Networked S-CPS: Ubiquitous Presence
Sensor Networks Everywhere

Wireless Sensor Networks (WSN) for infrastructure monitoring

- Environmental systems
- Structural health
- Construction projects
- Energy usage
Smart Grids in a Network Immersed World

Generation
- Conventional: Coal, Nuclear, Oil / Gas, Hydro
- Renewable: Solar, Wind

Transmission
- Smart Grid

Distribution
- Substation
  - Econometric models
  - Standards for process equipment energy

Utilization
- Residential/Commercial
  - Low-cost “embedded” energy sensors
  - Communications
  - Integrated control & energy mgmt.

• ACEEE estimates +2x energy savings
• Able to measure and manage carbon footprint per product line
Connected Cars: Internal

71 Sensors and 98 Switches

Color Key:
- Low Speed Sensor
- High Speed Sensor
- Safety Sensor

Engine
Connected Cars: External
**Key Challenge:** Humans

We are developing **novel frameworks** to include humans in this collaborative networked CPS environment
A Network Immersed World: Swarms and the Cloud
Online social network services (SNS)
- Permeate our lives with tremendous popularity
- Decision making via combining information from different sources
- Benefits SNS-based applications
  - Recommender Systems
  - Online Ad targeting

Trust relationships in SNS
- People put different levels of trust on others in SNS
- Important in decision making
  - People tend to accept suggestions from those they trust more

Our work: Semiring-Based Trust Evaluation for Information Fusion in Social Network Services
MBSE based HCMS for Diabetes II and its functional connectivity
Framework for MBSE: Key Challenges Addressed

• Methodology to develop integrated modeling hubs (IMH) for CPS – multi-physics and cyber
• Methodology to link IMHs with design space exploration via multi-criteria tradeoff methods and tools
• Linkage to component databases
• Working on the last remaining challenge: requirements management
• Developed new methods and tools to handle complexity in design space exploration
Integrated System Synthesis Tools - & Environments missing

Iterate to Find a Feasible Solution / Change as needed

Change structure/behavior model as needed

Assess Available Information

Define Requirements Effectiveness Measures

Create Behavior Model

Map behavior onto structure Allocate Requirements

Specifications Perform TradeOff Analysis

Create Sequential build & Test Plan

Generate derivative requirements metrics

Model - Based Information - Centric Abstractions

Integrated Multiple Views is Hard!

Model- Based Systems Engineering Components -- Architecture

Model- based
UML - SysML - GME - eMFLON
Rapsody
UPPAAL
Artist Tools
MATLAB, MAPLE
Modelica / Dymola
DOORS, etc
CONSOL-OPTCAD
CPLEX, ILOG SOLVER,
SIEMENS, PLM, NX, TEAM CENTER
Model Integration Challenge: Physics

Physical components are involved in multiple physical interactions (multi-physics)
Challenge: How to compose multi-models for heterogeneous physical components
Using System Architecture Model as an Integration Framework

- Security and Trust Models and Analytics
- Human Behavior Models
- Analysis Models
- Verification Models
- System Architecture Model
- Req’ts Allocation & Design Integration
- Market Models and Analytics
- Software Models
- Hardware Models
- Cost Models Financial Analytics
- Requirements Repository

\[ U(s) \rightarrow G(s) \rightarrow \int \]
The Challenge & Need:
Develop scalable holistic methods, models and tools for enterprise level system engineering

- Multi-domain Model Integration via System Architecture Model (SysML)
- System Modeling Transformations

BENEFITS
• Broader Exploration of the design space
• Modularity, re-use
• Increased flexibility, adaptability, agility
• Engineering tools allowing conceptual design, leading to full product models and easy modifications
• Automated validation/verification

APPLICATIONS
• Avionics
• Automotive
• Robotics
• Smart Buildings
• Power Grid
• Health care
• Telecomm and WSN
• Smart PDAs
• Smart Manufacturing

“Master System Model”
Update System Model
Tradeoff parameters
ADD & INTEGRATE
ILOG SOLVER, CPLEX, CONSOL-OPTCAD
Multiple domain modeling tools
• Tradeoff Tools (MCO & CP)
• Validation / Verification Tools
• Databases and Libraries of annotated component models from all disciplines
Digital Manufacturing Design Innovation Institute (DMDII)

• Announced February 25, 2014, 2014 by President Obama
  
  
• Headquartered in Chicago, Illinois
• Academic-Industry-Government “Mega Project” $320M co-funding, 5 years
• **Goal**: Revitalize manufacturing along the lines described in this lecture
• “Infinite number of virtual factories and an open-source manufacturing platform”
Crowdsourcing Manufacturing

- **Google’s Project ARA**: Smartphones are composed of modules (of the owner’s choice) assembled into metal frames.

- **Ubundu Edge Project**: Crowdsourcing the most radical smartphone yet “Why not look for the best upcoming tech and throw it together to stay ahead of the competition?”

- **Crowdsourcing** the development and manufacturing of **small unmanned aerial vehicles**.
“Democratizing” Manufacturing

• **Goal**: Transforming more ordinary people to “makers” of products and services

• Helping small and medium size companies to manufacture products and services – **bridge the “gap”** from innovation, prototyping, to manufacturing

  • General Electric (GE) opens manufacturing fab lab to spark ideas and participation in manufacturing through making

  • Several companies have also opened up similar “open” labs: Ford etc.

  • Several regional manufacturing centers (industry-university-government) are being established in various regions of USA

• “Industrial Internet” (USA) and “Industrie 4.0” (GE-EU) arrive
MBSE for Sensor Networks

- System Specifications
- WSNs Model Libraries (SysML)
- Environment Models (Simulink/Modelica)
- Control System Models (Simulink/Modelica)

Model Integration (BDD/IBD/Parametric Diagrams)

- <<StructuredSimulink Block>>
- SFunction/Simulink Model Generation
- SFunction and Simulink Model
- Matlab/Simulink Simulations

- Optimization Tools or Parametric Diagram Solvers

C/C++ Codes Generation

- Panel Diagrams
- Interactive Simulations
- Statecharts Animations

C/C++ Source Codes
CPS Architecture: Buildings

Architecture for earthquake resistance
Add computer controlled sensors, shock absorbers, material properties
CPS architecture?

Architecture for energy efficiency
Add computer controlled sensing, HVAC, etc.
CPS architecture?

Pearl River Tower Complex, Guangzhou
Smart Grid – Microgrids Architecture

Grid 1.0
Legacy Grid

Grid 2.0
Smart Grid

Grid 3.0
Future Grid
**CPS Architecture:**
Materials-Geometry-Controls

The 787 Dreamliner delivers:
- 20%* reduction in fuel and CO₂
- 28% below 2008 industry limits for NOx
- 60%* smaller noise footprint

*Relative to the 767

Architecture Logics, their Representation and Integration

**Composite wing** – new control algorithms
**All-electric platform** – new aircraft VMS

**Smart suit** – improve physical endurance & energy harvesting
Collaborative Autonomy
Approach: Four Pillars

The cognitive dialogue – a new architecture and formalism for cognitive systems
A dynamic attention mechanism that works through a combination of signal processing and symbolic processing of prior knowledge
The manipulation grammar and its associated parser
A three-layer architecture involving dynamically interacting multi-graphs and heterogeneous internal world models
Key problems:
Robots must make sense of cluttered audio-visual environments to execute autonomously and collaboratively tasks
Find and identify objects, tools, actions, based on multi-sensory input and prior knowledge
Represent and store prior knowledge
Search scene and knowledge in an efficient and organized manner
Humans utilize an elaborate attention system – need something similar, multimodal and adaptive
Need to learn, reason and communicate about objects, tools, actions

Key principle of our approach: Task-driven integration of perception, control and language
Also essential for human-robot collaboration
Focus: Manipulation Actions

Manipulation actions
Example 2: Robot Learning Manipulation Action Plans by “Watching” on-line Videos
Example 1: Learning Hand Movements from Markerless Demonstrations for Humanoid Tasks

- What tasks?
- How to learn tasks?
- Different situations?
Fig. 1. Overview of learning hand movements for humanoid tasks. Placing task is an example shown here, and the drawing on hand in demonstration video is indicating hand tracker.
From Logic/Semantics, to Timed Automata, to Action Execution

Must link:
- Abstract logical/semantic description of task
- Timed automata representation of actions in a composable manner
- Taking into consideration own kinematics constraints (embodiment)

Can this be done in a principled, automatic, repeatable, verifiable manner?

Challenges: Role and form of learning, fast execution, role of task description and performance metrics, tolerance and uncertainty models
Programming Language for Human Action

A motivating example

(a)

for o in in(bowl) {
    move(o, on(cutting_board))
    cut(o, knife, 2)
}
// Objects have changed because of cut!
for o in on(cutting_board) {
    move(o, in(bowl))
}

(b)

A simple action and its source code
Motion Planning with Temporal Constraints

• Q: How to generate trajectory/path based on temporal specifications such as ordering between actions, repetition of tasks, safety of the motions?

• State of art: motion planning with temporal constraints without duration, such as Linear Temporal Logic (LTL).

• We have proposed two methods for timed temporal logics, such as Metric Temporal Logic (MTL) for motion planning problem:
  – An optimization based method
  – A timed-automata based method

Always visiting area a,b,c and stay there for at least 2s. Always avoiding obstacles
Given: A dynamic workspace (environment),
    A time constrained task (φ),
    A cost function.

Objective: Find the suitable control input such that the robot completes the given task and minimizes the cost function.

Constraints: Avoiding collisions with all static and moving obstacles in the workspace.

Challenges/Innovations: metric temporal logic, finite automata specs, uncertainties with mixed logical/numerical representations, automatic verification, bridge the gap between action grammars and motion planning/controls independent of learning environment and platform execution.
Learning to Plan Manipulation Task Execution in New Environments

- Learn manipulator trajectory from demonstrations

- Adapt manipulator trajectory through planning with new constraints

- Learn preferences to adapt movement through feedback
Proposed System

Task Specification

Demonstrations

Embodiment

Perception

Movement Imitation

Imitation Trajectory

Movement Adaptation (Planning)

Adapted Trajectory

Robot System

Updated Weights

Rewards Learning

Robot Execution

User Feedback
Safety & Trust in Human-Robot Teams: Integrating Logic and Set-valued Analytics

- Space-time reachability analysis (now real time)

Translate these to analytics: model checking, contracts, theorem proving, set valued -- Trust values? Metrics? Timed Languages?
- Roles? Role-based trust management?
We propose to use reachable set for collision avoidance

- Reachable set of a dynamics is defined as set of states reachable from a bounded initial set, a control set and a disturbance set.

- The control sets are then synthesized collaboratively so that the reachable sets of the UAVs have no intersection.

- Existing studies in reachability literature exam the problem in a game theoretical setup such that other UAVs are treated as adversary. [I. Mitchell etc 2005] Commonly the collision avoidance is collaborative.

- Efficient reachable set computation normally uses convex approximations such as ellipsoids [A. B. Kurzhansk 2000] and polytopes.
We seek a control set design for aircraft A and B such that by using more constrained control sets than their initial ones, collision avoidance between sets are guaranteed.

Decompose the problem to two parts

First we seek a tighter control constraint set for aircraft B such that the reachable set are far away from that of aircraft A but at the same time the control set of B is still large enough.

At the second phase we seek a safe reachable tube for aircraft A so that the reachable tube will be apart from the reachable tube of aircraft B for at least the required separation.
- The top plot shows the reachable set of x y z location at time of collision. The two sets are overlapping.
- The darker colored ones in the center is the inner approximation of reachable set.
- The reachable tube of x, y position only in the bottom plot shows same idea.
If both UAVs have same priority, by tuning the scalarization factor, one can obtain control sets of similar size on the left.

The reachable tube can be then visualized as the right figure.
If one of the UAVs has higher priority, it can have larger control set, so that it maintains more freedom comparing to the other one.

The reachable tube can be then visualized as the one on the right.
Multiple Interacting Dynamic Multigraphs

- Multiple Interacting Multigraphs
  - **Nodes**: agents, individuals, groups
  - **Directed graphs**
  - **Links**: ties, relationships
  - **Weights on links**: value, strength
  - **Weights on nodes**: importance

- **Real-life problems**: Dynamic time varying graphs, relations, weights

- **Effects of connectivity topologies**
- **Taxonomies of multigraphs involved** -- performance
  - **Collaboration multigraph**: who collaborates with whom and when.
  - **Communication multigraph**: who communicates with whom and when

- **Need for different probability models**
- **Future**: Dynamic goal oriented planning, re-planning
Constrained Coalitional Games

• The nodes gain from collaborating
• But collaboration has costs (e.g. communications)
• Trade-off: gain from collaboration vs cost of collaboration

Vector metrics involved typically

Constrained Coalitional Games

• Example 1: Network Formation -- Effects on Topology
• Example 2: Collaborative robotics, communications
• Example 3: Web-based social networks and services
• Example 4: Groups of cancer tumor or virus cells

Future:
Introduce complex behavioral models, multiple-sensory perception, Language development and efficient communications, learning from collaboration, motifs, storing and recalling patterns, multiple internal models, complexity, trust in inference and control, composite trust
The Challenge & Need: DoD Collaborative Autonomous Networked Human-Machine Systems

Heterogeneous, dynamic, multi-scale, rapid technology changes, rapid threat changes

The Institute for Systems Research
University of Maryland

Fig. 1: MBSE process elements

Model-Based Systems Engineering

Fig. 2: Modeling and Analysis Tools Integration via SysML System Architecture Model

ADD & INTEGRATE

- New modeling environments
- Network models and semantics
- Reasoning, Validation and Tradeoff Tools
- Databases and Libraries of component models from all disciplines

BENEFITS

- Reduced cost and fielding time
- Modularity and re-use
- Increased agility in designing, modifying and fielding new systems
Demo 1: Synchronized Flight of Small Unmanned Aerial Systems

Our aircrafts use only basic onboard sensors and cameras, flying without the aid of motion tracking cameras that can be seen in many other experiments.

We follow MBSE (Model Based Systems Engineering) methods to create modular software.
Demo 2: Small Unmanned Aircraft Following a Target

Our aircraft use only basic onboard sensors and cameras, flying \textbf{without the aid} of motion tracking cameras that can be seen in many other experiments.

Our aircraft use a vision-based ROS package for the AR. The Drone aircraft automatically follow specific targets.
Distributed Cooperative Control of UAS
In Crowded Integrated Airspace with Safety

J. S. Baras -- baras@umd.edu

The Challenge & Need:
Cooperative Control Sense and
Avoid Technology for Autonomous
UAS in Dense Environments

Fig.: (a) Boid animation of birds in complex environments; (b) ‘bubles’ of
different shapes from slow to higher velocities; (c) diverse bubbles
navigating obstacles to a goal

APPROACH
- Biologically inspired control
  (swarms, birds)
- Control theoretic analytics
- Efficient and fast computations
- Aerodynamics
- Model predictive control of
  hybrid automata (switched
dynamical systems) including
temporal logic
- Formal safety verification
- Integrated modeling, simulation,
synthesis, operations tool-suite
for collaborating UAS

OUTCOMES / APPLICATIONS
- Dynamic bubble shapes for
  varying safety constraints
- Guaranteed safe
  operation of UAS teams
- Biologically inspired high
  performance collaborative and
  safe control of
  UAS/UAV/UGV
Distributed Coordination of Unmanned Underwater Vehicles (Baras)

**Motivation:**
- Networks of underwater vehicles for sensing, ocean mapping and exploration, surveillance
- Cooperating heterogeneous sensing
- Hybrid acoustic and RF communications – avoid surfacing

**Goals:**
- Adaptability to mission
- New communication schemes that explore idiosyncrasies of underwater channel: multiple dynamic waveguides, trapping waves, ducts, multipath. Use predictive opportunistic comms employing on-line ocean channel predictor
- Distributed dynamic behavior-based control

**Benefits:**
- Dynamic insertion and removal of mission elements during execution
- Sophisticated but energy efficient comms
- Longer collaborative, energy efficient missions

Use acoustic models to reduce comms requirements and increase efficiency
MBSE for Robotic Arms and Grippers

- Transcend areas of application: from space to micro robotics
- Include material selection in design
- Include energy sources, resilience, reliability, cost
- Include validation-verification and testing
- Use integrated SysML and Modelica environment
- Link it to tradeoff tools CPLEX and ILOG Solver
- Demonstrate reuse, traceability, change impact and management
Application to Microrobotics

- Micro-robots design and manufacturing require control algorithm and physical layer (material and geometry) co-design.
- This insect-like robot is modeled in Modelica language using Differential Algebraic Equation.
- We are working on a Model-Based Systems Engineering approach to perform analysis, modeling and tradeoff for robotics and its material and control parameters.

Siemens Tools Utilization

- Design and analysis CAD model at the design phase
- Guide requirement to implementation from CAD design to physical simulation
Modeling

- The particular microrobots we are interested in are small insect-like robots with microfeatures, more specifically with flexible joints.

Real microrobot prototype on the left with Modelica DAE based model virtualized in Dymola on the right. Dymola version has two distinct designs. (a) is the original design provided by D. E. Vogtmann, S. K. Gupta, and S. Bergbreiter [2012].
Sensory Perception and Cognition: Internal Models for Collaborative Autonomy

**REAL WORLD:**
Scenes, real 3D geometry, objects, images, sounds, visual clutter, sound clutter, real gestures, other robots, Humans, real actions, time

*Relations, dependencies, interactions*

**WORLD MODEL:**
Entities, Sensory Data Models, Abstractions, Models, Semantics, Dynamic Models, Time models and semantics, Information Models, Action Models, Hierarchies

*Symbolic relations, Logics, Graphs, Rules and Constraints, Metrics, Validation tolerances*

**CONTROL:**
Planning, Scheduling, Decision making, Task monitoring, Performance evaluation

**MODULES FOR:**
Sensory data processing, Scene analysis, Sensory data fusion, Attention selection, Data to Symbols, Symbols to data

**KNOWLEDGE BASE:**
Object Models, Action Models, Fusion patterns, Cognition Models, Symbolic to Associative links, Spatiotemporal patterns, Metrics, Prediction Models, Languages
CPS Architecture: Perception-Cognition and Co-Robots

The “pressure” of “P” on “C”
The return of analog computation?
Non-von Neumann Architectures?
Physics of computation?
Beyond Turing?

Cognition and knowledge generation from sensory perception – communicating with humans – collaboration
Not just obeying commands – the inverse problem
Future “Smart” Homes and Cities

• 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
Cars are Heavily Computerized: Electronics in Cars and Vulnerabilities

UW/UCSD Work:

Kosher et al., *IEEE Symposium on Security and Privacy, '10*
- Reach CAN bus through diagnostic port

Checkoway et. al., *USENIX Security, '11*
- Remote attacks
- Insert virus into computer system in mechanic shop
- Bluetooth
- Telematics unit
- CD player
Physical Layer Authentication: Key Ideas and Challenges

- Exploit characteristics (a.k.a. FINGERPRINTS) of physical layer (vastly ignored to date)
  - Waveform, RF and hardware peculiarities ⇒ lead to ‘unshakeable’ fingerprints
  - Embed artificial and stealthy ‘fingerprints’
  - **Authenticate the device to the network and then the user to the device** ⇒ reduces attack risk (fewer times through the net)

- Distribute assurance/trust function across software and hardware (increases difficulty to attacker significantly)
  - Trusted computing platform – architecture modifications to allow multiple sources input (including biometrics)
  - TPM – MTM chip ‘add on’ to portable devices and TCN
  - Remote software attestation
Experimental Validation

Demonstrated Very Low Power Authentication is Feasible
Trusted Computing

- **Trusted Platform Module technologies (TPM, MTM, TCN)**
  - A secure hardware
  - Protects the integrity and confidentiality of data with hardware support
  - Performs integrity measurements and reports them, thus attesting for the software running in the device

- **Provides a way to**
  - Understand the state of the platform,
  - Evaluate the state
  - Make a decision if the platform is appropriate for the task

**Source:** TCG Architecture Overview, http://www.trustedcomputinggroup.org
New Ideas: Hardware-Based Security

Using an external TPM?
- Initial idea: Use an existing component-of-the-shelf like a TPM or SmartCard as root-of-trust
  - But...
- Cost, PCB area,
- Quality requirements, availability of suitable components (e.g. temperature range) and
- Sensitivity to valid attacks
  - Reset attack (TPM is reset, manipulated µC continues operation)
  - Data exchange between µC and TPM not protected
Security Integration on the Portable Device

• The TPM/MTM is incorporated in the device

• Biometric information
  – protected in the TPM or
  – stored in the device but encrypted with keys that are managed by the TPM

• Hardened security encourages the use of the device

• Challenges:
  (a) How to use informative time varying pieces of the biometric
  (b) Develop anti-spoofing techniques using the sensor signature
  (c) System integration and validation of the various fingerprints and physical layer techniques
  (d) Proof methods that security is improved – Information theoretic methods
The Challenge & Need:
• Composite trust in distributed sensing and control systems (DSCS)
• Security and Trust-aware DSCS algorithms
• Universal compositional security
• Performance, security, energy, tradeoffs
• Vulnerability analysis and resilient system architectures

Fig. 1: Social (human agent) networks supported by technological networks

Fig. 2: Effects of trust on collaborative distributed control/operations (Baras 2005)

Fig. 3: Linked component-based executable, formal, performance models

APPROACH
• Security and trust aware network utility maximization
• Weighted multi-graphs
• Multiple ordered semi-rings,
• Physical layer security and authentication for universal compositional security
• Network game theory
• Distributed hybrid systems

APPLICATIONS
• Wireless communication and sensor networks
• Safety critical aircraft management systems
• Web-based social nets
• Power grid, smart grid, SCADA
• Smart buildings
• High integrity reputation and recommendation systems
• Resilience and robustness in the presence of adversaries

Compositional Security and Trust in Networked Multi-Agent Systems
J. S. Baras -- baras@umd.edu
Trust Semiring Properties: Partial Order

- Combined along-a-path weight should not increase:
  \[ a \otimes b \leq a, b \]

- Combined across-paths weight should not decrease:
  \[ a \oplus b \geq a, b \]
Computing Indirect Trust

• Path interpretation

\[ t_{i \rightarrow j} = \bigoplus_{\text{path } p:i \rightarrow j} t_{i \rightarrow j}^p \]

• Linear system interpretation

\[ t_{i \rightarrow j} = \bigoplus_{\text{User } k} t_{i \rightarrow k} \bigoplus W_{k \rightarrow j} \]

\[ \vec{t}_n = W \otimes \vec{t}_{n-1} \bigoplus \vec{b} \]

• Treat as a linear system
  – We are looking for its steady state.
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

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.
Consensus with Adversaries

• Solve the problem via detecting adversaries in networks of low connectivity.

• We integrate a trust evaluation mechanism into our consensus algorithm, and propose a two-layer hierarchical framework.
  – Trust is established via headers (aka trusted nodes)
  – The top layer is a super-step running a vectorized consensus algorithm
  – The bottom layer is a sub-step executing our parallel vectorized voting scheme.
  – Information is exchanged between the two layers – they collaborate

• We demonstrate via examples solvable by our approach but not otherwise

• We also derive an upper bound on the number of adversaries that our algorithm can resist in each super-step
Distributed sensor fusion. Goal: all agents reach consensus on ML estimate.


Distributed Coordination. Goal: all agents reach decision on same direction (location)
Malicious agent:

- Multiparty secure computation


Without considering failures, for certain nodes, the consensus problem in distributed control can be solved by simply iteratively calculating weighted averages of nodes’ neighboring states.

– Network of agents modeled by directed graph $G(k) = (V; E(k))$

- $V$ denotes the set of nodes and $E(k)$ the set of edges at time $k$
- $N_i(k) = \{ j \mid e_{ij}(k) \in E(k), j \neq i \}$ set of neighbor nodes of $i$
- “can hear from at time $k$”. $N_i^+(k) = N_i(k) \cup \{i\}$

– Nodes’ states (decisions, beliefs, opinions, etc.) evolve in time according to the dynamics:

$$x_i(k) = \sum_{j \in N_i(k)} w_{ij}(k)x_j(k-1) + w_{ii}(k)x_i(k-1)$$

$$X(k) = \{x_1(k), x_2(k), \ldots, x_N(k)\}^T \text{ $N$-dimensional vector of nodes’ states at time $k$.}$$

$W(k)$ is the updating matrix (weight matrix) at time $k$, rows sum to 1.
Trust-Aware Consensus

Trust Evidence

- Decision rules

Local Trust

- Trust Propagation

Global Trust

- Embed trust into consensus

Trust-Aware Consensus

Graph:

- Nodes 1, 2, 3, 4, 5, 6, 7
- Node 7 is marked as Malicious
- Node 1, 2, 3, 4, 5, 6 are marked as Good

University of Maryland Competition Sensitive
Trust-Aware Consensus

\[ x_i(k) = \frac{1}{A_i(k)} \sum_{j \in N_i} t_{ij}(k)x_j(k - 1) \]

\[ A_i(k) = \sum_{j \in N_i} t_{ij}(k) \]

\( t_{ij}(k) \) is “equilibrium” global trust values
Simulations

Adversary outputs constant message. Figure on the left has no trust propagation. Figure on the right has trust propagation.
Joint Content Delivery and Wireless Network Optimization

Existing System Design

Social Network
- Predict “rewards”.
- “Big Data”: various ML models.
- Slow in training, fast in computing.
- Asynchronous, centralized.

Wireless Network
- Schedule resource for delivery.
- Randomness of channel.
- Time-variant.
- Synchronized, distributed.

Different metrics/utilities:
- Ads: number of views (= ad payout).
- Videos: time spent.
- General: user satisfaction.
Three Scenarios/Problems

• Single base station, time-invariant reward.
  • Basic problem.
  • Establish foundation framework.
• Multiple base stations, time-invariant reward.
  • Many system configurations.
• Single base station, time-variant reward.
  • Time-variance specifically due to social dynamics.
# Comparison

<table>
<thead>
<tr>
<th></th>
<th>Traditional</th>
<th>Joint Optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What to deliver?</strong></td>
<td>Social optimal</td>
<td>Joint optimal</td>
</tr>
<tr>
<td><strong>How to deliver?</strong></td>
<td>Unicast</td>
<td>Multicast</td>
</tr>
<tr>
<td><strong>Fragmentation?</strong></td>
<td>Packet</td>
<td>Content package</td>
</tr>
</tbody>
</table>
Simulation Results – Overall System Rewards

Significant joint optimization gain. Myopic scheduling is sufficiently good. No significant improvement for look-ahead.

Number of contents $M=30$, $N=20$

Number of users

Joint Optimization

Social-only Optimization

$B=25$MHz

$B=15$MHz
Thank you!

baras@isr.umd.edu
301-405-6606
http://www.isr.umd.edu/~baras

Questions?