Cerințe generale

- Se folosește API-ul standard pentru accesul la stiva de comunicație (Berkeley sockets)
- Nu se folosesc biblioteci și module suplimentare care interacționează direct sau indirect cu stiva de comunicație
- Aplicația trebuie să dispună de o interfață grafică care să permită demonstrarea tuturor modurilor de funcționare (interfețe disponibile în Python – Tkinter, WxPython, PyQT etc.)
- Aplicația trebuie să stocheze informații despre pachetele trimise/recepționate pentru a putea analiza și validarea secvențelor de comunicație (listă în GUI, fișier log etc.)

Server DHCP

- Să poată fi configurate/activate/dezactivate cel puțin 10 opțiuni DHCP
- Implementarea tuturor mesajelor DHCP (Discover, Offer ...) și demonstrarea folosirii lor conform logicii funcționare a protocolului
- Pool de adrese configurabil
- Lease time configurabil
- Atribuire statică de IP-uri pe bază de adrese MAC
- Vizualizare clienți cu resurse alocate

Client DHCP

- Să poată cere de la server cel puțin 10 opțiuni (activate/dezactivate individual la nivel de cerere)
- Implementarea tuturor mesajelor DHCP (Discover, Offer ...) și demonstrarea folosirii lor conform logicii funcționare a protocolului
- Istoric pentru adresele IP alocate anterior – se va încerca refolosirea lor
- Implementarea mecanismului de reînoire bazat pe lease time
- Tratarea situațiilor în care există mai mult de un server DHCP în rețea
- Implementarea mecanismului de eliberare a resurselor

Server MQTT

- Listă de topic-uri configurabilă (creare/stergere din GUI, fisier config.)
- Autentificare și restricționare acces abonare/actualizare pentru clienți
- Tratarea clienților care vor să se aboneze la un topic inexistent
- Vizualizare clienți conectați și abonați, deconectare forțată client
- Vizualizarea istoricului pentru ultimele 10 valori publicate/topic

Client MQTT

- Două view-uri: unul pentru publicare, unul pentru abonare
- Autentificare cu utilizator și parolă
- Listă de abonare configurabilă (creare/stergere din GUI, fisier config.)
- Publicare manuală (din GUI) sau automată (periodic, configurabil) pentru una din următoarele categorii de topic-uri:
    - Numere aleatoare (ex. API random.org)
    - Curs valutar (ex. API infovalutar.org)
    - Starea vremii (ex. API openweathermap.org)
    - Știri (ex. newsapi.org)

Server CoAP / Client CoAP

- Cele două echipe trebuie să colaboreze în vederea implementării unei soluții personalizate pentru:
    - Numere aleatoare (ex. API random.org)
    - Curs valutar (ex. API infovalutar.org)
    - Starea vremii (ex. API openweathermap.org)
    - Știri (ex. newsapi.org)
    - Alte categorii de informații disponibile online
- Server-ul va prelua informațiile de pe Internet și va pune la dispoziție cel puțin 5 tipuri/categorii de informații/resurse (ex. temperatura, presiunea atmosferică, umiditatea)
- Clientul va fi capabil să interogheze resursele convenite între echipe (disponibile pe server) și nu numai (se dorește tratarea cazurilor de eroare atât la server, cât și la client)
- Cele două aplicații trebuie să suporte un număr de coduri (vezi formatul mesajului) care să includă - codul 0.00 (mesaj fără conținut), metodele GET, POST (vezi Method Codes) și o metodă nouă propusă de echipe în contextul temei (fiți inventivi!), codurile de răspuns relevante pentru aplicație
- Aplicațiile trebuie să suporte mecanismul de comunicație cu confirmare, cât și mesaje fără confirmare

Transfer fișiere – controlul congestiei

- Două view-uri: unul pentru transmisie, unul pentru recepție
- Două sau mai multe perechi de instanțe ale aplicației (tx-rx) trebuie să poată să ruleze pe aceleași mașini sau în același LAN
- Comunicația va fi implementată prin datagrame UDP
- Formatul pachetelor va fi stabilit de echipe
- View-ul pentru recepție să permită „pierderea” voită a unor pachete pentru a putea demonstra funcționarea mecanismului
- Posibilitatea de configurare a mecanismului de control

Transfer fișiere – fereastră glisantă

- Două view-uri: unul pentru transmisie, unul pentru recepție
- Două sau mai multe perechi de instanțe ale aplicației (tx-rx) trebuie să poată să ruleze pe aceleași mașini sau în același LAN
- Comunicația va fi implementată prin datagrame UDP
- Formatul pachetelor va fi stabilit de echipe
- View-ul pentru recepție să permită „pierderea” voită a unor pachete pentru a putea demonstra funcționarea mecanismului
- Posibilitatea de configurare a parametrilor de funcționare (dimensiune fereastră, temporizări ...)
- Se va alege una dintre următoarele variante de implementare: Go back n, Selective repeat fără Nack, Selective repeat cu Nack, implementarea TCP pentru controlul fluxului

Descoperire topologie – RIPv2

- Funcționarea va fi demonstrată într-o topologie ce conține mai multe LAN-uri: se va folosi o topologie de mașini virtuale
- Mesajele și logica de comunicație vor fi implementate conform RFC-ului RIPv2

- Interfața va permite configurarea parametrilor de funcționare (temporizări, lungime maximă rute ...)
- Interfața va permite selectare informațiilor care sunt comunicate celorlalte noduri pentru a putea demonstra că aplicația detectează în timp real reconfigurarea topologiei