What is DPI (Deep packet inspection), where it is used, what all mechanisms are available for the same?

Deep Packet Inspection (DPI) is a network traffic analysis technique that inspects the data part (and sometimes the header) of a packet as it passes through an inspection point. It is used in:

1. **Network Security**: Identifying and mitigating threats.
2. **Traffic Management**: Managing and prioritizing network traffic.
3. **Content Filtering**: Blocking access to specific content.
4. **Data Loss Prevention**: Monitoring and controlling sensitive data transfer.
5. **Law Enforcement**: Lawful interception of communications.

**DPI Mechanisms:**

1. **Pattern Matching**: Comparing packet contents against known patterns.
2. **Statistical Analysis**: Analysing traffic patterns for anomalies.
3. **Heuristic Analysis**: Using rules to identify suspicious packets.
4. **Behavioural Analysis**: Monitoring traffic behaviour over time.
5. **Deep Learning and AI**: Using machine learning for pattern recognition.
6. **Protocol Analysis**: Decoding specific protocols.
7. **Header Analysis**: Inspecting packet headers.

What is DPI analytics, how they are counted?

DPI analytics involves analyzing detailed packet data to extract insights on network traffic patterns, security threats, and user behavior. They are counted through metrics like packet and byte counts, flow and session analysis, protocol and application identification, and anomaly detection.

<https://netify.ai/> - Go through this - explain summary from this, how Netify is used by networking products

**Netify Summary**

Netify provides network intelligence and visibility using Deep Packet Inspection (DPI) and cloud-based analytics.

**Usage in Networking Products**

1. **DPI Engine**:
   * **Application Detection**
   * **Protocol Detection**
   * **QoS Management**
   * **Firewalls**: Enhanced security.
   * **SD-WAN**: Optimized routing.
   * **Device Discovery**: Inventories network devices.
   * **Digital Experience Monitoring**
   * **Network Alerts**
   * **Visibility and Tracking**
   * **Risk Management**
   * **Bandwidth Analysis**
   * **Forensics and Data Archiving**
   * **Compliance Integration**
   * **Geolocation Analysis**

**Benefits:**

* **Versatile**: Customizable insights.
* **Flexible Deployment**: Non-disruptive DPI agent.
* **Complete Visibility**: DPI combined with cloud analytics.
* **Data Forensics**: Deep network analysis.

In Kognitive edge system, what is probe settings, how the probe logic works.

**Kognitive Edge: Probe Settings and Logic :Purpose**: Monitor network performance (latency, packet loss, etc.).

**Configuration**: Define metrics to collect, collection frequency, and test destinations.

**Probe Logic:**

* **Data Collection**: Sends test packets to targets, measures response.
* **Analysis**: Evaluates performance, detects issues.
* **Reporting**: Results are visualized for decision-making.

How you will determine, this VLAN or network traffic is routed through the specific WAN or POP drop off tunnel?

By Checking the network configuration and use probes to monitor traffic paths and Using network tests (traceroute), review logs, and analyze traffic paths to identify routing through specific WANs or POP tunnels.

Write a program to check if specific website (eg. google.co.in) is reachable run this for 10 minutes and check every minute.

import requests

import time

def is\_website\_reachable(url):

    try:

        response = requests.get(url, timeout=10)

        return response.status\_code == 200

    except requests.RequestException as e:

        return False

def check\_website\_periodically(url, duration\_minutes, interval\_seconds):

    end\_time = time.time() + duration\_minutes \* 60

    while time.time() < end\_time:

        reachable = is\_website\_reachable(url)

        status = "reachable" if reachable else "not reachable"

        print(f"{url} is {status}")

        time.sleep(interval\_seconds)

if \_\_name\_\_ == "\_\_main\_\_":

    website\_url = "https://www.google.co.in"

    duration\_minutes = 10

    interval\_seconds = 60

    check\_website\_periodically(website\_url, duration\_minutes, interval\_seconds)

Write a program to get public IP programatically of your PC or device.

import requests

def get\_public\_ip():

try:

response = requests.get('https://api.ipify.org?format=json')

response.raise\_for\_status()

ip\_data = response.json()

return ip\_data['ip']

except requests.RequestException as e:

return f"Error fetching public IP: {e}"

if \_\_name\_\_ == "\_\_main\_\_":

public\_ip = get\_public\_ip()

print(f"Your public IP address is: {public\_ip}")