Nature-Inspired Techniques for Avoiding Congestion in Wireless Sensor Networks

Pavlos Antoniou
Ph.D. Defense

Supervisor: Prof. Andreas Pitsillides
“The great book, always open and which we should make an effort to read, is that of Nature”

Antoni Gaudi
Spanish architect, 1852-1926
Outline

• Introduction to Wireless Sensor Networks (WSNs)
• Problem of congestion in WSNs
• Motivation
• The Flock-based Congestion Control (Flock-CC) Approach
• The Lotka-Volterra based Congestion Control (LVCC) Approach
• Performance Evaluation and Results
• Conclusions and Future Work
Problem & Our Approaches

- **Congestion**: Phenomenon that occurs when load injected into network **is near or exceeds** the capacity of network resources → Degradation of performance
- **Congestion control and avoidance**: measures taken to avoid congestion, in order for the network to operate at acceptable performance levels (low packet loss, low delay)
- We successfully employed
  - obstacle avoidance behavior of bird flocks
  - competitive coexistence behavior of species in nature to avoid congestion in WSNs
Wireless Sensor Networks (1/2)

• Autonomous decentralized infrastructures
  – Comprise of small, cheap, cooperative nodes
  – Work without external intervention in dynamically changing conditions
  – Dynamic topology (e.g. due to failing nodes, mobility)
  – Individual node constraints:
    • Low computational capability
    • Limited buffer/memory space
    • Limited communication bandwidth
    • Constrained energy supply
Wireless Sensor Networks (2/2)

- small-scale (a few nodes) to very large scale (thousand nodes)
- distance between nodes can be few meters
- densities can be very high, e.g. 20 nodes/m²

Basic components of sensor nodes
- Sensing component • sense the environment, create data packets
- Buffer • store data packets before transmitting
- Communication module • transmit data packets
WSNs Insights

- Interactions:
  - environment-to-node → sense/control physical parameters
  - node-to-node → exchange information, forward data

- Network unpredictable behaviour:
  - Variable traffic load injection into network
    - typically WSNs operate under light load
    - large, sudden, correlated-synchronized impulses of data may suddenly arise in response to a detected event
  - Link capacity fluctuations
  - Topology modifications, e.g. due to
    - mobility
    - node failures

Overload/Congestion conditions

Symptoms:
- Wireless channel contention
- Buffer overflows
Congestion Consequences in WSNs

• Packet loss
  – buffer overflows
  – wireless channel collisions

• Retransmissions → energy waste → decrease of network lifetime → decomposition of network topology

• Throughput reduction, increased queueing delay
Motivation

• The problem of congestion in WSNs is getting increasingly complex
  – Not only due to problem definition, that mostly stays the same as in the Internet

• **Resources** to solve the problem are **limited**
  – Memory, computational capability, energy, bandwidth

• Underlying communication **channel** is **unpredictable**

• **Many-to-one** communication in hop-by-hop manner
  – Different than one-to-one in end-to-end manner

• WSNs cannot adopt centralized approaches
  – Slow reaction to changes
  – Non robust (single point of failure)

• Need for simple, decentralized approaches able to achieve robustness, self-adaptation, scalability
Two novel approaches

• Flock-based Congestion Control (Flock-CC) approach
  – motivated by the obstacle avoidance behavior of bird flocks
  – provides traffic load balancing over available (unexploited) network resources whilst avoiding congestion regions
  – targets large-scale, real-time, event-based WSNs apps

• Lotka-Volterra based Congestion Control (LVCC) approach
  – motivated the well known LV competition model of mathematical biology
  – provides smooth flow rate regulation and control
  – targets small-scale, streaming WSNs apps
The Flock-CC approach

- Draws inspiration from **Swarm Intelligence**
  - Excellent basis for computing environments that need:
    - Simplicity (at individual node level)
    - Decentralized operation
    - Robustness
    - Self-* properties: self-organization, self-adaptation, self-configuration, self-optimization, self-healing, etc.
  - Global properties **achieved collectively**, as a result of **evolutionary design**, without explicitly programming them into individual nodes or devices
Flock-CC: The Motivation

- Collective motion and obstacle avoidance behavior of bird flocks
- Employ *behavioral tendencies* (*attraction & repulsion forces*) observed among individuals within bird flocks
- Employ *orientation of migratory birds to a global attractor* (pole) under the influence of the magnetic field of Earth
- Goal: *emergent behaviour of minimizing congestion and directing information flow to the sink*
Flock-CC: The Concept

• Packets: individuals within a flock
• Goal: ‘guide’ packets to form flocks and flow towards a global attractor (sink), whilst trying to avoid obstacles (congestion regions and dead zones)
Bird flocking behavior

• Initial attempts by work of Craig Reynolds*
• According to Craig Reynolds, 3 basic rules governing interactions between neighboring particles in a swarm or flock are:
  – if too close, move apart (separation) – rule of repulsion
    • avoid collisions
  – if apart, move closer (cohesion) – rule of attraction
    • remain close to neighbors
  – attempt to match velocities (alignment)
    • move in the same direction as your neighbors
• With these three simple rules, the flock moves in an extremely realistic way, creating complex motion and interaction that would be extremely hard to create otherwise

* Graig Reynolds: artificial life and computer graphics expert, who created the Boids (simulated bird-like objects) in 1986. Boids were used in bat swarms and penguin flocks in Batman Returns (1992) and The Lion King (1994)
Flock-CC model development

• Need for defining neighborhood of each individual
• Use repulsion and attraction zones (as modeled by Couzin et al) in order to define the finite neighborhood of each individual
  – Couzin’s model, concentric zones around each individual
• Differing from Couzin’s model, in Flock-CC:
  1. apply on 2D topological (discrete) space defined by graph of nodes, whereas Couzin’s model formulated on metrical (continuous) 3D space.
  2. packets form flocks and move towards the sink, which necessitates a global field of attraction towards sink. In Couzin’s model (as well as Reynolds’) individuals form flocks and move constantly in given finite space without any attraction to a global target.
Flock-CC: Behavioral rules

• Achieve **congestion control** as a result of **4 simple rules** followed by each individual packet in flock:
  
  – **Rule 1: repel** from neighbouring packets on nodes at close distance (within zone of repulsion).
     
  • high queue loading (crowded nodes) can experience stronger repulsion

  – **Rule 2: attract** to neighbouring packets on nodes at medium distance (within zone of attraction)
     
  • low wireless channel contention can experience higher attraction

  – **Rule 3: orient** toward global **attractor** (sink)

  – **Rule 4: experience** perturbation (exploration)

• **Emergent behavior** expected to arise from simple behavioral rules followed by individual packets
Zone of Repulsion (ZoR)

- Packets at close distance (within the transmission range of current hosting node n)
- Strength of repulsion force proportional to packets residing in queues of grey-shaded nodes
- Number of packets in queues obtained through control packets broadcasted periodically (every T seconds – sampling period)
Zone of Attraction (ZoA)

- Packets at medium distance (two hops away from hosting node n)
- Strength of attraction force proportional to packets residing in queues of black-shaded nodes
  - cannot be obtained timely through control packets because black-shaded nodes outside of transmission range
  - use only locally available information
  - packet i can perceive packets ‘flying’ from nodes one hop away to nodes two hops away
Need for orientation

- Repulsion/attraction forces allow packets form flocks but move in any direction without orientation to a global attractor → Rooting loops in the network
- **Orientation and attractiveness to a global attractor** can be extracted from the orientational movement of migratory birds towards the poles
- Make sink artificial magnetic pole in a WSN

»Goal: guide packets move *sinkwards* under the influence of the artificial magnetic field
Magnetic field and Field of View (FoV)

- Artificial magnetic field should point to the sink
- Hop distance $h_n(k)$
  - # of hops between node $n$ and the sink at the kth sampling period
  - shows proximity to sink: nodes closer to sink have smaller hop distances and hence stronger ‘magnetic field’
  - sink extends forward in the direction of decreasing hop distance
- In order for birds to move sinkwards
  - turn their head toward the sink
- Mimic visual system of birds: FoV
  - FoV includes packets on nodes at equal or smaller hop distance to sink
Redefinition of Zones

• Zone of Repulsion and Zone of Attraction refined within the FoV
Recap: Bird Flocking Elements

- Rules followed by each packet:
  - repel from neighboring packets on nodes at close distance
  - attract to neighboring packets on nodes at medium distance
  - orient toward global attractor (sink)
  - experience perturbation (exploration)
  - Mimic visual system of birds (Limited visual perception: Field of View (FoV))
Flock-CC Protocol (1/4)

- At each node, each packet chooses its new hosting node (from $M$ nodes one hop away in FoV)
- Packet chooses its new hosting node on the basis of a **desirability function** for each node $m \in M$

$$D_{nm}(k) = s_{nm}(k) - q_{nm}(k)$$

- synthesizes the attraction and repulsion forces
- measures tendency of a packet on node $n$ to move towards each neighboring node $m$

- evaluated once per time period $k$ (every $T$ sec.)
  - $T$: sampling period
Flock-CC Protocol (2/4)

- **Attraction** to packets moving to nodes 2 hops away

\[
s_{nm}^{\text{norm}}(k) = \begin{cases} 
\frac{s_{nm}(k)}{s'_{nm}(k)} & \text{if } s'_{nm}(k) > 0 \\
\xi & \text{otherwise},
\end{cases}
\]

- \( s_{nm}^{\text{norm}}(k) \): measure of wireless channel loading around node \( m \)
  - \( \rightarrow 1 \) channel not congested, \( \rightarrow 0 \) channel congested

- \( s_{nm}(k) \): number of packets successfully transmitted from node \( m \) to nodes two hops away from node \( n \) (# of packets in ZoA) within period \( k \)

- \( s'_{nm}(k) \): number of total transmission attempts at node \( m \) within period \( k \)

- \( \xi \): spreading variable \([0,1]\) – allows attraction to idle nodes (at the borders of the flock)
  - low \( \xi \) values weak attraction to idle nodes; coherent flock motion (low spreading)
Flock-CC Protocol (3/4)

• Repulsion from packets on nodes 1 hop away

\[ q_{nm}^{\text{norm}}(k) = \frac{q_{nm}(k)}{Q_m} \]

- \( q_{nm}^{\text{norm}}(k) \): queue occupancy at node \( m \)
  - \( \rightarrow 1 \): high queue occupancy, \( \rightarrow 0 \) queue nearly empty
- \( q_m(k) \): number of packets in the queue of node \( m \)
  (# of packets in ZoR) within period \( k \)
- \( Q_m \): queue capacity of node \( m \)

\( \xi \)-spreading variable and T-sampling period are only two tuneable parameters – their behaviour is well understood
Flock-CC Protocol (4/4)

- **Orientation**: choosing the next hop hosting node:
  - choose set of nodes with shorter hop distance than the current hosting node having available buffer space
  - if this set is empty, choose set of nodes with equal hop distance having available buffer space
  - if this set is empty, choose set of nodes with longer hop distance

- Involve **perturbation** when selecting new hosting node from the chosen set (introduce exploration)
  - rank-based selection: rank nodes from chosen set by increasing desirability ($J = \text{no. of nodes in chosen set}$)
    - weakest node has fitness $f_i' = 1$
    - fittest node has fitness $f_i' = J$
  - probability to choose a node $\rightarrow p_i = \frac{f_i'}{\sum_{j=1}^{J} f_j'}$
Flock-CC implementation

• Every time a packet is about to be sent, the decision making process is invoked by the current hosting node to determine the new hosting node.

• The decision process employs three stages:
  a) selection of direction (forward, sideways, backwards) using the notion of the FoV and the magnetic fields,
  b) sorting of all nodes in the selected direction in descending order by their desirability function (calculated once per T), and
  c) probabilistic, biased (proportional to desirabilities) selection of the new hosting node.
Performance evaluations

- Performance evaluations focus on four directions:
  - Parameter selection ($\xi, T$)
  - Demonstration of:
    - emerging behavior
    - self-adaptation to changing network and traffic conditions
    - robustness against failing nodes
    - scalability as network size changes
  - Comparative evaluations
    - between previous and current Flock-CC models
    - against related (nature-inspired and conventional) congestion control approaches
Evaluation setup

- Evaluation topologies
  - Lattice (300 homog. nodes)
  - Random (300 homog. nodes)

- Evaluation parameters
  - Sampling period $T$: 0.5, 1, 1.5, 2 sec.
  - Node queue size: 50 packets
  - IEEE 802.11: 2Mbps, 250Kbps
  - Traffic load: light (25 pkts/s), medium (35 pkts/s), heavy (45 pkts/s)

- Evaluation measures:
  - Packet Delivery Ratio (PDR)
  - End-to-End Delay (EED)
  - Energy tax
  - Throughput

Diagram:
- Active nodes, Scenario 1: 20 nodes
  - Activated at $t=50$ s
  - Deactivated at $t=70$ s

- Active nodes, Scenario 2: 15 nodes
  - Activated at $t=10$ s
  - Reactivated at $t=70$ s

- Sink node

- Dead nodes, Scenario 3:
  - Failed at $t=40$ s
  - Activated at $t=50$ s
Results – Scenario 1 (35pkt/sec)

• Low $\xi \{0, 0.25\} \rightarrow$ low spreading $\rightarrow$ available paths left unexploited $\rightarrow$ high overload in popular paths $\rightarrow$ high number of collisions and buffer overflows $\rightarrow$ low PDR & high EED

• High $\xi \{1\} \rightarrow$ high spreading $\rightarrow$ high number of collisions

• Good compromise values: $\xi = \{0.5, 0.75\}$

• $T=1$ sec.: compromise between keep network updated without high control packet overhead
Results – Scenario 1 (35pkts/sec)

- High number of retransmissions for:
  - $\xi=0$ and $\xi=0.25$
    - buffer overflows & collisions
  - $\xi=1$
    - collisions

Good compromise values: $\xi = \{0.5, 0.75\}$
Results – Scenario 3 (failing nodes, 35 pkts/s)

• Buffer overflows followed same behavior with increase of $T$ as in scenario 1

• Unlike scen. 1, collisions increased with increase of $T$
  – High $T$: Infrequent control packet exchange and desirability evaluation $\rightarrow$ packet flock incapable of adapting to rapidly changing network conditions
  – Low $T$: fast adaptation of flock movement to network conditions
  – Good compromise value in failing node conditions: $T = 0.5$ sec.
Energy tax

- Lowest tax paid in scen 2
  - packets traveled shorter paths to the sinks
- Highest tax paid in scen 3
  - failing nodes => packets traveled longer paths to the sink whilst maneuvering around the “dead” zone
- Frequent updates (T=0.5s)
  - Highest tax for scens 1 & 2
    - Higher number of control pkts sent
  - Lowest tax for scenario 3
    - entities need to be updated about network state otherwise pkt drops, retransmissions
- Changes in energy tax fairly insensitive to $\xi$
Throughput

- Flock-CC achieves fairness between active nodes
  - active nodes achieve similar throughput
- Fluctuations in throughput as new active nodes added & network capacity reached
- Steep decline in throughput during extreme failing node phenomena

Fast adaptation to network conditions (10 sec after failures)
Results (Low data rates & random topologies)

• Low data rate WSNs, 250Kbps
  – Parameter setting similar to high data rate WSNs
  – Majority of packet loss attributed to collisions
    • Low rates $\rightarrow$ buffers rarely fill up

• Random topologies
  – Sparse and dense topologies of 300 nodes
  – High density topos $\rightarrow$ increased collisions $\rightarrow$ lower PDR
    • highest PDR + lowest EED $\rightarrow$ $\xi = 0.5$ and 0.75, $T = 2$ sec
  – Low density topos: limited network resources $\rightarrow$ limited paths to sink $\rightarrow$ increased buffer overflows (x10 more than dense topos)
    • up to 20% lower PDR compared to dense

• Overall recommended values:
  $\xi = 0.75$, $T = 0.5s$ or 1s
Need for adapting T value

• Value of T can be adapted to dynamically changing network conditions (e.g. failures)
• Initial (simple approach)
  – Initially set T=1s to avoid high control packet overhead
  – Change to T=0.5s after failures only nodes 1 hop away from failures for a small amount of time
    • If change to 0.5 for 2secs, results close to scen. having 0.5s
• Design choices need further study
  – Which nodes (number of hops away from failure point) will participate?
  – For how long?
  – Other rule-based/equation-based approach for optimally tuning T
Emergent behavior: visualizations

Packets form flocks and ‘fly’ over the network
A number of paths to the sink are exploited

Packets generated at the bottom create two subflocks that bypass the congested area

After deactivation of front nodes, subflocks re-join

Scenario 2
Evidence of emergent behavior

- Full Flock-CC, highest PDR

- Exclusion of randomization → reduced path exploration → deterioration of PDR (scen1: 9-17%, scen3: 5-11%)

- Exclusion of local interactions → lack of social activity → lack of knowledge on neighboring buffer & channel conditions → high number of overflows & collisions → deterioration of PDR (all scenarios)

- Exclusion of both features → further PDR deterioration
Robustness against failures

- Flock-CC approach achieves robustness against failures
- Packet flock exhibits the obstacle avoidance behavior of the bird flocks

Packets maneuver around the zone of dead nodes

When hole in the middle, packets re-align to include middle path to sink
Scalability

- Lattice topologies of 200, 300 and 400 nodes in same area
- Higher PDR in large scale nets
  - number of nodes scales up $\rightarrow$ available resources increase $\rightarrow$ flock spreads in network $\rightarrow$ packet losses reduced
  - small scale nets $\rightarrow$ packets “forced” to move in coherent formations
- Lower EED in large scale nets
  - large scale nets $\rightarrow$ multiple paths to sink $\rightarrow$ lower buffer occupancy $\rightarrow$ lower time to reach sink
- Graceful degradation
Comparative evaluations

- Flock-CC outperformed No Congestion Control (NCC) and Congestion-aware Routing (CAwR) protocols in all scens
- NCC sends over shortest paths
- CAwR allows multipath routing over shortest paths choosing node with lowest queue
- Scen1: 15%, 23%, 19% higher PDR than NCC for 25, 35, 45 pkts/sec
- Scen1: 2-8% higher PDR than CAwR
- Smaller differences in scens 2, 3
- Flock-CC allows for controlled packet spreading, exploits available resources through multiple paths to sink
- NCC, CAwR significantly higher number of overflows
Comparative evaluations

• **Qualitatively compared against AntHocNet and AntSensNet**
  – quite complicated protocols involving large number of parameters and equations (2x & 4x more respectively)
  – parameters have to be tuned for variety of network and traffic conditions; sensitive to environment
  – control packets much larger and a lot more (forward+backward ants) + need lots of memory space
  – AntSensNet requires modifications in the queueing policies of the underlying MAC protocol

• **Flock-CC approach**
  – quite simple involving only 2 parameters and 1 equation (desirability function),
  – much smaller and lot less control packets. No modification of the underlying protocols needed

[Comparison table](#)
Conclusions

• Control motion of packets through WSNs by mimicking synchronized group behavior of bird flocks and their ability to avoid obstacles (congested & “dead” nodes)

• Design embodying simple behavioural rules

• Results showed that congestion is alleviated by balancing the offered load through alternative (unexploited) paths to the sink

• Robust against failures, self-adaptable to variable network conditions

• Flock-CC outperformed related approaches in all traffic loads
Future Work

• Investigate alternative methods of evaluating attraction/repulsion forces and desirability functions
  – e.g. take direct account of energy in desirability

• Make design parameters adaptive to network changes
  – study when (immediate/delayed actions?) and how (rule-based, equation-based) to tune parameter values

• Investigate Flock-CC applicability in the presence of multiple sinks and/or mobile sinks
  – what happens if multiple magnetic poles?
    • devise criteria for differentiating the influence of each pole
  – devise moving strategy for mobile sinks
Lotka-Volterra Congestion Control (LVCC)
Lotka-Volterra competition model

• Lotka (1925) and Volterra (1926) independently developed a general model of competition between species

• Lotka-Volterra competition model
  – simple deterministic model of mathematical biology
  – describes how species population change over time as a result of species competition for some limiting resource (e.g. food, space)
  – detailed description
Ecosystems vs. WSNs

• **Ecosystem**
  - species live in nature
  - species interact with each other & non-living parts of their surroundings
  - compete for resources (e.g., food, water)
  - Result: co-existence of species

• **Sensor Network**
  - nodes initiate traffic flows
  - flows interact each other
  - flows compete for available resources located at each node (e.g., buffer, bandwidth)
  - Goal: co-existence of flows

![Diagram showing interactions between species and traffic flows]
LVCC: The concept

- Source nodes (SNs) compete for available buffer space at the parent (relay) node.
- SNs self-regulate and adapt the rate of their traffic flows so as to co-exist.
- SNs send packets to their parent node only when it has the available buffer space to hold the packets.
LVCC: Cong. detection & avoidance

• Congestion detection
  – Parent (relay) node measures its queue length
  – Broadcasts to all potential children (source nodes)

• Congestion avoidance
  – Rate adaptation
  – Every source node regulates and adapts its traffic flow rate on the basis of the LV competition model
    • queue length of parent node is taken into account
  – Goal: Avoid buffer overflow at parent (relay) node
Lotka-Volterra competition model

• Generalized Lotka-Volterra model for \( n \) species

\[
dx_i/dt = x_i \left[ r_i - \frac{\beta_i r_i}{K_i} x_i - \left( \sum_{j=1 \atop j \neq i}^{n} \alpha_{ij} x_j \right) x_i \right], \quad x_i(0) > 0, \ i = 1, \ldots, n
\]

\( x_i(t) \): biomass (population size) of species \( i \) at time \( t \) → number of bytes sent by each children node \( i \)

\( r_i \): growth rate of species \( i \)

\( \beta_i \): intra-specific competition coefficient (competitive effects among individuals of species \( i \))

\( a_{ij} \): is the inter-specific competition coefficient (competitive effects of species \( j \) on growth of species \( i \))

\( K_i \): is the carrying capacity of species (maximum number of individuals that can be sustained by the biotope in the absence of all other species competing for the same resource) → resource capacity

Species have same characteristics

\( r_i = r > 0, \ \forall i \in [1, n] \)

\( K_i = K > 0, \ \forall i \in [1, n] \)

\( a_{ij} = a > 0, \ \forall i, j \in [1, n], i \neq j \)

\( \beta_i = \beta > 0, \ \forall i \in [1, n] \)
Equilibria and stability analysis

• Equilibria of the generalised Lotka-Volterra model can be evaluated by:

\[ \frac{dx_i}{dt} = rx_i \left\{ 1 - \frac{\beta x_i}{K} - \frac{a}{K} \sum_{j=1, j \neq i}^{n} x_j \right\} = 0, \forall i \in [1, n] \]

• Coexistence non-negative equilibrium solution \( x_i^* = x^* \)

\[ x_i^* = \frac{K}{\alpha(n-1) + \beta}, i = 1, ..., n \]

• Stability analysis (using eigenvalues) of coexistence solution
  – all flows (species) co-exist (survive) when \( \beta > \alpha, \alpha \geq 1 \)
    • inter-specific competition is weaker than intra-specific competition
LVCC: Conditions to be satisfied

- Equilibrium stability conditions: \( \alpha \geq 1, \beta > \alpha \)

- Buffer overflow avoidance:
  - when system of \( n \) active nodes converges to the coexistence solution, \( x_i^* = \frac{K}{\alpha(n-1) + \beta}, i = 1, \ldots, n \)
  - each node \( i \) should send less than or equal to \( K/n \) bytes at equilibrium
  - denominator of \( x_i^* > n \)
    \[ \alpha(n-1) + \beta \geq n \quad \text{or} \quad \beta - \alpha \geq n \times (1 - \alpha) \]

- To ensure both conditions: \( \beta > \alpha, \quad \alpha > 1 \)
LVCC: Rate evaluation (1/2)

• Each node $i$ evaluates its flow rate using the solution of the LV differential equation
  – rate evaluation every period $T$

• Solution of LV differential equation by node $i$ requires:
  \[
  \frac{dx_i}{dt} = rx_i \left\{1 - \frac{\beta x_i}{K} - \frac{a}{K} \sum_{j=1}^{n} x_j \right\} = 0
  \]
  – knowledge of variables $r, K, \alpha, \beta$
  – number of bytes sent by node $i$ within previous period $T, x_i$
  – number of bytes sent by all other competing nodes $j$, $\sum_{j=1, j \neq i}^{n} x_j$
  within previous period $T$:
    • difficult to be obtained in a distributed decentralized network
    • set $C_i = \sum_{j=1, j \neq i}^{n} x_j = \text{parent node’s queue length} - x_i$
LVCC: Rate evaluation (2/2)

• Solution of LV differential equation:

\[ x_i(t) = \frac{w(0)x_i(0)}{\beta x_i(0) + [w(0) - \beta x_i(0)]e^{-\frac{w(0)r}{K}t}}, \quad w(0) = K - \alpha C_i(0) \]

– \( x_i(t) \): number of bytes send by node \( i \) at time \( t \)

• Discrete-time equation of \( x_i \) at the \( k+1 \)th period:

\[ x_i((k+1)T) = \frac{w(kT)x_i(kT)}{\beta x_i(kT) + [w(kT) - \beta x_i(kT)]e^{-\frac{w(kT)r}{K}T}}, \quad w(kT) = K - \alpha C_i(kT) \]

– used by source nodes (SNs)
– slightly modified equation used for relay nodes (RNs)
Performance evaluations

• Performance evaluations focus on three directions:
  – Parameter selection (α, β, r)
  – Demonstration of:
    • self-adaptation to changing network and traffic conditions
    • scalability as the network size changes
    • fairness among active nodes
  – Comparative evaluations
    • against related congestion control approaches
Evaluation setup

- Cluster-based evaluation topology (all links are wireless)

- Evaluation parameters
  - Buffer capacity (K): 35KB
  - Time period between successive sending rate evaluations: \( T = 1 \text{sec} \)
  - \( \alpha, \beta, r > 0, \beta > \alpha \)

- Evaluation measures
  - Bandwidth (number of pkts sent)
  - Packet delivery ratio
  - End-to-end delay
Simulations

• Control system type simulations (Matlab) for theoretical model analysis
  – Evaluate validity of analytical results

• Realistic network simulations (NS2)
  – Two-ray ground radio propagation model
  – CSMA-based IEEE 802.11 MAC, 1 Mbps

<table>
<thead>
<tr>
<th>Scenario</th>
<th>No. of active nodes</th>
<th>Active nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>5, 6, 10</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>5, 6, 10, 13, 14</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>5, 6, 10, 12, 13, 14, 21</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>5, 6, 10, 11, 12, 13, 14, 18, 21, 24</td>
</tr>
</tbody>
</table>
Matlab results

- Buffer overflows never occur
  - sending rates < buffer capacity

- Scalability
  - as # of active nodes scales up, their sending rates decrease
  - graceful performance degradation

- Adaptation
  - each active node self-adapts its sending rate
  - responsiveness to changes in the number of active nodes

- Fairness
  - Clusterheads’ buffer capacity is fairly shared among active cluster nodes

\[ x_i^* = \frac{K}{\alpha(n-1) + \beta} \]
Matlab results (cnt’d)

α=3, r=1, β=4

• no analytical upper bound for β
• β cannot grow unboundedly
  – Increase of β decreases coexistence solution => decrease of transmission rate

• α < β for system stability
• Increase of α :
  – decreases coexistence solution
  – smooth traffic sending rates are not preserved
    – close to stability limits

• Results showed that r can not grow unboundedly
• Smooth traffic sending rates are not preserved with the increase of r
• r ≤ 2 for system stability
NS2: Parameters setting, 3 nodes

- Decrease in PDR perceived for low values of $\alpha$ and $\beta$
- Mainly attributed to the increase in transmission rates at equilibrium:
  - increased traffic load provoked channel contention, packet loss.
- Sharp decrease in PDR was observed when the stability condition was threatened, e.g., $3.5<\alpha<4$ and $\beta=4$

\[ x_i^* = \frac{K}{\alpha(n-1) + \beta} \]
Validation of stability & buffer overflow avoidance conditions

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\beta &gt; \alpha$</th>
<th>$\alpha [n - 1] + \beta \geq n$</th>
<th>$x^*$ (Kbytes/sec) when all active</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.8</td>
<td>X</td>
<td>2.8 $\geq$ 3 X</td>
<td>-</td>
</tr>
<tr>
<td>0.25</td>
<td>0.5</td>
<td>$\checkmark$</td>
<td>1 $\geq$ 3 $\checkmark$</td>
<td>35</td>
</tr>
<tr>
<td>3.0</td>
<td>5.0</td>
<td>$\checkmark$</td>
<td>11 $\geq$ 3 $\checkmark$</td>
<td>3.18</td>
</tr>
</tbody>
</table>

(a) Calculated transmission rates
(b) Stream throughput measured at the sink
NS2: Parameters setting, 5 nodes

- In Matlab, stability was achieved for $r \leq 2$
- Realistic experiments showed that for $r < 1$ calculated transmission rate does not converge
- Extensive simulations showed that system stability is achieved for $1 \leq r \leq 2$
NS2: Parameters setting, 10 nodes

- Highest PDR (0.9) achieved for $6 \leq \beta \leq 7$ and $1.8 \leq \alpha \leq 2.1$
- Lowest EED (10μs) achieved for $6 \leq \beta \leq 7$ and $1.8 \leq \alpha \leq 2.1$
Parameter Setting

• Values of parameters \( \alpha, \beta \) and \( r \) should be chosen to ensure convergence, stability and buffer overflow avoidance

• \( r=1 \) : preserves convergence to equilibria and smooth flow rate regulation

• \( \alpha \) and \( \beta \) values depend on number of active nodes:

\[
\alpha = \begin{cases} 
1.6, & 1 \leq n \leq 5; \\
2.1, & 6 \leq n \leq 10. 
\end{cases}
\]

\[
\beta = \begin{cases} 
4.3, & 1 \leq n \leq 5; \\
7.0, & 6 \leq n \leq 10. 
\end{cases}
\]
Comparative Evaluations

- 3 and 5 active nodes
- LVCC vs. AIMD rate adaptation
- AIMD is involved in many recent CC protocols for WSNs
- LVCC achieved:
  - controlled behavior in wireless environments
  - smooth throughput
  - friendliness among competing flows
- AIMD caused saw-tooth behavior of traffic flow rates, proved ineffective for wireless streaming environments
Conclusions

- Lotka-Volterra competition model is employed in order to avoid congestive phenomena:
  - control of traffic flows originating from source nodes
  - avoid overwhelming parent node’s buffer
  - allow co-existence of multiple flows
- **Self-adaptation** of traffic flow rate at each source node is achieved
- **Responsiveness** to changes is maintained
- Available buffer capacity at parent node is *fairly shared* among active children
- For small configurations (<20 nodes), system *scales up with number of flows*, offering graceful performance degradation
Future work

• Adaptation of parameter values
  – analytically optimized using conventional techniques
  – Or adopt nature-inspired optimization techniques
• Modify LVCC approach to cope with different priority classes
  – Different kind of traffic flows – different species in nature
• Evaluation of LVCC approach on a real testbed
  – collaboration with Prof. Ahmet Sekercioglu, Monash University, Australia
  – Initial very-small scale experiments are encouraging, involve higher number of active nodes
Generalization of approaches (1/2)

• Generalization of both approaches to other man made systems

• Flock-CC
  – Road transportation
    • Capture interactions in an urban road transportation system
    • Flock-CC for navigating vehicles through congested road networks
    • Example: Google driverless car: like any car, but
      – Uses a series of cameras and laser radar to "see" its environment, react to other vehicles, stop signs, stop lights and other traffic signs
      – It can **steer** itself while looking out for **obstacles**, accelerate to the correct speed limit, stop and go based on any traffic condition
        » Nevada, US, 1st state to allow driverless vehicle to be legally operated on public roads, 1st license May 2012
  – Co-operation of a **swarm of robots** or **Unmanned Aerial Vehicles** (UAVs) moving towards a given target
Generalization of approaches (2/2)

- LVCC
  - Transportation engineering
    - Control of traffic flow injection into freeways/highways
    - Manage traffic flows on access ramps to freeways in order to avoid congestion phenomena, and thus delay for motorists
    - Autonomous Real-time Traffic Injection Control system
      - minimize the overall delay for motorists according to the traffic input load and freeway congestion situation
Thank you!

Are we there yet?
Publications

Book Chapters


Journal Papers


Submitted Journal Papers

Conference/Workshop Papers


Poster


Technical Reports


Repulsion forces

• Packet i repelled from packets on grey-shaded nodes

• Repulsion force proportional to the number of these packets
  – obtained through control packets* broadcasted periodically
  – control packets can be seen as means of transferring knowledge (propagate information) within the environment (sensor network) that is observable by birds' eyes

(*) Control packets are broadcasted periodically (every T seconds, sampling period)
Repulsion forces

Representation of a sensor network

packet $i$ on node $n$
Attraction forces

- Packet i attracted to packets on black-shaded nodes

- **Attraction force** proportional to the number of these packets
  - cannot be obtained timely through control packets
  - black-shaded nodes outside of transmission range
  - *use only locally available information*
  - packet i can perceive packets ‘flying’ from nodes one hop away to nodes two hops away
Attraction forces

Representation of a sensor network

packet $i$ on node $n$
Results – Scenario 1

- Low T: keeps network updated, frequent evaluation of desirabilities $\rightarrow$ desirable nodes change at a fast pace
  - Low number of buffer overflows/high number of collisions
  - Effective when packet spreading is enabled ($\xi=0.5, 0.75, 1$) and a high number of paths to the sinks are available
    - individuals in the flock are allowed to exploit the whole space and move on a balanced way over multiple paths to the sink
  - Ineffective at low $\xi$: coherent flock formation
    - next hop nodes belong to a very small number of closely located paths to the sink
    - proximity of these paths led to very high number of collisions
- High T: infrequent control packet exchanges and desirability evaluations
  - High number of buffer overflows/low number of collisions

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol functions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hop distance: $h_i$</td>
<td>average number of hops: $h_{ij}^{d}$</td>
<td>maximum packet loss rate: $pl(n)$</td>
</tr>
<tr>
<td>channel loading: $s_{nm}^{norm}$</td>
<td>regular pheromone value: $\tau_{ij}^{d}$</td>
<td>available memory: $ma(n)$</td>
</tr>
<tr>
<td>queue occupancy: $q_{mn}^{norm}$</td>
<td>virtual pheromone value: $\omega_{ij}^{d}$</td>
<td>queueing delay: $dl(n)$</td>
</tr>
<tr>
<td>desirability: $D_{nm}$</td>
<td>probability $i$ chooses $n$ (ant packets): $P_{in}^{d}$</td>
<td>normalized remaining energy: $re(n)$</td>
</tr>
<tr>
<td></td>
<td>probability $i$ chooses $n$ (data packets): $P_{nd}$</td>
<td>unicast path delay: $delay(P)$</td>
</tr>
<tr>
<td></td>
<td>reported pheromone value: $\nu_{ij}^{d}$</td>
<td>unicast path packet loss: $packetloss(P)$</td>
</tr>
<tr>
<td></td>
<td>bootstrapped pheromone value: $\kappa_{ji}^{d}$</td>
<td>unicast path energy: $energy(P)$</td>
</tr>
<tr>
<td></td>
<td>cost hopping from $i$ to $j$: $c_{ij}^{d}$</td>
<td>unicast path memory: $memory(P)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>objective function: $f(P_C)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>clustering pheromone value: $\Phi_c(n)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>probability of choosing neighbor (ant): $prop_c(j)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>energy pheromone value: $\epsilon_{r}^{k}(j)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>delay pheromone value: $\delta_{r}^{k}(j)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>packet loss pheromone value: $\epsilon_{l}^{k}(j)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>available memory pheromone value: $\mu_{r}^{k}(j)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>normalized pheromone value: $\Psi_{r}^{k}(j)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>probability of moving from CH to $i$: $P_{c}^{k}(j)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>normalized energy probability: $p_{e}^{k}(j)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>normalized delay probability: $p_{d}^{k}(j)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>normalized packet loss probability: $p_{l}^{k}(j)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>normalized available memory probability: $p_{m}^{k}(j)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol parameters</td>
<td>spreading factor: $\xi$</td>
<td>adaptation to new information: $\alpha$</td>
<td>delay weight factor: $\gamma^d_C$</td>
</tr>
<tr>
<td></td>
<td>sampling period: $T$</td>
<td>speed of adaptation: $\gamma$</td>
<td>packet loss rate: $\gamma^p_C$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>exploratory ant behavior control: $\beta_1$</td>
<td>residual energy ratio: $\gamma^r_C$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>exploratory ant character control: $\beta_2$</td>
<td>available memory for global QoS: $\gamma^m_C$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>forwarding control: $\beta_3$</td>
<td>memory capacity importance: $\alpha$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>energy importance: $\beta$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>energy pheromone improvement factor: $\rho_e$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>delay pheromone improvement factor: $\rho_d$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>packet loss pheromone improvement factor: $\rho_p$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>memory pheromone improvement factor: $\rho_m$</td>
</tr>
<tr>
<td>Types of control messages</td>
<td>Hello</td>
<td>Hello, Forward Ant, Backward Ant</td>
<td>Forward Ant, Backward Ant, Maintenance Ant, Data Ant</td>
</tr>
<tr>
<td>Topology cost for maintenance</td>
<td>None</td>
<td>None</td>
<td>Clustering formation and maintenance mechanisms</td>
</tr>
<tr>
<td>Control packet size</td>
<td>Constant</td>
<td>proportional to the number of hops from source to sink</td>
<td>proportional to the number of hops from source to sink</td>
</tr>
<tr>
<td>Modifications in MAC protocol</td>
<td>None</td>
<td>None</td>
<td>New queueing policies</td>
</tr>
</tbody>
</table>

21/5/2012 Pavlos Antoniou - Ph.D. Defence
2 species Lotka-Volterra model

- Start with logistic growth model for each of the two species.

- Population growth of species 1 depends on population size of species 1 (intra-specific comp.).

- Population growth of species 2 depends on population size of species 2 (intra-specific comp.).

- Now expand models so that growth depends on number of members of the same species and number of individuals of other competing species. (inter-specific)

- $\alpha$ and $\beta$ are termed the competition coefficients.

\[
\frac{dN_1}{dt} = r_1 N_1 \left( \frac{K_1 - N_1}{K_1} \right)
\]

\[
\frac{dN_2}{dt} = r_2 N_2 \left( \frac{K_2 - N_2}{K_2} \right)
\]

\[
\frac{dN_1}{dt} = r_1 N_1 \left( \frac{K_1 - N_1 - \alpha N_2}{K_1} \right)
\]

\[
\frac{dN_2}{dt} = r_2 N_2 \left( \frac{K_2 - N_2 - \beta N_1}{K_2} \right)
\]
2 species Lotka-Volterra model

\[
\frac{dN_1}{dt} = r_1N_1 \left( \frac{K_1 - N_1 - \alpha N_2}{K_1} \right) \quad \frac{dN_2}{dt} = r_2N_2 \left( \frac{K_2 - N_2 - \beta N_1}{K_2} \right)
\]

- \(\alpha\) is a measure of the effect of species 2 on growth of species 1.
- \(\beta\) is a measure of the effect of species 1 on growth of species 2.
- Competition coefficients measure strength of inter-specific competition effects relative to intra-specific competition.
- If \(\alpha > 1\), then competitive effect of species 2 on population growth of species 1 is greater than that of an individual of species 1.
- If \(\alpha < 1\), then competitive effect of species 2 on population growth of species 1 is less than that of an individual of species 1.
Equilibria and Linearization

• System of non-linear differential equations:
  \[ \frac{dx}{dt} = F(x, y), \frac{dy}{dt} = G(x, y) \]

• Study continuous models for two (or more) interacting populations: **linearization at equilibria** \( F(x^*, y^*) = 0, G(x^*, y^*) = 0 \)
  - behaviour of solutions near an equilibrium
  - periodic orbits cannot be revealed

• Classification of equilibria \( x^*, y^* \):
  - **Stable (node):** if every solution \( x(t), y(t) \) (with \( x(0), y(0) \) sufficiently close to equilibrium) remains close to equilibrium for all \( t \geq 0 \)
    - Asymptotically stable: solutions tend to equilibrium as \( t \to \infty \)
  - **Saddle point:** there is a curve through the equilibrium, orbits starting on this curve tend to the equilibrium, orbits starting off this curve cannot stay near the equilibrium
  - **Spiral point or focus:** every orbit wings around the equilibrium
  - **Center:** every orbit is periodic
  - **Unstable**
Equilibria and Linearization (cnt’d)

- Stability/Instability of an equilibrium for the linearization implies stability/instability of the equilibrium of the non-linear system.
- Asymptotic stability/instability for a linear system is determined using the community matrix of the system at the equilibrium.
- Describes the effect of the size of each species on the growth rate of itself and the other species at equilibrium.

\[
A = \begin{pmatrix}
F_x(x^*, y^*) & F_y(x^*, y^*) \\
G_x(x^*, y^*) & G_y(x^*, y^*)
\end{pmatrix}
\]

\[
\det(rI - A) = 0
\]
Classification of equilibria

- Stable point
- Unstable point
- Saddle point (unstable)
- Proper node (sink)
- Unstable point
- Proper node

Center (periodic orbit)
- Stable
- Stable spiral
- Unstable spiral
Lotka-Volterra Equilibria

- In general, model predicts coexistence of two species when inter-specific competition is weaker than intra-specific competition for both species.
- Otherwise, one species is predicted to exclude the other eventually.
- Equilibrium (steady state) population densities at which population growth for the two species stops:

\[
\begin{align*}
N_1^* &= 0 \\
N_2^* &= 0 \\
N_1^* &= K_1 \\
N_2^* &= K_2 \\
N_1^* &= 0 \\
N_2^* &= 0
\end{align*}
\]

\[
A = \begin{pmatrix}
    r_1 & 0 \\
    0 & r_2
\end{pmatrix}
\]

Unstable node

\[
A = \begin{pmatrix}
    -\frac{r_1}{K_1} & -\alpha \frac{r_1}{K_1} N_1^* \\
    -\frac{r_2}{K_2} & -\beta \frac{r_2}{K_2} N_2^*
\end{pmatrix}
\]
Lotka-Volterra Isoclines

- **Isoclines of zero population growth** are straight lines, where everywhere along the line population growth is stopped. ($dN_1/dt = 0$ and $dN_2/dt = 0$)

\[
N_1^* = K_1 - \alpha N_2
\]

Isocline for species 1

\[
N_2^* = K_2 - \beta N_1
\]

Isocline for species 2
Outcomes of Lotka-Volterra model

- Isoclines do not cross and isocline for species 1 lies above that of species 2.
- Species 1 wins (species 2 excluded) with equilibrium for species 1 at its carrying capacity.

<table>
<thead>
<tr>
<th>Case 1</th>
<th>Case 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stable steady state, N1 wins</td>
<td>Stable steady state, N2 wins</td>
</tr>
</tbody>
</table>

- Isoclines do not cross and isocline for species 2 lies above that of species 1.
- Species 2 wins (species 1 excluded) with equilibrium for species 2 at its carrying capacity.

**Competitive exclusion principle:** species less suited to compete for resources should either adapt or die out
Outcomes of Lotka-Volterra model

• Isoclines cross
• Intra-specific competition is stronger than inter-specific competition.
• Stable coexistence at equilibrium.

• Isoclines cross
• Inter-specific competition is stronger than intra-specific competition.
• Unstable equilibrium with eventual exclusion of one of the two species.

Coexistence

Competitive exclusion principle
Stability analysis

• Linearization (Taylor) at equilibrium point
  \[ x^* = \frac{K}{\alpha(n-1) + \beta} \]

• Stability is achieved if all eigenvalues of the community matrix \((A)\) are negative

  – \(n=2\): \( \det(\lambda I - A) = \det \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} - \begin{bmatrix} -\frac{\beta r}{\alpha + \beta} & -\frac{\alpha r}{\alpha + \beta} \\ -\frac{\alpha r}{\alpha + \beta} & -\frac{\beta r}{\alpha + \beta} \end{bmatrix} = \lambda^2 + \frac{2\beta r}{\alpha + \beta} + \frac{r^2(\beta^2 - \alpha^2)}{(\alpha + \beta)^2} = 0 \]

  \[ \lambda_{1,2} = \frac{r}{\alpha + \beta}(\beta \pm \alpha) < 0 \Rightarrow \alpha < \beta \]

  – \(n=3\): using Routh theorem \(\Rightarrow \lambda_{1,2,3} < 0 \text{ iff } \alpha < \beta \)

• Stability of
  \[ x^* = \frac{K}{\alpha(n-1) + \beta} \]
  is achieved when \(\alpha < \beta\)

• Model predicts coexistence of two (or more) species when inter-specific competition is weaker than intra-specific competition for all species
Extreme scenario (1/8)

Set of active nodes, 35 pkts/sec

$t = 37\text{ sec}$
Nodes failed at $t=40s$.
Nodes failed at $t=45s$
Extreme scenario (4/8)

Nodes failed at $t=50s$

$t = 54\ sec$
Extreme scenario (5/8)

Nodes failed at $t=55s$

$t = 59$ sec
Nodes failed at $t=60\text{ s}$
Extreme scenario (7/8)

Nodes failed at t=65s

Nodes failed at t=69 sec
Extreme scenario (8/8)

Nodes failed at t=70s

Nodes failed at t=82 sec
Set of active nodes, 35 pkts/sec

$t = 37$ sec
Extreme scenario 2 (2/11)

Nodes failed at t=40s

Nodes failed at t = 43 sec
Extreme scenario 2 (3/11)

Nodes failed at t=45s

Nodes failed at t = 49 sec
Nodes failed at $t=50s$

$t = 54$ sec
Nodes failed at t=55s
Extreme scenario 2 (6/11)

Nodes failed at $t=60s$

$t = 63s$
Extreme scenario 2 (7/11)

Nodes failed at $t=65s$

$t = 69\ sec$
The network has been almost cut in the middle

Nodes failed at $t=70s$
Some packets “wandering around” at a quest for an alternative path to the sink.
Some packets “discover” a new path to the sink
Two paths towards the sink have been established.
Causes of Congestion in WSNs

• Channel contention and interference
  – Wireless channel is shared among activated nodes
  – Contention occurs when two (or more) neighboring nodes attempt access to shared medium leading to collisions
  – Outgoing channel capacity becomes time variant

• Number of event sources
  – Higher number of event sources improve event detection efficiency
  – Closely located source nodes exacerbate the impact of contention
Causes of Congestion in WSNs

• Reporting rate
  – Increasing reporting rate causes network congestion even if local contention is minimized

• Many-to-one nature
  – Event communication between multiple sources and a single sink causes bottleneck around the sink
Bird flocking behavior

• Evidence that flocks and swarms are self-organising is provided by the ‘boid’ animations of Reynolds*

• Reynolds discovered that convincing animations can result from local, decentralized rules

• Collective group behavior is emergent because rules concerning the parts of the swarm do not contain any notion of the whole

• Early examples of behavioral animations using ‘boids’ include bat swarms and penguin flocks in Batman Returns (1992) and The Lion King (1994)

* Graig Reynolds: artificial life and computer graphics expert, who created the Boids (simulated bird-like objects) in 1986.
Flock-CC: Initial and current model

• Initial attempts for developing Flock-CC:
  – more complex model with
  – four tunable parameters
  – difficult to tune in a number of network and traffic conditions

• Current study:
  – improved Flock-CC model
  – mimics more faithfully bird flocking paradigm
  – simpler, involving two easily interpreted tunable parameters
  – easier to tune and thereafter to deploy
  – Comparably similar performance
Comparative evaluations

- Current Flock-CC vs. initial Flock-CC model
- Low traffic rates (25 pkts/sec): same performance
- Higher traffic rates (35, 45 pkts/sec): initial Flock-CC achieved 1-2% higher PDR and 0.5-1 sec. shorter EED
- Small gains tradeoff versus the complexity of tuning and its universality due the sensitiveness of parameters to the environment
ok class listen up. This problem might seem second nature to you but in some inferior species it is called a "hard" problem.