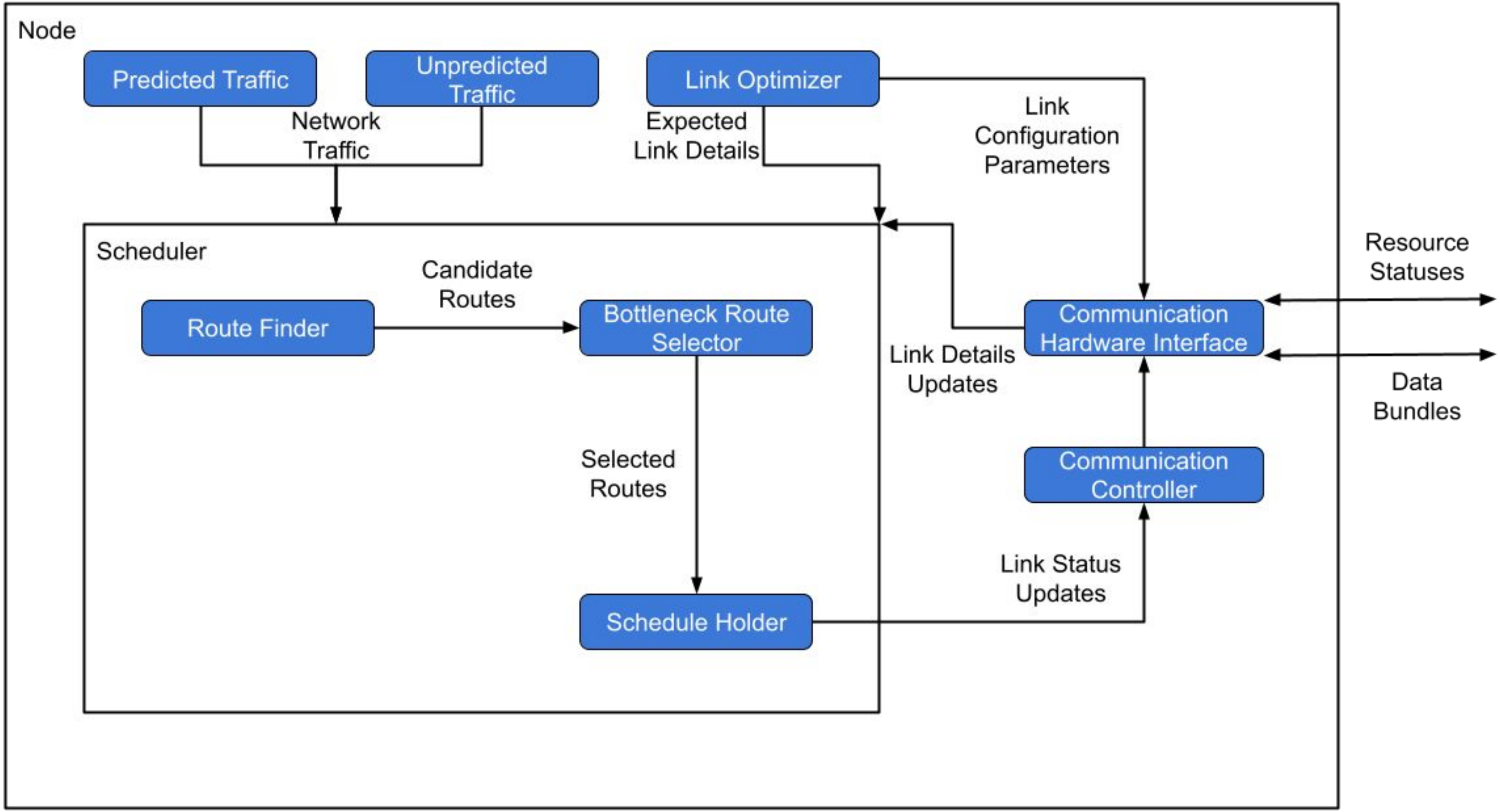
DREAMS DTN

June 21, 2023


Node
Predicted Traffic
Unpredicted Traffic
Link Optimizer
Network Traffic
Expected Link Details
Link Configuration Parameters
Scheduler
Candidate Routes
Route Finder
Bottleneck Route Selector
Selected Routes
Schedule Holder
Link Details Updates
Communication Hardware Interface
Resource Statuses
Data Bundles
Communication Controller
Link Status Updates

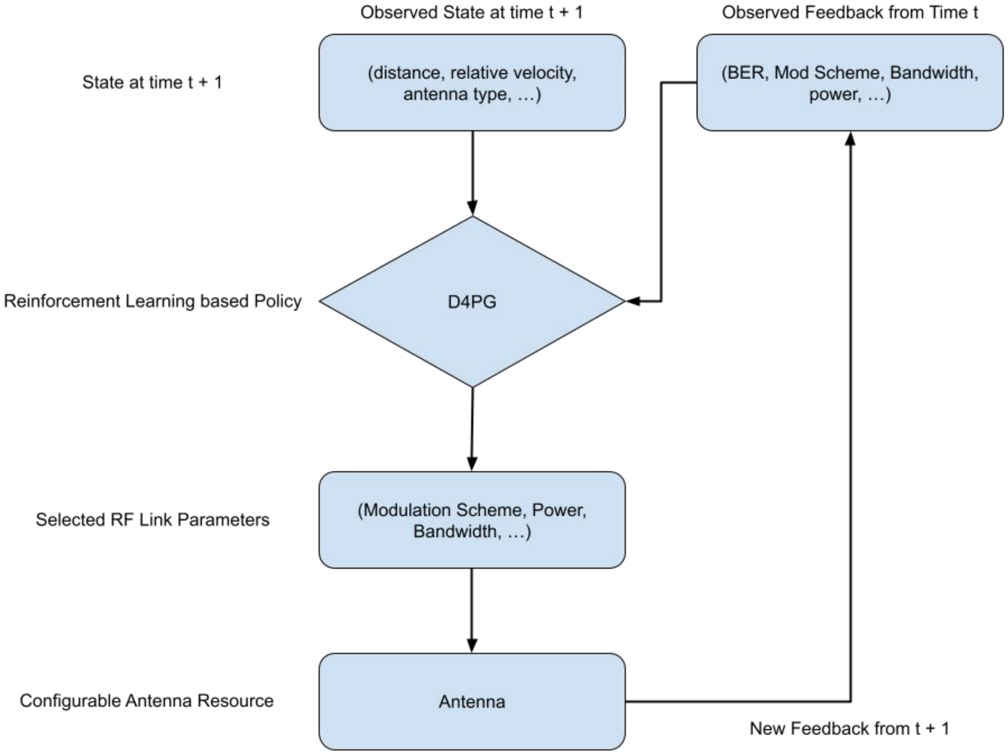
Overview of components

DREAMS

- Link optimization–optimizes links for quality and bandwidth
- Bundle prediction–predicts future scheduled and unscheduled network traffic
- Routing algorithm–routes bundles from one node to another
- Scheduling algorithm–schedules transmission times for bundles given routes
- Conflict resolution-resolves conflicts from distributed scheduling and reschedules bundles

Link Optimization

- Deep Reinforcement Learning
- Distributed Distribution Deterministic Policy Gradients (D4PG) to select link parameters (power, modulation scheme, coding scheme, symbol rate, etc.) to maximize Quality of Service
- Kratos's OpenSpace QuantumRadio Software Modem Capabilities
    - Tx/Rx modulation schemes, coding schemes, power, roll-off, frequency
    - Delay, varying matched filters, doppler effect, simple noise
    - Gives statistics such as EbN0, BER, effective bandwidth
- OpenAI Gym Environment using Kratos's API


Observed State at time t + 1
Observed Feedback from Time t
State at time t + 1
(distance, relative velocity, antenna type, …)
(BER, Mod Scheme, Bandwidth, power, …)
Reinforcement Learning based Policy
D4PG
Selected RF Link Parameters
(Modulation Scheme, Power, Bandwidth, …)
Configurable Antenna Resource
Antenna
New Feedback from t + 1

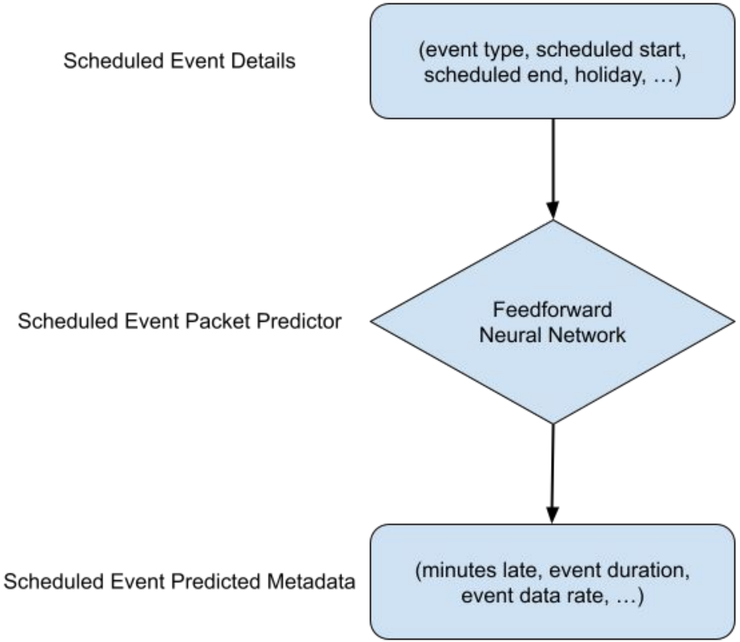
Bundles

- Bundles
    - Creation time
    - Source
    - Destination
    - Size
    - Deadline
    - Priority
    - Type
- Bundles -> SuperBundles with the same properties

Bundle Prediction

- Input: Traffic history (bundles)
- Neural network outputs, for a given scheduled event, the expected bundles and metadata
- Neural network outputs, the expected bundles not tied to any scheduled event and metadata
- Output: Predicted traffic, which is allocated in DREAMS's schedule

Scheduled Packet Predictor



Scheduled Event Details

(event type, scheduled start, scheduled end, holiday, …)

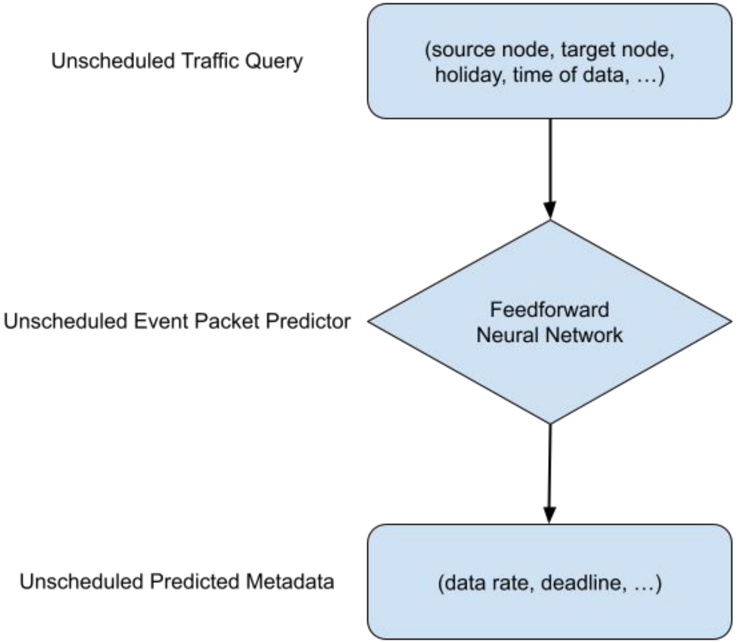
Scheduled Event Packet Predictor

Feedforward Neural Network

Scheduled Event Predicted Metadata

(minutes late, event duration, event data rate, …)

Unscheduled Packet Predictor



Unscheduled Traffic Query

(source node, target node, holiday, time of data, …)

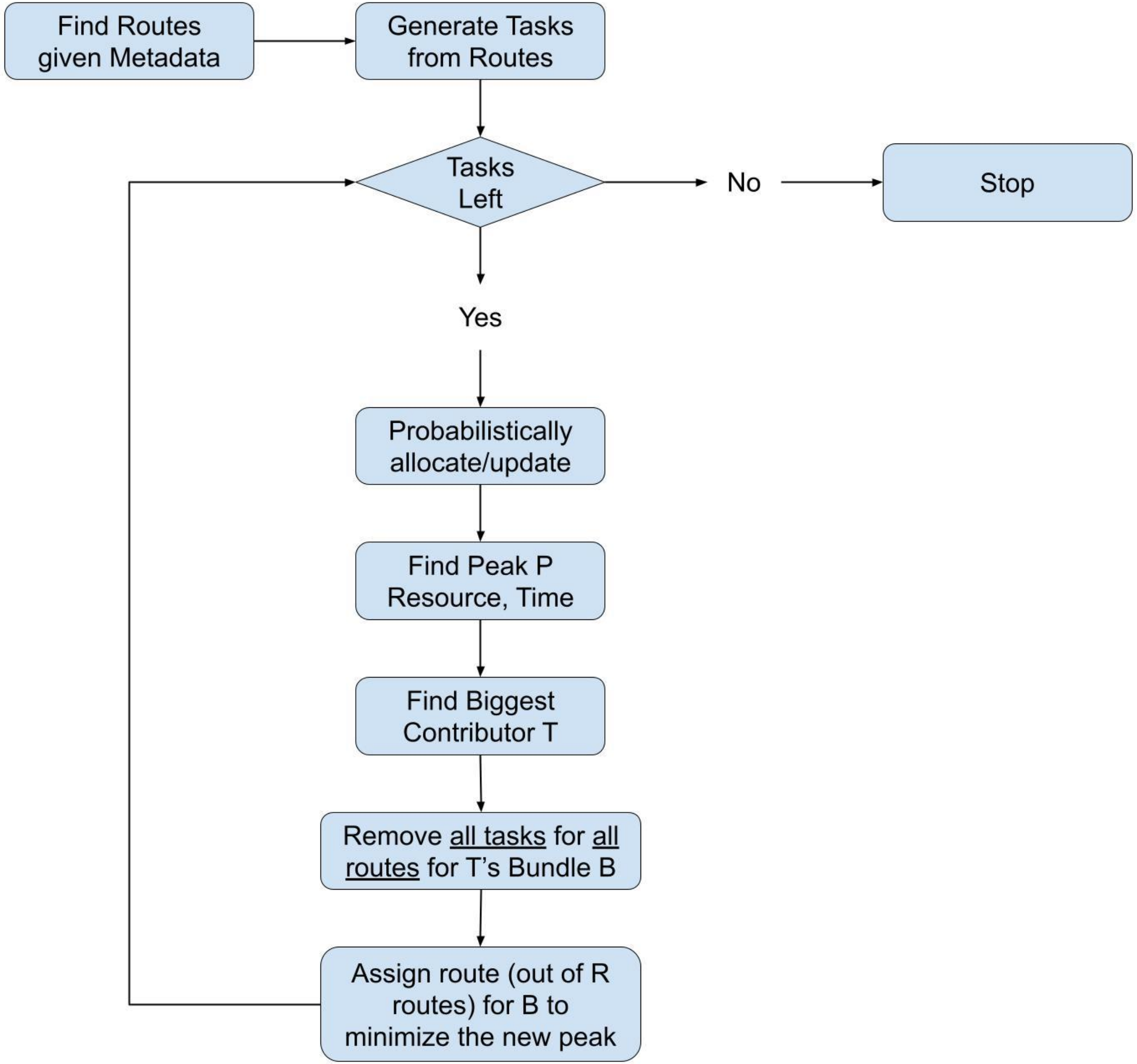
Unscheduled Event Packet Predictor

Feedforward Neural Network

Unscheduled Predicted Metadata

(data rate, deadline, …)

DREAMS Scheduling


Find Routes given Metadata
Generate Tasks from Routes
Tasks Left
No
Stop
Yes
Probabilistically allocate/update
Find Peak P Resource, Time
Find Biggest Contributor T
Remove all tasks for all routes for T’s Bundle B
Assign route (out of R routes) for B to minimize the new peak

Routing and Scheduling

- Two components:
  - Route Finder (only finds feasible (i.e. meet deadlines) routes)
    - N fast routes
    - M less-congested routes
    - B balanced routes
  - Bottleneck Scheduler (BNS)
    - Uses the BNA algorithm **to optimize overall throughput** by utilizing low congestion routes for each bundle when possible

Routing

- Viz-Dijkstra
  - Visibility aware Dijkstra's shortest path algorithm
  - Mark nodes/vertices with visit times, only consider visibilities after arrival time for next hop
  - Optimal if *FIFO* constraint on cost functions, e.g., cost proportional to time
  - Cost is a convex combination of a visibility congestion term and time
- Apply Viz-Dijkstra Temporal Graph
  - Vertices are now nodes in a slice of time (edges updated accordingly)
  - Better approximates optimal cost when FIFO violated (optimal for sufficiently small steps)
- Subset of nodes routing
  - NP-Hard (equivalent to Steiner Tree problem)
  - Approximate with Dijkstra's + post-pruning (tighter approximation schemes tend to be too slow)
- Multi-route finding
  - Adjust Congestion/Time cost for proposing low-congestion vs fast routes
  - Remove targeted edges and reroute to propose multiple decorrelated routes

Traditional Priority-Based Scheduling Algorithms

Sort tasks/bundles by priority, break ties with deadlines.

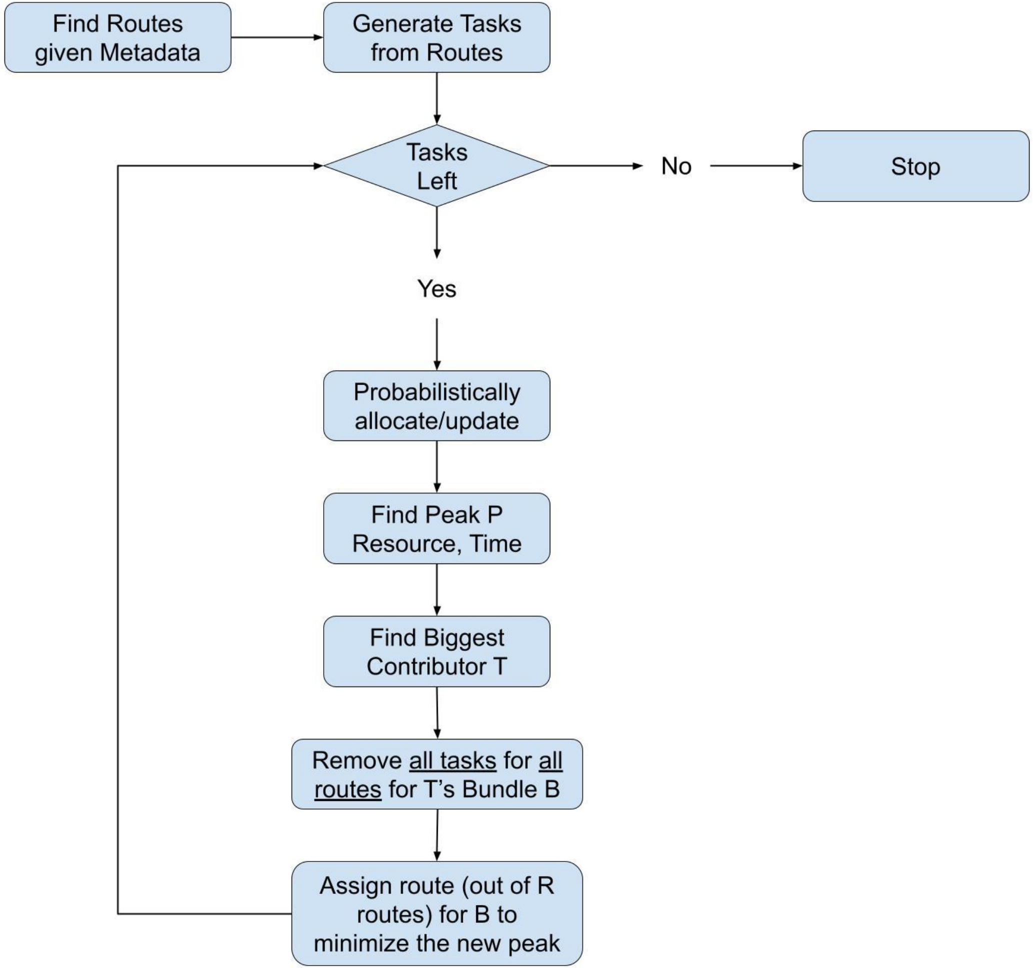
Schedule tasks / transmit bundles in the above order, greedily choose best (usually = fastest) route based on local information

First thing everyone thinks of; Simple; Easy to implement; Runs fast (linear)

But it is notoriously a very bad algorithm and produces severely suboptimal results

Problem is that being greedy and only using local information is not particularly good for overall schedule

BNS Best Route Selector


Find Routes given Metadata
Generate Tasks from Routes
Tasks Left
No
Stop
Yes
Probabilistically allocate/update
Find Peak P Resource, Time
Find Biggest Contributor T
Remove all tasks for all routes for T's Bundle B
Assign route (out of R routes) for B to minimize the new peak

Centralized Scheduling Scenario Performance

- Satellites and groundstations located on and around the Earth and Moon with realistic orbits
- 2 hour scenario
- 22 nodes, 300 links, 360 visibilities
- Routed: ~ 3562000 packets ->3562 bundles routed
- Java Runtime: 55 seconds (scales linearly with respect to scenario size in each direction) working on additional optimizations, including porting to an optimized C++ implementation
- Semi-Naive (fastest route only but included lookahead with BNS) Routed: 2792 / 3562= 78% (similar results in other scenarios)

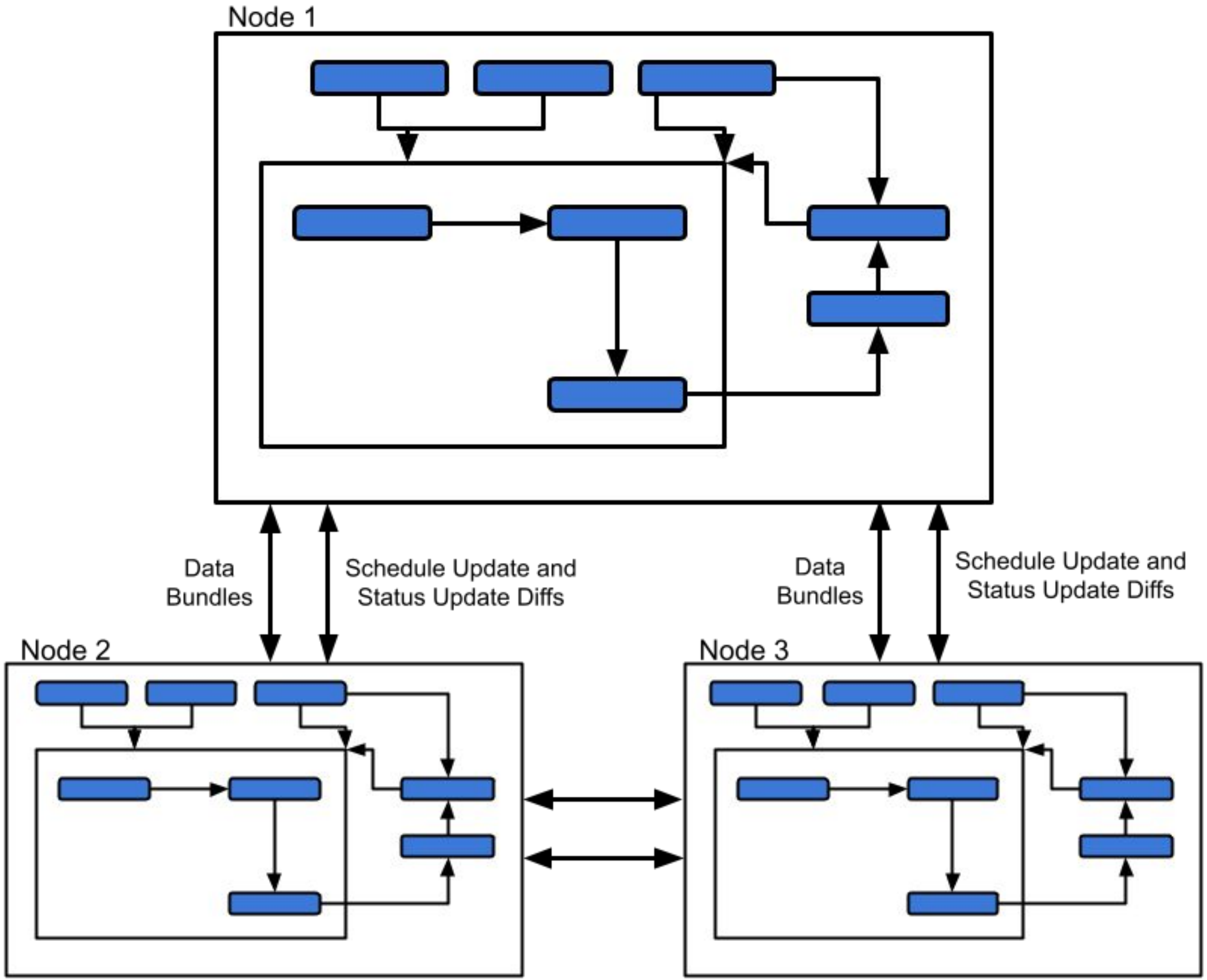
Distributed Scheduling

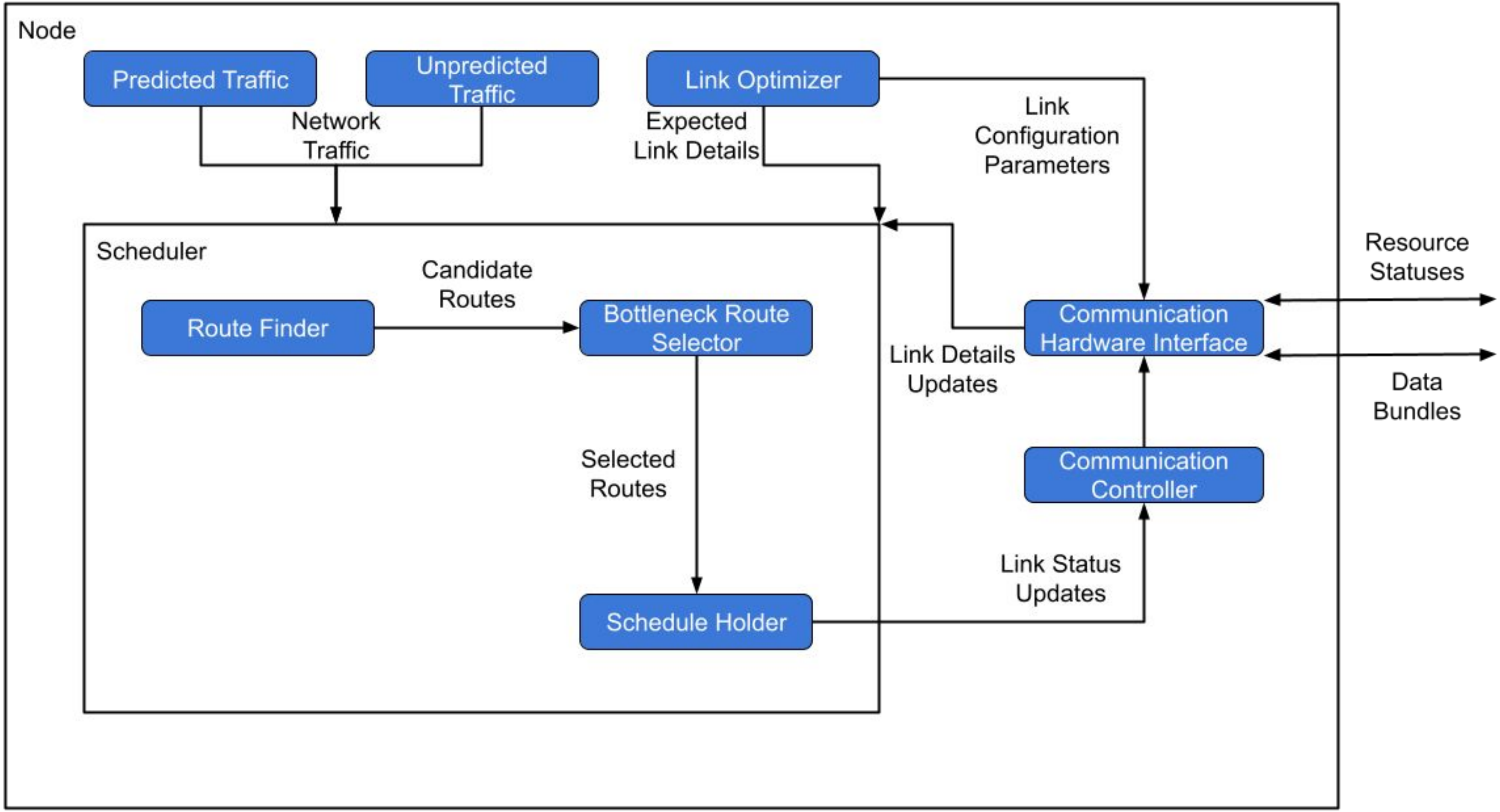
Distributed Simulation Environment

- CORE+EMANE
  - Realistic network emulator
  - Packet Generator feeds into CORE+EMANE as a custom "service"
- HDTN implemented as a service
- DREAMS fulfills router and scheduler roles in HDTN

Distributed Architecture

- DREAMS’s scheduling is inherently a distributed algorithm
- Deploy one instance of DREAMS on each node
- Use diffs for updating schedule to minimize network overhead


Node 1
Data Bundles
Schedule Update and Status Update Diffs
Data Bundles
Schedule Update and Status Update Diffs
Node 2
Node 3


Node
Predicted Traffic
Unpredicted Traffic
Link Optimizer
Network Traffic
Expected Link Details
Link Configuration Parameters
Scheduler
Candidate Routes
Route Finder
Bottleneck Route Selector
Resource Statuses
Communication Hardware Interface
Link Details Updates
Data Bundles
Selected Routes
Communication Controller
Link Status Updates
Schedule Holder

Distributed Conflict Handling

- Centralized scheduling for predicted traffic (and baseline connectivity tasks)
  - Expected to be a high percentage of the overall traffic
- Unpredicted traffic scheduled by originating node
  - Resource conflicts unavoidable
- Nodes only reschedule descendant tasks and handle immediate conflicts
- Diffs generated and propagated to the rest of the network