# A Mission Management Framework for Unmanned Autonomous Vehicles

**Mobilware**

28 April 2009

Eskindir Asmare, Anandha Gopalan, Morris Sloman,
Naranker Dulay and Emil Lupu

Department of Computing, Imperial College London

# Presentation Outline

- Introduction
- Motivation
- Architecture
- Failure Management
- Communication Management
- Evaluation
- Conclusion

# Introduction

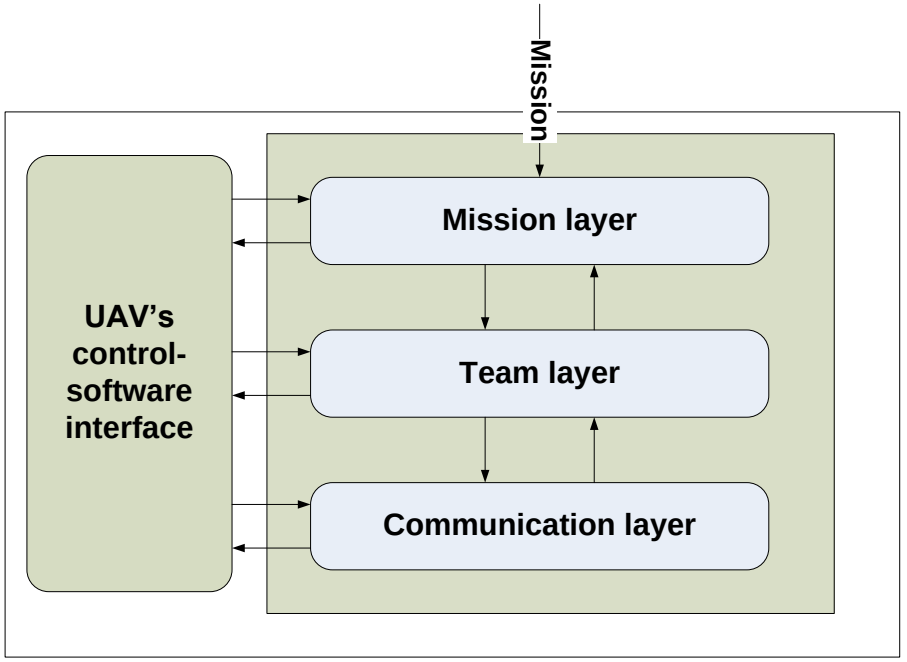
- Unmanned Autonomous Vehicle (UAV)
  - A mobile robot
  - Could be aerial, ground, underwater etc.
  - Collaborative team of UAVs used in missions which are unsuitable for humans
- Self management is of crucial importance for UAVs
  - Our work focuses on management issues

# Motivation

- Robot control architectures
  - Organise intelligence
  - Do not usually focus on management issues
- To accomplish real life missions, UAVs:
  - Should be self managing
    - Adapt to component failures and context changes
    - Optimise performance
    - Seamlessly incorporate new components and UAVs
  - Should cooperate and manage their cooperation
    - Maintain communication link to enable cooperation
    - Manage team

# Architecture

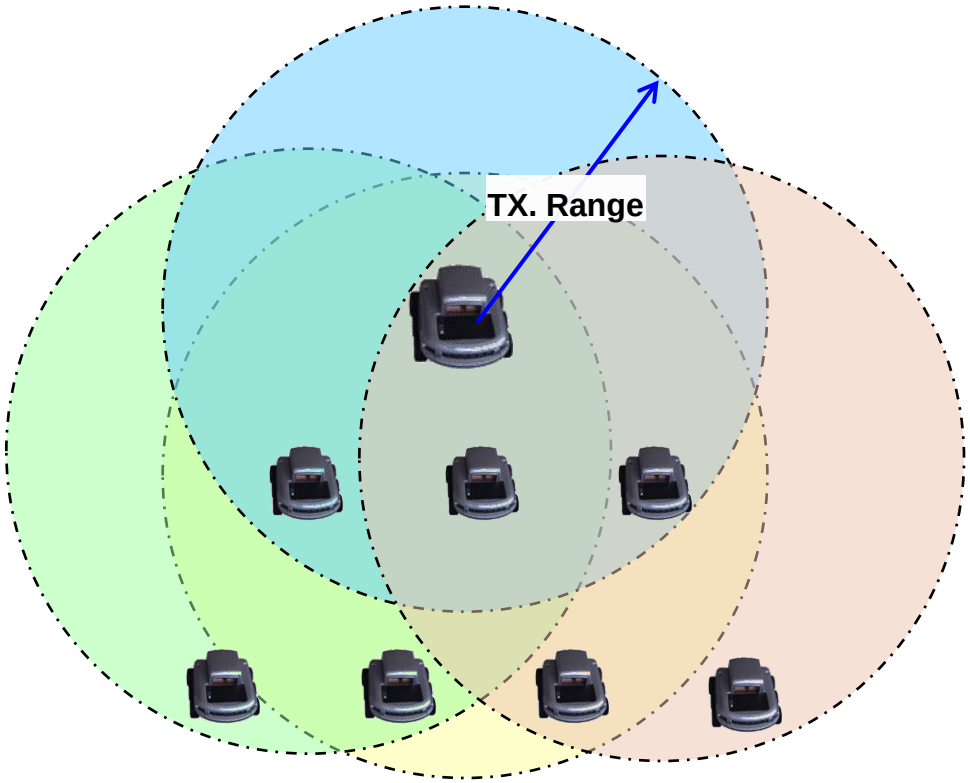
- Management architecture
  - Mission management
    - Roles
    - Mission Specification
    - Capabilities
  - Team management
    - Admission
    - Maintenance
    - Cooperation
  - Communication management
    - Formation control



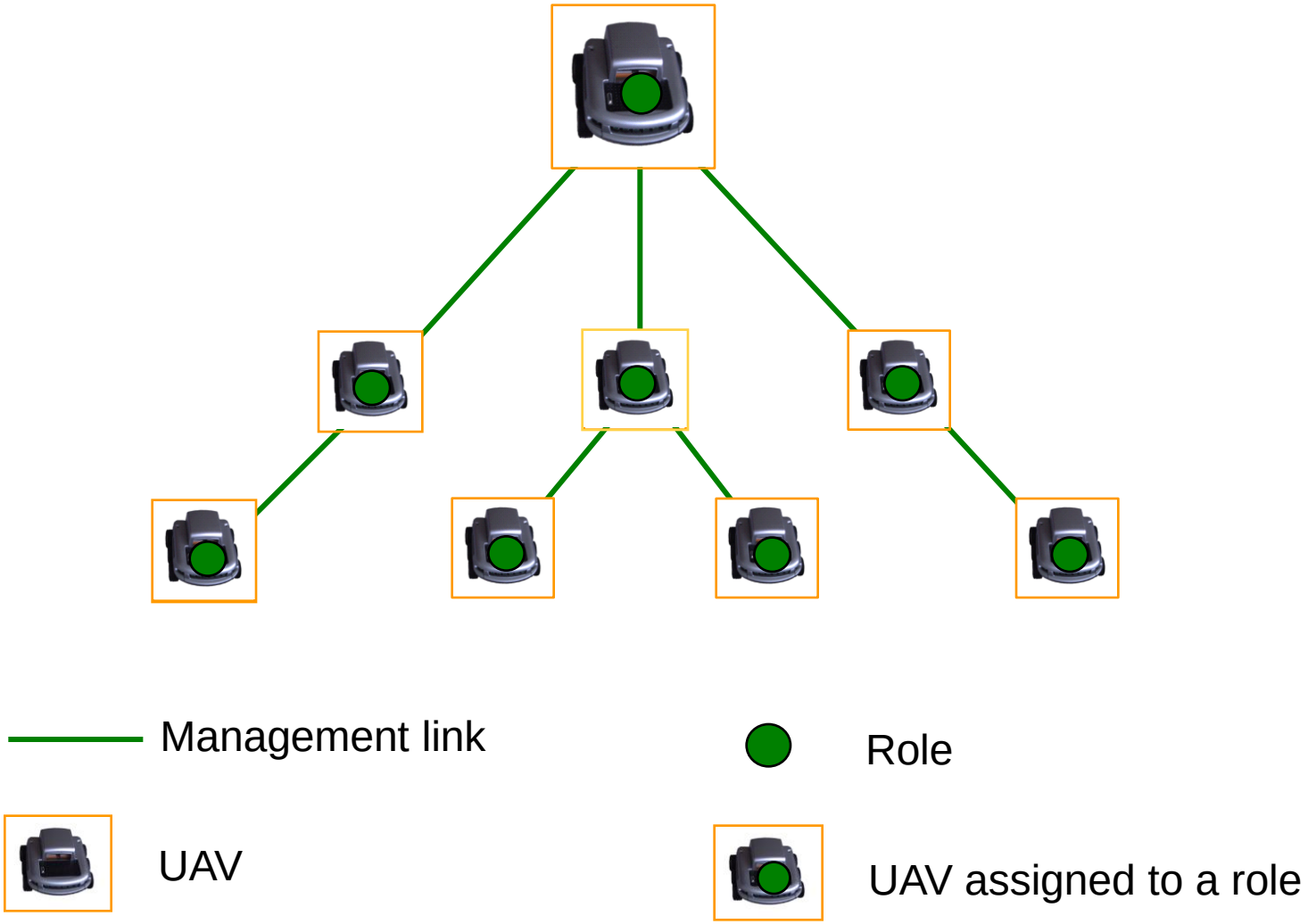
# Management Structure and Mechanism

- Hierarchical
  - Captures the management semantics without loss of distribution
  - Low message complexity
  - Suitable for optimisation during role assignment
- We use a hierarchical (tree) management structure with a policy based management mechanism

# UAVs forming an ad hoc network



# Management Tree



# Security of UAVs

- Authenticate a UAV before it joins a mission and protect group communication
- Uses the Certificate Public Key Infrastructure
- During the course of the authentication, a common secret key is generated using the Diffie-Hellman protocol for secure channels
- Authorisation policies control access to resources

# Failure Management

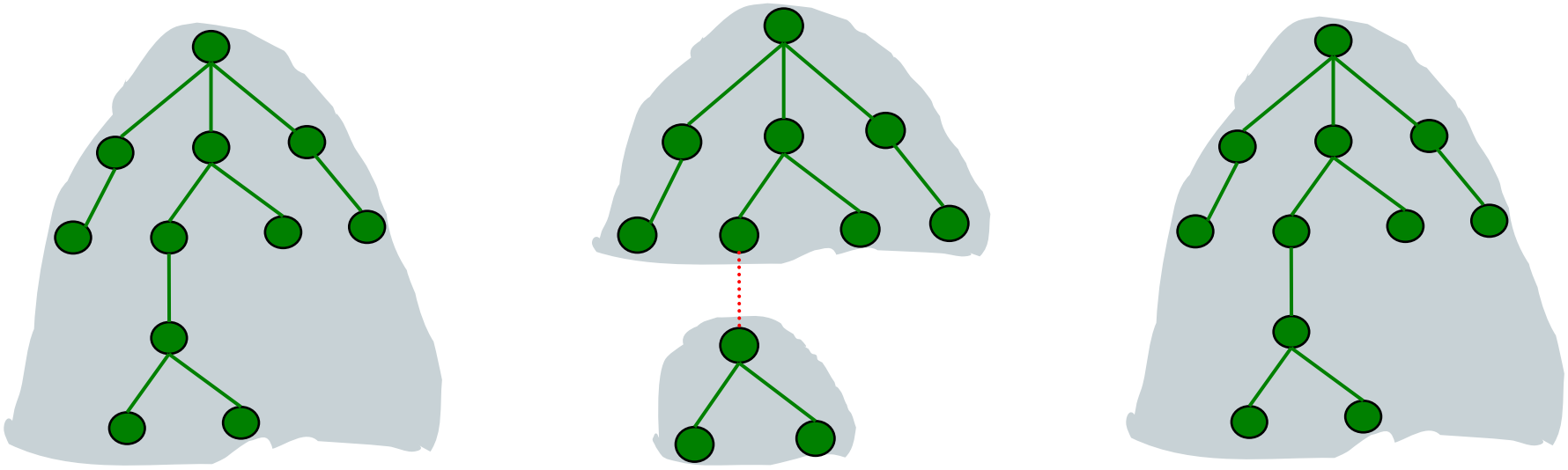
- Intermittent disconnection
  - Temporary partitioning of the logical (overlay) mission-management network
- Complete link/UAV failure
  - Partitioning of the logical (overlay) mission-management network as well as loss of roles

# Failure Management Scheme

- Detect and differentiate between failure types using time outs
- Use identity for UAVs in order to:
  - Identify a mission and the hierarchy level of a UAV in the management tree
  - Facilitate merging and re-joining of partitioned teams
- [M|H|S]
  - M: mission ID
  - H: hierarchy level
  - S: a numbering system which puts all the UAVs in the management hierarchy in a total order
  - The identity lasts throughout the team configuration

# Adapting to Intermittent Disconnection

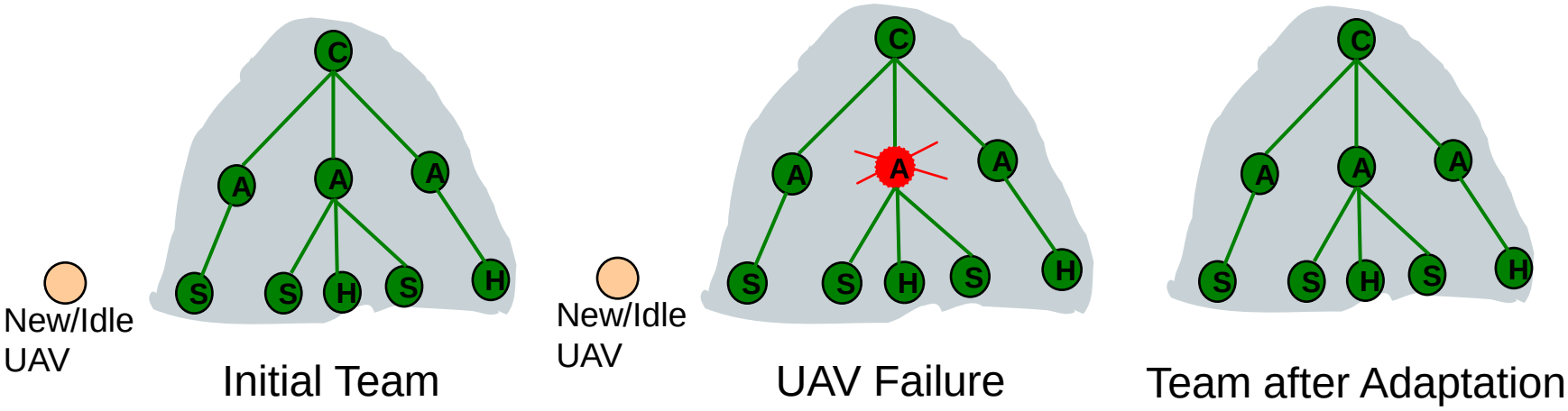
- Continue mission execution with disconnected operations and resolve inconsistencies when communication link reappears



Initial Team | Disconnection | Link Reappears

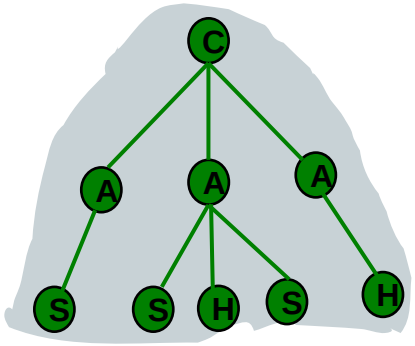
# Adapting to Link/UAV Failure

- Use newly discovered or idle UAVs to replace lost role (s)
  - The topology of the management tree remains the same
  - Role state information migrates

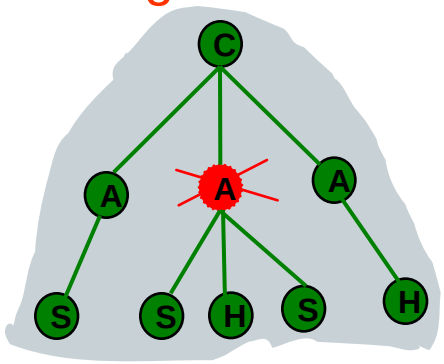


# Adapting to Link/UAV Failure

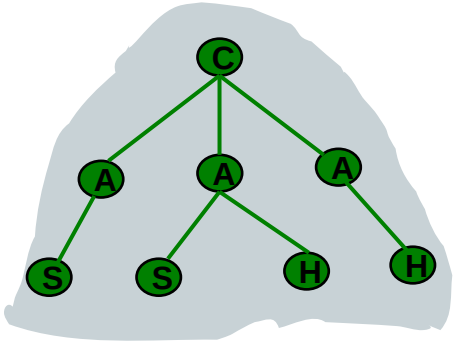
- Reconfigure the team to replace less crucial roles by failed more crucial roles
  - Used when there are no idle/new UAVs available
  - The topology of the management tree changes
  - Role state information migrates




Initial Team



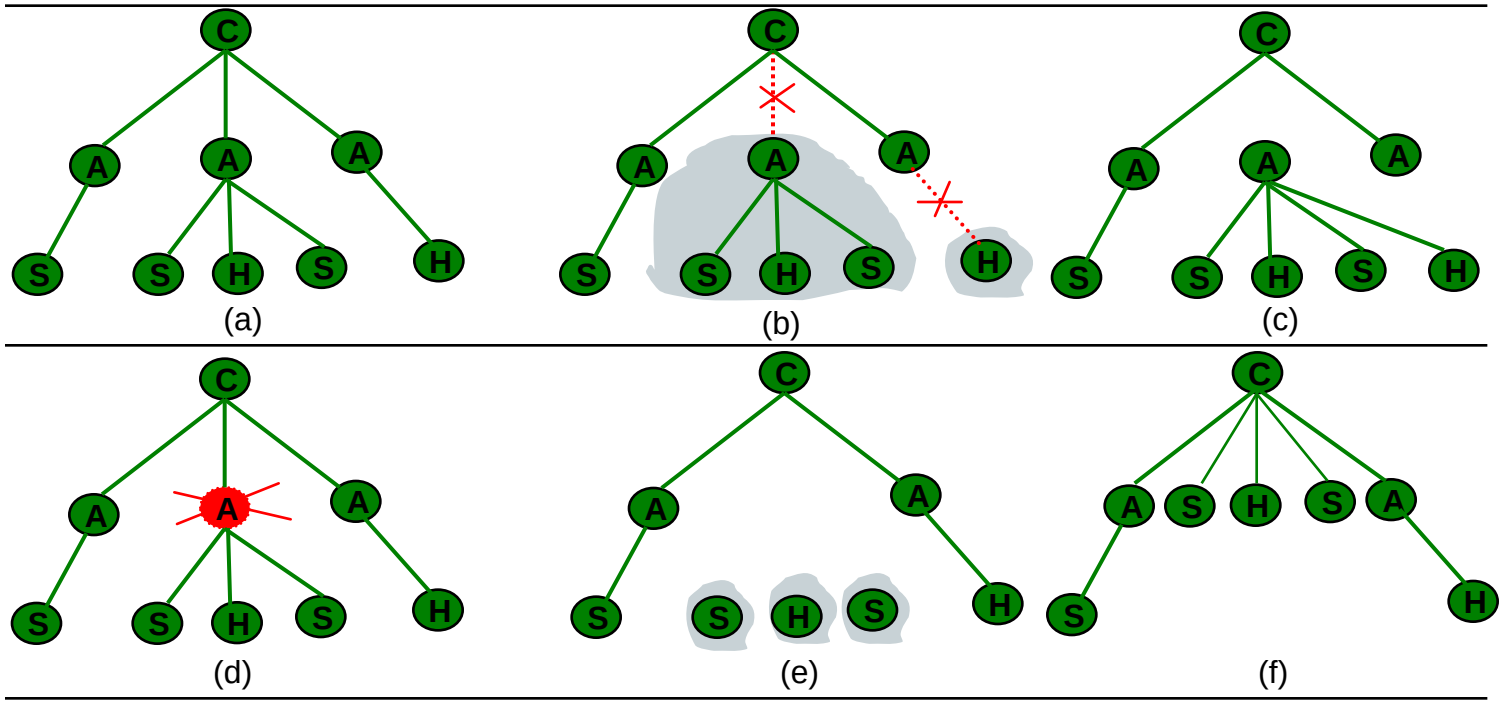

UAV Failure




Team after Adaptation

•Role S reassigned to role A

# Adaptation to Failure (more examples)



# Reassignment on Failure – Example policy

**on** Failure (UAV, role)
**do** reassign (role)
**when** role (surveyor)

```
policy event: /event/UAVFailure;
condition:[:role| role=="surveyor"];
action: [ :role :name|
            root/role/commander reassign:role scheme:"default"].
```

# Communication Management

- Why is this necessary ?
  - Desirable to have members of a mission exchange information continuously
    - Maintain state information
    - Maintain team information
- Our approach
  - Two pronged
    1. Adapt movement to maintain communication
    2. Rendezvous to restore communication
  - These approaches work in conjunction with the failure management scheme

# Approach 1

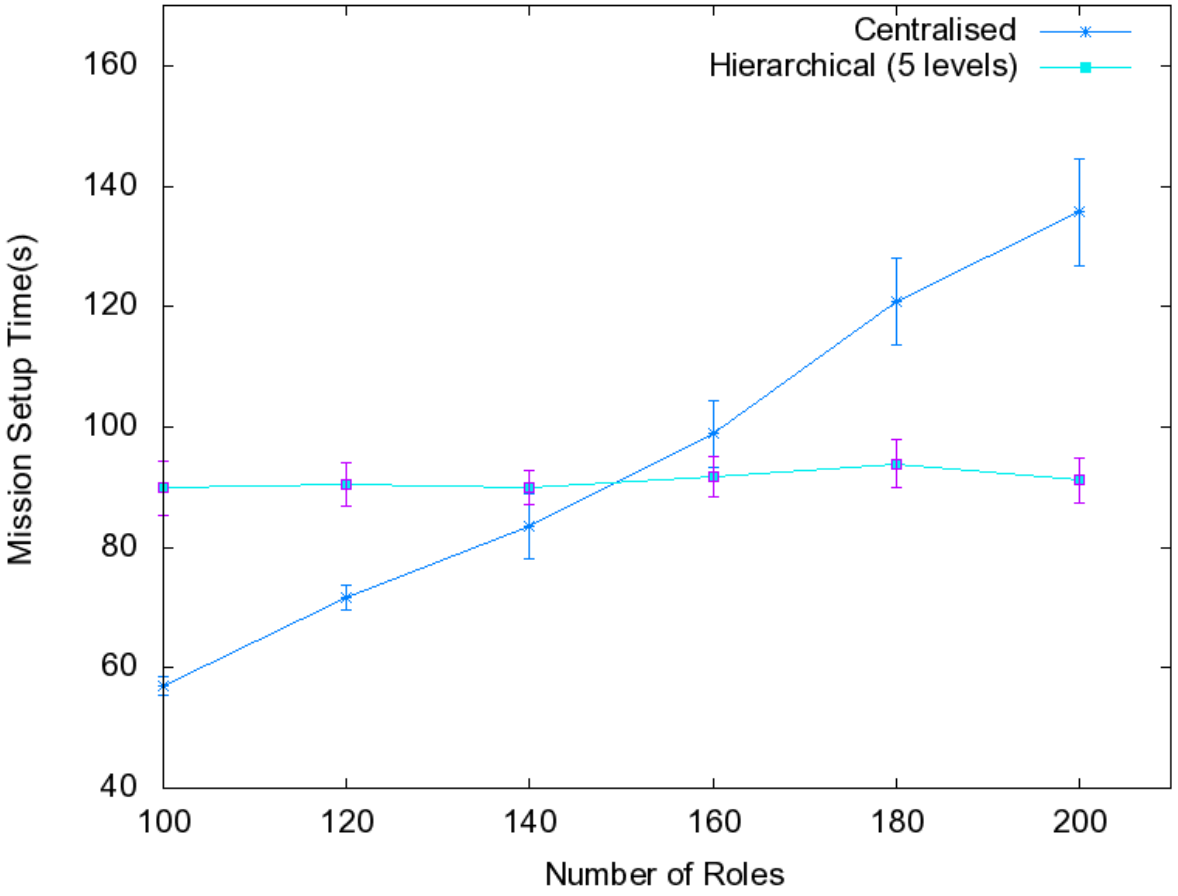
- Control movement of UAV so that they stay within communication range
- Uses location and speed of the UAVs
- When an UAV (for e.g.: *A*) is about to go out of the communication range, the UAV closest to *A* will move so as to keep *A* within communication range
  - If it is not possible to cover *A*, the protocol uses Approach 2
- Drawbacks
  - Restrictive with respect to the movement of the UAVs

# Approach 2

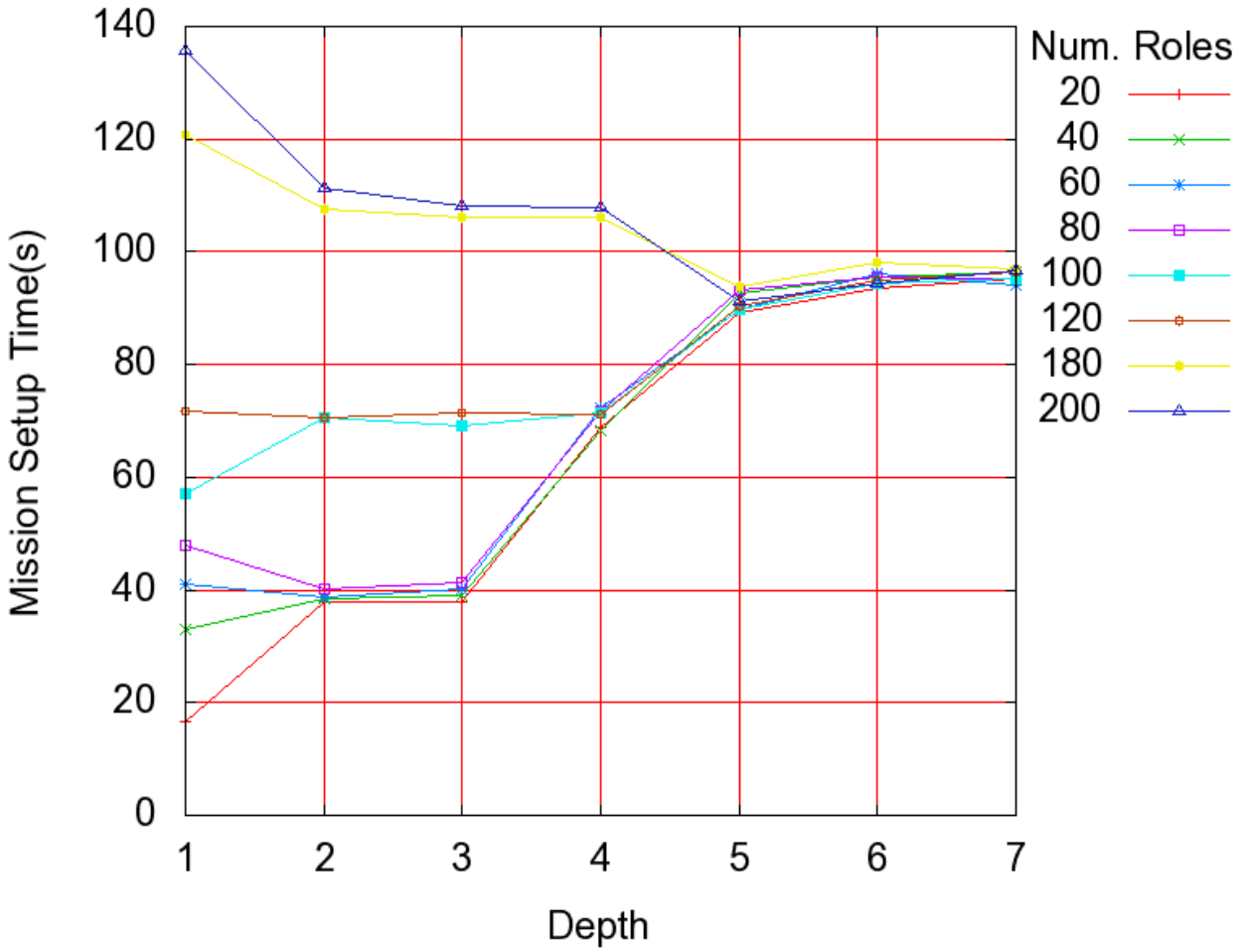
- Allows UAVs to perform disconnected individual operations, while maintaining the team structure
  - Tries to ensure that all members of the mission regardless of destination or task, communicate at intermittent intervals
- Started when the distance between nodes is greater than the *range threshold*
  - *Rendezvous area* is calculated by averaging the direction and speed of travel of all UAVs
    - In the event that an UAV is unable to reach the rendezvous area, it is assumed to have failed

# Evaluation: Mission setup time vs. Number of roles

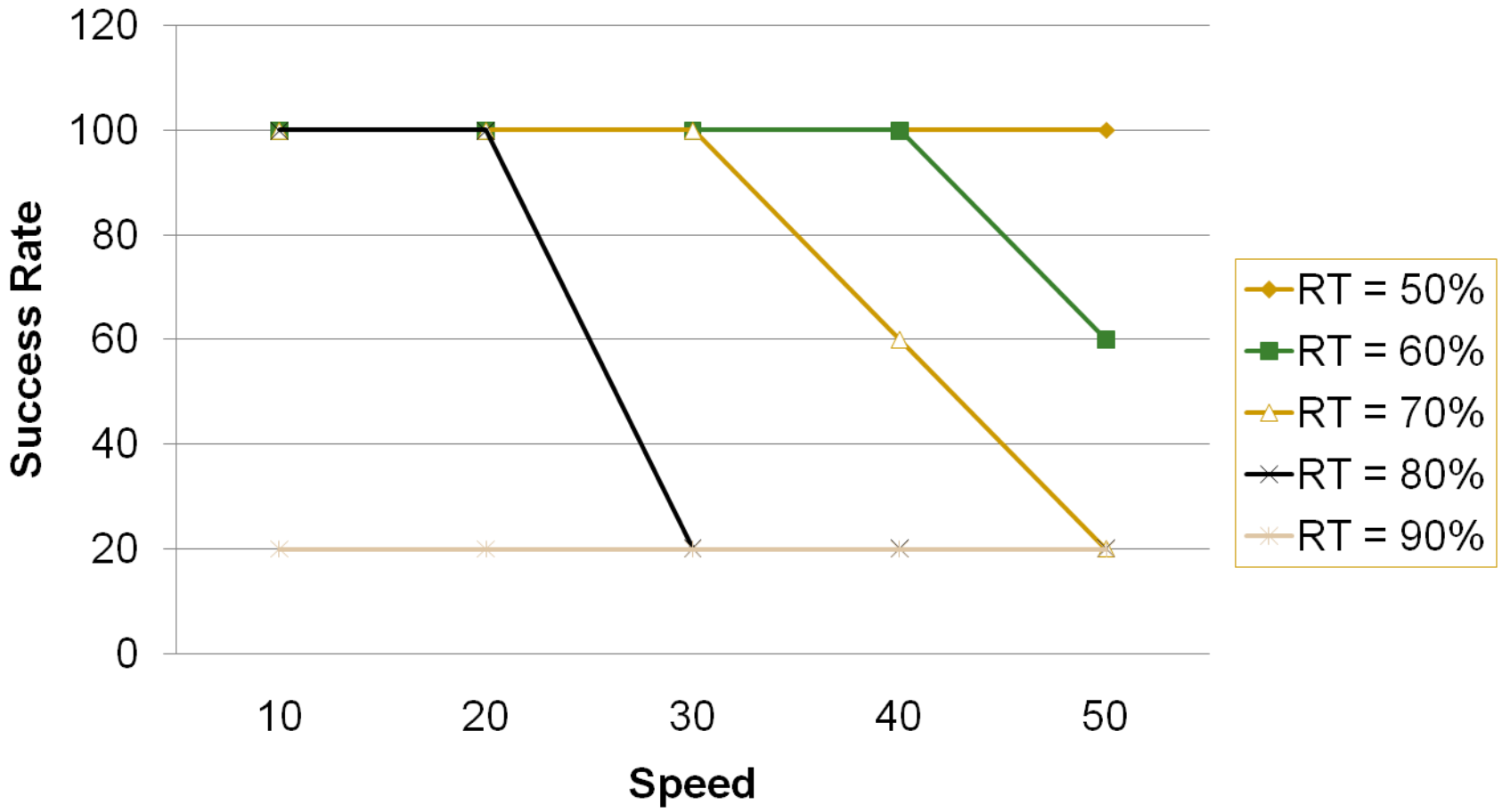
- Mission setup time
  - UAV discovery
  - Role assignment
  - Downloading policy from the repository
  - Loading policy and starting the role



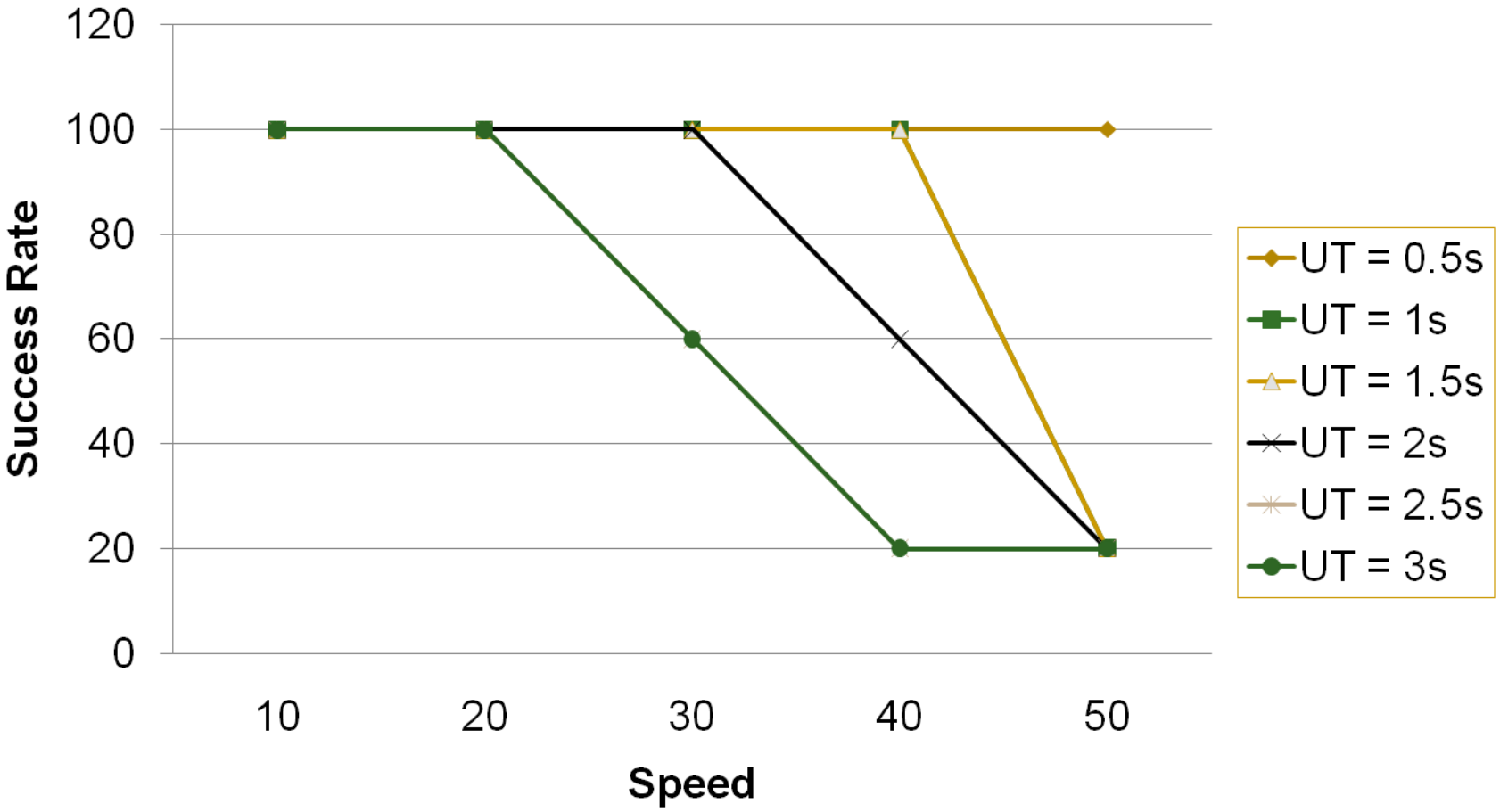
# Evaluation: Mission setup time vs. Depth



# Results: Effect of Range Threshold



# Results: Effect of Update Time



# Conclusion

- Management architecture for distributed self management of mobile autonomous systems
  - Failure management
  - Communication management
- Future work will focus mainly on evaluation and optimisation

Thank you.

{e.asmare, a.gopalan, m.sloman, n.dulay, e.c.lupu}
@imperial.ac.uk

http://www-dse.doc.ic.ac.uk