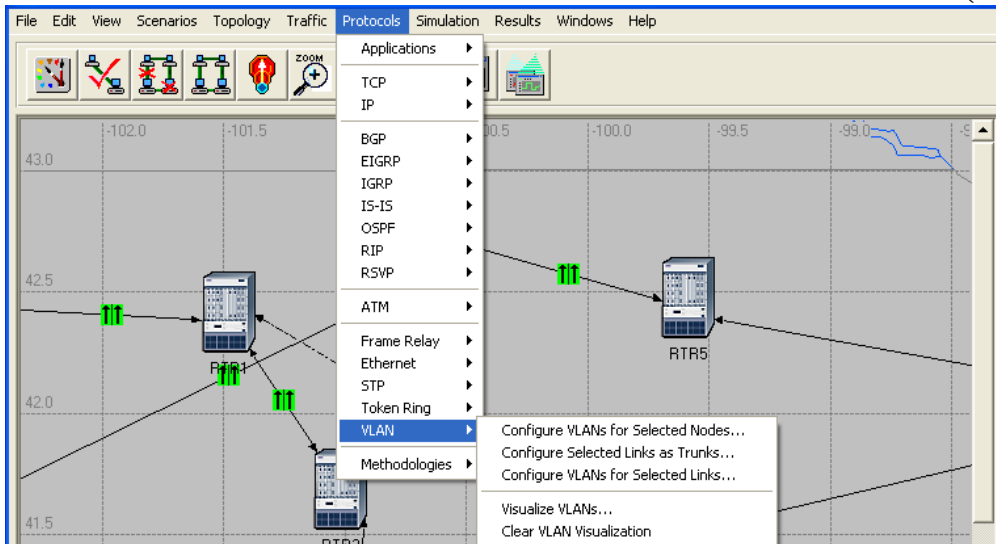


Retele de calculatoare – Informatica anul 3 (2019-2020)

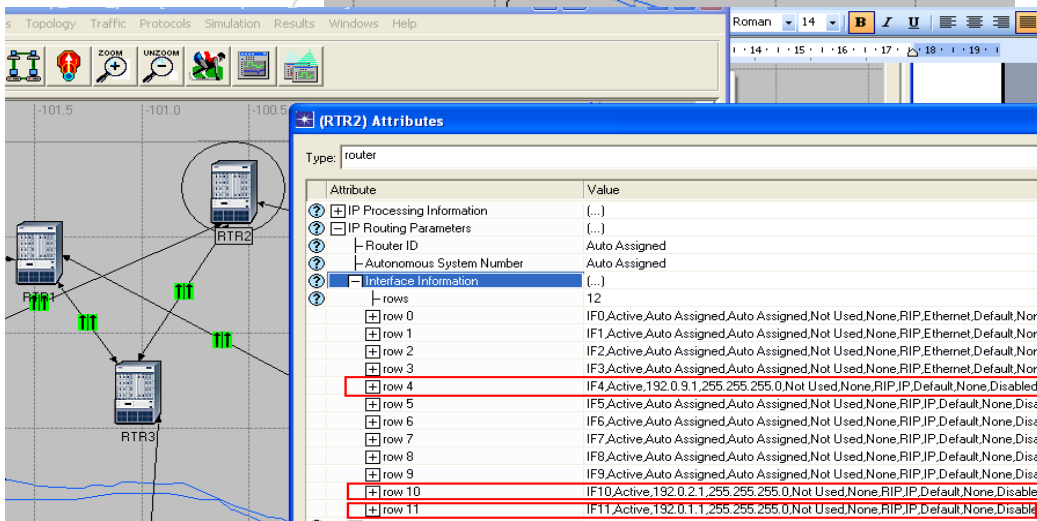
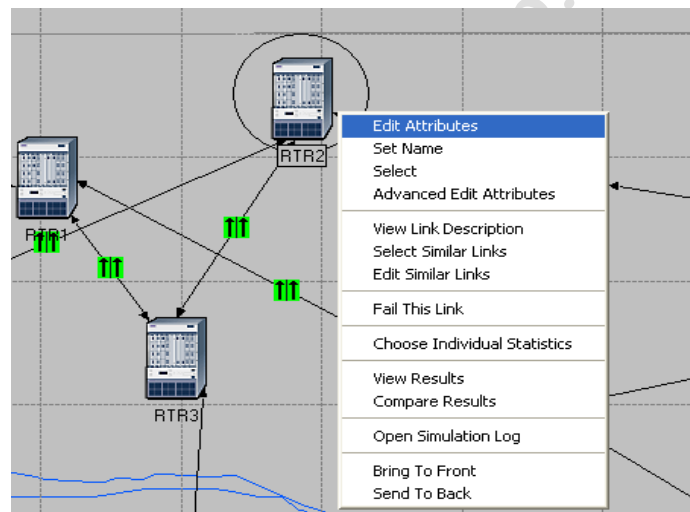
- Se vor identifica meniurile si sub-meniurile de mai jos, in contextul utilizarii ulterioare a acestor informatii.



Retele de calculatoare – Informatica anul 3 (2019-2020)



- Pe baza informatiilor privind adresarea IP pentru fiecare nod in parte, se va completa Tabelul de adrese IP, conform modelului de la pag 14, punctul 3.4.1.



3.7. Aplicatii de retea in Python

3.7.1. Recapitulare (Lab_02, Lab_03)

- Python_intro
- Programare_Python
- Byte-of-python
- [pyshark](#) – pachet Python care permite analiza PDU-urilor folosind decodarea Wireshark

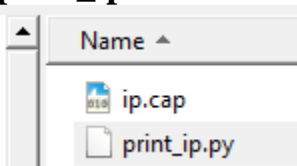
3.7.2. Aplicatie: Afisarea adreselor IP sursa si destinatie corespunzatoare pachetelor din fisierul wireshark [ip.cap](#)

Indicatii:

print_ip.py

```
1  # print_ip.py
2
3  import pyshark
4
5  #cap = pyshark.FileCapture('http.cap', display_filter="dns")
6  #for pkt in cap:
7  #    print(pkt)
8
9  cap = pyshark.FileCapture('ip.cap', keep_packets=False)
10
11 def print_info_layer(packet):
12     print("[Protocol:] "+packet.highest_layer+" [Source IP:] "
13           +packet.ip.src+" [Destination IP:] "+packet.ip.dst)
14
15 cap.apply_on_packets(print_info_layer)
16
```

print_ip:



Output:

```
Anaconda Prompt (Anaconda3)

(base) C:\          >d:
(base) D:\>cd D:\          \print_ip
(base) D:\          \print_ip>python print_ip.py
[Protocol:] TCP [Source IP:] 145.254.160.237 [Destination IP:] 165.208.228.223
[Protocol:] TCP [Source IP:] 65.208.228.223 [Destination IP:] 145.254.160.237
[Protocol:] TCP [Source IP:] 145.254.160.237 [Destination IP:] 165.208.228.223
[Protocol:] HTTP [Source IP:] 145.254.160.237 [Destination IP:] 165.208.228.223
[Protocol:] TCP [Source IP:] 65.208.228.223 [Destination IP:] 145.254.160.237
[Protocol:] TCP [Source IP:] 65.208.228.223 [Destination IP:] 145.254.160.237
[Protocol:] TCP [Source IP:] 145.254.160.237 [Destination IP:] 165.208.228.223
[Protocol:] TCP [Source IP:] 65.208.228.223 [Destination IP:] 145.254.160.237
[Protocol:] TCP [Source IP:] 145.254.160.237 [Destination IP:] 165.208.228.223
```

Retele de calculatoare – Informatica anul 3 (2019-2020)

3.7.3 Aplicatie python: Afisarea adreselor IP sursa / destinatie filtrate cu ip ca data de intrare, corespunzatoare pachetelor din fisierul wireshark [ip.cap](#)

Exemple:

| | | | |
|--|----------------|---|-----------------|
| Input: ip sursa = 145.254.160.237 | | Input: ip destinatie = 145.254.160.237 | |
| Output: | | Output: | |
| 145.254.160.237 | 65.208.228.223 | 65.208.228.223 | 145.254.160.237 |
| 145.254.160.237 | 65.208.228.223 | 65.208.228.223 | 145.254.160.237 |
| 145.254.160.237 | 65.208.228.223 | 65.208.228.223 | 145.254.160.237 |
| 145.254.160.237 | 65.208.228.223 | 65.208.228.223 | 145.254.160.237 |
| 145.254.160.237 | 65.208.228.223 | 65.208.228.223 | 145.254.160.237 |
| 145.254.160.237 | 65.208.228.223 | 65.208.228.223 | 145.254.160.237 |
| 145.254.160.237 | 145.253.2.203 | 65.208.228.223 | 145.254.160.237 |
| 145.254.160.237 | 65.208.228.223 | 65.208.228.223 | 145.254.160.237 |
| 145.254.160.237 | 216.239.59.99 | 145.253.2.203 | 145.254.160.237 |
| 145.254.160.237 | 65.208.228.223 | 65.208.228.223 | 145.254.160.237 |
| 145.254.160.237 | 65.208.228.223 | 65.208.228.223 | 145.254.160.237 |
| 145.254.160.237 | 65.208.228.223 | 65.208.228.223 | 145.254.160.237 |
| 145.254.160.237 | 216.239.59.99 | 216.239.59.99 | 145.254.160.237 |
| 145.254.160.237 | 65.208.228.223 | 216.239.59.99 | 145.254.160.237 |

Pentru aplicatiile 3.7.2 si 3.7.3:

- Documentarea solutiilor ([pyshark](#), etc)
- Challenge:** Interfata grafica

Recomandare: Qt Designer , cu Designer din Anaconda prompt). <http://pythonforengineers.com/your-first-gui-app-with-python-and-pyqt/>, <https://www.codementor.io/deepak Singh04/design-simple-dialog-using-pyqt5-designer-tool-ajskrd09n>, <https://wiki.python.org/moin/PyQt/Tutorials>

Anexa 4: The Programming Process !!!

Observatii TEMA!!!!

DE ANALIZAT [readme-ul readme_mod_work_dir.pdf](#) (si un numai!... de exemplu si [readme_lab_modeler.pdf](#)) de la adresa <http://www.cdsd.ro>

Observatii

1. Atentie (Modeler) – Proiectul creat se salveaza implicit in:

C:\Users\student(NUMÉ user)\op_model\NUMÉ_PROIECT

NUMÉ_PROIECT contine proiectul modeler propriu-zis

VARIANTA

se arhiveaza intreg folderul *Folder creat mai jos...el contine proiectul opnet propriu-zis*

- In directorul\Studenti\Info3\Nume_Prenume se creează directorul (pentru punctul 3.5) \L5_3.6_Nume_Prenume folosind:
 - File → New → Folder
- Se lansează în execuție Modeler

Retele de calculatoare – Informatica anul 3 (2019-2020)

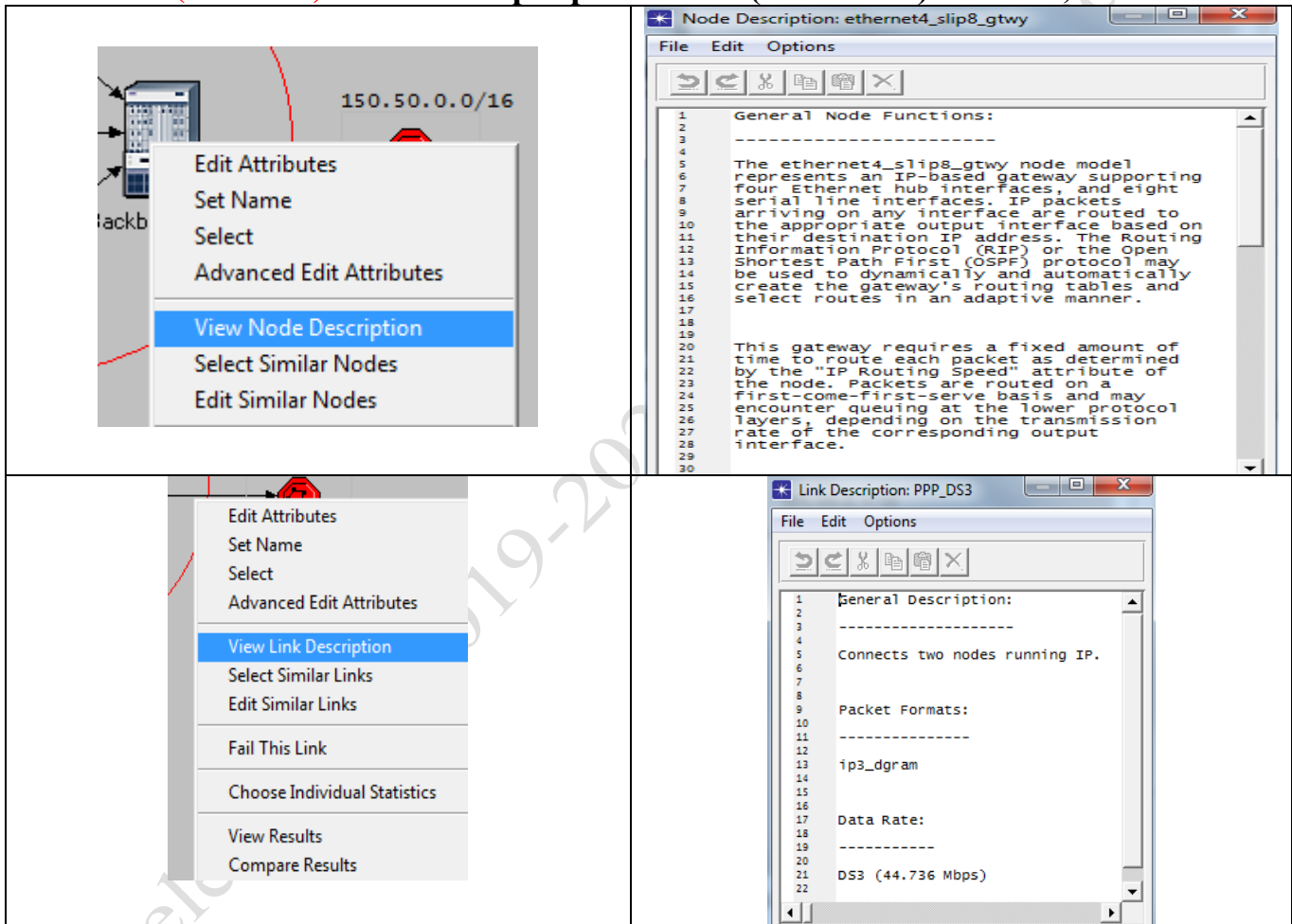
- Se selectează directorul în care vor fi plasate fișierele proiectului.
 - **File** → **Model Files** → **Add Model Directory**
 - Se selectează directorul în care se va lucra (în acest director vor fi salvate fișierele proiectului curent)
 - Se arhiveaza **L5_3.6_Nume_Prenume**

Atentie (Modeler Academic Edition) – se foloseste

readme_mod_work_dir.pdf (este prezent in arhiva **Lab_01**)

pentru a identifica folderul op_models in care se salveaza default *proiectul Modeler*.

2. Atentie (Modeler) : Click dreapta pe “obiect” (ex. Router)...”Judec, deci exist!”



.....similar omnet++..... (<http://www.omnetpp.org>)

4. Tema:

- Toate punctele din sectiunea 3 “partea practica” se vor relua de catre cursanti, folosind etapele de lucru indicate. **Rezultatele experimentale:**
 - **L5_nume+prenume_Modeler (folder)** - contine proiectul Modeler/Omnet++(3.6) si L5_nume+prenume_Modeler.doc (document .doc): rezultatele experimentale: comentarii insotite de capturi corespunzatoare proiectului Modeler/ Omnet++ (3.6) pasi intermediari importanti/topologia fizica, rezultate/capturi pentru View node description si View link

Retele de calculatoare – Informatica anul 3 (2019-2020)

description (obs.2 anterioara), exercitiile rezolvate, raspunsuri la intrebari, rezultate finale, observatii finale)

- **L5_nume+prenume_java** : folder cu fisierele Java pentru fiecare aplicatie (fisierele sursa .java, fisierul **.bat**, insotite de un *readme.txt* pentru particularitati de rulare + snipping tool) - 3.1, 3.2, 3.3, 3.5 (**Challenge: Versiuni Python**).
- **L5_nume+prenume_IP**: folder cu rezolvarile (.doc/ .png) ale exercitiilor 3.4.
- **L5_nume+prenume_Python (folder)** – cu subfolderele 3.7.2 si 3.7.3 (fiecare din acestea contine scripturile .py si document .doc (snipping tool) pentru aplicatiile Python, conform temelor indicate la aplicatiile de la punctul 3.7).

Se vor arhiva cu numele **L5_nume+prenume_info3.rar**. ARHIVA **se va trimite prin e-mail** (**VARIANTA:** <http://www.wetransfer.com>) la adresa retelecdsd@gmail.com precizandu-se la subject: **L5_nume+prenume_info3**, pana pe data de **7 noiembrie 2019**, ora **08.00 a.m.** (**Atentie, gmail nu "prea vrea" .rar in .rar** <http://www.makeuseof.com/tag/4-ways-email-attachments-file-extension-blocked/>).

Cursantii sunt incurajati sa analizeze si sa comenteze rezultatele obtinute, studiind si materialele (**ACTUALIZATE AICI:** <http://www.cdscd.ro> !) indicate in bibliografie si anexe. (+ **Recapitulare Lab_01, Lab_02, Lab_03, Lab_04**)

DE ANALIZAT **readme-ul** **readme_mod_work_dir.pdf** (si un **numai!...** de exemplu si **readme_lab_modeler.pdf**) **de la adresa** <http://www.cdscd.ro>

Observatie: **Studentii** dornici de afirmare....pot opta sa foloseasca pe langa Modeler (sau ca varianta), **framework-ul Omnet++** www.omnetpp.org/ , cu pastrarea scenariilor pentru aplicatie, descrise in laborator. **Se acorda "bonusuri! ...SUBSTANTIALE!"**

ATENTIE (pregatire **testul grila!!!**): [2 CRC Error detection.pdf](#); [3 CRC exercises.pdf](#); Subnetting; VLSM

Obs:

| Punctaj maxim (Data trimiterii temei) | | | |
|---------------------------------------|-------------|------------|------------|
| <= 07.11. 2019 | 11.11. 2019 | 15.11.2019 | 19.11.2019 |
| 100 pct | 80 pct | 60 pct | 50 pct |



Sursa: <http://www.funnfun.in/wp-content/uploads/2013/06/steps-of-success-encouraging-quote.jpg>

Retele de calculatoare – Informatica anul 3 (2019-2020)

How to send an e-mail

<http://lifehacker.com/5803366/how-to-send-an-email-with-an-attachment-for-beginners>

<https://support.google.com/mail/answer/6584?hl=en> “As a security measure to prevent potential viruses, Gmail doesn't allow you to send or receive [executable files](#) (such as files ending in .exe).”

<https://support.google.com/mail/answer/2480713?hl=en>

<http://fastupload.ro/free.php>

<http://www.computerica.ro/siteuri-transfer-fisiere-mari-upload/>

Bibliografie

Lab_01, Lab_02, Lab_03, Lab_04, TL_01

<http://www.cdsd.ro/cursuri>

efg' Mathematics, <http://www.efg2.com/Lab/Mathematics/CRC.htm>

Java API, <https://docs.oracle.com/javase/8/docs/api/>

Java Tutorial, Writing Your Own Filtered Streams <http://www.rgagnon.com/javadetails/java-0416.html>

http://en.wikipedia.org/wiki/Cyclic_redundancy_check

<http://www34.brinkster.com/dizzyk/crc32.asp>

<http://www.createwindow.com/programming/crc32/crcfile.htm>

<http://webnet77.com/cgi-bin/helpers/crc.pl>

<http://www.softpedia.com/get/Others/Miscellaneous/CRC32-Calculator.shtml>

<http://www.wikiera.net/EthernetCRC-readytouseexample.html>

http://www.wireshark.org/docs/wsug_html_chunked/ChAdvChecksums.html

Modeler Tutorials

https://rpmapps.riverbed.com/ae/4dcgi/SIGNUP_NewUser

<https://supportkb.riverbed.com/support/index?page=content&id=S24443>

https://rpmapps.riverbed.com/ae/4dcgi/DOWNLOAD_HOME

https://rpmapps.riverbed.com/ae/4dcgi/REG_TransactionCode

- Install Riverbed Modeler 17.5 Windows 10, 8.1, 8 and 7 (<https://www.youtube.com/watch?v=TpenN2jYbHQ>)
- Install Riverbed Modeler (<https://www.youtube.com/watch?v=DQ3XhHYuFGA>)
- How to activate riverbed modeler 17.5 (<https://www.youtube.com/watch?v=h-lmeJMqiSA>)
- How to solve invalid activation of Opnet Modeler 17.5 (<https://www.youtube.com/watch?v=13ZBcXkW46s>)
- Riverbed Modeler 17.5 Tutorial - Switched Lan (<https://www.youtube.com/watch?v=XdebwQLrr0w>)
- 6-Virtual LAN (VLAN) configuration in OPNET Riverbed (<https://www.youtube.com/watch?v=Ajz7bVO5WJM>)
- Riverbed Modeler Configuracion VLAN (<https://www.youtube.com/watch?v=rP3jPMcyEFk>)
- Ethernet (lab_04)
- Riverbed Opnet 17.5 Tutorial - The Ethernet network (https://www.youtube.com/watch?v=fS_J6ApFJtc)
- 6-Virtual LAN (VLAN) configuration in OPNET Riverbed (<https://www.youtube.com/watch?v=Ajz7bVO5WJM>)
- Riverbed Modeler Tutorial 3 Configuracion VLAN (<https://www.youtube.com/watch?v=rP3jPMcyEFk>)

<https://www.python.org/>

Retele de calculatoare – Informatica anul 3 (2019-2020)

Anexa 1 :

Atentie ! Pentru aplicatiile Java, la fiecare laborator (n) va fi creat un proiect in NetBeans (ex: Ln_info3) , care va fi salvat in directorul personal – F:\info3\<nume student>.

1. Crearea unui proiect nou

Pentru a crea un proiect nou selectam File -> New Project. Se va deschide o fereastră cu tipurile de proiecte pentru care se ofera suport. Se va alege General -> Java Application si se apasa Next.

Se va deschide o fereastră in care trebuie introduse numele si locatia noului proiect. Se deselecteaza caseta CreateMainClass, si se apasa Finish. Se va observa in fereastră Projects noul proiect creat, care contine: un director pentru fisierele sursa, un director pentru fisierele test si doua directoare cu alte informatii utilizate in proiect.

2. Crearea claselor

Pentru a crea o noua clasa, se va selecta New -> New File. Se va deschide o fereastră cu mai multe tipuri de aplicatii.

Funcție de tipul de clasa necesar, se alege optiunea si se apasa Next. Se introduce numele noii clase create, se selecteaza proiectul in cadrul caruia se creaza clasa (din lista Projects) si se apasa Finish. Se va observa faptul ca numele clasei este adaugat in directorul cu fisiere sursa din fereastră Projects.

3. Compilarea si rularea unui program Java

Se deschide clasa ce trebuie compilata, alegand-o din fereastră Projects, directorul Source Packages. Pentru compilare:

- Se selecteaza Build -> Compile *nume_clasa.java* (sau se apasa F9)
- Se executa click dreapta pe clasa selectata, si se selecteaza Compile File

Se observa in fereastră Output, situata in partea inferioara a interfetei NetBeans, rezultatul compilarii programului: erorile (daca sunt) sau mesajul BUILD SUCCESSFUL.

In ultimul caz, se va selecta Run -> Run Main Projects (daca se doreste lansarea in executie a intregului proiect) sau Run File (daca se doreste doar rularea clasei respective).

Rularea clasei se face si prin comanda Run File din meniul pop-up afisat la executarea de click dreapta pe clasa selectata.

4. Rularea unui program Java impreuna cu o lista de argumente pentru metoda main()

Se selecteaza din fereastră Projects proiectul care contine clasa ce trebuie rulata. Se face click dreapta, si se alege optiunea Properties.

Se alege categoria Run, si apoi numele clasei principale din proiect (clasa se poate selecta utilizand butonul Browse). Se introduc argumentele metodei principale in campul Arguments. Se apasa Finish.

Retele de calculatoare – Informatica anul 3 (2019-2020)

How do I run a Java program using a .bat file in Windows?

<https://www.quora.com/How-do-I-run-a-Java-program-using-a-bat-file-in-Windows>

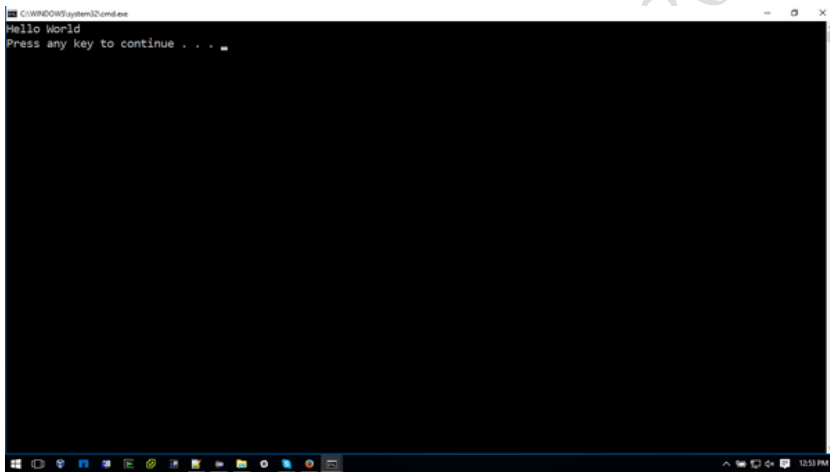
Step 1: Create java file what you want to execute and place that file in some location. For example I am creating Test.java in my desktop.

```
1. public class Test {
2.     public static void main(String[] args) {
3.         System.out.println("Hello World");
4.     }
5. }
```

Step 2: Create batch file and write following snippet and then save the file with some name. Here I am using Sample.bat and saved in my desktop.

```
1. @ECHO OFF
2. set CLASSPATH=.
3. javac Test.java
4. java Test
5. pause
```

Step 3 : Run the Sample.bat file then your java program automatically compiled and run. Batch file automatically opens the command prompt.



That's it. Happy coding..

Anexa 2

2.1. Checksums (Sursa: <http://ftp.uni-kl.de/pub/wireshark/docs/user-guide-us.pdf>)

Several network protocols use checksums to ensure data integrity.

Tip!

Applying checksums as described here is also known as **redundancy checking**.

What are checksums for?

Retele de calculatoare – Informatica anul 3 (2019-2020)

Checksums are used to ensure the integrity of data portions for data transmission or storage. A checksum is basically a calculated summary of such a data portion. Network data transmissions often produce errors, such as toggled, missing or duplicated bits. As a result, the data received might not be identical to the data transmitted, which is obviously a bad thing.

Because of these transmission errors, network protocols very often use checksums to detect such errors. The transmitter will calculate a checksum of the data and transmits the data together with the checksum. The receiver will calculate the checksum of the received data with the same algorithm as the transmitter. If the received and calculated checksums don't match a transmission error has occurred.

Some checksum algorithms are able to recover (simple) errors by calculating where the expected error must be and repairing it. If there are errors that cannot be recovered, the receiving side throws away the packet. Depending on the network protocol, this data loss is simply ignored or the sending side needs to detect this loss somehow and retransmits the required packet(s).

Using a checksum drastically reduces the number of undetected transmission errors. However, the usual checksum algorithms cannot guarantee an error detection of 100%, so a very small number of transmission errors may remain undetected.

There are several different kinds of checksum algorithms; an example of an often used checksum algorithm is CRC32. The checksum algorithm actually chosen for a specific network protocol will depend on the expected error rate of the network medium, the importance of error detection, the processor load to perform the calculation, the performance needed and many other things. Further information about checksums can be found at: <http://en.wikipedia.org/wiki/Checksum>.

2.2. Wireshark checksum validation

Wireshark will validate the checksums of several protocols, e.g.: IP, TCP, UDP, ... It will do the same calculation as a "normal receiver" would do, and shows the checksum fields in the packet details with a comment, e.g.: [correct], [invalid, must be 0x12345678] or alike.

Checksum validation can be switched off for various protocols in the Wireshark protocol preferences, e.g. to (very slightly) increase performance. If the checksum validation is enabled and it detected an invalid checksum, features like packet reassembling won't be processed. This is avoided as incorrect connection data could "confuse" the internal database.

7.8.2. Checksum offloading

The checksum calculation might be done by the network driver, protocol driver or even in hardware. For example: The Ethernet transmitting hardware calculates the Ethernet CRC32 checksum and the receiving hardware validates this checksum. If the received checksum is wrong Wireshark won't even see the packet, as the Ethernet hardware internally throws away the packet. Higher level checksums are "traditionally" calculated by the protocol implementation and the completed packet is then handed over to the hardware.

Recent network hardware can perform advanced features such as IP checksum calculation, also known as checksum offloading. The network driver won't calculate the checksum itself but will simply hand over an empty (zero or garbage filled) checksum field to the hardware.

Note!

Checksum offloading often causes confusion as the network packets to be transmitted are handed over to Wireshark before the checksums are actually calculated. Wireshark gets these "empty" checksums and displays them as invalid, even though the packets will contain valid checksums when they leave the network hardware later.

Checksum offloading can be confusing and having a lot of [invalid] messages on the screen can be quite annoying. As mentioned above, invalid checksums may lead to unreassembled packets, making the analysis of the packet data much harder. You can do two things to avoid this checksum offloading problem:

- Turn off the checksum offloading in the network driver, if this option is available.
- Turn off checksum validation of the specific protocol in the Wireshark preferences.

Anexa 3: Testarea unui program de tip *Checksum verifier*

Link util: https://emn178.github.io/online-tools/crc32_checksum.html

Anexa 4: The Programming Process

1. Identify the Problem - **What** Are You Trying To Do?
 - Requirements
 - Specification
2. Design a Solution - **How** Is It Going To Be Done?
3. Write the Program - **Teaching** the Computer
 - Code
 - Compile
 - Debug
4. Check the Solution - **Testing** it Understands You