**T.C.**

**SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

AĞ PROGLAMLAMA

B211210095 – Emre ZENGİN

B211210557 – Merve KARTAL

B211210087-Merve ŞENTÜRK

|  |  |  |
| --- | --- | --- |
| Bölüm  Danışman | :  : | BİLGİSAYAR MÜHENDİSLİĞİ  Dr. Öğr. Üyesi MUSA BALTA |

**1. Uygulama Tanıtımı**

Bu proje, ICMP (Internet Control Message Protocol) protokolünü kullanarak basit bir istemci-sunucu iletişimi gerçekleştirmeyi amaçlar. İstemci (icmp\_client), belirli bir hedefe ICMP mesajları gönderirken, sunucu (icmp\_server) ise gelen mesajları analiz ederek uygun yanıtlar üretir. Proje, ICMP protokolünün ham paket düzeyinde nasıl çalıştığını anlamaya yönelik bir uygulama sağlar.

**1.1. Kullanılan Teknolojiler**

* **C Programlama Dili**
* **libpcap**: Ham paketleri yakalamak ve göndermek için kullanılır.
* **Raw Sockets / pcap\_inject**: Doğrudan Ethernet/IP/ICMP başlıkları elle inşa edilir.

**1.2. Proje Dosyaları**

1. **icmp\_client.c**
   * Basit bir ICMP istemcisi.
   * `send\_icmp\_packet` fonksiyonuyla “ICMP Echo Request” oluşturur ve **pcap\_inject** ile gönderir.
   * packet\_handler callback’i ile gelen **ICMP Echo Reply** (veya diğer ICMP tipleri) ekrana log basar.
2. **icmp\_server.c**
   * Gelen ICMP paketlerini yakalayan “sunucu” rolünde.
   * packet\_handler içinde, paketin **ICMP** mi olduğuna bakar; type=8 ise **ICMP Echo** olarak algılar ve type=0 (Echo Reply) yanıtı gönderir. Timestamp, hata mesajları vb. senaryolar da benzer şekilde işlenir.

**1.3. Derleme ve Çalıştırma**

1. **Derleme**:

gcc icmp\_client.c -o icmp\_client -lpcap

gcc icmp\_server.c -o icmp\_server -lpcap

1. **Sunucu**:

sudo ./icmp\_server enp0s3

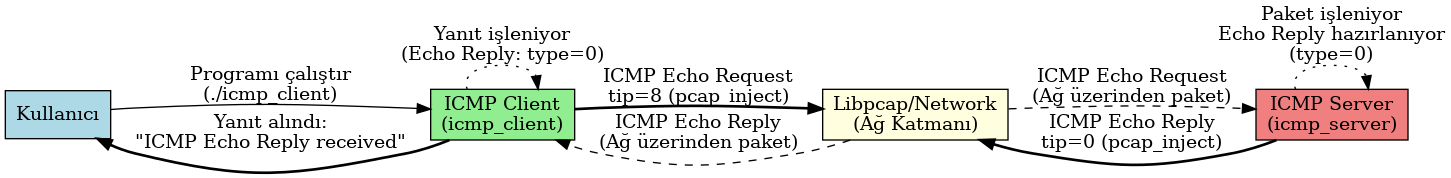
* + Sunucu, ICMP paketlerini **dinlemeye** başlar ve “ICMP Echo Request” gibi paketleri yakalayıp, **Echo Reply** yanıtı verir.

1. **İstemci**:

sudo ./icmp\_client enp0s3

* + “192.168.1.1” gibi sabit bir hedef IP’ye ICMP Echo Request gönderir (kod içinde değiştirilebilir).
  + Gelen yanıtları pcap\_loop içinde yakalayarak ekrana log basar.

1. **UML Sıralama Şeması (Sequence Diagram)**



/\*\*

\* Aktör ve katılımcıları tanımlıyoruz

\* - Kullanıcı, icmp\_client (Istemci)

\* - Libpcap/Network (ara katman)

\* - icmp\_server (Sunucu)

\*\*/

actor "Kullanıcı" as U

participant "ICMP Client\n(icmp\_client)" as C

participant "Libpcap\n(PCAP/Network)" as P

participant "ICMP Server\n(icmp\_server)" as S

'Uygulama akışı

U -> C: icmp\_client başlat

C -> P: \nICMP Echo Request (type=8)\npaketini pcap\_inject ile gönder

activate C

P -> S: \nICMP Echo Request\niçeren ham paket

activate S

S -> S: paket\_handler()\n(type=8 ise Echo Request)

S -> S: Echo Reply hazırlanır (type=0)

S -> P: \npcap\_inject ile Echo Reply (type=0) gönder

deactivate S

P -> C: ICMP Echo Reply (type=0)

C -> C: paket\_handler()\n(type=0 ise 'ICMP Echo Reply received...')

deactivate C

U: Konsolda \n"ICMP Echo Reply received..." \nmesajını görür

@enduml

**2.1. Diyagramdaki Adımların Açıklaması**

1. **Kullanıcı**, icmp\_client programını çalıştırır (sudo ./icmp\_client enp0s3).
2. **icmp\_client** kodu, send\_icmp\_packet fonksiyonuyla bir **ICMP Echo Request** (type=8) inşa eder ve **libpcap** üzerinden “ham paket” olarak ağa yollar (pcap\_inject).
3. Ağ katmanına düşen paket, **icmp\_server** uygulamasına (aynı makine veya ağ üzerinden) iletilir. packet\_handler fonksiyonunda “type=8” kontrol edilir ve “Echo Reply” oluşturulur (type=0).
4. Sunucu, ürettiği Echo Reply paketini yine pcap\_inject ile gönderir.
5. İstemcinin packet\_handler callback’i, gelen “**ICMP Echo Reply (type=0)**” paketini tespit eder ve “ICMP Echo Reply received…” mesajını terminale yazar.

**3. Sonuç**

* **icmp\_client** ve **icmp\_server** uygulamaları, **libpcap** kullanarak ICMP paketlerini ham düzeyde yönetmeyi göstermektedir.
* İstemci paket gönderir, sunucu da tipik “ping” davranışını (Echo Request -> Echo Reply) taklit eder. Ayrıca Timestamp, Destination Unreachable, Time Exceeded gibi mesajlar da benzer yaklaşım ile kodlanabilir.
* **UML sıralama şeması**, bu akışı görsel olarak özetlemektedir. Proje, “ham paket” düzeyinde programlamanın temel prensiplerini ve **ICMP** protokolünün işleyişini **uygulamalı** şekilde göstermektedir.

