

Multiple Graphs Models for Networked CPS

John S. Baras

Institute for Systems Research

Department of Electrical and Computer Engineering

Fischell Department of Bioengineering

Applied Mathematics, Statistics and Scientific Computation Program

University of Maryland College Park

NSF CPS: Science of Integration for Cyber Physical Systems

Annual Review Meeting

April 23, 2013

Vanderbilt University

Acknowledgments



- **Joint work with:** Anup Menon, Brian Wang, Ion Matei, Shah-An Yang, Tao Jiang
- **Other collaborators:** Christoforos Somarakis, Sasa Rakovic, Doohyun Sung, Evripidis Paraskevas, Vladimir Ivanov, Shalabh Jain, Johnny Ta
- **Co-funding sponsors:** NIST, AFOSR MURI

Networked CPS



Infrastructure / Communication Networks

Internet / WWW
MANET

Sensor Nets

Robotic Nets

Hybrid Nets

Robotic and
Human Nets

Smart Grid Nets

Social / Economic Networks

Social

Interactions

Collaboration

Social Filtering

Economic

Alliances

Web-based

social systems

Biological Networks

Community

Epidemic

Cellular and

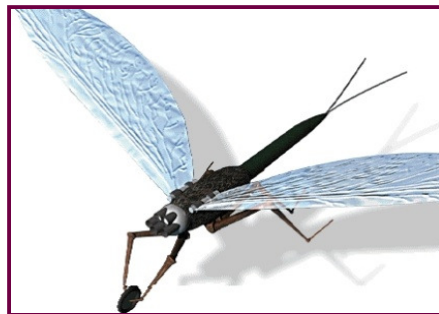
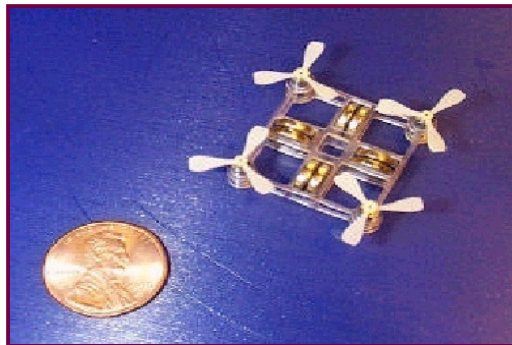
Sub-cellular

Neural

Insects

Animal Flocks

Collaborative Robotic Swarms



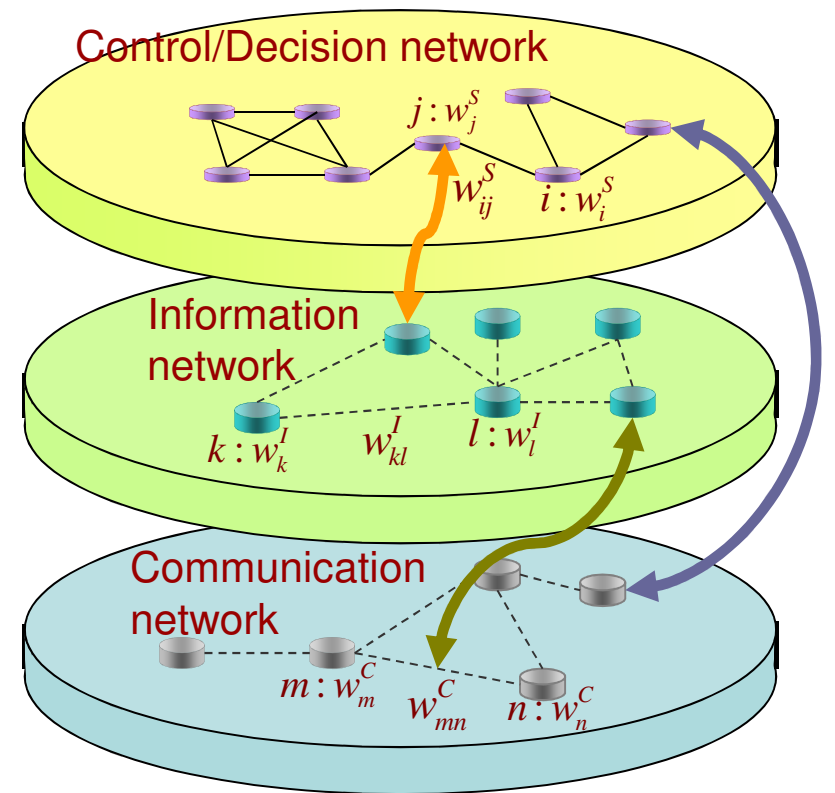
Outline



- **Multiple interacting dynamic graphs models for networked CPS**
- **Networks and Collaboration**
Constrained Coalitional Games
- **Component Based Networking**
- **Topology Matters**
- **Conclusions and Future Directions**

Multiple Interacting Dynamic Graphs

- Multiple Interacting Graphs
 - **Nodes**: agents, individuals, groups, organizations
 - Directed graphs
 - **Links**: ties, relationships
 - **Weights on links** : value (strength, significance) of tie
 - **Weights on nodes** : importance of node (agent)
- Value directed graphs with weighted nodes
- Real-life problems: **Dynamic, time varying graphs, relations, weights, policies**



**Networked System
architecture & operation**

Expander Graphs as Information Patterns for Distributed Control



Most of the literature in distributed control is devoted to –

- Given a distributed plant and an information exchange pattern amongst the control stations, when is the optimal controller linear or the synthesis convex?
- Sufficiency conditions like nested information structures and quadratic invariance that give an affirmative answer are known.

We are interested in the following design question-

Given a plant with a set of (decentralized) control stations, design a “minimal” information exchange pattern that provides desirable control performance.

Main Obstacles

- Optimizing over information patterns is **combinatorially hard** →
Understand features of the 'right' information pattern
- Given an information pattern, optimal controller is **not necessarily linear/convex** →
Make context dependent simplifying assumptions

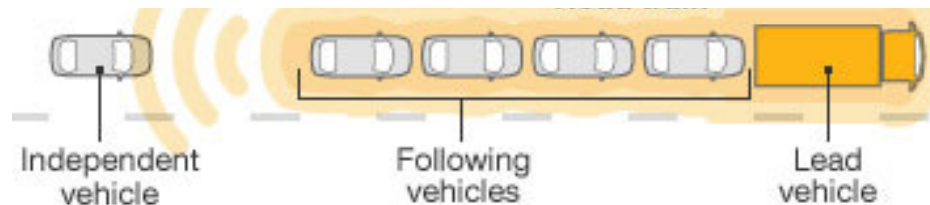
An Example problem of the Interaction between the Control Graph and the Communication Graph

An Example: Vehicle Platooning

Consider an Intelligent Vehicle Highway System (IVHS) where a number of vehicles heading to a common destination form a platoon or a road train.

Advantages-

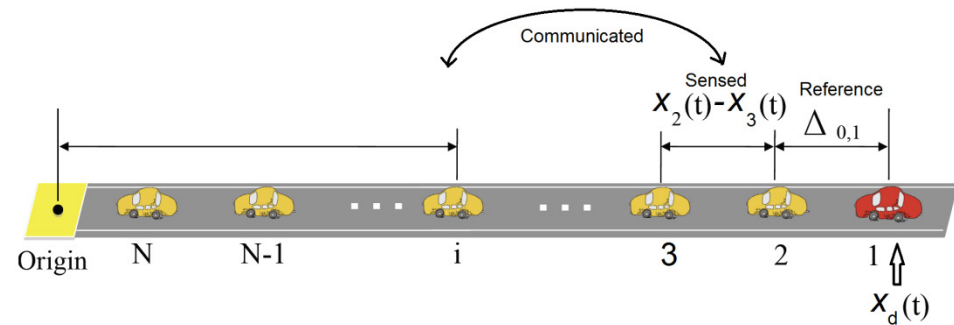
- improved highway throughput and
- reduced fuel consumption.^[1]



High Speeds, Close Spacing and Multiple Vehicles
➔ Need Automatic Distributed Control

- Vehicles have identical linear dynamics
- Only lead vehicle is given desired trajectory information $x_d(t)$.
- **Symmetric Control**: i applies a linear feedback law based information available

$$u_i = \frac{1}{deg(i)} \sum_{j \in \mathcal{N}(i)} [-k(x_i - x_j - \Delta_{i,j}) - b(\dot{x}_i - \dot{x}_j)] + \delta(1, i)[-k(x_1 - x_{1,d}) - b(\dot{x}_1 - \dot{x}_{1,d})]$$



Control objective:
Regulation- maintain prescribed reference inter-vehicle spacing.

If the information is restricted to the **nearest neighbor type**, then

- The least damped eigenvalue of the closed loop matrix scales as $O(1/N^2)$ [2].
- **String instability** is inevitable- disturbances acting on an individual grow without bounds in the size of the platoon [3].
- It is **not possible to achieve coherence** or resemblance to a rigid lattice as the formation moves[4].

Bottom line:
Nearest neighbor type information patterns lead to inadequate control performance.

[2] He Hao, Prabir Barooah, J.J.P. Veerman. "Effect of Network Structure on the Stability Margin of Large Vehicle Formation with Distributed Control", CDC2010

[3] Seiler, P. Pant, A. Hedrick, K. "Disturbance propagation in vehicle strings", Trans. Automatic Control 2004

[4] Bamieh B., Jovanović M.R., Mitra P., Patterson S., "Coherence in Large-Scale Networks: Dimension-Dependent Limitations of Local Feedback", 2012, To appear in Trans. Automatic Control 9

Our Approach: Towards More General Information Patterns

- Let the agents (vehicles) exchange information over an arbitrary symmetric graph G .

- The resulting closed loop system (with the controls in the previous slide) is

$$\dot{z} = (I \otimes A_1 + (L + D_{ext}) \otimes A_2)z = A_{cl}z$$

where $A_1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $A_2 = \begin{bmatrix} 0 & 1 \\ -k & -b \end{bmatrix}$, $z = \begin{bmatrix} x_1 - x_{1,d} \\ \dot{x}_1 - \dot{x}_{1,d} \\ \vdots \\ x_N - x_{N,d} \\ \dot{x}_N - \dot{x}_{N,d} \end{bmatrix}$,

L is the normalized Laplacian and D_{ext} is a diagonal matrix with the first element in the diagonal being 1 rest all 0s.

- A_{cl} is Hurwitz with spectrum $\sigma(A_{cl}) = \bigcup_{\gamma \in \sigma(L + D_{ext})} \{\sigma(A_1 + \gamma A_2)\} = \bigcup_{\gamma \in \sigma(L + D_{ext})} \left\{ \sigma \begin{bmatrix} 0 & 1 \\ -k\gamma & -b\gamma \end{bmatrix} \right\}$.
- The least damped eigenvalue of A_{cl} is proportional to the smallest eigenvalue of $L + D_{ext}$. Thus, we define the **stability margin** of the closed loop system to be

$$\gamma_{min} = \min \sigma(L + D_{ext}) .$$

Main Result [5]

Theorem: Let λ_{\min} be the second smallest eigenvalue of the normalized Laplacian L . Then

$$\frac{\lambda_{\min}}{4N} < \gamma_{\min} \leq \lambda_{\min}.$$

Proof sketch: Recall $\gamma_{\min} = \min \sigma(L + D_{\text{ext}})$. We exploit the special structure of the Laplacian.

➤ Since L is symmetric, there exists an orthogonal matrix P such that $P^T L P$ is a diagonal. Special structure: last column of P is along the vector $[1, \dots, 1]^T$.

➤ Apply similarity transformation $P^T (L + D_{\text{ext}}) P$.

➤ By performing suitable column transformations represented by the matrix \tilde{T} ,

$$\tilde{T}^{-1} P^T (L + D_{\text{ext}}) P \tilde{T} = \begin{bmatrix} \tilde{\Lambda} & r_k(N) \tilde{r}_k \\ \frac{1}{r_k(N)} \tilde{r}_k^T \tilde{\Lambda} & 1 \end{bmatrix}, \text{ where } \tilde{\Lambda} \text{ is a diagonal matrix with the nonzero eigenvalues of } L \text{ and } r_k \text{ are rows of } P.$$

➤ Next, we exploit the special structure of the above matrix to calculate its characteristic polynomial. Solving a first order Taylor approximation around zero yields the desired lower bound. Upper bound follows from standard inequalities.

Choosing the Right Information Pattern

Information pattern	Communication load $\sim \text{Edges} $	Stability margin
Nearest neighbor type	$O(N)$	$O(1/N^2)$
Complete graph	$O(N^2)$	At most $O(1/N)$

- Is there something in between? Does there exist a “family” of graphs such that one can get improved control performance while limiting the communication load?
- **An Expander family** is a sequences of d -regular graphs with increasing number of vertices such that the second smallest eigenvalue of the Laplacian is bounded away from zero.
- An immediate consequence of the Theorem from the previous slide is:

Expander families	$O(N)$	At most $O(1/N)$
-------------------	--------	------------------

Expander Families or Expander Graphs



- **Edge expansion, $h(G)$** , is a quantitative measure of connectivity of the graph $G=(V,E)$ given by

$$h(G) = \min_{S \subset V, 0 < |S| \leq |V|/2} \frac{|E(S, V - S)|}{|S|}.$$

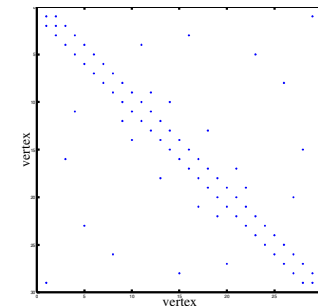
- Combinatorial definition: An **expander family** of graphs is a sequence of d -regular graphs $\{G_n\}$ with $|V_n| \rightarrow \infty$ as $n \rightarrow \infty$ such that there exists an $\epsilon > 0$ and $h(G_n) > \epsilon$ as $n \rightarrow \infty$.
- Equivalent characterization due to Cheeger: A sequence of graphs such that there exists an $\epsilon > 0$ such that $\lambda_{min} > \epsilon$ as $n \rightarrow \infty$.
- There has been extensive research on expanders. They've found applications in theoretical computer science, in constructing good error correcting codes, random walks converge faster on them, etc.
- Examples of known **explicit constructions** include
 - Cayley graphs of certain finite groups and
 - based on a technique called zig-zag product of graphs.

Expander Family for the Platooning Problem

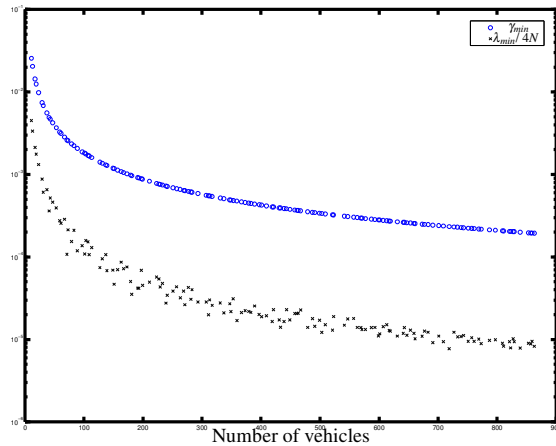
- How to pick the right expander family for the platooning problem?
 - involves considerations about the **communication network** used to implement the **information pattern**.
- Assuming the vehicles use a multi-hop wireless network to communicate, this involves considerations such as reducing the maximum hop length corresponding to the “edges” in the information pattern.
- We do not address this problem directly and only give an example of an 3-regular expander family from [6]. The desirable feature of this family is that two of the three links of each vertex correspond to the sensed distance from its neighbors. However the construction is only valid for a prime number of vertices.

An example expander family- Let $\{p_i\}$ be an infinite sequence of increasing primes. The 3-regular family of graphs $\{S_{p_i}\}$, $S_{p_i} = (V_i, E_i)$ with $V_i = Z_{p_i}$ and for every $a \in V_i$, $(a, a + 1)$, $(a, a - 1)$ and $(a, a^{-1}) \in E_i$ is an expander family.

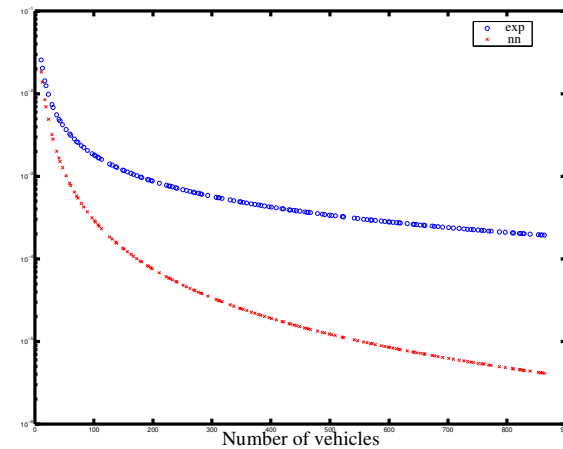
Sparcity plot of an element of the expander family-



Some Numerical Simulations and Next Steps



- An experimental verification of the main result stated earlier. The plot of the stability margin γ_{min} is above the lower bound $\frac{\lambda_{min}}{4N}$.



- Experimental verification that expanders outperform nearest neighbor type information patterns. Plot of stability margins with expanders serving as information pattern is above that with nearest neighbor type.

Next steps-

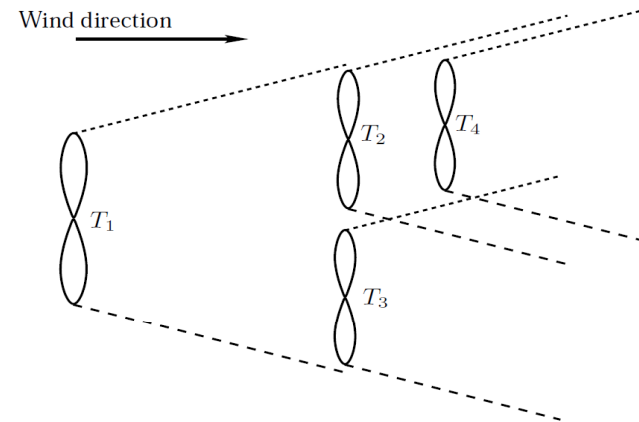
- Investigate the problem under other metrics of control performance like string stability, coherence etc.
- How to synthesize the right expander family? Formulating a way of incorporating communication network constraints in the problem of choosing expanders.
- More general scenarios for answering the question **“the right information pattern for a given collaborative control task”**.

Interaction Between Control and Communication Graphs: Agents Learn What is Best for the Team

Example: Maximizing Power Production of a Wind Farm^[1]



A Wind Farm. Courtesy:<http://www.dis.anl.gov/>



Schematic representation of a wind farm depicting individual turbine wake regions.

- Aerodynamic interaction between different turbines is not well understood.
- Need on-line decentralized optimization algorithms to maximize total power production.

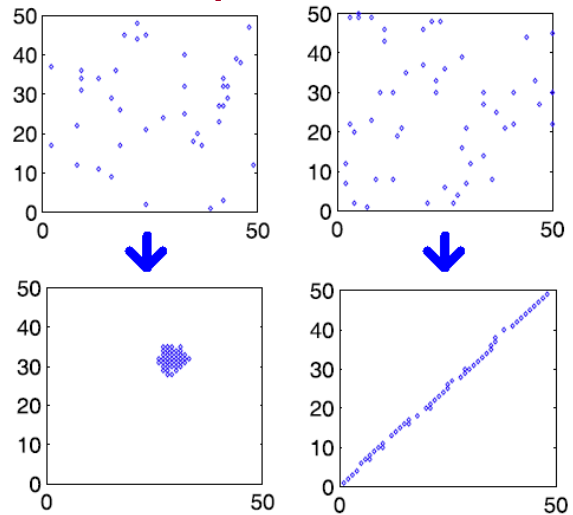
Assign individual utility

$u_i(t)$ = power produced by turbine i at time t
such that maximizing $\sum_i u_i(t)$ leads to desirable behavior.

[1]. Marden et. al., "A model free approach to wind farm control using game theoretic methods", 2012, under review.

Interaction Between Control and Communication Graphs

Example: Formation Control of Robotic Swarms



Simulation results demonstrating rendezvous and gathering along a line^[2]

- Deploy a robotic swarm in unknown environment: obstacles, targets etc. have to be discovered.^[3]
- The swarm must form a prescribed geometric formation.
- Robots have limited sensing and communication capabilities.

For rendezvous, design individual utility

$$u_i(s_i) = \frac{1}{|\{s_j \in S: |s_i - s_j| < r\}|} - \alpha \text{dist}_r(s_i, \text{obstacle}),$$

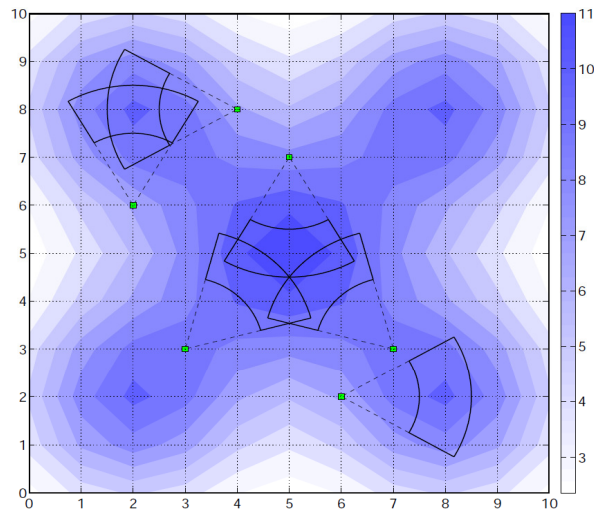
such that minimizing $\sum_i u_i(t)$ leads to desirable behavior.

[2] Xi, Tan and Baras, "Decentralized coordination of autonomous swarms using parallel Gibbs sampling", Automatica, 2010.

[3]. Baras et. al., "Decentralized Control of Autonomous Vehicles", Proc. of 42nd IEEE CDC, 2003.

Interacting Control, Information and Communication Graphs

Example: Mobile Visual Sensor Network Deployment



Darker the shade of blue, more the interest in the site. Sectors represent sensor

- We wish to monitor events in different sites of varying interest levels.
- All robots monitoring a small set of high interest sites is undesirable w.r.t. coverage.
- Cost associated with information processing.
- How to deploy so “effective coverage” is ensured at “reasonable cost”.

Design individual utility

$$u_i(s, c) = \sum_{s' \in NB(s, c)} \frac{q(s')}{n(s')} - f_i(c),$$

such that maximizing $\sum_i u_i(t)$ leads to desirable behavior.

(here $q(s)$ = interest in observing s , $n(s)$ = number of agents observing s , $NB(s, c)$ = subset of S observable from s when camera viewing angle = c , and $f_i(c)$ = processing cost when the camera viewing angle is c .)

Game Theoretic Control: A Multi-agent Control Paradigm



1. Assign *individual utility*, u_i , based on local measurements/information such that a solution concept such as Nash Equilibrium (NE) corresponds to desirable *system-wide* behavior.
2. Prescribe *Learning Rules*, studied in the repeated games literature, to agents to enable them to learn to play actions that correspond to such concepts.

Shortcomings of existing learning rules:

- Need **assumptions on structure of game** (like potential, weakly-acyclic, congestion games etc.).
- Equilibrate to **NE**; if utilities are not designed carefully these **can be inefficient**.
- Gradient based rules require knowledge of **functional form** of utilities and converge to **local extrema**.

Our approach is complementary to the existing work- instead of focusing on designing utilities with special structure, we devise distributed algorithms that are:

- **online** (functional form of utilities not needed);
- **no assumption on structure of utilities**;
- lets **agents learn welfare optimal** (Pareto dominant if maximal) actions.

Setup



- N agents, indexed by i
- Agent i picks actions from a finite set A_i
- Agent i receives/measures private utility $u_i: A \rightarrow \mathbb{R}$ where $A = \prod_{i=1}^N A_i$ is the set of joint actions.
- We wish to maximize the welfare function $W(a) = \sum_{i=1}^N u_i(a)$ over the set A i.e. we seek *efficient actions*

$$A^* = \{\mathbf{argmax}_{a \in A} W(a)\}.$$

Difficulties-

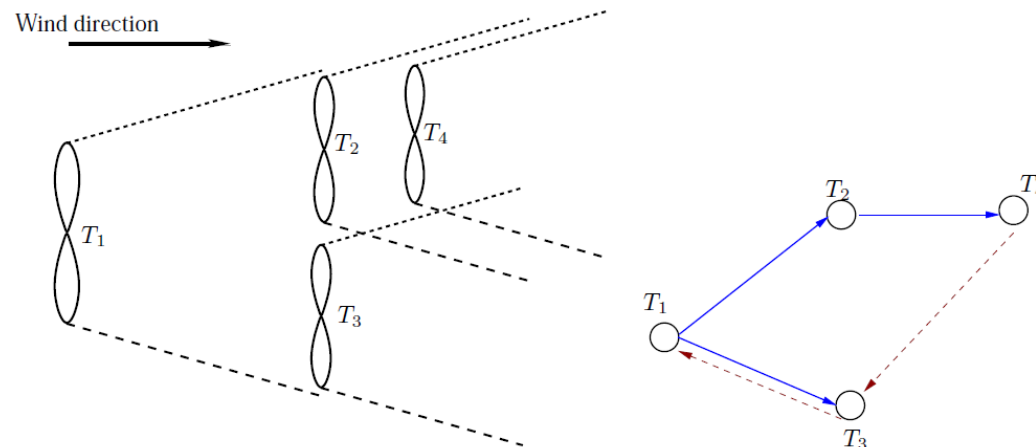
- No information exchange between agents.
- No a priori information about the structure of the game like potential game, weakly acyclic game etc.
- It's a general combinatorial problem and is NP hard.
- Given no explicit information exchange, how would agents even identify an element of A^* ?

System Designer's Perspective

Like agents, system designer does not know exact functional form of the payoffs.
→ The system designer may have “coarse information” about which agents' action can affect which others.

Interaction graph models such coarse information: It's a directed graph where a link from i to j implies actions of agent i affect the payoff of agent j .

Communication graph models explicit inter agent communications: It's a directed graph where a link from i to j implies agent i can send information to agent j .

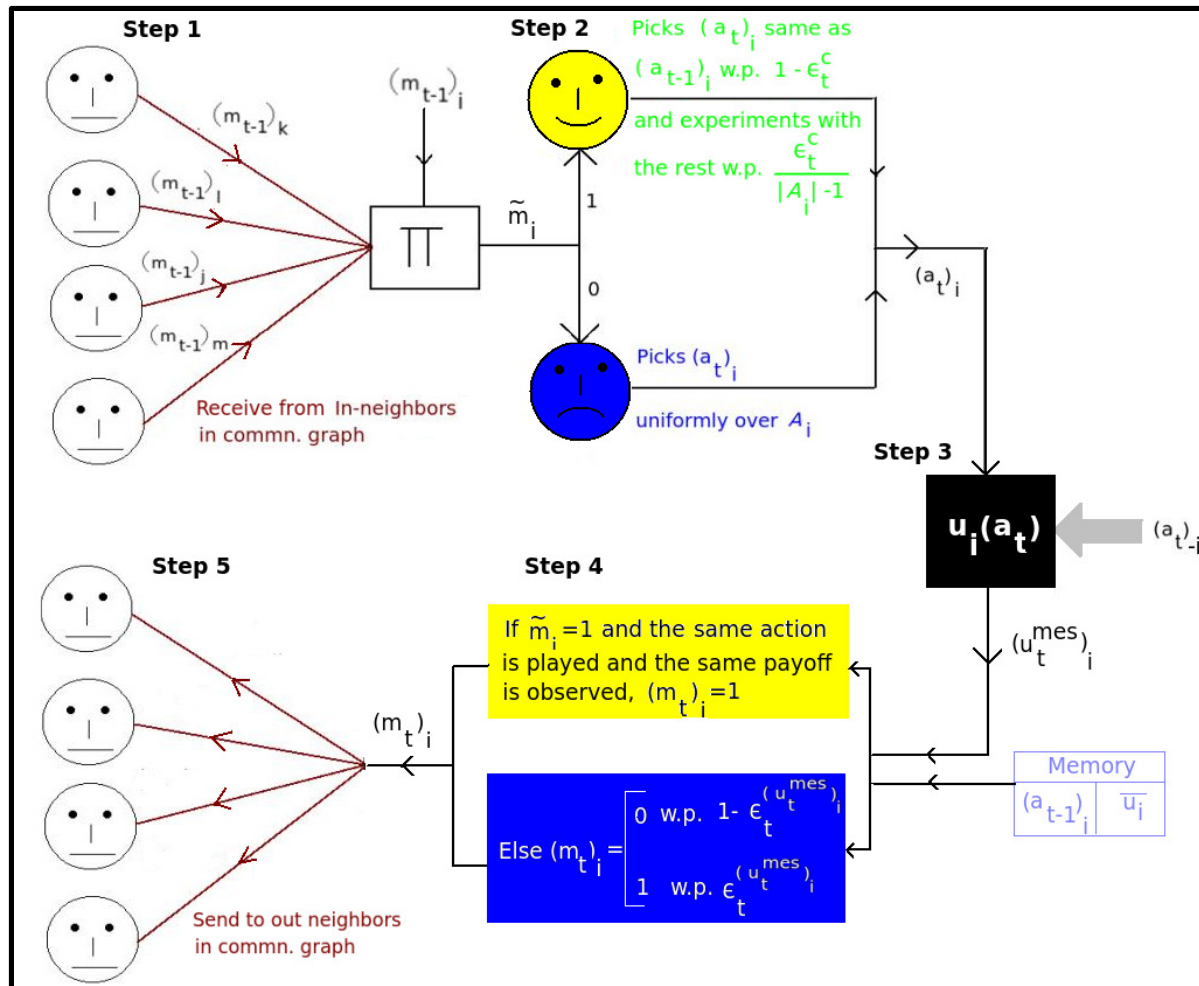


The wind farm example is considered in the figure:

- blue lines are edges in the interaction graph and,
- the red lines in the communication graph.

Proposed Algorithm

Based on Marden et al^[5], endow each agent with a state $x_i = (a_i, m_i)$; $m_i \in \{0,1\}$ is the mood of agent i , with 1 corresponding to a “content” agent and 0 to a “discontent” one.



Differences from the algorithm in [5] :

- No explicit inter-agent communication is used in [5].
- Some assumptions on utilities are made in [5] to prove feasibility.
- ϵ_t is held constant for some $\epsilon > 0$ in [5].

[5]. Marden, Young and Pao, “Achieving Pareto optimality through distributed learning”, 2011, Under review.

Theorem^[6]

Assume

1. $\sum_{t=1}^{\infty} \varepsilon_t^c = \infty$ and;
2. for each $a \in A$, $G_c(a) \cup G_I(a)$ is strongly connected.

Then,

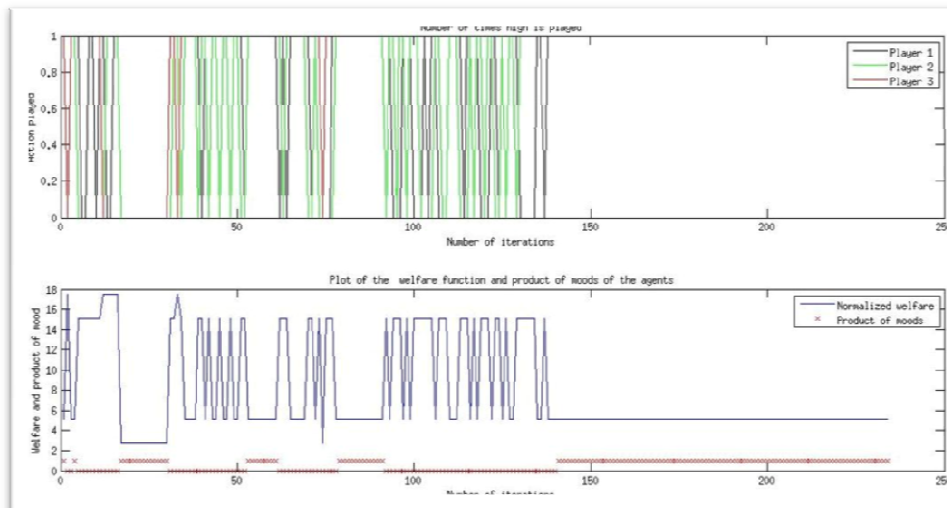
$$\lim_{t \rightarrow \infty} P(a_t \in A^*) = 1.$$

Where $G_c(a)$ is the communication graph and $G_I(a)$ is the information graph when joint action is a .

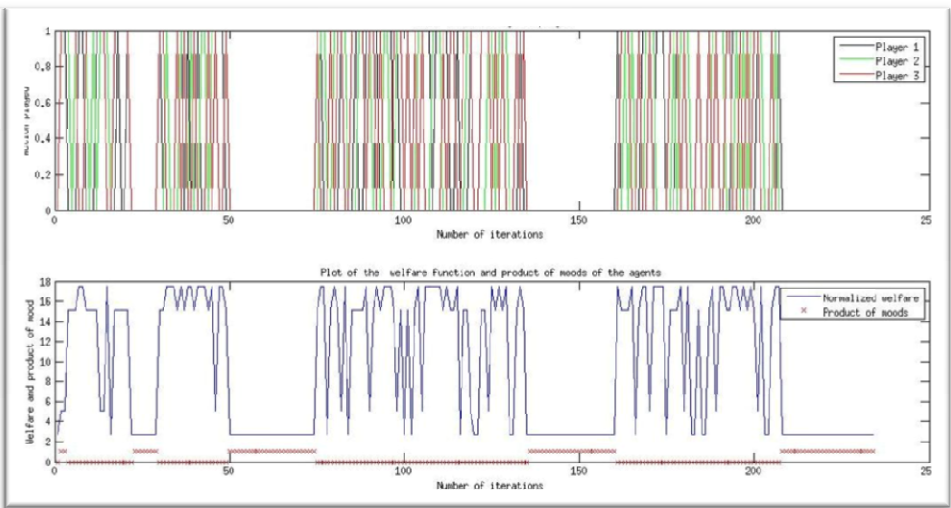
- The algorithm is model free; if nothing is known about $G_I(a)$, design $G_c(a)$ strongly connected for all a to ensure convergence.
- The more “knowledge” about $G_I(a)$ the designer has, the lesser the demands on $G_c(a)$.
- Communication is only bit-valued: simple implementation.
- The scheduling law $\varepsilon_t = \frac{1}{c\sqrt{t}}$ works.

An Example

Player 3 →		l		h	
		Player 2			
Player 1		l		h	
		l		h	
	l	(0.1, 0.1, 2.5)	(5, 10, 2.5)	(7.5, 7.5, 0.1)	(10, 5, 0.1)
	h	(10, 5, 2.5)	(7.5, 7.5, 2.5)	(5, 10, 0.1)	(2.5, 2.5, 0.1)



No Communication: Since 3 cannot be influenced, it learns to play high. The actions converge to (h; h; h) with suboptimal welfare value 5.6.



Communication from 1 to 3: The agents learn to play (l; l; l) with minimal welfare value 2.7.

Proof Sketch: Overview

Fix $\varepsilon_t \equiv \varepsilon > 0$.
The algorithm describes a
irreducible, aperiodic
Markov chain $P(\varepsilon)$.

Theory of Perturbed
Markov Chains
developed by Young



Stationary distribution of
 $P(\varepsilon)$, $\mu(\varepsilon)$, converges $\mu(\varepsilon) \rightarrow$
 $\mu(0)$, as $\varepsilon \rightarrow 0$.
Support of $\mu(0)$, are states
where $a \in A^*$, $m_i = 1 \forall i$.



Rate conditions on ε_t to
ensure ergodicity of $\mathbf{P}(t)$ with
 $\mu(0)$ as stationary
distribution.

Results on ergodicity
of nonhomogeneous
Markov chains



Let ε vary as ε_t . Obtain
nonhomogeneous Markov
chain $\mathbf{P}(t) = P(\varepsilon_t)$.

Conclusions and Future Directions

- Relevance of Welfare extremum seeking.
- A model free, on-line, decentralized algorithm is proposed.
- Explicit inter-agent communication eliminates assumptions on utility functions.
- Introduced interaction graph for analysis and help identify extent of explicit communication.
- Guarantees for convergence in probability.

Future work:

- Nothing about the rate yet! How long does it take to converge?
- What effect does the structure of communication have on the rate of convergence?
- Similar algorithms for the continuous actions space setting?
- With some model information, gradient descent etc. can help converge faster.

Storage-centric Wireless Sensor Networks for Smart Buildings

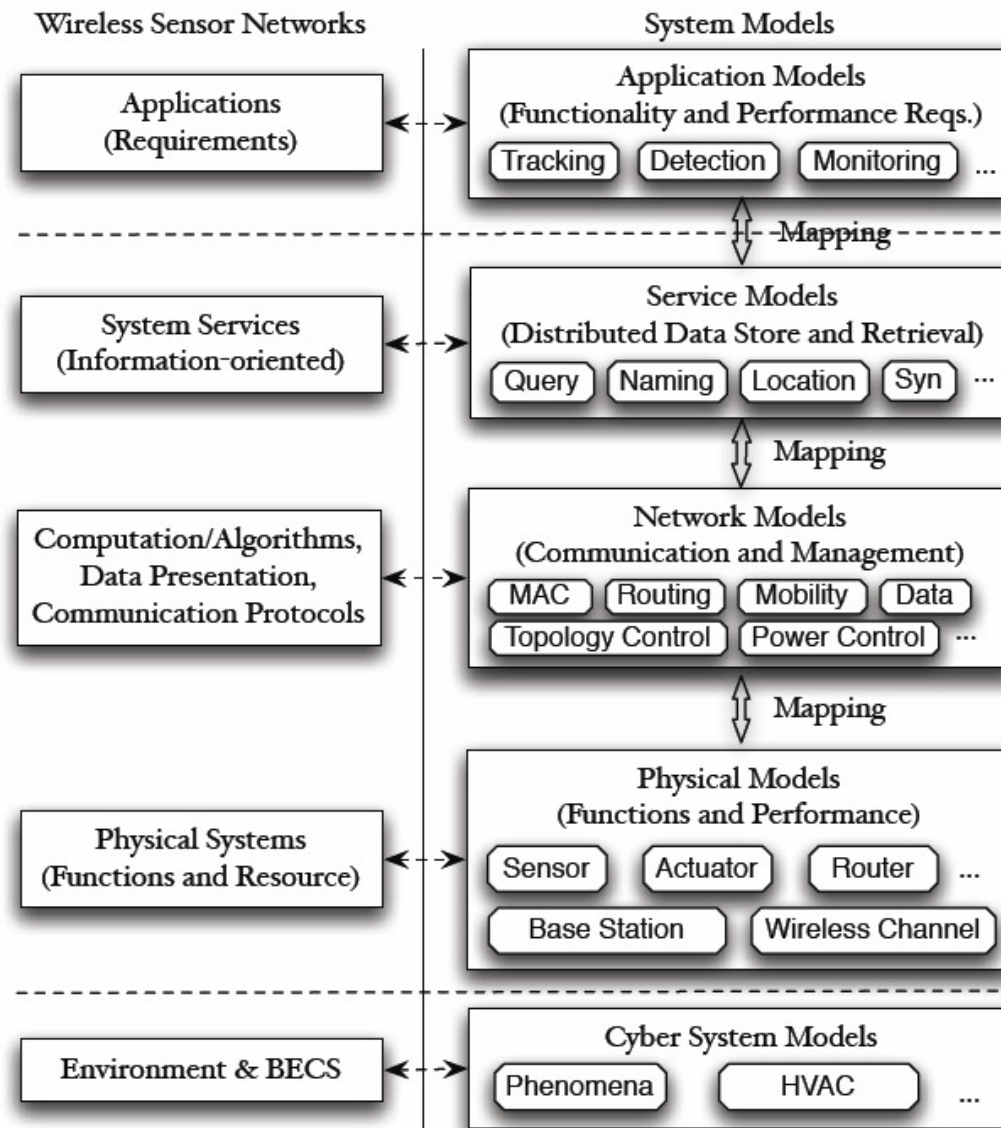


Interacting Information and Communication Graphs

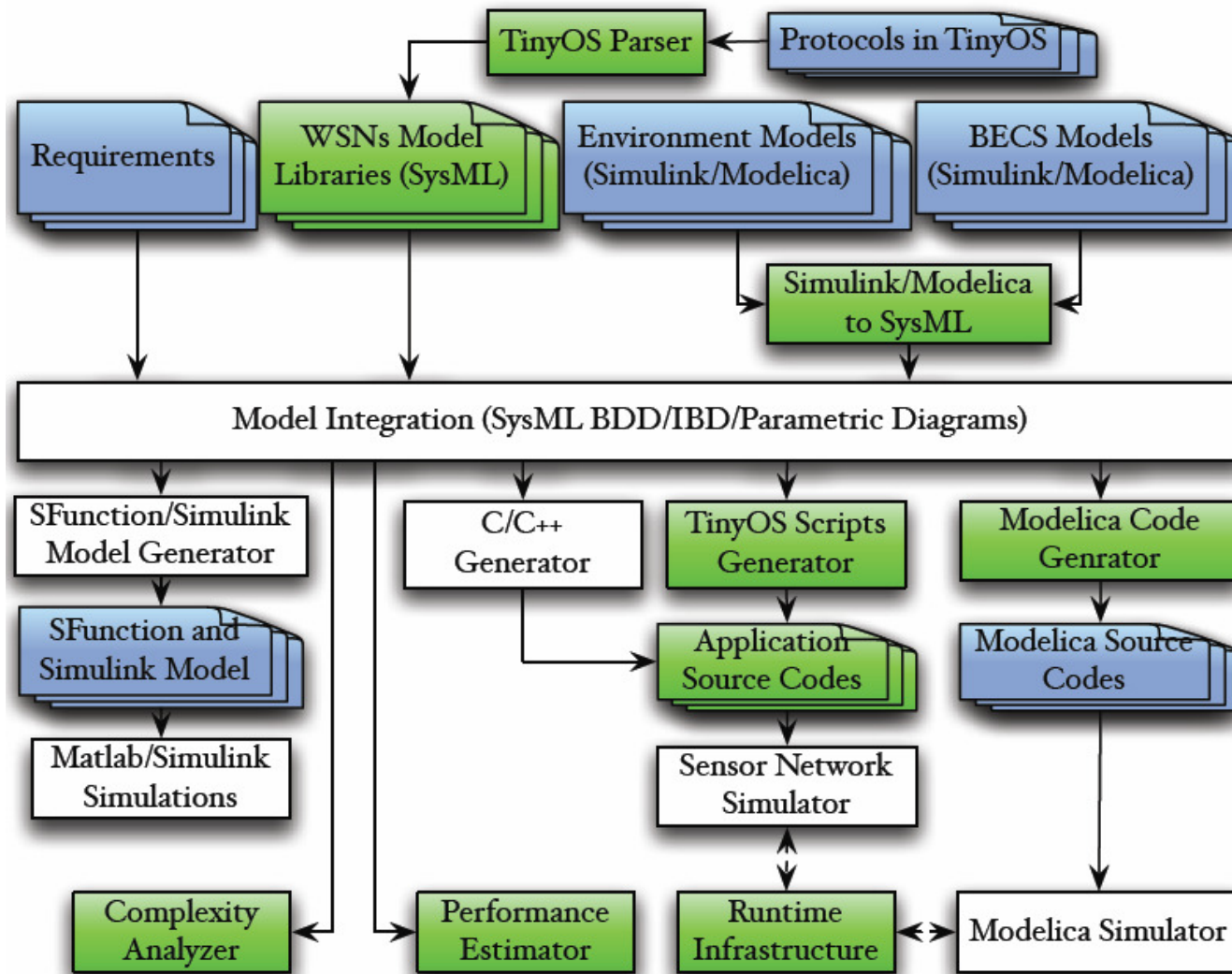
- Buildings consume 40% of total energy in the US
 - Main: HVAC and lighting systems
 - Electric plug-loads: nearly 30% in commercial buildings
- WSNs are critical for Smart Buildings
 - Collect **real-time** data for smart HVAC and lighting
 - Collect **historical** data for energy usage pattern analysis
- Difficult to design efficient and reliable WSNs
 - Collaboration across multiple engineering domains
 - Complex cyber-physical interactions
 - Component reusability
 - Massive data collection and processing

- Q1: How to develop an integrated framework for the design across multiple engineering domains?
 - **WSNDesign: Model-based Systems Design Framework**
 - Model libraries and integration
 - Theoretical performance estimation
 - Automatic code generation and integrated simulation
 - Reduce the complexity of system analysis
- Q2: How to store and retrieve large amount of sensor readings efficiently?
 - **Flash-based Data Storage and Retrieval**
 - Node-level energy-efficient data storage system
 - Distributed database system supporting approximate querying

WSN Design Overview: Model Libraries



WSNDesign Overview: Design Flow



Model Libraries

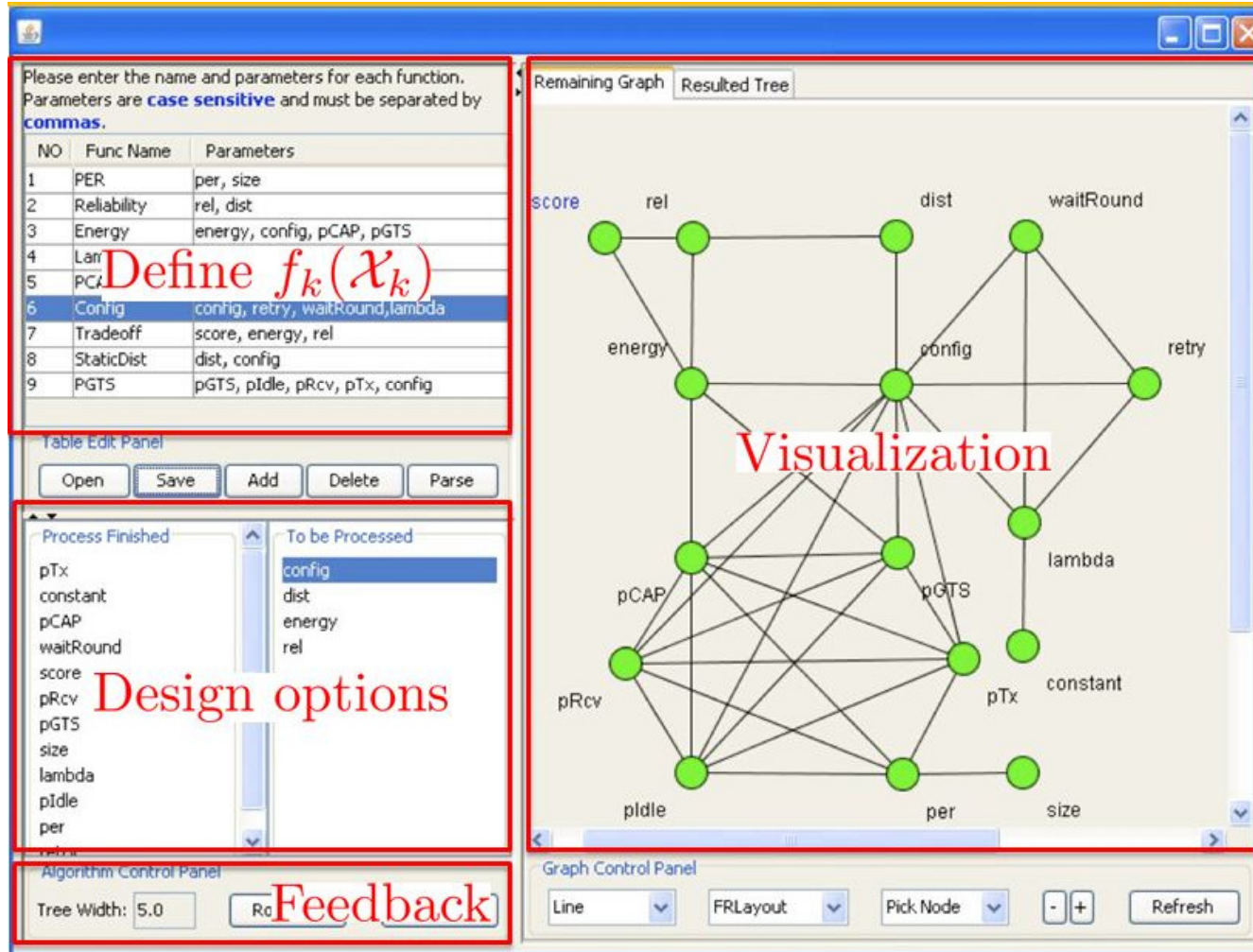


- Develop model libraries using SysML, Modelica and Simulink
- Model Wireless Sensor Networks
 - Physical platforms: CPU, sensor, RF transceiver, and battery
 - MAC layer protocols: Low Power Listener, CSMA/CA Channel Access, CSMA/CA Sender, MAC Controller, Slot Manager, Queue Manager, TDMA Sender, Receiver...
 - Wireless channels: radio propagation models, channel fading models, and bit error rate
- Model Cyber Systems
 - Phenomenon: interface between the event-triggered domain and continuous-time domain
 - Environment: propagation of phenomenon signals
 - Control logic
- Case study: building thermal control system

Complexity Analysis and Reduction

- **Complexity of system analysis**
 - Exponential exploration space: D^N
 - Overall system = local analysis + composition rules
 - Reduced complexity: $\sum_{i=1}^k D^{n_j}$
- **Drawbacks of existing work**
 - Ad-hoc partitioning
 - Rely upon general rules of thumb
 - Rely upon the expertise of systems engineers
- **Our Objective**
 - Visualize and quantitate the complexity of system analysis
 - Help system engineers understand the impact of decisions
 - Give hints to system engineers to improve their designs

Complexity Analysis and Reduction



Please enter the name and parameters for each function. Parameters are **case sensitive** and must be separated by **commas**.

NO	Func Name	Parameters
1	PER	per, size
2	Reliability	rel, dist
3	Energy	energy, config, pCAP, pGTS
4	Lam	
5	PCAP	
6	Config	config, retry, waitround, lambda
7	Tradeoff	score, energy, rel
8	StaticDist	dist, config
9	PGTS	pGTS, pIdle, pRcv, pTx, config

Define $f_k(\mathcal{X}_k)$

Table Edit Panel

Process Finished
 pTx
 constant
 pCAP
 waitRound
 score
 pRcv
 pGTS
 size
 lambda
 pIdle
 per

To be Processed
 config
 dist
 energy
 rel

Design options

Algorithm Control Panel
 Tree Width: 5.0 **Feedback**

Remaining Graph Resulted Tree

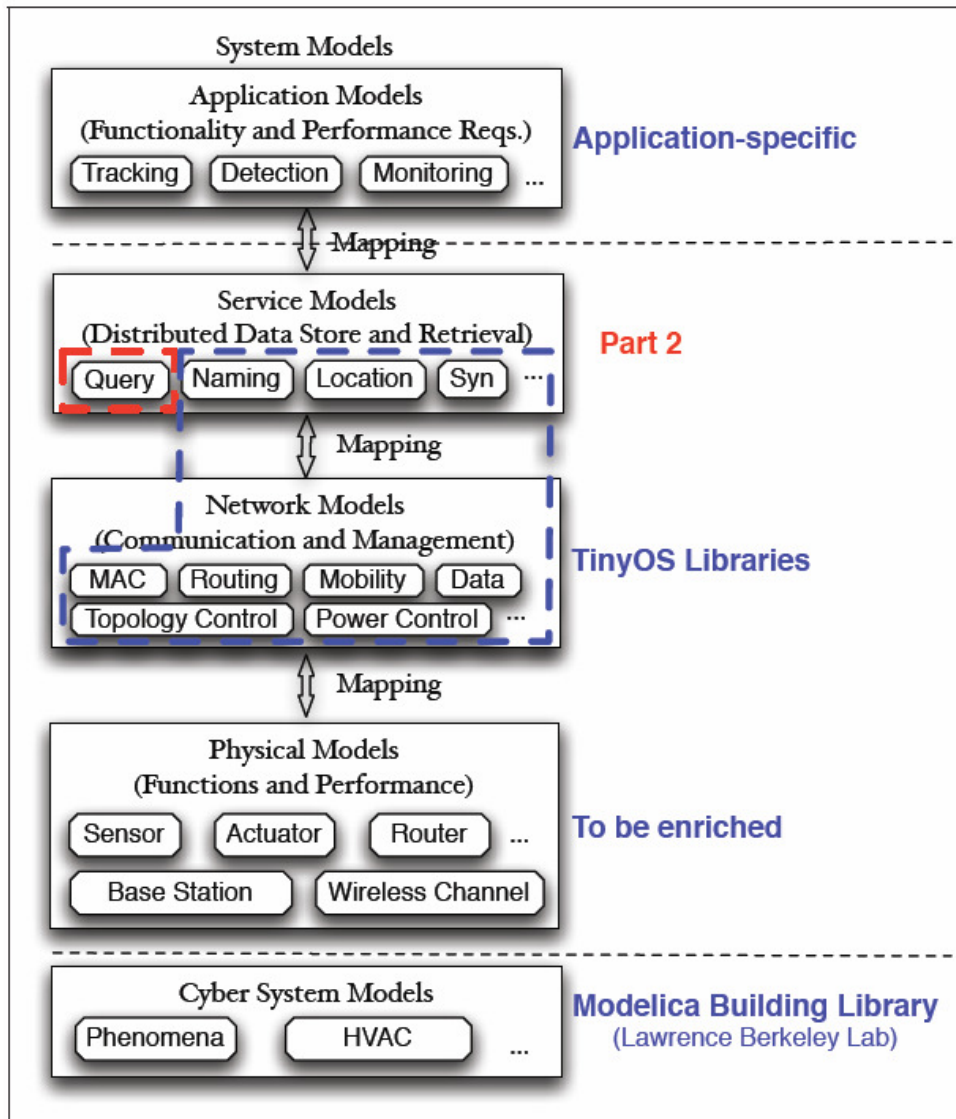
score rel dist waitRound
 energy config retry
 pCAP pGTS lambda
 pRcv pTx constant
 pIdle per size

Visualization

Graph Control Panel
 Line FRLayou Pick Node - + Refresh

- **Model Library Development**
 - Transform and import TinyOS libraries
 - Integrate System-Modelica profile with IBM Rhapsody
 - Enrich the Physical Model Library
- **Code Generation and Simulation Integration**
 - Generate TinyOS configuration components
 - Generate Modelica wrapper components
 - Synchronize the IBM Rhapsody (**SysML**) with TOSSIM (**TinyOS**) and OpenModelica simulator (**Modelica**)
- **Complexity Reduction**
 - Parse hierarchical SysML Parametric Diagrams
 - Generate improve block diagrams from analytical results

Flash-based Data Storage and Retrieval



- Not every readings is needed
 - Aggregated data
 - Approximate data
- Data storage schemes
 - **Centralized** data collection
 - **In-situ** data storage
 - Flash memory: high capacity, energy efficient

Problem Formulation



- N nodes, sampling periodically
- **Approximate Querying**
 - Retrieve a subset of a given original dataset
 - Compute an approximate dataset from this subset
 - The maximum error of each record is bounded
- **Drawbacks of Approximate Querying**
 - Error bound must be specified by users
 - What error bound can lead to satisfactory result?
 - Too tight: over-qualified result, energy waste
 - Too loose: re-issue the query, duplicated data retrieval

- **Incremental Approximate Querying**
 - Divide a query to a sequence of sub-queries
 - Reduce the error bound gradually
 - Retrieve an **incremental** dataset for each sub-query
 - Issue sub-queries until a satisfactory result is got
- **Our Approach**
 - Q2.1: How to get a dataset satisfying the given selection predicate? → **HybridStore**
 - Q2.2: How to compute an incremental set efficiently? → **HybridDB**

HybridStore Interface

- `insert(float key, void* record, uint8_t length)`
- `select(uint32_t t1, uint32_t t2, float k1, float k2)`

- **HybridStore Features**

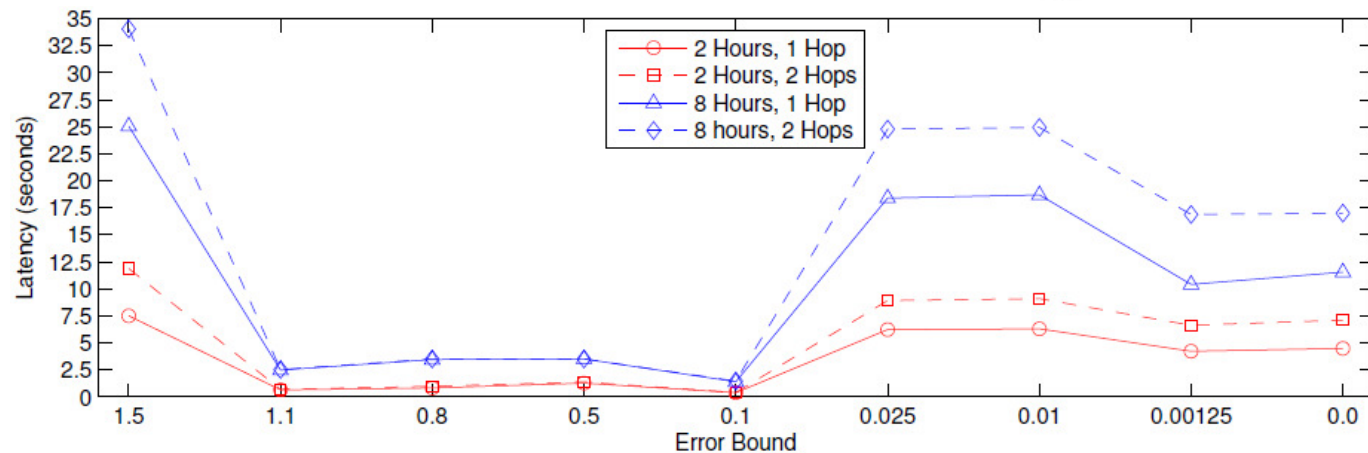
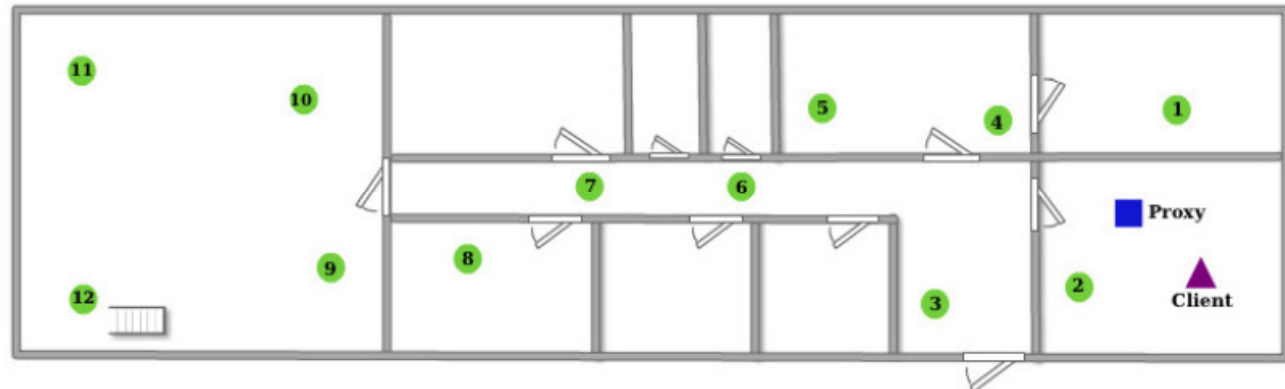
- All NAND pages are **fully occupied** and written **sequentially**
- In-place updates and out-of-place writes are **completely avoided**
- Process typical joint queries efficiently, even on **large-scale** datasets
- **Data aging** without overhead
- **Sensor-friendly**: 16.5KB ROM and 3.2KB RAM in TinyOS

HybridDB Interface [TOSN '13]

- `approxQuery(uint32_t t1,1, uint32_t t1,2, float k1, float k2, float ϵ_1)`
- `approxUpdate(uint8_t queryID, uint32_t ti,1, uint32_t ti,2, float ϵ_i)`
- **HybridDB Features**
 - Support refinement and zoom-in sub-queries
 - Retrieve an approximate dataset with arbitrary error bound
 - Balance trade-offs
 - Energy consumption: sensor \leftrightarrow proxy
 - Response time: current sub-query \leftrightarrow following sub-queries

Implementation and Evaluation

- TinyOS implementation: 22.5KB ROM, 3.76KB RAM
- Benefits: significant energy savings, much better user experience



Conclusions



- Proposed a mode-based design framework for the design of WSNs in the context of Smart Buildings
 - Model libraries, transformation and integration
 - Composition rules, and system performance estimation
 - Code generation, and multi-simulator integration
 - Reduction of system analysis complexity
- Proposed and implemented an abstraction for in-situ data storage and retrieval
 - Efficient light-weight data storage system
 - Distributed incremental approximate querying

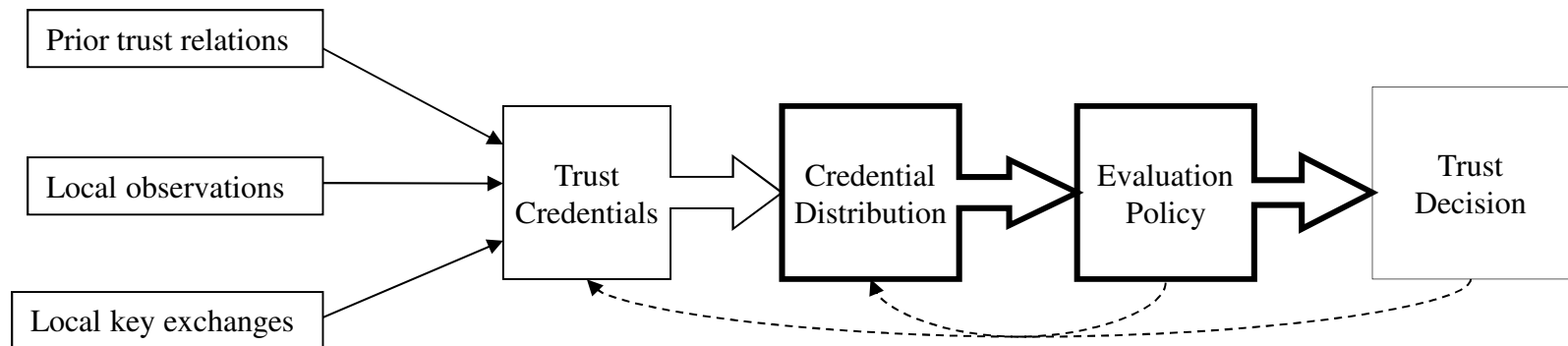
Composite Trust in Networked CPS



Interacting Information and Communication Graphs

- Trust components are derived from different network environments
 - Represented by several metrics and/or parameters
 - Numerous couplings in various notions of trust
 - Numerical trust evaluation is not sufficient, may include constraints on trust policy, etc.
- Three networks we consider
 - Trust in control/decision networks
 - Trust in information networks
 - Trust in communication networks

Components of Trust Management System



- **Trust credentials**
 - From different network domains of various representation forms
- **Credential distribution**
 - Require a negotiation process that includes identifying required credentials, locating credentials and formatting credentials in the way that can be securely transmitted across domains
- **Evaluation policy**
 - More than just numerical trust evaluation

Trust Metrics



- **Multiple trust metrics**
 - Represented as vector weights on links and nodes
 - Multi-value directed graphs with weighted nodes
- **Numerical representation of trust weights**
 - Discrete numerical value: e.g. {full, marginal, untrustworthy, don't know} in PGP trust model
 - Continuous value: e.g. $[0, 1]$
 - Vector value: e.g. $[0, 1]^2$
 - Probability: subjective or objective probability
 - Entropy in information theory

Trust Metrics (cont.)



- **More than just numerical metrics, such as**
 - Level of the confidence
 - Trustor's inclination to take risks
 - Location of trustor and trustee in the network
 - Trustor's access to information
 - Degree of tolerance of potential disappointment
- **More abstract mathematical model**
 - Include logical variables in the form of constraints
 - Allow rule-based/behavioral models and various constraints

Evaluation of Composite Trust



- A general algebraic structure is required for the extended value directed graph
- **Partially ordered semiring**
 - A semiring is a tuple $\langle A, +, \times, 0, 1 \rangle$
 - Partially ordered semiring: $+$ and \times are weakly monotonic
 - Similar to Network Calculus (Max $+$ algebra, LeBoudec & Thiran, 2004)
 - Handle constraints and constrained based reasoning

- **Constraint Satisfaction Problem (CSP)**
 - A set of problem variables
 - A domain of possible values for each variable
 - A set of constraints
 - Specify acceptable combinations of values for the problem variables
- **Hard constraint vs. soft constraint**
 - Hard constraint must be satisfied
 - Soft constraints express preferences or prioritized constraints
 - Soft constraints is common in the distributed environment with multiple domains, such as networked MAS
- **Semiring-based CSP is proposed to address CSP with soft constraints**

Semiring-Based CSP



- A **semiring-based constraint system** is a tuple $\langle S, D, V \rangle$
 - S is a semiring
 - D is a finite set
 - V is an ordered set of variables
- A **constraint** over such a system is a tuple $\langle \text{def}, \text{con} \rangle$
 - $\text{con} \subseteq V$ is known as the type of the constraint
 - **def**: $D^k \rightarrow A$ (where k is cardinality of V) is the value of the constraint.
 - **def** assigns a value as strength of preference each combination of the variables on con
- The solution of an SCSP is a tuple $\langle C, v \rangle$ where $v \subseteq V$ and C is a set of constraints
 - NP-complete problem
 - Algorithms to yield approximate solutions efficiently for certain SCSP

Trust Semiring



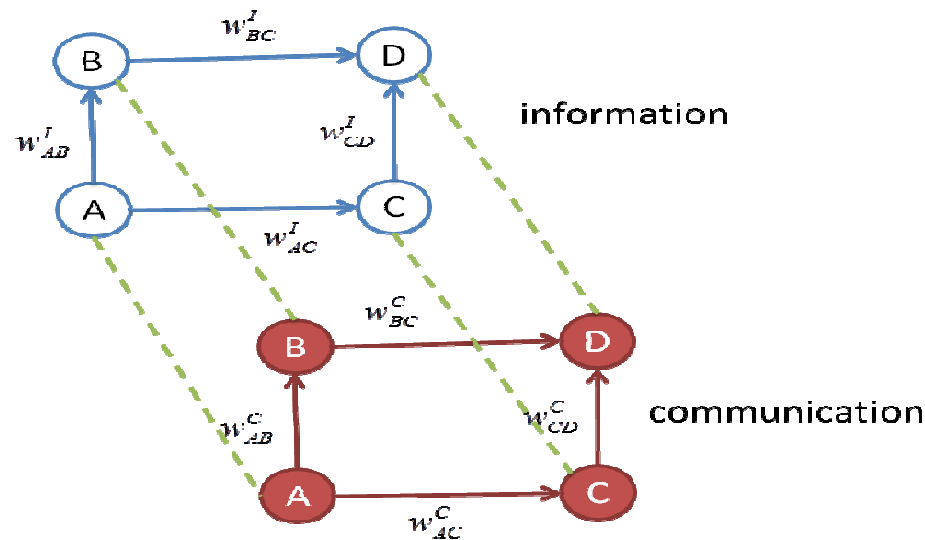
- **Trust Semiring** $\langle W, +_{\text{trust}}, \times_{\text{trust}}, 0, 1 \rangle$
- Based on intuitive concepts about trust evaluation in a network, the binary operators have the following properties
 - Trust **deteriorates along a path**:

$$a \times_{\text{trust}} b \leq_S a, b$$
 - Trust quality **improves via aggregation across paths**, since we have multiple opinions:

$$a +_{\text{trust}} b \geq_S a, b$$
 - The 0 element (identity element for $+_{\text{trust}}$, absorbing for \times_{trust}) corresponds to the opinion “I don’t know” (not the most negative opinion).
 - The element 1 (identity element for \times_{trust}) is the “best” trust weight that can be assigned to a node.
- One example of the trust semiring is
 $\text{TS} = \langle [0, 1], \max, \min, 0, 1 \rangle$.

Example

- Graphs at two-level with trust weights



- **Information semiring** is $\langle W^I, \max, \min, 0, 1 \rangle$
- **Communication semiring** is $\langle W^C, \max, \min, 0, 1 \rangle$
- Trust semiring is $TS = \langle W^I \times W^C, +_{\text{trust}}, \times_{\text{trust}}, 0, 1 \rangle$

Example (cont.)

- Two different set of constraint preferences

- *Information preferred*

$$(w_1^I, w_1^C) +_{trust} (w_2^I, w_2^C) = \begin{cases} (w_1^I, w_1^C) & \text{if } w_1^I > w_2^I \\ (w_2^I, w_2^C) & \text{if } w_1^I < w_2^I \\ (w_1^I, \max(w_1^C, w_2^C)) & \text{if } w_1^I = w_2^I \end{cases}$$

$$(w_1^I, w_1^C) \times_{trust} (w_2^I, w_2^C) = (\min(w_1^I, w_2^I), \min(w_1^C, w_2^C))$$

- *Communication preferred*

$$(w_1^I, w_1^C) +_{trust} (w_2^I, w_2^C) = \begin{cases} (w_1^I, w_1^C) & \text{if } w_1^C > w_2^C \\ (w_2^I, w_2^C) & \text{if } w_1^C < w_2^C \\ (\max(w_1^I, w_2^I), w_1^C) & \text{if } w_1^C = w_2^C \end{cases}$$

$$(w_1^I, w_1^C) \times_{trust} (w_2^I, w_2^C) = (\min(w_1^I, w_2^I), \min(w_1^C, w_2^C))$$

Example (cont.)



- This specific trust SCSP has a **distributed solution** where the following algorithm is carried out at every node in the network

Algorithm: The distributed solution to solve the SCSP.

Repeat

$$X_k^{n+1}(D) = \sum_{l \in \mathcal{N}_k} w_{kl} \times_{trust} X_l^n(D)$$

Until $X_k^n(D)$ converges.

- $X_l^n(D)$ represents the evaluated trust to target D via a chain of n direct trust relations
- $\sum = \times_{trust}$

Future Research



- Scale to large networks, including social networks
- Effects of communication topology on trust integrity and resiliency
- Composite trust dynamics
- Applications to automotive, aerospace CPS

Future “Smart” Homes, Cities, Factories, Infrastructures, Transportation

- UI for “Everything”
 - Devices with Computing Capabilities & Interfaces
- Network Communication
 - Devices Connected to Home Network
- Media: Physical to Digital
 - MP3, Netflix, Kindle eBooks, Flickr Photos
- Smart Phones
 - Universal Controller in a Smart Home
- Smart Meters & Grids
 - Demand/Response System for “Power Grid”
- Wireless Medical Devices
 - Portable & Wireless for Real-Time Monitoring



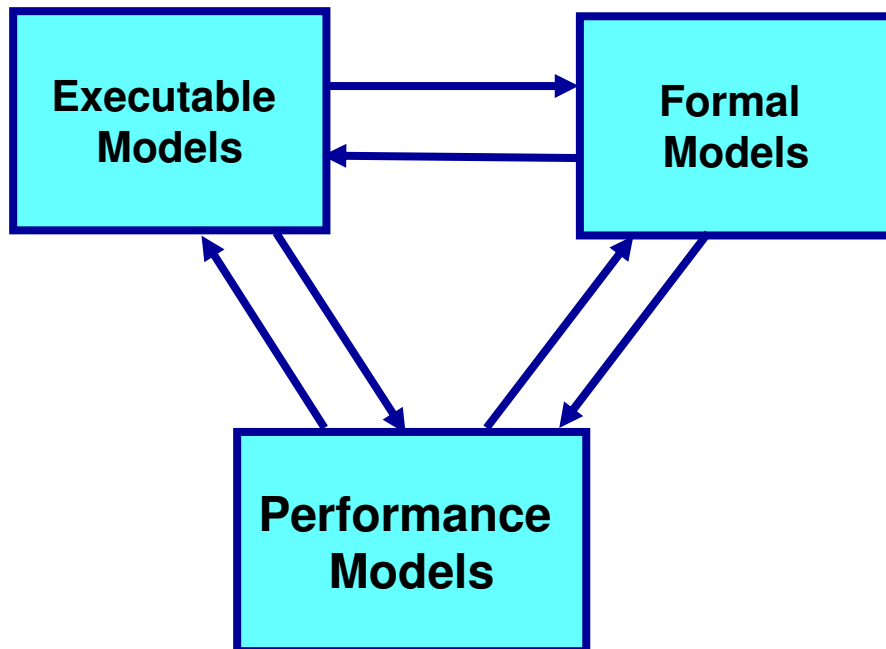
Component Based Networking and Security



- **Component Based Networking** : Leads to a compositional approach to the synthesis and operation of networks. Fits very well MANET, WNAN (and WAND), beyond.
- **Does away with classical layers and with classical cross-layer**
- **Compositionality**, and **Compositional Synthesis**
- **Cross linked executable, formal and performance models is addressing this challenging problem directly.**

*Interacting Control, Information and Communication
Graphs*

Component-base Networks and Compositional Security



**Studying compositionality
is necessary!
Compositional Security is
critical for all CPS!**

Universally Composable Security of Network Protocols:

- Network with many agents running autonomously.
- Agents execute in mostly asynchronous manner, concurrently several protocols many times. Protocols may or may not be jointly designed, may or may not be all secure or secure to same degree.

Key question addressed :

- Under what conditions can the composition of these protocols be provably secure?
- Investigate time and resource requirements for achieving this

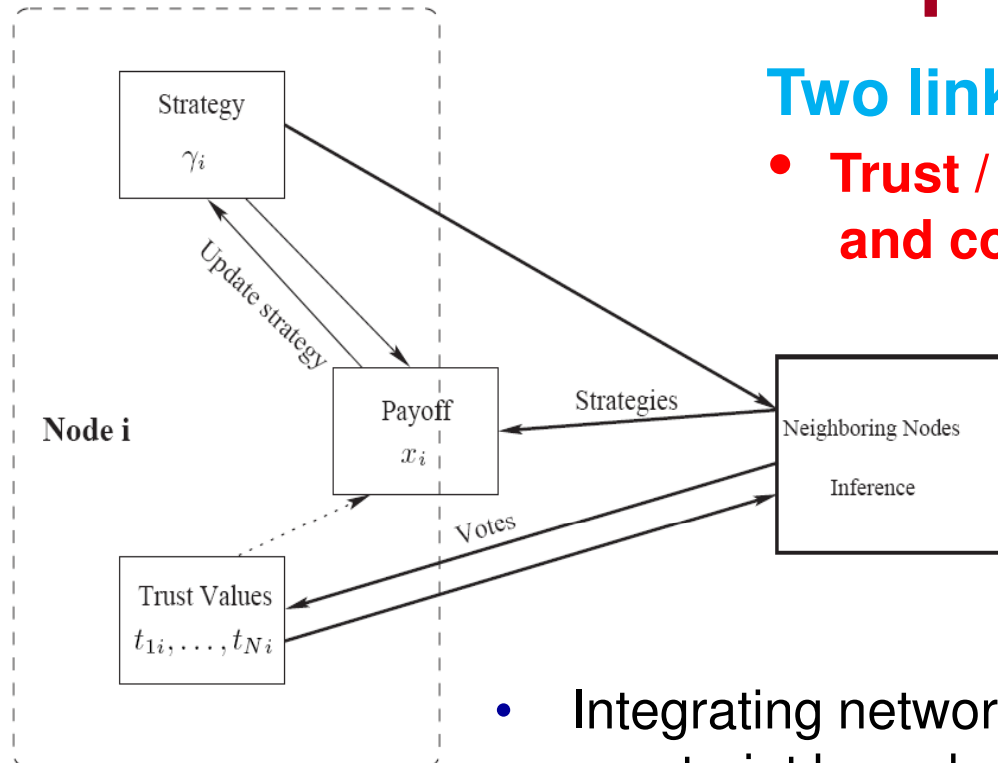
Universally Composable Security (UCS)



Results todate (Canetti, Lindell, ...) :

- When there is a clear majority of well behaving nodes (i.e.2/3) **almost any functionality is secure under UCS**
- When there is no clear majority then UCS is **impossible** to achieve unless there are pre-conditions – typically some sort of trust mechanism
- Introducing **special structure in the network** (e.g. overlay structure, small subset of absolutely trusted nodes) helps substantially in establishing UCS, even without preconditions
- **Many applications:** military networks, health care networks, sensor networks, SCADA and energy cyber networks
- **The challenge and the hope:** Use “tamper proof hardware” (physical layer schemes, TPM etc.) even on a small subset of nodes to provably (validation) establish UCS – role of fingerprints and physical layer techniques.
- **Establish it and demonstrate it?**

Trust and Collaborative Control/Operation



Two linked dynamics

- **Trust / Reputation propagation and collaborative control evolution**

$$\begin{aligned}\gamma_i(t+1) &= f^i(x_i(t), \gamma_i(t), \gamma_j(t), t_{ij}(t)) \\ t_{ik}(t) &= g^i(t_{ij}(t), v_{jk}(t)) \quad \forall k \in N \\ x_i(t) &= h^i(\gamma_i(t), \gamma_j(t)) \\ v_{ij}(t) &= p^i(\gamma_j(t), t_{ji}(t))\end{aligned}$$

- Integrating network utility maximization (NUM) with constraint based reasoning and coalitional games
- Beyond linear algebra and weights, semirings of constraints, constraint programming, soft constraints semirings, policies, agents
- Learning on graphs and network dynamic games: behavior, adversaries
- Adversarial models, attacks, constrained shortest paths, ...

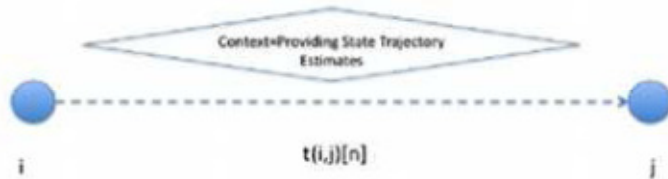
Interacting Control, Information and Communication Graphs

Example: MANET Trust Aware Routing – Trust/Reputation Systems

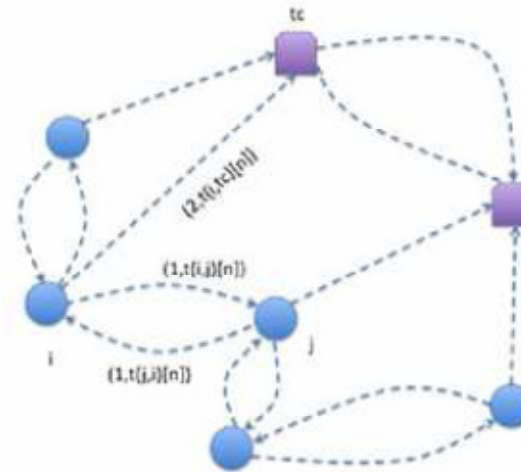


- Components
 - Monitoring and Detection Modules
 - Reputation Control System
 - Response Modules
- **Our approach is different:** build and use a **Trusted Sentinel Sub-Network (SSN)**, which is responsible for monitoring and flooding reputation measures
- Logically decouple the Trust/Reputation System from other network functionalities
- Demonstrate that **logical constraints on the SSN translate to constraints on the communication graph topology** of the network
- **Trade-off analysis between security and performance**

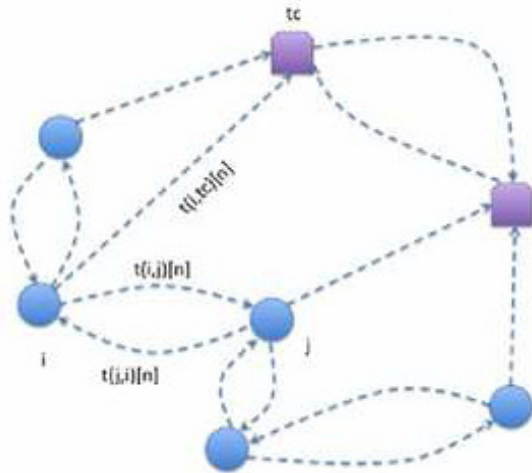
Trust and Induced Graphs



Trust relation



Induced Graph $G(V, A)$



Weighted Directed Dynamic Trust Graph $G_t(V, A_t)$

$$V_{tc} \subset V$$

$$w(i, j) = (c(i, j), t(i, j)[n])$$

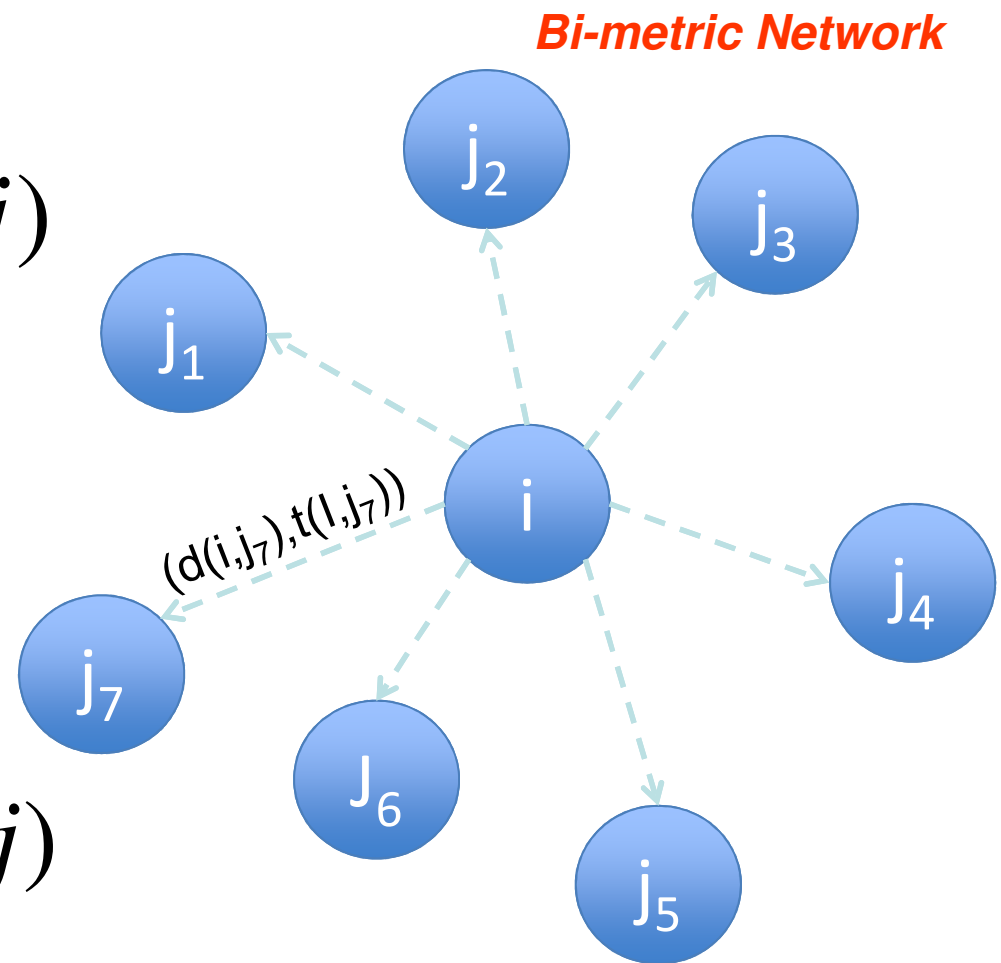
Trust Aware Routing – Multi-Criteria Optimization Problem

- **Delay** of a path “p”

$$d(p) = \sum_{(i,j) \in p} d(i,j)$$

- **Trust** of a path “p” –
bottleneck trust

$$t(p) = \min_{(i,j) \in p} t(i,j)$$

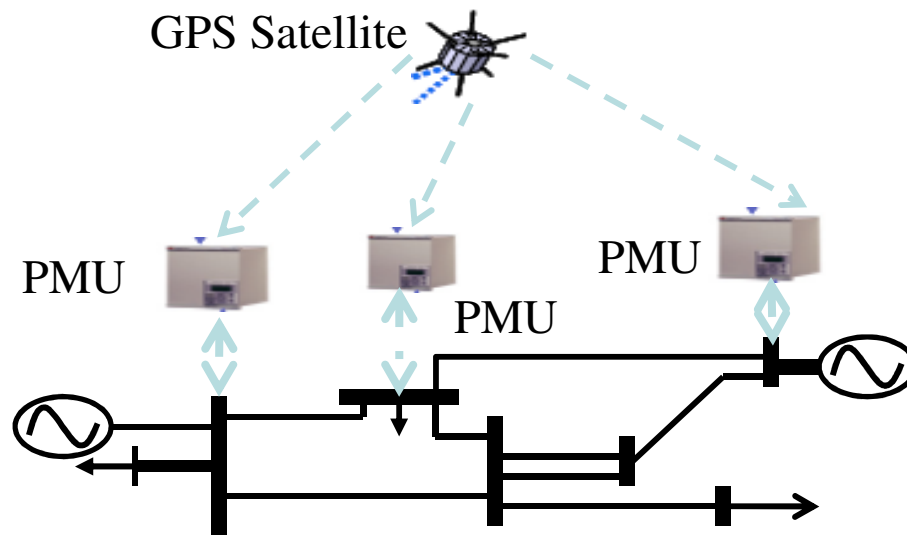


Example: Power Grid Cyber- security



- **Inter-area oscillations (modes)**
 - Associated with large inter-connected power networks between clusters of generators
 - Critical in system **stability**
 - Requiring **on-line** observation and control
- Automatic estimation of modes
 - Using currents, voltages and angle differences measured by PMUs (Power Management Units) that are distributed throughout the power system

Distributed Estimation



N multiple recording sites (PMUs) to measure the output signals

- To compute an accurate estimate of the state $x(k)$, using:
 - **local measurements** $y_j(k)$;
 - information received from the PMUs in its **communication neighborhood**;
 - confidence in the information received from other PMUs provided by the **trust model**

Thank you!

baras@umd.edu

301-405-6606

<http://www.isr.umd.edu/~baras>

Questions?