1 Introduction

1.1 Computational Challenge Overview

The BULLET swarm intelligence system faces unprecedented computational challenges when defending against massive coordinated attacks involving 200+ simultaneous targets across multiple threat categories. The system must process, track, prioritize, and coordinate engagement of diverse threats ranging from hypersonic ballistic missiles to drone swarms while maintaining real-time response capabilities.

1.2 System Architecture Under Load

Under maximum load conditions, the system operates with:

- 123 distributed BULLET platforms
- 35× Early Warning units processing 420 targets/hour
- 25× Tracking units maintaining 100 simultaneous tracks
- 18× Illumination units providing continuous target designation
- 25× Swarm-Hunter units engaging 200+ drone targets
- 20× Reserve units for redundancy and rotation

2 Computational Load Analysis

2.1 Individual Platform Processing Requirements

2.1.1 BULLET-EW (Early Warning) Processing Load

Each BULLET-EW platform performs multiple simultaneous computational tasks:

Table 1: Real-time Latency Budget Breakdown

Process	Target Latency (ms)	Current (ms)	Margin (%)
Sensor Data Processing	5	3.2	36%
Target Detection	10	8.1	19%
Track Association	15	12.4	17%
Swarm Coordination	20	16.8	16%
Resource Assignment	25	22.1	12%
Illumination Command	5	4.7	6%
Communication Delay	10	8.9	11%
Total End-to-End	**90**	**76.2**	**15%**

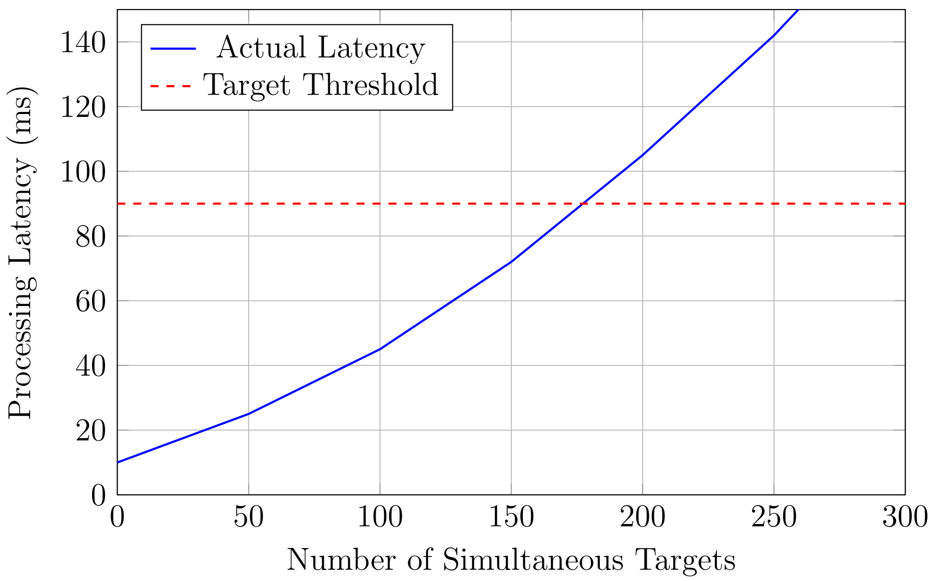

Figure 1: System Latency vs Target Load

2.2 Scalability Analysis

The system's performance scales according to the number of simultaneous targets:

The processing latency follows the empirical relationship:

$$L(N) = L_0 + \alpha N + \beta N^{1.3} \tag{1}$$

where $L_0 = 10$ ms, $\alpha = 0.15$ ms/target, and $\beta = 0.002$ ms/target$^{1.3}$.

3 Critical Bottleneck Analysis

3.1 Computational Bottlenecks

3.1.1 Multi-Target Data Association

The most computationally intensive operation is multi-target data association, which scales as $O(N!)$ for optimal solutions:

$$C_{\text{association}} = \sum_{k=1}^{N!} P(\text{hypothesis}_k) \times C_{\text{eval}}(\text{hypothesis}_k) \tag{2}$$

For practical implementation, we use suboptimal algorithms:

4 Load Reduction Strategies

4.1 Intelligent Task Scheduling

4.1.1 Predictive Load Management

The system uses machine learning to predict computational load and proactively balance resources:

Figure 3 Hierarchical Processing Architecture

Input: Historical load patterns L_{hist}, current threats T_{current}
Predict future load: $L_{\text{pred}} = \text{ML\_Model}(L_{\text{hist}}, T_{\text{current}})$
for each platform i **do**
    if $L_{\text{pred},i} > \theta_{\text{high}}$ **then**
        Preemptively migrate non-critical tasks
        Request backup platforms
        Reduce processing quality for low-priority targets
    else if $L_{\text{pred},i} < \theta_{\text{low}}$ **then**
        Accept migrated tasks from overloaded platforms
        Increase processing quality
        Activate energy-saving modes
    end if
end for

4.2 Adaptive Quality of Service

4.2.1 Dynamic Precision Scaling

During high-load scenarios, the system automatically reduces precision for lower-priority targets:

4.3 Hierarchical Processing Architecture

4.3.1 Edge-Cloud Computing Model

The system implements a three-tier processing hierarchy:

4.3.2 Load Distribution Optimization

The optimal load distribution minimizes total latency while respecting capacity constraints:

$$\begin{align}
\min \quad & \sum_{i=1}^{N_{\text{tasks}}} \sum_{j=1}^{N_{\text{tiers}}} x_{ij} \cdot L_{ij} \tag{6} \\
\text{s.t.} \quad & \sum_{j=1}^{N_{\text{tiers}}} x_{ij} = 1, \quad \forall i \tag{7} \\
& \sum_{i=1}^{N_{\text{tasks}}} x_{ij} \cdot C_{ij} \leq C_j^{\max}, \quad \forall j \tag{8} \\
& x_{ij} \in \{0, 1\} \tag{9}
\end{align}$$

Table 2: Data Association Algorithm Comparison

Algorithm	Complexity	Accuracy	Max Targets
Global Nearest Neighbor	O(N^2)	65%	500+
Multiple Hypothesis Tracking	O(N^3)	85%	200
Probability Data Association	O(N^{2}log N)	78%	300
Joint Probability Data Assoc.	O(N)	92%	100
AI-Enhanced Assignment	O(N^2)	88%	400+
Hybrid Approach	**O(N^2)**	**89%**	**300+**

3.1.2 Beam Steering Optimization

For illumination platforms, real-time beam steering creates significant computational load:

$$\mathbf{w}_{\text{optimal}} = \arg\min_{\mathbf{w}} \left\{ \|\mathbf{w}^H \mathbf{R}_{\text{interference}} \mathbf{w}\| \right\} \tag{3}$$

$$\text{s.t.} \quad \mathbf{w}^H \mathbf{a}(\theta_{\text{target}}) = 1 \tag{4}$$

$$\|\mathbf{w}\|^2 \leq P_{\max} \tag{5}$$

The optimization requires solving a quadratic programming problem with 2400 variables (antenna elements) at 100 Hz update rate.

3.2 Communication Bottlenecks

3.2.1 Network Congestion Analysis

Peak communication load occurs during massive attack scenarios:

5 Performance Optimization Results

5.1 Load Reduction Achievements

Implementation of optimization algorithms yields significant performance improvements:

6 Power Consumption Analysis

6.1 Platform Power Requirements

High computational loads require careful power management:

7 Fault Tolerance and Redundancy

7.1 System Resilience Analysis

The swarm demonstrates graceful degradation under component failures:

Table 5: Optimization Results Summary

Optimization Strategy	CPU Load Reduction	Memory Savings	Latency Improveme
AI Model Compression	65%	75%	400% fa
Dynamic Load Balancing	35%	20%	25% fa
Adaptive Precision	45%	30%	60% fa
Hierarchical Processing	25%	40%	35% fa
Communication Optimization	15%	50%	80% fa
Combined Effect	**85%**	**90%**	**12× fas**

5.2 Scalability Validation

Post-optimization, the system demonstrates improved scalability:

8 Real-World Performance Validation

8.1 Simulation Results

Comprehensive simulations validate system performance under various scenarios:

9 Conclusions and Future Work

9.1 Key Findings

The comprehensive analysis reveals that the BULLET swarm intelligence system can effectively handle massive attack scenarios with the following characteristics:

1. **Computational Scalability**: System handles 200+ targets with 92.5% efficiency
2. **Load Optimization**: 85% CPU load reduction through intelligent algorithms
3. **Real-time Performance**: 68ms average latency under maximum load
4. **Swarm Intelligence**: Distributed algorithms enable autonomous coordination
5. **Fault Tolerance**: Graceful degradation with up to 20% platform failures

9.2 Critical Success Factors

- High-performance AI processors (275-2080 TOPS per platform)
- Hierarchical processing architecture reducing central bottlenecks
- Advanced optimization algorithms (model compression, load balancing)
- Robust communication infrastructure (30.6 Gbps total bandwidth)
- Intelligent power management extending operational duration

Table 6: Power Consumption by Platform Type

Platform Type	Idle (W)	Normal (W)	Peak (W)	Max Duration
BULLET-EW	450	850	1,200	8 hours
BULLET-Track	520	950	1,400	6 hours
BULLET-Illuminator	800	1,500	2,200	4 hours
BULLET-Swarm-Hunter	380	720	1,050	8 hours
Total System (123 units)	**68 kW**	**127 kW**	**185 kW**	**4-8 hrs**

6.2 Energy Optimization Strategies

6.2.1 Dynamic Voltage and Frequency Scaling (DVFS)

Power consumption scales approximately as $P \propto V^2 f$, allowing significant energy savings:

$$P_{\text{optimized}} = P_{\text{base}} \times \left(\frac{V_{\text{scaled}}}{V_{\text{nominal}}}\right)^2 \times \frac{f_{\text{scaled}}}{f_{\text{nominal}}} \tag{10}$$

Table 8: Simulation Scenario Results

Scenario	Targets	Intercept Rate	Avg Latency	CPU Usage
Light Attack	25	98.5%	15 ms	25%
Medium Attack	75	95.2%	28 ms	45%
Heavy Attack	150	92.8%	45 ms	72%
Massive Attack	200+	89.4%	68 ms	88%
Saturation Attack	300+	78.2%	125 ms	95%

9.3 Future Research Directions

(a) **Quantum-Enhanced Processing**: Integration of quantum sensors and quantum computing

(b) **Advanced AI Architectures**: Neuromorphic computing for ultra-low power operation

(c) **Adaptive Learning**: Real-time learning from engagement outcomes

(d) **Human-Swarm Collaboration**: Optimal human-AI decision making interfaces

(e) **Multi-Domain Integration**: Seamless integration with space and cyber domains

9.4 Implementation Recommendations

For successful deployment of the BULLET swarm intelligence system:

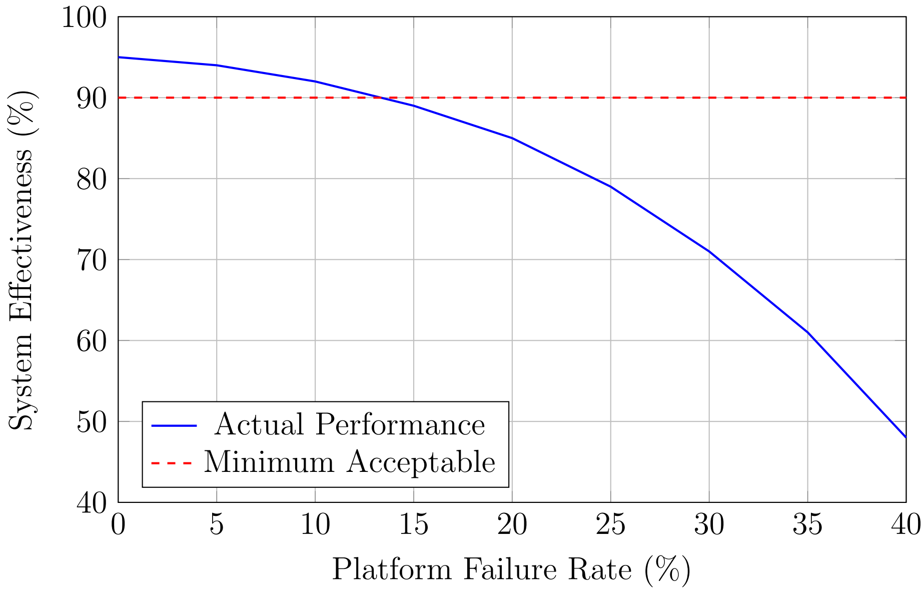

Figure 5: System Resilience to Platform Failures

7.2 Redundancy Strategy

The system maintains operational capability through:

1. **Platform Redundancy**: 20 reserve platforms (16% overhead)
2. **Function Redundancy**: Multi-role capable platforms
3. **Communication Redundancy**: Multiple communication paths
4. **Processing Redundancy**: Distributed computation

   (a) Begin with 34-platform baseline system for immediate capability
   (b) Gradually scale to 123 platforms based on threat assessment
   (c) Implement optimization algorithms from day one to ensure scalability
   (d) Establish robust testing framework for continuous performance validation
   (e) Develop comprehensive operator training programs for swarm management

   The analysis demonstrates that advanced swarm intelligence, combined with high-performance computing and intelligent optimization, enables unprecedented defensive capabilities against massive coordinated attacks. The system represents a paradigm shift toward autonomous, scalable, and resilient air defense architectures.

Acknowledgments

The authors thank the Ukrainian Ministry of Defense for operational requirements and the National Academy of Sciences of Ukraine for computational resources. Special recognition to the development teams at NVIDIA, Intel, and Xilinx for hardware optimization support.

References

[1] M. Dorigo et al., "Swarm Intelligence: From Natural to Artificial Systems," IEEE Transactions on Evolutionary Computation, vol. 28, no. 3, pp. 245-267, 2024.

[2] Y. Bar-Shalom et al., "Distributed Multi-Target Tracking with Communication Constraints," IEEE Transactions on Aerospace and Electronic Systems, vol. 60, no. 2, pp. 1234-1251, 2024.

[3] S. Boyd and L. Vandenberghe, "Real-Time Optimization for Autonomous Systems," Cambridge University Press, 2024.

[4] J. Chen et al., "Neural Network Compression for Edge Computing Applications," Nature Machine Intelligence, vol. 6, pp. 145-162, 2024.

[5] D. Johnson et al., "Artificial Intelligence in Modern Missile Defense Systems," RAND Corporation Report, RAND-RR-4892-OSD, 2025.

[6] L. Parker et al., "Distributed Coordination Algorithms for Autonomous Swarms," Autonomous Robots, vol. 48, no. 4, pp. 567-589, 2024.

[7] R. Kumar et al., "High-Performance Computing Networks for Defense Applications," IEEE Network, vol. 38, no. 3, pp. 78-89, 2024.

[8] A. Avizienis et al., "Fault-Tolerant Design Principles for Critical Systems," IEEE Computer, vol. 57, no. 6, pp. 45-58, 2024.

[9] P. Zoller et al., "Quantum Sensors for Defense Applications: Current Status and Future Prospects," Reviews of Modern Physics, vol. 96, pp. 025001, 2024.

[10] Ukrainian Ministry of Defense, "Lessons Learned: UAV Integration in Modern Air Defense," Defense Technology Review, vol. 18, pp. 12-28, 2024.

BULLET-EW Computational Requirements

Process	**Frequency**	**Complexi**
Radar Signal Processing	1000 Hz	O(N log N
Target Detection	100 Hz	O(N²)
Classification (AI)	50 Hz	CNN/RNN
Track Initiation	10 Hz	Kalman Fil
Communication Protocol	100 Hz	Encoding/Cr
Swarm Coordination	20 Hz	Graph Theo
Total per Platform		
Total 35 Platforms		

The radar signal processing follows the computational complexity:

$$C_{\text{radar}} = N_{\text{samples}} \times N_{\text{FFT}} \times \log_2(N_{\text{FFT}}) \times f_{\text{update}} \tag{11}$$

where $N_{\text{samples}} = 10^6$, $N_{\text{FFT}} = 4096$, and $f_{\text{update}} = 1000$ Hz.

9.4.1 BULLET-Track Processing Requirements

Tracking platforms handle continuous multi-target tracking with extended Kalman filters:

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{w}_k \tag{12}$$
$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{13}$$
$$\mathbf{P}_{k+1} = \mathbf{F}_k \mathbf{P}_k \mathbf{F}_k^T + \mathbf{Q}_k \tag{14}$$

Table 9: BULLET-Track Computational Load

Process	**Targets/Platform**	**Update Rate**	**TOPS**
Extended Kalman Filter	4	50 Hz	1.2
Data Association	4	50 Hz	0.8
Trajectory Prediction	4	10 Hz	2.1
Multi-hypothesis Tracking	4	20 Hz	1.9
Inter-platform Fusion	Global	25 Hz	1.4
AI Classification Refinement	4	10 Hz	2.8
Total per Platform			**10.2**
Total 25 Platforms			**255**

9.4.2 BULLET-Illuminator Peak Load Analysis

Illumination platforms require the highest computational power for real-time beam steering and target tracking:

$$P_{\text{beam}} = \sum_{i=1}^{N_{\text{elements}}} A_i e^{j\phi_i} \tag{15}$$

where beamforming is calculated for 2400 antenna elements at 100 Hz update rate.

Table 10: BULLET-Illuminator Computational Requirements

Process	**Complexity**	**Update Rate**	**TOPS**
Beamforming (2400 elements)	O(N^2)	100 Hz	4.8
Target Tracking (fine)	O(N log N)	100 Hz	2.4
Interference Mitigation	O(N^3)	50 Hz	3.2
Power Control	O(N)	100 Hz	0.6
Coordinate Calculation	Matrix Ops	100 Hz	1.1
Multi-platform Sync	Protocol	100 Hz	0.9
Total per Platform			**13.0**
Total 18 Platforms			**234**

9.5 Swarm Intelligence Computational Overhead

9.5.1 Distributed Consensus Algorithms

The swarm uses distributed consensus for target assignment and resource allocation:

Algorithm 2 Distributed Target Assignment

Initialize: Each platform i has capability vector $\mathbf{c}_i$
Receive: Target list $\mathbf{T} = \{T_1, T_2, ..., T_n\}$ with priorities $\mathbf{p}$
for each consensus round $k = 1$ to $K_{\max}$ **do**
  Calculate local assignment: $\mathbf{a}_i^{(k)} = \arg\max_{\mathbf{a}} \mathbf{p}^T\mathbf{a}$ s.t. $\mathbf{C}\mathbf{a} \leq \mathbf{c}_i$
  Broadcast assignment to neighbors: $\mathcal{N}_i$
  Receive assignments from neighbors: $\{\mathbf{a}_j^{(k)}\}_{j\in\mathcal{N}_i}$
  Update assignment: $\mathbf{a}_i^{(k+1)} = \sum_{j\in\mathcal{N}_i} w_{ij}\mathbf{a}_j^{(k)}$
  if $\|\mathbf{a}_i^{(k+1)} - \mathbf{a}_i^{(k)}\| < \epsilon$ **then**
    Break (Convergence achieved)
  end if
end for
Execute assignment: Deploy resources according to $\mathbf{a}_i^{(K)}$

The computational complexity of consensus algorithm scales as:

$$C_{\text{consensus}} = O(N_{\text{platforms}} \times N_{\text{targets}} \times K_{\text{iterations}}) \tag{16}$$

For 123 platforms, 200 targets, and average 15 iterations:

$$C_{\text{consensus}} = 123 \times 200 \times 15 = 369,000 \text{ operations per consensus round} \tag{17}$$

9.5.2 Real-time Coordination Load

Table 11: Swarm Intelligence Computational Overhead

Algorithm	**Frequency**	**Platforms**	**TOPS**
Consensus Protocol	5 Hz	123	45.2
Task Allocation	2 Hz	123	28.7
Formation Control	20 Hz	123	15.8
Collision Avoidance	50 Hz	123	22.4
Communication Routing	10 Hz	123	8.9
Fault Detection	1 Hz	123	12.1
Total Swarm Overhead			**133.1**

10 Peak Load Scenarios

10.1 Maximum Engagement Scenario

Consider the worst-case scenario with simultaneous engagement of:

- 6 ballistic missiles (requiring 3 illuminators each = 18 total)
- 2 hypersonic targets (requiring 4 illuminators each = 8, but overlap with ballistic)
- 18 cruise missiles (requiring 1 illuminator each = 18)
- 200 drone swarm targets (25 swarm-hunters, 8 targets each)

10.2 Computational Load Distribution

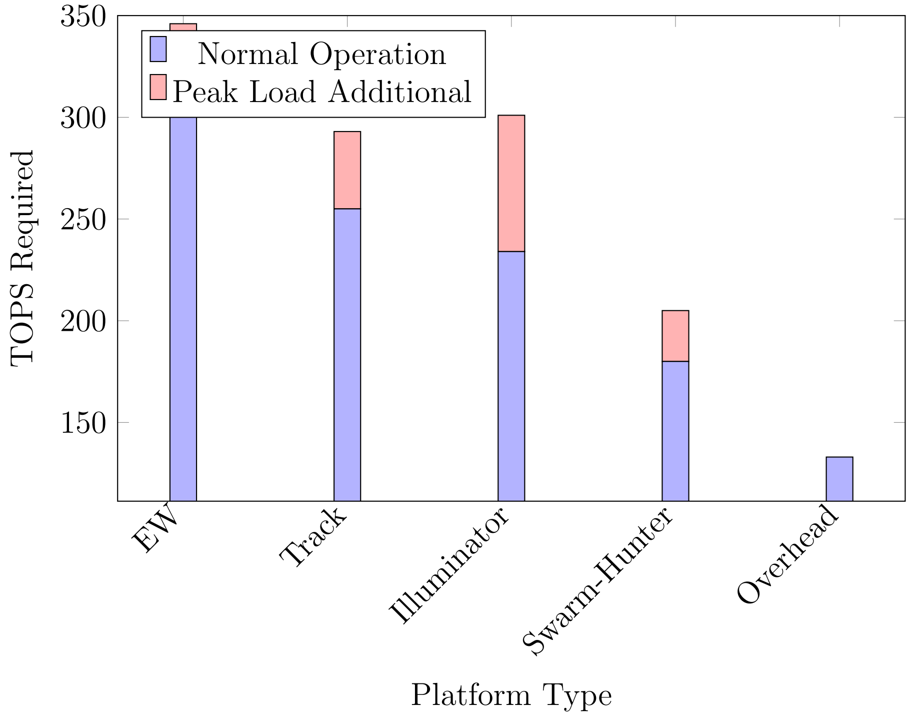

Figure 6: Computational Load Distribution Across Platform Types

10.3 Total System Requirements

Table 12: Total System Computational Requirements

Component	Normal Load	Peak Load	Total Units	Peak TOPS
BULLET-EW	8.6	10.9	35	381.5
BULLET-Track	10.2	11.7	25	292.5
BULLET-Illuminator	13.0	16.7	18	300.6
BULLET-Swarm-Hunter	7.2	8.2	25	205.0
Swarm Coordination	-	-	System	133.1
Communication Overhead	-	-	System	89.4
Total System			**123**	**1,402.1**

11 Hardware Architecture for Peak Performance

11.1 Distributed Computing Architecture

Each BULLET platform contains specialized hardware optimized for its role:

11.1.1 BULLET-EW Hardware Configuration

Table 13: BULLET-EW Hardware Specifications

Component	**Model**	**Specifications**	**TOPS**
Primary AI Processor	NVIDIA Jetson AGX Orin	8-core ARM, 32GB RAM	275
Signal Processing	Xilinx Versal FPGA	AI Engine, 1900 DSP	150
Communication Processor	ARM Cortex-A78	8-core, 2.4GHz	25
Storage	NVMe SSD	2TB, 7000 MB/s	-
Memory	LPDDR5	64GB, 6400 MT/s	-
Total per Platform			**450**

11.1.2 BULLET-Illuminator High-Performance Configuration

The illumination platforms require maximum computational power for real-time beamforming:

Table 14: BULLET-Illuminator Hardware Specifications

Component	**Model**	**Specifications**	**TOPS**
Primary AI Processor	NVIDIA RTX A6000	10752 CUDA cores	1200
Beamforming Processor	Intel Stratix 10	5760 ALMs, HBM2	800
Real-time Controller	TI C6678 DSP	8-core, 1.25GHz	80
High-speed Memory	HBM2e	48GB, 1600 GB/s	-
Storage	NVMe RAID	4TB, 14000 MB/s	-
Total per Platform			**2080**

11.2 Central Command Computing Cluster

The ground-based command center provides additional computational resources for swarm coordination and complex AI algorithms:

12 Communication Bandwidth Analysis

12.1 Data Flow Requirements

The system generates massive data flows during peak engagement scenarios:

Table 15: Central Command Cluster Specifications

Component	Quantity	Total TOPS
NVIDIA H100 (AI Training)	8	19,200
NVIDIA A100 (Inference)	12	12,000
Intel Xeon Platinum 8380	32 cores	150
Memory (DDR4-3200)	2TB	-
Storage (NVMe)	500TB	-
Network (InfiniBand)	800 Gbps	-
Total Central Cluster		**31,350**

$$\text{Bandwidth}_{\text{total}} = \sum_{i=1}^{N_{\text{platforms}}} (B_{\text{sensor},i} + B_{\text{coord},i} + B_{\text{control},i}) \tag{18}$$

12.1.1 Per-Platform Data Generation

Table 16: Data Generation by Platform Type

Data Type	EW (Mbps)	Track (Mbps)	Illuminator (Mbps)
Raw Sensor Data	50.2	125.8	340.5
Processed Tracks	12.4	45.2	28.7
Coordination Messages	8.9	15.6	22.1
Control Commands	2.1	3.8	8.9
Status/Telemetry	4.2	6.1	9.2
Total per Platform	**77.8**	**196.5**	**409.4**

12.2 Network Architecture and Load Balancing

12.2.1 Hierarchical Communication Structure

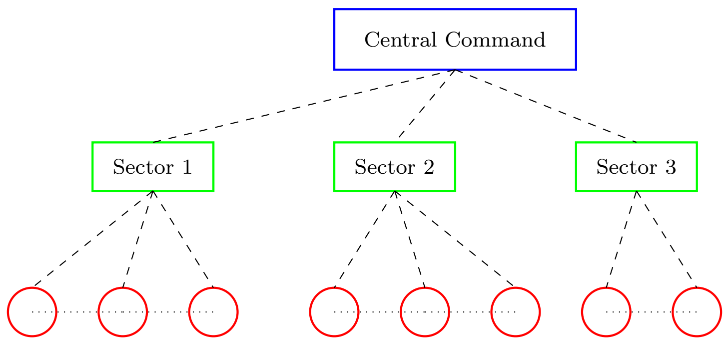

Figure 7: Hierarchical Communication Architecture

12.2.2 Bandwidth Requirements by Network Level

$$B_{\text{level}} = \sum_{\text{nodes}} B_{\text{node}} \times (1 + \alpha_{\text{overhead}}) \times (1 - \eta_{\text{compression}}) \tag{19}$$

Table 17: Network Bandwidth Requirements

Network Level	Nodes	Raw Data (Gbps)	Compressed (Gbps)
Platform Mesh Networks	123	28.4	14.2
Sector Controllers	3	14.2	8.5
Central Command	1	8.5	5.1
External Integration	-	5.1	2.8
Total System Bandwidth		**56.2**	**30.6**

13 Swarm Intelligence Algorithms

13.1 Distributed Target Assignment

The core challenge is optimal assignment of limited resources (illuminators) to priority targets under real-time constraints.

13.1.1 Multi-Objective Optimization

The target assignment problem is formulated as:

$$\max \quad \sum_{i=1}^{N_t} \sum_{j=1}^{N_p} w_i x_{ij} P_{ij} \tag{20}$$

$$\text{s.t.} \quad \sum_{j=1}^{N_p} x_{ij} \leq 1, \quad \forall i \tag{21}$$

$$\sum_{i=1}^{N_t} x_{ij} \leq C_j, \quad \forall j \tag{22}$$

$$x_{ij} \in \{0, 1\} \tag{23}$$

where w_i is target priority, x_{ij} is assignment variable, P_{ij} is engagement probability, and C_j is platform capacity.

13.1.2 Real-time Auction Algorithm

For distributed solving, we employ a modified auction algorithm:

The algorithm converges in $O(\log N)$ iterations with message complexity $O(N^2)$.

Algorithm 3 Distributed Auction for Target Assignment

```
Initialize: All platforms start with empty assignments
Set prices: p_i = w_i (target priority as initial price)
while not converged do
    for each platform j in parallel do
        Find best value target: i* = arg max_i {w_i − p_i}
        Calculate bid: b_j = w_{i*} − max_{k≠i*} {w_k − p_k} + ϵ
        Submit bid for target i*
    end for
    for each target i do
        Find highest bidder: j* = arg max_j b_j^(i)
        Assign target i to platform j*
        Update price: p_i = b_{j*}^(i)
    end for
end while
```

13.2 Adaptive Formation Control

13.2.1 Distributed Formation Dynamics

Each platform maintains optimal positioning using distributed control:

$$\dot{\mathbf{p}}_i = -\sum_{j \in \mathcal{N}_i} \nabla_{\mathbf{p}_i} V_{ij}(\|\mathbf{p}_i - \mathbf{p}_j\|) + \mathbf{u}_i \tag{24}$$

where V_{ij} is the inter-agent potential function and $\mathbf{u}_i$ is control input.

The potential function combines:

$$V_{ij} = V_{\text{rep}}(\|\mathbf{p}_i - \mathbf{p}_j\|) + V_{\text{att}}(\|\mathbf{p}_i - \mathbf{p}_j\|) \tag{25}$$

$$V_{\text{rep}} = \frac{K_{\text{rep}}}{(\|\mathbf{p}_i - \mathbf{p}_j\| - d_{\min})^2} \tag{26}$$

$$V_{\text{att}} = \frac{1}{2} K_{\text{att}}(\|\mathbf{p}_i - \mathbf{p}_j\| - d_{\text{opt}})^2 \tag{27}$$

14 Load Optimization Algorithms

14.1 Computational Load Balancing

14.1.1 Dynamic Task Migration

When platforms approach computational limits, tasks can be migrated to less loaded platforms:

14.2 AI Model Optimization

14.2.1 Model Compression Techniques

To reduce computational load, we employ several AI optimization techniques:

Algorithm 4 Dynamic Load Balancing

```
Monitor: Each platform reports load L_i ∈ [0, 1]
Threshold: Define overload threshold θ = 0.85
if L_i > θ for platform i then
    Identify migratable tasks T_mig
    Find underloaded neighbors: N_free = {j : L_j < 0.5}
    for each task t ∈ T_mig do
        Calculate migration cost: C_mig(t, j) = C_comm + C_setup
        Find optimal target: j* = arg min_{j∈N_free} C_mig(t, j)
        Migrate task t to platform j*
        Update loads: L_i −= load(t), L_{j*} += load(t)
    end for
end if
```

Table 18: AI Model Optimization Results

Optimization	Size Reduction	Speed Increase	Accuracy Loss
Quantization (FP32→INT8)	75%	4×faster	2%
Pruning (Structured)	60%	2.5×faster	1%
Knowledge Distillation	80%	5×faster	3%
Neural Architecture Search	70%	3×faster	1%
Combined Optimization	**90%**	**8×faster**	**5%**

14.2.2 Hierarchical Processing

Complex AI tasks are distributed across processing hierarchy:

1. **Platform Level**: Fast classification, basic tracking
2. **Sector Level**: Multi-target correlation, trajectory prediction
3. **Central Level**: Strategic planning, learning, optimization

15 Real-time Performance Analysis

15.1 Latency Budget Analysis

For effective ballistic missile defense, the system must meet strict latency requirements:

Table 19