Robust Machine Learning-enabled Routing for Highly Mobile Vehicular Networks with PARRoT in ns-3

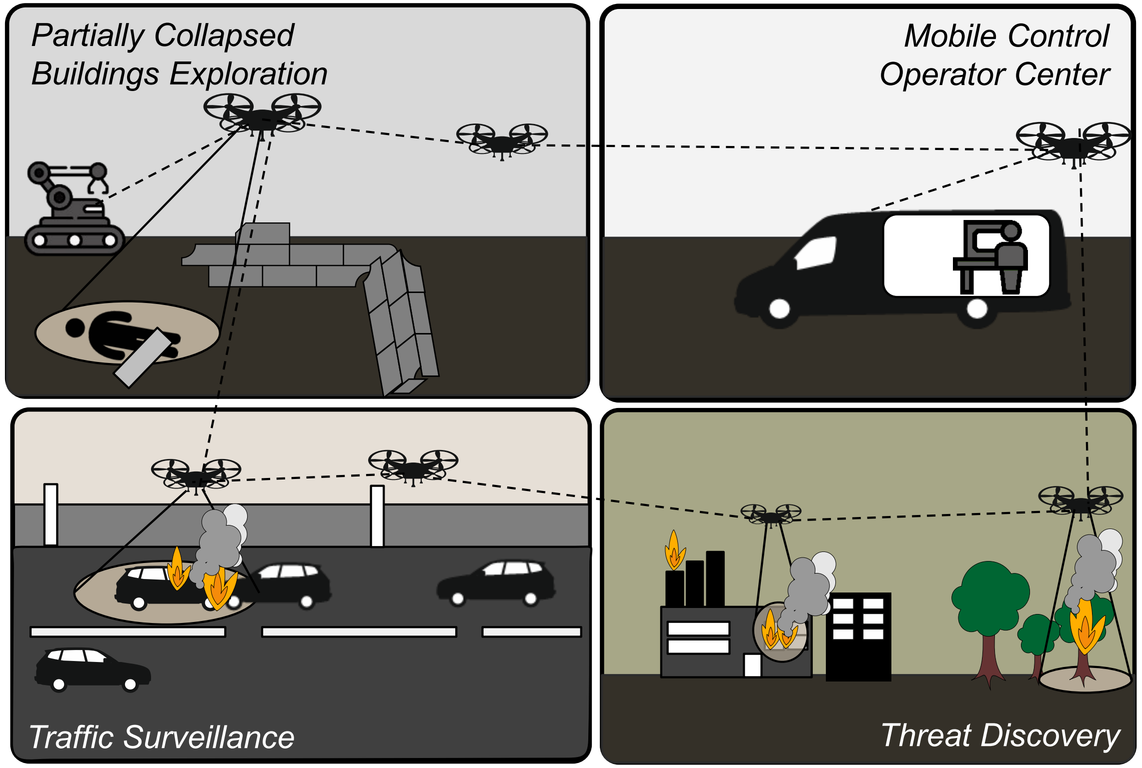
Cedrik Schüler, Manuel Patchou, Benjamin Sliwa and Christian Wietfeld

Virtual Conference: June 24, 2021

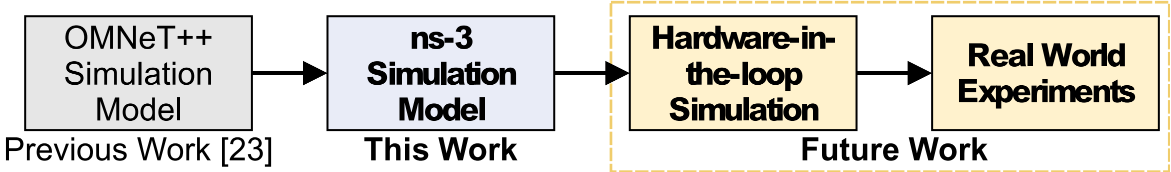
Faculty of Electrical Engineering and Information Technology
Communication Networks Institute
Prof. Dr.-Ing. Christian Wietfeld

Motivation

- Remotely operated or autonomous robotics
- Spontaneously deployable ad-hoc networks
- High mobility → Dynamically changing networks → Fast adaptation needed
- Reinforcement Learning-based protocols omit complex modeling and learn from context features



- Previous work PARRoT [23]
- ns-3 simulation model as preparation for future HiL simulations and real-world experiments



[23] B. Sliwa, C. Schüler, M. Patchou, C. Wietfeld, **“PARRoT: Predictive ad-hoc routing fueled by reinforcement learning and trajectory knowledge”**, In *2021 Vehicular Technology Conference (VTC-Spring)*

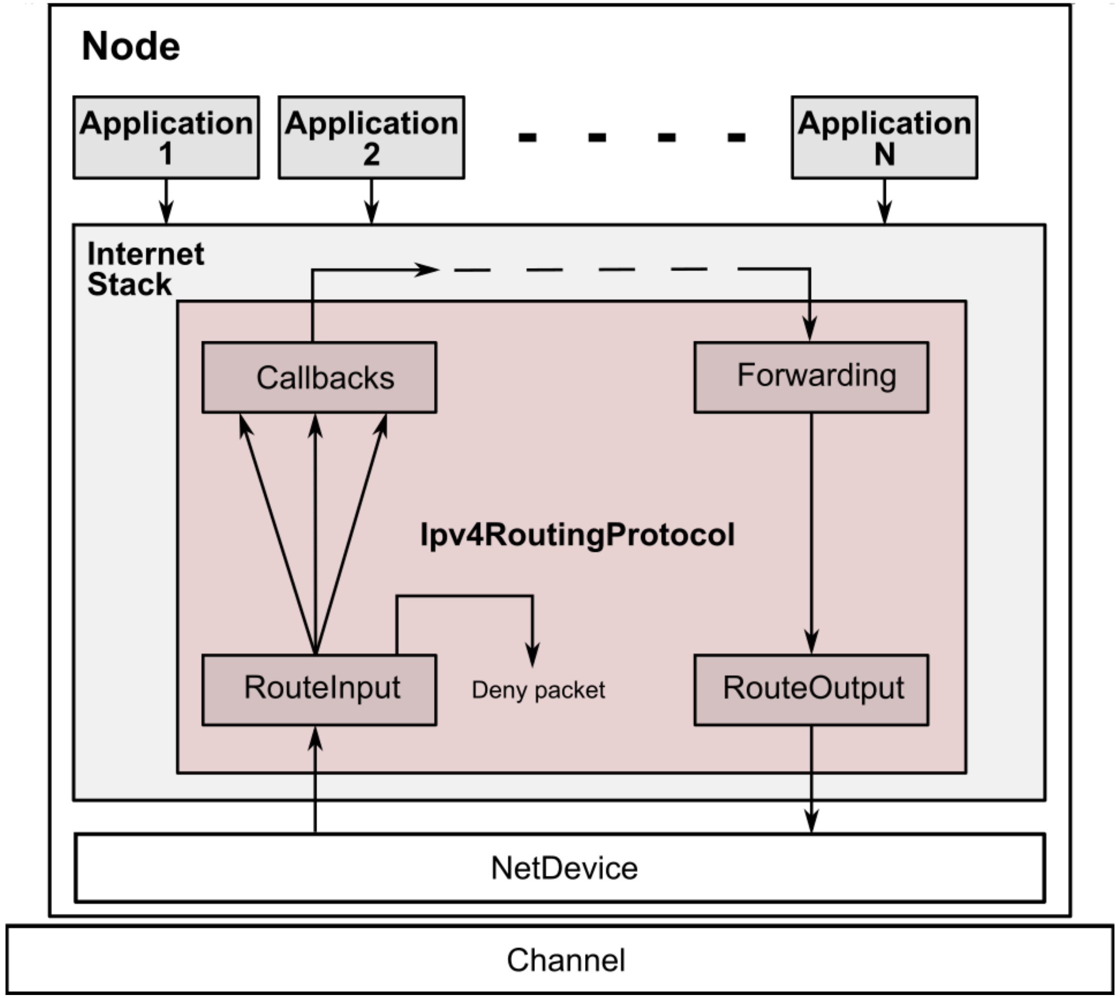
HiL – Hardware-in-the-Loop

Agenda

- Motivation
- Routing Protocol Structure in ns-3
- PARRoT Fundamentals and ns-3 Implementation
- Simulation Setup for Evaluation
- Results
  - Parameter Optimization of PARRoT
  - Consumption Analysis
- Conclusion and Outlook

Routing Protocol Structure in ns-3

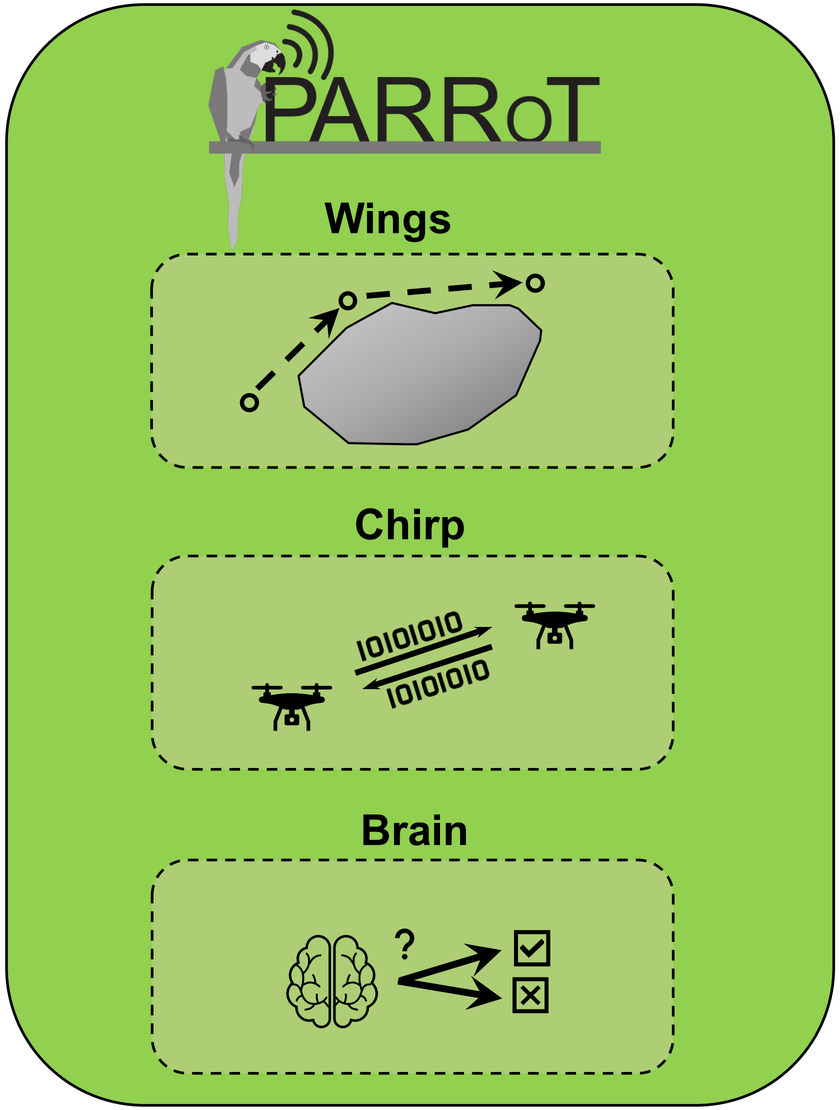
- Subclasses of *ns3::Ipv4RoutingProtocol*
- Located within the internet protocol (IP) stack
- Closely related to Linux' routing functions
- Required implementations:
  - *Notify{InterfaceUp, InterfaceDown, AddAddress, RemoveAddress}*
  - *RouteInput*
    - Processing of incoming packets
  - *Route Output*
    - Route lookup for outgoing packets
- Installed on every *ns3::Node*



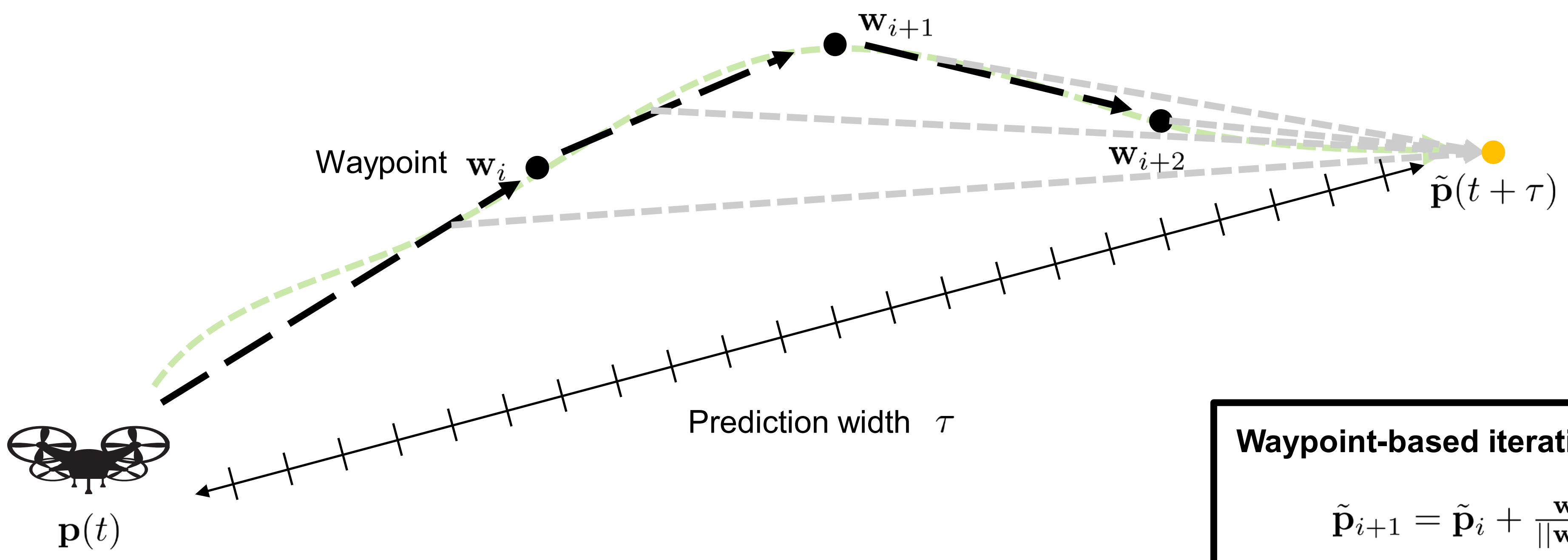
PARRoT Routing Protocol

- **PARRoT:** Predictive Ad-hoc Routing Fueled by Reinforcement Learning and Trajectory Knowledge
- Presented and recipient of the best student paper award at VTC-Spring 2021

- **Wings:** *Trajectory prediction by leveraging cross-layer knowledge*
- **Chirp:** *Cooperative distribution of periodic information to maintain a decentralized network architecture*
- **Brain:** *Autonomous forwarding selection based on anticipated topologies and cohesion assessments*



Wings: Trajectory Prediction by Leveraging Cross-layer Knowledge



Waypoint-based iteration:

$$\tilde{\mathbf{p}}_{i+1} = \tilde{\mathbf{p}}_i + \frac{\mathbf{w}_k - \tilde{\mathbf{p}}_i}{||\mathbf{w}_k - \tilde{\mathbf{p}}_i||} \cdot v \cdot \Delta t$$

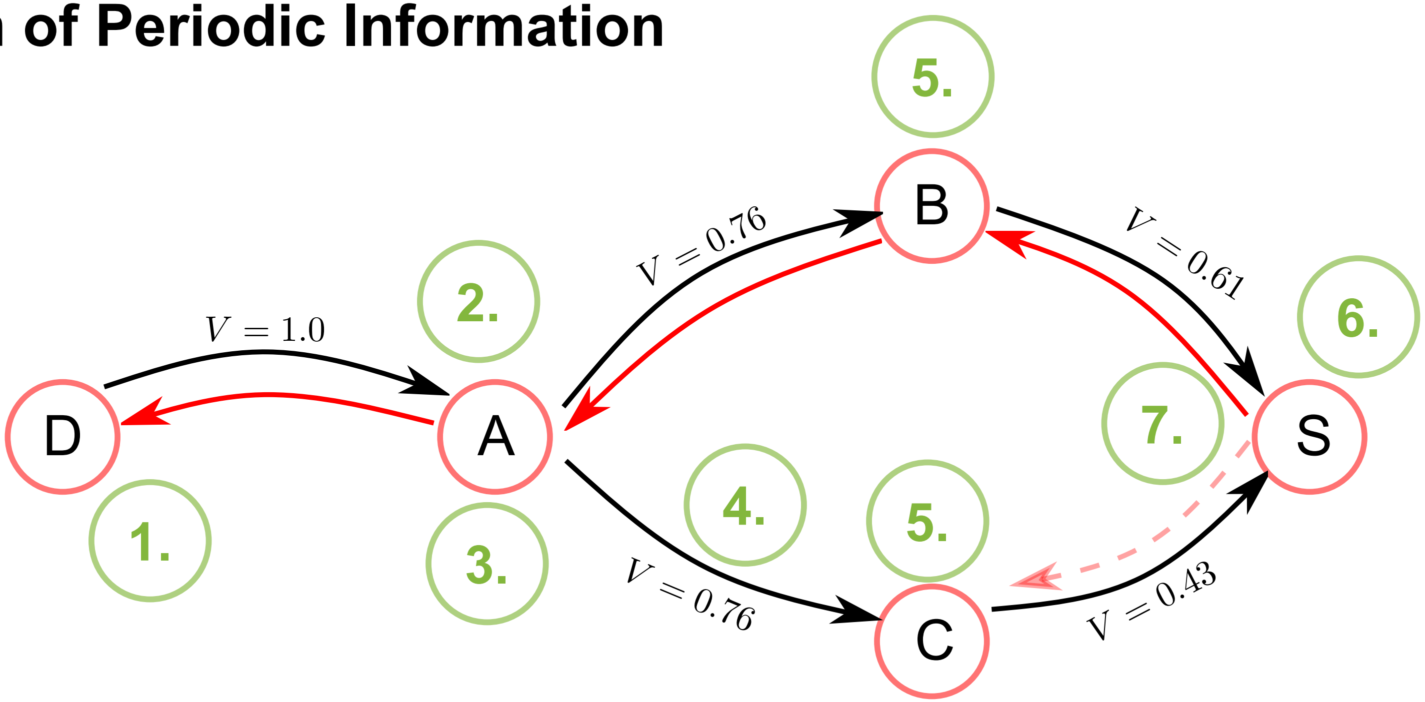
History-based iteration:

$$\tilde{\mathbf{p}}_{i+1} = \tilde{\mathbf{p}}_i + \frac{t_{i+1} - t_i}{N_e - 1} \sum_{j=0}^{N_e - 2} \frac{\mathbf{p}_{i-j} - \mathbf{p}_{i-j-1}}{t_{i-j} - t_{i-j-1}}$$

(*) B. Sliwa et al., **„B.A.T.Mobile: Leveraging mobility control knowledge for efficient routing in mobile robotic networks"**, *IEEE Globecom Workshops (GC Wkshps) 2016*

Chirp: Cooperative Distribution of Periodic Information

1. Chirp initialization
2. Packet processing
3. Reinforcement learning
4. Forward updated chirp
5. Repeat 2. – 4.
6. Reverse route building
7. Greedy hop selection



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Originator Address				SEQ		TTL		Cohesion Φ_{Coh}				Reward V			
$\mathbf{p}.x$				$\mathbf{p}.y$				$\mathbf{p}.z$				$\tilde{\mathbf{p}}.x$			
$\tilde{\mathbf{p}}.y$				$\tilde{\mathbf{p}}.z$											

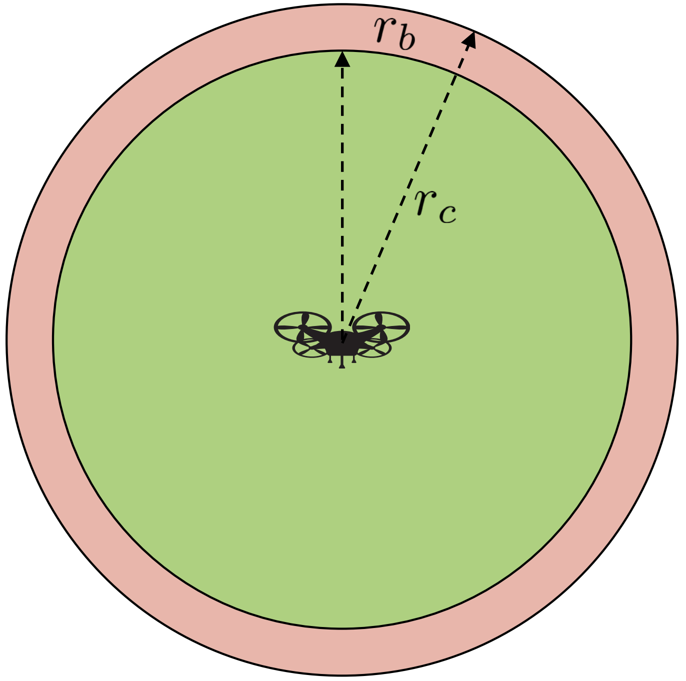
500 ms Δt_u

40 Byte

Brain: Autonomous Forwarding Selection

- Routing tables built through Q-Learning
- Metric calculation for destination ***d*** via neighbor ***j***
- Refrain from direct rewards to reduce communication overhead
- Learning from long-term reward
- Multi-metric approach
  - Link expiry time
  - Neighbor cohesion

d	j	$Q(d,j)$
D	**C**	**0.79**
D	B	0.77
⋮	⋮	⋮
S	S	0.86



$$Q(\overset{\text{Destination}}{\boxed{d}},\underset{\text{Neighbor}}{\boxed{j}}) = Q(d,j) + \overset{\text{Learning rate}}{\boxed{\alpha}}\overbrace{\left[\underset{\text{Reward}}{\boxed{R(d,j)}} + \overset{\text{Neighbor-specific metric}}{\boxed{\gamma(j)}}\underset{\text{Long-term reward}}{\boxed{\max_{j'\in V_i} Q(d',j')}} - Q(d,j)\right]}^{\text{Temporal difference}}$$

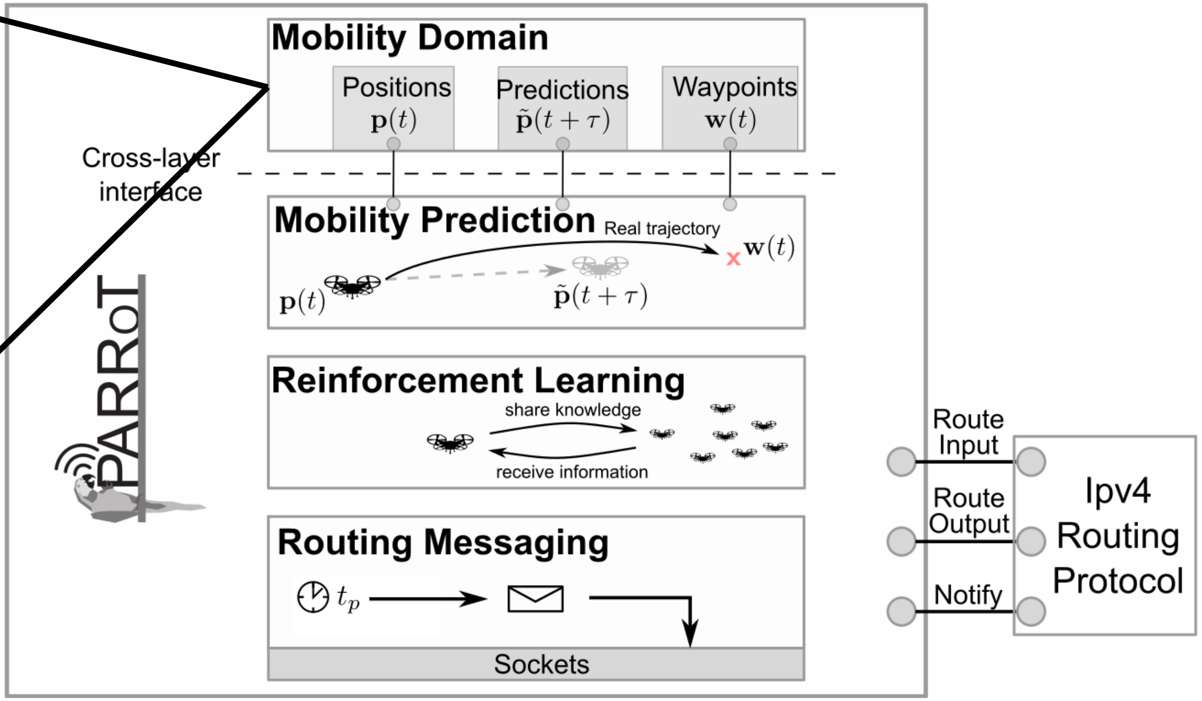
(*) G. Oddi et al., **„A proactive link-failure resilient routing protocol for MANETs based on reinforcement learning“**, 20th Mediterranean Conference on Control & Automation (MED), Barcelona, 2012

Key Implementation Aspects of PARRoT in ns-3

```
std::deque<Vector>
ControlledRandomWaypointMobilityModel::gcWP (int n)
{
  std::deque<Vector> res;
  for (int i = 0; i < std::min (n, (int) m_buffer.size ()); i++)
    {
      res.push_back (m_buffer[i]);
    }
  return res;
}
```

Returns up to n available future waypoints

Buffered Waypoints



Key Implementation Aspects of PARRoT in ns-3

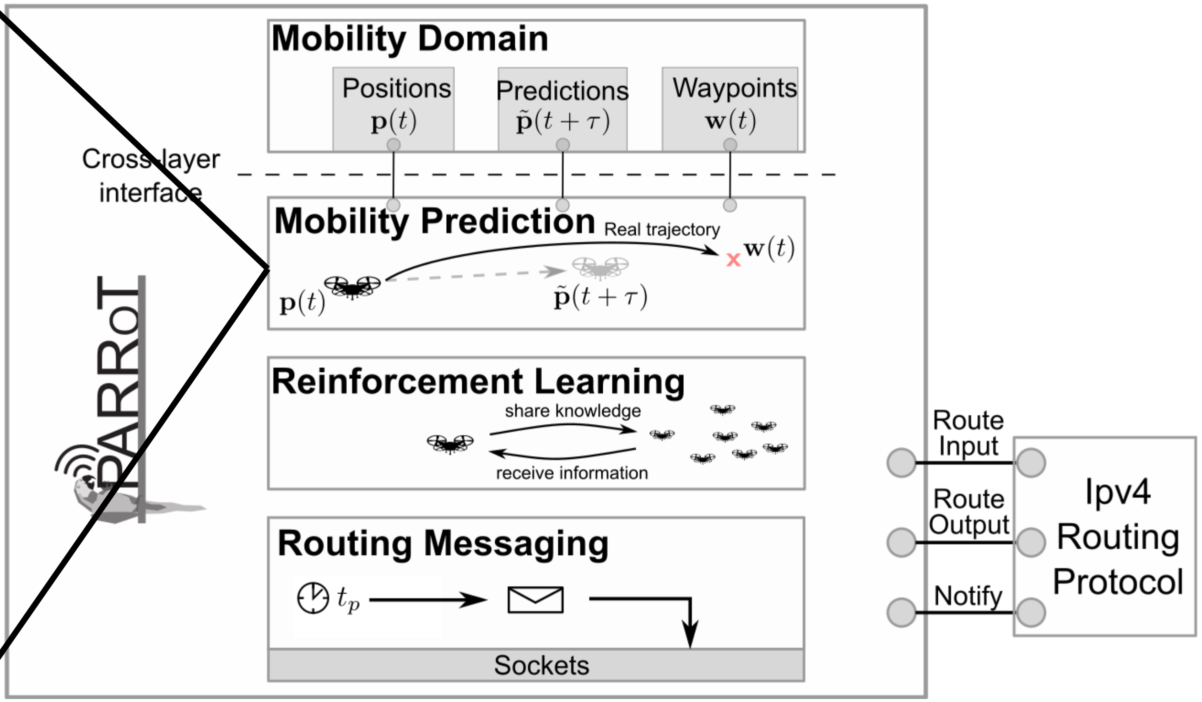
```
for (int i = 0; i < (int) floor (m_tau.GetSeconds () /
                                    m_tp.GetSeconds ());
     i++)
  {
    time_ms += m_updateInterval_ms;
    if (plannedWaypoints.size () > 0)
      {
        currentData =
            predictWithTarget (currentData, m_updateInterval_ms,
                                     plannedWaypoints[0]);
      }
    else
      {
        currentData = predictWithHistory (history, hist_times,
                                   time_ms);
      }
    predictions.push_back (currentData);
    history.push_back (currentData);
    history.pop_front ();
  }
return predictions[predictions.size () - 1];
```

Number of iteration steps

Predict with next Waypoint if possible

Predict with history as fallback

Return latest predicted position



Key Implementation Aspects of PARRoT in ns-3

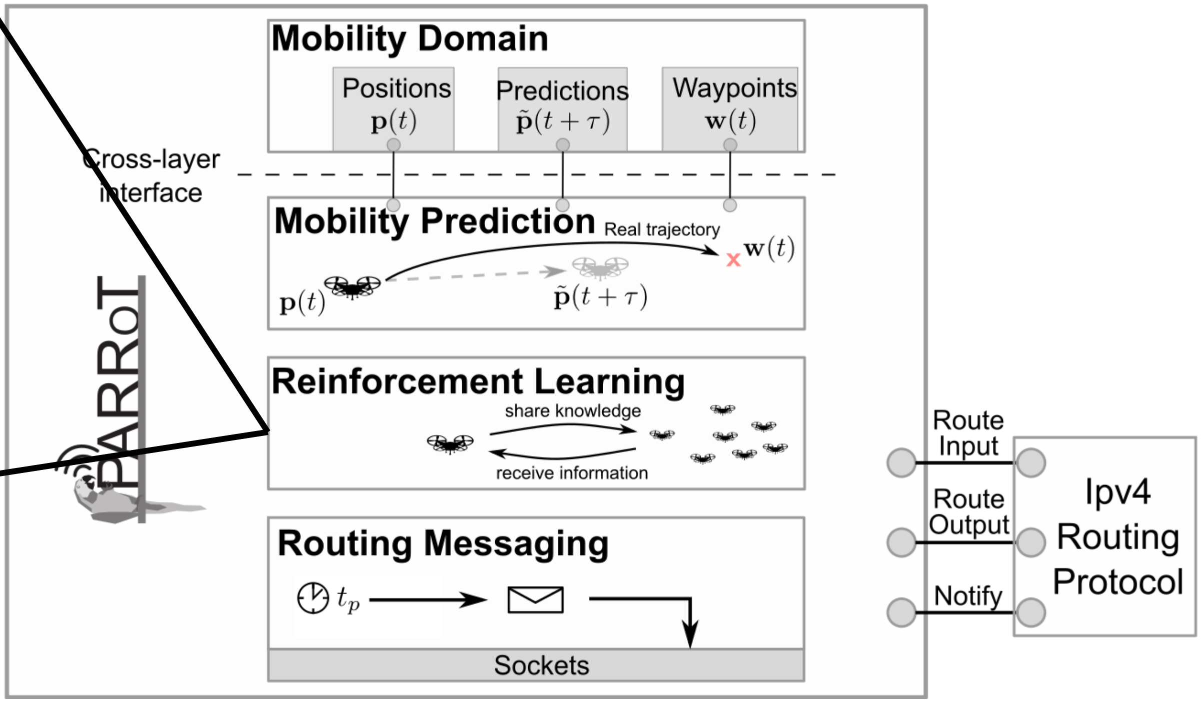
Depends on target node and possible next hop

```cpp
double
RoutingProtocol::qFunction (Ipv4Address target, Ipv4Address hop)
{
  std::vector<double> discounts;
  discounts.push_back (m_basicDiscount);
  discounts.push_back (Phi_LET (hop));
  discounts.push_back (Vi.at (hop)->Phi_Coh ());

  return (1 - m_alpha) * m_QTable.at (target).at (hop)->Q () +
          m_alpha * (combineDiscounts (discounts) * m_QTable.at
                                 (target).at (hop)->V ());
}
```

Multi-metric approach

Calculation of Q-Value



Key Implementation Aspects of PARRoT in ns-3

```
RoutingProtocol::SendMultiHopChirp ()
{
  MultiHopChirp chirp;

  Vector3D p = mobility->GetPosition ();
  trackPosition (p);
  Vector3D p_hat = predictPosition ();

  chirp.SetOrig (m_selfIpv4Address);
  chirp.SetP (p);
  chirp.SetP_hat (p_hat);

  chirp.SetV (1.0);

  updatePhi_Coh ();
  chirp.SetPhi_Coh (m_Phi_Coh);

  chirp.SetSeq (m_squNr);
  m_squNr++;

  chirp.SetTtl (m_maxHops);

  // ... Serialize, add header to packet and send out
}
```

Create object

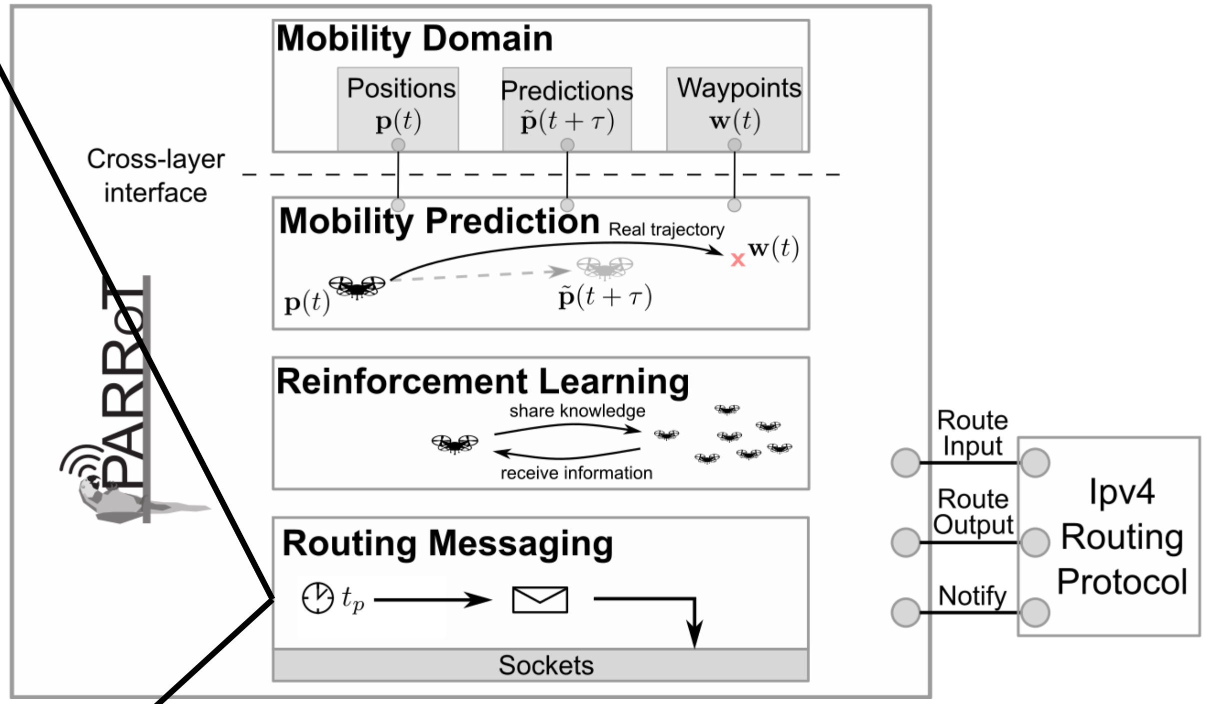
Mobility management

Set identification and position data

Initialize with maximum long-term reward

Update and set remaining fields

Further processing

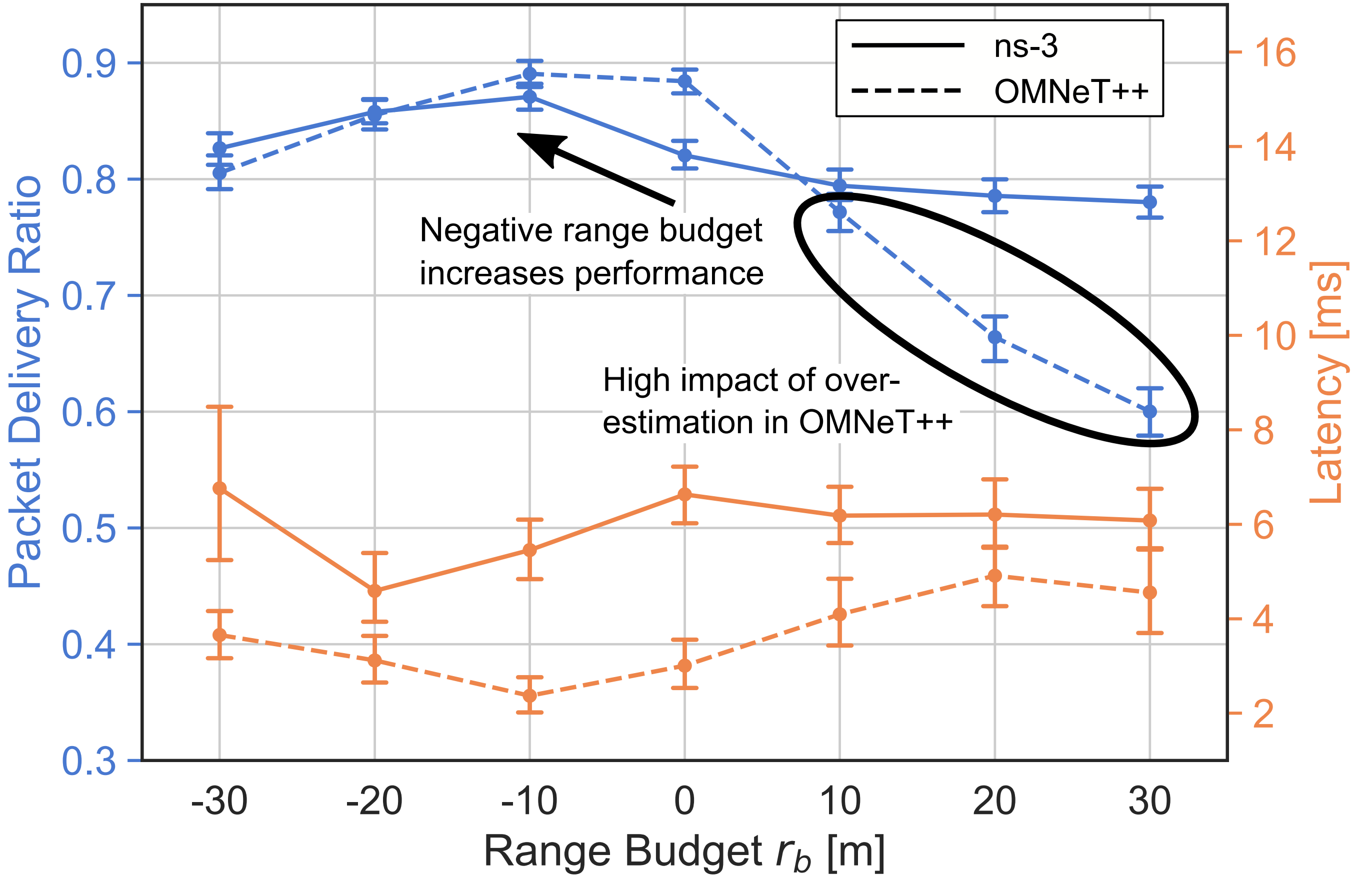


Methodology

- Aim to reproduce OMNeT++ setup [23]
- 3-dimensional playground with controlled waypoint mobility
- 2 Mbps stream with constant bit rate between two agents
- **Data collection** with *ns3::FlowMonitor*
  - *Packet delivery ratio*
  - *End-to-end latency*
- **Consumption analysis** triggered by external Python tool
  - Output and logging disabled
  - Execution time
  - Memory consumption monitored using *pidstat*

Parameter	**Value**
Runs	25
Simulation time	900 s
Start phase duration	5 s
MAC	802.11g
Bit Error model	NistErrorModel
Noise Figure	0 dB
Rate Control	IdealRateControlManager
Transmission power	20 dBm
Receiver sensitivity	-85 dBm
Channel model	Friis ($\eta = 2.75$)
Mobility model	Controlled waypoint
Playground size	500 m x 500 m x 250 m
Number of hosts	10
Speed	50 $\frac{km}{h}$
Traffic	UDP constant bit rate (2 Mbps)

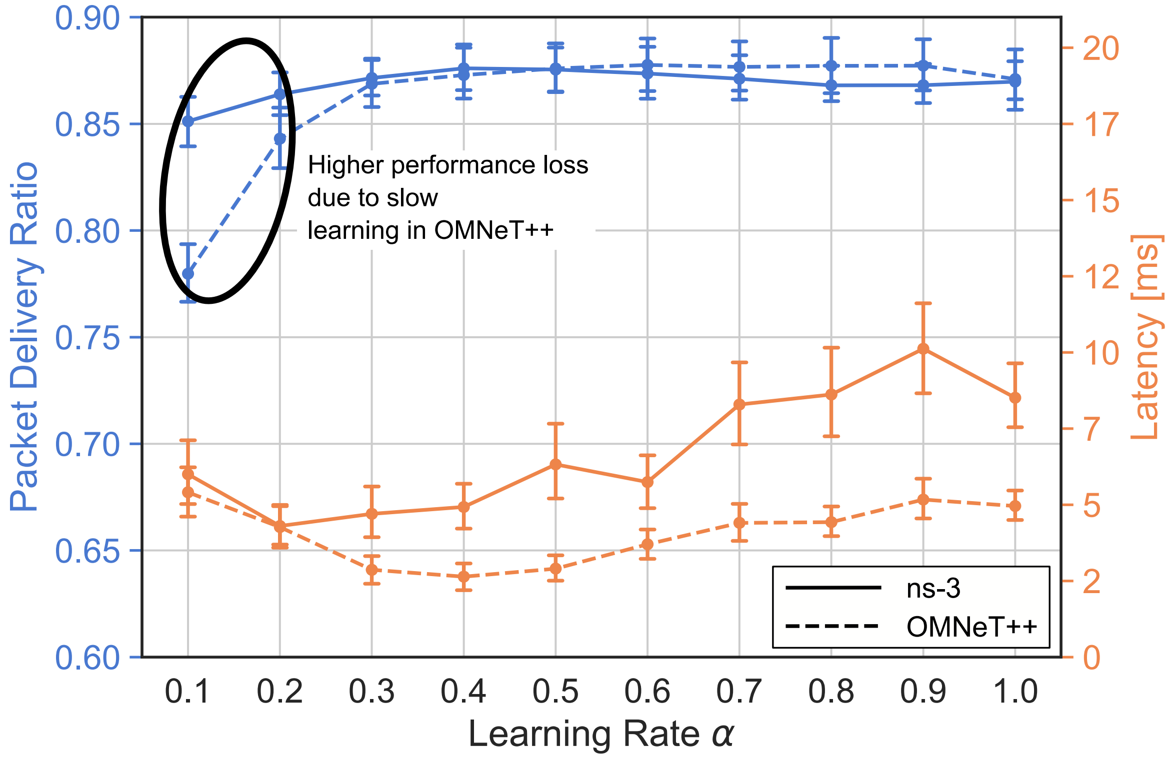
Sensitivity Analysis of Assumed Communication Range

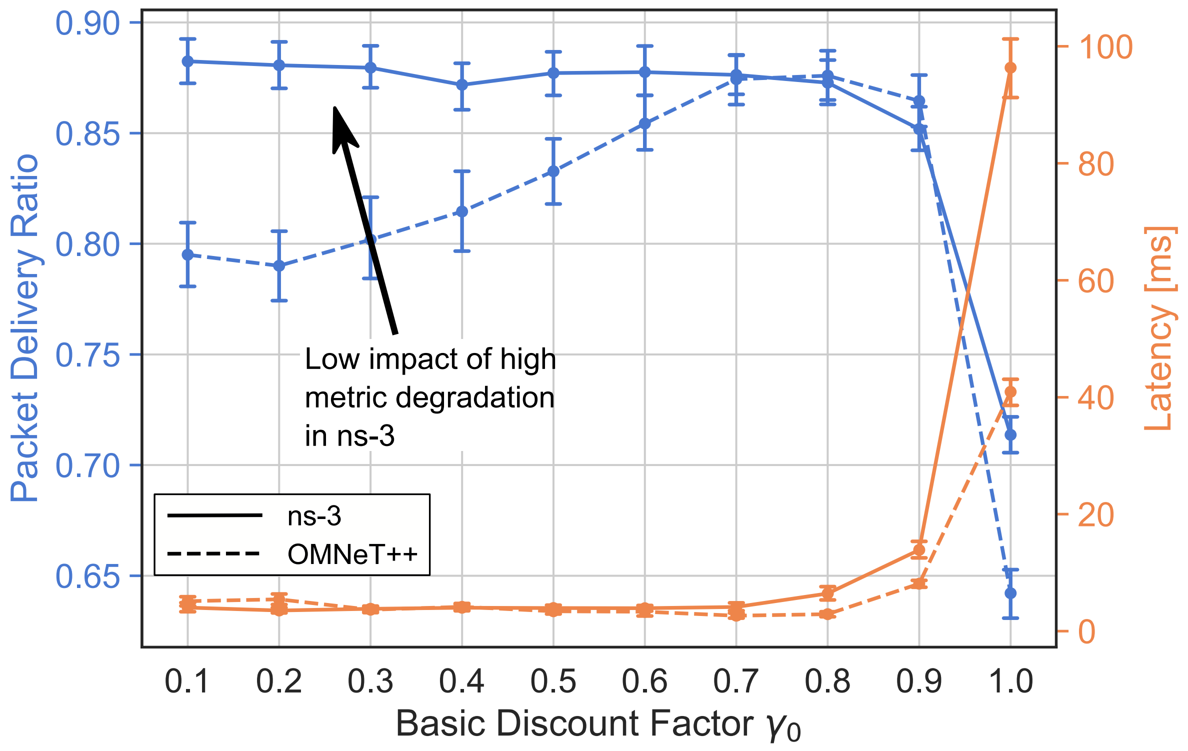


- A range of $r_c \approx 230m$ is assumed
- Applying a range budget of $-30m$ leads to a range estimation of $r_c \approx 200m$
- By range overestimation, links are falsely assumed to be available

Range overestimation leads to performance drains

Optimization of the Reinforcement Learning Parameters

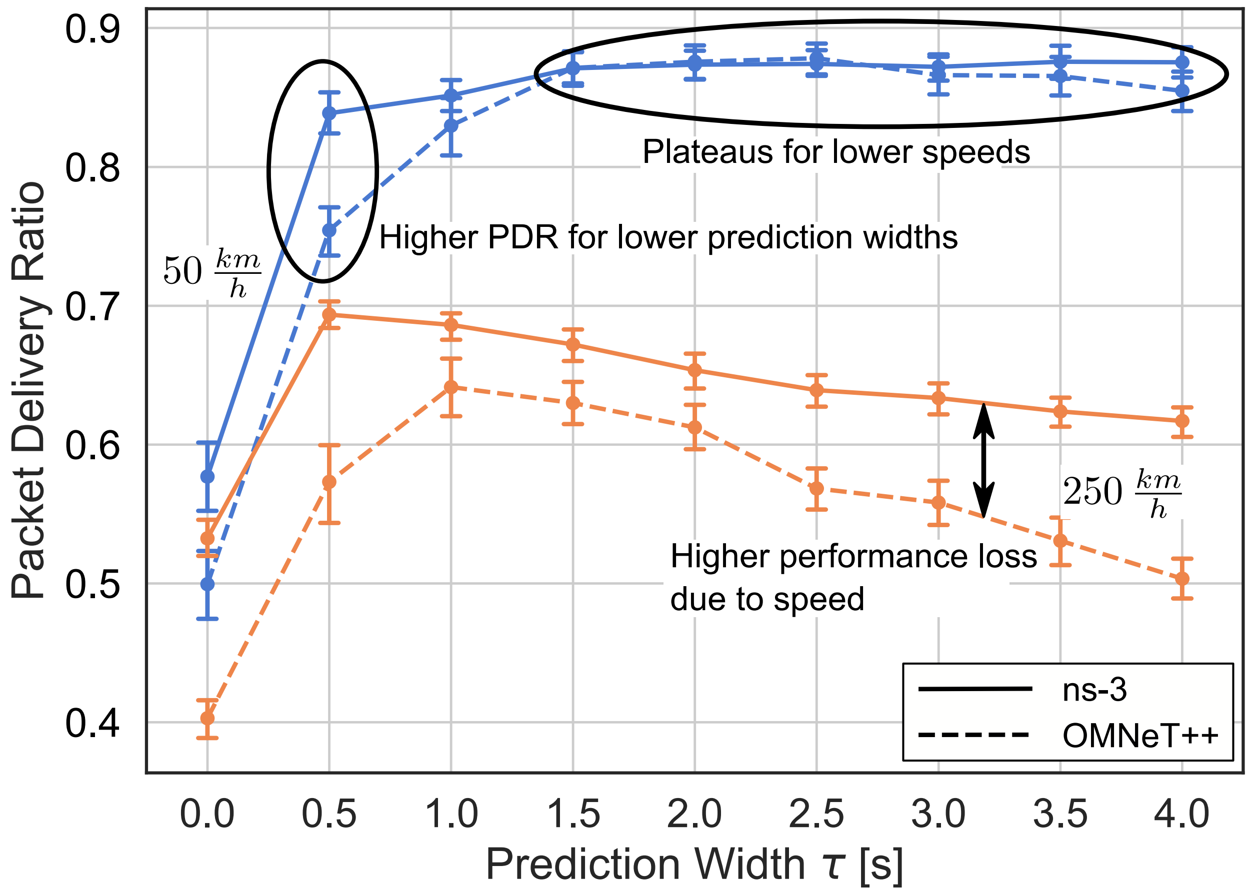




Similar hyperparameters for reinforcement learning components

Prediction Widths for Different Speed Profiles

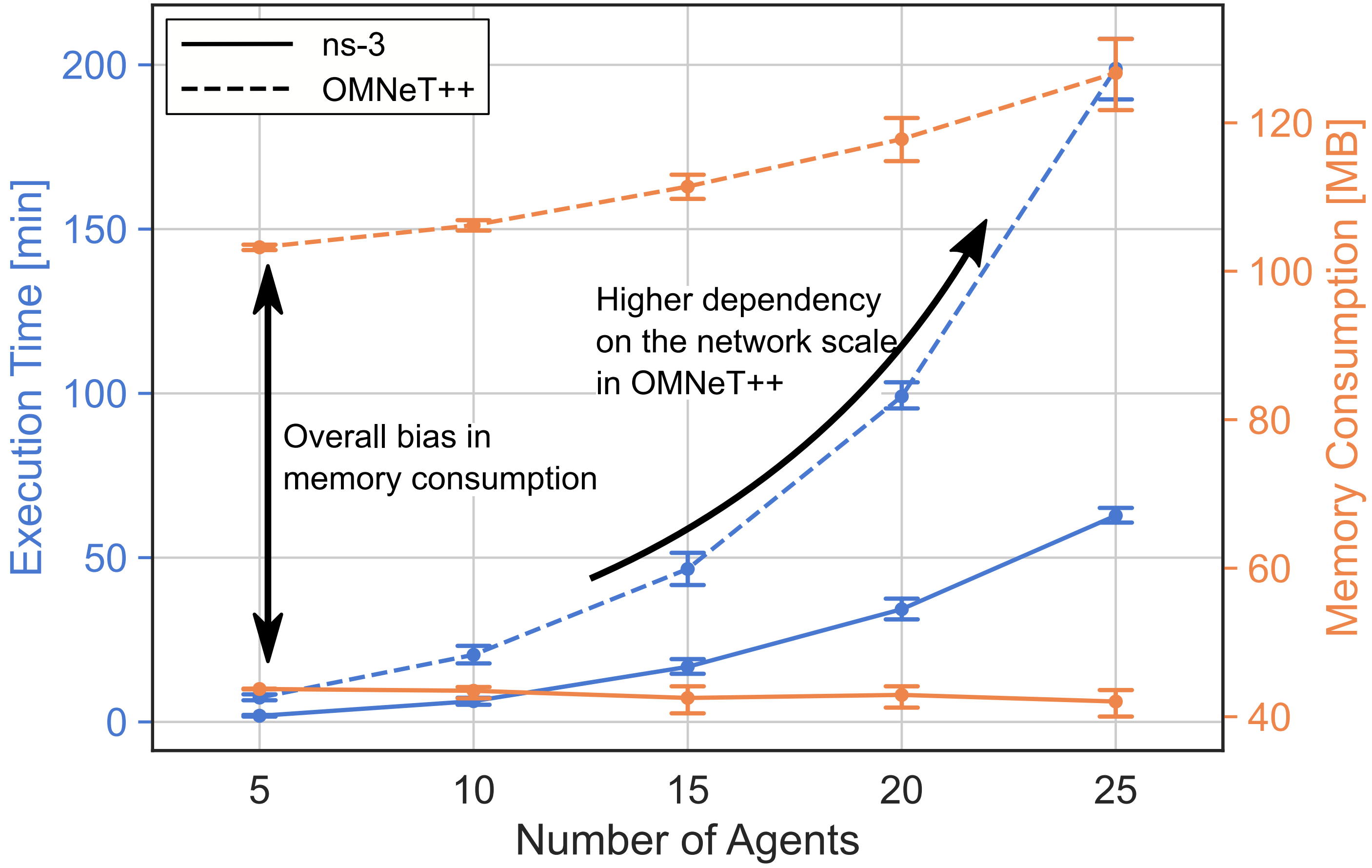
- Prediction width τ represents:
  - Not only the width of mobility prediction
  - But also the time constraint for routing entries
- Different speed profiles considered: $50km/h$ and $250km/h$
- Higher agent speed decreases reliability
- Early plateauing for smaller τ



Suitable prediction width is function of movement speed

Analysis of Time and Memory Consumption for PARRoT Scenarios

- 15 minutes of simulated time
- Scaling up the simulation by increasing agent count
- ➔ More generated events
- Smaller and nearly constant memory usage for ns-3
- Sustainable increase of execution time



Leaner simulation overhead in ns-3

Conclusion and Outlook

Proposal: Implementation of PARRoT in ns-3

- Derived from existing implementation in OMNeT++
- Parameter optimization analysis
- Comparative resource consumption analysis

Future Work

- Hardware in the simulation loop evaluations to further approach real-world performance
- Integration with robotics simulations

Thank you for your attention!

Head of Institute
Prof. Dr.-Ing. Christian Wietfeld

Point of Contact (POC)
Cedrik Schüler
fax: +49 231 755 6136
e-mail: cedrik.schueler@tu-dortmund.de
internet: https://www.cni.tu-dortmund.de

Address:
TU Dortmund
Communication Networks Institute
Otto-Hahn-Str. 6
44227 Dortmund

Germany

DRZ
DEUTSCHES RETTUNGSROBOTIK ZENTRUM

SFB 876 Providing Information by Resource-Constrained Data Analysis

Funded by
DFG Deutsche Forschungsgemeinschaft
German Research Foundation

5G.NRW Competence Center

Ministerium für Wirtschaft, Innovation, Digitalisierung und Energie des Landes Nordrhein-Westfalen

Plan & Play
Zuverlässige ad-hoc 5G-Campusnetze für den temporären Einsatz