First, the basic details

- Class meets every Tuesday and Thursday between 11:00AM and 12:15PM in room 425 CSB

- **Instructor:** Lorenzo De Carli (ldecarli@colostate.edu; room 360 CSB)

- **Lab:** no separate lab hours - we are going to discuss assignments in class

- **Office hours:** contact instructor to make an appointment

- **Course web page** (for general info, syllabus and schedule): [http://www.cs.colostate.edu/~cs557](http://www.cs.colostate.edu/~cs557)

- **Course Canvas page** (for questions, discussions, assignments and grades): [https://colostate.instructure.com/courses/59406](https://colostate.instructure.com/courses/59406)
What is this class about?

• A review of advanced concept in networking

• Broad structure:
  
  • 1st half (~until first midterm): foundational concepts (network design, routing, etc.)
  
  • 2nd half: recent topics in networking
  
  • (We are going to make an exception for Software-Defined Networking, which is a recent topic but also the topic of the first class project)
What is this class not about?

- This class is **not** an introduction to networking and the Internet
  - I am going to assume you already know the basics… and then some (if you don’t, you should enroll in CS457 instead)
  - Instead, we are going to review **advanced concepts on which the field is built**, and novel ideas
What are the goals of this class?

1. Ensure students understand what are the principles behind modern networking: *why are things the way they are?*

2. Teach students to **think critically** about network design, configuration, security, etc.

3. Prepare students to become **productive researchers** (aka “people who solve problems yet unsolved”) in the field of networking
How is this class organized?

• This course is set up as a research-focused class.
  
  • No textbook
  
  • The bulk of the class will consist in reading scientific papers and discussing them critically
  
  • The class will include two midterms which will evaluate the students’ knowledge and understanding of the reading material (papers)

  • Student will also be expected to complete a mini-project and a research-oriented final project (more on this later)
Participation

• Attendance is **required** for local students

• It is acceptable to miss **up to 2 classes** - students who miss more than that without a valid reason will receive a **failing grade**

• **Bottom line:** if you miss more than 2 classes, contact me as soon as possible!
Paper reviews

• Before each class, students must **read the assigned paper** (see [http://www.cs.colostate.edu/~cs557/schedule.html](http://www.cs.colostate.edu/~cs557/schedule.html)) and **post a review** on the class Canvas page ([https://colostate.instructure.com/courses/59406](https://colostate.instructure.com/courses/59406))

• During each class, the paper will be presented in details and students will be encouraged to engage in critical discussion

• Remote students are encouraged to ask questions and post observations on Canvas
Paper reviews

Only “assigned readings” are mandatory - “Additional readings” are going to be discussed in class (and it is a good idea to take a look at them), but you are not expected to review them.
Paper reviews - II

- Discussion topics will be opened the Friday before the week in which each paper will be discussed.

- Discussions will be due at the end of the day before the class section in which each paper will be discussed (exception: first review, which will remain open until the end of this week).

- Make sure you are posting your review under the correct discussion topic!

Click on the discussion topic referring to the paper you just read; submit your review.
Each review will receive a score between 0 and 100. The three lowest score will be dropped.

0 points will be assigned for reviews which are not submitted; no late submissions will be accepted.

Reviews are required - Failing to submit more than three reviews will cause the student to receive a failing grade.
How to read papers

• In order to successfully understand and review scientific papers, it is important to learn how to read them

• Common mistake: read a paper sequentially the way you would read a book, dedicating equal attention to all parts

• It almost never works - you will drown in the details, miss the big picture, and get very frustrated

• Instead, try reading a paper selectively, focusing on the most important parts first
How to read papers - II

• The trick is to read a paper **multiple times**:

  • **First step:** read abstract, intro and conclusion; skim the beginning of every section in between. **Goal:** understand which problem the paper aims to solve, and get a very basic understanding of how it gets solved

  • **Second step:** read the paper end-to-end, trying to understand the technical details of how problems are solved.

  • For particularly complex papers, **you may need to repeat the second step multiple times**, every time delving deeper into the technical details

• **Additional readings** (linked on the course web page):

  • M. Hanson, “Efficient Reading of Papers in Science and Technology”

  • S. Keshav, “How to Read a Paper”
How to read papers - III

• **Summarizing:** skim the paper first to understand what is the problem being solved, and then try to refine your understanding with further readings

• What happens if you follow this method:

  • **Pass 1:** *this paper proposes some hardware which can execute different networking algorithms*

  • **Pass 2:** *this paper propose a dataflow-based processor which can efficiently run multiple forwarding and classification algorithms, overcoming the limitations of traditional network ASICs*
How to read paper - IV

- What happens if you don’t follow this method and try to understand everything at once:
  - Pass 1: this paper solves a problem in networking
  - Pass 2: this paper solves a problem in networking
  - Pass 3: this paper solves a problem in networking
  - Pass 4: this paper solves a problem
  - Pass 5: this paper
  - Pass 6: ZZzzzzz…
How to read papers - V

• Other tips:

  • Take notes

  • Form informal reading groups with your classmates so you can read and discuss the paper together

  • Reviews must still be individual!

  • Come to the lecture prepared for discussion (especially if there are aspects of a paper which you did not understand)
How to review papers

• A paper review is a discussion of the paper topics, merits, and issues

• Reviews are fundamental for modern science - they allow the scientific community to determine the correctness and relevance of scientific work

• For the purpose of this course, writing reviews will force you to truly reason about each paper, and ensure that you understood the core ideas
How to review papers - II

• A review of a paper is not an unsupported subjective judgement in the style of Youtube comments (“This paper is great”, “This work sucks”, etc.)

• A review of a paper is not a summary, however long - it must contain your original assessment of the paper merits and issues
How to review papers - III

• In order to assess the technical merits of a paper, you must ask yourself questions such as:

  • Is the problem solved by the paper relevant? What is the impact of the work presented here?

  • Do the experiments satisfyingly back up the paper’s claims?

  • Are the experiments sound?

  • Is the paper theoretically sound?
How to review papers - IV

• How can I judge if a paper is well-written and clearly presented?
  • Does the paper clearly state the reasoning and insight behind the solution, and the lessons learned?
  • Does the paper put the results in context (i.e. does it provide background and motivation)?
  • Is the paper written in a clear and concise manner?
How to review papers - V

- For the purpose of this class, reviews must include:
  
  - A summary of the contents of the paper, which must include an outline of the core challenges and how they are solved
  
  - A discussion of the positives, if any
  
  - A discussion of the negatives, if any
  
  - A conclusion summarizing your thoughts on the paper
  
- Each section must consist of a short paragraph; overall length of each review should be approximately between 300 and 500 words
Review grading

• Each review will receive a score between 0 and 100

• Aspects I will evaluate:

  • **Length:** is the review too short or too long (“too long” means that you could have expressed the same concepts in a much smaller number of words)

  • **Structure:** does the review well-structured as summary of paper/discussion of positives and negatives/conclusion?

  • **Insight:** does the review formulates an original, insightful assessment of the paper, or is it just a summarizing and/or repeating what stated in the paper?

  • **Plagiarism:** Is the review copied and pasted from the results of a Google search?
Bad review examples

• “This paper was great” - 0 points

• “This paper was great, it showed how TCP congestion issues can be alleviated by incorporating slow start and congestion avoidance” - 0 points

• “This paper was great, it showed how TCP congestion issues can be alleviated by incorporating slow start and congestion avoidance, but it also had some issue for example some points were not clearly explained. Overall, still pretty cool!” - 0 points

• “This paper presents” <elaborate summary of the paper w/o any critical consideration> - 0 points
This article describes the slow start and congestion avoidance algorithms, through which TCP is able to tune the rate at which data are sent on the network. The aim of the authors was to create a protocol that (i) could make the best use of the available bandwidth and (ii) could quickly respond to network congestion by reducing the sending rate.

One of the most interesting aspects of the proposed approach is that it is designed for a completely uncooperative network. TCP works without explicit knowledge of the link it is using - it gathers the information it needs while the connection is running. Moreover, congestions are detected without the need of explicit congestion notifications from the gateways. This radical application of the end-to-end principle makes TCP somewhat limited in its features but also very flexible - it does not need any "help" from the underlying levels, so it works almost everywhere. Also, despite its complexity, the TCP state machine requires limited processing power. Therefore, it can be implemented on low-end machines such as embedded devices.

The main problem of Van Jacobson's approach is that packet losses are always interpreted as a signal of congestion. While this is reasonable - as the author say, this kind of "signal" is always delivered by any type of network - it can cause unnecessary slowdowns in modern WiFi networks. In fact, such radio links are prone to packet losses that are unrelated to congestion and should not trigger congestion avoidance. The consequence is that, on lossy wireless link, often TCP is not able to exploit all the available bandwidth. However, this limitations should be seen in historical perspective: when TCP was designed, the main cause of packet loss was congestion, so the decision makes sense.

Overall, I really enjoyed this paper. Van Jacobson's algorithm is solidly based on previous work on queuing and network congestion, detailed mathematical analysis and sparkling intuitions. The language is plain and clear, with vivid images and a touch of irony which keeps the attention of the reader. Contributions and ideas from other authors are always clearly recognized.
In-class questions

• The more you engage in **questions** and **discussion** during the class, the more you will benefit from the class

• Distance students will benefit too since a good discussion always bring up original points of view and interesting remarks

• If you are a distance student, you are encouraged to **contribute** by posting your comments and questions on the **Canvas discussion** for the paper

• **Tips for discussion:** be prepared to articulate your point of view and to discuss it against objections; be prepared to change your mind; **be respectful of classmates**
Exams

- **2 midterms**, each covering roughly half of the class topics
  - Second midterm may include questions about fundamental concepts *reviewed in the first half of the class*
- **No final exam** (replaced by the final project)
- Refer to class schedule for midterm dates
- **No review sessions planned**, but we may have them if the class feels strongly about it
Mini-project

- The mini-project will consist of a standardized medium-size network implementation assignment in C/C++ (same for all students)
  
  - Will most likely be related to Software-Defined Networking

- **Goal:** ensure that you have the programming skills and the familiarity with networking concepts necessary for successfully complete the final project

- Due early in the semester (refer to schedule for details)

- Will post details on the website soon and discuss them in class
Final project

• For the final project, students will be expected to complete research-oriented work (you may work individually or in pairs; a two-student project is expected to be more substantial than a one-student project)

• Each team must work on a different project idea

• You may either come up with your own project idea, or come talking to me for advice

• Possible project types: preliminary evaluation of a novel idea, repeatability study on published results

• You are not limited to the topics discussed in class - you can make your project about any topic as long as it is networking-related!

• If you are working on networking-related research projects with your advisor, you may consider choosing a project related to your research

• DO NOT REUSE MATERIAL FROM OTHER CLASS PROJECTS
Final project - II

- Completing the project will involve: (i) performing any implementation and evaluation work the project may require; (ii) give a project presentation in front of the class; and (iii) write a final project report detailing the work you did.

- You should decide on a project topic and have it approved by me by early March

- I am going to meet with each team regularly to ensure everyone is making progress

- We will have project presentations during the last two lectures (distance students will need to send me a recording of the presentation, which I will play in class)

- The final project report will be due by end of the day on 5/9/2018

- More details to come soon
Grading

- Graded paper reviews: 20%
- Midterm #1: 20%
- Midterm #2: 20%
- Mini-project: 15%
- Final project: 25%
Some considerations

• This class is designed to be “fun”:

  • Not laser-tag fun

  • More like this type of fun:
Some considerations - II

• This class is going to be challenging in some respects, especially if you are new to paper-reading and research

• ... but that’s the whole point: challenge you so you can learn skills you did not possess before

• For this reason, there is a zero-tolerance policy on plagiarism: if you present online material as your own, have someone else do the work for you, etc., for even one assignment - you will receive a failing grade. You may also incur in serious consequences at the university level
Some considerations - III

• **More on plagiarism:** plagiarism is the act of presenting someone else’s work as your own. *This include copying and pasting text from any online source* (including Wikipedia) without referencing the source.

• Please be honest and do not disappoint me!
Some considerations - IV

• Class attendance and all graded actives (reviews, midterms, projects) must be completed in order to receive a passing grade

• Exceptions: you may miss up to 2 lectures and up to 3 paper reviews

• If you cross one of the thresholds above, contact me immediately to discuss the situation
Quick review
What is a network?

- A computer network can be recursively defined as:

  - Two or more nodes, connected by a link...
  - Two or more networks connected by one or more nodes

...OR...

Diagram showing two or more nodes and two or more networks connected by one or more nodes.
Nodes and links

- A **node** is pretty much any computing device that can communicate with other devices (PC, phone, smart device, car, etc.)

- A **link** is a component (physical or logical) which two or more nodes use to exchange data

- **Examples of links:**

  - ![Point-to-point](image1)
  - ![Wireless](image2)
Circuit vs Packet Switching

- **Circuit switching:** data between two nodes follows a static path which must be reserved and set up prior to communicating. Concurrent communication between different host pairs must use different paths.
Circuit vs Packet Switching - II

- **Packet switching**: data is broken into packets which are forwarded across links between nodes. Communication flows between different host pairs can be multiplexed on the same link, simply by alternating packets from different flows on the wire.

- Most modern networks are based on packet switching technology (although *virtual circuits* are sometimes used).
Protocol stack

- Modern computer networks are packet-switched and based on a stack of protocols which provide features at increasingly high level of abstraction.

- The hourglass design:

  - Physical (characteristics of physical signal)
  - Data-link (transmission of frames on link)
  - Network (routing across multiple networks)
  - Transport (reliable transmission of packets)
  - Application (high-level data stream)
The Internet

• The Internet is a world-scale network which interconnects a large number of smaller networks around the globe

• The Internet includes a large number of heterogeneous devices and wildly different link technologies

• What is the glue that keeps everything together? Protocols!

• As long as your device can speak TCP/UDP and IP, it can communicate on the Internet
A bit of history

- The Internet grew incrementally from a small set of hosts interconnected as a result of military and academic projects.

The Internet, 1969

The Internet, 2011
The Internet has structure

ISP

Backbones (national, global)

Regionals

Campus LANs, Business

Users
The Internet is a physical object!
How does anyone find anyone else on the Internet?

• Each node on the Internet is assigned a unique identifier (IP address) (not really, but acceptable simplification)

• IP address are numerical and hard to remember - hence public nodes are typically associated with a mnemonic: a string in the domain name system (DNS)
I have the IP address, now?

Routers send packet to next closest point

H: Hosts
R: Routers
Yes, but how does it work?

- Routers maintain forwarding tables which specify how to reach all portions of the assigned public IP address space.
Routing tables

• In the beginning, manually updated

• Now, routers coordinate using gateway protocols such as BGP

• Routing tables are big and must be looked up for each new packet - doing this at TBs/s creates enormous problems for hardware and software architects
### Challenges

- The Internet is one of the most complex objects ever built by mankind
- Building something of this scale and complexity requires solving many challenges:

<table>
<thead>
<tr>
<th>Time</th>
<th><strong>Existential challenges:</strong></th>
<th><strong>Scalability challenges:</strong></th>
<th><strong>Security challenges:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>How do I connect multiple computing devices together?</td>
<td>How do I ensure I can reach anyone in any location?</td>
<td>How to protect against worms, DDoS, etc.?</td>
</tr>
<tr>
<td></td>
<td>How do I ensure everyone can speak to everyone else?</td>
<td>How do I ensure flows share BW fairly and efficiently?</td>
<td>How to secure web applications?</td>
</tr>
<tr>
<td></td>
<td><strong>Packet switching</strong></td>
<td><strong>IP routing</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>TCP congestion avoidance</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>New paradigms: IPv6, CDN, NDN, etc.</strong></td>
<td></td>
</tr>
</tbody>
</table>
Challenges - II

• **Cross-cutting issue:** the need to support wildly different applications (some still to come!)

  • File transfer: loss sensitive, not delay sensitive, bursty

  • Audio: low-bw, jitter-sensitive, loss tolerant, long sessions

  • Video: high bandwidth, bursty, somewhat loss-tolerant
Internet evolution = sequence of disruptions

• New application disrupt business as usual

• Not easy to predict
  
  • Some technology can have a hard time being adopted despite having clear advantages (think of IPv6)…
  
  • …while new paradigms can come out of the blue and suddenly become very popular (think of P2P in the late ‘90s)

• Now the killer app is video-on-demand
  
  • Netflix generates 37% of North American Internet traffic (2015)

• What’s next?
Networking ⊃ Internet

• Plenty of interesting problems in specialized contexts!
  • Datacenter/cloud
  • Sensor networks
  • Mobile networks
  • …
• We are going to look at some of those in this class
See you on Thursday!