Coordination in Decentralized Grids

Prof. David Bakken

School of Electrical Engineering and Computer Science

Washington State University

Pullman, Washington, USA

bakken@wsu.edu

WASHINGTON STATE UNIVERSITY

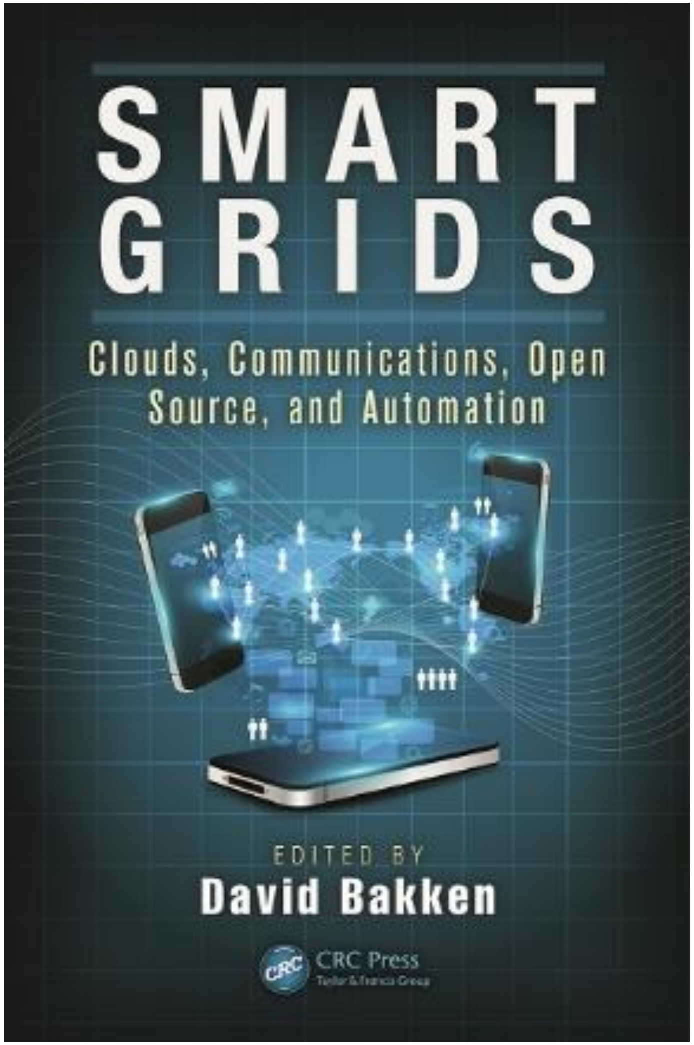
PSERC Webinar

Sept. 3, 2019

GridICT.net

Context

- IANAPP (power person): Computer Scientist
  - Core background: fault-tolerant **distributed computing**
  - Research lab experience (BBN) with wide-area middleware with QoS, resilience, security, …. for DARPA/military
  - Working with Anjan Bose since 1999 on **wide-area** data delivery issues appropriate **for RAS and closed-loop applications**
    - **GridStat (1999-present)**
    - **GridSim (2009-2014)**
    - **GridCloud (2012-2015)**
    - **DCBlocks (2014-present)**
    - **GridFog (2016-present)**



May, 2014 | ISBN: 1482206110

My Theses

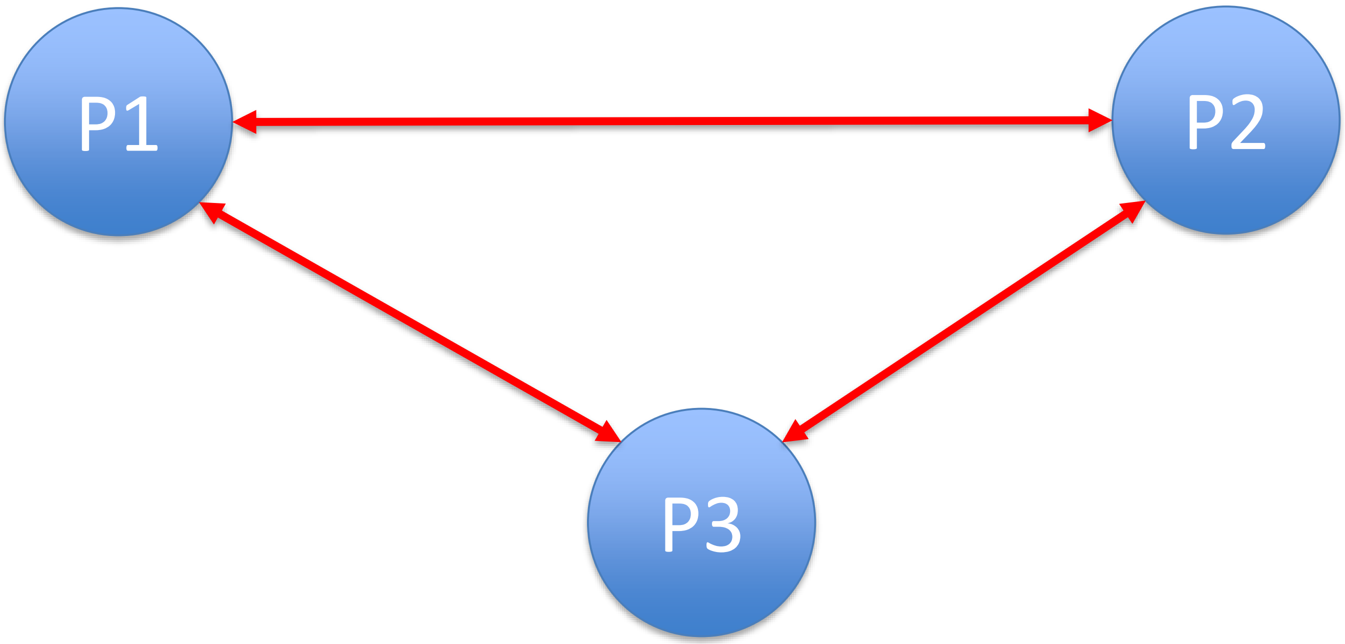
1. **Grids are getting more decentralized**
2. **Distributed coordination (consensus, others) is extremely complicated, even for computer scientists**
3. **Platform support is crucial** for coordination patterns
   - Deal with complexities of distributed computing
   - Reuse of experts' software
   - Future-proofing: future mechanisms
   - E.g., RTI DDS has ~100 QoS-related settings

Outline

- **Distributed Coordination 101**
- DCBlocks Platform Summary
- Coordination in the Electric Grid
- Fog/Edge Computing to the Rescue
- Next-Generation Platform Requirements

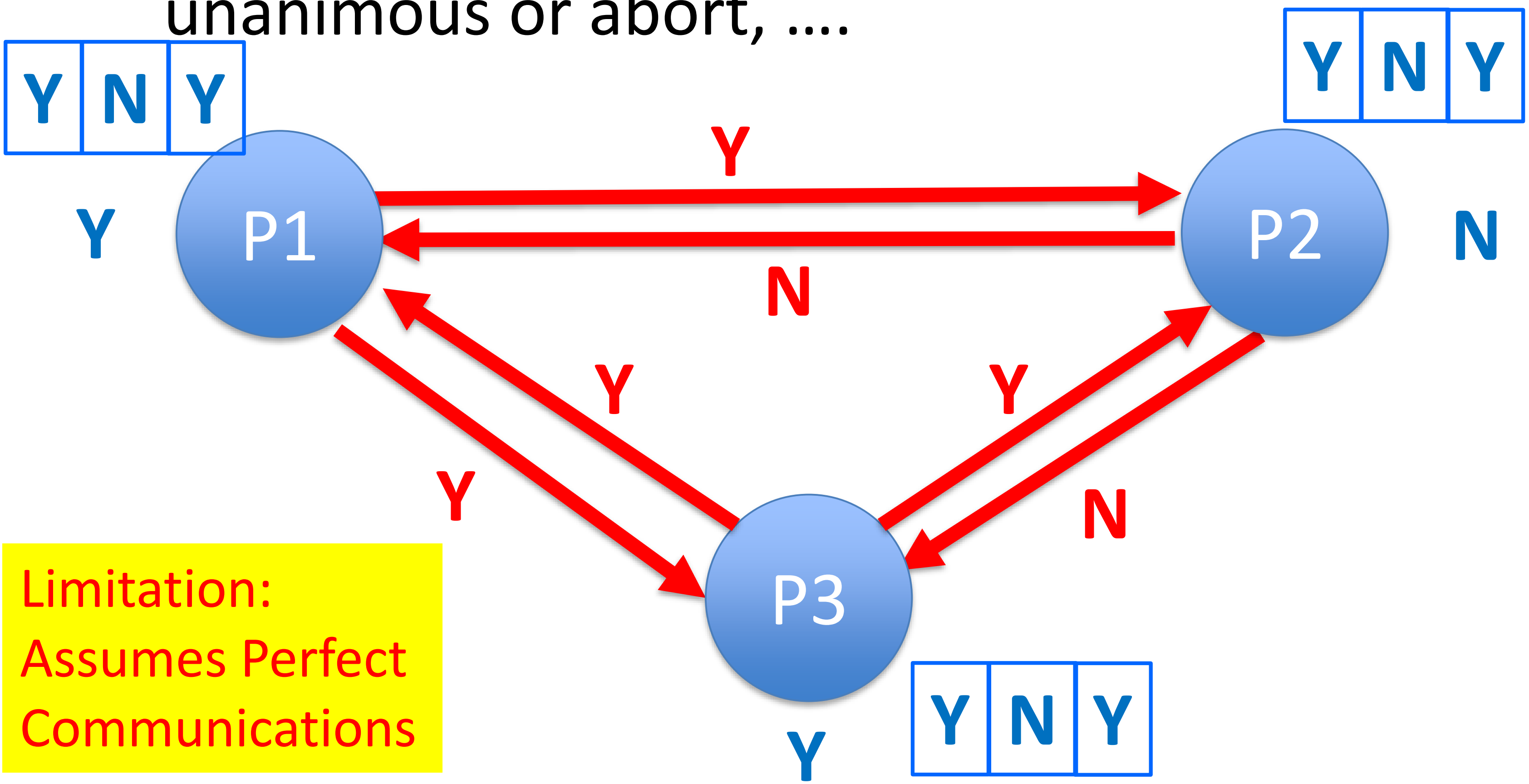
How to agree on a value?

- "Yes" or "No"
- Only based on messages seen locally



Algorithm #0: Naïve

- Each process sends all others their choice/vote
- Combine in some deterministic way: majority, unanimous or abort, ….

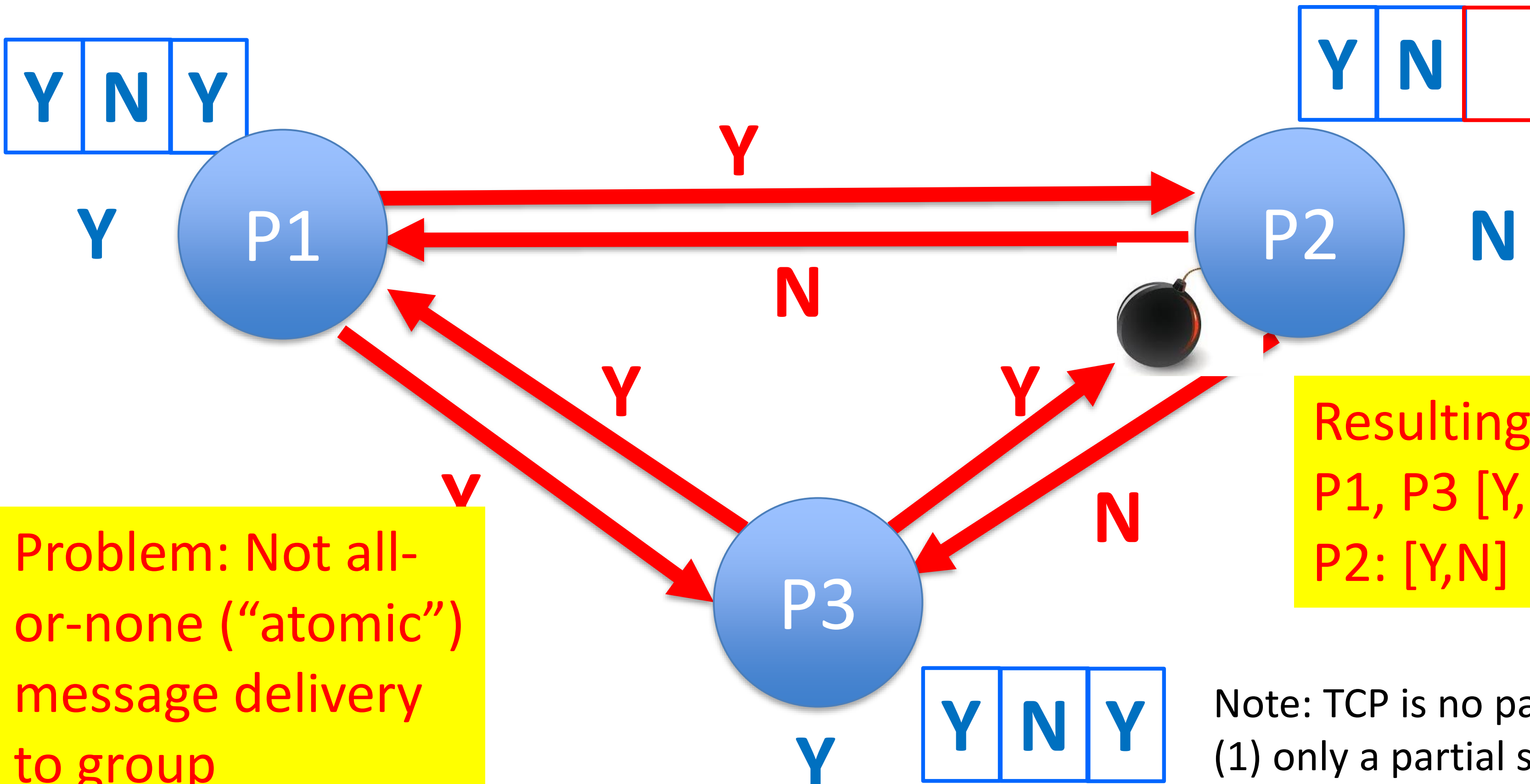


Limitation:
Assumes Perfect
Communications

Distributed Computing is HARD

- Two fundamental facts of life:
  - Variable (computer) network delay
  - Partial failures
- So different cooperating processes can see
  - Different message arrival order (at different processes)
  - Different failures (at any give time)
  - Different timeouts (at any given process)
  - Different group membership
- Very subtle boundary cases for programmers to program!
  - Can result in inconsistent decisions.

Problem #1a for Naïve Algorithm #0



Resulting vectors:
P1, P3 [Y,N,Y]
P2: [Y,N]

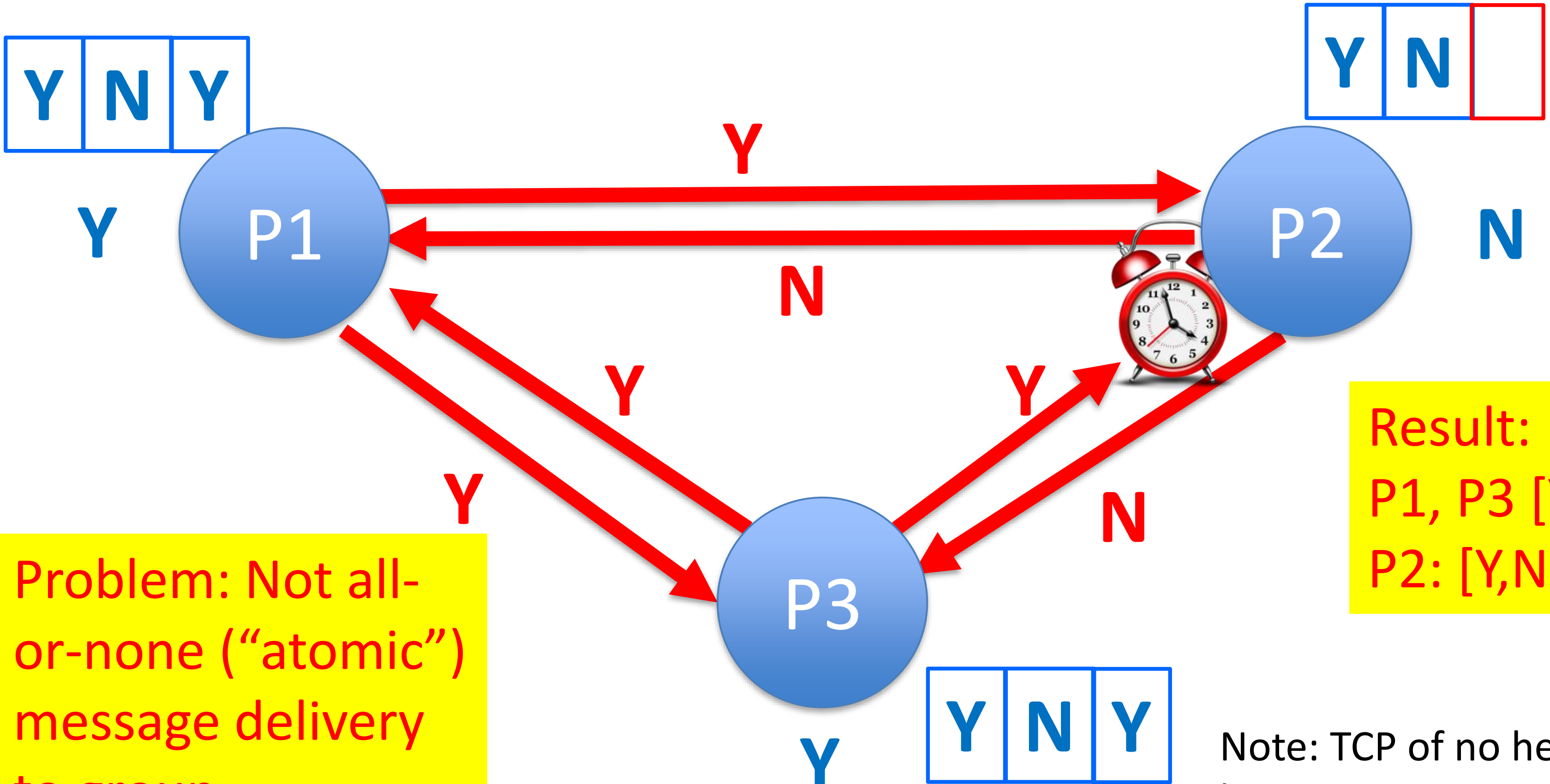
Problem: Not all-or-none ("atomic") message delivery to group

Note: TCP is no panacea
(1) only a partial solution
(2) not generally used due to inefficiency/ delays
(3) can't use with multicast

Problem #1b for Naïve Algorithm #0

Issue: can't wait forever for a message (may never come)

Normal solution: set a timeout, give up if it fires



Problem: Not all-or-none ("atomic") message delivery to group

Result:
P1, P3 [Y,N,Y]
P2: [Y,N]

Note: TCP of no help here: it has to time out at some point

Possible Solution: Failure Detection Service

- Add a failure detection service "off to the side"
- That way you should know how big your vector should be (??)
- Oops: failure notifications ("view changes") can be delivered at different points in the message stream
- Solution: Virtual synchrony (Ken Birman, ~1984)

WASHINGTON STATE UNIVERSITY

gridstat.net

WAY more than Problem 1 can Go Wrong (Redux)

- Two huge facts of life:
  - Variable (computer) network delay
  - Partial failures
- So different cooperating processes can see/detect
  - Different message arrival order
  - Delivery of a message to some, but not all, processes
  - Different failures
  - Different timeouts
  - Different group membership notifications at different logical times
- Very subtle boundary cases for power engineers to program! **Can result in inconsistent decisions**.

Distributed Coordination R&D

- Since 1979!
- Theoretical papers, most algorithms never programmed AFAIK
- Papers very hard for a MSCS student to understand
  - Impossible for a power engineer (or even BSCS IoT application engineer) to deeply understand, or even find
  - Ergo invariably lots of boundary cases being missed across the industry
  - Results in less resilient systems

Complexity of Dist. Coord. Algorithms

- Consensus – Processes agree on one or more values from a set of proposed values.

Name	Failure Model	Computational Complexity	
		Message complexity	Time complexity (Number of rounds)
Scalar Consensus (also Vector, Matrix)	Crash, Omission, Byzantine	N(F + 1)	F + 1
K-set Agreement	Crash, Omission, Byzantine	F/K + 1	F/K + 1
Paxos Concensus	Crash, Omission, Byzantine	(2F+1)(N- 3)	4T
2 Phase Commit	None	2(N – 1)	2
3 Phase Commit	Crash	3(N – 1)	3 to 6

Where,

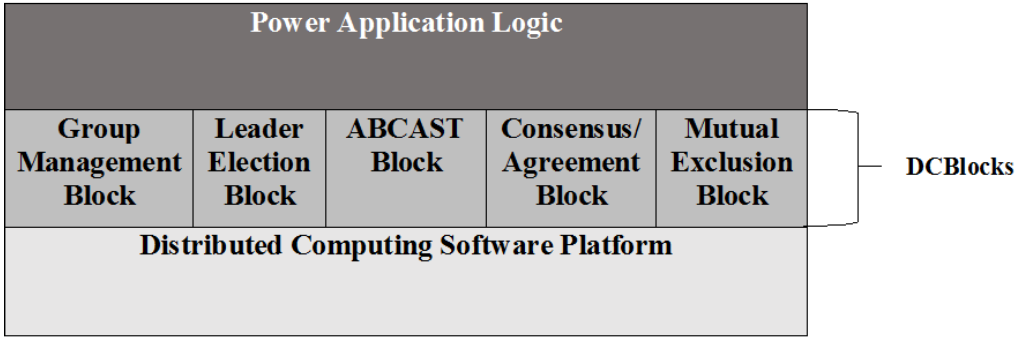
- N = Number of processes
- **F** = Number of faulty processes
- **K** = Max possible decision values in K-set algorithm
- **T** = Message delay

Outline

- Distributed Coordination 101
- **DCBlocks Platform Summary**
- Coordination in the Electric Grid
- Fog/Edge Computing to the Rescue
- Next-Generation Platform Requirements

DCBlocks Platform Summary

- DCBlocks: Decentralized Coordination Blocks
  - “**Building blocks**” for a program: **coordination patterns**
- Packaged up and made useable solutions to the most useful coordination probs (“blocks”); implemented in Akka Java
  - Group management
  - Agreement/consensus
  - Leader Election
  - Voting
  - Reliable multicast
  - Mutual Exclusion



- Version 0: Shwetha Niddodi, Decentralized Coordination Building Blocks (DCBlocks) for Dedentralized Monitoring and Control of Smart Grids, WSU MS Thesis, December 2015.

Outline

- Distributed Coordination 101
- DCBlocks Platform Summary
- **Coordination in the Electric Grid**
- Fog/Edge Computing to the Rescue
- Next-Generation Platform Requirements

NEWS: Renewables Destabilize Grids!

- I wish that even 10% of politicians and regulators understood this!
  - It is not politically correct
  - But it is THE LAW (physics)
- Renewables do not provide any reactive power
  - Can (must) be provided locally by ancillary services with extra hardware
  - This has to be locally coordinated and FAST

Renewables Destabilize Grids! (2)

- Renewables do not provide physical inertia

  - Turbines from hydro, natural gas, and coal have a HUGE amount of rotational inertia
  - Can soak up a LOT of energy from faults
  - Faults travelling farther and faster
    - Ca. 2000: ~100 miles/hour
    - Today: much closer to speed of light

Renewables Destabilize Grids! (3)

- Renewables have first access to the markets (zero marginal costs), at least in Europe
- "Existential threat" to the business models of the companies providing the stable baseline power (*The Economist*)

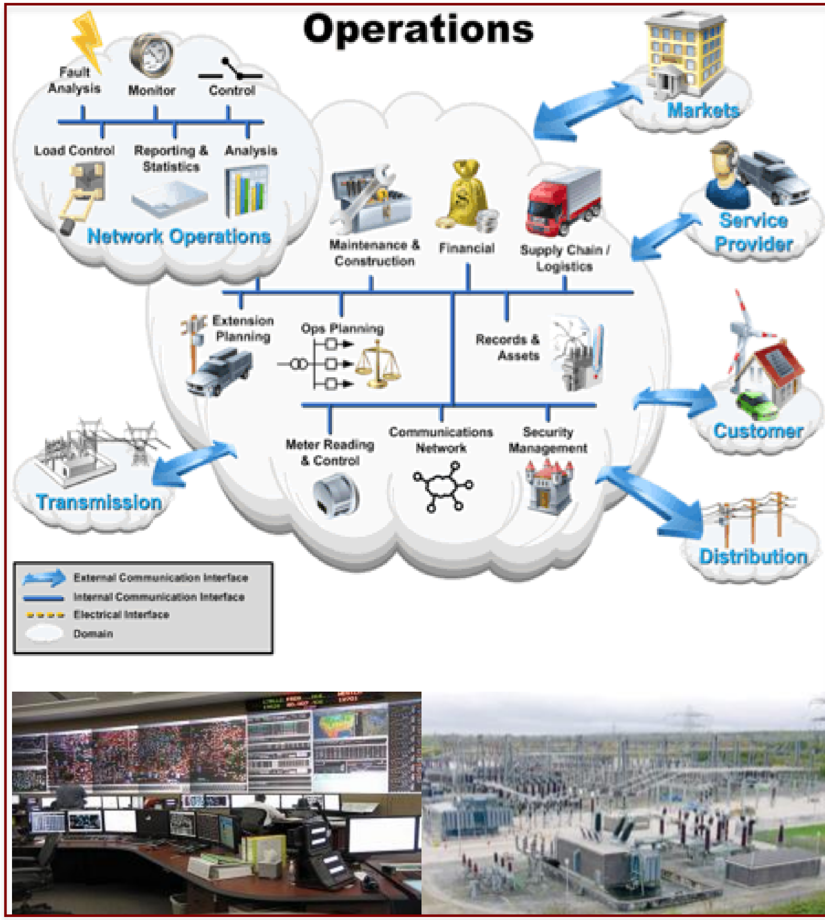
Grid Architecture

- Dominant architecture in power grid: centralized control center (CC)
  - With limited local control: protection, transformer tap changes, reactive power control
- Big changes in the "smart grid" need decentralized apps
  - Renewables
  - Much larger #sensors in field
  - Faster response than round trip to CC
  - Intermittent loads (batteries: charging and DR)
  - CC is single point of failure AND cyber-attack

Challenges Abound!

- Challenges for CC based monitoring and control
  - Large amount of measurement data
  - Large set of system variables
  - Intermittent nature of DERs (e.g. wind farms) and battery operated loads (e.g. EVs)
  - Slow (relatively) control action response
  - (Soon) too many DERs to send to one location
    - Problems in Germany
- Challenges for completely local control
  - Based on local disturbance and limited network visibility
  - Possible cascading effect on the neighboring areas

Vision for Decentralized Apps



Centralized	Local
Slow	Fast
Not Scalable	Non-optimal
Optimal	Hard coded
Prone to failures	May fail for unexpected

Existing Monitoring and Control

Distributed, Coordinated and Hierarchal

- Fast
- Scalable
- Sub-Optimal
- Fault-tolerant
- Supports Big Data
- Supports IoT

Proposed Monitoring and Control

WASHINGTON STATE UNIVERSITY

gridstat.net

Decentralized Power Apps

- Decentralized apps are here!
- "**consensus**" appearing in power papers
  - **Issue**: how to combine one value from each cooperating decentralized process into the same scalar value at each process
  - **Power definition**: math to merge a vector of values into a scalar
    - Assumes perfect communications ☹
  - **Computer science definition**: how to have a group of cooperating processes agree the scalar value, despite
    - Variable network delays
    - Dropped Messages
    - Failures of processes, network links
    - Etc. etc. etc ….
- Make Remedial Action Schemes (RAS) dynamic
  - Now configured on install: inflexible, not adaptive
  - What happens when
    - power topology and operating point changes?
    - Cyber topology changes (node or link failure)?
- Can be hierarchical based on power topology
- Can have dynamic groups based on power entities

Attributes of a Candidate Power Alg.

- Decentralized/decentralizeable decisions
- Decision in one place but may rotate/change (e.g., with ICT failure or changing power or ICT conditions)
- Race conditions possible
  - "Inconsistent" decision possible with *ad hoc* techniques
  - For a RAS scheme, that could be catastrophic

WASHINGTON STATE UNIVERSITY

Decentralized Linear State Estimation

- Traditional Centralized State Estimation may need additional time to converge and may not converge at all.
- DLSE is developed as an alternative solution. The main idea of DLSE is to divide the power system computational data into a set of **<u>groups</u>**, and process in distributed manner to reduce the computational burden.
- The linear state estimation is a noniterative method, and much faster than traditional centralized methods
  - Can also use MUCH more local data:
    - Almost all would come from that substation
    - A little comes from parent group and peer groups

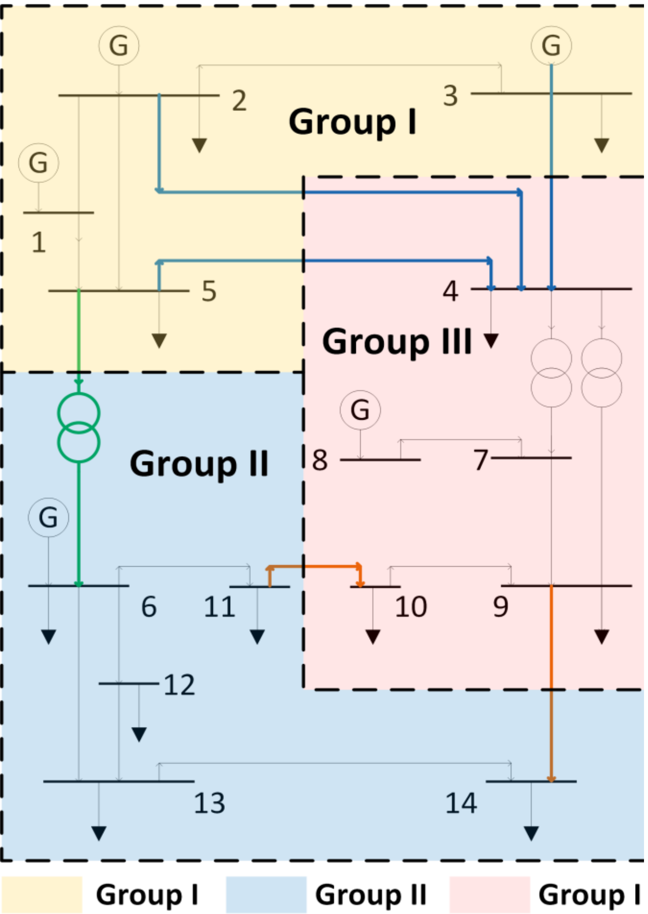
WASHINGTON STATE UNIVERSITY

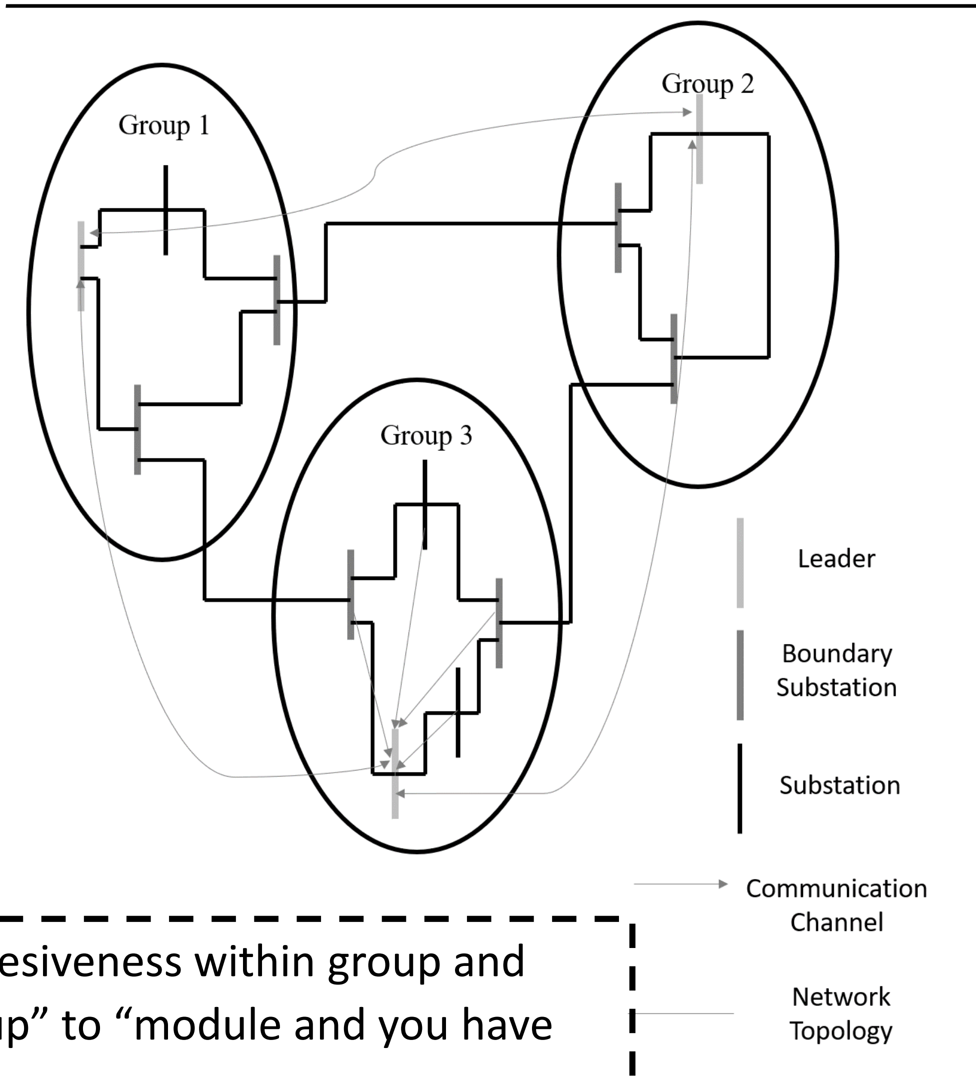
gridstat.net

Forming Groups in Decentralized Linear State Estimation (DLSE)

- Power system divide into groups based on:
  - Electrical Distance
  - Connectivity of Boundary Bus
  - Computational Ability
  - Requirements of Supported Applications
- NOTE: Both Cyber and Physical criteria!
  - Important for DCBlocks++

Example Groups in Decentralized Linear State Estimation





Note: figure on right shows high internal cohesiveness within group and loose coupling between them. Change “group” to “module and you have one old lesson for SW design!

WASHINGTON STATE UNIVERSITY

gridstat.net

Centralized RAS

- Normal power **protection** involves monitoring one power bus (line) at substations at each end
- Remedial Action Scheme (link)
  - Involve 3 or more substations
  - A last-resort “tripwire” to prevent a large blackout
  - AKA “Special Protection Scheme” or “System Integrity Protection Scheme”
- The objective of one centralized RAS algorithm is to reduce wind energy output to prevent overloads on the transmission line.
- The algorithm also aims to minimize the amount of renewable energy that is curtailed.
- This can be mathematically formulated as an optimization problem.

WASHINGTON STATE UNIVERSITY

gridstat.net

Distributed Remedial Action Scheme

- The proposed Distributed RAS logic is designed to be implemented in multiple controllers at electric substations connected to data sources in a decentralized manner.
- In order to achieve the distributed implementation of the RAS, it is required to mathematically distribute the overall objective and constraints among the various computing nodes.
- The actors also monitor the power system states and check whether there is any overload in the transmission line, and calculate the minimum curtailment value in case of overloads

Power Use Cases (So Far…)

Decentralized Power Apps	Applicable DC Blocks	Pub
Distributed Voltage Stability	Group Membership, Leader Election, ABCAST	[LNS+16]
Distributed State Estimation	Group Membership , Leader Election , ABCAST	[LSA+16]
Distributed Remedial Action Schemes	Group Membership , Scalar Consensus, ABCAST	[LSA+16]
Decentralized Wind Power Monitoring and Control	Group Membership , Leader Election , ABCAST	[KLA+17]
Distributed Frequency Control	Group Discovery, Group Membership , Leader Election , Matrix Consensus, ABCAST	[BAS+17]
Decentralized Optimal Power Flow	Group management, Matrix Consensus, ABCAST	[KGL+18]
Decentralized Reactive Power Control	Group management, Vector Consensus	[KLA+17]
Decentralized Inverter Control	Group management	[BAS+17]

Other Notes [Lee17,Gop18]

- Groups can be **dynamic**
- Groups can be **hierarchical**

Outline

- Distributed Coordination 101
- DCBlocks Platform Summary
- Coordination in the Electric Grid
- **Fog/Edge Computing to the Rescue**
- Next-Generation Platform Requirements

Edge and Fog Computing

- **Edge computing** has been around for a while
  - Management of edge devices, including installing/configuring apps, etc
- **Fog Computing** is newer (~2 years of hype)
  - **Goal:** push cloud services to/towards the edges
  - Silicon Valley views this as the way to manage IoT devices (2017 "Fog World Congress").
- Can **support decentralized grids** with its very sophisticated management infrastructures (e.g., Cisco FogDirector)
  - You can buy a Cisco router with a CPU in it to load your apps/services on

IoT is Here

- "Extending Internet connectivity beyond the usual servers/PCs/mobile platforms to traditionally 'dumb' devices and everyday objects"
- It is big and getting bigger (still early in hype cycle)

Year	#connected devices
1990	0.3 million
1999	90 million
2010	5.0 billion
2013	9.0 billion
2025	1.0 trillion

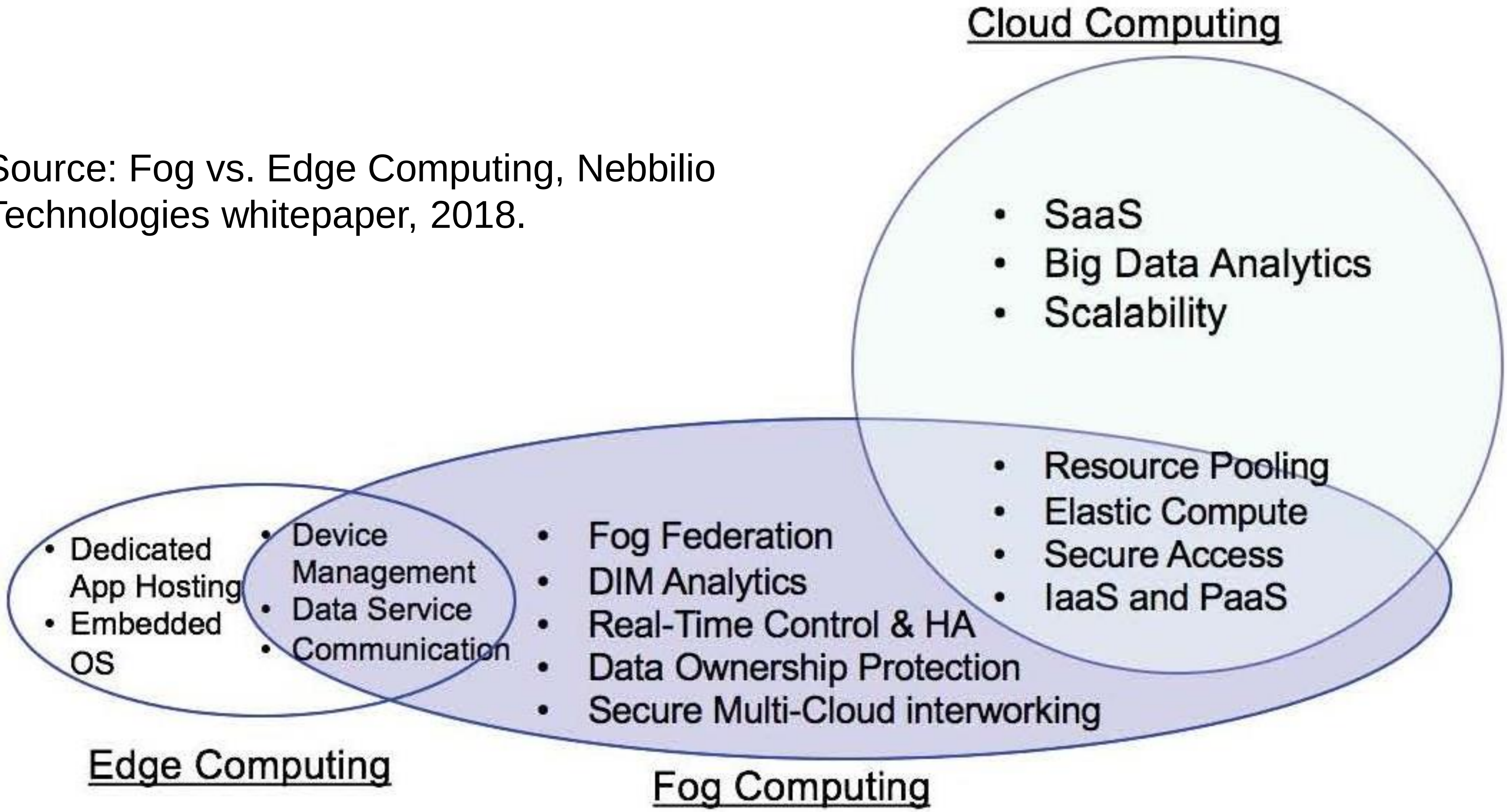
Source: HP

IoT Application Domains

- Smart appliances
- Smart energy meters
- Smart homes
- Elder care
- Manufacturing
- Wearable Devices
- Connected Vehicles
- Smart Healthcare Devices
- Smart Grids
- Smart Cities
- Retail: smart checkout & inventory management
- .....

From Cloud to Fog to Edge

Source: Fog vs. Edge Computing, Nebbilio Technologies whitepaper, 2018.



Outline

- Distributed Coordination 101
- DCBlocks Platform Summary
- Coordination in the Electric Grid
- Fog/Edge Computing to the Rescue
- **Next-Generation Platform Requirements**

IoT_Coord Platform Goals

- Extensible: can load a new mechanism in field
- Support **multiple implementations** of the same "block" offering **different tradeoffs**
  - High-level list of few parameters (tradeoff choices) are mapped down onto the particular mechanism's parameters by the platform expert
- Switch between multiple implementations of same block to get best tradeoff for present conditions

IoT_Coord Platform Goals (cont.)

- Suport hybrid (multiple) failure models & switching between
- Cyber-Physical
  - Including criteria for leader election
- Support DEEP platform-specific security

Hybrid Failures

- TOLERATE benign failures by default
- MONITOR for evidence of security breaches (“Byzantine” behavior)
- SWITCH failure models (and of course implementations of Blocks such as agreement that use them)
  - Disable some in Byzantine: leader election, others?
- Some open research issues here: how to cleanly switch agreement (etc) instances between
  - Different implementations w/same failure model
  - Different failure models
    - Maybe simple: just stop new agreements/blocks, flush ongoing/pending ones, switch failure assumptions, restart..

WASHINGTON STATE UNIVERSITY

gridstat.net

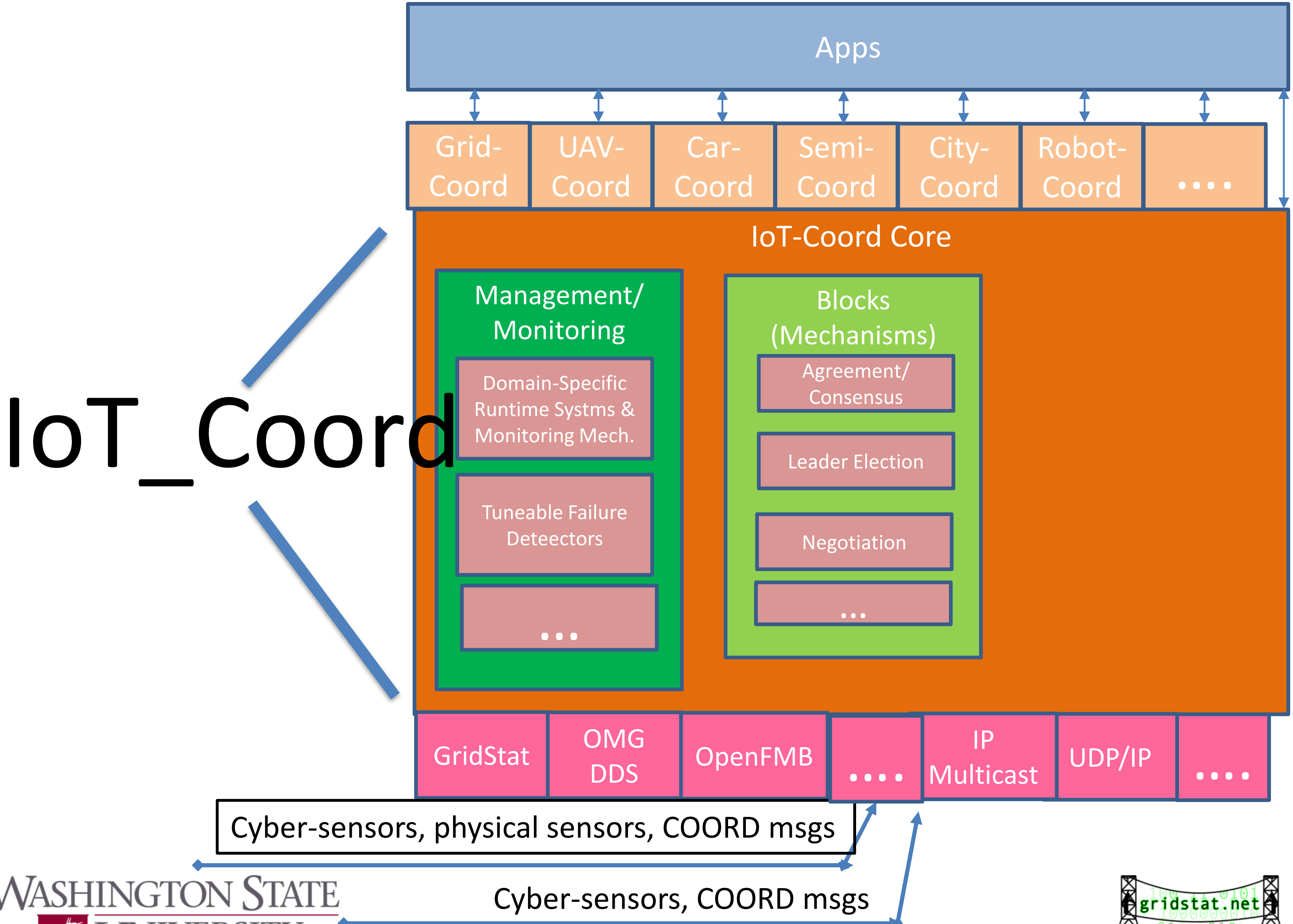
Reusing Existing Software

- Bakken's (sole or joint)
  - Condition: QuO-RTS and SysCond Objects
  - Block::Voting: Voting VM, Mr. Fusion
  - Pub/Sub and reliable delivery: GridStat (can also do 1:1)
  - Security mechanisms: managed chain of filters from GridStat data plane
  - … (bunch more from BBN I am not aware of, most likely)
  - IoTp2p baseline should NOT reuse code: DCBlocks
    - "Plan on throwing one version away. You're going to, anyway", Fred Brooks, *The Mythical Man-Month*, ~1975
- Others'
  - Agreement: RAFT, Paxos, …
  - Group Communication: comes with Akka Java (which we built DCBlocks with)

Conclusions

- Grids are getting more decentralized, mainly due to renewables and other DERs
- Coordination is sometimes required
- Distributed coordination is very tricky to program
- Platform support

Note: Anurag and I are organizing a workshop on grid decentralization with Siemens, EU, and others: Europe in 1Q20. Email if interested


IoT_Coord
Apps
Grid-Coord
UAV-Coord
Car-Coord
Semi-Coord
City-Coord
Robot-Coord
• • • •
IoT-Coord Core
Management/ Monitoring
Domain-Specific Runtime Systms & Monitoring Mech.
Tuneable Failure Deteectors
• • •
Blocks (Mechanisms)
Agreement/ Consensus
Leader Election
Negotiation
• • •
GridStat
OMG DDS
OpenFMB
• • • •
IP Multicast
UDP/IP
• • • •
Cyber-sensors, physical sensors, COORD msgs
Cyber-sensors, COORD msgs

What Can Domain Layers Add?

- Domain-specific APIs: simplified, specialized, …
- Domain-specific policy "languages": Domain-specific coordination/constraint languages (that generate domain-specific API code; ~FSA-like?)
- Domain-specific state machines to describe negotiations (generates code to fill container)
- Domain-specific management/monitoring mechanisms
- Semantic info from the domain to help customize/configure/manage IoT_Coord below
- …. more TBD ….

AI/ML Opportunities

1. Tuning Runtime Management of Different Coordination Mechanisms*
2. Tuning Failure Detectors and other QoS Meta-Data with Situational Awareness
3. Allowing Application to Make a Better Consensus Choice/Vote
4. Allowing Consensus with a Lower Threshold (!?)

- Q: do these need separate containers, or are they just part of the app?

* Hoffert J., Mack D., and **Schmidt D.**, "Integrating Machine Learning Techniques to Adapt Protocols for QoS-Enabled Distributed Real-Time and Embedded Publish/Subscribe Middleware," *International Journal of Network Protocols and Algorithms: Special Issue on Data Dissemination for Large-scale Complex Critical Infrastructures*, vol. 2, no. 3, pp. 1-33, 2010.

Deep Domain-Specific Security

- Obviously, need very deep security (esp. authentication) baked in, with domain-specific and possibly app-specific info
  - App-specific ➔ platform should support

References

[**BAS+17**] D. Bakken, A. Askerman, A. Srivastava, P. Panciatici, M. Seewald, F. Columbus, and S. Jiang. "Towards Enhanced Power Grid Management via More Dynamic and Flexible Edge Communications", in *Proceedings of the Fog World Congress, IEEE and OpenFog*, Santa Clara, CA, Oct. 30, 2017. - http://www.gridict.net/GridFog-FogCongress2017-Extended-v1.5.pdf

[Gop18] Shyam Gopal. "Building Distributed Applications for Monitoring and Control on the Resilient Information Architecture Platform (RIAPS) for Smart Grids". MS Thesis, Washington State University, expected December 2018.

[**KBH+09**] Christophe Kieny, Boris Berseneff, Nouredine Hadjsaid, Yvon Besanger, Joseph Maire. "On the concept and the interest of virtual power plant: Some results from the European project Fenix." In *Proceedings of the 2009 IEEE PES General Meeting*, July 26-30, 2009, Calgary, AB, Canada.

https://ieeexplore.ieee.org/document/5275526

[**KGL+18**] V. Krishnan, S. Gopal, R. Liu, A. Srivastava, and D. Bakken. "Resilient Cyber Infrastructure for the Minimum Wind Curtailment Remedial Control Scheme", *IEEE Transactions on Industry Applications*, 2018, to appear. - https://ieeexplore.ieee.org/abstract/document/8453874

References

[**LeeX17**] HyoJong Lee. *Development, Modeling, and Application of PMUs*. PhD Dissertation, Washington State University, 2017.

[**LNS+16]** Hyojong Lee, Shwetha Niddodi, Anurag Srivastava, and David Bakken. "Decentralized Voltage Stability Control in the Smart Grid using Distributed Computing Architecture", in Proceedings of the 2016 IEEE Industry Applications Society Annual Meeting (IAS), Portland, OR 2-6 October, 2016. - https://ieeexplore.ieee.org/document/7731871

[**LSA+16**] Ren Liu, Anurag Srivastava, Alexander Askerman, David Bakken and Patrick Panciatici, "Decentralized State Estimation and Remedial Control Action for Minimum Wind Curtailment Using Distributed Computing Platform", IEEE Transactions on Industry Applications, 23(6), 2017 https://ieeexplore.ieee.org/abstract/document/8012530

WASHINGTON STATE UNIVERSITY

gridstat.net

References

[**KLA+17**] V. Krishnan, R. Liu, A. Askerman, A. Srivastava, D. Bakken, and P. Panciatici, "Resilient Cyber Infrastructure for the Minimum Wind Curtailment Remedial Control Scheme", in *Proceedings of the 2017 IEEE Industry Applications Society Annual Meeting*, Cincinnatti, Ohio, USA, October 1-5, 2017. https://ieeexplore.ieee.org/document/8101783

[**Nid15**] Shwetha Niddodi, *Decentralized Coordination Building Blocks (DCBlocks) for Decentralized Monitoring and Control of Smart Power Grids*. MS Thesis, Washington State University, December 2015. https://www.eecs.wsu.edu/~bakken/Nid15.pdf

[**PRS07**] Pudjianto, D.; Ramsay, C.; Strbac, G., "Virtual power plant and system integration of distributed energy resources", *IET Renewable Power Generation*, 1(1), pp10 - 16, March 2007