CARLETON UNIVERSITY

SYSC 5104 — Assignment 1

**Student:** RACHID FAHMI (101415381)

Email: RachidFahmi@cmail.carleton.ca

**Date:** February 19, 2026

# Modeling UAV Swarm Communication with Relay Handoff using DEVS

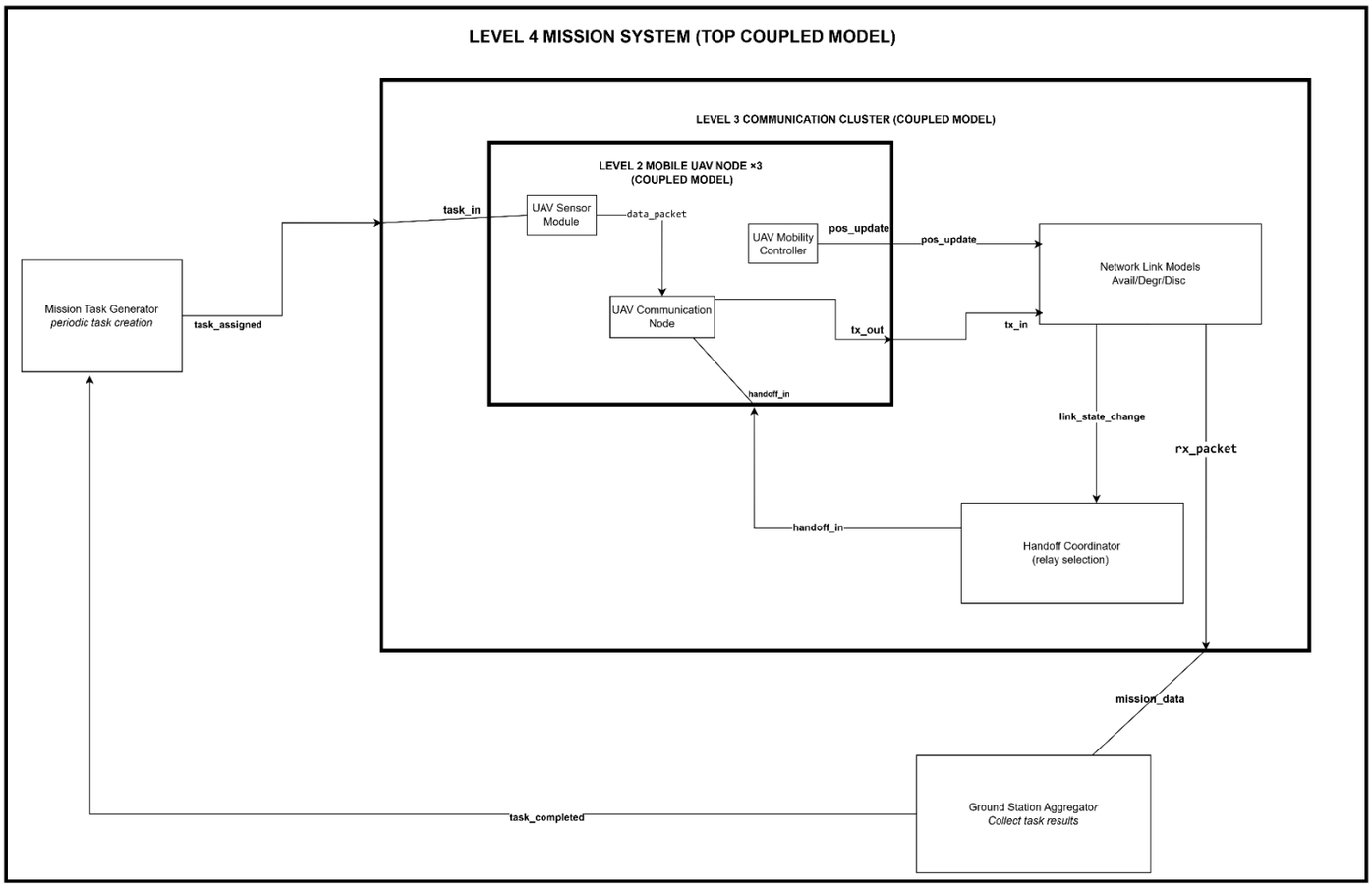
## UAV Swarm Surveillance — Conceptual Model

### Problem Description

UAV swarms used in surveillance need to keep sending data back to a ground station even as they move around. The problem is that the further a UAV gets from the station, the worse the link gets — and eventually it drops entirely. The question this model tries to answer is simple: if we detect that a link has gone down and automatically reroute traffic through a relay UAV, does that actually help? And by how much compared to doing nothing?

---

### Model Structure




The system is built as a four-level DEVS hierarchy.

**Level 4 — Mission System (Top Coupled Model)**
The top level ties everything together. The Mission Task Generator sends tasks down into the cluster, the cluster processes them, and when enough data has been collected, the Ground Station reports back up that the mission task is done.

**Level 3 — Communication Cluster (Coupled Model)**
This is where most of the interesting behavior lives. Three UAV nodes operate alongside their respective Network Link Models and a single Handoff Coordinator. The cluster handles all communication between UAVs and the ground, including detecting link failures and triggering relay handoffs.

**Level 2 — Mobile UAV Node (Coupled Model, ×3)**
Each UAV is its own coupled model containing three components: a Sensor Module, a Mobility Controller, and a Communication Node. From the outside, the node takes in task assignments and handoff commands, and produces transmitted packets and position updates.

**Level 1 — Atomic Models**
Seven atomic components, each handling one specific piece of behavior.

### Atomic Models

**Mission Task Generator**
Sends out a task_assigned every 60 seconds on a fixed schedule. It does receive task_completed on its input port, but that signal doesn't change its behavior — the periodic timer keeps running regardless. One state: active, Ta = 60 s.

- I: task_completed — O: task_assigned

**UAV Sensor Module**
Waits in idle until a task_in arrives, then spends 5 seconds sensing before emitting a data_packet and going back to idle. If another task_in comes in while it's already sensing, it's dropped — there's no buffer.

- I: task_in — O: data_packet

**UAV Mobility Controller**
Has no inputs at all. Every 5 seconds it emits a pos_update with the UAV's current (x, y) position. It never changes state — just keeps broadcasting position autonomously for the whole simulation.

- I: (none) — O: pos_update

**UAV Communication Node**
Takes a data_packet and spends 1 second transmitting it before emitting tx_out. While it's transmitting, any new packet that arrives gets dropped. handoff_in is the exception — it can arrive at any time and quietly updates where the next packet gets sent (direct vs. relay), without interrupting the current transmission.

- I: data_packet, handoff_in — O: tx_out

**Network Link Model**
The most complex model in the system. Tracks the UAV's distance from the ground station and maintains one of three stable states: Available, Degraded, or Disconnected, based on thresholds d1 and d2. Two transient zero-time states — Forwarding and Notify — handle packet forwarding and link change notifications. Packets that arrive while disconnected are silently dropped.

- I: tx_in, pos_update — O: rx_packet, link_state_change

**Handoff Coordinator**
Watches for link_state_change events. Available and Degraded events are filtered out completely. When a Disconnected event comes in, the model waits 1 second then sends a handoff_in to the affected UAV's Communication Node before going back to monitoring.

- I: link_state_change — O: handoff_in

**Ground Station Aggregator**
Counts incoming mission_data packets. Once three arrive, it immediately outputs task_completed (Ta = 0, no delay), resets its counter, and goes back to waiting for the next batch.

- I: mission_data — O: task_completed

**Key Interactions**

- **Task flow:** Generator emits task_assigned → sent to all 3 UAVs as task_in → each UAV starts a sensing cycle
- **Data flow:** Sensor produces data_packet → passed to Communication Node → transmitted via Network Link → received at Ground Station → task_completed sent back to Generator
- **Mobility:** Each Mobility Controller independently pushes pos_update every 5 s → Network Link Model uses this to update distance and switch states if needed
- **Link failure and handoff:** Network Link emits link_state_change(Disconnected) → Handoff Coordinator issues handoff_in after 1 s → Communication Node switches to relay mode
- **Relay simplification:** Since building a full two-hop relay path would require extra ports and couplings beyond this assignment's scope, relay mode is modeled as a threshold relaxation on the affected Network Link Model (d1 = 200 m, d2 = 400 m). The change takes effect on the next pos_update. This is an acknowledged simplification and is documented in the experimental framework.

**Assumptions**

- UAVs move at constant speed along fixed trajectories throughout the simulation
- Link quality is determined entirely by distance from the ground station: $d = \sqrt{x^2 + y^2}$, with the station at (0, 0)
- Three UAVs are used — enough to produce meaningful cluster behavior without making the coupling tables unmanageable
- Handoff only triggers on full disconnection; a degraded link on its own is not enough
- The relay mechanism is a threshold relaxation rather than a true forwarding path — this is a deliberate design choice documented as a limitation
- Communication Nodes hold one packet at a time with no queuing
- Threshold updates from handoff take effect on the next pos_update event, not instantaneously, to stay DEVS-compliant

**Objective**

The model compares two configurations under the same failure conditions: one where UAVs only use direct links, and one where a handoff to relay mode is triggered on disconnection. The metrics being tracked are packet delivery ratio, number of completed mission tasks, and how often handoffs are activated. The point is to see whether even a simplified relay mechanism — with its 1-second delay and threshold-based approximation — makes a measurable difference in how many packets actually reach the ground station.

## Formal DEVS Specification of the UAV Swarm Model

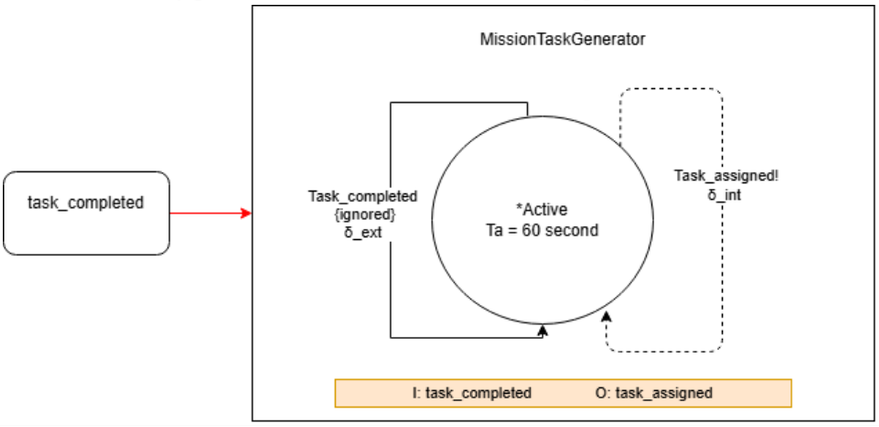
**1. Model Architecture Refinement**

The Part I model defined a four-level DEVS hierarchy: Mission System, Communication Cluster, Mobile UAV Nodes, and their internal components. For Part II, the Privacy Filter and Coverage Monitor were removed, bringing the atomic model count from nine to seven and keeping only the components directly relevant to the handoff scenario. The swarm was fixed at three UAVs to avoid dynamic coupling. The Communication Node was reduced to a single-packet buffer with a binary mode flag, and the Network Link was limited to three UAV-to-Ground links and up to two relay links, with state changes driven by position events from the Mobility Controller.

**2. Atomic Model Specifications**

**2.1 Mission Task Generator**

**Role:** Periodically generates surveillance tasks and assigns them to UAVs.

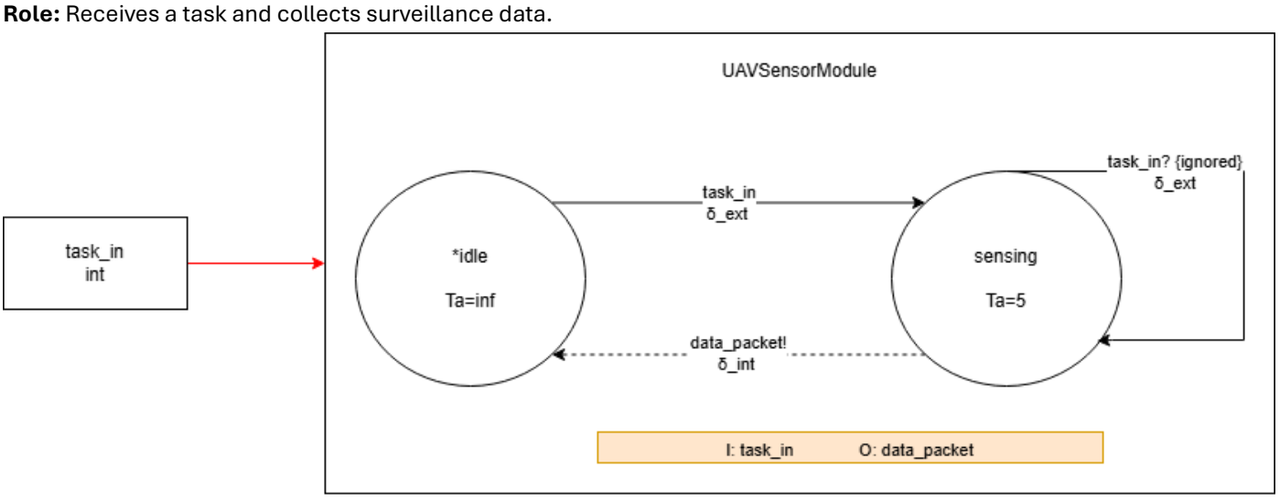



$X = \{ task\_completed \}$
$Y = \{ task\_assigned \}$
$S = \{ active \}$
$Initial\ state: active$
$ta(active) = T\_task = 60\ s$
$\delta ext(active, e, x) = active \quad // ignore\ all\ external\ inputs$
$\delta int(active) = active$
$\lambda(active) = task\_assigned$

**Behavior**: Fires every 60 seconds regardless of external input. task_completed is received but ignored.

**2.2 UAV Sensor Module**

**Role:** Receives a task and collects surveillance data.



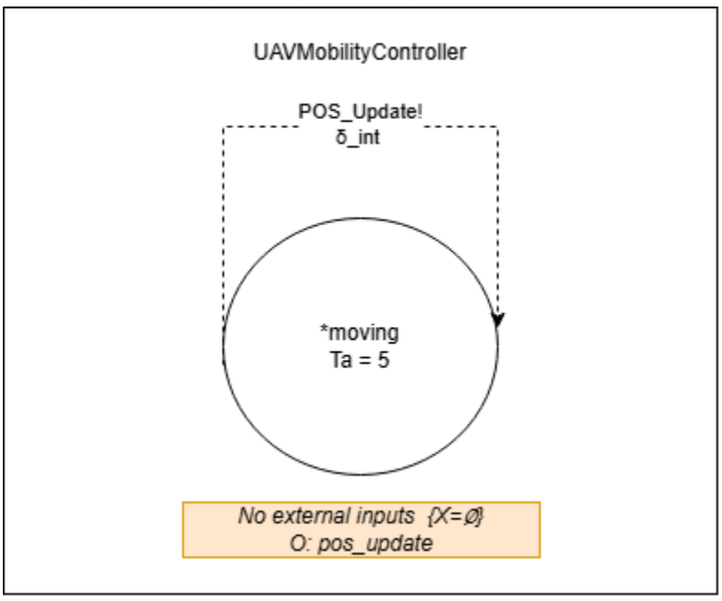
$X = \{ task\_in \}$
$Y = \{ data\_packet \}$
$S = \{ idle, sensing \}$
$Initial\ state: idle$
$ta(idle) = \infty$
$ta(sensing) = T\_sense = 5\ s$
$\delta ext(idle, e, task\_in) = sensing$
$\delta ext(sensing, e, x) = sensing \quad // ignore\ all\ inputs\ while\ sensing$
$\delta int(sensing) = idle$
$\lambda(sensing) = data\_packet$

**Behavior:** On task_in, enters sensing for 5 seconds, outputs data_packet, then resets to idle. Any input during sensing is ignored.

---

**2.3 UAV Mobility Controller**

**Role:** Updates UAV position periodically and notifies the Network Link Model of each position change.



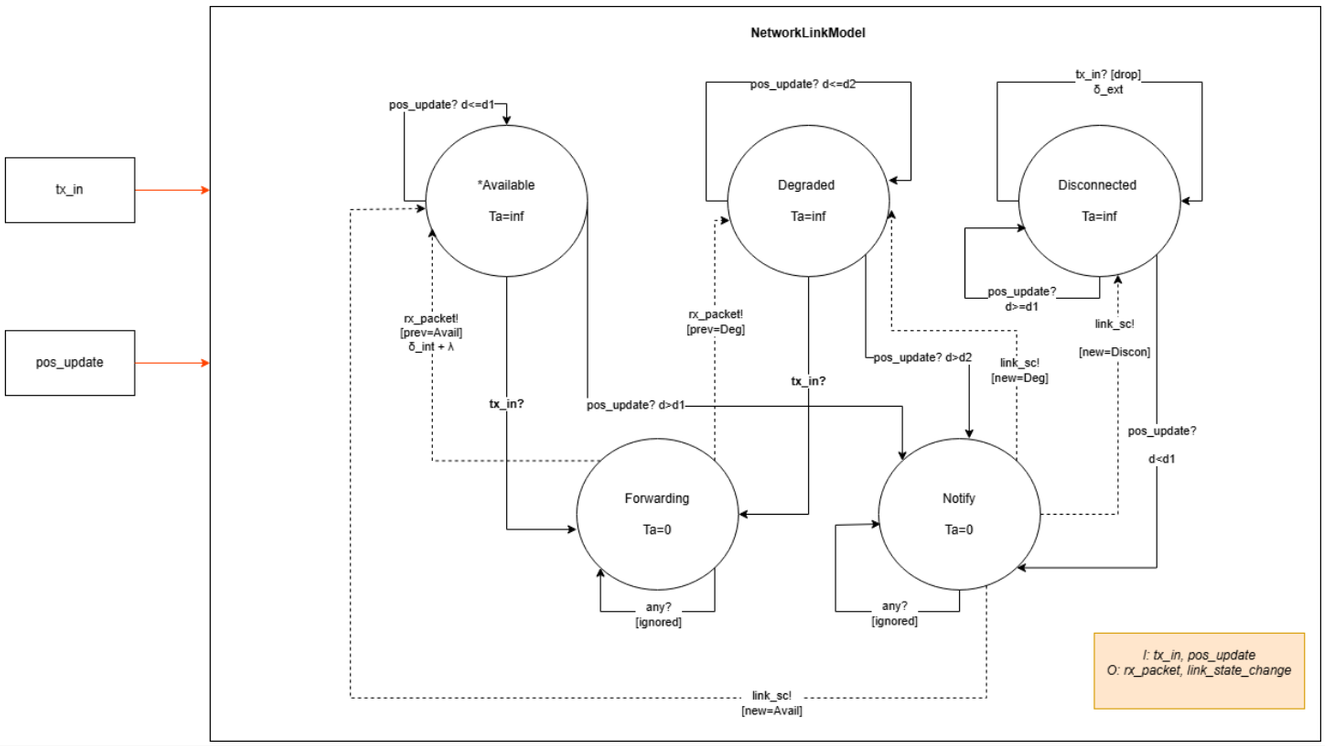

$X = \emptyset$
$Y = \{ pos\_update \}$
$S = \{ moving \}$
$Initial\ state: moving$
$ta(moving) = T\_move = 5\ s$
$\delta int(moving) = moving$
$\lambda(moving) = pos\_update$

**Behavior:** Fires pos_update every 5 seconds. No external inputs.

**2.4 Network Link Model**

**Role:** Represents a UAV-to-ground communication link. Reacts to position updates to change link state and forwards or drops packets accordingly.

**Initial state**: Available

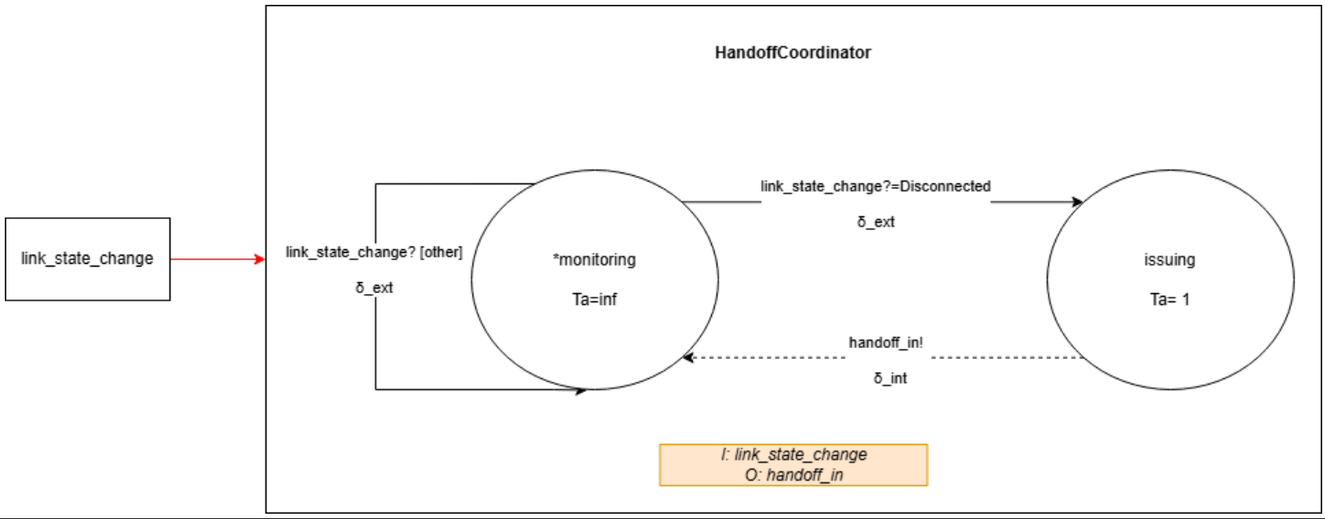



$X = \{ tx\_in, pos\_update \}$
$Y = \{ rx\_packet, link\_state\_change \}$
$S = \{ Available, Degraded, Disconnected, Forwarding, Notify \}$
$// Packet\ handling \rightarrow enter\ Forwarding\ (zero-time), \lambda\ fires, \delta int\ returns$
$\delta ext(Available, \quad e, tx\_in) = Forwarding\ (prev\_state = Available)$
$\delta ext(Degraded, \quad e, tx\_in) = Forwarding\ (prev\_state = Degraded)$
$\delta ext(Disconnected, e, tx\_in) = Disconnected\ // drop\ silently, no\ state\ change, no\ output$
$// Threshold\ crossed \rightarrow enter\ Notify\ (zero-time), \lambda\ fires, \delta int\ returns$
$\delta ext(Available, \quad e, pos\_update \rightarrow d > d1) = Notify\ (new\_state = Degraded)$
$\delta ext(Degraded, \quad e, pos\_update \rightarrow d > d2) = Notify\ (new\_state = Disconnected)$
$\delta ext(Disconnected, e, pos\_update \rightarrow d < d1) = Notify\ (new\_state = Available)$
$// No\ threshold\ crossed \rightarrow stay, no\ output$
$\delta ext(Available, \quad e, pos\_update \rightarrow d \leq d1) = Available$
$\delta ext(Degraded, \quad e, pos\_update \rightarrow d \leq d2) = Degraded$
$\delta ext(Disconnected, e, pos\_update \rightarrow d \geq d1) = Disconnected$
$// Ignore\ concurrent\ inputs\ in\ transient\ states$
$\delta ext(Forwarding, e, x) = Forwarding$
$\delta ext(Notify, \quad e, x) = Notify$
$ta(Available) \quad = \infty$
$ta(Degraded) \quad = \infty$
$ta(Disconnected) = \infty$
$ta(Forwarding) \quad = 0$
$ta(Notify) \quad = 0$
$\delta int(Forwarding) = prev\_state \quad // return\ to\ stable\ link\ state$
$\delta int(Notify) \quad = new\_state \quad // transition\ to\ new\ link\ state$
$\lambda(Forwarding) = rx\_packet$
$\lambda(Notify) \quad = link\_state\_change(new\_state)$

**Behavior**: Each pos_update recomputes d. If a threshold is crossed, the link changes state and emits link_state_change. Packets are forwarded as rx_packet in Available or Degraded, and dropped silently in Disconnected.

**2.5 Handoff Coordinator**

**Role:** Triggers a relay handoff when a link disconnects.

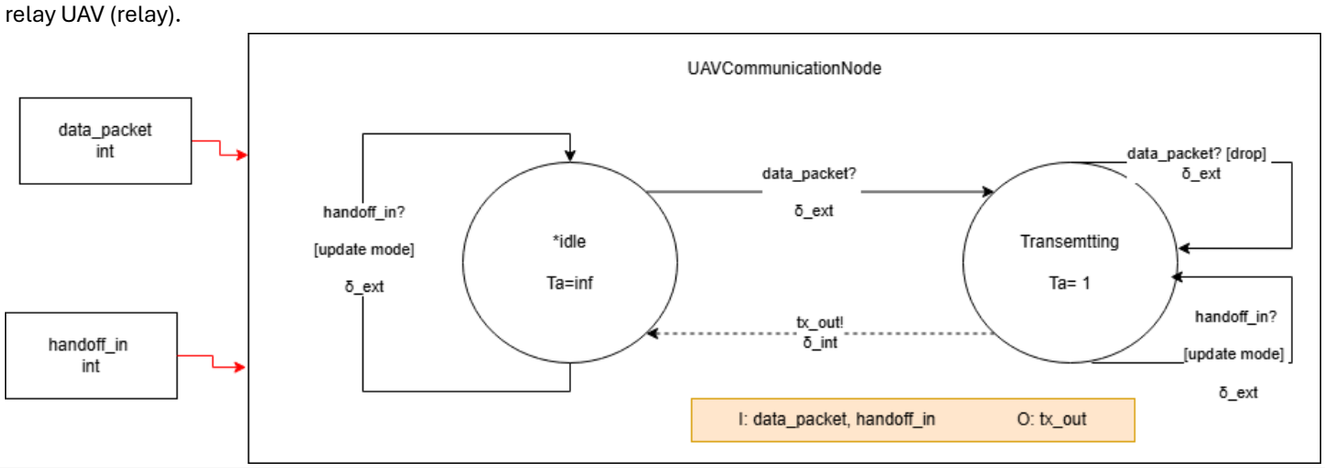



$X = \{ link\_state\_change \}$
$Y = \{ handoff\_in \}$
$S = \{ monitoring, issuing \}$
$Initial\ state: monitoring$
$Internal\ variable: relay\_target\_id$
$ta(monitoring) = \infty$
$ta(issuing) = T\_handoff = 1\ s$
$\delta ext(monitoring, e, link\_state\_change = Disconnected) = issuing$
$\delta ext(monitoring, e, other) = monitoring$
$\delta int(issuing) = monitoring$
$\lambda(issuing) = handoff\_in$

**Behavior:** On Disconnected, moves to issuing, waits 1 second, outputs handoff_in, then resets to monitoring.

### 2.6 UAV Communication Node

**Role:** Sends data packets to the Network Link Model. Mode selects whether the destination is the ground station (direct) or a relay UAV (relay).

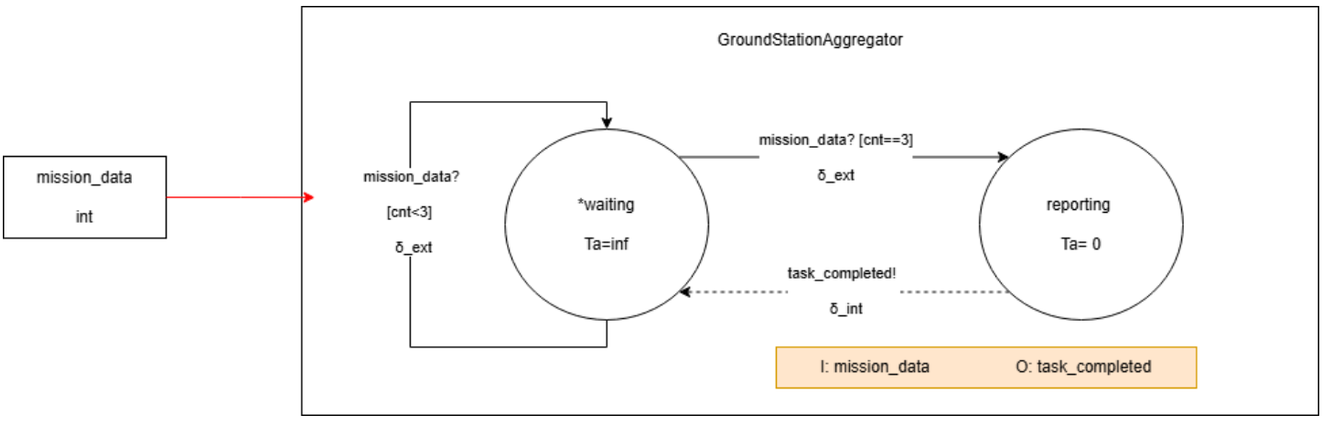



$X = \{ data\_packet, handoff\_in \}$
$Y = \{ tx\_out \}$
$S = \{ idle, transmitting \}$
$Initial\ state: idle$
$Internal\ variables: mode \in \{direct, relay\},\quad relay\_target\_id$
$ta(idle) = \infty$
$ta(transmitting) = T\_tx = 1\ s$
$\delta ext(idle, e, data\_packet) = transmitting$
$\delta ext(idle, e, handoff\_in) = idle$
$\delta ext(transmitting, e, data\_packet) = transmitting$
$\delta ext(transmitting, e, handoff\_in) = transmitting$
$\delta int(transmitting) = idle$
$\lambda(transmitting) = tx\_out(packet, destination)$

**Behavior:** Buffers one packet at a time. On data_packet, transmits for 1 second and outputs tx_out with destination set by mode — ground station if direct, relay UAV if relay. Extra packets during transmission are dropped. handoff_in updates mode and relay_target_id without interrupting the current state.

### 2.7 Ground Station Aggregator

**Role:** Receives mission data from UAVs and signals task completion when all expected packets arrive.




$X = \{ mission\_data \}$
$Y = \{ task\_completed \}$
$S = \{ waiting, reporting \}$
$Initial\ state: waiting$
$Internal\ variables: received\_count, \quad expected\_count = 3$
$ta(waiting) = \infty$
$ta(reporting) = 0$
$\delta ext(waiting, e, mission\_data):$
$\quad received\_count++$
$\quad if\ received\_count == expected\_count$
$\quad\quad \rightarrow reporting$
$\quad else$
$\quad\quad \rightarrow waiting$
$\delta int(reporting):$
$\quad received\_count = 0$
$\quad \rightarrow waiting$
$\lambda(reporting) = task\_completed$

**Behavior**: Counts incoming packets. When all 3 arrive, outputs task_completed instantly, resets the counter, and waits for the next round.

**3. Coupled Model Specifications**

There are three coupled models in the hierarchy: Level 2 (Mobile UAV Node), Level 3 (Communication Cluster), and Level 4 (Mission System — top model).

**3.1 Level 2 — Mobile UAV Node**

**Components:** UAV Sensor Module (SM), UAV Mobility Controller (MC), UAV Communication Node (CN)

**Ports:**

$X\_L2 = \{ task\_in \}$
$Y\_L2 = \{ tx\_out, pos\_update \}$

**EIC**

| From (external) | To (internal) |
|---|---|
| L2.task_in | SM.task_in |

**IC**

| From | To |
|---|---|
| SM.data_packet | CN.data_packet |

**EOC**

| From (internal) | To (external) |
|---|---|
| CN.tx_out | L2.tx_out |
| MC.pos_update | L2.pos_update |

**3.2 Level 3 — Communication Cluster**

**Components:** Mobile UAV Node ×3 (UAV1, UAV2, UAV3), Network Link Model ×3 (NL1, NL2, NL3), Handoff Coordinator (HC)

*Note: handoff_in is broadcast to all UAVs in this model. Targeted per-UAV handoff routing is deferred to future work.*

**Ports:**

$X\_L3 = \{ task\_in \}$
$Y\_L3 = \{ rx\_packet \}$

**EIC**

| From (external) | To (internal) |
|---|---|
| L3.task_in | UAV1.task_in |
| L3.task_in | UAV2.task_in |
| L3.task_in | UAV3.task_in |

**IC**

| From | To | Purpose |
|---|---|---|
| UAV1.tx_out | NL1.tx_in | UAV1 packets → Link1 |
| UAV2.tx_out | NL2.tx_in | UAV2 packets → Link2 |
| UAV3.tx_out | NL3.tx_in | UAV3 packets → Link3 |
| UAV1.pos_update | NL1.pos_update | UAV1 position → Link1 |
| UAV2.pos_update | NL2.pos_update | UAV2 position → Link2 |
| UAV3.pos_update | NL3.pos_update | UAV3 position → Link3 |
| NL1.link_state_change | HC.link_state_change | Link1 state → Handoff |
| NL2.link_state_change | HC.link_state_change | Link2 state → Handoff |
| NL3.link_state_change | HC.link_state_change | Link3 state → Handoff |
| HC.handoff_in | UAV1.handoff_in | Handoff → UAV1 comm node |
| HC.handoff_in | UAV2.handoff_in | Handoff → UAV2 comm node |
| HC.handoff_in | UAV3.handoff_in | Handoff → UAV3 comm node |

**EOC**

| From (internal) | To (external) |
|---|---|
| NL1.rx_packet | L3.rx_packet |
| NL2.rx_packet | L3.rx_packet |
| NL3.rx_packet | L3.rx_packet |

**3.3 Level 4 — Mission System (Top Model)**

**Components:** Mission Task Generator (MTG), Communication Cluster (L3), Ground Station Aggregator (GSA)

**Ports:**

$X\_L4 = \emptyset \quad (top\ model\ -\ no\ external\ inputs)$

$Y\_L4 = \emptyset \quad (top\ model\ -\ no\ external\ outputs)$

**EIC**

None (X_L4 = ∅)

**IC**

| From | To | Purpose |
|---|---|---|
| MTG.task_assigned | L3.task_in | Tasks → UAV swarm |
| L3.rx_packet | GSA.mission_data | Received packets → aggregator |
| GSA.task_completed | MTG.task_completed | Completion feedback → generator |

**EOC**

$None\ (Y\_L4 = \emptyset)$

## Cadmium Implementation and Simulation Results

### Experimentation Strategy

The experimentation strategy follows an incremental approach. Testing begins with individual atomic models executed in isolation to verify state transitions, timing behavior, and output correctness. It then progresses to coupled simulations to evaluate system-level interactions. Three experiments are defined: (1) atomic unit tests for each model, (2) a baseline coupled simulation without handoff to measure packet loss under link degradation, and (3) a full system simulation with handoff enabled for comparison. This layered approach ensures correctness at each level before introducing additional coupling complexity.

### Specification Refinements During Implementation

A few adjustments were made during implementation to better align the code with the intended system behavior. The HandoffMsg was changed from a single integer to a structured message {affected_uav_id, relay_target_id}. This made it clear which UAV should react to a handoff event and avoided ambiguity when multiple UAVs were present.

The initial sigma value in the UAVMobilityController was also corrected from 0 to 5 seconds so that the first pos_update occurs at t = 5, consistent with the Ta = 5 s specification in the state diagram.

Additionally, input parsers for PosUpdateMsg and LinkStateMsg were updated to validate comma-separated formatting. This prevents malformed input lines from being silently accepted.

For Experiment 2, a separate coupled model (CommunicationClusterNoHandoff) was created. Instead of disabling the handoff logic at the top level, the HandoffCoordinator was removed directly from the cluster wiring. This ensured that the baseline experiment was structurally isolated in a clean and consistent way.

### 2. Atomic Model Test Results (Experiment 1)

Each atomic model was driven individually via an IEStream input file and output logged to logs/exp1x_<model_name>.log.

**Exp 1a — MissionTaskGenerator**: task_assigned output confirmed at t=60, 120, 180 (Ta=60 s). External task_completed inputs received and correctly ignored — sigma did not reset. PASS.

$60; 18; mission\_task\_generator; task\_assigned; 1$
$60; 18; mission\_task\_generator; ; sigma: 60$
$120; 18; mission\_task\_generator; task\_assigned; 1$
$180; 18; mission\_task\_generator; task\_assigned; 1$

**Exp 1b — UAVSensorModule**: On task_in, sensing:true entered immediately, data_packet emitted at t=5 (Ta=5 confirmed). Mid-sense inputs silently dropped. Returns to idle correctly after output. PASS.

$0; 6; sensor\_module; ; sensing: true\ sigma: 5$
$5; 6; sensor\_module; data\_packet; 1$
$5; 6; sensor\_module; ; sensing: false\ sigma: inf$

**Exp 1c — NetworkLinkModel**: Starting Available. On pos_update crossing d1, transient Notify state entered (sigma:0), link_state_change(Degraded) emitted, transitions to stable Degraded. On further crossing d2, transitions to Disconnected. Packets in Disconnected state silently dropped — no rx_packet output. PASS.

$20; 4; nl0; ; phase: Notify\ stable: Available\ sigma: 0$
$20; 4; nl0; link\_state\_change; 0, Degraded$
$20; 4; nl0; ; phase: Degraded\ stable: Degraded\ sigma: inf$
$45; 4; nl0; link\_state\_change; 0, Disconnected$

**Exp 1d — GroundStationAggregator**: Counter increments on each mission_data. When count ≥ expected (3), transitions to reporting (Ta=0), emits task_completed immediately, resets counter to 0, returns to waiting. PASS.

$66; 1; ground\_station\_aggregator; ; count: 3\ reporting: true\ sigma: 0$
$66; 1; ground\_station\_aggregator; task\_completed; 1$
$66; 1; ground\_station\_aggregator; ; count: 0\ reporting: false\ sigma: inf$

**Exp 1e — HandoffCoordinator**: link_state_change(Degraded) filtered — stays in monitoring. link_state_change(Disconnected) triggers issuing (Ta=1). After 1 second, handoff_in emitted, returns to monitoring. PASS.

45; 3; $handoff\_coordinator$; ; $issuing$: $true$ $affected$: 0 $relay$: 1 $sigma$: 1
46; 3; $handoff\_coordinator$; $handoff\_in$; $affected$: 0 $relay$: 1
46; 3; $handoff\_coordinator$; ; $issuing$: $false$ $affected$: −1 $relay$: −1 $sigma$: $inf$

**Exp 1f — UAVCommunicationNode**: data_packet triggers 1-second transmission; tx_out emitted at Ta=1. Concurrent data_packet during transmission dropped. handoff_in updates mode flag in both idle and transmitting states without interrupting current transmission. PASS.

5; 7; $comm\_node$; ; $transmitting$: $true$ $mode$: $direct$ $relay\_target$: −1 $pending$: 1 $sigma$: 1
6; 7; $comm\_node$; $tx\_out$; 1
6; 7; $comm\_node$; ; $transmitting$: $false$ $mode$: $direct$ $relay\_target$: −1 $pending$: 1 $sigma$: $inf$

**Exp 1g — ``````````````````**: First pos_update at t=5 (initial sigma=5). Subsequent outputs at t=10, 15, 20, 25, 30, 5-second period confirmed throughout. No external inputs. PASS.

5; 7; $mobility\_controller$; $pos\_update$; 40,0
10; 7; $mobility\_controller$; $pos\_update$; 55,0
15; 7; $mobility\_controller$; $pos\_update$; 70,0

**3. Experiment 2 — Baseline: No Handoff**

*Configuration:* MissionSystemNoHandoff using CommunicationClusterNoHandoff — HandoffCoordinator removed entirely from cluster wiring. UAV2 starts at (30,0) with vx=10 m/step; UAV0 and UAV1 static. NL0 thresholds: d1=50 m, d2=100 m.
*Key log trace (exp2_baseline_no_handoff.log):*

$t = 20$ $nl0$: $Notify \rightarrow Degraded$ $(UAV2\ pos\_update: 60{,}0 \rightarrow d = 60 > d1 = 50)$
$t = 45$ $nl0$: $Notify \rightarrow Disconnected$ $(UAV2\ pos\_update: 110{,}0 \rightarrow d = 110 > d2 = 100)$
$[no\ handoff\ issued - HC\ not\ present]$
$t = 60$ $task\_assigned$ $(round\ 1)$
$t = 65$ $data\_packet \times 3$; $nl0\ comm\_node\ transmitting: true\ mode: direct$
$t = 66$ $nl1\ rx\_packet, nl2\ rx\_packet - nl0\ drops\ UAV2\ packet$ $(Disconnected)$
$GSA\ count: 2$
$t = 120$ $task\_assigned$ $(round\ 2)$
$t = 126$ $nl1\ rx\_packet, nl2\ rx\_packet \rightarrow GSA\ count: 4 \geq 3 \rightarrow task\_completed, reset\ to\ 0$
$t = 180$ $task\_assigned$ $(round\ 3)$
$t = 186$ $nl1\ rx\_packet, nl2\ rx\_packet \rightarrow GSA\ count: 2$ $(no\ task\_completed)$

*Metrics: 9 packets sent, 6 received, PDR =* ***67%****, 1 task_completed event (t=126), 0 handoff activations.*

***4. Experiment 3 — With Handoff***

*Configuration: Full MissionSystem with CommunicationCluster including HandoffCoordinator. Identical UAV positions and thresholds.*
*Key log trace (exp3_with_handoff.log):*

$t = 20$ $nl0$: $Notify \rightarrow Degraded$ $(same\ position\ trajectory)$
$t = 45$ $nl0$: $Notify \rightarrow Disconnected$
$t = 46$ $HC$: $issuing: true\ affected: 0\ relay: 1$ $(fires\ after\ 1\ s\ delay)$
$handoff\_in(affected: 0, relay: 1)\ broadcast\ to\ all\ UAVs$
$UAV2\ comm\_node: mode: relay\ relay\_target: 1 \leftarrow mode\ updated$
$UAV0, UAV1: mode: direct\ unchanged$ $(id\ filter\ working)$
$t = 60$ $task\_assigned$ $(round\ 1)$
$t = 65$ $UAV2\ comm\_node\ transmitting: true\ mode: relay\ relay\_target: 1$
$t = 66$ $nl1\ rx\_packet, nl2\ rx\_packet - nl0\ still\ Disconnected, drops\ UAV2\ packet$
$GSA\ count: 2$
$t = 120$ $task\_assigned$ $(round\ 2)$
$t = 126$ $nl1\ rx\_packet, nl2\ rx\_packet \rightarrow GSA\ count: 4 \geq 3 \rightarrow task\_completed, reset\ to\ 0$
$t = 180$ $task\_assigned$ $(round\ 3)$
$t = 186$ $nl1\ rx\_packet, nl2\ rx\_packet \rightarrow GSA\ count: 2$

*Metrics:* 9 packets sent, 6 received, PDR = **67%**, 1 task_completed event (t=126), **1 handoff activation.**

**5. Comparative Analysis**

| Metric | Exp 2 (No Handoff) | Exp 3 (With Handoff) |
|---|---|---|
| Packets sent | 9 | 9 |

| Metric | Exp 2 (No Handoff) | Exp 3 (With Handoff) |
|---|---|---|
| Packets received at GSA | 6 | 6 |
| PDR | 67% | 67% |
| task_completed events | 1 (t=126) | 1 (t=126) |
| Handoff activations | 0 | 1 |
| mode:relay switches | 0 | 1 |

The packet delivery ratio (PDR) and the mission completion time are the same in both experiments. This is expected. Even when relay mode is activated, it only changes an internal mode flag inside UAV2's Communication Node. The actual connections in the CommunicationCluster do not change. UAV2's packets are still sent through nl0. Since nl0 stays in the Disconnected state, those packets are dropped before they can reach the ground station.

Relay mode would only improve delivery if the cluster wiring actually changed when a handoff happened. In DEVS, this would require changing the port connections during simulation, which was not implemented in this model.

What the handoff experiment does show is that the control logic works correctly. The Network Link detects when the distance crosses the threshold and sends a link_state_change(Disconnected) message. The HandoffCoordinator receives this message, waits exactly one second (disconnect at t=45, handoff at t=46), and then sends a handoff_in message. The correct UAV updates its communication mode without interrupting any ongoing transmission. The signaling mechanism works as designed.

A future improvement would be to implement a true two-hop relay path so that packets can actually be rerouted through another UAV when a link fails.

**6. Changes Relative to Part II Specification**

| Change | Reason |
|---|---|
| HandoffMsg upgraded from int to {affected_uav_id, relay_target_id} | Needed explicit per-UAV targeting for mode flag filter |
| Initial sigma of UAVMobilityController: 0 → 5 | Diagram consistency (Ta=5 s) |
| Comma validation added to LinkStateMsg / PosUpdateMsg parsers | Robustness against malformed input |
| CommunicationClusterNoHandoff added as separate coupled model | Required to properly isolate Exp 2 baseline at the correct structural level |
| Relay implemented as CommNode mode flag only (not true rerouting) | Full dynamic recoupling not DEVS-compliant; documented as acknowledged limitation |