In [2]:

```
import email
from email import policy
from email.parser import BytesParser
import re
from ipwhois import IPWhois

# Sample email header
email_header = """Delivered-To: recipient@example.com
Received: by 2002:a05:6510:1b10:b0:1f7:abc:2d70 with SMTP id v10csp4711552ejb;
        Thu, 2 Aug 2024 07:00:00 -0700 (PDT)
X-Received: by 2002:a0c:e818:0:b0:7f:feff:feff with SMTP id j18mr6423450eja.30.1690
        Thu, 02 Aug 2024 07:00:05 -0700 (PDT)
Return-Path: <sender@example.com>
Received: from mail.example.com (mail.example.com. [192.0.2.1])
        by mx.google.com with ESMTPS id s7si2345678ejb.324.2024.08.02.07.00.00
        for <recipient@example.com>
        (version=TLS1_3 cipher=TLS_AES_128_GCM_SHA256 bits=128/128);
        Thu, 02 Aug 2024 07:00:00 -0700 (PDT)
Received-SPF: pass (google.com: domain of sender@example.com designates 192.0.2.1 a
Authentication-Results: mx.google.com;
       dkim=pass header.i=@example.com header.s=selector1 header.b=d4t3Ef6T;
       spf=pass (google.com: domain of sender@example.com designates 192.0.2.1 as p
       dmarc=pass (p=NONE sp=NONE dis=NONE) header.from=example.com
From: Sender <sender@example.com>
To: Recipient <recipient@example.com>
Date: Thu, 2 Aug 2024 07:00:00 -0700
Subject: Sample Email
"""

def parse_email_header(header_str):
    msg = BytesParser(policy=policy.default).parsebytes(header_str.encode())
    header_info = {
        "From": msg["From"],
        "To": msg["To"],
        "Date": msg["Date"],
        "Subject": msg["Subject"],
        "Received": msg.get_all("Received", [])
    }
    return header_info

def extract_ip_from_received(received_lines):
    ip_pattern = re.compile(r'\[([\d\.]+)\]')
    ips = []
    for line in received_lines:
        match = ip_pattern.search(line)
        if match:
            ips.append(match.group(1))
    return ips

def get_ip_info(ip):
    try:
        ip_info = IPWhois(ip).lookup_rdap()
        network_info = ip_info['network']
        name = network_info.get('name', 'N/A')
        country = network_info.get('country', 'N/A')
        return name, country
    except Exception as e:
        return str(e), 'N/A'

def analyze_email_header(header_str):
    header_info = parse_email_header(header_str)
    print("Email Header Analysis:")
    print(f"From: {header_info['From']}")
    print(f"To: {header_info['To']}")
    print(f"Date: {header_info['Date']}")
    print(f"Subject: {header_info['Subject']}")

    ips = extract_ip_from_received(header_info["Received"])
    print("\nExtracted IP Addresses:")
    for ip in ips:
        print(f"IP: {ip}")
        name, country = get_ip_info(ip)
        print(f"  Organization: {name}")
        print(f"  Country: {country}")

if __name__ == "__main__":
    analyze_email_header(email_header)
```

```
Email Header Analysis:
From: Sender <sender@example.com>
To: Recipient <recipient@example.com>
Date: Fri, 02 Aug 2024 07:00:00 -0700
Subject: Sample Email

Extracted IP Addresses:
IP: 192.0.2.1
  Organization: IPv4 address 192.0.2.1 is already defined as TEST-NET-1 via RFC 573
7.
  Country: N/A
```

```
In [ ]:
```