National University of Singapore
School of Computing
CS5229: Advanced Computer Networks
Semester I, 2021/2022

General Information

Correct as at: 20th August 2021

Welcome to CS5229. This document provides an overview of the course. Students should henceforth refer to the course module on Coursemology (`http://coursemology.org`). The lesson plan, announcements, assignment due dates, and more detailed information for the module will be disseminated on Coursemology.

Aims & Objectives

This course covers advanced fundamental principles of computer networks and techniques for networking. The goal of this course is to teach these fundamentals/techniques that will remain important and relevant regardless of the hot topics in networking. There will be some differences in the topics covered from year to year, but they would include advanced network architecture and design principles, protocol mechanisms, implementation principles and software engineering practices, network algorithmic, network simulation techniques and tools, performance analysis and measurement, and protocol specification/verification techniques.

We will discuss topics that are generally beyond what is covered in a basic undergraduate networking class, up until state-of-the-art, with a focus on the principles and abstractions. But clearly, we cannot teach everything so this module will be a broad survey of the field. In addition to knowledge, we also want students to learn practical skills that are broadly applicable beyond networking. To this end, students will be provided with programming exercises and a final project that would help them pick up practical skills in terms of running experiments and analyzing data.

Course Materials

There is currently no required textbook for the course. The material for this course will be taken from published research papers.

Students are expected to already have some background on undergraduate networking. For reference, they might consider checking out *Computer Networks: A Systems Approach* by Profs Larry Peterson and Bruce Davie available free online: `https://book.systemsapproach.org`.

Course materials can be found on Coursemology. On Coursemology, you will also find general announcements, a detailed syllabus and lecture schedule, assignments, and other useful documents. We will be using Microsoft Teams instead of a forum to address questions from students.

Teaching Staff

Lecturer:	Dr Ben Leong COM2 #03-20 Tel: 6516 4240	`benleong@comp.nus.edu.sg`
Teaching Assistants:	1. Raj Joshi 2. Ayush Mishra	`rajjoshi@comp.nus.edu.sg` `ayush@comp.nus.edu.sg`

Teaching Mode

Lectures:	2 hours per week (Zoom)

Given the safe distancing measures, lectures will be conducted online using Zoom this semester. Recordings will be made available through Coursemology but students are encouraged to attend the actual lecture "live". Before each lecture, an automatically-graded online quiz will be released on the Coursemology platform to encourage students to read the readings for the week.

You do not have to read every research paper in full detail. Doing the first two "passes" as described by Keshav [1] should be sufficient for the online quiz. After each lecture, students will also have to complete a simple online exercise called "Lecture training" on Coursemology in order to reinforce the concepts being taught during the lecture. Lecture trainings will consist of multiple-choice and short-answer questions.

Scheduled Examination

Exam	Date/Time	Venue
Course Assessment	12 Nov (Friday) 6.30 pm - 8.30 pm	TBA

Please note down the time in your calendar now!

The course assessment to be held in Week 13 is currently planned as an *open-book* written exam. The exact arrangements will however be determined by the prevailing safe-distancing policy and the availability of exam halls nearer the date. Students are allowed to bring along *any* written reference materials.

Grading Scheme

Item	Weightage
Online quizzes/exercises (CA)	15%
Programming Assignments	25% (2% + 8% + 15%)
In-class assessment	30%
Term Project	30%
Total	100%

We will be using the gamification system in Coursemology for the Continuous Assessment (CA) component. The goal of the CA is not to assess the students' mastery of the material. Instead it is used to encourage students to complete the readings and turn up for class. Experience points (EXP) will be awarded for lesson preparation, online

quizzes and for turning up during lectures (attendance). With enough EXP, students will increase in level. The final level capped at 15 will be the CA grade for CS5229.

For the term project, students would be expected to build an in-depth understanding of a research paper and reproduce the main results reported in the paper. This provides students with the opportunity to learn one topic in state-of-the-art research (of their choice) in detail and pick up valuable skills in running experiments, processing data and plotting graphs. Ideally, this process will also allow them to develop intuitions about the problem(s) studied and potentially encourage some to do follow-on research.

Please do not worry about grades in this course. Please focus on learning.

Lesson Plan

The following is the tentative lesson plan and readings for the semester. Given that we are revamping CS5229 this semester, this list is subject to change and students should refer to Coursemology each week for the authoritative list of readings.

Week	Dates	Topic(s)	Readings	Optional
1	13 Aug 2021	Introduction to CS5229 & Internet Architecture	[2, 3]	[4, 5, 6]
2	20 Aug 2021 1	Window-based End-to-End Congestion Control	[7, 8]	[9, 10]
3	27 Aug 2021	Active Queue Management & Buffer Sizing	[11, 12, 13]	[14]
4	3 Sep 2021	Rate-based End-to-End Congestion Control	[15, 16]	[17, 18, 19]
5	10 Sep 2021	Network-assisted Congestion Control	[20, 21, 22]	
6	17 Sep 2021	Data Centre Networking	[23, 24, 25, 26]	[27, 28]
-	24 Sep 2021	**– RECESS WEEK –**		
7	1 Oct 2021	Software Defined Networks	[29, 30]	[31, 32, 33, 34]
8	8 Oct 2021	Wireless Ad Hoc Networks	[35, 36, 37]	
9	15 Oct 2021	Wireless Mesh Networking	[38, 39, 40]	
10	22 Oct 2021	Peer-to-Peer & Distributed Systems	[41, 42, 43]	
11	29 Oct 2021	Network/Distributed Co-Design	[44, 45]	
12	5 Nov 2021	The Last Lecture		
13	12 Nov 2021	In-class Assessment		
-	19 Nov 2021	**– READING WEEK –**		

References

[1] Srinivasan Keshav. How to read a paper. *SIGCOMM Computer Communication Review*, 37(3):83–84, 2007.

[2] Jerome H Saltzer, David P Reed, and David D Clark. End-to-end arguments in system design. *ACM Transactions on Computer Systems (TOCS)*, 2(4):277–288, 1984.

[3] David Clark. The design philosophy of the DARPA Internet protocols. In *Symposium proceedings on Communications architectures and protocols*, pages 106–114, 1988.

[4] David D Clark, John Wroclawski, Karen R Sollins, and Robert Braden. Tussle in cyberspace: defining tomorrow's internet. In *Proceedings of the 2002 conference on*

Applications, technologies, architectures, and protocols for computer communications, pages 347–356, 2002.

[5] Mark Handley. Why the internet only just works. *BT Technology Journal*, 24(3):119–129, 2006.

[6] Barry M Leiner, Vinton G Cerf, David D Clark, Robert E Kahn, Leonard Kleinrock, Daniel C Lynch, Jon Postel, Larry G Roberts, and Stephen Wolff. A brief history of the internet. *ACM SIGCOMM Computer Communication Review*, 39(5):22–31, 2009.

[7] Van Jacobson. Congestion avoidance and control. *ACM SIGCOMM computer communication review*, 18(4):314–329, 1988.

[8] Ayush Mishra, Xiangpeng Sun, Atishya Jain, Sameer Pande, Raj Joshi, and Ben Leong. The great internet TCP congestion control census. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(3):1–24, 2019.

[9] Dah-Ming Chiu and Raj Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN systems*, 17(1):1–14, 1989.

[10] Sangtae Ha, Injong Rhee, and Lisong Xu. Cubic: a new TCP-friendly high-speed TCP variant. *ACM SIGOPS operating systems review*, 42(5):64–74, 2008.

[11] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on networking*, 1(4):397–413, 1993.

[12] Kathleen Nichols and Van Jacobson. Controlling queue delay: A modern aqm is just one piece of the solution to bufferbloat. *Queue*, 10(5):20–34, 2012.

[13] Nick McKeown, Guido Appenzeller, and Isaac Keslassy. Sizing router buffers (redux). *ACM SIGCOMM Computer Communication Review*, 49(5):69–74, 2019.

[14] Guido Appenzeller, Isaac Keslassy, and Nick McKeown. Sizing router buffers. *ACM SIGCOMM Computer Communication Review*, 34(4):281–292, 2004.

[15] Lawrence S Brakmo, Sean W O'Malley, and Larry L Peterson. TCP vegas: New techniques for congestion detection and avoidance. In *Proceedings of the conference on Communications architectures, protocols and applications*, pages 24–35, 1994.

[16] Neal Cardwell, Yuchung Cheng, C Stephen Gunn, Soheil Hassas Yeganeh, and Van Jacobson. Bbr: Congestion-based congestion control: Measuring bottleneck bandwidth and round-trip propagation time. *Queue*, 14(5):20–53, 2016.

[17] Sally Floyd, Mark Handley, Jitendra Padhye, and Jörg Widmer. Equation-based congestion control for unicast applications. *ACM SIGCOMM Computer Communication Review*, 30(4):43–56, 2000.

[18] Wai Kay Leong, Zixiao Wang, and Ben Leong. TCP congestion control beyond bandwidth-delay product for mobile cellular networks. In *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*, pages 167–179, 2017.

[19] Venkat Arun and Hari Balakrishnan. Copa: Practical delay-based congestion control for the internet. In *15th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 18)*, pages 329–342, 2018.

[20] Dina Katabi, Mark Handley, and Charlie Rohrs. Congestion control for high bandwidth-delay product networks. In *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 89–102, 2002.

[21] Nandita Dukkipati, Masayoshi Kobayashi, Rui Zhang-Shen, and Nick McKeown. Processor sharing flows in the internet. In *International Workshop on Quality of Service*, pages 271–285, 2005.

[22] K Ramakrishnan, Sally Floyd, and D Black. RFC3168: The addition of explicit congestion notification (ECN) to IP, 2001.

[23] Albert Greenberg, James R Hamilton, Navendu Jain, Srikanth Kandula, Changhoon Kim, Parantap Lahiri, David A Maltz, Parveen Patel, and Sudipta Sengupta. Vl2: A scalable and flexible data center network. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, pages 51–62, 2009.

[24] Arjun Singh, Joon Ong, Amit Agarwal, Glen Anderson, Ashby Armistead, Roy Bannon, Seb Boving, Gaurav Desai, Bob Felderman, Paulie Germano, et al. Jupiter rising: A decade of clos topologies and centralized control in google's datacenter network. *ACM SIGCOMM computer communication review*, 45(4):183–197, 2015.

[25] Mohammad Alizadeh, Albert Greenberg, David A Maltz, Jitendra Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta, and Murari Sridharan. Data center TCP (DCTCP). In *Proceedings of the ACM SIGCOMM 2010 Conference*, pages 63–74, 2010.

[26] Radhika Mittal, Vinh The Lam, Nandita Dukkipati, Emily Blem, Hassan Wassel, Monia Ghobadi, Amin Vahdat, Yaogong Wang, David Wetherall, and David Zats. Timely: Rtt-based congestion control for the datacenter. *ACM SIGCOMM Computer Communication Review*, 45(4):537–550, 2015.

[27] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, et al. B4: Experience with a globally-deployed software defined wan. *ACM SIGCOMM Computer Communication Review*, 43(4):3–14, 2013.

[28] Chi-Yao Hong, Subhasree Mandal, Mohammad Al-Fares, Min Zhu, Richard Alimi, Chandan Bhagat, Sourabh Jain, Jay Kaimal, Shiyu Liang, Kirill Mendelev, et al. B4 and after: managing hierarchy, partitioning, and asymmetry for availability and scale in google's software-defined wan. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 74–87, 2018.

[29] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM computer communication review*, 38(2):69–74, 2008.

[30] Pat Bosshart, Glen Gibb, Hun-Seok Kim, George Varghese, Nick McKeown, Martin Izzard, Fernando Mujica, and Mark Horowitz. Forwarding metamorphosis: Fast programmable match-action processing in hardware for sdn. *ACM SIGCOMM Computer Communication Review*, 43(4):99–110, 2013.

[31] David L Tennenhouse and David J Wetherall. Towards an active network architecture. In *Proceedings DARPA Active Networks Conference and Exposition*, pages 2–15. IEEE, 2002.

[32] Martin Casado, Michael J Freedman, Justin Pettit, Jianying Luo, Nick McKeown, and Scott Shenker. Ethane: Taking control of the enterprise. *ACM SIGCOMM computer communication review*, 37(4):1–12, 2007.

[33] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, et al. P4:

Programming protocol-independent packet processors. *ACM SIGCOMM Computer Communication Review*, 44(3):87–95, 2014.

[34] David Wetherall and David Tennenhouse. Retrospective on" towards an active network architecture". *ACM SIGCOMM Computer Communication Review*, 49(5):86–89, 2019.

[35] Josh Broch, David A Maltz, David B Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 85–97, 1998.

[36] Brad Karp and Hsiang-Tsung Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254, 2000.

[37] Ben Leong, Barbara Liskov, and Robert Tappan Morris. Geographic routing without planarization. In *NSDI*, volume 6, page 25, 2006.

[38] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level measurements from an 802.11 b mesh network. In *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 121–132, 2004.

[39] Douglas SJ De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 134–146, 2003.

[40] Sanjit Biswas and Robert Morris. Opportunistic routing in multi-hop wireless networks. *ACM SIGCOMM Computer Communication Review*, 34(1):69–74, 2004.

[41] Ion Stoica, Robert Morris, David Karger, M Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM Computer Communication Review*, 31(4):149–160, 2001.

[42] Bram Cohen. Incentives build robustness in bittorrent. In *Workshop on Economics of Peer-to-Peer systems*, volume 6, pages 68–72. Berkeley, CA, USA, 2003.

[43] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels. Dynamo: Amazon's highly available key-value store. *ACM SIGOPS operating systems review*, 41(6):205–220, 2007.

[44] Jialin Li, Ellis Michael, Naveen Kr Sharma, Adriana Szekeres, and Dan RK Ports. Just say {NO} to paxos overhead: Replacing consensus with network ordering. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 467–483, 2016.

[45] Jialin Li, Jacob Nelson, Ellis Michael, Xin Jin, and Dan RK Ports. Pegasus: Tolerating skewed workloads in distributed storage with in-network coherence directories. In *14th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 20)*, pages 387–406, 2020.