

**Year: M. Tech. I (Semester – I)**

**Subject Name:** Intelligent Networks

**Subject Code:** MTCO14105

**Type of course:** Professional Elective - I

**Prerequisite:** Basics knowledge of Computer Network

**List of Courses where this course will be prerequisite: --**

**Rationale:** The Internet's routing and switching speeds have been on super-linear increase to cope with the exponentially growing traffic demand. To respond to the increasing complexity in the core, the networking community has moved to techniques with cheaper management costs via more virtualization and centralization of the protocol functions. This trend came with separation of control and data plane networking functions, a paradigm named as 'software-defined networking' (SDN). This course covers fundamentals of SDN, its basic design trade-offs, potential applications, and relations with network function virtualization (NFV). It explores emerging applications of SDN in network management, traffic engineering, and software-defined exchanges (SDXes). The course offers hands-on exercises of SDN protocols such as OpenFlow and ONOS.

**Teaching and Examination Scheme:**

| Teaching Scheme |   |   |   | Theory Marks |     |     | Practical Marks |     | Total |
|-----------------|---|---|---|--------------|-----|-----|-----------------|-----|-------|
| L               | T | P | C | TEE          | CA1 | CA2 | TEP             | CA3 |       |
| 3               | 0 | 2 | 4 | 60           | 25  | 15  | 30              | 20  | 150   |

CA1: Continuous Assessment (assignments/projects/open book tests/closed book tests CA2: Sincerity in attending classes/class tests/ timely submissions of assignments/self-learning attitude/solving advanced problems TEE: Term End Examination TEP: Term End Practical Exam (Performance and viva on practical skills learned in course) CA3: Regular submission of Lab work/Quality of work submitted/Active participation in lab sessions/viva on practical skills learned in course

**Contents:**

| Sr.No. | Contents   | Total Hrs |
|--------|--|-----------|
| 1      | INTRODUCTION TO SDN: SDN Origins and Evolution, Basic Packet Switching Terminology, The Modern Data Centre, Traditional Switch Architecture, Autonomous and Dynamic Forwarding Tables, Can We Increase the Packet Forwarding IQ? Why SDN? The Genesis of SDN | 10        |



|   |   |    |
|---|---|----|
| 2 | HOW SDN WORKS? : Fundamental Characteristics of SDN, SDN Architecture, SDN Operation, SDN Devices, SDN Controller, SDN Applications, Alternate SDN Methods                                  | 08 |
| 3 | OPENFLOW : OpenFlow Overview, OpenFlow 1.0 and OpenFlow Basics, Version of OpenFlow, Switch-Controller Interaction, Flow Table, Packet Matching, Actions and Packet Forwarding, Limitations | 05 |
| 4 | NETWORK FUNCTION VIRTUALIZATION: Definition of NFV, What Can We Virtualize? Standards, OPNFV, Leading NFV Vendors, SDN vs NFV, In-Line Network Functions, NFV Orchestration                 | 06 |
| 5 | SDN APPLICATIONS AND USE CASES : SDN in the Data Centre, SDN in other Environments, SDN Applications  | 07 |
| 6 | SECURITY IN SDN : Security requirements, Threats to SDN, SDN security, NFV Security and its techniques  | 03 |
| 7 | SDN EMULATOR AND CONTROLLERS : Mininet, POX, NOX, RYU, Flood Light, Open Daylight   | 04 |
| 8 | SDN'S FUTURE AND PERSPECTIVES : Current State of Affairs, SD-WAN, Potential Novel Applications of Open SDN  | 02 |

**Reference Books:**

| Sr.No. | Title of book /article                              | Author(s)                      | Publisher and details like ISBN | Year of publication | Publication Edition |
|--------|---|--------------------------------|---------------------------------|---------------------|---------------------|
| 1      | Software Defined Networks: A Comprehensive Approach | Paul Goransson and Chuck Black | Morgan Kaufmann                 | 2014                | First               |
| 2      | SDN - Software Defined Networks                     | Thomas D. Nadeau & Ken Gray    | O'Reilly                        | 2013                | Second              |
| 3      | Software Defined Networking with OpenFlow           | Siamak Azodolmolk              | Packt Publishing                | 2013                | First               |



|   |                                 |                        |          |      |       |
|---|---------------------------------|------------------------|----------|------|-------|
| 4 | Network Function Virtualization | Ken Gray Thomas Nadeau | Elsevier | 2016 | First |
|---|---------------------------------|------------------------|----------|------|-------|

**Course Outcomes (CO):**

| Sr. No. | CO statement   | Marks % weightage |
|---------|--|-------------------|
| 1       | Differentiate between traditional networks and software defined networks   | 10%               |
| 2       | Interpret the characteristics of SDN controllers and their implications to learn the board aspects of security, overlay and network model. | 35%               |
| 3       | Elaborate network function virtualization and network virtualization   | 15%               |
| 4       | Learn how to use software programs to perform varying and complex networking tasks   | 15%               |
| 5       | Expand upon the knowledge learned and apply it to solve real world problems.   | 25%               |

**List of Open learning website:**

1. <https://www.coursera.org/learn/sdn>
2. <https://opennetworking.org/sdn-definition/>
3. [https://github.com/p4lang/tutorials/blob/master/exercises/basic\\_tunnel](https://github.com/p4lang/tutorials/blob/master/exercises/basic_tunnel)
4. [https://nptel.ac.in/courses/pdf\\_download.php?&subjectId=106105166&lectid=33&lang=English](https://nptel.ac.in/courses/pdf_download.php?&subjectId=106105166&lectid=33&lang=English)
5. [www.mininet.org](http://www.mininet.org)
6. [www.opendaylight.org](http://www.opendaylight.org)

**List of Open-Source Software: --**

1. Mininet
2. Open Daylight Controller





**List of Experiments:**

| Sr. No | Practical  |
|--------|--|
| 1      | Introduction to Mininet  |
| 2      | Write a python script to create the topology.  |
| 3      | Create Linear, Tree and Fat tree topology and measure the performance.   |
| 4      | Implement Custom Topologies in POX, ODL, Floodlight 3 Controllers  |
| 5      | Write a python code for topology discovery and shortest path calculation.  |
| 6      | Execute the python script of P4 Language tutorials and analyse. <ul style="list-style-type: none"><li>• Basic Forwarding</li><li>• Basic Tunnelling</li><li>• P4 Runtime</li></ul> |
| 7      | Study the parametric comparison of different controllers.  |
| 8      | Case study of B4 -Experience with a Globally-Deployed Software Defined WAN   |

