dortmund
university

OMNeT++ Community Summit 2016

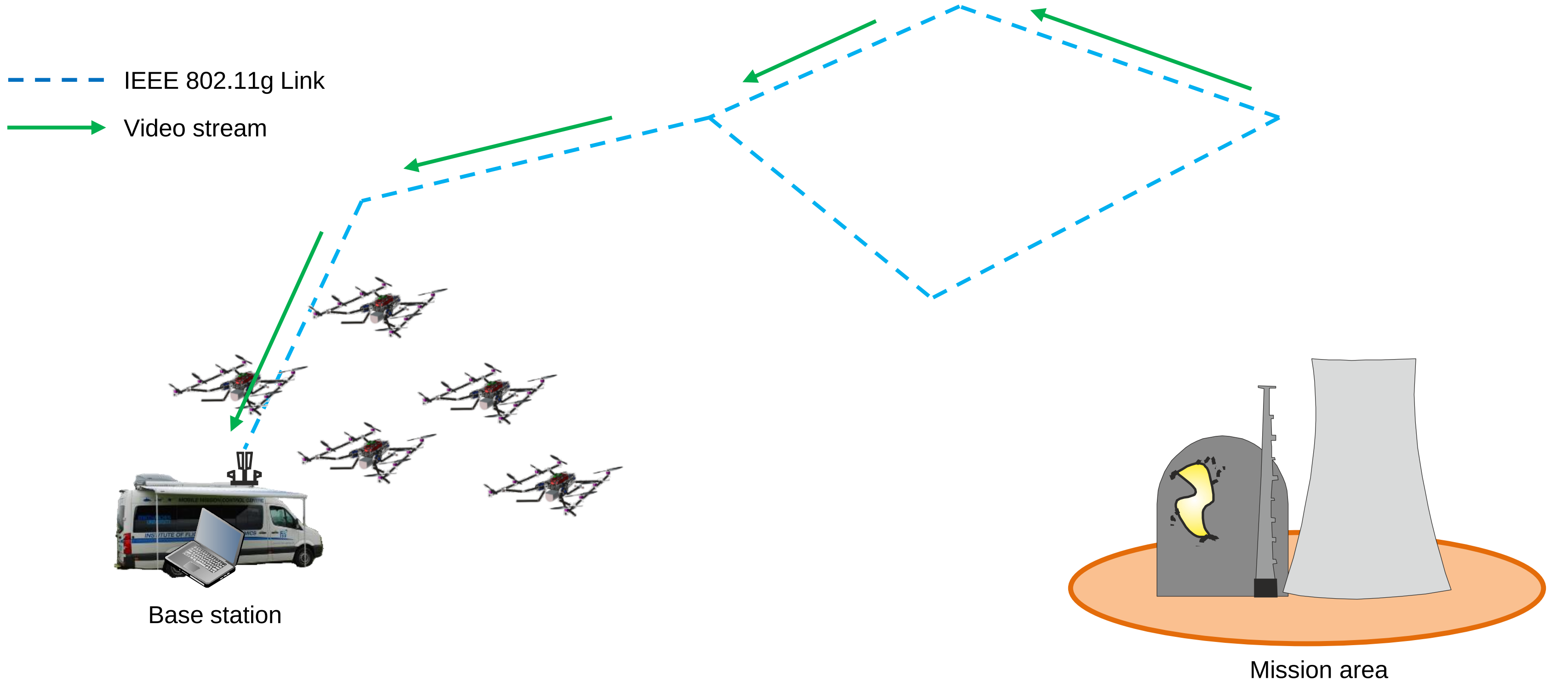
An OMNeT++ based Framework for Mobility-aware Routing in Mobile Robotic Networks

Benjamin Sliwa, Christoph Ide and Christian Wietfeld

September 16, 2016

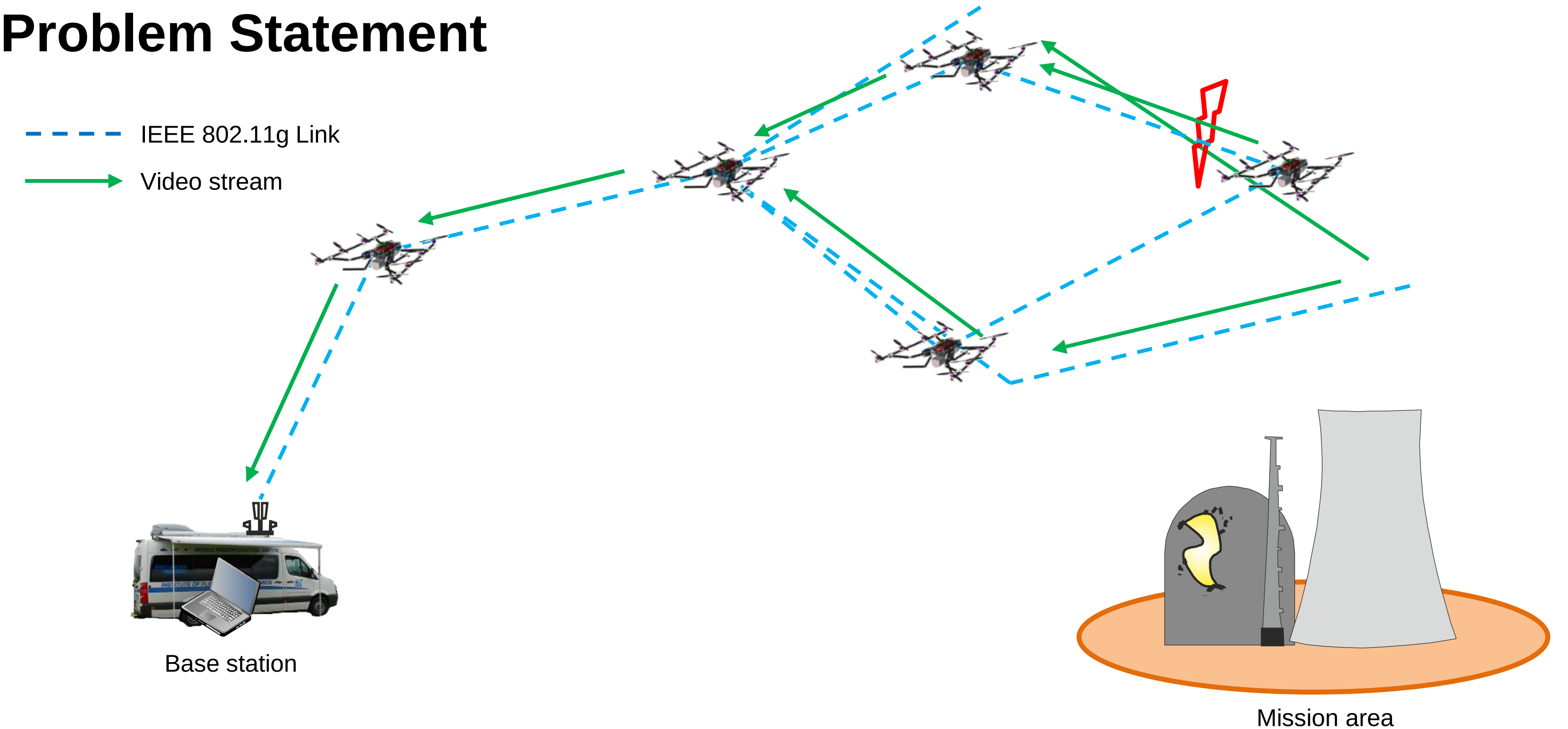
Faculty of Electrical Engineering & Information Technology
Communication Networks Institute
Prof. Dr.-Ing. Christian Wietfeld

Problem Statement



- Application of autonomous agents for exploration of hazardous areas

Problem Statement

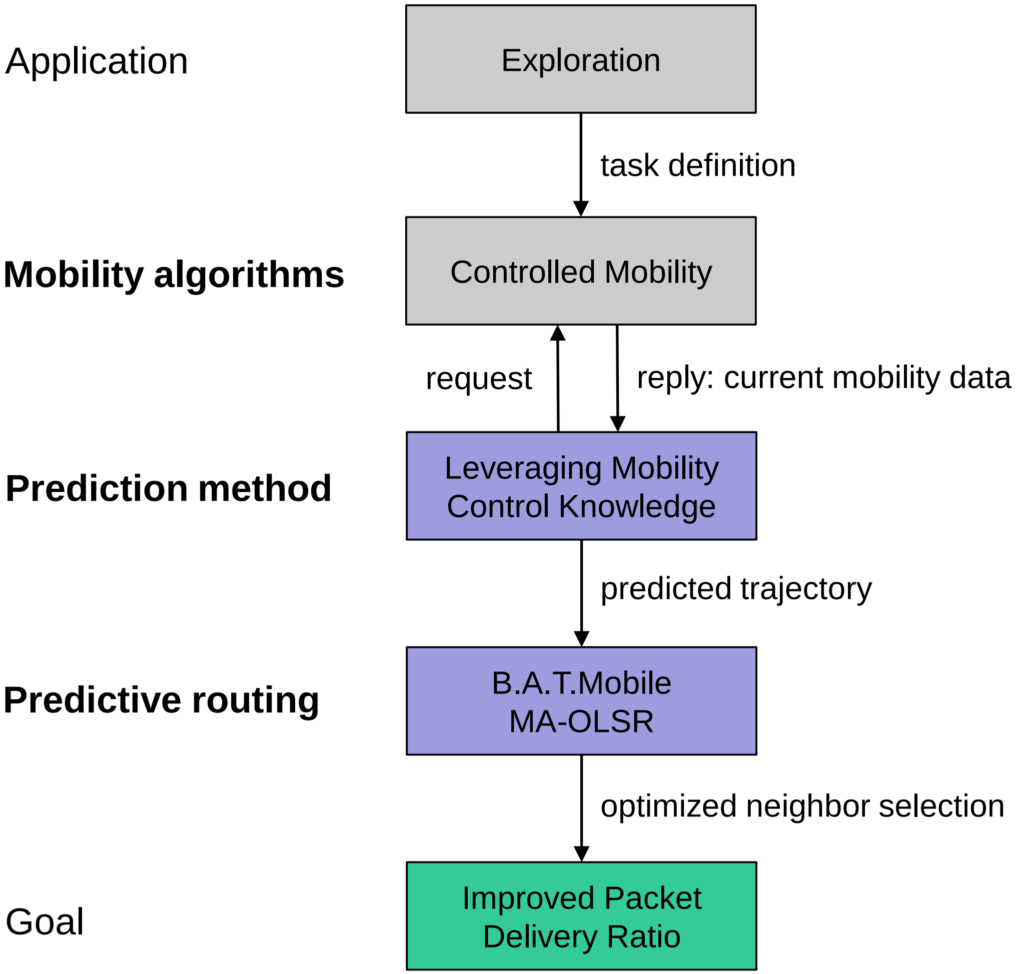


- Application of autonomous agents for exploration of hazardous areas

- High mobility causes frequent changes of the network topology
- ➢ Stressed routing protocols and losses of the communication link

Solution approach: using mobility control knowledge for intelligent forwarder selections

Solution Approach

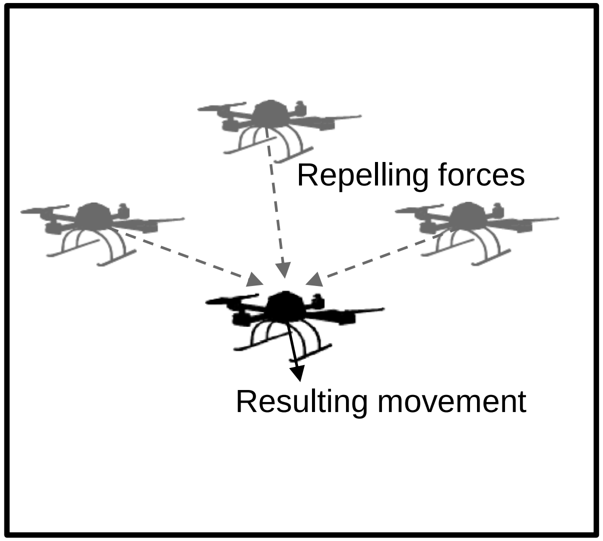


- Meta-model for realistic modelling of cooperative agents
- Trajectory prediction for precise position estimation
- Mobility-aware routing approaches
- Proof of concept evaluation

OLSR: Optimized Link State Routing

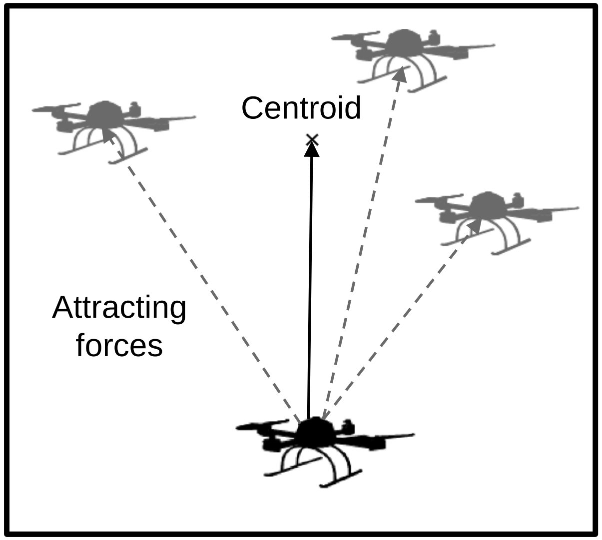
Modelling the Mobility Behavior of Autonomous Agents

Three basic Rules for the Behavior of Individual Agents Travelling in Swarms



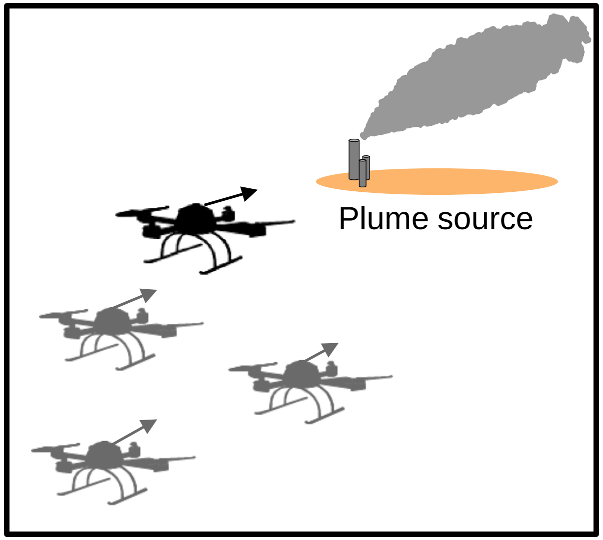
Separation

- **Collision avoidance**
- Potential fields / position-based



Cohesion

- **Swarm coherence**
- Potential fields / position-based

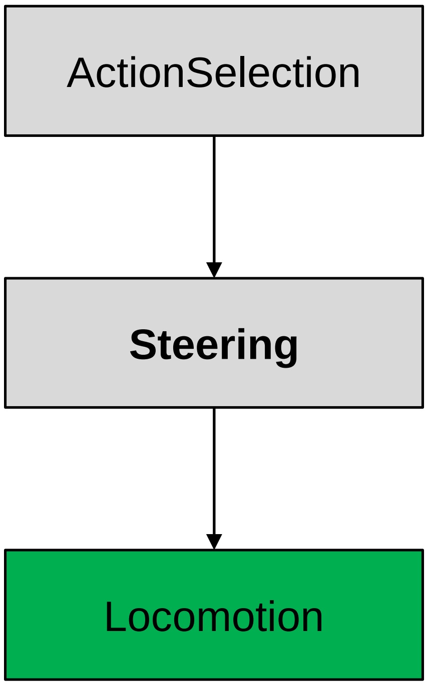


Alignment

- **Task fulfillment**
- Cooperative mobility algorithms

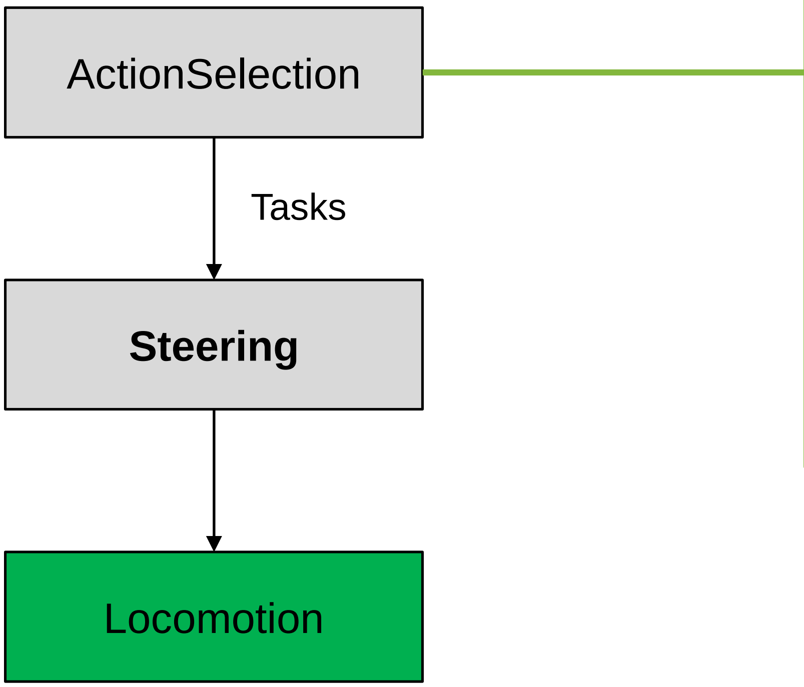
Real-world swarming scenarios require multiple mobility algorithms to be executed in parallel
→ Meta-model for mobility

Layers of the Reynolds Mobility Meta-model



[2] Reynolds, C. W., “Steering behaviors for autonomous characters,” in *Game developers conference,* San Francisco, California 1999

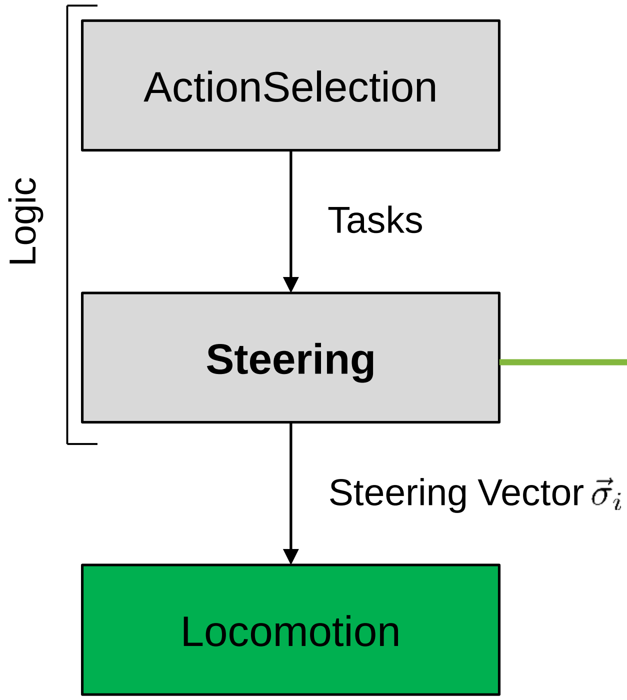
Layers of the Reynolds Mobility Meta-model



- Task definition
  - Exploration
  - Network Provisioning
- Agent classes
  - Scout
  - Relay

[2] Reynolds, C. W., “Steering behaviors for autonomous characters,” in *Game developers conference,* San Francisco, California 1999

Layers of the Reynolds Mobility Meta-model



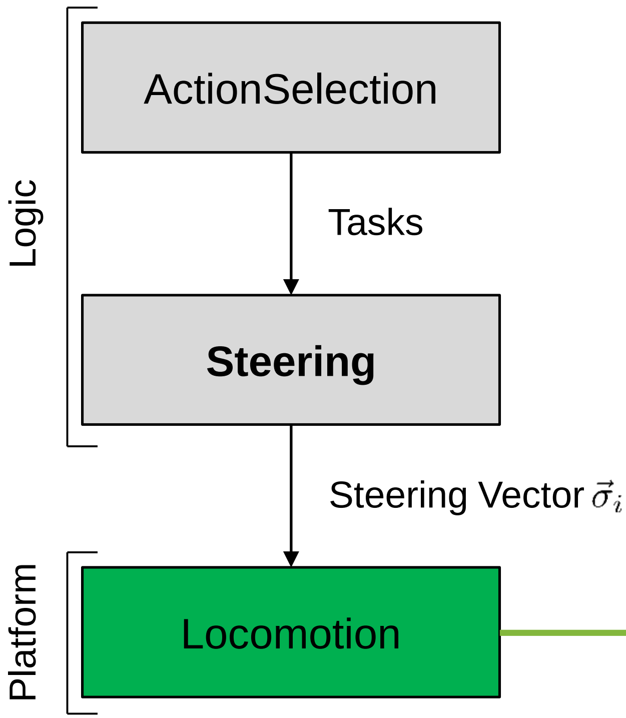
- Fulfillment of individual tasks
  - Path planning
  - Collision avoidance
  - Swarm coherence

Mobility algorithms

- Steering Vector is the **desired movement vector** for the next iteration

[2] Reynolds, C. W., "Steering behaviors for autonomous characters," in *Game developers conference,* San Francisco, California 1999

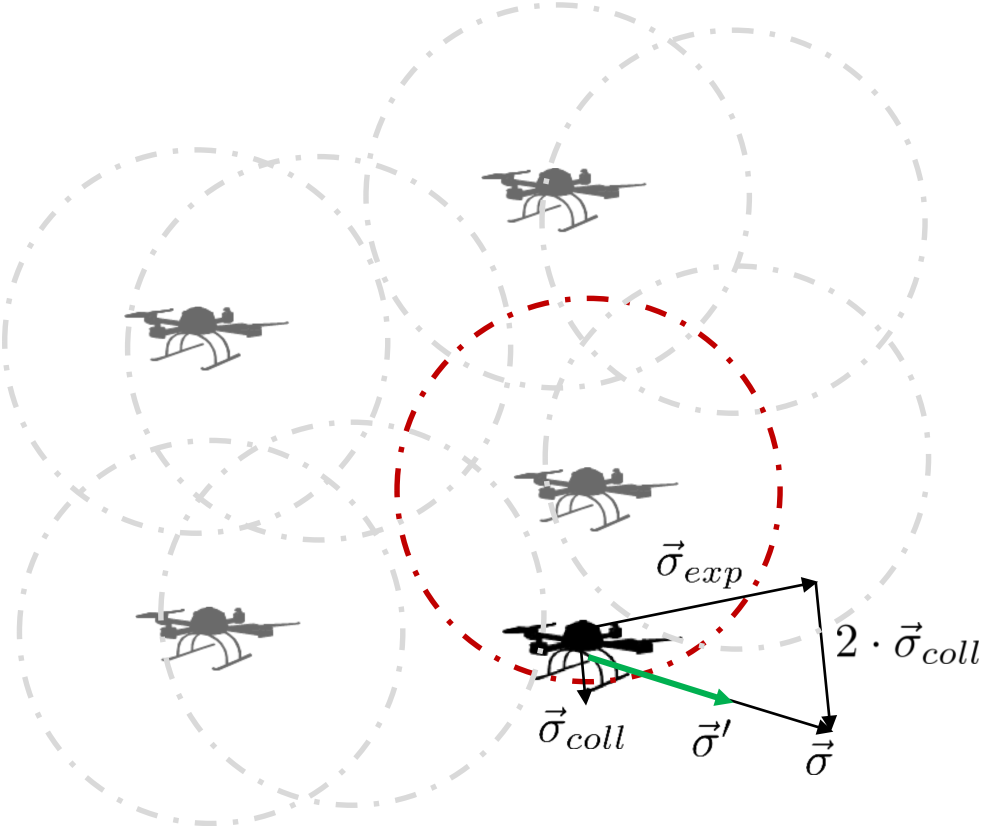
Layers of the Reynolds Mobility Meta-model

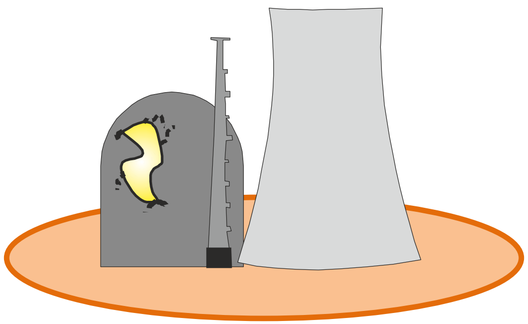


- Maps the desired movement vector to a **travelled vector**
- Physical realization of the movement
  - Simulation: calculation of the next position
  - Real vehicle: motor control
  - Acceleration and braking
- Implementation of the **vehicle type**
- Increases portability of the mobility algorithms / steerings

[2] Reynolds, C. W., “Steering behaviors for autonomous characters,” in *Game developers conference,* San Francisco, California 1999

Example Behavior of the Reynolds Mobility Meta-modell





Mission area

Steering	Weight
Exploration	1
Collision Avoidance	2

- **Step 1**: Handle the Exploration steering
- **Step 2**: Handle the Collision Avoidance steering
- **Step 3**: Compute the total Steering Vector with the assigned weights
- **Step 4**: Handle the locomotion and limit the movement vector with respect to the physical capabilities of the vehicle

Implementation of a Mobility Algorithm as a Steering

```
class MobilityAlgorithm : public Steering
{
public:
    MobilityAlgorithm();

protected:
    SteeringVector update();
};



SteeringVector MobilityAlgorithm::update()
{
    SteeringVector vector;

    // do fancy stuff
    vector = …

    return vector;
}
```

MobilityAlgorithm.h/.cc

```
module MobilityAlgorithm extends Steering
{
    parameters:

}
```

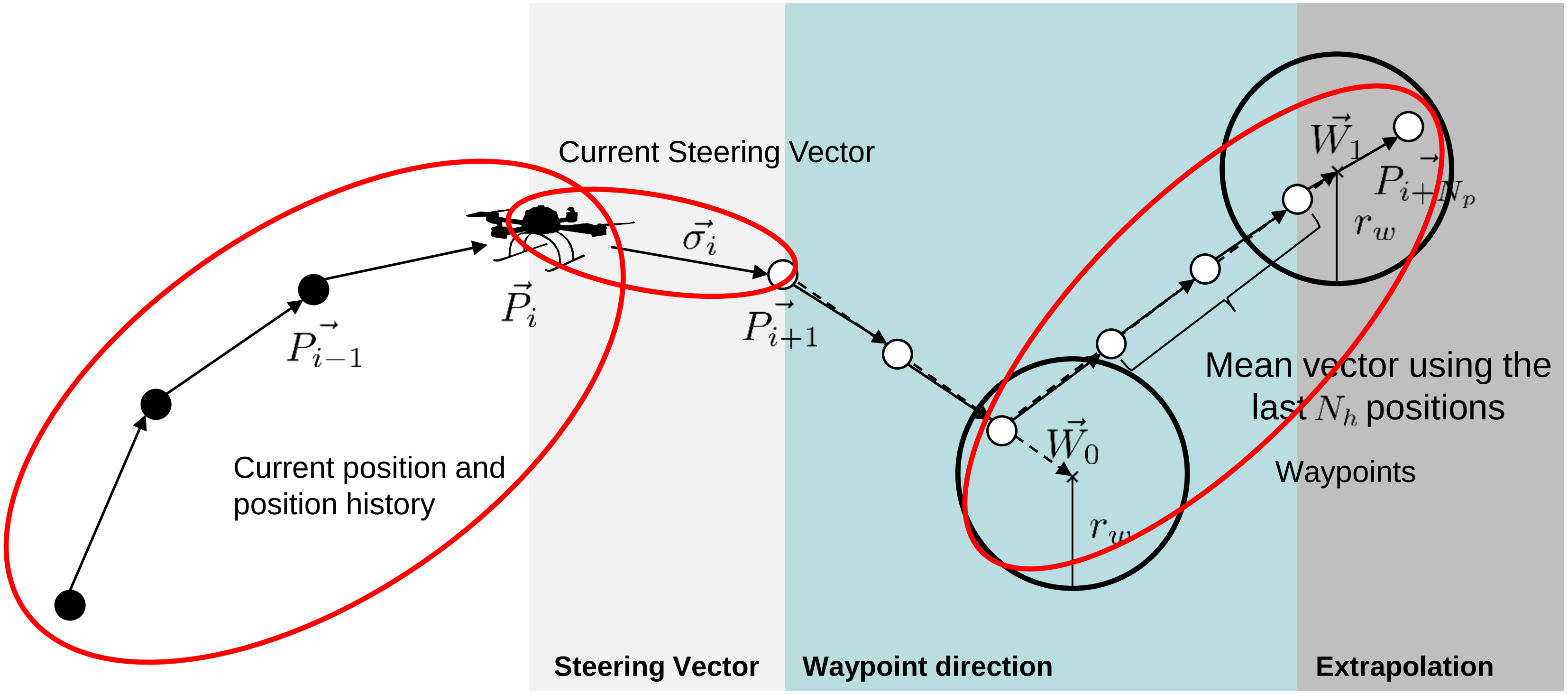
MobilityAlgorithm.ned

```
**.host*.mobilityType = "ReynoldsMobilityModel"
**.host*.numSteerings = 2
**.host*.steering[0].typename = "MobilityAlgorithm"
**.host*.steering[0].weight = 2
**.host*.steering[1].typename = "AnotherMobilityAlgorithm"
**.host*.steering[1].weight = 1
**.host*.locomotion.typename = "UAVLocomotion"
```

omnetpp.ini

Mobility Prediction

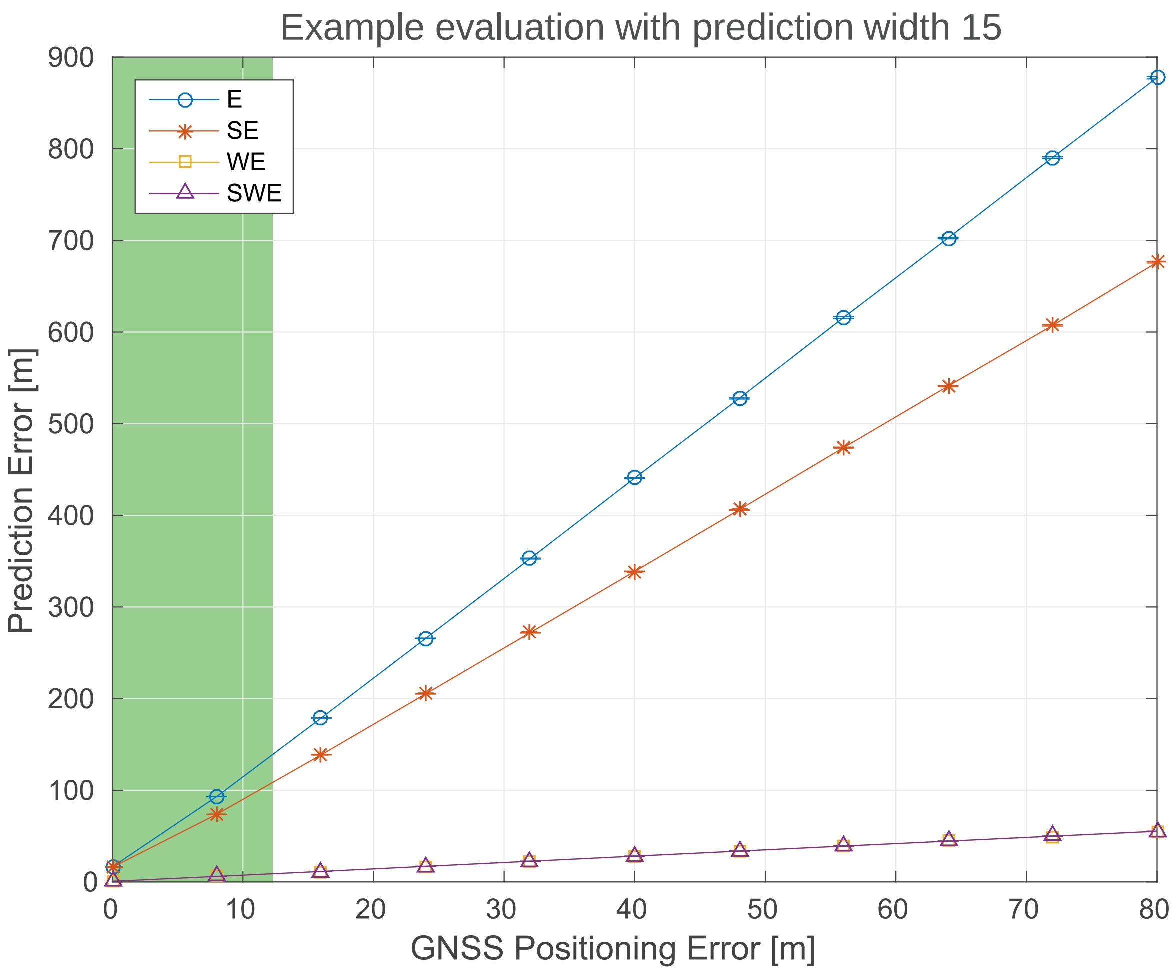
Leveraging Mobility Control Knowledge for Precise Position Predictions



- Iterative method to predict the position at a defined target iteration (here $N_p = 7$)
- Usage of the most precise available mobility information in each step

[3] B. Sliwa, D. Behnke, C. Ide, C. Wietfeld, "B.A.T.Mobile: Leveraging Mobility Control Knowledge for Efficient Routing in Mobile Robotic Networks", *In IEEE GLOBECOM 2016 Workshop on Wireless Networking, Control and Positioning of Unmanned Autonomous Vehicles (Wi-UAV)*, Washington D.C., USA, Dezember 2016, accepted for presentation. [Online]. Available: http://arxiv.org/abs/1607.01223

Accuracy of the Prediction Method



- Steering Vector integration decreases the prediction error
- Waypoints act as static orientation points
- Steering Vector is only beneficial if waypoints are not available

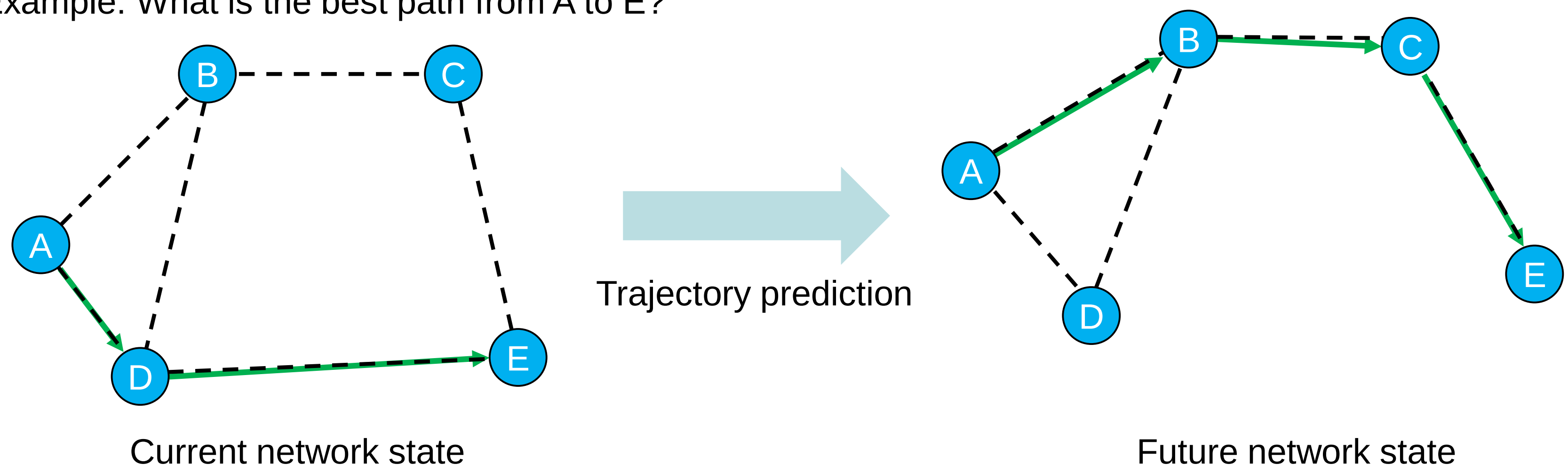
W: Waypoints
S: Steering Vector
E: Extrapolation
GNSS: Global Navigation Satellite System

The prediction stability can highly be increased by integrating waypoint information

Mobility-aware Routing

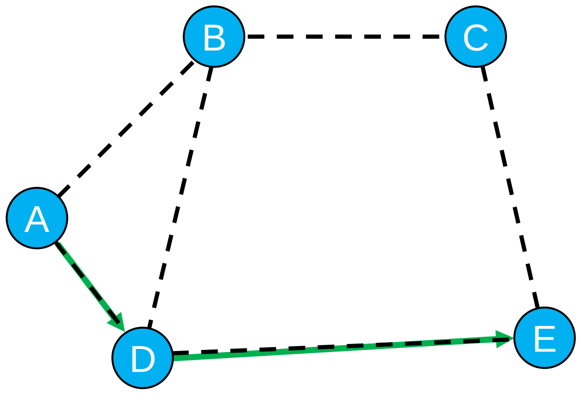
Predictive Path Planning with Mobility-aware OLSR

Example: What is the best path from A to E?



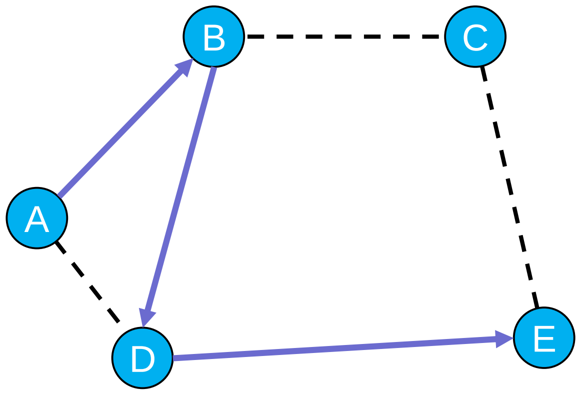
- Trade-off: shortest path vs stability
- Choose the shortest path with the highest availability
- Requires mobility information of **all nodes** → high overhead

Stigmergic Approach with B.A.T.Mobile



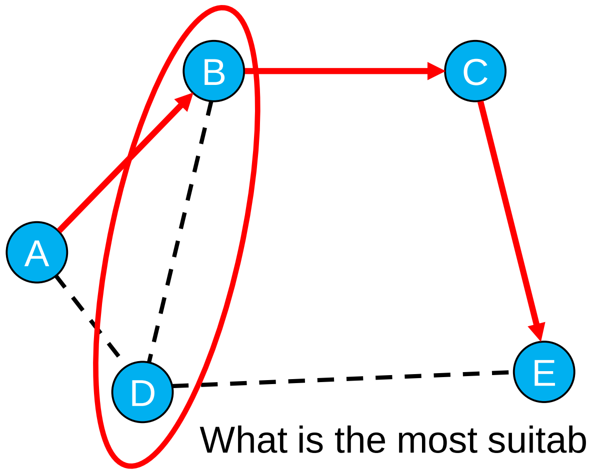
What is the best path from A to E?

Stigmergic Approach with B.A.T.Mobile



What is the best path from A to E?

Stigmergic Approach with B.A.T.Mobile

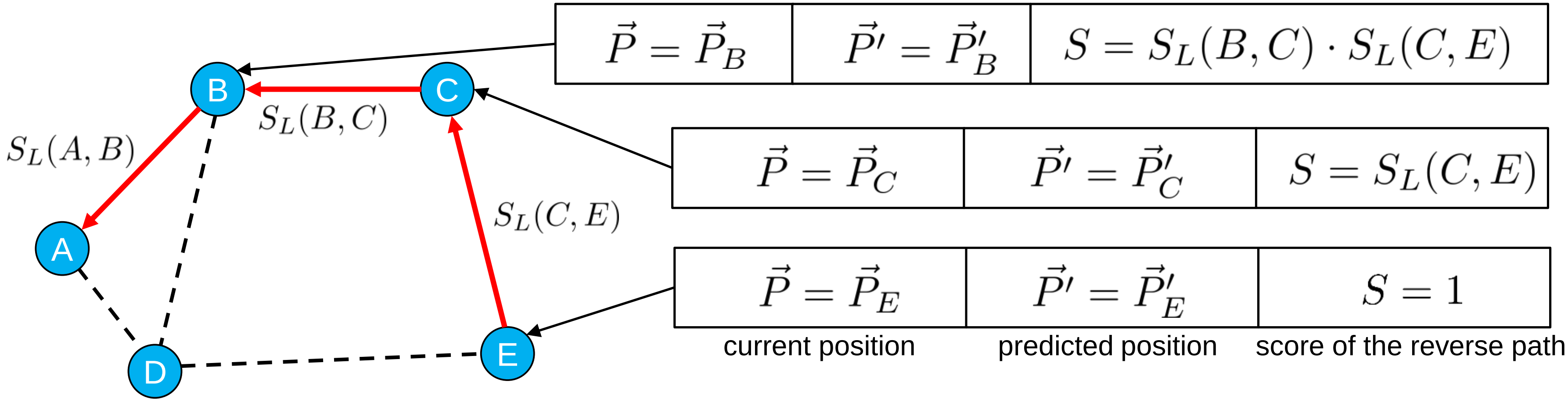


~~What is the best path from A to E?~~

What is the most suitable forwarder for A to E?

- Periodic flooding of routing messages → data is received on multiple paths
- Include a measurement for **the path quality of the reverse path** to the sender

Stigmergic Approach with B.A.T.Mobile



$$S_{C|B} = S_L(A,B) \cdot S_L(B,C) \cdot S_L(C,E)$$

- Periodic flooding of routing messages → data is received on multiple paths
- Include a measurement for **the path quality of the reverse path** to the sender

- Intermediate nodes update the score with information about the link quality to the forwarder

$$S_{new} = S_{received} \cdot S_L$$

- Link score takes current distance and **predicted distance development** into account

- Routing decision: choose the forwarder with the best path score to the destination

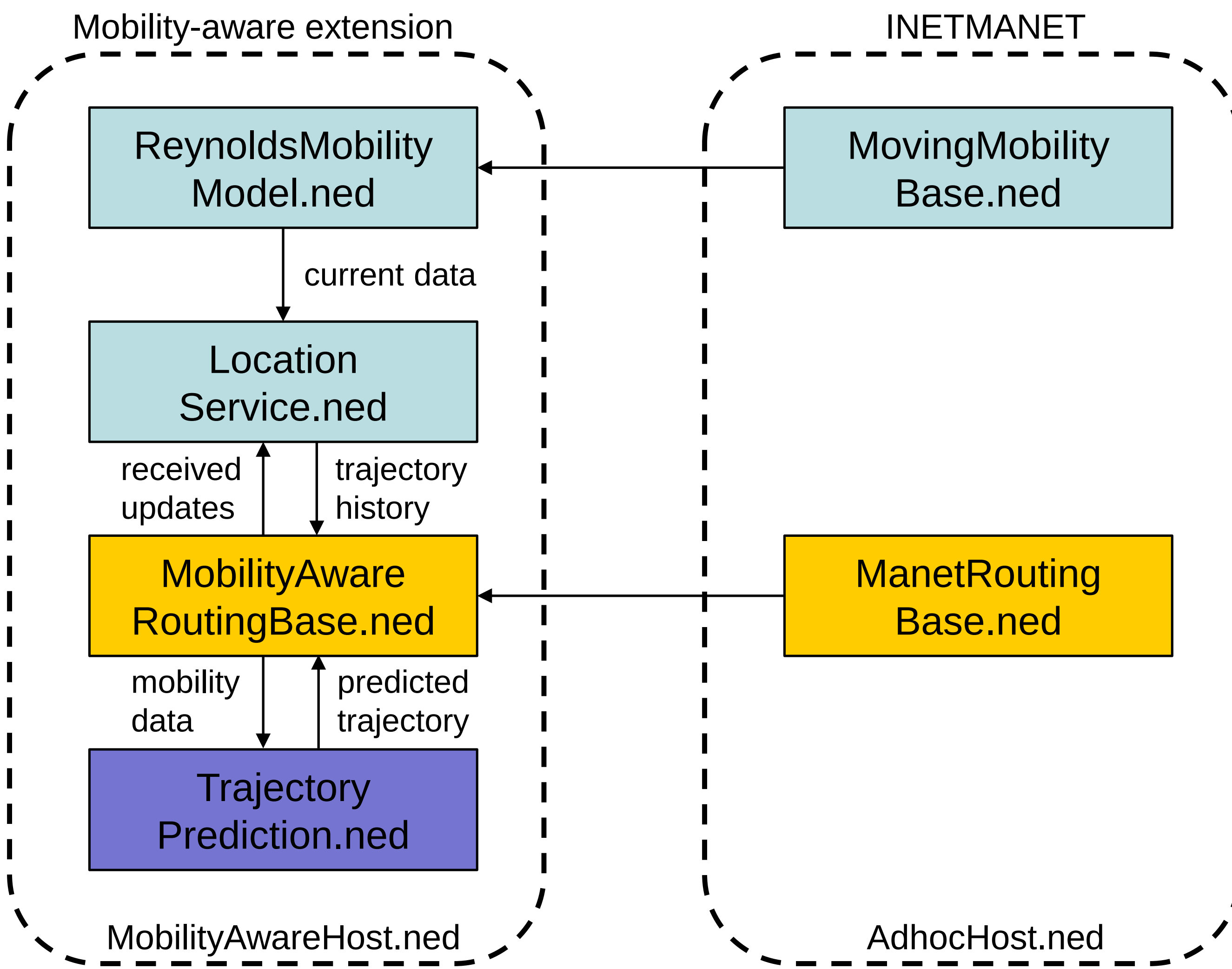
Implementation in OMNeT++/INETMANET

Mobility control

Mobility data access

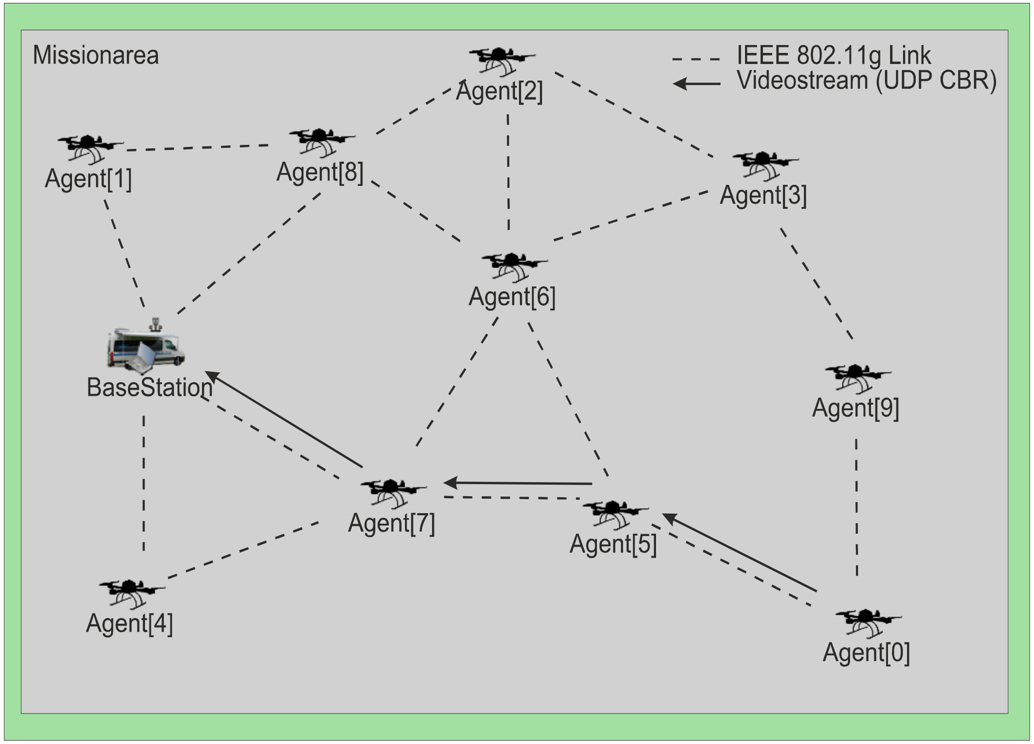
Routing protocols

Prediction method



Proof of Concept Evaluation

Simulation Parameters and Reference Scenario

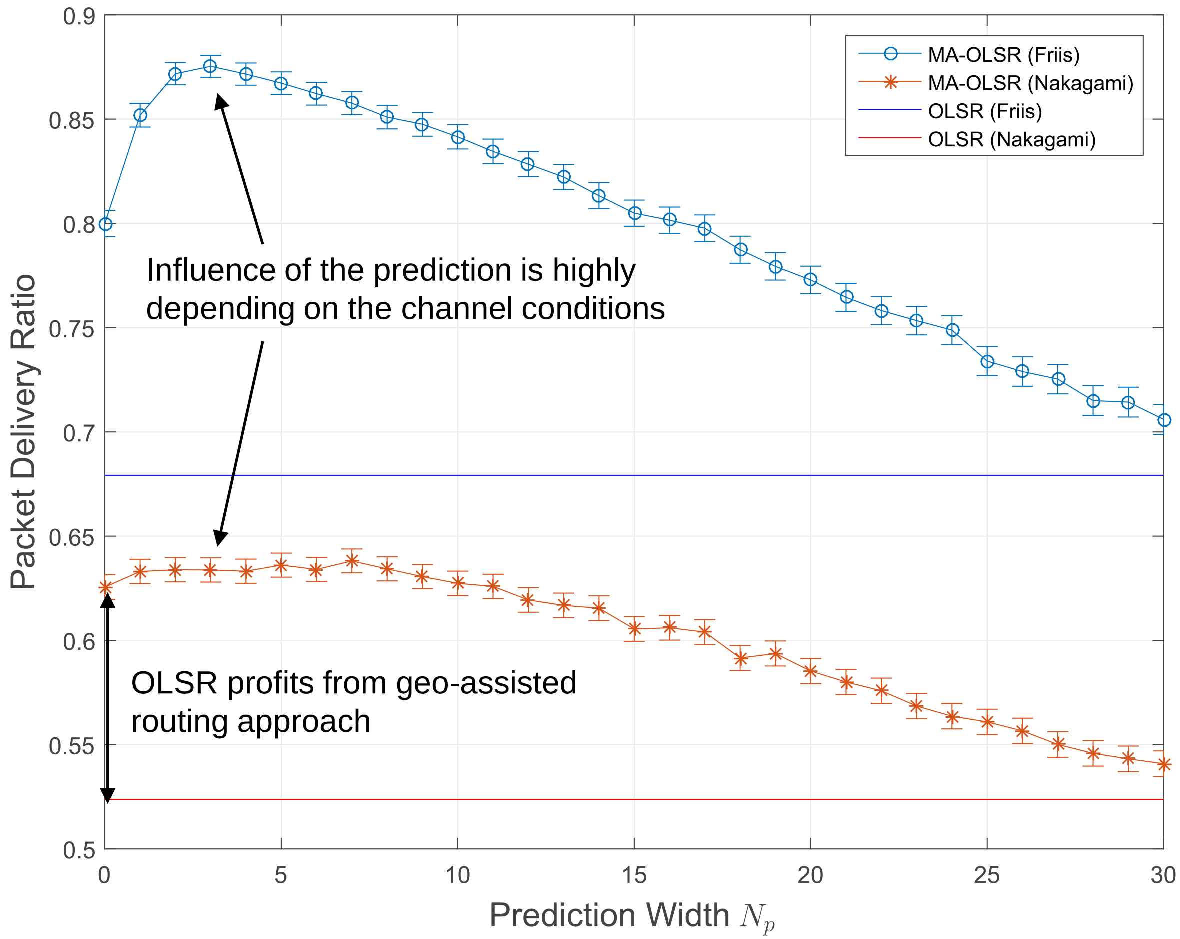


Simulation parameter	Value
Mission area	500 m x 500 m x 250 m
Number of agents	10
Steering[0] (Exploration)	Controlled Waypoint
Steering[0].weight	1
Steering[1] (Collision Avoidance)	LocationBasedCollAvoidance
Steering[1].weight	10
Steering[1].minDistance	30 m
Locomotion	UAVLocomotion
Mobility update interval Δt_u	250 ms
Movement speed	50 km/h
Channel model	Friis / Nakagami ($\alpha = 2.75$)
Video stream bitrate	2 Mbit/s
Video stream packet size	1460 Byte

- Which approach achieves a higher benefit from mobility-awareness?
- How do the channel conditions influence the routing performance?

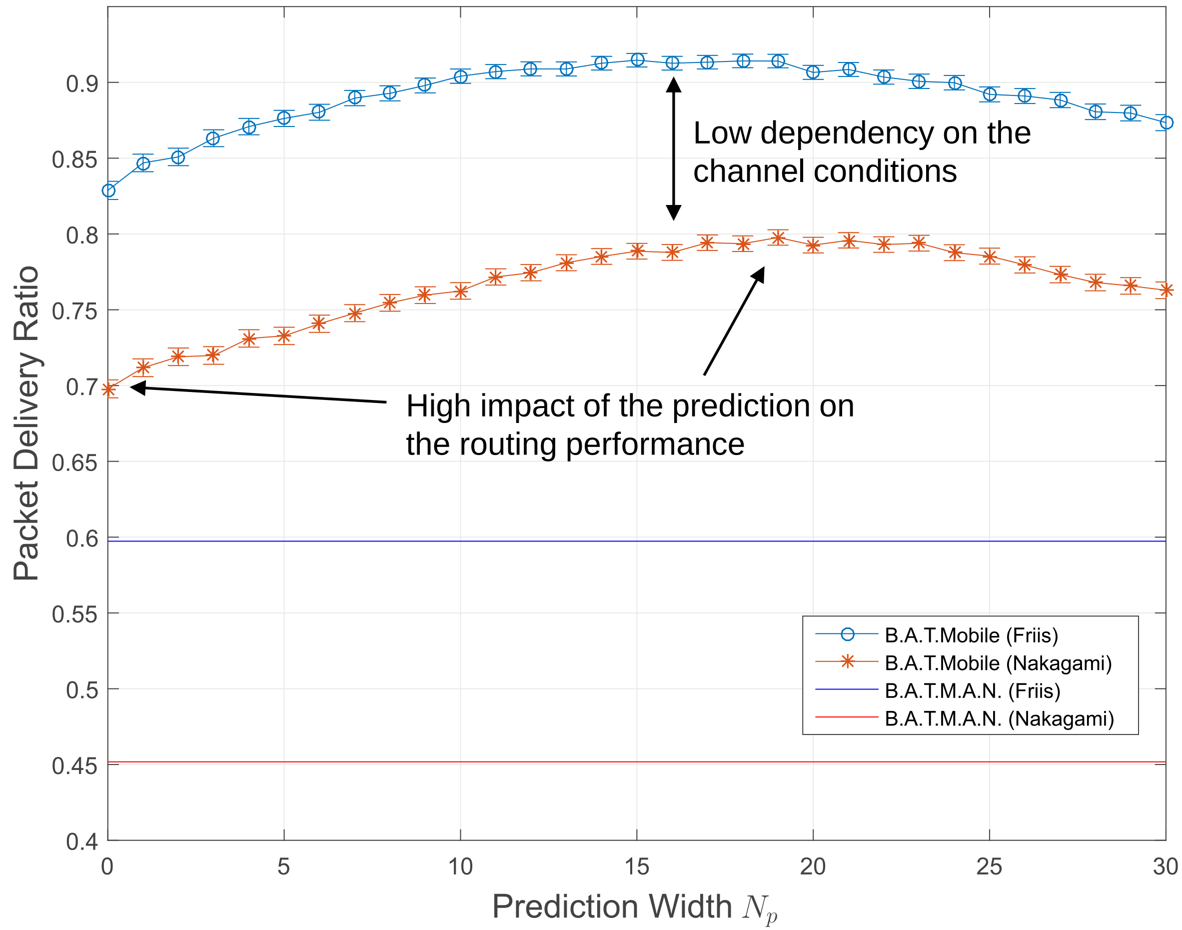
CBR: Constant Bitrate

Predictive Path Planning with Mobility-aware OLSR



Predictive Path Planning is only useful in envrionments with low packet loss probabilities

Stigmergic Approach with B.A.T.Mobile



Stigmergic approach achieves a high robustness against challenging channel conditions

Conclusion

- Reynolds meta-model for mobility behavior of autonomous agents
  - Steerings implement mobility algorithms
  - Locomotion represents the vehicle class
- Trajectory prediction method
  - Leverages mobility control knowledge
  - Integration of waypoints highly improves prediction accuracy
- Mobility-aware routing approaches
  - Predictive Path Planning is only useful in environments with low packet loss probabilities
  - Stigmergic approach achieves a high robustness against challenging channel conditions

Mobility-aware extension to the INETMANET framework of OMNeT++
https://github.com/BenSliwa/mobilityaware-inetmanet

Outlook

- Integration of further mobility algorithms as steerings
  - Communication-aware mobility
- Provision of additional vehicle classes as locomotion implementations
- Extension of more routing protocols with mobility-aware capabilities
- Real-world protocol evaluation with experimental testbed

Thanks for your attention