Software Systems for Smart Worlds: An Evolution for IoT to CPS **

Kaliappa Ravindran and Michael Iannelli
Department of Computer Science
City University of New York
(CUNY – Graduate Center & City College)

Email: *ravi@cs.ccny.cuny.edu; miannelli@gc.cuny.edu*

** Other Ph.D. student participants at CUNY

Arun Adiththan, Mohammad Rabby

Outline of talk

• Networks of objects (IoT) as a cyber-physical system

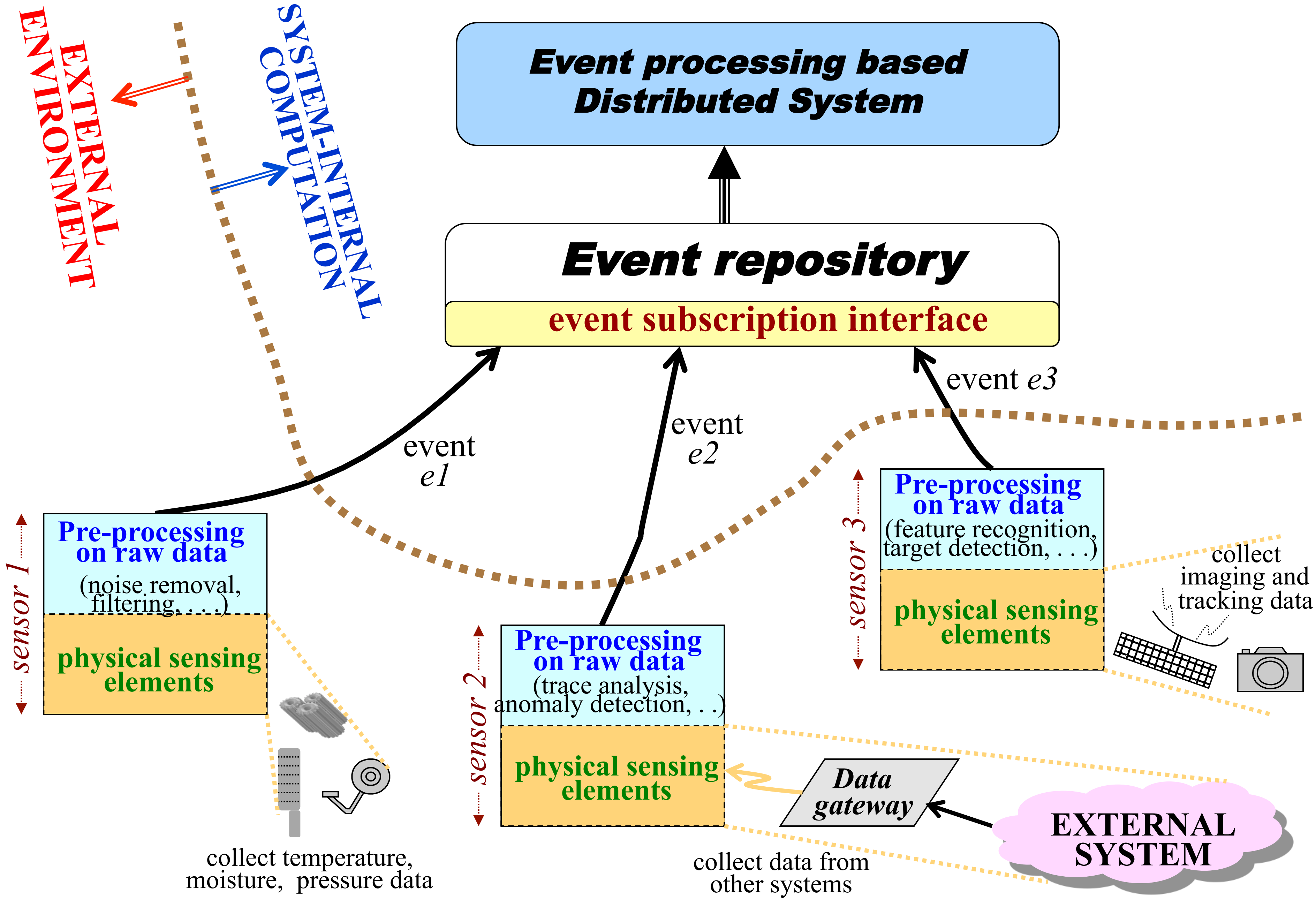
•Support mechanisms for *Smart IoT infrastructure*
[infusion of *control* to the M2M & D2D communication substrate]

•Key element of "control" functionality:
quality of sensing

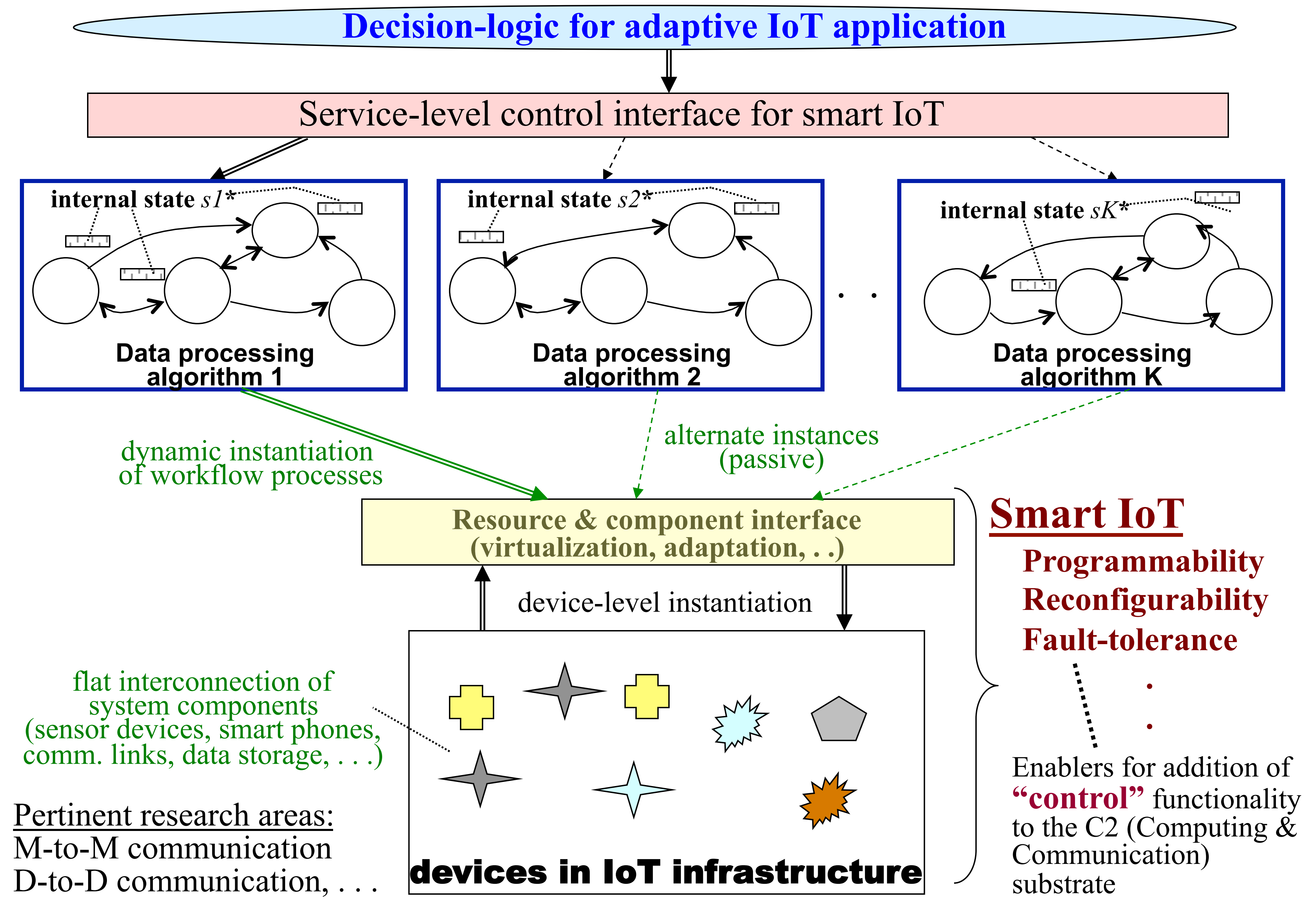
•Fuzziness of input data and system uncertainty
(replicated sensing, distributed sensing)

•Case study of OCR-based text summarization

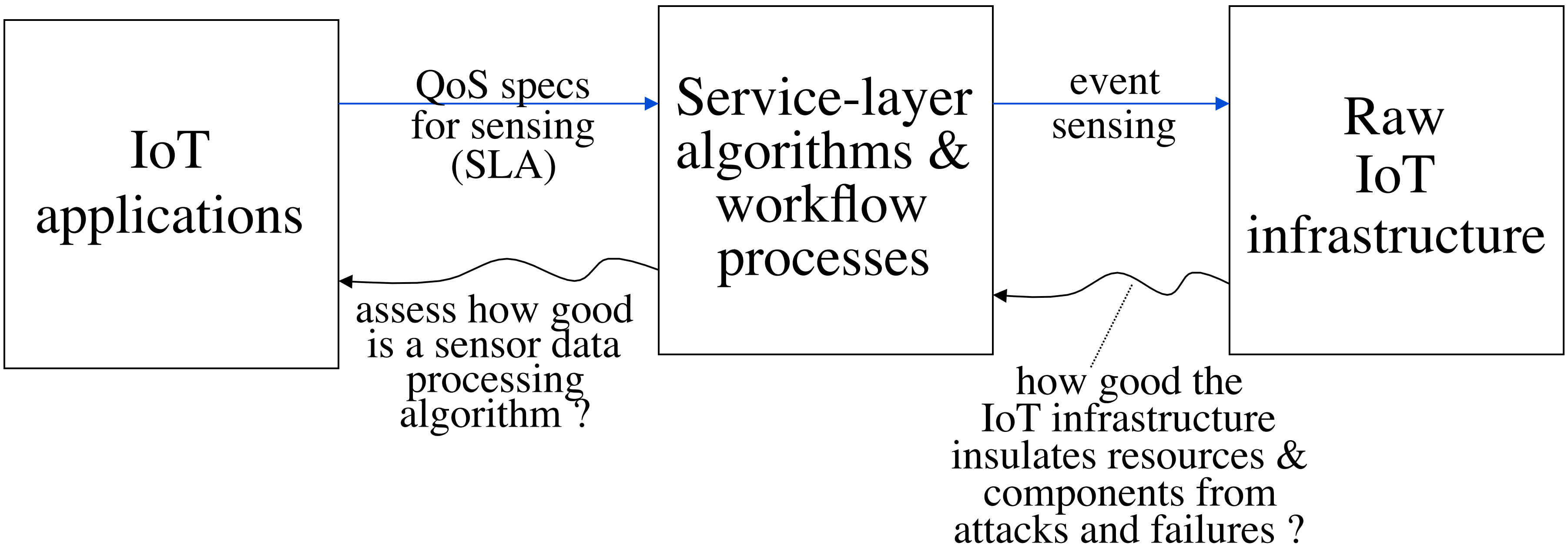
A bird's view of IoT system



Architecture to transition from C2 paradigm to C3 paradigm



Trust relationship between sub-systems in an IoT



A system-level design question:

How much confidence can be bestowed on a sensing sub-system ?

➔ Requires a quantitative benchmarking of the quality of sensors (we need probabilistic techniques)

Our goal in transitioning from IoT to CPS

Design of verifiable IoT software systems where the "event sensing" processes can be managed and reasoned about in *probabilistic* ways

A cyber-physical system *S* that cannot be *verified* about "what it is sensing from the external physical environment" is useless in meeting the mission-critical quality needs (however good *S* has been designed to be)

Verifiability is an integral part of good design practice (from a software engineering angle) !!

QoS-oriented goals of event sensing in IoT system

- Provision of ***programmable*** quality of data delivery as IoT service

  Application-level QoS attributes are:

  ^^ Timeliness

  ^^ Accuracy

  ^^ Fault-tolerance

  .

  .

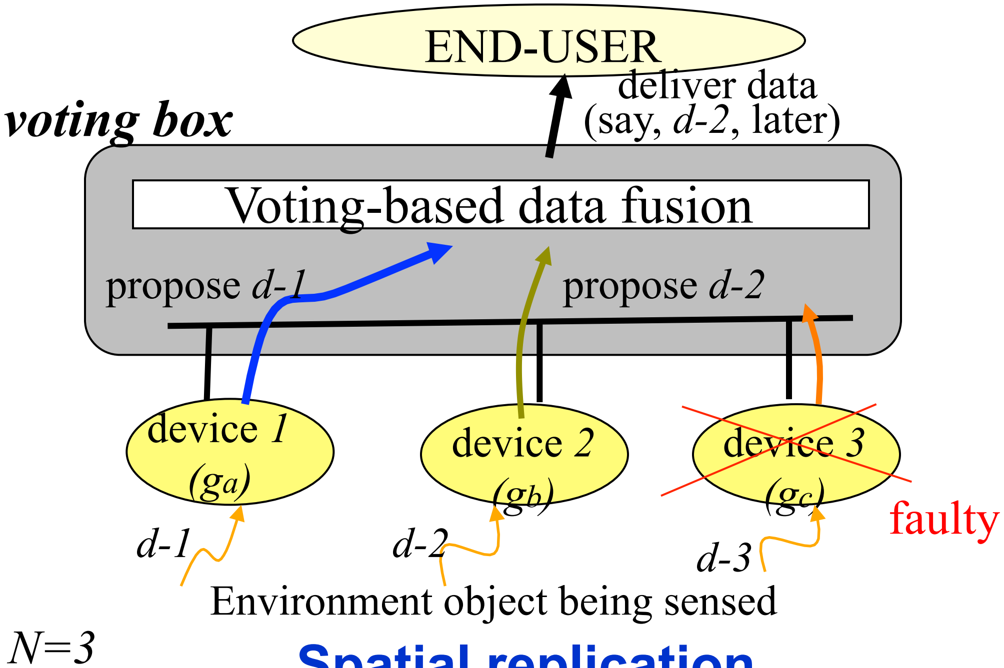
- Realization of service-layer algorithms to enforce ***in-time*** and ***sufficiently-accurate*** data delivery in a backdrop of failures/attacks

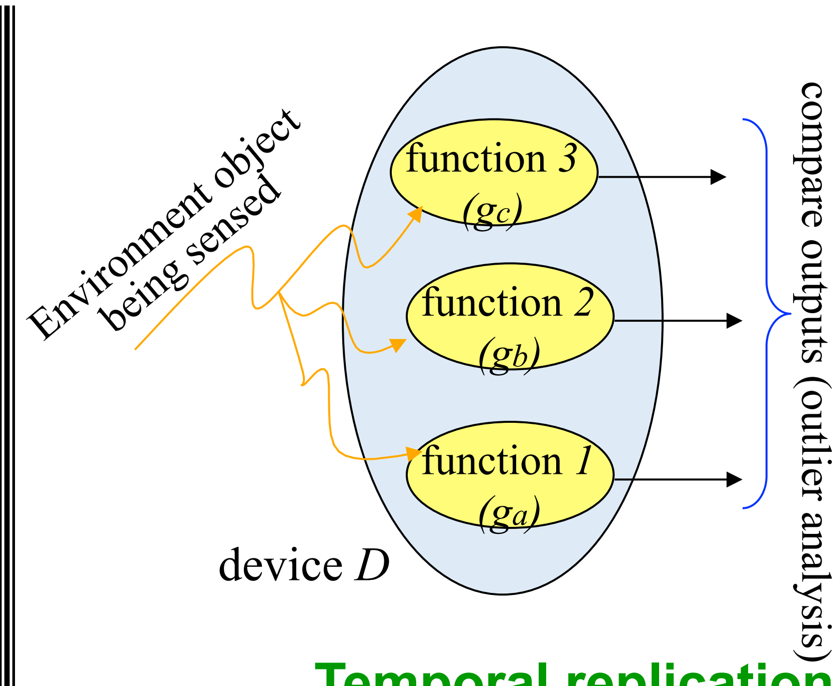
  Resource-optimal protocols for data delivery

  ^^ Reduce message overhead incurred by network

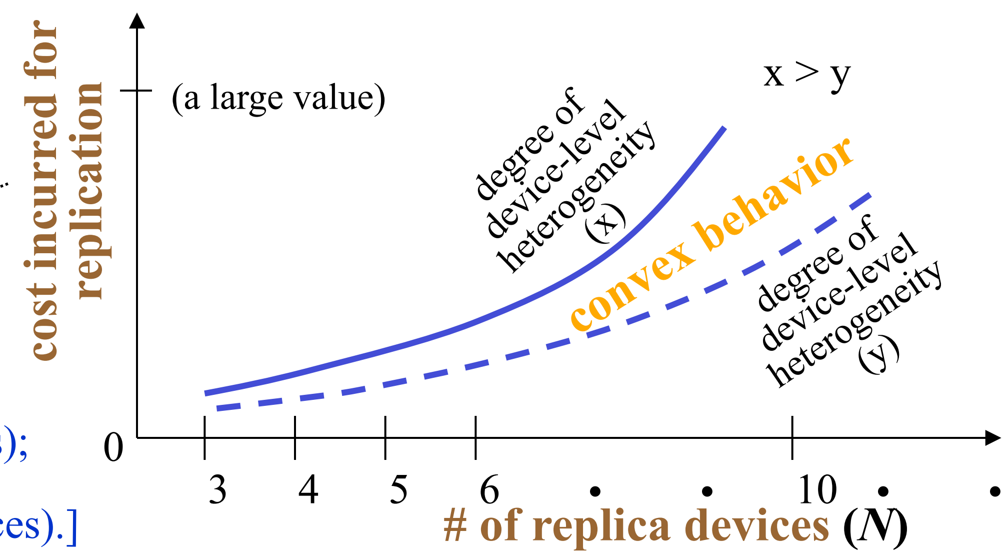
  ^^ Reduce computational cycles expended at devices

Component replication at sub-system interfaces

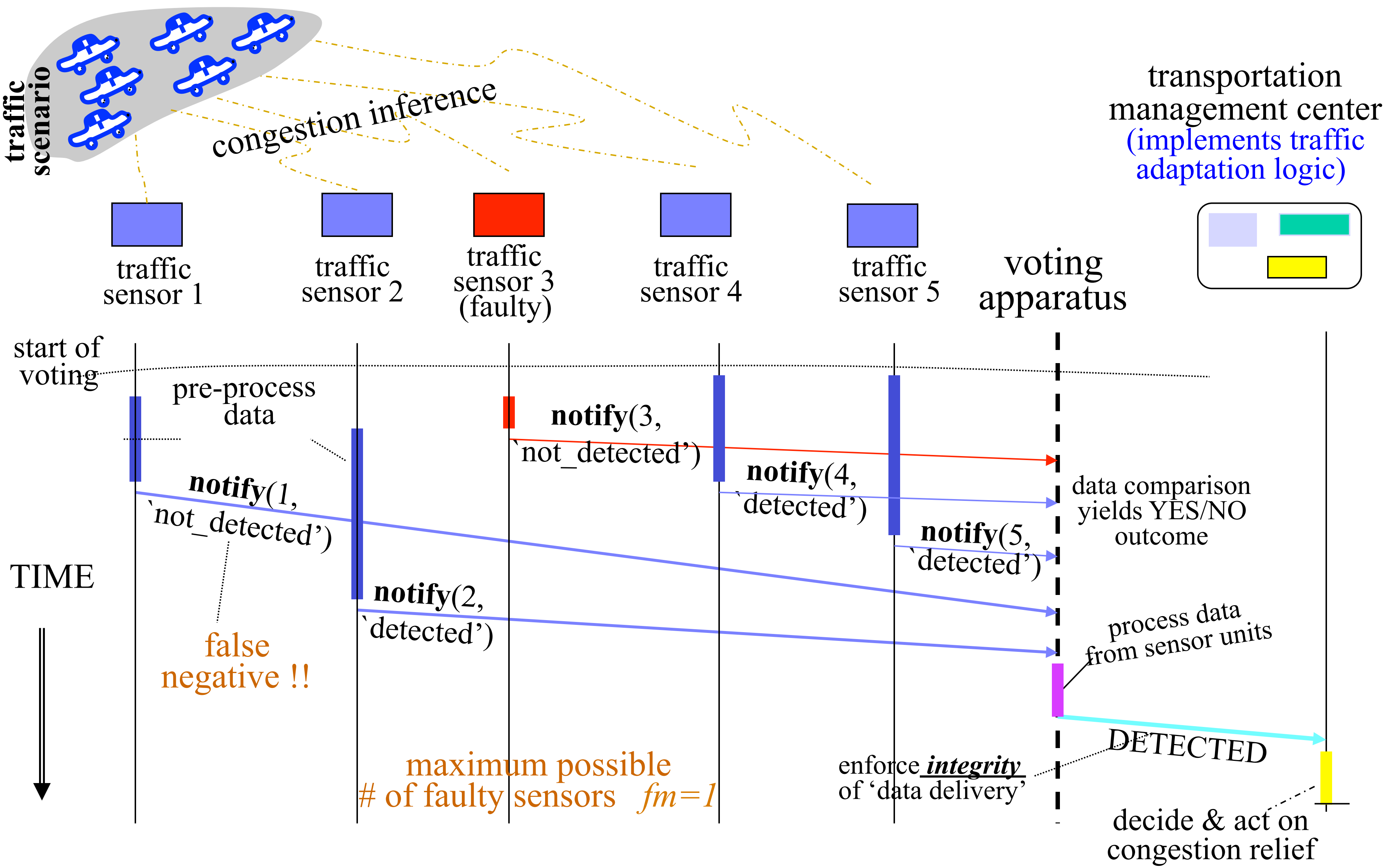




['cost' is determined by the amount of:
1. computational efforts expended at system design level (to replicate device functions);
2. deployment/maintenance efforts expended in physical world (to install multiple devices).]



Example: A sensor replication based traffic congestion inference



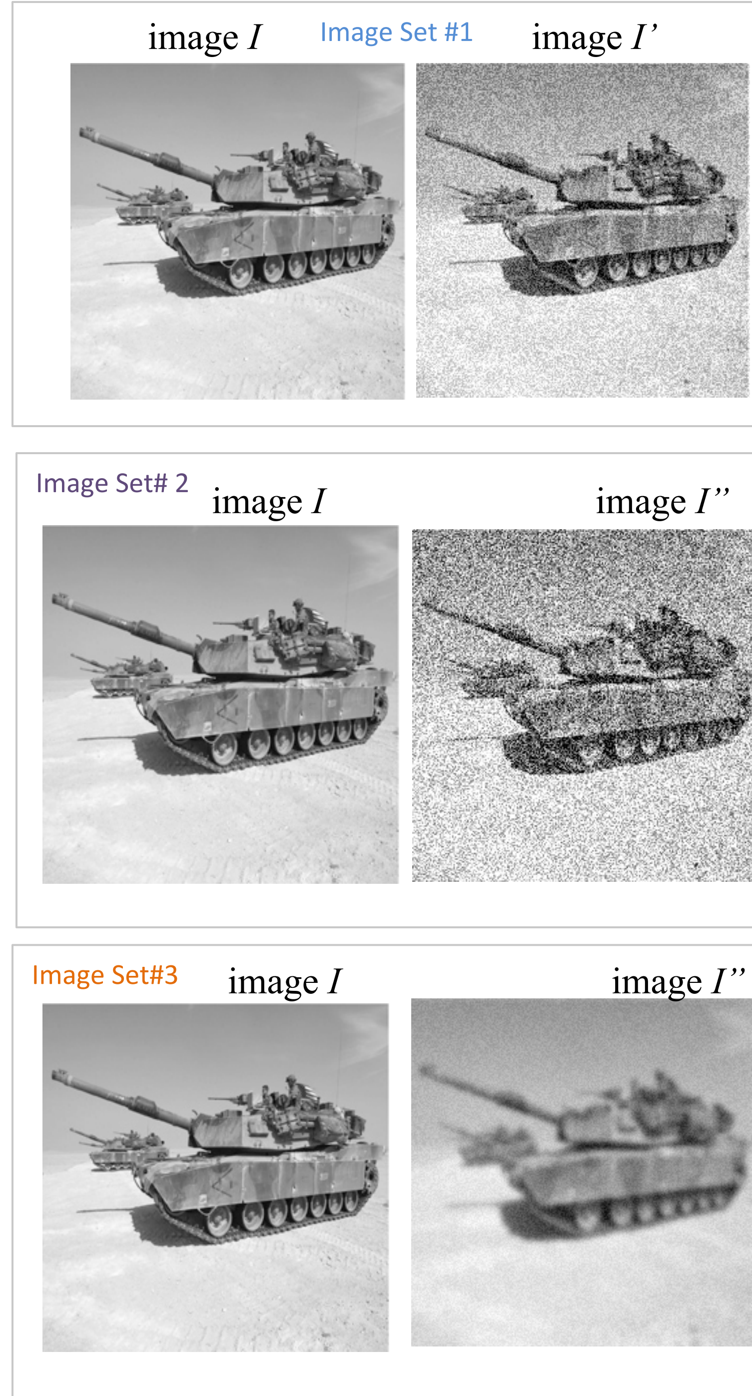
We have also done studies on multiple cameras detecting a tank in battle terrain

Total # of grids=32; Grid size: 8X8 pixels; Image size: 256X256 pixels

δ : Per-grid mis-match tolerance in average pixel intensity

	Threshold(% of Grid Match) (g_{min})	Image Set #1	Image Set#2	Image Set #3
TABLE-A $\delta = 5\%$	84%	NO	NO	NO
	82%	NO	NO	YES
	80%	NO	NO	YES
	78%	NO	NO	YES
	76%	YES	NO	YES
	74%	YES	NO	YES
	72%	YES	YES	YES
TABLE-B $\delta = 3\%$	84%	NO	NO	NO
	82%	NO	NO	NO
	80%	NO	NO	NO
	78%	NO	NO	NO
	76%	NO	NO	YES
	74%	NO	NO	YES
	72%	NO	NO	YES
	70%	YES	NO	YES
	68%	YES	NO	YES
	66%	YES	NO	YES
	64%	YES	NO	YES
	62%	YES	NO	YES
	60%	YES	YES	YES
TABLE-C $\delta = 2\%$	84%	NO	NO	NO
	82%	NO	NO	NO
	80%	NO	NO	NO
	78%	NO	NO	NO
	76%	NO	NO	NO
	74%	NO	NO	NO
	72%	NO	NO	NO
	70%	NO	NO	NO
	68%	NO	NO	NO
	66%	NO	NO	YES
	64%	NO	NO	YES
	62%	NO	NO	YES
	60%	YES	NO	YES
	58%	YES	NO	YES
	56%	YES	YES	YES

I: original image; *I'*,*I''*,*I'''*: noise-injected/blurred images



Impact of resource constraints and data characteristics on accuracy of event reporting

1. Resource constraints of data collection devices

→ limited amount of CPU cycles, memory size, battery energy in hand-held devices

→ Limited bandwidth in network paths

2. Large dimensionality of data collected from environment

→ 100's of object features may need to be sensed and processed by data collection devices

'data classifier' algorithm $M(F,L)$ embodied in device V represents the computational processing for object detection over the object space $\boldsymbol{O}$

F: feature vector to describe object images

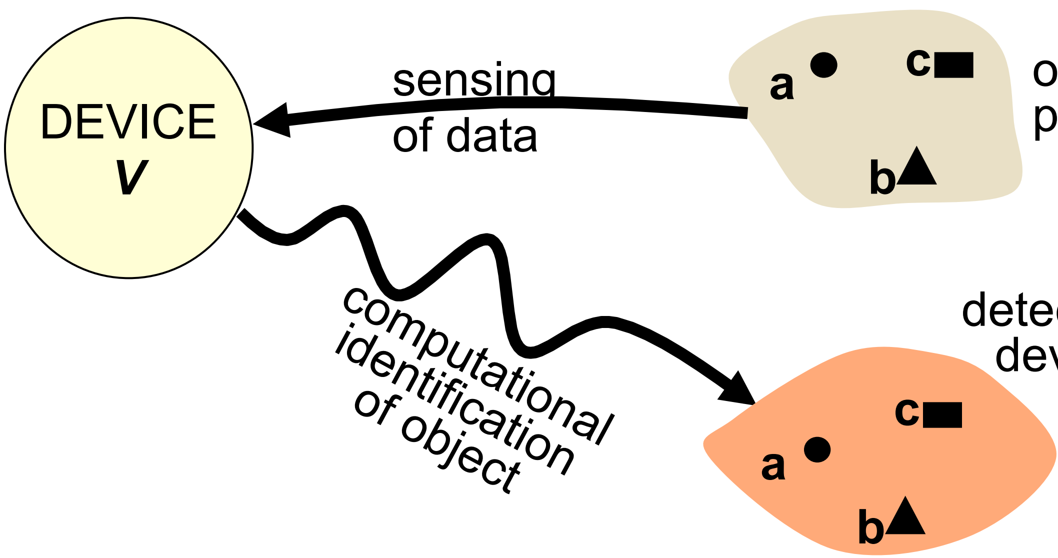
L: logical formulas to relate features

e.g., aircraft detection: features of target: {length l, height h, speed s}

logical formula: "plane_detected $\equiv$ ($l > 90'$ AND $h > 12'$ AND $s > 300\ mph$)"

1 and 2 impact the system capabilities to accurately represent the data about an object from various devices and then vote on these data within a stipulated time

Probabilistic fault models for fuzzy data in IoT systems



detected object $\boldsymbol{x'}$ =
compute_object (sensed data $\boldsymbol{x}$ from physical world)

Existing models of voting

Premise of ***exact voting***, i.e., a non-malicious device *V* is 100% certain that the object it claims to have detected, based on *V*'s locally computed data, matches with the actual object in physical world (within reasonable accuracy ε):

$$|x'| - |x| < \varepsilon \quad \forall\, x \in \boldsymbol{O}$$

New model of voting relevant for IoT systems

Fuzzy voting: the device *V* is less-than-100% certain on the accuracy of object it claims to have detected, relative to the actual object in physical world

$p(x)$ = probability (detected_object=x' | actual_object=x) for $|x'| - |x| < \varepsilon$, where

$$0.0 \ll p(x) < 1.0 \quad \forall\, x \in \boldsymbol{O}$$

confusion matrix to capture uncertainty in object detection

actual object / sensed object	$x1$	$x2$	. . .	xk
$x1$				
$x2$				
. .			. .	
xk				

EXAMPLE (aircraft detection) **

aircraft	F-15	T-38A	falcon-20	falcon-100
F-15	0.97	0.03	0.0	0.0
T-38A	0.03	0.96	0.0	0.01
falcon-20	0.0	0.01	0.87	0.12
falcon-100	0.0	0.01	0.12	0.87

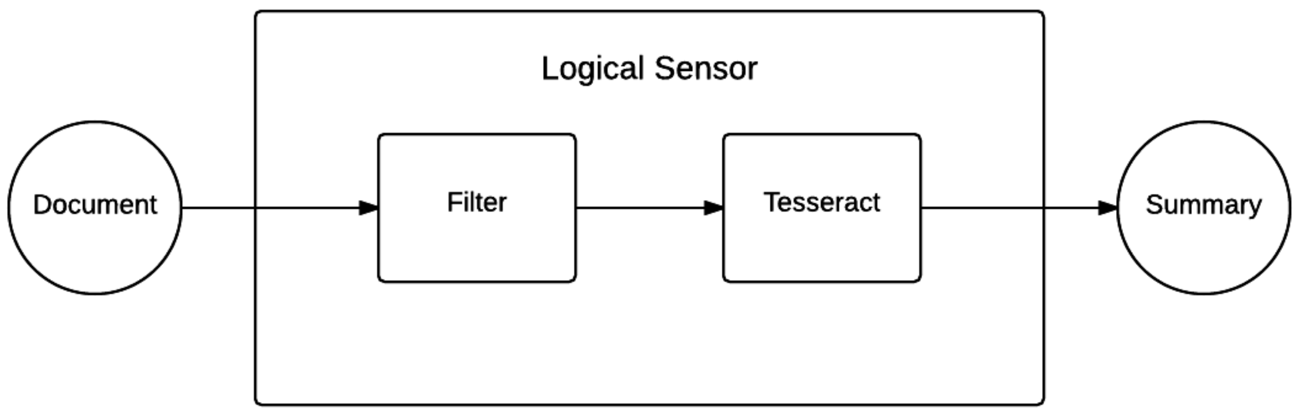
ij-th element of matrix: conditional probability that the object detected is xi when the object occurrence in physical world is xj

i.e., probability (sensed_object = xi | actual_object = xj) $\forall\ xi, xj \in \boldsymbol{O}$

[** Refer to ***Automated Target Recognition Using Passive Radar and Coordinated Flight Models***, Lisa M. Erhman and Aaron D. Lanterman, proc. SPIE 5094, *Automated Target Recognition XIII,* Orlando (FL), April 2003]

Sensor calibration is also useful in obstacle sensing on roads, traffic congestion reporting, etc [switching to different sensors under different observed conditions: e.g., night-time *vs* day-time, distance to obstacle, visibility under fog, etc]

Case study:
OCR (optical Character recognition) as a sensor



A logical sensor was created that would accept an image of a document as input, run this input through an arbitrary filter, run the filtered output through Tesseract, and return the OCR string output summary. The output summary was then compared to the ground truth using *tf-idf* vectors to measure the accuracy of each sensor. Five such logical sensors were created and tested on three documents.

Application example: Law-enforcement in major cities examining a large # of social media postings, text extraction from YouTube clips, etc

Case study of OCr (. . . contd.)

An example Document Input/Output

Representation of text as image
(hard to look for specific words/phrases)

Helvetica

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Representation of text in ASCII
(easily searchable texts/phrases)

Helvetica

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Measurement results from OCR processing on 3#s of PlanetLab nodes

OCR – Local Machine
19.39308

Table 1: Time (seconds) to Compute OCR Task on Local Machine

	OCR – Wifi	OCR - AirCard
planetlab2.tsuniv.edu	25.61008371	28.51807368
pl2.rcc.uottawa.ca	27.741041	27.61074164
planetlab1.cs.colorado.edu	31.64578674	60.43584437

Table 2: Time (seconds) to Compute OCR Task on 3 Different Remove VMs

	Data Transfer – Wifi	Data Transfer - AirCard
planetlab2.tsuniv.edu	44.98775117	46.84633965
pl2.rcc.uottawa.ca	8.424560809	3.419660211
planetlab1.cs.colorado.edu	76.36823809	91.05481637

Table 3: Time (seconds) to transfer 2.5 MB file from 3 Different Remote VMs

The OCR task for the 2.5MB bitmap file took an average of 19.39 seconds on the local machine. When run as a surrogate computation one the 3 remote VMs, it took only slightly longer, accounted for by latency between client and server and time to download necessary software packages. Lastly, time to transfer the 2.5MB file was tested between the client and remote VMs.

OCR experiments: controlled injection of fuzziness in raw input data

Filters Used

- Gaussian Blur – A low pass filter that reduces image noise and detail. It blurs the input image by a Gaussian function.
- Smooth – Points in the data set are modified so that adjacent points are closer to each other in magnitude. Points higher than adjacent points are decreased, while those that are lower are increased. For example, a moving average will accomplish this effect.
- Sharpen – A high pass filter that exaggerates smaller details. The opposite of a blur filter.
- Edge Enhance – Enhances the edge contrast of an image to improve its sharpness.

Sample timeliness & accuracy assessment of OCR sensors

Sensor	Image	Accuracy	Time to Process(s)
Sensor 1	Image 1	0.0	11.95
	Image 2	0.0	2.67
	Image 3	0.0	0.37
Sensor 2	Image 1	.32	13.23
	Image 2	.83	8.72
	Image 3	0.0	1.82
Sensor 3	Image 1	.74	10.84
	Image 2	.92	11.11
	Image 3	.06	4.27
Sensor 4	Image 1	.83	11.36
	Image 2	.93	10.72
	Image 3	.07	4.82
Sensor 5	Image 1	.37	13.20
	Image 2	.61	11.14
	Image 3	0.0	1.86

Sample timeliness & accuracy assessment of OCR sensors

Confusion Matrix

	Image 1	Image 2	Image 3
Sensor 1	0.0	0.0	0.0
Sensor 2	.32	.83	0.0
Sensor 3	.74	.92	.06
Sensor 4	.83	.93	.07
Sensor 5	.37	.61	0.0

Analysis of sensing quality with device replication & heterogeneity

Define two metrics: 1. ***confidence*** that a data *Xi* delivered to the user is good --- Θ;
2. ***availability*** of data *Xi* to the user in the presence of uncertainty about the goodness of *Xi* --- ζ

Let K_{1_l} → minimum # of YES votes needed to attain a confidence higher than the per-device confidence level *p(Xi)*

^^ Delivery occurs after receiving a certain # of YES votes K_{1_h} in order to attain a confidence Θ

$$1 - \binom{N\text{-}1\text{-}fm}{K_{1_h}\text{-}fm}\,[1\text{-}\,p(Xi)]^{K_{1_h}\text{-}fm+1\text{-}K_2} > \Theta, \quad \text{where} \quad \Theta > p(Xi) \text{ and } K_{l_h} \geq K_{l_l}$$

^^ Estimate how often *Xi* is not delivered even though K_{1_l} YES votes have been received

$$\zeta \leq (1.0 - \sum_{r=0}^{K_{1_h}-1-K_{1_l}} \binom{N-1-K_{1_l}-f_m}{r} [p(X_i)]^r [1-p(X_i)]^{N-1-K_{1_l}-f_m-r}).$$

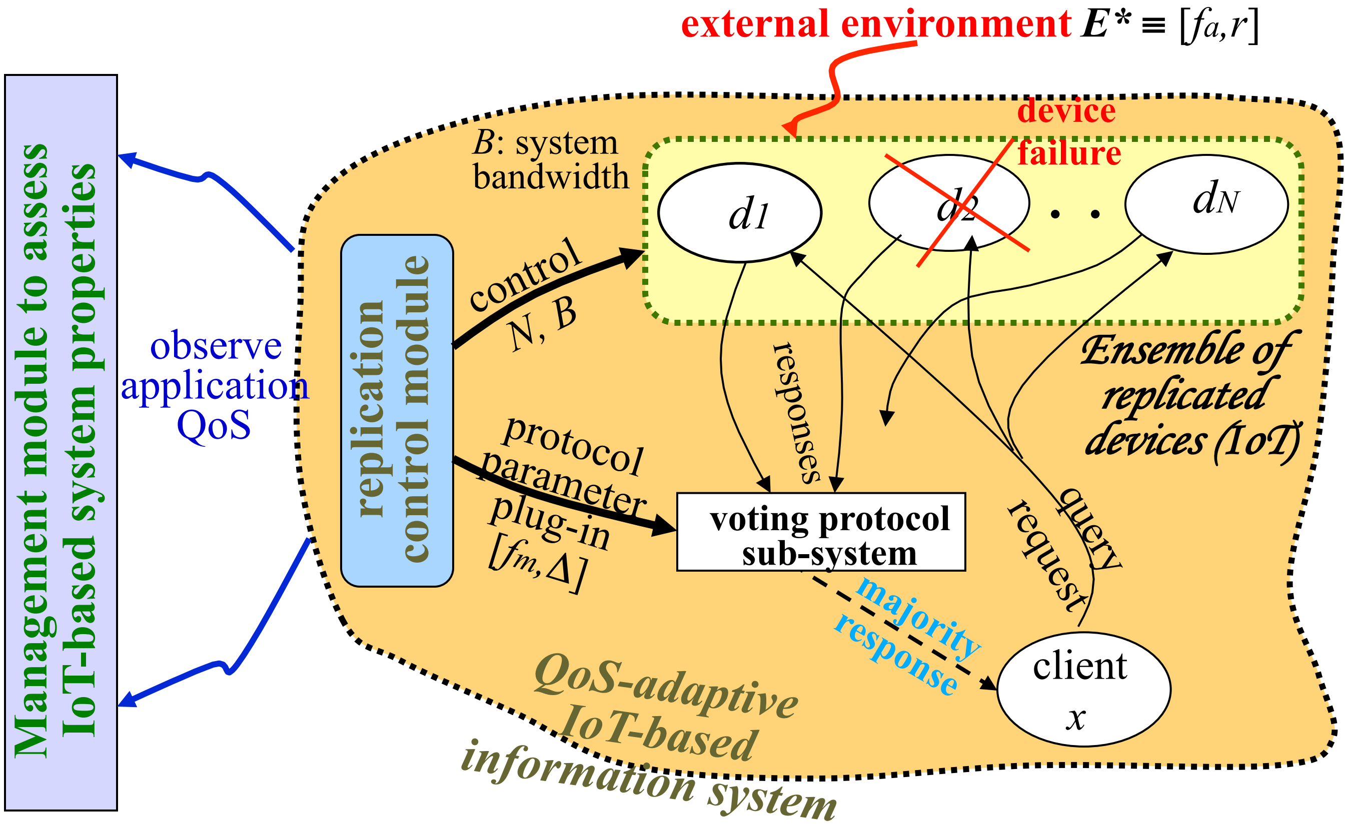
Sample estimates

Input parameters:
$N=17$; $fm=2$;
$p(Xi)=0.8$

K_{1_l}	K_{1_h}	Confidence in data delivery($\geq \Theta$)	Data availability ($\geq \zeta$)
11	***	35.94 %	***
12	12	98.72 %	100 %
12	13	99.98 %	96 %
12	14	99.99 %	64 %

*** : not possible

Experimental study of IoT management (with device replication)



f_m: Max. # of device failures assumed by voting protocol

r: degree of faulty behavior of a failed device ($0 < r \leq 1$)

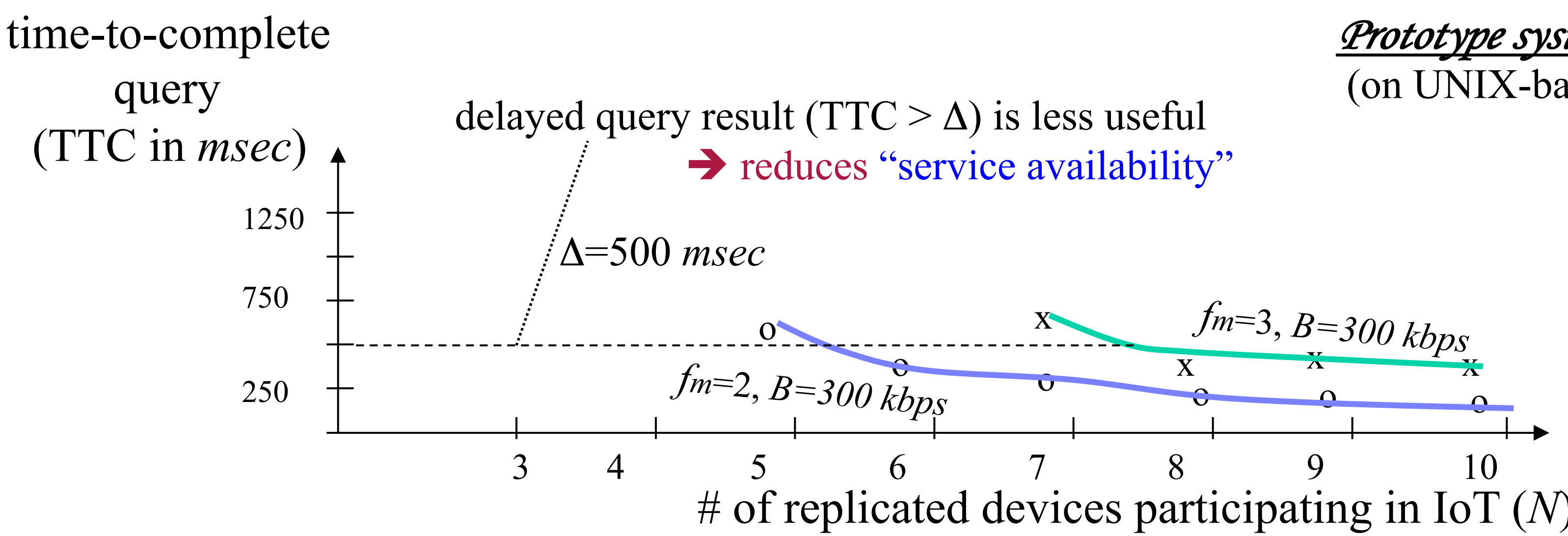
f_a: Actual # of device failures

K: Total device pool size($K \geq N$)

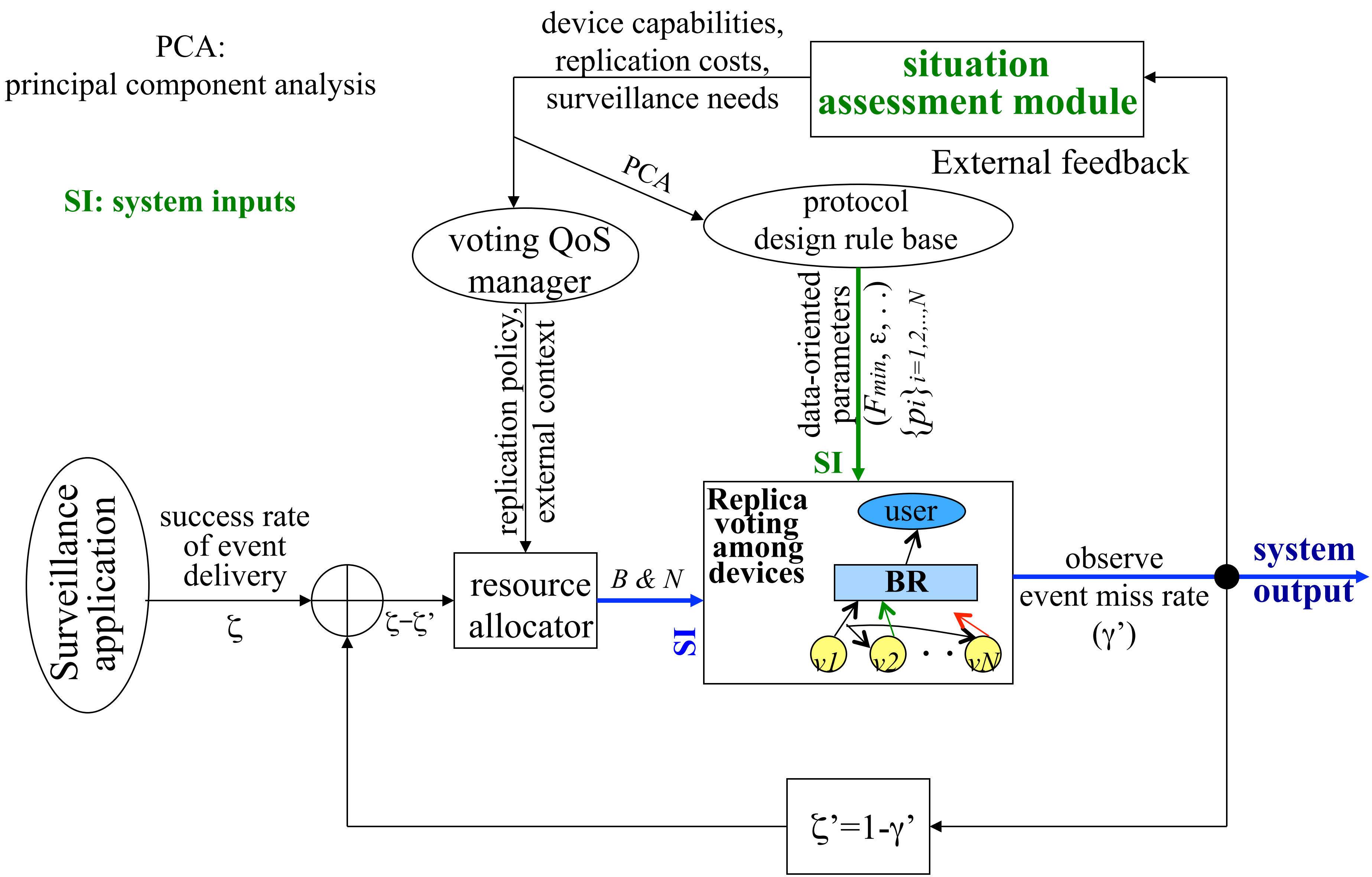
$$1 \leq f_m < \left\lceil \frac{N}{2} \right\rceil; \quad 0 \leq f_a \leq K$$

Prototype system results

(on UNIX-based LAN)



Autonomic management of device replication



N; number of replicas; pi: per-device confusion probability

B: device-level CPU cycles & network bandwidth for voting

Future research plans

- Injection of attack, stressor events on IoT system being tested
- Incorporation of system utility functions and QoS penalty as part of dependability analysis of IoT systems
- Identification of probabilistic measures of sensing quality
- Machine-intelligence and Markov decision processes for sensor system analysis
- Software cybernetics methods for autonomic system improvement