**Amar Soni**

**411702**

**CN Assignment 3:**

**Q1) What is ns2?**

**Ans)** NS2 stands for network simulator (Version 2). It is an event-driven simulation that has been proved to be useful for the dynamic nature of communication networks. It provides users with a way of specifying nature of such network protocols and simulating their corresponding behavior. It’s primarily used for analyzing the performance of congestion control algorithms on the basis of their various factors (packet delay, packet loss, error control).

Features:

* NS2 is an open source simulator.
* It is flexible and has modular nature.
* NS2 is UNIX based.
* Simulates both wired and wireless networks.
* It supports protocols- TCP, UDP, FTP, https and DSR.
* 2 Lang – C++ and Object-oriented Tool Command Language (OTCL).
* TCL – used as scripting language.
* Discrete event scheduler.

NS2 has gained constant popularity since it has been introduced in 1989. The developers and researchers of the community are constantly working to keep NS2 strong and versatile.

**Q3) Why IPv6 hasn’t replaced IPv4 yet?**

**Ans)** IPv6 uses 128 bit addressing ( 8 groups of 4 hexadecimal digits separated by semicolons) which can support approximately 340 trillion trillion(2^128) .

But the adoption of IPv6 has been delayed due many reasons.

1. Conversion requires replacing all the core routers with **expensive** hardware.(only when replacing the current equipment)
2. **NAT,** which is already present, allows extension of the present number of addresses, increasing the lifetime of IPv4 over IPv6.
3. **Backward compatibility** is not available in IPv6 protocol which disables the communication of the system undergone transition with those that still run on IPv4.

**Q2) Differences between NS2 and NS3.**

|  |  |  |
| --- | --- | --- |
| Properties | NS2 | NS3 |
| 1. Implementation | * Combination of C++ (core of simulation) and oTCL(for scripts describing network topology). | * C++ |
| 1. Scripting Language | * OTCL, which takes lesser compilation time than C++. But it has overheads with large simulations. | * C++ scripting and limited support of python. * Compilation time was not an issue due to modern hardware compatibilities. |
| 1. Memory Management | * Basic manual C++ memory management functions. | * All C++ memory management functions. * Automatic de-allocation of objects is supported using reference counting. (especially useful when dealing with packet objects) |
| 1. Debugging | * Complex due to bi-language system. | * Easy, single language system is more robust in longer run. |
| 1. Packets | * 2 distinct regions – one for headers and the other payload data. | * Consists of single buffer of bytes and optionally, collection of small tags with meta-data. |