Course Code	PN
Department	CSE
Course Name	Programmable Networking
Credits	4
Course Offered to	UG/PG
Whether the course is to be counted towards M.Tech specialization. If yes, please select the specialization towards which it is to be counted	CSE+Mobile Computing
If the course is to be counted towards other B.Tech programs(For Ex if a course with CSE no. satisfies the requirement of 32 credits of B. Tech ECE program that students have to do in last 4 semesters, then the drop down answer should be ECE)	CSAI, CSD, CSAM
Course Description	There has been a massive revolution in networking research in the last decade, leading to newer networking infrastructure designs that are highly performant, scalable, and flexible. This evolution has opened up exciting opportunities for both networking and non-networking application domains. This course will discuss networking advancements such as Software-defined Networks (SDN), programmable network hardware such as smartNICs and programmable switches, kernel bypass networking, RDMA, and high-speed host network stacks. We will discuss a few seminal research papers to understand each networking design advancement.

Pre-requisites		
Pre-requisite (Mandatory)	Pre-requisite (Desirable)	Pre-requisite(other)
CSE102, CSE231, CSE232	Fluency in working with scripts, C/C++ and python programming. Readiness to learn a new programming language	CSE222, CSE319/CSE519

*Please insert more rows if required

Post Conditions*(For suggestions on verbs please refer the second sheet)				
CO1	**CO2**	**CO3**	**CO4**	**CO5**
Students will be able to recall the basics of network layer. They will be able to understand the design principles for a high-performant, scalable, and flexible networking infrastructure.	Students will be able to anlayze the reasons for the performance overheads in traditional network packet processing. They will be able to identify and exploit the benefits of kernel-bypass, in-network, and in-kernel packet processing techniques.	Students will understand the research ideas and the challenges addressed to build a high-speed, flexible networking infrastructure. They will be able to analyze research findings to gauge the impact.	Students will be able to construct novel ideas for networking/non-networking applications. They will be able to design, and implement these applications for SDN/programmable data plane infrastructure, and evaluate the performance improvements of in-network packet processing over traditional packet processing.	

Weekly Lecture Plan

Week Number	**Lecture Topic**	**COs Met**	**Assignment/Labs/Tutorial**	
1	**Introduction:** networking infrastructure evolution, network layer recap, working of a router **Software-defined networking I:** motivation and concept, Openflow SDN protocol	CO1, CO2		
2	**Software-defined networking II:** Centralized SDN controllers, Distributed SDN controllers, Google's software-defined WAN (B4), SDN limitations	CO1, CO3, CO4	**Assignment 1:** **Goal:** Implement a MAC-based learning and forwarding switch **Components:** mininet, OpenVSwitch (OVS), POX SDN controller **Intent:** This assignment will help understand SDN concepts. The mininet infrastructure set up will help set up the subset of infrastructure for assignment 2	Note: The course project requires infrastructure setup and aquaintance to new tools. These preliminary requirements are satisfied via the two assignments. Students will build their project over the infrastructure set up during the assignments. Therefore, the students technically start their project from week 2.
3	**Programmable data planes:** programmable data plane concepts and architecture **Hardware programmable data planes:** programmable switching device hardware, reconfigurable match-action tables	CO1, CO2, CO3		
4	**Language for programmable hardware I:** architecture model, P4 data types, expressions, statements, packet parser, control blocks, packet deparser	CO4	**Assignment 2:** **Goal:** Implement a stateless firewall using P4 **Components:** mininet, bmv2 switch, thrift API **Intent:** This is a warm-up exercise that helps building the basic infrastructure for the course project	
5	**Language for programmable hardware II:** control plane API, P4 stateful elements (tables, registers, counters, and meters), P4Runtime	CO4	**Course project:** Topic selection deadline	
6	**Introduction to programmable hardware infrastructure:** FPGA based SmartNICs (e.g., Xilinx NetFPGA-SUME), SoC based SmartNICs (e.g., Netronome), ASIC based programmable switch (e.g., Intel Tofino)	CO2, CO4	**Course project:** Problem statement and preliminary design document submission deadline	
7, 8	**Offload network functionality to programmable hardware:** network telemetry, heavy-hitter detection for traffic engineering, ML model for traffic classification (any two)	CO3, CO4		
9	**Offload application to programmable hardware I:** caching, application level load balancing (any one)	CO3, CO4	**Course project:** Phase 1 presentations **Intent:** track project progress and resolve roadblocks (if any)	
10	**Offload application to programmable hardware II:** fault tolerance, consensus management (any one)	CO3, CO4		
11	**Software data planes I:** performance overheads of a standard kernel-based network stack, motivation, kernel bypass techniques (DPDK), host network stacks (mTCP or IX)	CO2, CO3, CO4		
12	**Software data planes II:** in-kernel packet processing (eBPF, XDP)	CO2, CO3, CO4		
13	**Introduction to current trends:** The 5G mobile network and edge computing, offload network stack to NIC hardware using Remote Direct Memory Access (RDMA)	CO2, CO3, CO4	**Course project:** Final presentations and report submission	

*Please insert more rows if required

Weekly Lab Plan

Week Number	**Laboratory Exercise**	**COs Met**	**Platform (Hardware/Software)**
Course does not have a lab component.			

*Please insert more rows if required

Assessment Plan	
Type of Evaluation	% Contribution in Grade
Project	**30**
Assignment	**20**
Research paper review	**15**
Mid-sem	**15**
End-sem	**20**
*Please insert more row for other type of Evaluation	

Resource Material	
Type	**Title**
Books	There are no official textbooks for this course. Students are expected to read research papers. The following text books provide the fundamentals. (1) Computer Networking: A Top-Down Approach (7th edition), by James F. Kurose and Keith W. Ross. (2) Computer Networks: A Systems Approach, by Larry Peterson and Bruce Davie. Online link: https://book.systemsapproach.org/ (3) 5G Mobile Networks: A Systems Approach, by Larry Peterson and Oguz Sunay. Online link: https://5g.systemsapproach.org/
Web references	The following web references will help the students in programming assignments and projects. (1) Mininet & Openflow (VM+ tutorial) https://github.com/mininet/openflow-tutorial/wiki/Installing-Required-Software (2) P4 infrastructure setup tutorial https://github.com/p4lang/tutorials/tree/sigcomm19 (3) P4-16 specifications https://p4.org/p4-spec/docs/P4-16-v1.2.2.html (4) P4Runtime specifications https://p4.org/p4-spec/p4runtime/v1.3.0/P4Runtime-Spec.html (5) Sample P4 programs for bmv2 platform https://github.com/p4lang/tutorials (6) Sample P4 programs for Netronome smartNIC https://github.com/open-nfpsw/
Research papers	The following research papers will help the students learn the course topics. (1) Albert Greenberg, et al. A clean slate 4D approach to network control and management. ACM SIGCOMM Computer Communication Review 35.5 (2005): 41-54. (2) Nick McKeown, et al. OpenFlow: enabling innovation in campus networks. ACM SIGCOMM computer communication review 38.2 (2008): 69-74. (3) Natasha Gude, et al. NOX: towards an operating system for networks. ACM SIGCOMM computer communication review 38.3 (2008): 105-110. (4) Sushant Jain, et al. B4: experience with a globally-deployed software defined WAN. In Proceedings of the ACM SIGCOMM 2013. (5) Pat Bosshart, et al. Forwarding metamorphosis: Fast programmable match-action processing in hardware for SDN. ACM SIGCOMM Computer Communication Review 43.4 (2013): 99-110. (6) Pat Bosshart, et al. P4: programming protocol-independent packet processors. SIGCOMM Computer Communication Rev. 44, 3 (July 2014), 87–95. (7) Dan RK Ports, and Jacob Nelson. When should the network be the computer? Proceedings of the Workshop on Hot Topics in Operating Systems. 2019. (8) James McCauley, et al. Thoughts on load distribution and the role of programmable switches. ACM SIGCOMM Computer Communication Review 49.1 (2019): 18-23. (9) Ran Ben Basat, et al. Pint: Probabilistic in-band network telemetry. Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication. 2020. (10) Vibhaalakshmi Sivaraman, et al. Heavy-hitter detection entirely in the data plane. Proceedings of the Symposium on SDN Research. 2017. (11) Zhaoqi Xiong, and Noa Zilberman. Do switches dream of machine learning? Toward in-network classification. Proceedings of the 18th ACM workshop on hot topics in networks. 2019. (12) Xin Jin, et al. Netcache: Balancing key-value stores with fast in-network caching. Proceedings of the 26th Symposium on Operating Systems Principles. 2017. (13) Rui Miao, et al. Silkroad: Making stateful layer-4 load balancing fast and cheap using switching ASICs. Proceedings of the Conference of the ACM Special Interest Group on Data Communication. 2017. (14) Marcos A. M. Vieira, et al. Fast Packet Processing with eBPF and XDP: Concepts, Code, Challenges, and Applications. ACM Computing Surveys. 53, 1, Article 16 (May 2020), 36 pages. (15) Aleksandar Dragojević, et al. FaRM: Fast remote memory. 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14). (16) YoungGyoun Moon, et al. AccelTCP: accelerating network applications with stateful TCP offloading. In Proceedings of the 17th Usenix Conference on Networked Systems Design and Implementation (NSDI 20). (17) EunYoung Jeong, et al. mTCP: a highly scalable user-level TCP stack for multicore systems. 11th {USENIX} Symposium on Networked Systems Design and Implementation (NSDI 14). (18) Adam Belay, et al. IX: A Protected Dataplane Operating System for High Throughput and Low Latency. 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14).