CT420 REAL-TIME SYSTEMS

LECTURE OUTLINE

Dr. Michael Schukat

Dr. Jawad Manzoor

Ollscoil na Gaillimhe
University of Galway

About Michael

- Background:
  - M.Sc. Computer Science
  - Dr. rer. nat. (Computer Science)
  - Senior Lecturer in the School of Computer Science at the University of Galway
  - Senior Embedded Systems Design Engineer (Ireland)
  - Embedded Systems Design Engineer (Germany)
  - Junior Lecturer and Researcher (Germany)
- Research Interests:
  - Many, including real-time systems
- Contact:
  - michael.schukat@universityofgalway.ie
  - Office CSB3002

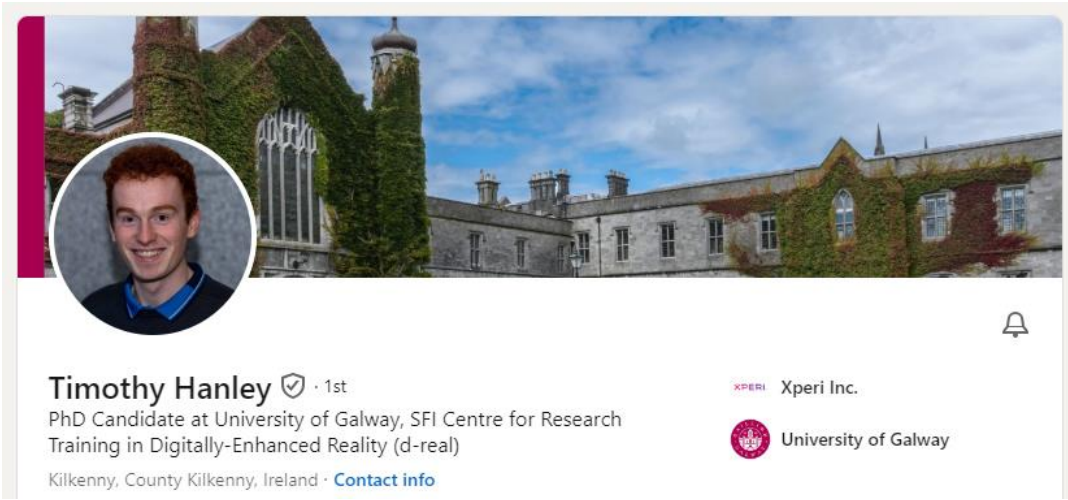
About Jawad

- Education:
  - MSc. (Distributed Computing) Erasmus Mundus scholar at Universitat Politècnica de Catalunya (UPC) Barcelona, Spain + KTH Royal Institute of Technology, Sweden
  - PhD (Computer Science) Erasmus Mundus fellow at Université catholique de Louvain (UCL), Belgium + UPC, Spain.
- Background
  - Lecturer at University of Galway
  - Lecturer at Atlantic Technological University (ATU Donegal)
  - Lecturer at Air University, Pakistan
  - Research Engineer and CA Technologies, Spain
- Research Interests:
  - Analysis of network protocols, network traffic measurements, network security
- Contact:
  - jawad.manzoor@universityofgalway.ie
  - Office CSB3012

Your Support Team / Lab Tutors

- James Wei:
  M.Sc. (AI) student

- Timothy Hanley:
  2nd year PhD student



Some Housekeeping …

Use of Canvas

- Announcements
  - **Please check regularly**
- Information
  - Contains module outline, breakdown of marks, etc.
- Learning Materials
- Assessment
  - Assignment & quizzes
- Discussion Forum
  - Mainly used for assignment-related questions
- Quickly attendance (used later for every lecture)
- Virtual Classroom
  - Possibly used for virtual labs
- End-of term student feedback questionnaire

Lecture Organisation / Breakdown of Marks

- 2 hours of lectures per week
  - Wednesday 15:00 – 16:00 in the Tyndall Theatre
  - Thursday 10:00 – 11:00 in CSB1006 (formerly CT250)
- 2 hours of (F2F and virtual) labs per week (from week 3, tbc)
- There is a 70/30 breakdown between exam and CA
- The CA component consists of
  - 2 assignments, worth 10% each
  - 2 in-class quizzes between week 7 and week 12, worth 5% each
- Occasionally we'll be also using **Mentimeter** or **Vevox** for in-class activities

Flipped Learning

- In some lectures we'll apply the concept of **flipped learning:**
  - You'll be notified via Canvas and are asked to study the learning materials prior to the weekly lectures
  - If you have specific questions about content, please let us know in good time, so that we can incorporate them into our lecture slots that week

Assignment Content Overview

- Mostly practical & hands on
- Use your own laptop / computer
- The assignments will include the following:
  1. Software development and benchmarking
  2. Network protocol benchmarking and analysis
- The assignments will require you to run VMs or container

Main Learning Outcomes

On successful completion of this module you will have:

1. A solid understanding of time synchronisation protocols including NTP and PTP
2. Hands-on experience with the time synchronisation protocol NTP
3. Hands-on experience with the POSIX API to design real-time systems
4. A solid understanding of the design principles for hard and soft real-time software systems
5. An in-depth understanding of the the design and performance of QUIC and HTTP3 protocols
6. A solid knowledge soft real time systems and multimedia protocols

Learning Materials and Textbooks

- Weekly presentations
- There's no primary textbook used in this lecture
- Instead, we'll provide you with links to additional sources, e.g.
  - articles
  - eBooks
  - source code

  as we go along

CT420 Main Topics Week 7 - 12

- Protocols for Soft Real-Time Systems: RTP, VoIP, WebRTC, DASH
- Emerging Application Layer Protocols: Web QoS, Web evolution, HTTP2 protocol
- Emerging Transport Layer Protocols: TCP performance issues, QUIC protocol
- QUIC Congestion Control
- QUIC Logging, Debugging and Visulisation (qlog, qviz)
- Web Performance Benchmarking

CT420 Main Topics Week 1 - 6

- Time Synchronisation algorithms and protocols for Real-time systems (RTS)
- Real-time systems engineering
  - Design approaches
  - Process scheduling models
  - Programming models and OS support for RTS
    - Real-time OS, example POSIX.4 RT
    - RT Java / C
- Safety-critical systems engineering
  - Hardware, software and data perspective

Recap: Functional versus non-functional Requirements

Functional Requirement	Non-functional Requirement (NFR)
Product feature	Product property
Describes the actions with which the user's work is concerned	Describe the experience of the user while doing the work
A feature or function that can be captured in use-case	A global constraint and as such difficult to capture in use case
A behaviour that can be analysed via sequence diagrams or state machines	A software quality
Can be usually traced back to a single module / class / function	Usually cannot be implemented in a single module or even program

Typical NFR include:
Availability, maintainability, **performance**, privacy, **reliability**, scalability, **responsiveness**

What is a Real-Time System (RTS)?

- *A system is said to be real-time if the total correctness of an operation depends not only upon its **logical correctness**, but also upon **the time** in which it is performed*
  - Functional (logical correctness) versus non-functional (time constraints) requirements
  - Fancy example: Boston Dynamics robots, see
    - https://www.youtube.com/watch?v=_EZQx87DyzM
    - https://www.youtube.com/watch?v=50eli-eOPO4

RTS Categories

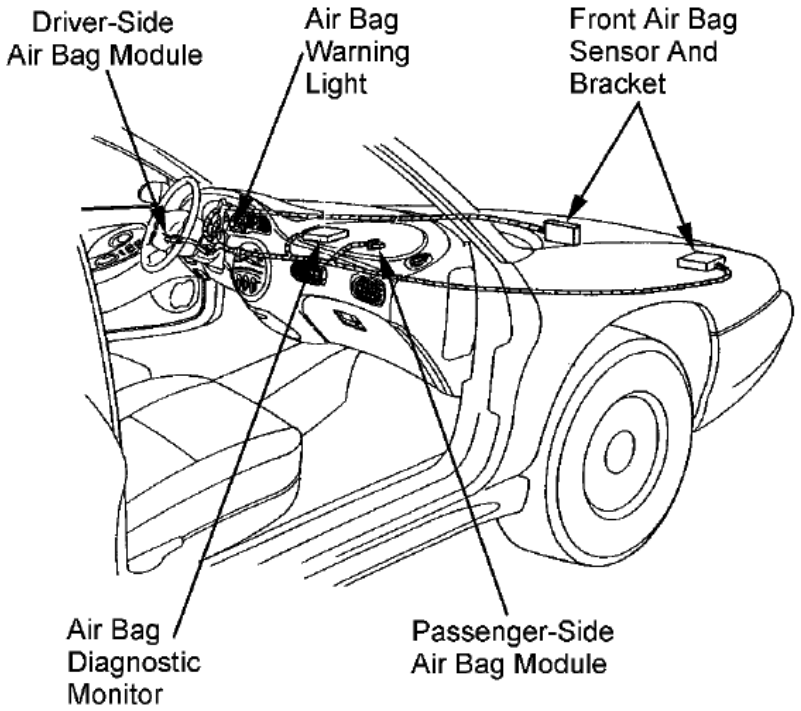
- Two major categorisation factors: criticality and speed
  - Criticality:
    - Hard RTS - deadlines (responsiveness) is critical. Failure to meet these have severe to catastrophic consequences (e.g. injury, damage or death)
    - Soft RTS - deadlines are less critical; in many cases significant tolerance can be permitted
  - Speed:
    - Fast RTS - responses in the microseconds to hundreds of milliseconds
    - Slow RTS - responses in the range seconds to days
- What type of RTS are the following:
  - Airbag control system
  - Missile defense system
  - Video conferencing system
  - Instant messaging tool

What is a Safety-Critical System (SCS)?

Features

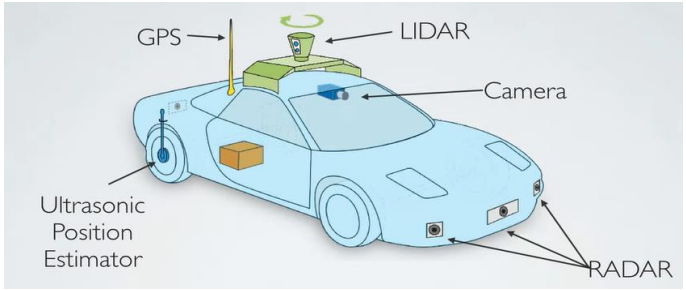
- A safety-critical system or life-critical system is a system whose failure or malfunction may result in one (or more) of the following outcomes:
  - death or serious injury to people
  - loss or severs damage to equipment/property
  - environmental harm

Example Airbag



Example Autonomous Vehicles

- https://www.youtube.com/watch?v=4sCK-a33Nkk
- https://www.youtube.com/watch?v=taMP_n3wL7M
- https://www.youtube.com/watch?v=3mnG_Gbxf_w
- Real-time system
- Safety-critical system
- Precise time
- Precise timing



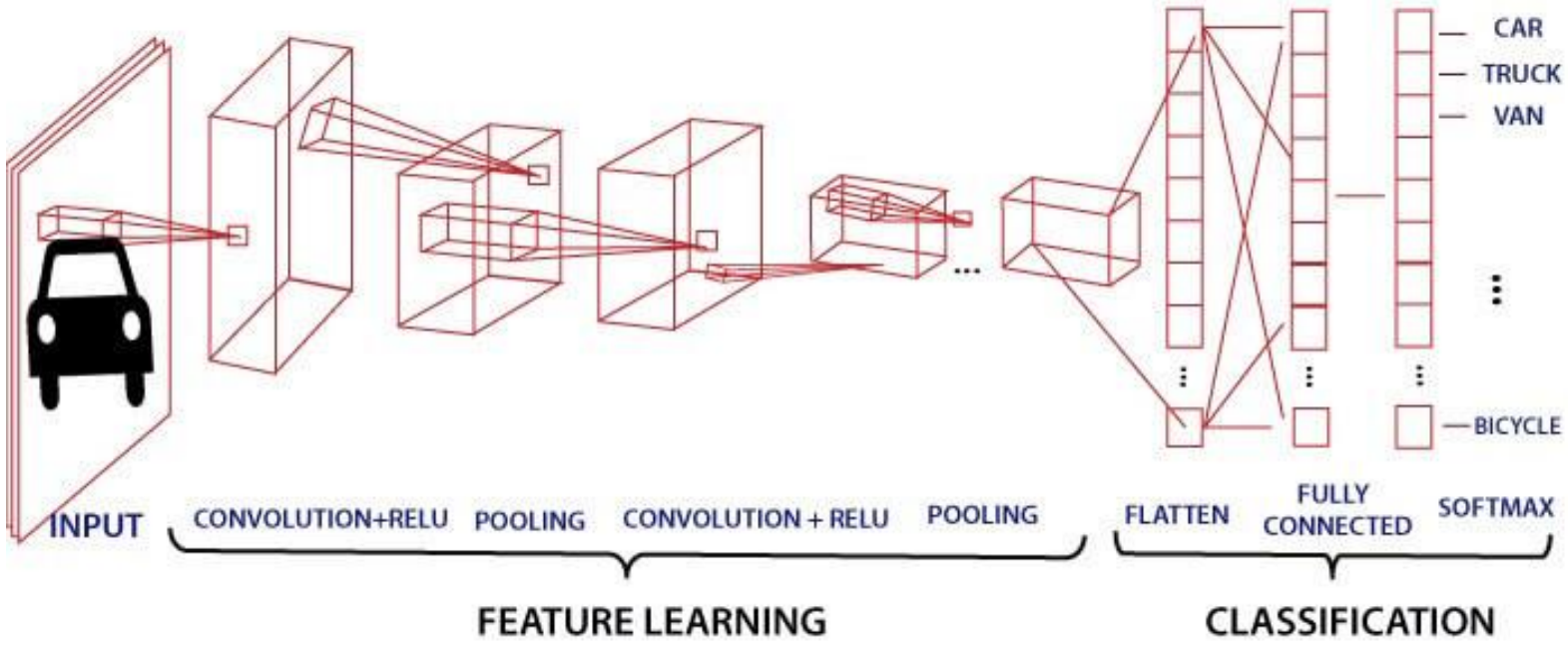
Autonomous Vehicle: Real-Time Aspects

Autonomous Vehicle: Safety-Critical Aspects

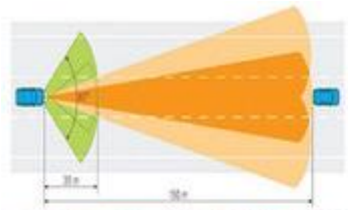
How to fool AI: https://medium.com/swlh/how-to-fool-artificial-intelligence-fcf230bf37e

FYI: Convolutional Neural Networks and Image Classification

- Source: https://www.javatpoint.com/pytorch-convolutional-neural-network
- Black Box AI versus explainable AI



Autonomous Vehicle: Precise Timing Aspects



RADAR (long range)

RADAR (short range)

LiDAR

Vision

GPS, speedometer, accelerometer, compass, ..

Thermal imager

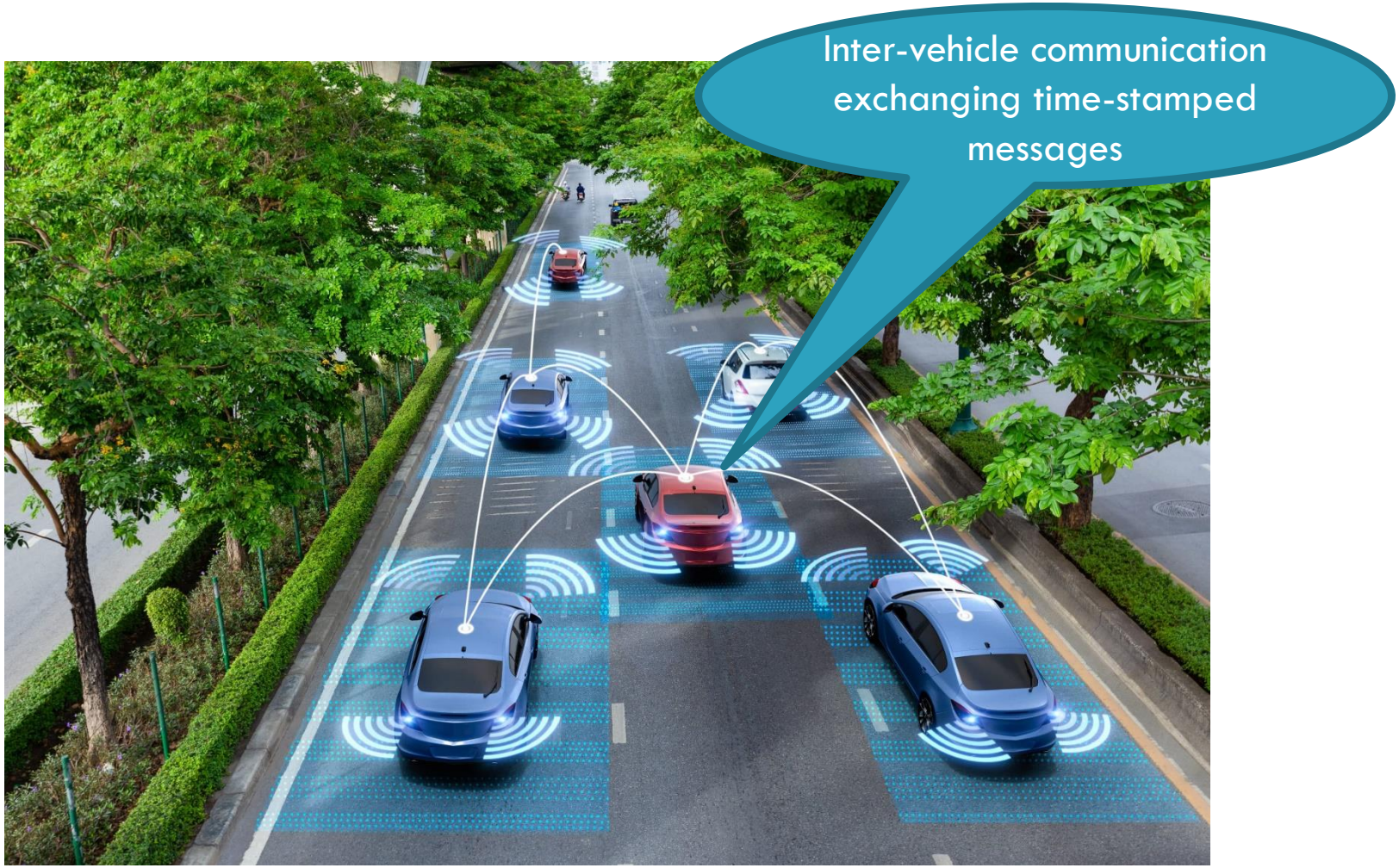
Sensor Fusion

Sensor data must be synchronised

Map

V2X
"seeing around corners"
Other traffic, road conditions

Autonomous Vehicle: Precise Time Aspects



Outlook for next Lectures

- The Essence of Time: From Measurement to Navigation and Beyond
- Time Synchronisation in Distributed Systems
- Time Synchronisation Protocols (NTP and PTP)