Drone Communication Protocol (DCP)

Department of Computer Science

Peter Franchina, Matt Mayer, Cam Pepin, Joe Puzzo
Advisor: Radim Bartos

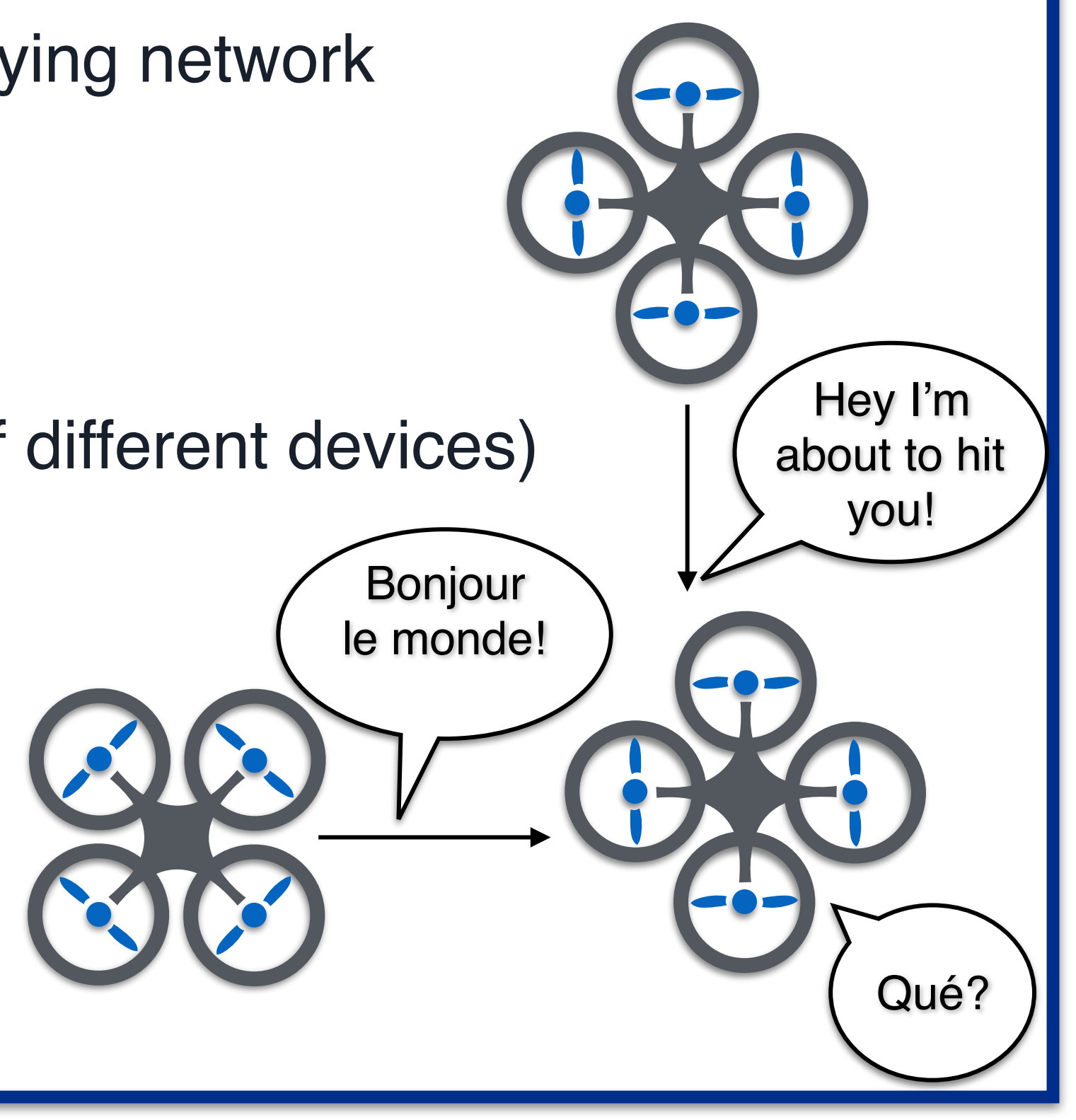
Background

"existing technology does not appear to provide a way to resolve the "see and avoid" problem with small UAS operations without maintaining human visual contact with the small unmanned aircraft during flight." (FAA-2015-0150)

Drones are becoming increasingly relevant in both commercial and recreational spaces. As more drones take to the skies, we must begin to consider a standard method in which they can communicate. We propose a coordination protocol and network stack that enables drones to avoid each other, stay within their allowed airspace, report problems, share information, and receive orders.

Project Goals

- Define a standardized method for communication
  - Develop application layer protocol
  - Research and define underlying network stack
- Implement protocol
  - Easily extensible
  - Highly portable (multitude of different devices)
- Simulate use of protocol
  - Full stack
  - Real time
  - Scalable



Network Stack

Application

The Drone Communication Protocol provides a method for drones to share data with neighboring nodes through a variety of different message types. The DCP envelope is designed for extensibility by allowing implementors to insert their own vendor-specific payloads.

The HIA (Here I Am) message is an example of a required protocol message. It holds basic information pertaining to a drones location and is broadcast repeatedly.

```
{"envelope":{
   "version":1.0,
   "local":true
   “date”:”2012-04-23…”,
   "to":{
      "id": “drone1”,
      "port": 8082,
      "address": 3000::2
   },
   "from":{
      "id": “drone2”,
      "port": 8082,
      "address": 3000::1
   },
   "payload":[
      {"message":{
         "type":"hia",
         "lat":46.49829336,
         "lon":7.567411672,
         "alt":1343.127
      }}
   ]
}}
```

Figure 1: example DCP packet

Transport

Due to the constantly changing network topology we used UDP for our transport layer. The stateless nature of UDP reduces network overhead required by session establishment and maintenance and allows broadcast data.

Network

IPV4 multicast packets
Multicast packets were utilized to send broadcast data.

Routing
- Optimized Link State Routing (OLSR)
- Enables ad-hoc communication
- OLSR keeps track of all 1 hop 2 hop neighbors and stores link information
- Configured implementation for faster route discovery
- Nodes determine which routes to use based on the network topology information they receive from their neighbors

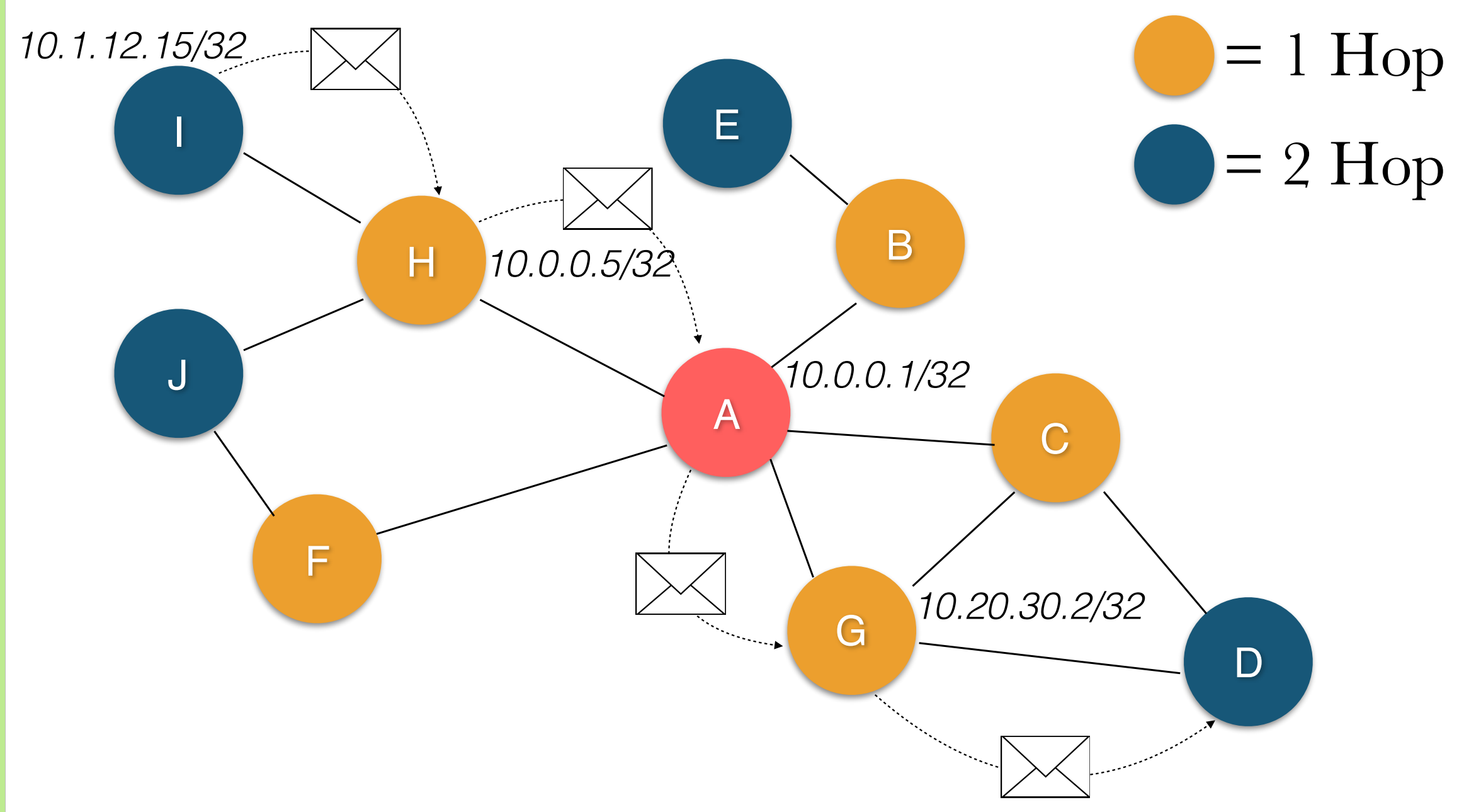


Figure 2: node A neighbor discovery and example transmission from I to D

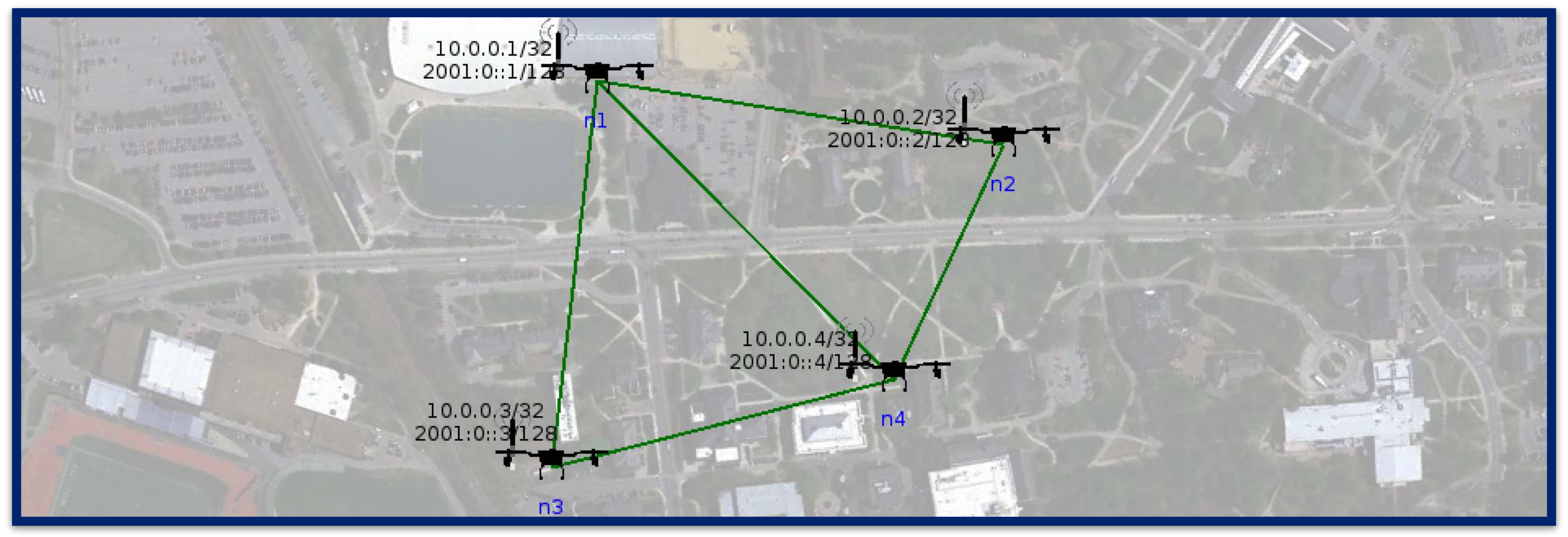
Link & Phy

WiFi

Simulation

CORE (Common Open Research Emulator)

- Used to simulate network and network conditions
- Provides an easy to use interface capable of spawning custom Linux containers
- Built in node mobility functionality
- Each Linux container runs its own network stack, thereby allowing us to simulate physical devices in real time



Applications

- Collision Avoidance
- Swarm Navigation
- Smart Airspace
- Delivery
- Crowdsource Information
- Shared Internet access

Collision Avoidance

- Operates through the transmission of trajectory-segments
- Collisions are avoided by inserting pause times into flight plans

Implementation

Transmission and reception of multicast UDP packets is implemented with JavaScript running on top of Node.js. This was conducive to the lightweight, simple, and extensible nature of our implementation. The code shown below illustrates the sending of a HIA message from a node 'A' to another node 'B'.

```
var sender = new Messenger(‘A');
var receiver = new Messenger('B');
var hia  = new Messages.HIA( lat, lon, alt );
receiver.on( 'hia', function( msg, from ) {
             console.log('Received HIA from’,from.id);
         });
sender.sendLocal([hia], {id: 'B'});
```

Evaluation

- Tested on Raspberry Pis
- Built an interface for communicating with Drone flight controller

Future Work

- Test in physical environment with actual drones
- Extend the core functionality (message types) in DCP
- Format design document to IETF standards