(CT(DE)-21014) Computational Geometry Laboratory

Teaching Scheme
Laboratory: 2 Hrs / Week

Examination Scheme
Continuous evaluation: 50 Marks
End Semester Exam: 50 Marks

Course Outcomes

Students will be able to:

1. Implement basic and advanced algorithms in the area of computational geometry
2. Demonstrate ability to write code taking care of precision of numbers, boundary conditions, degeneracy etc
3. Verify the implementations for correctness and efficiency by using suitable test data
4. Decide the best possible solution for a given problem under the given constraints
5. Apply the algorithms/concepts studied in the theory course to real life problems

Suggested List of Assignments

1. Area of a polygon
2. Finding closest pair of points
3. Triangulating a polygon
4. Convex hull in two dimensions
5. Convex hull in three dimensions
6. Delaunay Triangulation
7. Segment/ray-segment intersection
8. Segment/ray-triangle intersection
9. Point in polygon
10. Point in polyhedron
11. Intersecting convex polygons
12. Minkowski convolution with a convex polygon
13. Multilink robot arm reachability

This is a suggested list. The instructor is expected to continuously update it.

(CT(DE)-21015) Recent Trends in Computer Networks

Teaching Scheme:
Lectures: 3 Hrs/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

Course Outcomes

Students will be able to

1. Explain issues in the design of network topologies and design network systems
2. Analyse different possible solutions for communications at each network layer
3. Simulate working of wired and wireless networks to Explain networking concepts
4. Develop solutions by applying knowledge of mathematics, probability, and statistics to network design problems
5. Explain and Compare various storage and networking technologies
6. Explain the recent advancements in the networking technologies

Course Contents

Internetworking: Routing algorithms evaluations, TCP variants, Quality of Service, Active Queue Management, High-Speed Networks, Performance Modelling and Estimation

[7 Hrs]

IPv6: IPv4 deficiencies, patching work done with IPv4, IPv6 addressing, multicast, Anycast, ICMPv6, Neighbour Discovery, Routing, header compression

[5 Hrs]

Software-Defined Networking and OpenFlow: Centralized and Distributed Control and Data Planes, SDN Controllers, Data Center Concepts, Network Function Virtualization, Mininet, Programming SDNs, Openflow Switch, Wire Protocol, Openstack Neutron plug-in

[6 Hrs]

Ad Hoc Wireless Networks: MAC Protocols for Ad Hoc Wireless Networks, Routing Protocols for Ad Hoc Wireless Networks, Multicast routing in Ad Hoc Wireless Networks, Transport Layer and Security Protocols for Ad Hoc Wireless Networks, Quality of Service in Ad Hoc Wireless Networks.

[8 Hrs]

Storage and Networking: Storage and Networking Concepts, Fiber Channel Internals, Fiber Channel SAN Topologies, Fiber Channel Products, IP SAN Technology, IP SAN Products, Management of SANs, SAN Issues

[7 Hrs]

Advancements in CN: Fundamentals of MPLS (RFC 3031) and GMPLS, SNMP, 5G architecture, Named Data Networks (NDN), Content-Centric Networking (CCN), IoT

[8 Hrs]

Text Books

- Thomas D Nadeau and Ken Grey, Software Defined Networking, O'Reilly, 2013
- https://tools.ietf.org/html/rfc8200
- https://tools.ietf.org/html/rfc3031
- Mani Subramanian, Timothy A. Gonsalves, N. Usha Rani; Network Management: Principles and Practice; Pearson Education India, 2010 References
- William Stallings, High-Speed Networks and Internets, Pearson Education, 2nd Edition, 2002.
- C. Siva Ram Murthy, B.S. Manoj, Ad Hoc Wireless Networks: Architectures and Protocols, Prentice Hall, 2004
- Muthukumaran B, Introduction to High-Performance Networks, Tata Mc Graw Hill, 2008
- Tom Clark, Designing Storage Area Networks, A Practical Reference for Implementing Fibre Channel and IP SANs, Addison-Wesley Professional, 2nd Edition, 2003.
- For 5Garchitecture: https://www.3gpp.org/
- For NDN: https://named-data.net/
- Recent papers and RFCs on advancement in Computer Networks

(CT(DE)-21016) Recent Trends in Computer Networks Laboratory

Teaching Scheme:
Laboratory: 2 Hrs /week

Examination Scheme:
Term Work: 50 marks
Oral Examination: 50 marks

Course Outcomes

1. Relate theory with practice by performing programming assignments
2. Get proficiency in designing network topology solutions
3. Get proficiency in a variety of tools and environments like ns-3, NeST
4. Analyse various networking algorithms and implementation to solve a networking problem
5. Work on the individual assignments to get the
6. Imbibe good programming practices Suggested

Suggested List of Assignments

1. Create a simple dumbbell routing topology setup over the Linux kernel
2. Evaluate the performance of intra-domain routing algorithms using ns-3 or Linux kernel
3. Evaluate the performance of different TCP variants over ns-3 or ip netns
4. Perform a Wireshark packet sniffing experiment.
5. Do the network analysis using NMAP - the Network Mapper.
6. Using Mininet tool for the SDN topology
7. Implement a web proxy that passes requests and data between multiple web clients and web servers.
8. Create a topology using ip netns with the help of IPv6 addresses and perform the operations related to different next headers
9. Evaluate the performance of different Ad Hoc routing protocols e.g., AODV, DSDV, etc,.
10. Setup the NDN testbed (https://named-data.net/)

This list is a guideline. The instructor is expected to improve it continuously.

(CT-21010) Operating Systems

Teaching Scheme:
Lectures: 3 Hrs/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

Course Outcomes

Students will be able to:

1. Write programs to manipulate processes, files, and hardware resources using appropriate system calls.
2. Illustrate the design issues, solutions and complexity of operating system by compiling, modifying an OS kernel, tracing the sequence of activities on processor, data structures of a file system, race conditions, locking mechanisms and storage techniques.
3. Correlate the computer architecture features with operating system design issues.
4. Make design choices for an operating systems with given constraints