Network-Aware Automated Planning and Plan Execution

Kyle Usbeck

A Thesis Submitted to the Faculty of Drexel University in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

2009-07-07
Motivation

April 2009:

- 75% of coalition force casualties in Afghanistan are from roadside bombs.
- 40% of coalition force casualties in Iraq are from roadside bombs.

Source: Tom Vanden Brook, USA Today
April 2009:

- 75% of coalition force casualties in Afghanistan are from roadside bombs.
- 40% of coalition force casualties in Iraq are from roadside bombs.

Source: Tom Vanden Brook, USA Today
Motivating Scenario

- IED Detection.
- Monitor Locations.
- Techniques.
- Actors.
- Resources.
- Evaluators.
Motivating Scenario

- IED Detection.
- Monitor Locations.
- Techniques.
- Actors.
- Resources.
- Evaluators.
Motivating Scenario

- IED Detection.
- Monitor Locations.
- Techniques.
- Actors.
- Resources.
- Evaluators.
Motivating Scenario

- IED Detection.
- Monitor Locations.
- Techniques.
- Actors.
- Resources.
- Evaluators.
Motivating Scenario

- IED Detection.
- Monitor Locations.
- Techniques.
- Actors.
- Resources.
- Evaluators.
Motivating Scenario

- IED Detection.
- Monitor Locations.
- Techniques.
- Actors.
- Resources.
- Evaluators.
Motivating Scenario

Heterogeneous Network
multiple different network technologies are combined to work together simultaneously.

Network-Centric System
a distributed system where performance is dependent on the quality of the underlying network communication links.
Contributions

1. Qualitatively-different plans:
   - Generating plans over a range of evaluation criteria;
   - Visualizing plan evaluations.
   - Improve plan selection.

2. Network-Aware Agents:
   - Classical planning domains for distributed service composition;
   - Measuring the performance and effectiveness of planning, execution, and monitoring agents;
   - Incorporating network-awareness.
Contributions

1. Qualitatively-different plans:
   - Generating plans over a range of evaluation criteria;
   - Visualizing plan evaluations.
   - Improve plan selection.

2. Network-Aware Agents:
   - Classical planning domains for distributed service composition;
   - Measuring the performance and effectiveness of planning, execution, and monitoring agents;
   - Incorporating network-awareness.
Contributions

1. Qualitatively-different plans:
   - Generating plans over a range of evaluation criteria;
   - Visualizing plan evaluations.
   - Improve plan selection.

2. Network-Aware Agents:
   - Classical planning domains for distributed service composition;
   - Measuring the performance and effectiveness of planning, execution, and monitoring agents;
   - Incorporating network-awareness.
Contributions

1. Qualitatively-different plans:
   - Generating plans over a range of evaluation criteria;
   - Visualizing plan evaluations.
   - Improve plan selection.

2. Network-Aware Agents:
   - Classical planning domains for distributed service composition;
   - Measuring the performance and effectiveness of planning, execution, and monitoring agents;
   - Incorporating network-awareness.
Contributions

1. Qualitatively-different plans:
   - Generating plans over a range of evaluation criteria;
   - Visualizing plan evaluations.
   - Improve plan selection.

2. Network-Aware Agents:
   - Classical planning domains for distributed service composition;
   - Measuring the performance and effectiveness of planning, execution, and monitoring agents;
   - Incorporating network-awareness.
Contributions

1 Qualitatively-different plans:
   - Generating plans over a range of evaluation criteria;
   - Visualizing plan evaluations.
   - Improve plan selection.

2 Network-Aware Agents:
   - Classical planning domains for distributed service composition;
   - Measuring the performance and effectiveness of planning, execution, and monitoring agents;
   - Incorporating network-awareness.
Contributions

1. Qualitatively-different plans:
   - Generating plans over a range of evaluation criteria;
   - Visualizing plan evaluations.
   - Improve plan selection.

2. Network-Aware Agents:
   - Classical planning domains for distributed service composition;
   - Measuring the performance and effectiveness of planning, execution, and monitoring agents;
   - Incorporating network-awareness.
Contributions

1. Qualitatively-different plans:
   - Generating plans over a range of evaluation criteria;
   - Visualizing plan evaluations.
   - Improve plan selection.

2. Network-Aware Agents:
   - Classical planning domains for distributed service composition;
   - Measuring the performance and effectiveness of planning, execution, and monitoring agents;
   - Incorporating network-awareness.
“Service composition is the linking... of existing services so that their aggregate behavior is that of a desired service (the goal)” [Hoffmann et al. 09].

- Requires Semantic Web Services [Sirin et al. 04].
- QoS Assurance [Gu et al. 03].

Assumes static networking.
Service Composition to Automated Planning

Definition

“Service composition is the linking... of existing services so that their aggregate behavior is that of a desired service (the goal)” [Hoffmann et al. 09].

- Requires Semantic Web Services [Sirin et al. 04].
- QoS Assurance [Gu et al. 03].

Assumes static networking.
Service Composition to Automated Planning

Definition

“Service composition is the linking... of existing services so that their aggregate behavior is that of a desired service (the goal)” [Hoffmann et al. 09].

- Requires Semantic Web Services [Sirin et al. 04].
- QoS Assurance [Gu et al. 03].

Assumes static networking.
Agents in Planning

Agents:
- Planning Agent.
- Execution Agent.
- Monitoring Agent.

[Tate 93]
Agents in Planning

Agents:
- Planning Agent.
- Execution Agent.
- Monitoring Agent.

[Tate 93]
Agents in Planning

Agents:
- Planning Agent.
- Execution Agent.
- Monitoring Agent.

[Tate 93]
Planning Under Uncertainty

Restrictive Assumptions:

- Determinism.
- Full observability.
- Reachability goals.

[Nau et al. 04]

Sources of Uncertainty:

- Partial observability.
- Unreliable resources.
- Measurement variance.
- Inherently vague concepts.
Planning Under Uncertainty

**Restrictive Assumptions:**
- Determinism.
- Full observability.
- Reachability goals.

[Nau et al. 04]

**Sources of Uncertainty:**
- Partial observability.
- Unreliable resources.
- Measurement variance.
- Inherently vague concepts.
Fault Detection & Isolation (FDI)

Types of FDI:
- Analytic.
- Data-driven.
- Knowledge-based.

[Pettersson 05]
Fault Detection & Isolation (FDI)

Types of FDI:
- Analytic.
- Data-driven.
- Knowledge-based.

[Pettersson 05]
Fault Detection & Isolation (FDI)

Types of FDI:
- Analytic.
- Data-driven.
- Knowledge-based.

[Pettersson 05]
Approach

1. Modify planner to improve the quality of the plans it produces based on evaluation criteria.
2. Add network-awareness to planning, execution, and monitoring agents.

Purpose
To improve network-centric automated planning and execution.
Approach

1. Modify planner to improve the quality of the plans it produces based on evaluation criteria.
2. Add network-awareness to planning, execution, and monitoring agents.

Purpose
To improve network-centric automated planning and execution.
Approach

1. Modify planner to improve the quality of the plans it produces based on evaluation criteria.

2. Add network-awareness to planning, execution, and monitoring agents.

Purpose
To improve network-centric automated planning and execution.
Outline

1. Introduction
   - Motivation
   - Background
   - Approach

2. Formalization
   - Problem Statement

3. Technical Approach
   - Planning Agents
   - Execution Agents
   - Monitoring Agents
   - Mixed-initiative UI

4. Experiments
   - Plan Evaluation Benchmarking
   - Network-Aware Agent Combinations
   - Discussion
Σ is the planning domain — the model of the world passed as input to the planner.

Σ is a Tuple

- $S$: set of states;
- $A$: set of actions;
- $E$: set of events;
- $\gamma$: transition function $\gamma: S \times A \rightarrow S$. 
Σ is the planning domain — the model of the world passed as input to the planner.

Σ is a Tuple

- $S$ set of states;
- $A$ set of actions;
- $E$ set of events;
- $\gamma$ transition function $\gamma : S \times A \rightarrow S$. 
Formal Problem Statement

Σ is the planning domain — the model of the world passed as input to the planner.

Σ is a Tuple

\[ \Sigma = (S, A, E, \gamma) \]

- \( S \) set of states;
- \( A \) set of actions;
- \( E \) set of events;
- \( \gamma \) transition function \( \gamma : S \times A \rightarrow S \).
$\Sigma$ is the planning domain — the model of the world passed as input to the planner.

$\Sigma$ is a Tuple

- $S$ set of states;
- $A$ set of actions;
- $E$ set of events;
- $\gamma$ transition function $\gamma : S \times A \rightarrow S$. 
The functions on planning actions:

For \( a \in A \)

- \( \text{precond}(a) \): preconditions of \( a \);
- \( \text{effects}^+(a) \): positive effects of \( a \);
- \( \text{effects}^-(a) \): negative effects of \( a \);
- \( \text{host}(a) \): the single host \( h \) from \( a \);
- \( \text{resources}(a) \): the set of resources (parameters) of action \( a \).
The functions on planning actions:

For $a \in A$

- $\text{precond}(a)$: preconditions of $a$;
- $\text{effects}^+(a)$: positive effects of $a$;
- $\text{effects}^-(a)$: negative effects of $a$;
- $\text{host}(a)$: the single host $h$ from $a$;
- $\text{resources}(a)$: the set of resources (parameters) of action $a$. 
Formal Problem Statement

The functions on planning actions:

For $a \in A$

- $\text{precond}(a)$: preconditions of $a$;
- $\text{effects}^+(a)$: positive effects of $a$;
- $\text{effects}^-(a)$: negative effects of $a$;
- $\text{host}(a)$: the single host $h$ from $a$;
- $\text{resources}(a)$: the set of resources (parameters) of action $a$. 
Formal Problem Statement

The functions on planning actions:

For $a \in A$

- $\text{precond}(a)$: preconditions of $a$;
- $\text{effects}^+(a)$: positive effects of $a$;
- $\text{effects}^-(a)$: negative effects of $a$;
- $\text{host}(a)$: the single host $h$ from $a$;
- $\text{resources}(a)$: the set of resources (parameters) of action $a$. 
The functions on planning actions:

For $a \in A$

- $\text{precond}(a)$ preconditions of $a$;
- $\text{effects}^+(a)$ positive effects of $a$;
- $\text{effects}^-(a)$ negative effects of $a$;
- $\text{host}(a)$ the single host $h$ from $a$;
- $\text{resources}(a)$ the set of resources (parameters) of action $a$. 
The planning agent receives the tuple, $I_P$, and creates a set of plans, $P_I$.

$I_P$ is a Tuple

- $\Sigma$: automated planning domain;
- $s_0$: initial state;
- $S_g$: set of goal state(s);
- $H$: set of hosts (nodes) on the network;
- $\omega_H$: host link weighting.
The planning agent receives the tuple, $I_P$, and creates a set of plans, $P_I$.

$I_P$ is a Tuple

- $\Sigma$ automated planning domain;
- $s_0$ initial state;
- $S_g$ set of goal state(s);
- $H$ set of hosts (nodes) on the network;
- $\omega_H$ host link weighting.
The planning agent receives the tuple, $I_P$, and creates a set of plans, $P_I$.

$I_P$ is a Tuple

- $\Sigma$: automated planning domain;
- $s_0$: initial state;
- $S_g$: set of goal state(s);
- $H$: set of hosts (nodes) on the network;
- $\omega_H$: host link weighting.
The planning agent receives the tuple, $I_P$, and creates a set of plans, $P_I$.

$I_P$ is a Tuple

- $\Sigma$ automated planning domain;
- $s_0$ initial state;
- $S_g$ set of goal state(s);
- $H$ set of hosts (nodes) on the network;
- $\omega_H$ host link weighting.
The planning agent receives the tuple, $I_P$, and creates a set of plans, $P_I$.

$I_P$ is a Tuple

- $\Sigma$ automated planning domain;
- $s_0$ initial state;
- $S_g$ set of goal state(s);
- $H$ set of hosts (nodes) on the network;
- $\omega_H$ host link weighting.
Formal Problem Statement

Problem

To find and execute $p_I \in P_I$ where $p_I = \{a_0, a_1, \ldots, a_{|p_I|}\}$ and execution of $p_I$ yields the best domain-dependent and network-centric evaluations.

Network-Awareness

An agent exhibits network-awareness if changes to $\omega_H$ cause the agent’s output to change while all other inputs remain constant.
Formal Problem Statement

Problem
To find and execute $p_I \in P_I$ where $p_I = \{a_0, a_1, \ldots, a_{|p_I|}\}$ and execution of $p_I$ yields the best domain-dependent and network-centric evaluations.

Network-Awareness
An agent exhibits network-awareness if changes to $\omega_H$ cause the agent’s output to change while all other inputs remain constant.
Formal Problem Statement
Formal Problem Statement

- Planning Agent
- Execution Agent
- Plans
- Faults
- Service Calls

Host 1

Host 2

Host 3

Host 4

Host 5
Formal Problem Statement

Planning Agent → Plan(s) → Execution Agent → Monitoring Agent

- [major fault] Fault
- [minor fault] Fault
## Outline

1. **Introduction**
   - Motivation
   - Background
   - Approach

2. **Formalization**
   - Problem Statement

3. **Technical Approach**
   - Planning Agents
   - Execution Agents
   - Monitoring Agents
   - Mixed-initiative UI

4. **Experiments**
   - Plan Evaluation Benchmarking
   - Network-Aware Agent Combinations
   - Discussion
Planning Domain Extensions

Operator distribution

- \textit{e.g., NODE1\textsc{Action}(parameters)}
- Implicit constraints.

Resource distribution

- \textit{e.g., ACTION(node1, parameters)}
- \(s_0 \leftarrow s_0 \cup \{\text{TYPE}(node1) = \text{NETWORK}\textsc{Node}\}\)
- \(s_0 \leftarrow s_0 \cup \{\text{ACTION}(node1) = \text{true}\}\)
Planning Domain Extensions

Operator distribution
- \textit{e.g.}, \textsc{Node1Action}(parameters)
- Implicit constraints.

Resource distribution
- \textit{e.g.}, \textsc{Action}(node1, parameters)
- $s_0 \leftarrow s_0 \cup \{\textsc{Type}(node1) = \textsc{NetworkNode}\}$
- $s_0 \leftarrow s_0 \cup \{\textsc{Action}(node1) = \texttt{true}\}$

Complexity
- Operator distribution increases the number of actions in $\Sigma$ to $|H| \times |A|$ in the worst case.
- Resource distribution increases the number of constraints in the world-state.
Planning Domain Extensions

Operator distribution

- *e.g.*, \texttt{NODE1\text{ACTION}(parameters)}
- Implicit constraints.

Resource distribution

- *e.g.*, \texttt{ACTION(node1, parameters)}

\[
\begin{align*}
s_0 & \leftarrow s_0 \cup \{\text{TYPE}(node1) = \text{NETWORKNODE}\} \\
\end{align*}
\]

\[
\begin{align*}
s_0 & \leftarrow s_0 \cup \{\text{ACTION}(node1) = \text{true}\} \\
\end{align*}
\]
Planning Domain Extensions

**Operator distribution**
- *e.g.*, NODE1ACTION(*parameters*)
- Implicit constraints.

**Resource distribution**
- *e.g.*, ACTION(node1, *parameters*)
- $s_0 \leftarrow s_0 \cup \{\text{TYPE}(\text{node1}) = \text{NETWORKNODE}\}$
- $s_0 \leftarrow s_0 \cup \{\text{ACTION}(\text{node1}) = \text{true}\}$
Planning Domain Extensions

**Operator distribution**
- *e.g.*, \texttt{NODE1ACTION(parameters)}
- Implicit constraints.

**Resource distribution**
- *e.g.*, \texttt{ACTION(node1, parameters)}
- \( s_0 \leftarrow s_0 \cup \{ \text{TYPE(node1) = NETWORK\textsc{node}} \} \)
- \( s_0 \leftarrow s_0 \cup \{ \text{ACTION(node1) = true} \} \)

Complexity
Operator distribution increases the number of actions in \( \Sigma \) to \(|H| \times |A| \) in the worst case.

Resource distributed increases the number of constraints in the world-state.
Planning Domain Extensions

### Operator distribution
- *e.g.*, `NODE1 ACTION(parameters)`
- Implicit constraints.

### Resource distribution
- *e.g.*, `ACTION(node1, parameters)`

#### Complexity
- Operator distribution increases the number of actions in $\Sigma$ to $|H| \times |A|$ in the worst case.
- Resource distributed increases the number of constraints in the world-state.

```plaintext
s_0 \leftarrow s_0 \cup \{ \text{TYPE(node1) = NETWORK NODE} \}
s_0 \leftarrow s_0 \cup \{ \text{ACTION(node1) = true} \}
```
Planning Agents

Agent Types:
- Domain-Independent.
- Random.
- Guided.

Plan Evaluators:
- Steps.
- Alternatives.
- Longest temporally ordered path.
- Duplicate plans.
Planning Agents

Plan Evaluators:

- (none).

Agent Types:
- Domain-Independent.
- Random.
- Guided.
Planning Agents

Agent Types:
- Domain-Independent.
- Random.
- Guided.

Plan Evaluators:
- IED detection accuracy.
- Plan execution time.
- Network link quality.
- Network bandwidth usage.
## Domain-Independent Planning Agent

- Uses I-Plan’s default strategy.

---

### I-Plan

University of Edinburgh, Tate et al.’s plan-space HTN planner which is built on an intelligent agent framework, I-X.

---

### Process

1. Traverses search space depth-first.
2. Encounter an alternative whose constraints cannot be satisfied.
Domain-Independent Planning Agent

- Uses I-Plan’s default strategy.

**I-Plan**

University of Edinburgh, Tate et al. ’s plan-space HTN planner which is built on an intelligent agent framework, I-X.

**Process**

1. Traverses search space depth-first.
2. Encounter an alternative whose constraints cannot be satisfied.
Domain-Independent Planning Agent

- Uses I-Plan’s default strategy.

**I-Plan**

University of Edinburgh, Tate *et al.* ’s plan-space HTN planner which is built on an intelligent agent framework, I-X.

**Process**

1. Traverses search space depth-first.
2. Encounter an alternative whose constraints cannot be satisfied.
Domain-Independent Planning Agent

- Uses I-Plan’s default strategy.

I-Plan

University of Edinburgh, Tate et al. ’s plan-space HTN planner which is built on an intelligent agent framework, I-X.

Process

1. Traverses search space depth-first.
2. Encounter an alternative whose constraints cannot be satisfied.
Domain-Independent Planning Agent

- Uses I-Plan’s default strategy.

I-Plan

University of Edinburgh, Tate et al. ’s plan-space HTN planner which is built on an intelligent agent framework, I-X.

Process

1. Traverses search space depth-first.
2. Encounter an alternative whose constraints cannot be satisfied.
Random Planning Agent

- DFS with random branching.

**Process**

\[\text{CONSTRUCT RANDOM PLAN}(I_P)\]

1: \(\text{toVisit.push}(s_0)\)
2: \(\textbf{while } \neg \text{toVisit.empty()} \land \neg \text{solution(toVisit.peek()) } \textbf{do}\)
3: \(v \leftarrow \text{toVisit.pop()}\)
4: \(\textbf{if } v \notin \text{visited } \textbf{then}\)
5: \(\text{visited.add}(v)\)
6: \(r \leftarrow \text{randomize}(v.\text{children}())\)
7: \(\text{toVisit.push}(r)\)
8: \(\textbf{end if}\)
9: \(\textbf{end while}\)
10: \(\textbf{return } \text{toVisit.peek()}\)
Generates qualitatively-different plans over:

- Domain-dependent criteria, and
- Network-centric criteria.

**Process**

1. A priority queue exists for each evaluator.
2. Every partial-plan is evaluated by all evaluators and placed in their respective priority queues.
3. The partial-plan at the head of each priority queue is used for the next step.
Guided Planning Agent

Generates qualitatively-different plans over:
- Domain-dependent criteria, and
- Network-centric criteria.

Process

1. A priority queue exists for each evaluator.
2. Every partial-plan is evaluated by all evaluators and placed in their respective priority queues.
3. The partial-plan at the head of each priority queue is used for the next step.
Guided Planning Agent

Guided Planning Agent

Guides qualitatively different plans over:
- Domain-dependent criteria, and
- Network-centric criteria.

**Process**

1. A priority queue exists for each evaluator.
2. Every partial-plan is evaluated by all evaluators and placed in their respective priority queues.
3. The partial-plan at the head of each priority queue is used for the next step.
Guided Planning Agent

Generates qualitatively-different plans over:
- Domain-dependent criteria, and
- Network-centric criteria.

Process

1. A priority queue exists for each evaluator.
2. Every partial-plan is evaluated by all evaluators and placed in their respective priority queues.
3. The partial-plan at the head of each priority queue is used for the next step.
Guided Planning Agent

- Planner
- Evaluator 1
  - Priority Queue
- Evaluator 2
  - Priority Queue
- Evaluator 3
  - Priority Queue
- Evaluator 4
  - Priority Queue

next partial-plan

evaluations
Guided Planning Agent

- Planner
- Evaluator 1: Priority Queue
- Evaluator 2: Priority Queue
- Evaluator 3: Priority Queue
- Evaluator 4: Priority Queue

Flow:
- Planner outputs a new partial-plan.
- The partial-plan is evaluated by the Evaluator agents and the Priority Queues.
Execution Agents

Agent types:
- Naïve.
- Reactive.
- Proactive.

Defined by:
- Service invocation.
- Error handling.
Execution Agents

Agent types:
- Naïve.
- Reactive.
- Proactive.

Defined by:
- Service invocation.
- Error handling.
Naïve Execution Agent

Naïve Execution Agent Properties

**Service Invocation** Invokes services exactly as described by $p_I$. The naïve agent requires that
\[
\forall \text{ actions } a \in p_I, \text{ host}(a) \neq \emptyset \land \text{ resources}(a) \neq \{}.
\]

**Error Handling** Ignores execution errors.

- **Not** network-aware.
Naïve Execution Agent

Naïve Execution Agent Properties

Service Invocation  Invokes services exactly as described by $p_I$. The naïve agent requires that
\[ \forall \text{ actions } a \in p_I, \text{host}(a) \neq \emptyset \land \text{resources}(a) \neq \{\}. \]

Error Handling  Ignores execution errors.

Not network-aware.
Naïve Execution Agent

Naïve Execution Agent Properties

Service Invocation  Invokes services exactly as described by \( p_I \).

\[ \forall \text{ actions } a \in p_I, \text{ host}(a) \neq \emptyset \land \text{ resources}(a) \neq \{\} \].

Error Handling  Ignores execution errors.

- Not network-aware.
Reactive Execution Agent Properties

**Service Invocation** Invokes services exactly as described by $p_I$. The reactive agent requires that

\[ \forall \text{ actions } a \in p_I, \text{host}(a) \neq \emptyset \land \text{resources}(a) \neq \{\}. \]

**Error Handling** Repairs the failed $p_I$ by replacing failed service call(s) with new ones, creating $p'_I$.

- Network-aware recovery — plan repair.
- Uses routing protocol neighbors & link quality.
Reactive Execution Agent Properties

**Service Invocation** Invokes services exactly as described by $p_I$. The reactive agent requires that

$$\forall \text{ actions } a \in p_I, \text{ host}(a) \neq \emptyset \land \text{ resources}(a) \neq \emptyset.$$

**Error Handling** Repairs the failed $p_I$ by replacing failed service call(s) with new ones, creating $p'_I$.

- Network-aware recovery — plan repair.
- Uses routing protocol neighbors & link quality.
Reactive Execution Agent Properties

**Service Invocation**  Invokes services exactly as described by $p_I$. The reactive agent requires that

$$\forall \text{ actions } a \in p_I, \text{ host}(a) \neq \emptyset \land \text{resources}(a) \neq \{\}.$$  

**Error Handling**  Repairs the failed $p_I$ by replacing failed service call(s) with new ones, creating $p'_I$.

- Network-aware recovery — plan repair.
- Uses routing protocol neighbors & link quality.
Reactive Execution Agent Properties

**Service Invocation**  Invokes services exactly as described by $p_I$. The reactive agent requires that
\[ \forall \text{ actions } a \in p_I, \ \text{host}(a) \neq \emptyset \land \text{resources}(a) \neq \emptyset. \]

**Error Handling**  Repairs the failed $p_I$ by replacing failed service call(s) with new ones, creating $p'_I$.

- Network-aware recovery — plan repair.
- Uses routing protocol neighbors & link quality.
Reactive Execution Agent

Flowchart:
- **Start**
  - Accept Plan
    - has more actions? (Yes/No)
    - If Yes, go to **Execute Next Action**
    - If No, go to **Success**
  - If Start, has more actions?
    - No: go to **Failure**
    - Yes: go to **Repair Plan**
- **Success**
  - repaired?
    - Yes: go to **Repair Plan**
    - No: go to **Failure**

- **Execute Next Action**
  - failed?
    - Yes: go to **Repair Plan**
    - No: go to **Repair Plan**
Proactive Execution Agent Properties

**Service Invocation**  Invokes services using network-aware logic to choose the host and resources at execution time. The proactive execution agent uses only service descriptions from actions $a \in p_I$, meaning 

$$\forall a \in p_I, \text{host}(a) = \emptyset \land \text{resources}(a) = \{\}$$

**Error Handling**  Repairs the failed $p_I$ by replacing failed service call(s) with new ones, creating $p'_I$.

- Network-aware host/resource grounding.
Proactive Execution Agent Properties

**Service Invocation**  Invokes services using network-aware logic to choose the host and resources at execution time. The proactive execution agent uses only service descriptions from actions $a \in p_I$, meaning $\forall a \in p_I, \text{host}(a) = \emptyset \land \text{resources}(a) = \{\}$

**Error Handling**  Repairs the failed $p_I$ by replacing failed service call(s) with new ones, creating $p'_I$. 

- Network-aware host/resource grounding.
Proactive Execution Agent Properties

Service Invocation  Invokes services using network-aware logic to choose the host and resources at execution time. The proactive execution agent uses only service descriptions from actions $a \in p_I$, meaning $\forall a \in p_I$, $\text{host}(a) = \emptyset \land \text{resources}(a) = \{\}$

Error Handling   Repairs the failed $p_I$ by replacing failed service call(s) with new ones, creating $p_I'$.

- Network-aware host/resource grounding.
Outline

1. Introduction
   - Motivation
   - Background
   - Approach

2. Formalization
   - Problem Statement

3. Technical Approach
   - Planning Agents
   - Execution Agents
   - Monitoring Agents
   - Mixed-initiative UI

4. Experiments
   - Plan Evaluation Benchmarking
   - Network-Aware Agent Combinations
   - Discussion
Monitoring Agents

Methods of FDI

1. Analytic.
2. Data-driven.
Monitoring Agents

Methods of FDI

1. Analytic. ← Active Monitor
2. Data-driven. ← Passive Monitor
Given the ordered plan \( p_I = \{a_0, a_1, \ldots, a_{|p_I|}\} \)

An analytic monitoring agent:

1. Constructs \( p_M = \{m_0, m_1, \ldots, m_{|p_I|+1}\} \), an ordered set of monitoring actions;
2. Creates the new execution plan \( p'_I = \bigcup_{i=0}^{n} \{m_i, a_i\} \);
3. The result is \( p'_I = \{m_0, a_0, m_1, a_1, \ldots, m_{|p_I|}, a_{|p_I|}, m_{|p_I|+1}\} \);
4. Each \( m \in p_M \) calculates the residual between expected and actual bytes transferred.
Analytic Monitoring Agent

Given the ordered plan $p_I = \{a_0, a_1, \ldots, a_{|p_I|}\}$

An analytic monitoring agent:

1. Constructs $p_M = \{m_0, m_1, \ldots, m_{|p_I|+1}\}$, an ordered set of monitoring actions;
2. Creates the new execution plan $p'_I = \bigcup_{i=0}^{n} \{m_i, a_i\}$;
3. The result is $p'_I = \{m_0, a_0, m_1, a_1, \ldots, m_{|p_I|}, a_{|p_I|}, m_{|p_I|+1}\}$.
4. Each $m \in p_M$ calculates the residual between expected and actual bytes transferred.
Given the ordered plan $p_I = \{a_0, a_1, \ldots, a_{|p_I|}\}$

An analytic monitoring agent:

1. Constructs $p_M = \{m_0, m_1, \ldots, m_{|p_I|+1}\}$, an ordered set of monitoring actions;
2. Creates the new execution plan $p'_I = \bigcup_{i=0}^{n} \{m_i, a_i\}$;
3. The result is $p'_I = \{m_0, a_0, m_1, a_1, \ldots, m_{|p_I|}, a_{|p_I|}, m_{|p_I|+1}\}$.
4. Each $m \in p_M$ calculates the residual between expected and actual bytes transferred.
Given the ordered plan $p_I = \{a_0, a_1, \ldots, a_{|p_I|}\}$

An analytic monitoring agent:

1. Constructs $p_M = \{m_0, m_1, \ldots, m_{|p_I|+1}\}$, an ordered set of monitoring actions;

2. Creates the new execution plan $p'_I = \bigcup_{i=0}^{n} \{m_i, a_i\}$;

3. The result is $p'_I = \{m_0, a_0, m_1, a_1, \ldots, m_{|p_I|}, a_{|p_I|}, m_{|p_I|+1}\}$.

4. Each $m \in p_M$ calculates the residual between expected and actual bytes transferred.
Given the ordered plan $p_I = \{a_0, a_1, \ldots, a_{|p_I|}\}$

An analytic monitoring agent:

1. Constructs $p_M = \{m_0, m_1, \ldots, m_{|p_I|+1}\}$, an ordered set of monitoring actions;
2. Creates the new execution plan $p'_I = \bigcup_{i=0}^{n} \{m_i, a_i\}$;
3. The result is $p'_I = \{m_0, a_0, m_1, a_1, \ldots, m_{|p_I|}, a_{|p_I|}, m_{|p_I|+1}\}$.
4. Each $m \in p_M$ calculates the residual between expected and actual bytes transferred.
Analytic Monitoring Agent

Execution Agent  Monitoring Agent 1  Monitoring Agent n

Starting Action

Action Execution

Ending Action

Monitoring Agent 1 Residual

Monitoring Agent n Residual

Fault Detection
Data-driven Monitoring Agent

- Execution Agent
- Data-driven Monitoring Agent 1
- Data-driven Monitoring Agent 2
- ... Data-driven Monitoring Agent n
- Faults
Data-driven Monitoring Agent

- Multivariate monitor.
- Data packets.
- Retransmission timeouts.
### Data-driven Monitoring Agent

- Multivariate monitor.
- Data packets.
- Retransmission timeouts.

#### Network Statistics During Host Disconnection

<table>
<thead>
<tr>
<th>Data Type</th>
<th>0</th>
<th>10</th>
<th>100</th>
<th>1000</th>
<th>10000</th>
<th>100000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packets Sent</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Packets</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Retransmitted Data Packets</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Received Packets</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCP ACKs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Duplicate TCP ACKs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Completely Duplicate Packets</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Old Duplicate Packets</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Window Update Packets</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Segments Updated RTT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Retransmission Timeouts</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(See legend for units)
Outline

1. Introduction
   - Motivation
   - Background
   - Approach

2. Formalization
   - Problem Statement

3. Technical Approach
   - Planning Agents
   - Execution Agents
   - Monitoring Agents
   - Mixed-initiative UI

4. Experiments
   - Plan Evaluation Benchmarking
   - Network-Aware Agent Combinations
   - Discussion
Plan Evaluation Criteria Statistics

Aspects

- Range (effective and theoretic).
- Direction (minimize or maximize).
- Statistics (e.g., mean, median, mode, standard deviation).

Benefit

Plans can be positioned along an absolute continuum of evaluation values.
Plan Evaluation Criteria Statistics

Aspects

- Range (effective and theoretic).
- Direction (minimize or maximize).
- Statistics (e.g., mean, median, mode, standard deviation).

Benefit

Plans can be positioned along an absolute continuum of evaluation values.
Plan Evaluation Criteria Statistics

Aspects
- Range (effective and theoretic).
- Direction (minimize or maximize).
- Statistics (e.g., mean, median, mode, standard deviation).

Benefit
Plans can be positioned along an absolute continuum of evaluation values.
Plan Evaluation Criteria Statistics

**Aspects**
- Range (effective and theoretic).
- Direction (minimize or maximize).
- Statistics (e.g., mean, median, mode, standard deviation).

**Benefit**
Plans can be positioned along an absolute continuum of evaluation values.
Dominant Plans

**Definition**

A plan, \( p \), is **dominant** to a set of other plans, \( P^- \) in respect to two or more plan evaluators \( e_1...k \in E \) when

\[
\forall e \in E, p^- \in P^- [e(p) \geq e(p^-)]
\]
Plan Evaluation Visualization

- Option Tree
  - Option 1.1
  - Option 1.10
  - Option 1.100
  - Option 1.1

- Option Graph
  - network bandwidth usage
  - transportation cost vs. network bandwidth usage
  - est. plan execution time vs. network bandwidth usage
  - network hcp count vs. network bandwidth usage

- Option Ranges
  - ED detection accuracy: 4.52 ± 3.82
  - network bandwidth usage
  - transportation cost
  - est. plan execution time
  - network hcp count

- Mixed-initiative UI

Network-Aware Automated Planning

Kyle Usbeck

Drexel University
Experiment: Plan Evaluation Benchmarking

- Location 1
- Location 2
- Location 3
- Location 4

- Node 1
- Node 2
- Node 3
- Node 4
- Node 5

- Camera 1
- Camera 2
## Plan Evaluation Benchmarking

<table>
<thead>
<tr>
<th>Action</th>
<th>Providing Hosts</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHYSICALMOVE</td>
<td>all</td>
</tr>
<tr>
<td>ACQUIRECAMERA</td>
<td>all</td>
</tr>
<tr>
<td>TAKEPHOTO</td>
<td>all</td>
</tr>
<tr>
<td>GETOLDPHOTO</td>
<td>all</td>
</tr>
<tr>
<td>RELEASECAMERA</td>
<td>all</td>
</tr>
<tr>
<td>CHECKFORIEDAT</td>
<td>1, 2, and 5</td>
</tr>
<tr>
<td>MANUALSEARCH</td>
<td>1, 2, 3, and 4</td>
</tr>
<tr>
<td>PHOTOGRAPHICSEARCH</td>
<td>3, 4, and 5</td>
</tr>
<tr>
<td>PHOTOARCHIVE</td>
<td>5</td>
</tr>
<tr>
<td>PHOTOCOMPARE</td>
<td>4 and 5</td>
</tr>
<tr>
<td>RESULTREPORT</td>
<td>2 and 5</td>
</tr>
</tbody>
</table>
Plan Evaluation Benchmarking

<table>
<thead>
<tr>
<th>Camera</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera 1</td>
<td>3.2 MP</td>
</tr>
<tr>
<td>Camera 2</td>
<td>8.0 MP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Node</th>
<th>Speed (max mph)</th>
<th>Transportation Cost ($ per mile)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node 1</td>
<td>30</td>
<td>6.0</td>
</tr>
<tr>
<td>Node 2</td>
<td>40</td>
<td>6.5</td>
</tr>
<tr>
<td>Node 3</td>
<td>20</td>
<td>5.1</td>
</tr>
<tr>
<td>Node 4</td>
<td>10</td>
<td>4.9</td>
</tr>
<tr>
<td>Node 5</td>
<td>45</td>
<td>6.2</td>
</tr>
</tbody>
</table>
Plan Evaluation Benchmarking Results

Each planning algorithm ran in I-Plan for five minutes.

### Plan Evaluations

<table>
<thead>
<tr>
<th>Plan Type</th>
<th>$\omega_H$</th>
<th>Bandwidth</th>
<th>IED Acc.</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-Plan Default</td>
<td>0.949</td>
<td>0.759</td>
<td>291.4</td>
<td>8216</td>
</tr>
<tr>
<td>Random</td>
<td>1.647</td>
<td>1.476</td>
<td>177.9</td>
<td>7220</td>
</tr>
<tr>
<td>Guided</td>
<td>1.916</td>
<td>1.141</td>
<td>392.6</td>
<td>14050</td>
</tr>
</tbody>
</table>

### Dominant Plans

<table>
<thead>
<tr>
<th>Search Strategy</th>
<th>% Dominant Plans Produced</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-Plan Default</td>
<td>7.4%</td>
</tr>
<tr>
<td>Random</td>
<td>33.3%</td>
</tr>
<tr>
<td>Guided</td>
<td>59.3%</td>
</tr>
</tbody>
</table>
Each planning algorithm ran in I-Plan for five minutes.

### Plan Evaluations

<table>
<thead>
<tr>
<th>Plan</th>
<th>$\omega_H$</th>
<th>Bandwidth</th>
<th>IED Acc.</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-Plan Default</td>
<td>0.949</td>
<td>0.759</td>
<td>291.4</td>
<td>8216</td>
</tr>
<tr>
<td>Random</td>
<td>1.647</td>
<td>1.476</td>
<td>177.9</td>
<td>7220</td>
</tr>
<tr>
<td>Guided</td>
<td>1.916</td>
<td>1.141</td>
<td>392.6</td>
<td>14050</td>
</tr>
</tbody>
</table>

### Dominant Plans

<table>
<thead>
<tr>
<th>Search Strategy</th>
<th>% Dominant Plans Produced</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-Plan Default</td>
<td>7.4%</td>
</tr>
<tr>
<td>Random</td>
<td>33.3%</td>
</tr>
<tr>
<td>Guided</td>
<td>59.3%</td>
</tr>
</tbody>
</table>
Outline

1. Introduction
   - Motivation
   - Background
   - Approach

2. Formalization
   - Problem Statement

3. Technical Approach
   - Planning Agents
   - Execution Agents
   - Monitoring Agents
   - Mixed-initiative UI

4. Experiments
   - Plan Evaluation Benchmarking
   - Network-Aware Agent Combinations
   - Discussion
### Experiment: Network-Aware Agent Combinations

<table>
<thead>
<tr>
<th>Agent</th>
<th>Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Random</td>
</tr>
<tr>
<td></td>
<td>Domain-independent (I-Plan)</td>
</tr>
<tr>
<td></td>
<td>Guided</td>
</tr>
<tr>
<td>Execution</td>
<td>Naïve</td>
</tr>
<tr>
<td></td>
<td>Reactive</td>
</tr>
<tr>
<td></td>
<td>Proactive</td>
</tr>
<tr>
<td>Monitoring</td>
<td>Data-driven</td>
</tr>
<tr>
<td></td>
<td>Analytic</td>
</tr>
<tr>
<td></td>
<td>(none)</td>
</tr>
</tbody>
</table>
Experimental Setup

- Multi-objective Optimization (MOO) Function.
- Implemented agents with I-X and I-Plan.
- Network emulation.
- Mobility models.

**MOO function**

\[
\text{MOO}(p_I) = \text{IEDDetectAcc}(p_I) + 3 \times \text{TranspCost}(p_I) + 5 \times \text{ExecTime}(p_I) + \text{LinkQuality}(p_I) + \text{BandwidthUse}(p_I)
\]
Experimental Setup

- Multi-objective Optimization (MOO) Function.
- Implemented agents with I-X and I-Plan.
- Network emulation.
- Mobility models.

**MOO function**

$$\text{MOO}(p_I) = \text{IEDDetectAcc}(p_I) + 3 \times \text{TranspCost}(p_I) + 5 \times \text{ExecTime}(p_I) + \text{LinkQuality}(p_I) + \text{BandwidthUse}(p_I)$$
Experimental Setup

- Multi-objective Optimization (MOO) Function.
- Implemented agents with I-X and I-Plan.
- Network emulation.
- Mobility models.

**MOO function**

\[
\text{MOO}(p_I) = \text{IEDDetectAcc}(p_I) + 3 \times \text{TranspCost}(p_I) + 5 \times \text{ExecTime}(p_I) + \text{LinkQuality}(p_I) + \text{BandwidthUse}(p_I)
\]
Experimental Setup

- Multi-objective Optimization (MOO) Function.
- Implemented agents with I-X and I-Plan.
- Network emulation.
- Mobility models.

**MOO function**

\[
\text{MOO}(p_1) = \text{IEDDetectAcc}(p_1) + 3 \times \text{TranspCost}(p_1) + 5 \times \text{ExecTime}(p_1) + \text{LinkQuality}(p_1) + \text{BandwidthUse}(p_1)
\]
- Boeing’s Common Open Research Emulator.
- FreeBSD network stack emulation.
- Simple Multicast Forwarding (SMF).
- Open Shortest Path First (OSPF).
- Boeing’s Common Open Research Emulator.
- FreeBSD network stack emulation.
- Simple Multicast Forwarding (SMF).
- Open Shortest Path First (OSPF).
- Boeing’s Common Open Research Emulator.
- FreeBSD network stack emulation.
- Simple Multicast Forwarding (SMF).
- Open Shortest Path First (OSPF).
Boeing’s Common Open Research Emulator.

FreeBSD network stack emulation.

Simple Multicast Forwarding (SMF).

Open Shortest Path First (OSPF).
Mobility Models

**Purpose**

- Dictate geographical node locations.
- Dynamic $\omega_H$.

**Mobility Patterns**

1. Local.
2. Static.
3. Dynamic.
Mobility Models

Purpose
- Dictate geographical node locations.
- Dynamic $\omega_H$.

Mobility Patterns
1. Local.
2. Static.
3. Dynamic.
## Mobility Models

### Purpose
- Dictate geographical node locations.
- Dynamic $\omega_H$.

### Mobility Patterns
1. **Local.**
2. **Static.**
3. **Dynamic.**
4. **Partition-merge.**
Mobility Models

Purpose

- Dictate geographical node locations.
- Dynamic $\omega_H$.

Mobility Patterns

1. Local.
2. Static.
3. Dynamic.
Mobility Models

Purpose

- Dictate geographical node locations.
- Dynamic \( \omega \).

Mobility Patterns

1. Local.
2. Static.
3. Dynamic.
Mobility Models

Purpose
- Dictate geographical node locations.
- Dynamic $\omega_H$.

Mobility Patterns
1. Local.
2. Static.
3. Dynamic.
Mobility Models

**Purpose**

- Dictate geographical node locations.
- Dynamic $\omega_H$.

**Mobility Patterns**

1. Local.
2. Static.
3. Dynamic.
Mobility Models

Purpose
- Dictate geographical node locations.
- Dynamic
  \( \omega \)
  \( H \).

Mobility Patterns
1. Local.
2. Static.
3. Dynamic.
Domain-independent Plan

- checkForIEDAt location1
- manualSearch node1 location1
- physicalMove node1 location1
- conductScan node1 location1
- physicalMove node1 location1
- reportResults node1 location1
- checkForIEDAt location2
- manualSearch node1 location2
- physicalMove node1 location2
- conductScan node1 location2
- physicalMove node1 location2
- reportResults node2 location2
- checkForIEDAt location3
- manualSearch node1 location3
- physicalMove node1 location3
- conductScan node1 location3
- physicalMove node1 location3
- reportResults node2 location3
Random Plan

cHECKFORIEDAt location1
photographICSearch node3 location1
physicalMoveToCamera node3 cameral
acquireCamera node3 location1 cameral
physicalMove node3 location1
getoIDPhoto node5 to photo-0
takePhoto node3 location1 cameral to photo-1
cOMPAREPhotos node4 photo-1 photo-0
reportResults node2 location1
cHECKFORIEDAt location2
manualSearch node1 location2
physicalMove node1 location2
conductScan node1 location2
physicalMove node2 location2
reportResults node2 location2
cHECKFORIEDAt location3
manualSearch node1 location3
physicalMove node1 location3
conductScan node1 location3
physicalMove node2 location3
reportResults node2 location3
Guided Plan

checkForIEDAt location1
photographicSearch node5 location1
physicalMoveToCamera node5 camera2
acquireCamera node5 location1 camera2
physicalMove node5 location1
getOldPhoto node5 to photo-0
takePhoto node5 location1 camera2 to photo-1
comparePhotos node5 photo-1 photo-0
reportResults node5 location1
checkForIEDAt location2
manualSearch node3 location2
physicalMove node3 location2
conductScan node3 location2
physicalMove node5 location2
reportResults node5 location2
checkForIEDAt location3
manualSearch node4 location3
physicalMove node4 location3
conductScan node4 location3
physicalMove node2 location3
reportResults node2 location3
Local Results: Mean Time

- Network not a factor.
- Network-awareness did not hurt.
Local Results: Mean IED Detection Accuracy

Ideal values of IED detection accuracy.
Network disruptions adversely effect plan execution times.

Guided was 16.7% faster than I-Plan and 28.8% faster than random in part-merge.
Execution Agent Effectiveness

Planning Agent: domain-independent (I-Plan default)

- Naïve agent has the lowest IED detection accuracy and exec. time.
- Reactive and proactive agents achieved ideal IED detection accuracies.
Planning Agent: random

- Naïve agent failed most often.
- Proactive agent finished considerably faster than reactive.
Execution Agent Effectiveness

Planning Agent: guided (network-aware)

- Naïve agent failed most often.
- The guided algorithm advice significantly helped the execution agent.
Execution Agent Performance

- Proactive agent uses slightly more network transmissions under connected mobility patterns.
- Under part-merge, the proactive agent sent fewer than half as many packets as the reactive agent.
Analytic Monitoring Agent

- High percentage of false-positives.
- Communication errors → incorrect residuals.
- Active monitor.

Analytic monitors are less-suitable for network-centric domains.
Monitoring Agent Comparisons

Analytic Monitoring Agent

- High percentage of false-positives.
- Communication errors $\rightarrow$ incorrect residuals.
- Active monitor.

- Analytic monitors are less-suitable for network-centric domains.
Monitoring Agent Comparisons

Analytic Monitoring Agent

- High percentage of false-positives.
- Communication errors → incorrect residuals.
- Active monitor.

- Analytic monitors are less-suitable for network-centric domains.
Monitoring Agent Comparisons

Analytic Monitoring Agent

- High percentage of false-positives.
- Communication errors → incorrect residuals.
- Active monitor.

- Analytic monitors are less-suitable for network-centric domains.
Normal execution:

![Graph showing data-driven monitoring agent during dynamic link mobility. The graph plots the total number of data packets processed against elapsed time (in seconds) for two categories: number of data packets and TCP retransmit timeouts. The y-axis represents the total number of data packets processed and TCP retransmit timeouts, ranging from 0 to 160 and 0 to 7, respectively. The x-axis represents elapsed time (in seconds), ranging from 0 to 250. The graph shows a linear increase in the number of data packets processed and a slight fluctuation in TCP retransmit timeouts over time.](image-url)
Data-driven Monitoring Agent

Network disconnection:

![Graph showing data packets and TCP retransmit timeouts over time](image)

- In 54 trials...
- 9.25% false-positives (type I error).
- 1.85% false-negatives (type II error).

Kyle Usbeck

Network-Aware Automated Planning

71/75
Data-driven Monitoring Agent

Network disconnection:

In 54 trials...

- 9.25% false-positives (type I error).
- 1.85% false-negatives (type II error).
Outline

1. Introduction
   - Motivation
   - Background
   - Approach

2. Formalization
   - Problem Statement

3. Technical Approach
   - Planning Agents
   - Execution Agents
   - Monitoring Agents
   - Mixed-initiative UI

4. Experiments
   - Plan Evaluation Benchmarking
   - Network-Aware Agent Combinations
   - Discussion
Main Contributions

1. Qualitatively-different plan generation:
   - Qualitatively different plans over a range of plan evaluation criteria.
   - Visualizing plan evaluations.

2. Network-aware agents:
   - Network-aware planning agent.
   - Network-aware execution agents.
   - Network-aware monitoring agents.
Main Contributions

1. Qualitatively-different plan generation:
   - Qualitatively different plans over a range of plan evaluation criteria.
   - Visualizing plan evaluations.

2. Network-aware agents:
   - Network-aware planning agent.
   - Network-aware execution agents.
   - Network-aware monitoring agents.
Main Contributions

1. Qualitatively-different plan generation:
   - Qualitatively different plans over a range of plan evaluation criteria.
   - Visualizing plan evaluations.

2. Network-aware agents:
   - Network-aware planning agent.
   - Network-aware execution agents.
   - Network-aware monitoring agents.
Main Contributions

1. Qualitatively-different plan generation:
   - Qualitatively different plans over a range of plan evaluation criteria.
   - Visualizing plan evaluations.

2. Network-aware agents:
   - Network-aware planning agent.
   - Network-aware execution agents.
   - Network-aware monitoring agents.
Main Contributions

1. Qualitatively-different plan generation:
   - Qualitatively different plans over a range of plan evaluation criteria.
   - Visualizing plan evaluations.

2. Network-aware agents:
   - Network-aware planning agent.
   - Network-aware execution agents.
   - Network-aware monitoring agents.
Main Contributions

1. Qualitatively-different plan generation:
   - Qualitatively different plans over a range of plan evaluation criteria.
   - Visualizing plan evaluations.

2. Network-aware agents:
   - Network-aware planning agent.
   - Network-aware execution agents.
   - Network-aware monitoring agents.
Main Contributions

1. Qualitatively-different plan generation:
   - Qualitatively different plans over a range of plan evaluation criteria.
   - Visualizing plan evaluations.

2. Network-aware agents:
   - Network-aware planning agent.
   - Network-aware execution agents.
   - Network-aware monitoring agents.
Future Work

- Knowledge-based monitoring agents.
- Incorporate the effects of planning actions into heuristics.
Future Work

- Knowledge-based monitoring agents.
- Incorporate the effects of planning actions into heuristics.
Acknowledgements

- **Advisor:**
  - Dr. William C. Regli

- **Committee:**
  - Dr. Rachel Greenstadt
  - Dr. Ani Hsieh

- **Conceptual Contributors:**
  - Prof. Austin Tate
  - Dr. Gerhard Wickler
  - Jeff Dalton

- **Critiquors:**
  - Ilya Braude
  - Matt Chase
  - Patrick Freestone
  - Joe Kopena
  - Duc Nguyen
  - Rob Lass
  - Evan Sultanik

- **Family & Friends**

- **Co-workers**

- **\LaTeX, Vim, opensource software**
IED Detection Accuracy and Bandwidth Usage

The graph illustrates the relationship between IED Detection Accuracy (%) and Network Bandwidth Usage (Mbps) across different planning strategies: Guided, I-Plan Default, and Random. The data points show a general trend where higher IED Detection Accuracy is associated with lower Network Bandwidth Usage, although the specific patterns and correlations would require further analysis.

Network-Aware Agent Combinations

Introduction
Formalization
Technical Approach
Experiments
Plan Evaluation Benchmarking
Discussion

Kyle Usbeck
Network-Aware Automated Planning
IED Detection Accuracy and Execution Time

![Graph showing IED Detection Accuracy vs. Plan Execution Time]

- Guided
- I-Plan Default
- Random

Plan Evaluation Benchmarking
Network-Aware Agent Combinations
Discussion
Network Bandwidth Usage and Execution Time

Network Bandwidth Usage vs. Plan Execution Time

Guided
I-Plan Default
Random

Network Bandwidth Usage (Mbps)
Plan Execution Time (sec)

Network Bandwidth Usage (Mbps)
Plan Execution Time (sec)
Network Hops and IED Detection Accuracy

![Network Hops vs. IED Detection Accuracy Graph]

**Guided**
- I-Plan Default
- Random

Number of Network Hops (full data path) vs. IED Detection Accuracy (%)
Network Hops vs. Network Bandwidth Usage

- Guided
- I-Plan Default
- Random

Network Bandwidth Usage (Mbps) vs. Number of Network Hops (full data path)

Kyle Usbeck
Network-Aware Automated Planning
Network Hops and Execution Time

![Network Hops vs. Plan Execution Time](chart.png)

- Guided
- I-Plan Default
- Random

Plan Execution Time (sec) vs. Number of Network Hops (full data path)

- Kyle Usbeck
- Network-Aware Automated Planning
Plan Eval. Benchmarking Execution Time Distribution

Guided, Random, iPlan execution times distribution.
Plan Eval. Benchmarking IED Detect. Acc. Distribution

![Bar chart showing distribution of guided, random, and iplan methods for IED detection accuracy. The y-axis represents the number of methods, and the x-axis represents different values. The chart illustrates the performance of each method across varying scenarios.]
Plan Eval. Benchmarking Link Quality Distribution

![Graph showing link quality distribution for guided, random, and iplan methods]

- **Guided**
- **Random**
- **Iplan**
Plan Eval. Benchmarking Bandwidth Usage Distribution

![Plan Evaluation Benchmarking Graph]

- **Guided**
- **Random**
- **iplan**

Plan Evaluation Benchmarking Bandwidth Usage Distribution

Guided

Random

Iplan

3.1 3.7 4.3 4.8 5.4 6.0 6.6 7.1 7.7 8.3

0 5 10 15 20 25 30

Kyle Usbeck

Network-Aware Automated Planning