A Synergy of Wireless Sensor Networks and Data Center Systems

Ke Hong

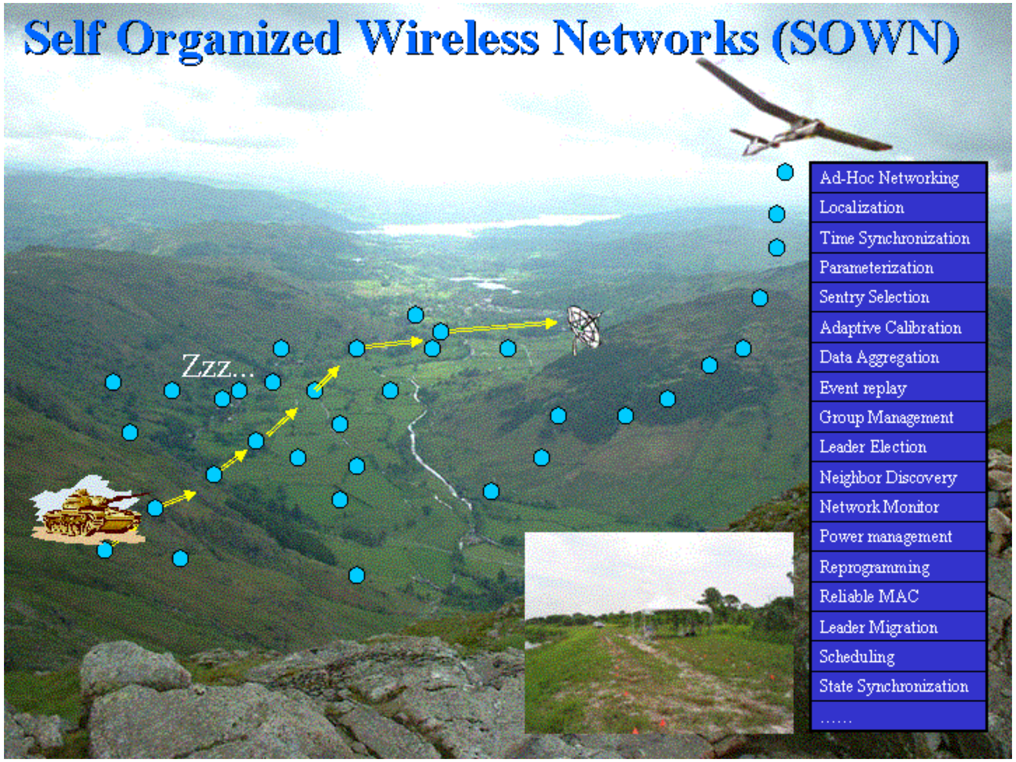
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology

MPhil Thesis Defense

Dec 17, 2013

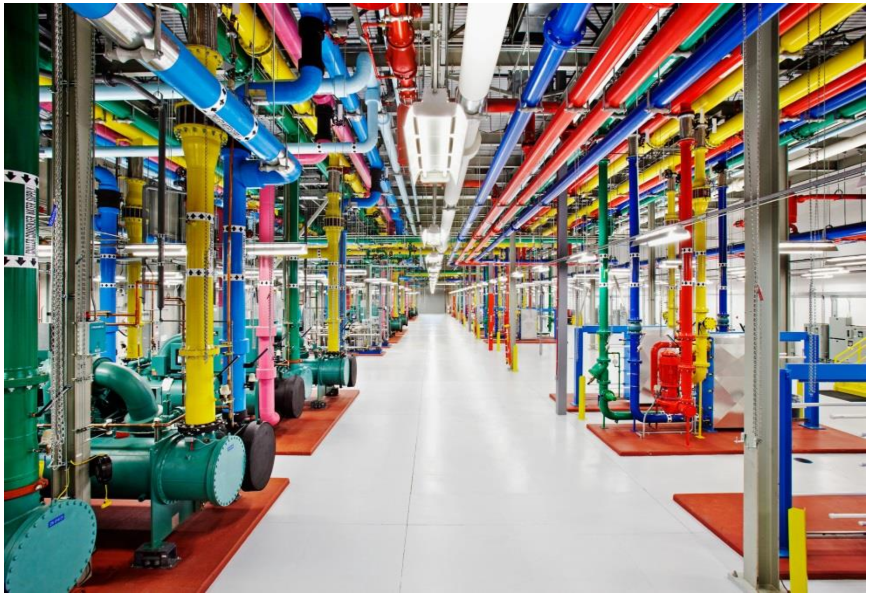
Data Center vs. Sensornet

- Both distributed, dense, scalable
  - 300 nodes in VigilNet, hundreds in GreenOrbs, 1000+ in ExScal



Data Center vs. Sensornet

- Both distributed, dense, scalable
  - 300 nodes in VigilNet, hundreds in GreenOrbs, 1000+ in ExScal
  - Thousands of compute servers organized in racks [Google, Microsoft Quincy]



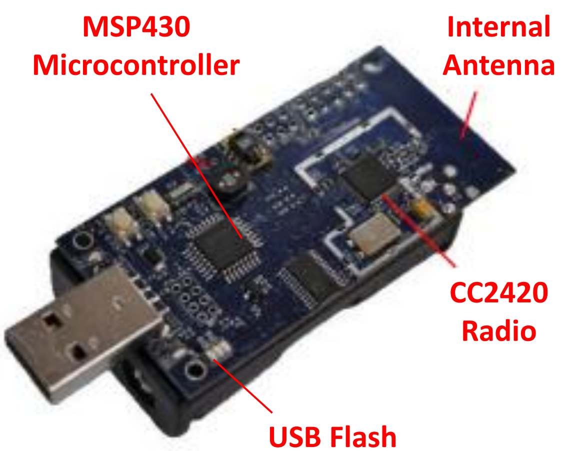
Data Center vs. Sensornet

- Both distributed, dense, scalable
  - 300 nodes in VigilNet, hundreds in GreenOrbs, 1000+ in ExScal
  - Thousands of compute servers organized in racks [Google, Microsoft Quincy]

- Low-end and high-end of computation
  - Limited computing resource on each sensor node
  - Abundant computing resources on rack servers

Related Work

- Sensornet in data centers
  - "Cool" scheduling [USENIX '05]
  - RACNet [SenSys '09]
  - Thermocast [KDD '11]



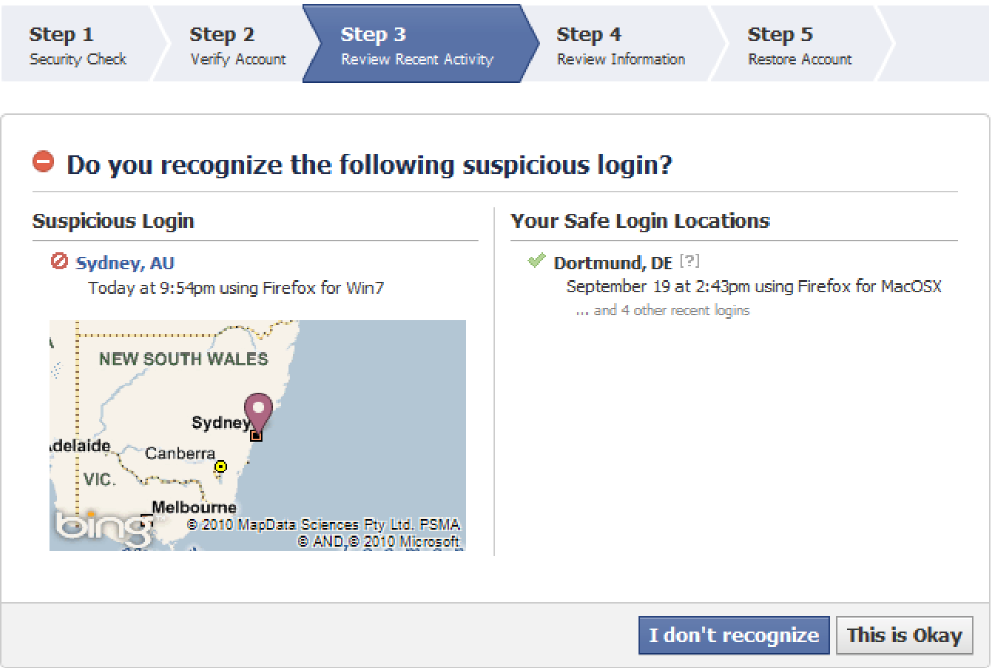
- The combined computational and networking capability of a sensornet enables it to interact with compute clusters in a more sophisticated way

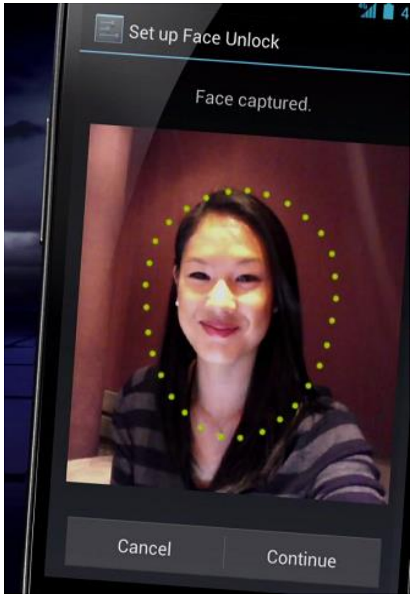
Management in Data Centers

- Software reprogramming on compute servers
  - System settings, configuration files, software upgrade
  - Usually performed on a management station
  - Require certain manual operations

- Why not wirelessly broadcast commands and small files via a sensornet?
  - Wireless as a convenient and flexible broadcast medium

Security Hints





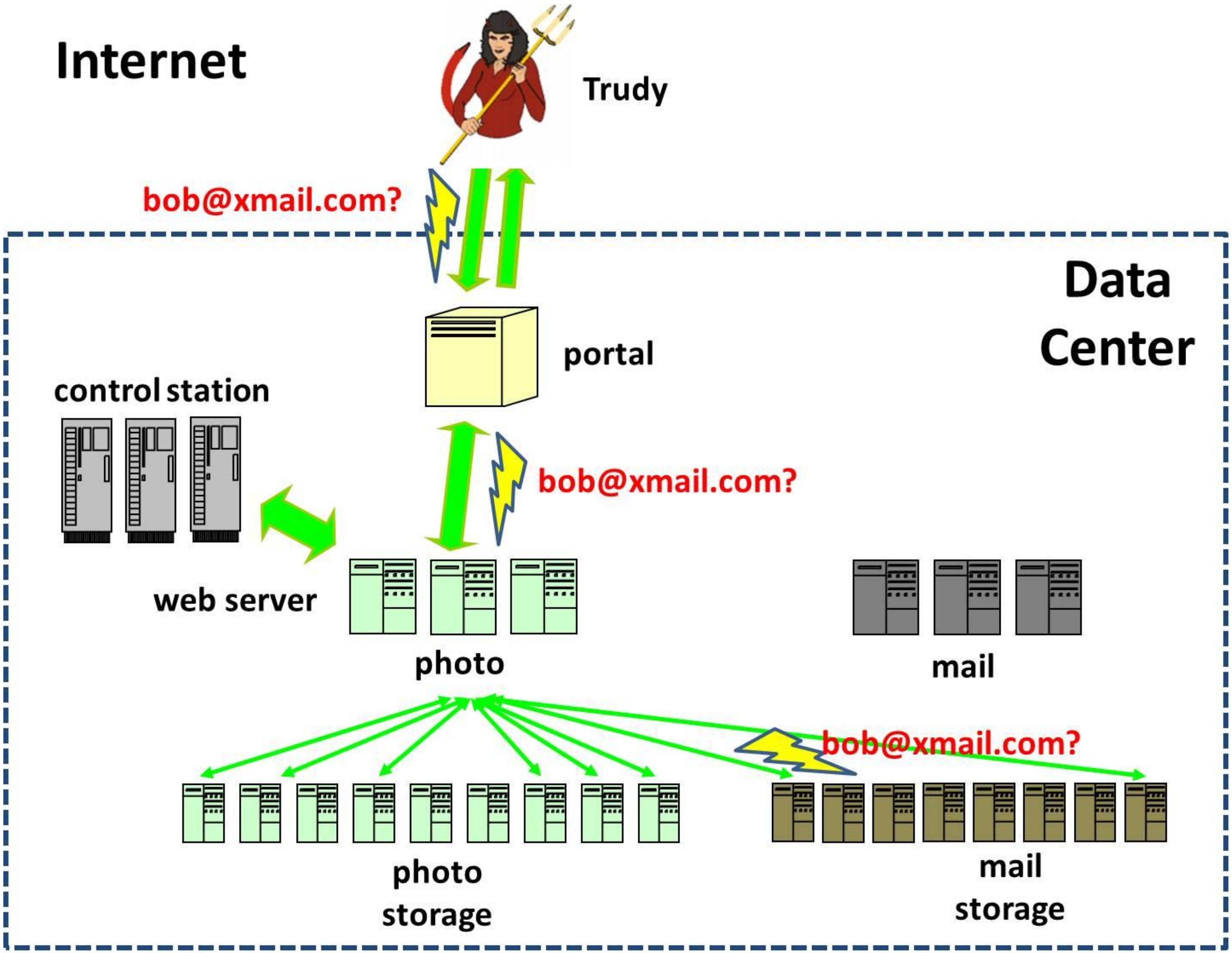
Two-step verification adds an extra layer of protection to your account. Whenever you sign in to the Dropbox website or link a new device, you'll need to enter both your password and a security code sent to your mobile phone.

Security in Data Centers

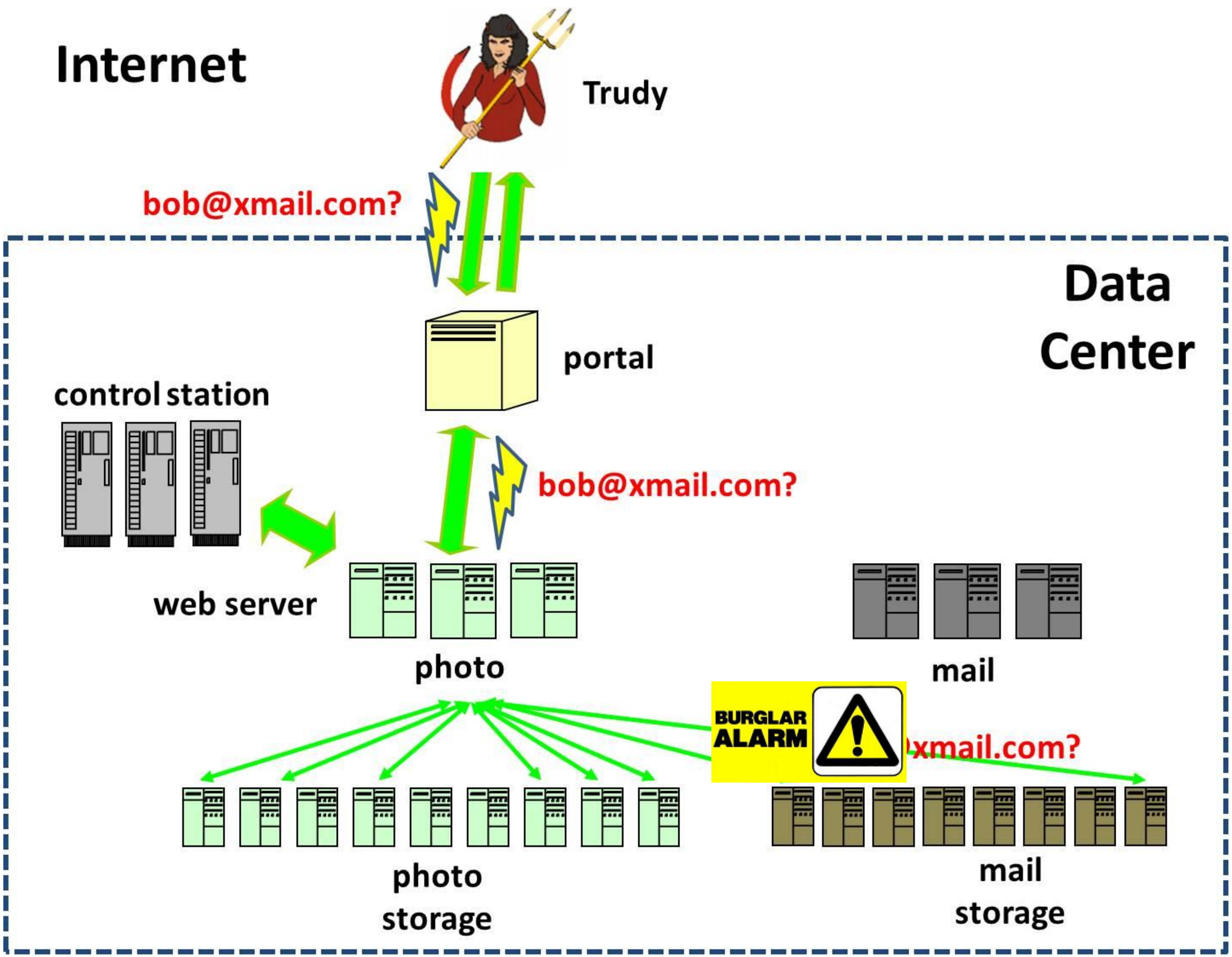
- Existing cryptologic methods do not entirely ensure the operational security of data centers
  - User account leakage at *Yahoo!*, *Sony PlayStation Network* and *Qriocity*
  - Need additional measures for security monitoring

- New security hint: **servers' physical presence**
  - Servers in data centers usually serve different roles (i.e. management, web agent, mail agent, storage)
  - Alarm triggered upon request from strange roles

Access Path Verification



Access Path Verification



Cluster Network in DCs

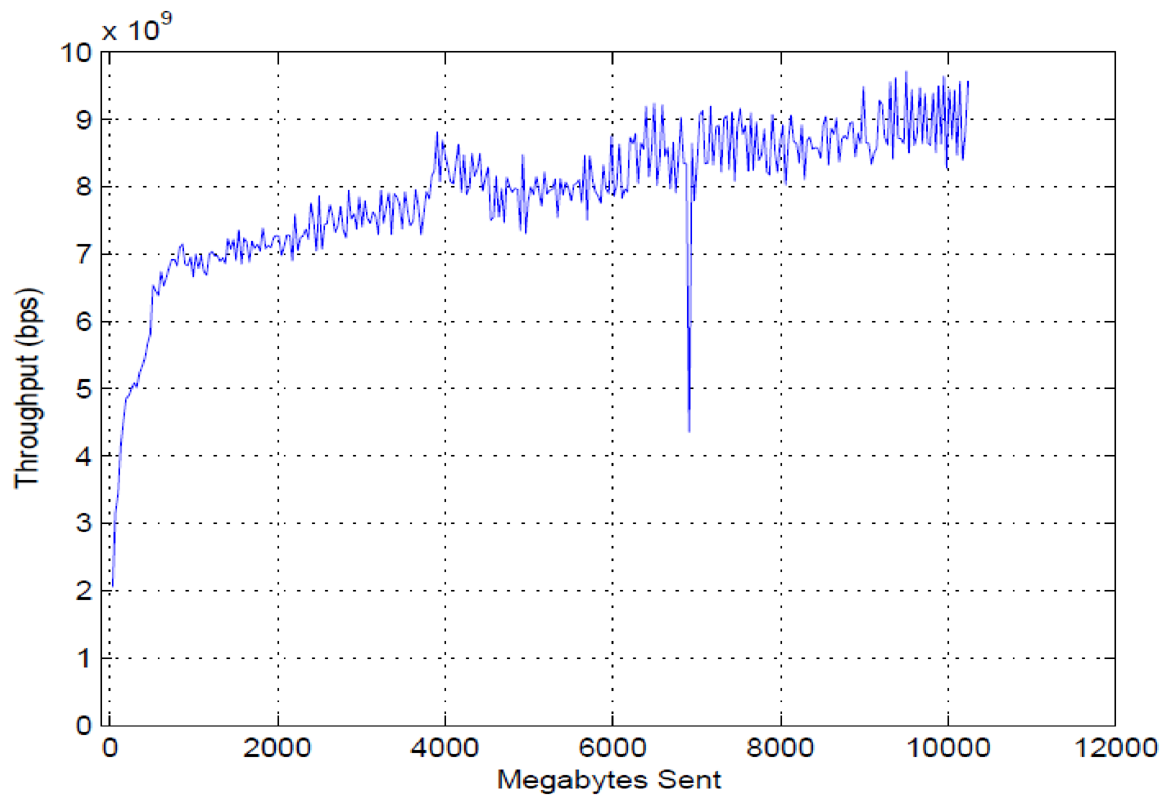
- Cluster network in data centers
  - Ethernet-based
  - High-bandwidth (1Gbps, 10Gbps, 40Gbps)
  - Low-latency (< several hundred microseconds)

- Characterization of data center traffic
  - 99.91% of flows are TCP flows
  - 99% of TCP flows of size < 100MB
  - Low degree of flow multiplexing

TCP in 10GbE Cluster Network

- Low utilization rate in data center traffic
  - CUBIC's inefficacy in coping with the small RTTs

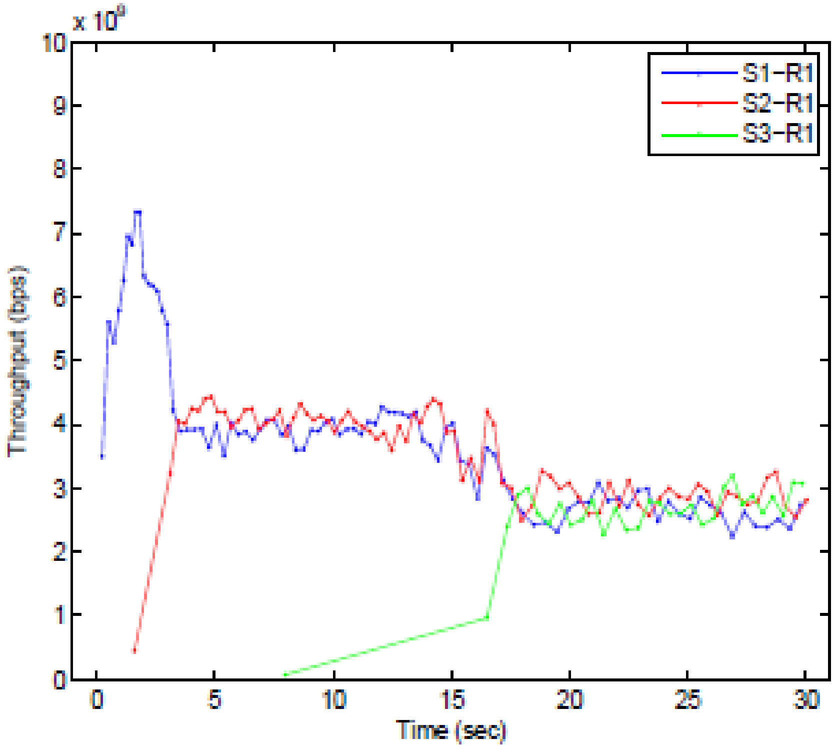
$$W(t) = C(t-K)^3 + W_{max}$$

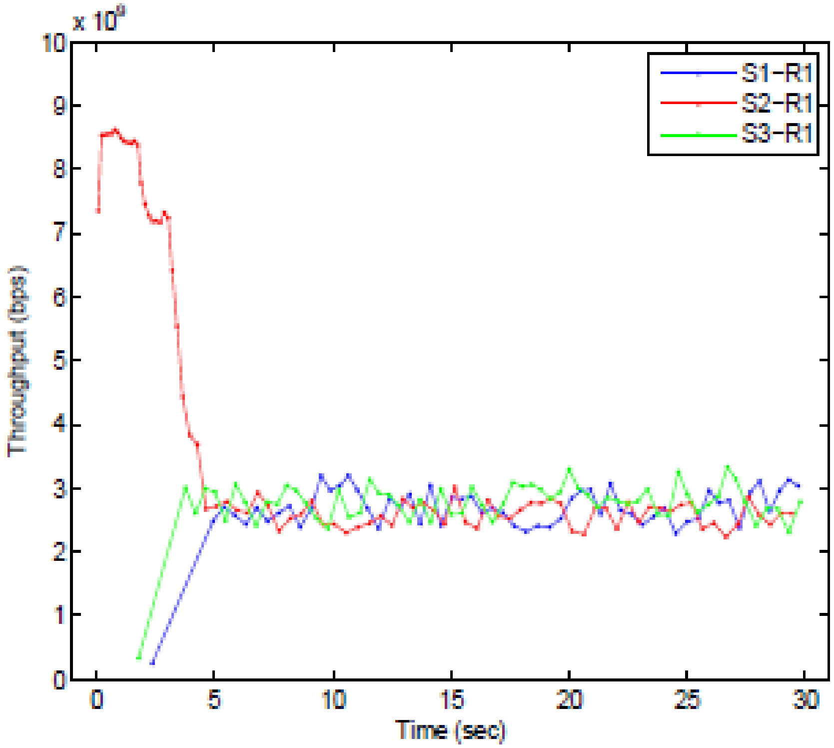


TCP in 10GbE Cluster Network

Unpredictable bandwidth sharing

– ECN congestion signaling in DCTCP $cwnd = cwnd * (1 - \frac{\alpha}{2})$

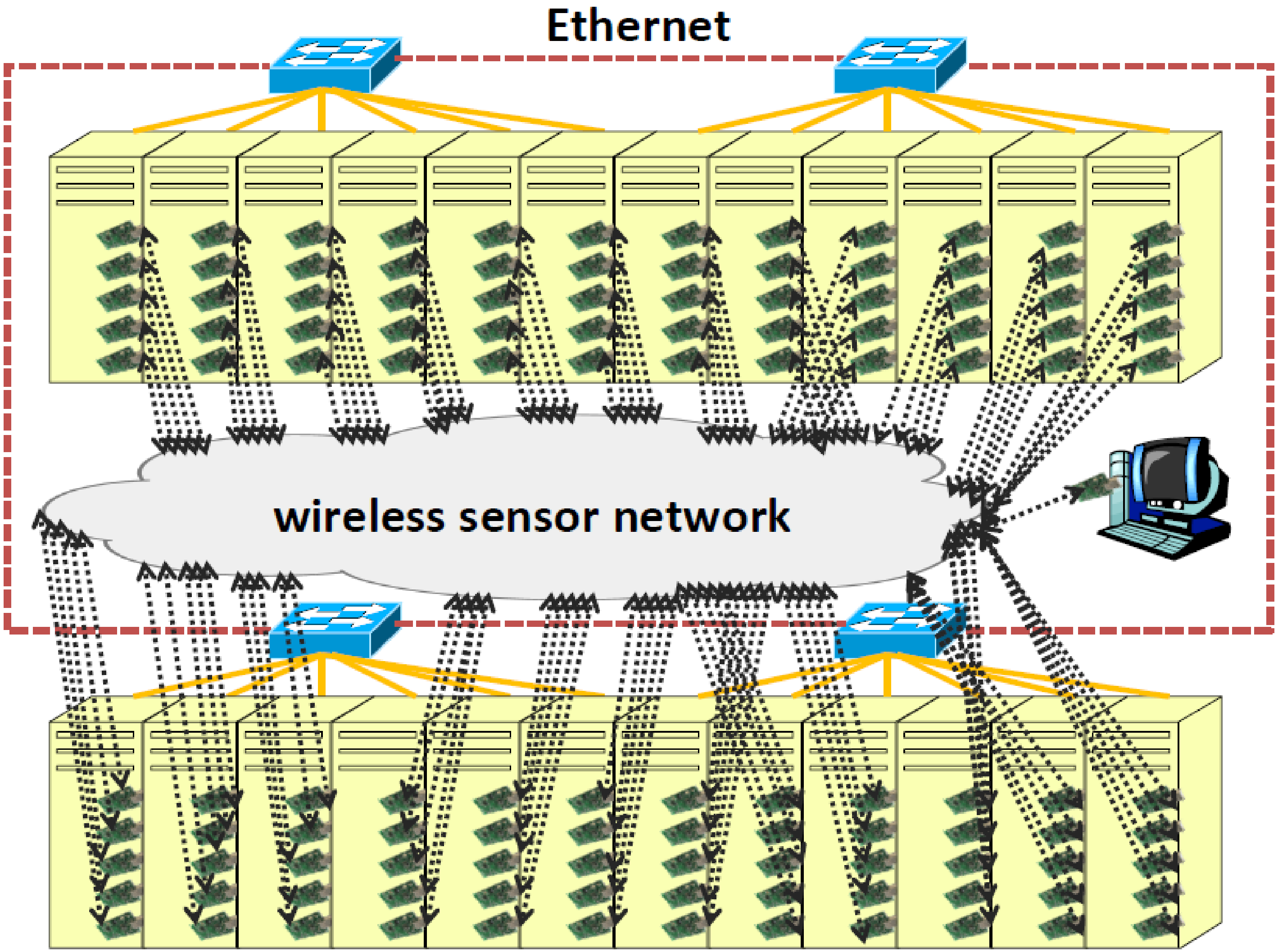




Cluster-Area Sensor Network

- CASN as a complementary solution
  - To improve the cluster management
  - To enhance the operational security
  - To improve the bandwidth performance of TCP
- CASN achieves
  - Cluster-wide command dissemination
  - Verification of server’s physical presence
  - Wireless traffic signaling for TCP

CASN Architecture

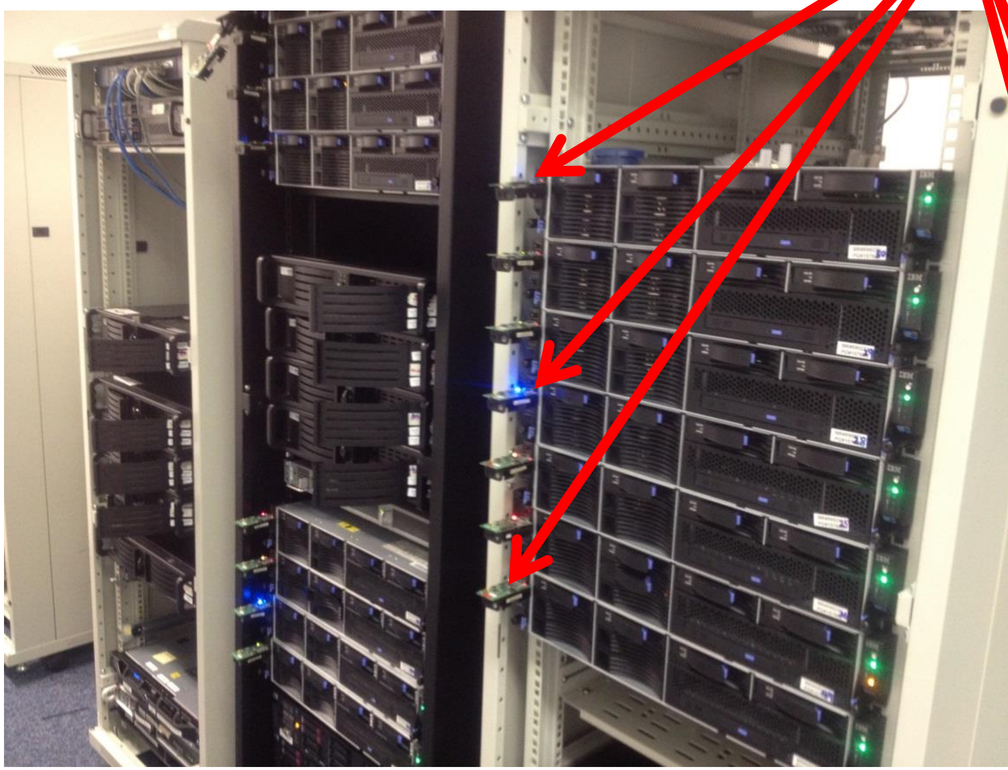


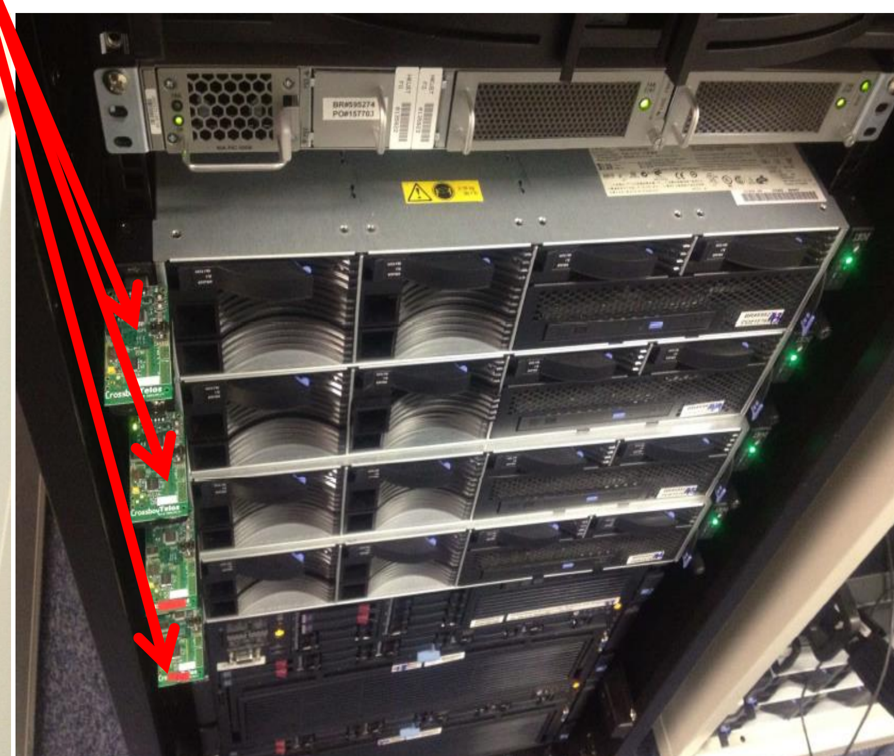
Prototype Implementation

Telos B motes attached to servers via USB interfaces

Prototype Implementation

Telos B motes attached to servers via USB interfaces





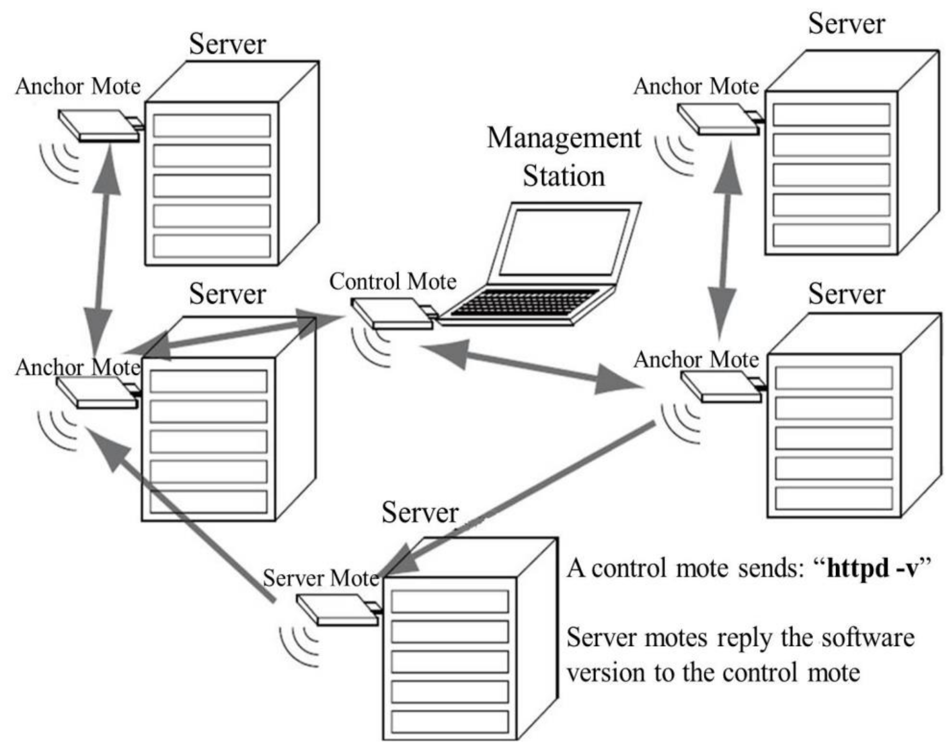
Cluster-Area Sensor Network

- Cluster-wide command dissemination
- Verification of server's physical presence
- Wireless traffic signaling

Command Dissemination

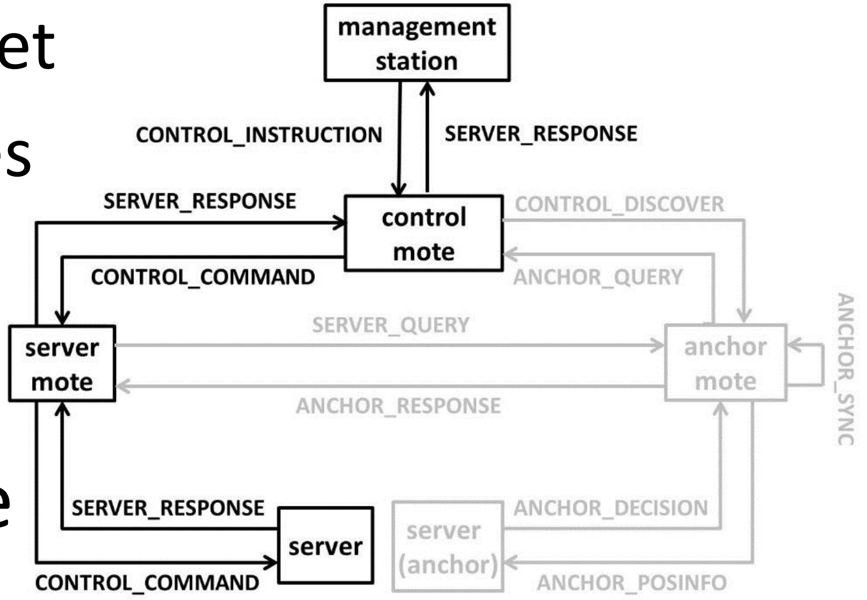
- System components
  - Wireless motes
  - Compute servers

- Three types of motes
  - Control motes
  - Anchor motes
  - Server motes



Command Dissemination

- Workflow of command dissemination
  - Issued from the management station
  - Forwarded to the control mote
  - Broadcasted via sensornet
  - Received by server motes
  - Executed on servers
- Command-line interface



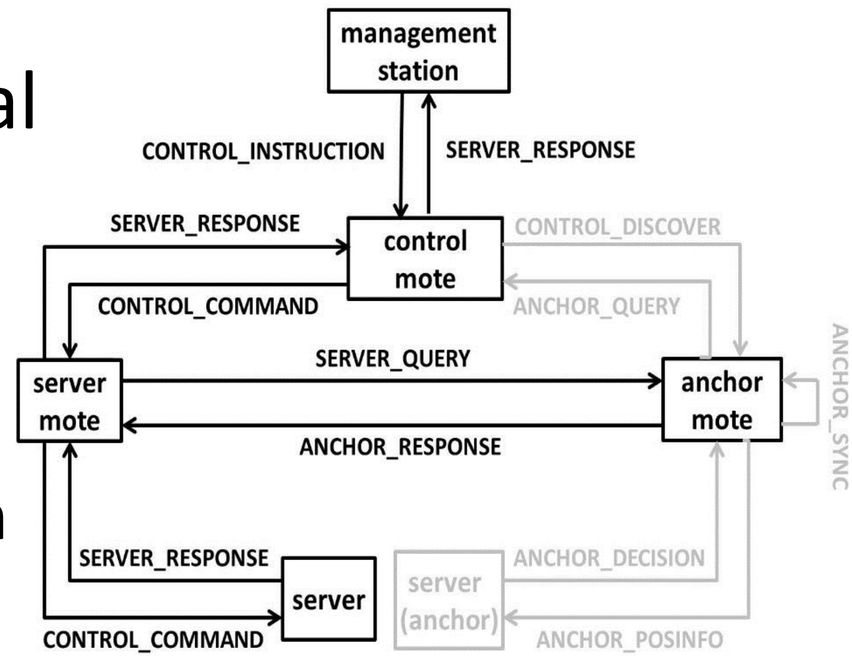
Cluster-Area Sensor Network

- Cluster-wide command dissemination
- Verification of server’s physical presence
- Wireless traffic signaling

Verification of Physical Presence

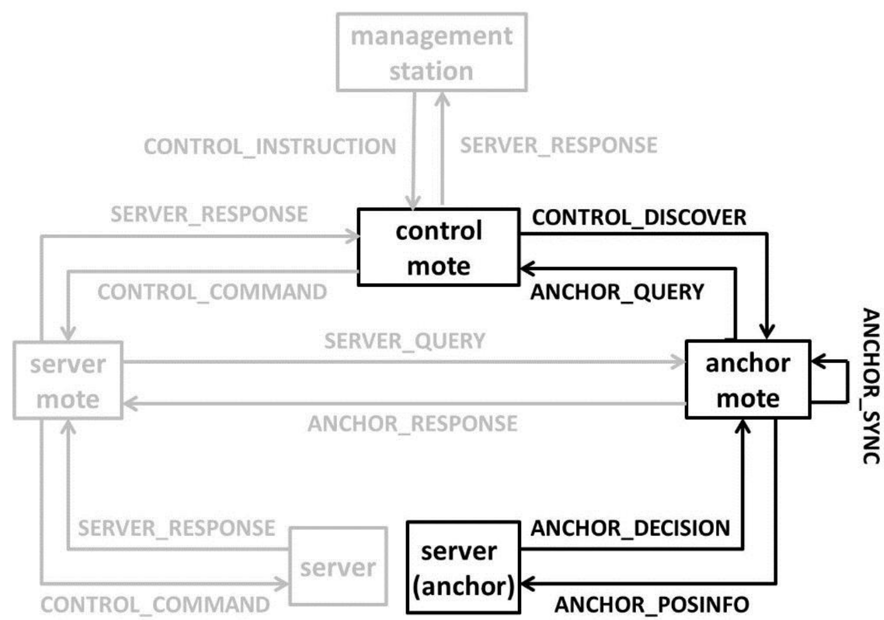
- Operations in data center are yet to be secure
  - Example: impersonating the management station

- Example: verify the physical location of a control mote
  - Before execution, server motes query anchor motes for the legitimacy of certain control mote



Localizing Control Motes

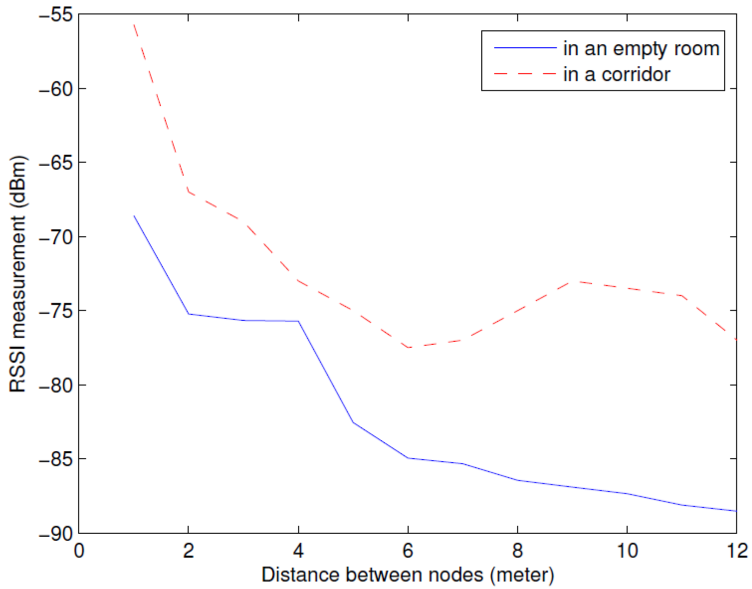
- Workflow of physical localization
  - Passive discovery: anchor motes periodically query the location of control motes
  - Active discovery: control mote initiates discovery upon its arrival
  - Anchor motes together localize a control mote to determine its legitimacy
- Suffice with 4 anchors



Radio-based Localization

- Coarse-grained radio-based localization
  - Suffice even at 5-meter precision
  - Inefficacy of RSSI-based ranging approach

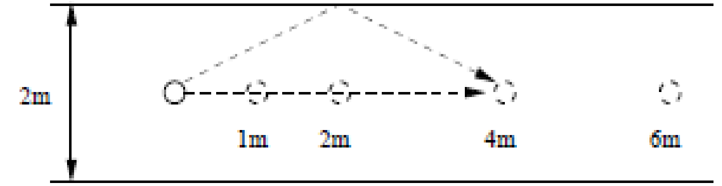
$$P(d) = P(d_0) - 10n \log\left(\frac{d}{d_0}\right)$$



- Necessity for empirical RSSI modeling in a data center environment

Empirical Localization Model

- Cope with the multipath effect by considering indirect signals



$$P(d) = P(d_0) - 10n\log\left(\frac{\sum_{i=1}^{k} r_i d_i}{d_0}\right)$$

  - $\boldsymbol{R}$ = $[r_1\ r_2\ \ldots\ r_k]$ as the amplitude coefficients of signal components
  - $\boldsymbol{D}$ = $[d_1\ d_2\ \ldots\ d_k]$ as discretized distances of signal components

- Rician distribution used to model amplitudes of indirect signals

$$R(x|\gamma,\sigma) = \frac{x}{\sigma} e^{\frac{-(x^2+\sigma^2)}{2\sigma^2}} I_0\left(\frac{x\gamma}{\sigma^2}\right)$$

Probabilistic Ranging

- Solving ***R*** in $\boldsymbol{R} * \boldsymbol{D} = d_0 * 10^{\frac{P(d_0)-P(d)}{10n}}$
  - Consider only the 5 shortest reflected signals
  - d_{AB} as the distance between the transmitter A and receiver B (i.e. 2 meters)

$$r_i = \begin{cases} 0 & if\ d_i < d_{AB}\ or\ d_i - d_{AB} \geq 2 \\ 1 & if\ d_i = d_{AB} \\ a_i * R(d_i - d_{AB}) & if\ d_i > d_{AB}\ and\ d_i - d_{AB} < 2 \end{cases}$$

- Localization: after obtaining the probabilistic ranging results, compute the most plausible location using trilateration

Localization Delay & Accuracy

- To evaluate the localization delay and accuracy inside a 4m x 4m square field
  - Localization error $e = \sqrt{(x' - x)^2 + (y' - y)^2}$
- Results
  - Overall delay: 8-12 seconds
  - Acceptable with 30-sec localizing period
  - 88% of localization errors within 5 meters
  - Errors for positions inside the square within 2 meters

Cluster-Area Sensor Network

- Cluster-wide command dissemination
- Verification of server’s physical presence
- Wireless traffic signaling

Wireless Signaling

- Wirelessly Assisted TCP (WSTCP)
  – Goal 1: high bandwidth utilization
  – Goal 2: responsive to fair bandwidth sharing

- To achieve coordinated traffic control among multiple hosts on TCP
  – Coordinating co-flows’ traffic transmission
  – Co-flows: congestion-coupled active flows

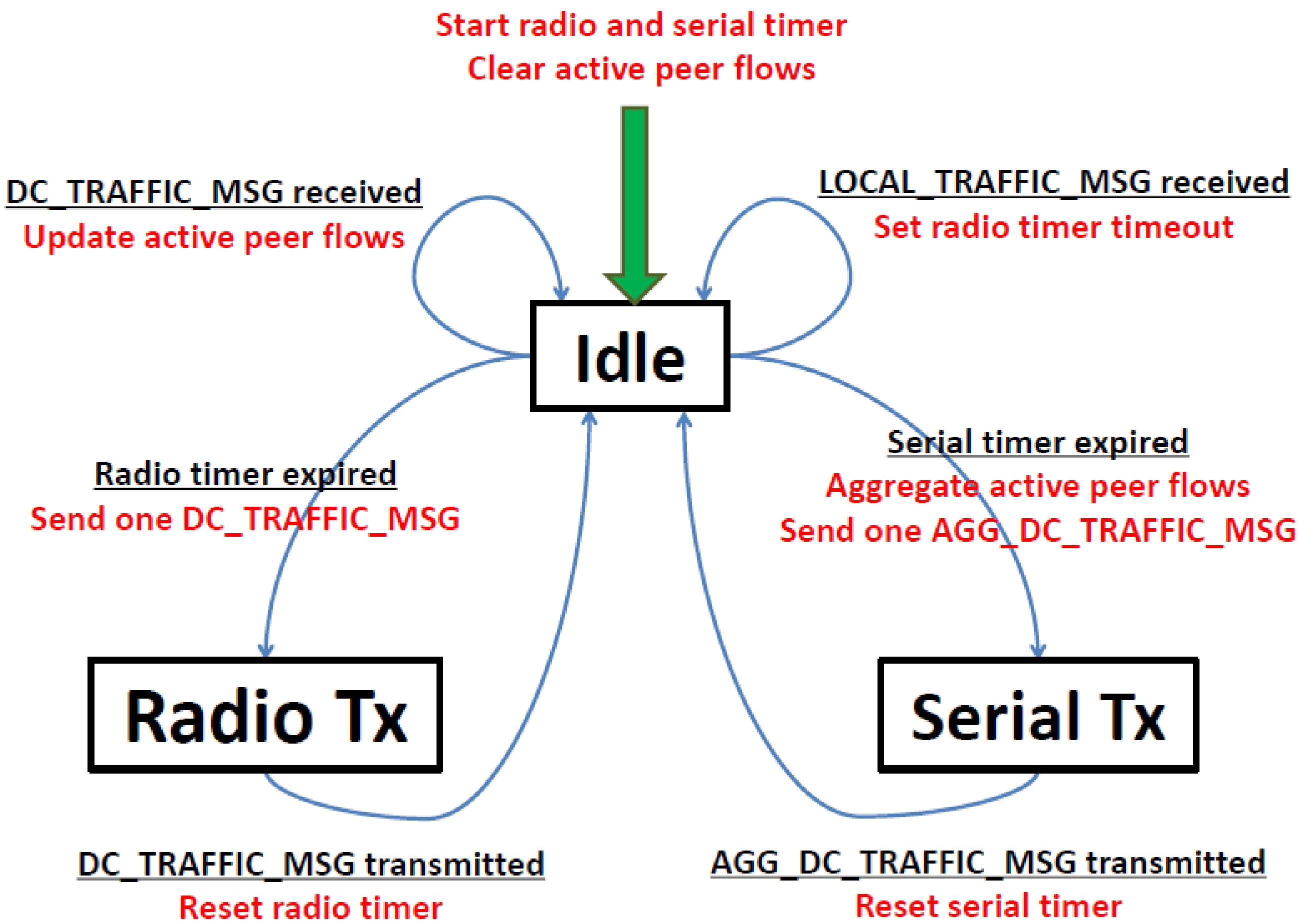
General Approach

- Per-flow traffic detection and signaling
  - Classify a set of active flows $T(f_i) = \begin{cases} 1 & \text{if } sock\_rate(f_i) > \epsilon \\ 0 & \text{otherwise} \end{cases}$
  - Broadcast their traffic information to sensornet

- Congestion coupling of active flows
  - Identify a coupling set S_{CAF}^{i} for each active flow f_i
  - Compute an aggregate congestion level for each active flow $A(f_i, TF) = \sum_{f_k \in (F - \{f_i\})} [T(f_k) * h(f_k, f_i)]$
  - Apply the aggregate congestion level to tune *cwnd*

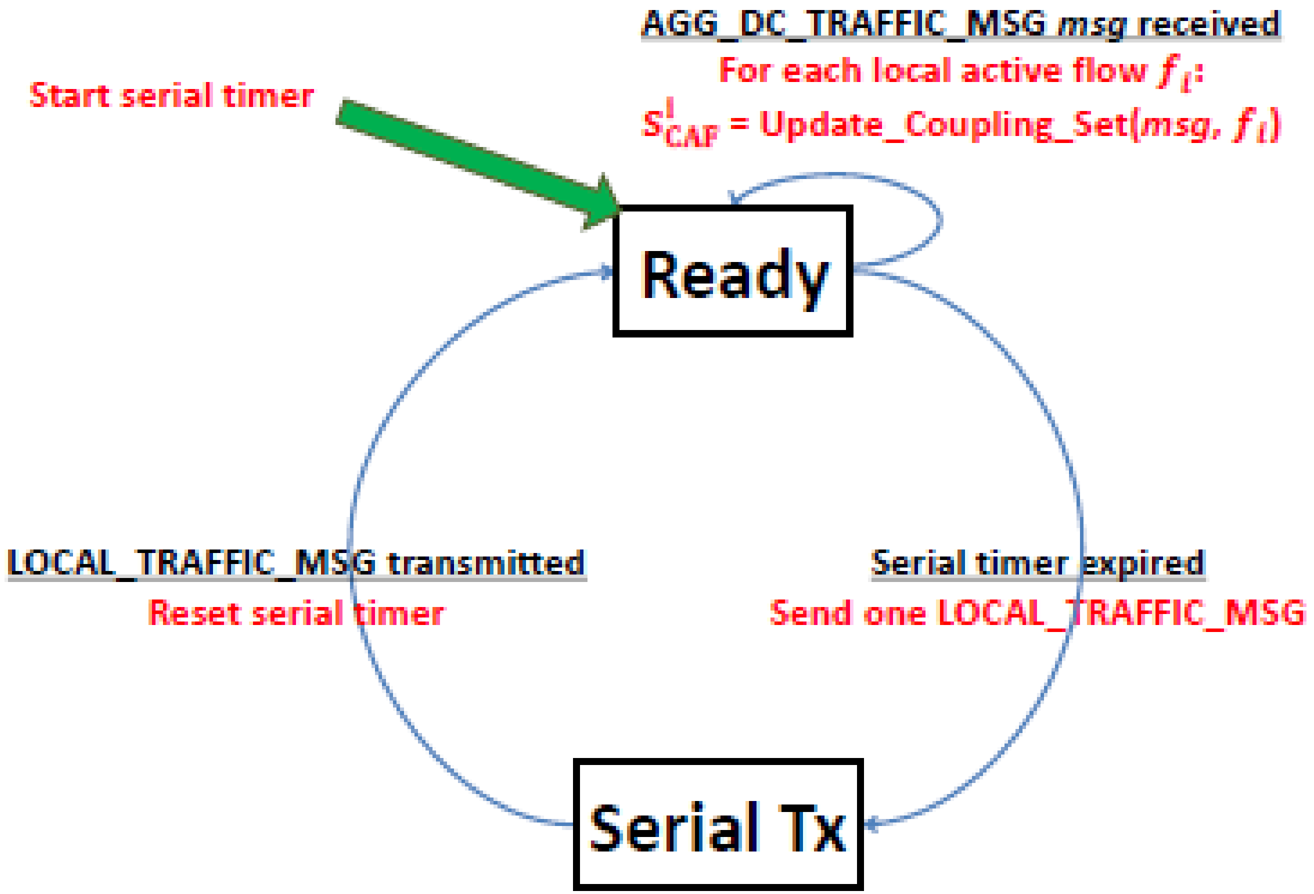
Active Flow Signaling

Sensornet controller



Active Flow Signaling

Ethernet controller



Identify Coupling Set

Having received an AGG_DC_TRAFFIC_MSG (i.e. *msg*), identify a congestion coupling set for each active flow in *msg*

Input: $AGG\_DC\_TRAFFIC\_MSG$ *msg*

$S^i_{ACF} = \{f_i\}$

for all active flow f': $src' -> dest'$ in msg **do**

  if $corr(f_i, f') == 1$ **then**

    $S^i_{ACF} = S^i_{ACF} \cup f'$

  end if

end for

Output S^i_{ACF}

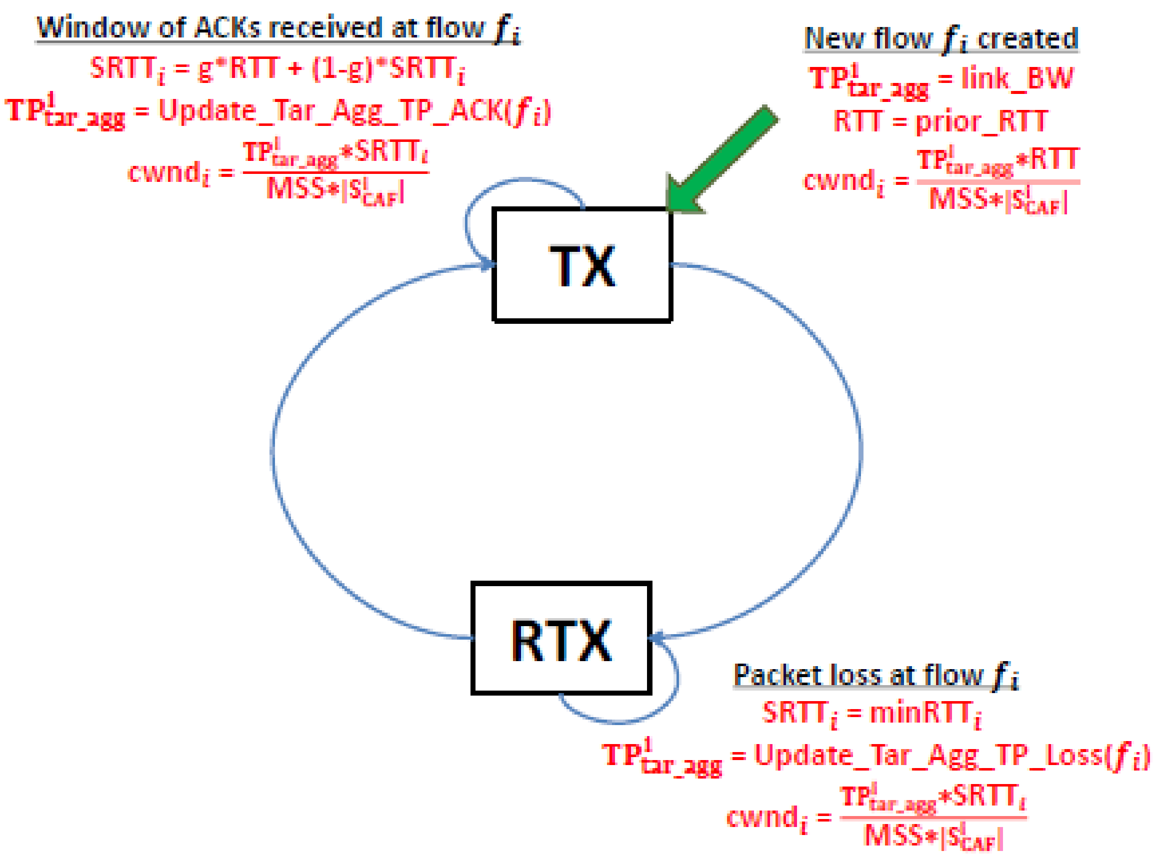
Congestion Coupling

- Given a pair of congestion-coupled flows, their throughput normally affects each other if they coexist

- Throughput-based profiling
  - L_a, L_b: a pair of end-to-end links
  - P_a, P_b: throughput of a single flow on L_a and then L_b
  - P_a', P_b': throughput of 2 concurrent flows on L_a, L_b
  - L_a and L_b are congestion-coupled if $(P_a' + P_b') < \beta(P_a + P_b)$, where $0.8 \leq \beta < 1$

Congestion Control

Update *cwnd* upon ACKs and loss



Shared Link Estimation

Update target aggregate throughput upon ACKs

$$TP^i_{act\_agg} = \frac{TP^i_{head} + TP^i_{tail}}{2} * |S^i_{ACF}|$$

if $TP^i_{tar\_agg} < TP^i_{act\_agg}$ **then**

$\quad TP^i_{tar\_agg} = TP^i_{act\_agg}$

else if $TP^i_{tar\_agg} - TP^i_{act\_agg} <$ D **then**

$\quad TP^i_{tar\_agg} = TP^i_{tar\_agg} + \frac{TP^i_{head}}{TP^i_{act}} * |S^i_{ACF}|$

else

$\quad TP^i_{tar\_agg} = TP^i_{tar\_agg} - |S^i_{ACF}| * F$

end if

Output $TP^i_{tar\_agg}$

Shared Link Estimation

Update target aggregate throughput upon loss

$$prevTP^i_{tar\_agg} = TP^i_{tar\_agg}$$

$$newTP^i_{tar\_agg} = \frac{TP^i_{tar\_agg} + prevTP^i_{tar\_agg} * (|S^i_{ACF}| - 1)}{|S^i_{ACF}|}$$

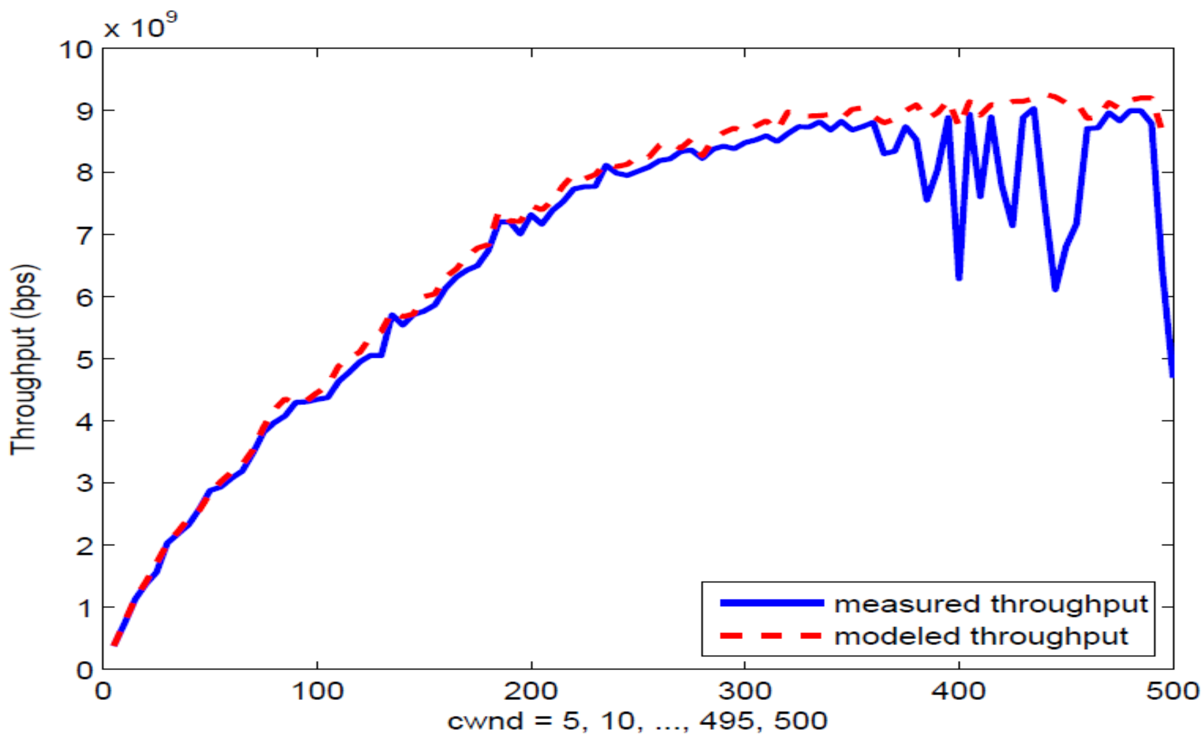
$$TP^i_{tar\_agg} = \min(newTP^i_{tar\_agg}, TP^i_{tar\_agg} - \frac{TP^i_{act}}{TP^i_{head}} C)$$

Output $TP^i_{tar\_agg}$

Tuning CWND

Following the bandwidth-delay product model

$$TP^i_{tar} = \frac{TP^i_{tar\_agg}}{|S^i_{ACF}|} \Rightarrow cwnd_i = \frac{TP^i_{tar} * RTT_i}{MSS}$$



Active Peer Flow

- Sensor signaling may not be reliable
  - May miss some co-flow elements
  - Smaller coupling set overestimates *cwnd*
- Moving average does not help
  - Aggressively increasing aggregate congestion level
  - Conservatively decreasing aggregate congestion level

$$|S^i_{ACF}|'_t = max^k_{j=0}\{|S^i_{ACF}|_{t-j}\}$$

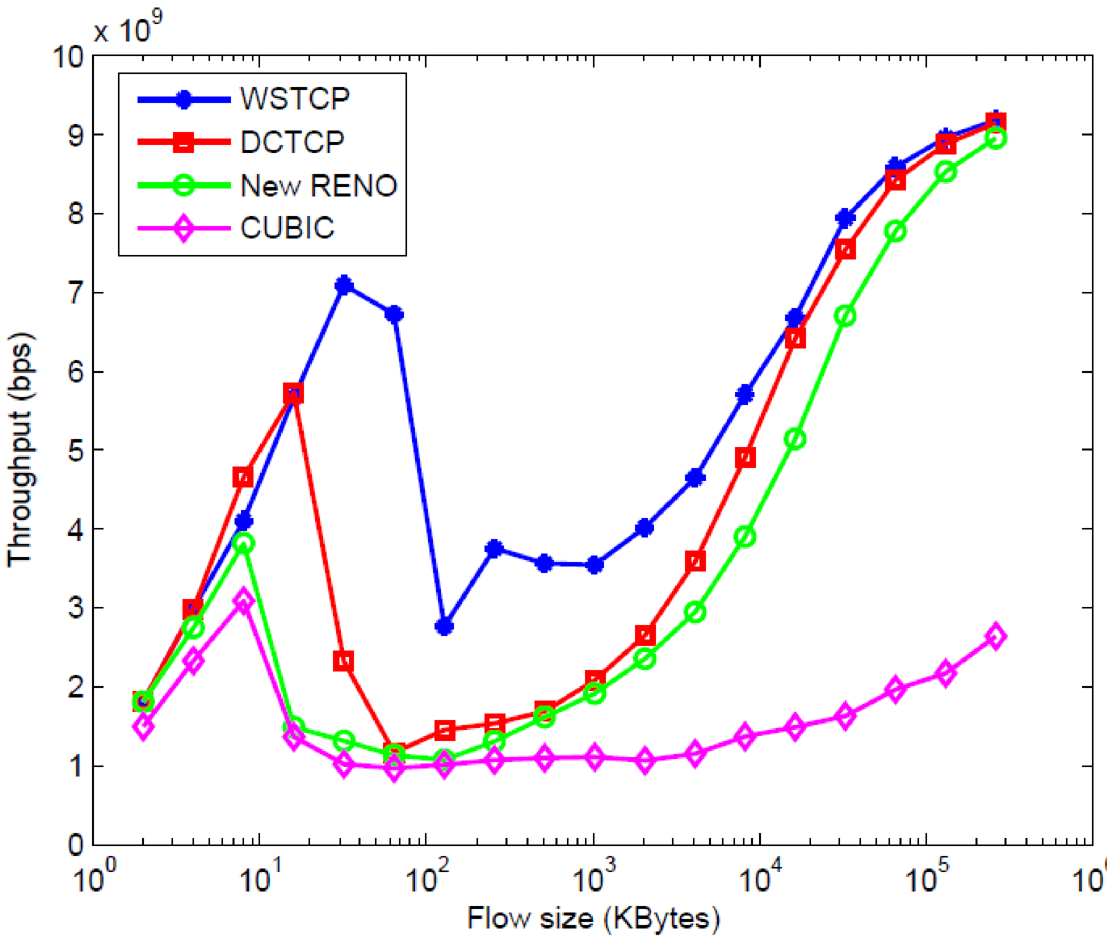
Flow Control

- Sending rate bounded by *min(cwnd, swnd)*
  - Initial *swnd* normally set as several MSSs
  - Low bandwidth utilization even with large *cwnd*
  - Short flows take multiple RTTs to complete

- Modify flow control based on traffic signaling
  - Initial swnd enlarged (< 64 KB) given idle traffic signals from sensornet

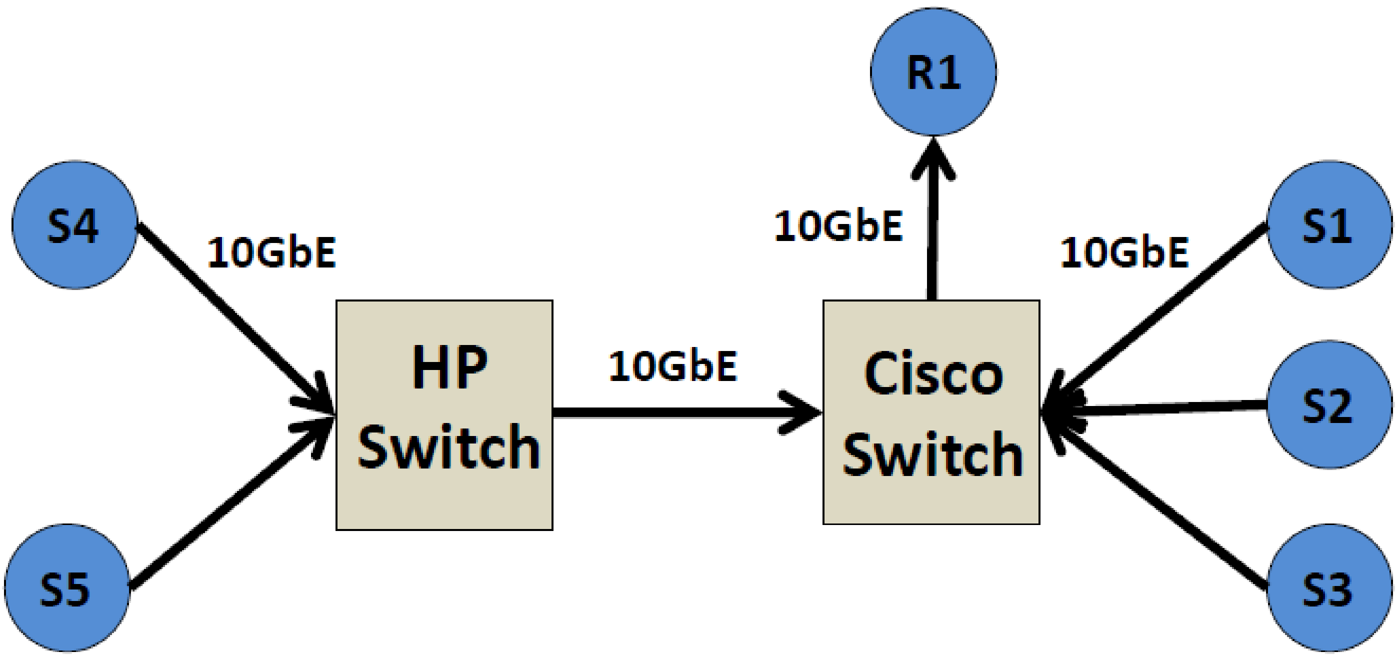
Bandwidth Utilization

One-to-one short flows (2KB~256MB)



Bandwidth Sharing

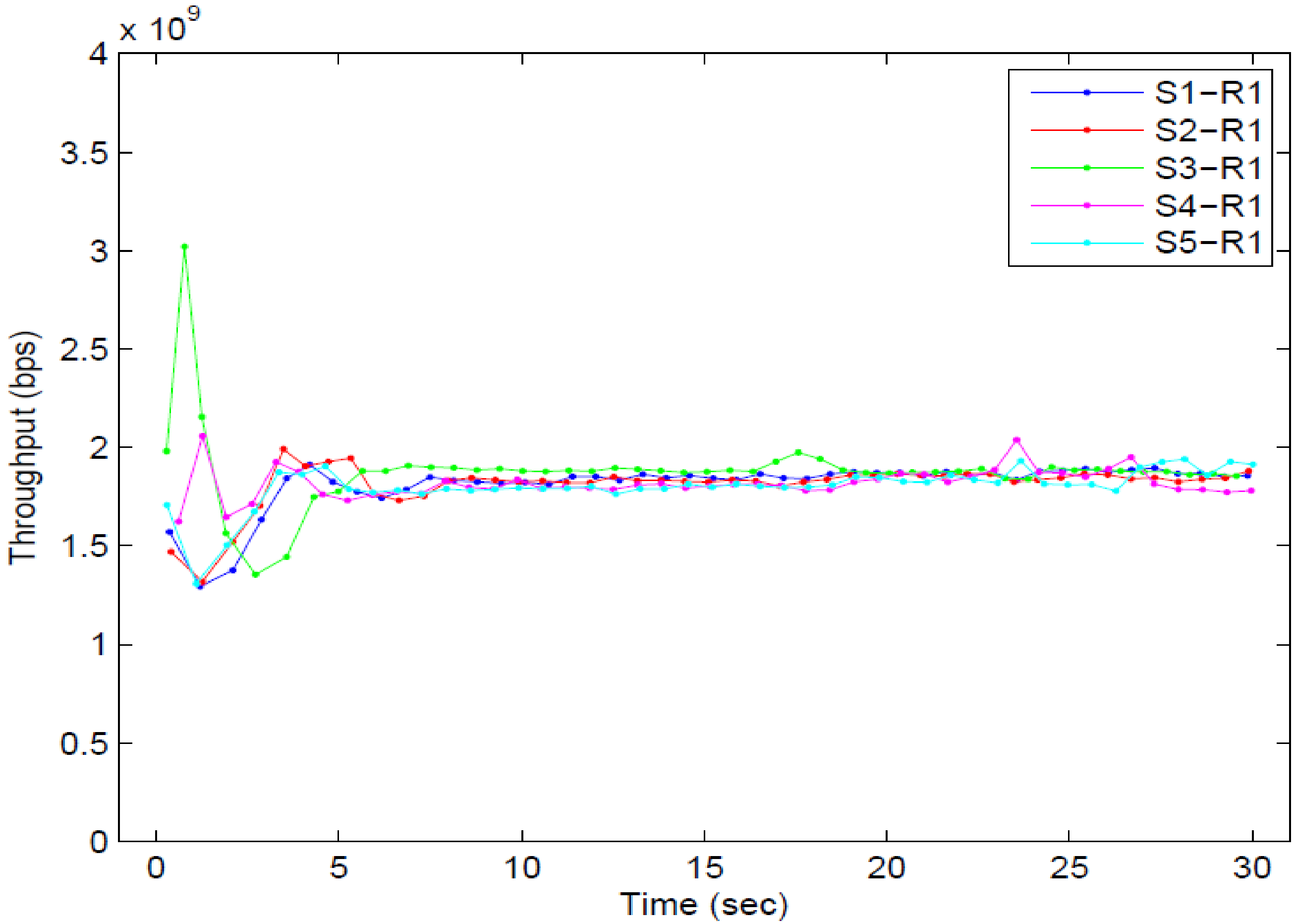
Many-to-one synchronous long-lived flows



Each of five senders S1, S2, S3, S4, S5 send one flow to receiver R1.
The two flows from S4, S5 are called cross-switch flows.
The three flows from S1, S3, S3 are called intra-switch flows.

Bandwidth Sharing

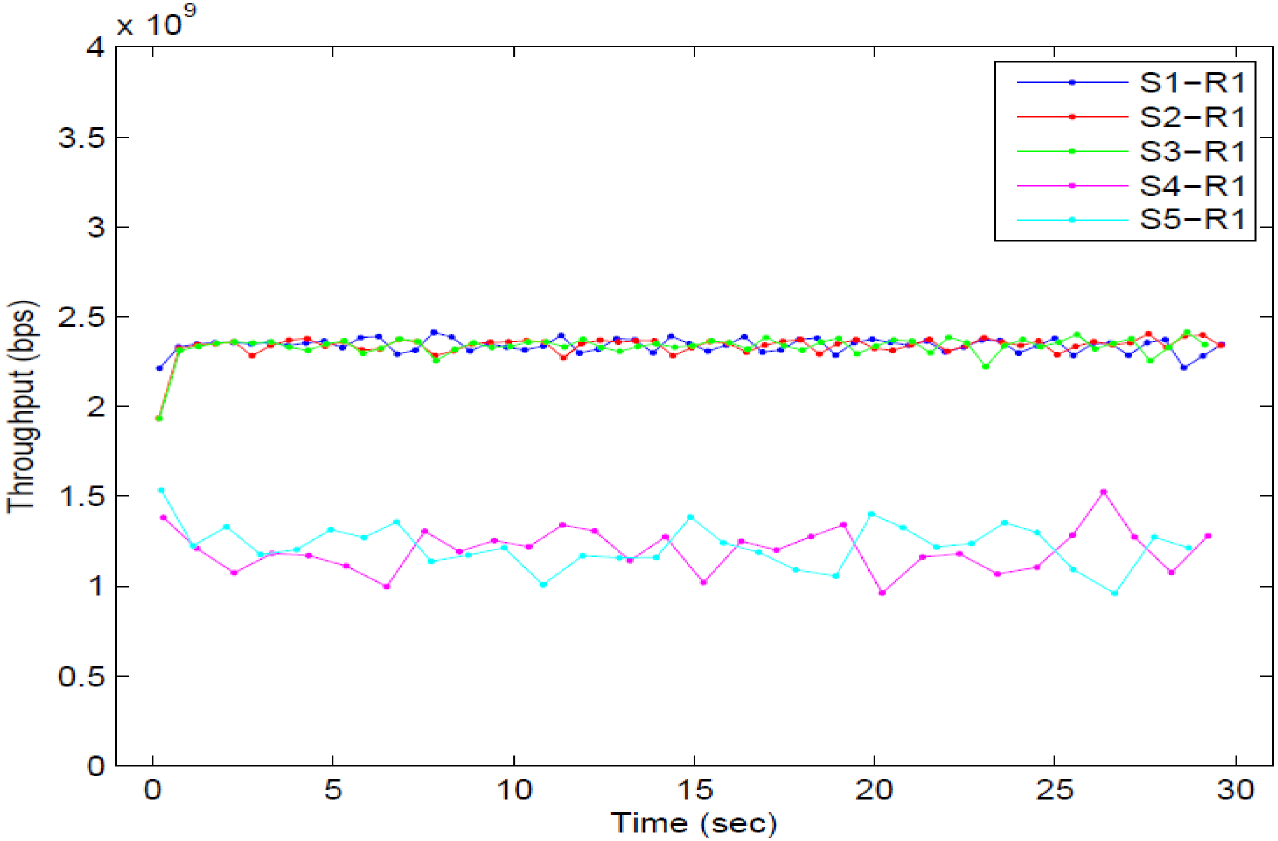
Many-to-one synchronous long-lived flows



Bandwidth Sharing

Many-to-one synchronous long-lived flows

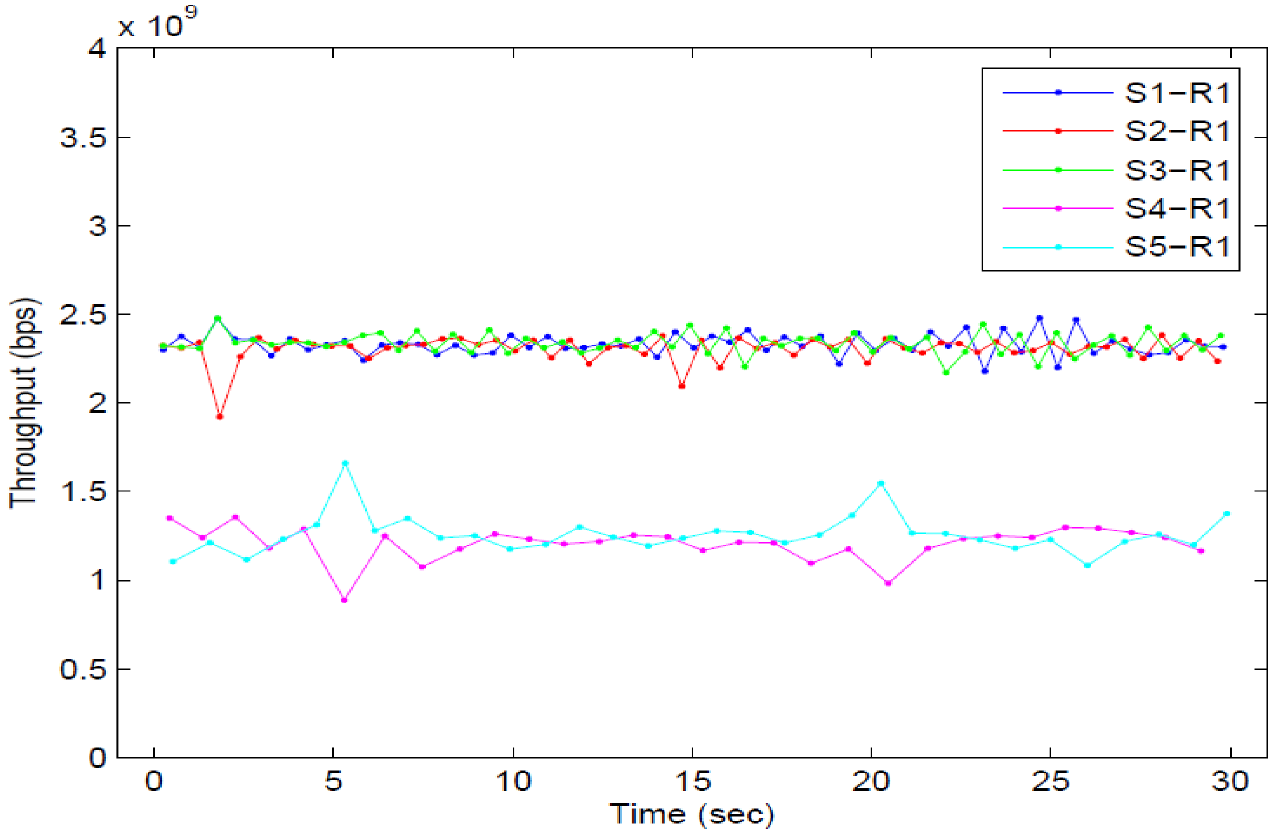
- Compared to CUBIC



Bandwidth Sharing

Many-to-one synchronous long-lived flows

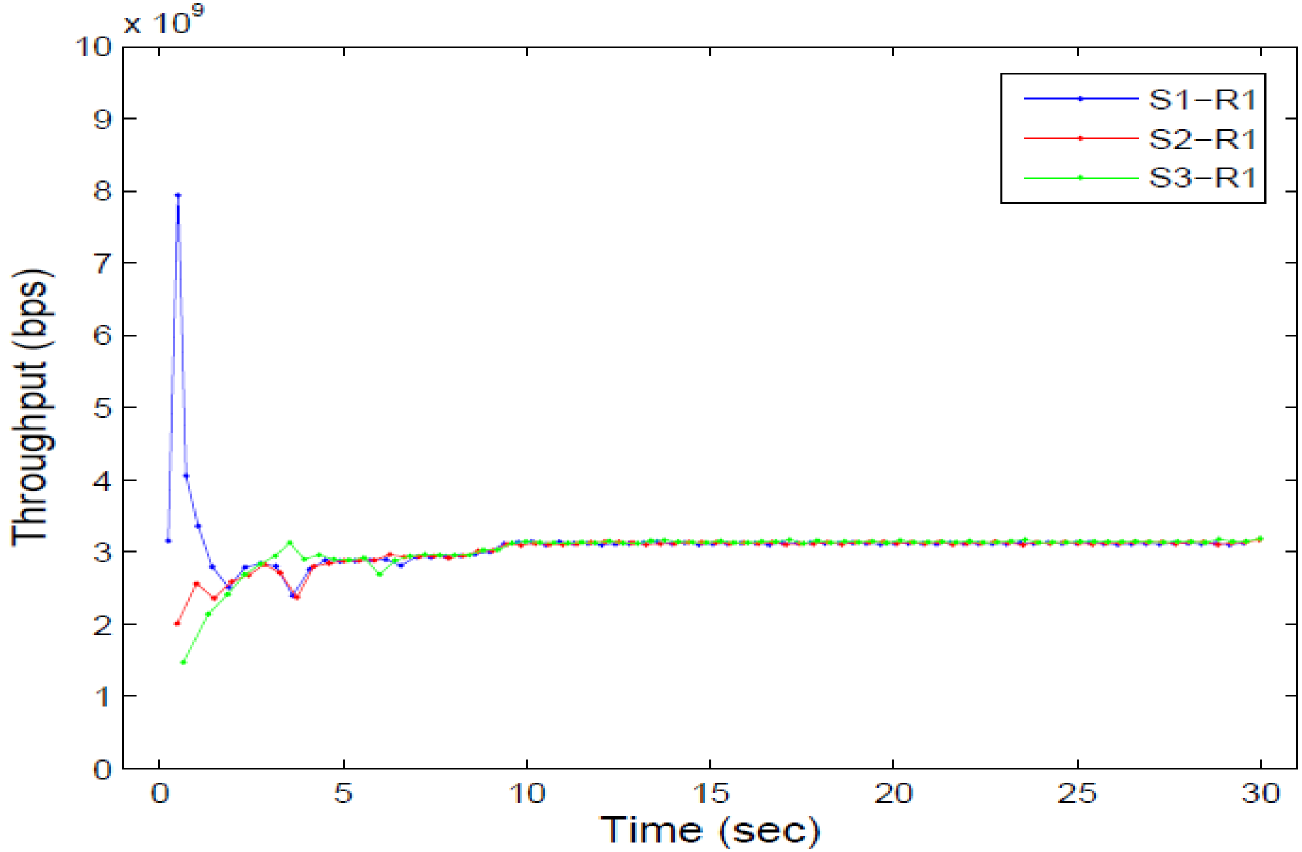
– Compared to New Reno



WSTCP vs. ECN Control

3-to-1 synchronous long-lived flows

– Compared to ECN-based DCTCP



Summary

- We design and implement CASN -- a cluster-area sensor network
  - Wireless cluster-wide command dissemination
  - Verification of server's physical presence
  - Networking signaling for TCP

- Future work
  - CASN with fingerprint-based localization
  - More sophisticated congestion coupling approach

Reference

- J. Moore, J. Chase, P. Ranganathan, and R. Sharma, “Making scheduling "Cool": temperature-aware workload placement in data centers,” in USENIX ATC ’05.
- C.-J. M. Liang, J. Liu, L. Luo, A. Terzis, and F. Zhao, “RACNet: a high-fidelity data center sensing network,” in ACM SenSys ’09.
- L. Li, C.-J. M. Liang, J. Liu, S. Nath, A. Terzis, and C. Faloutsos, “Thermocast: a cyber-physical forecasting model for datacenters,” in ACM SIGKDD ’11.
- S. Ha, I. Rhee, and L. Xu, “CUBIC: a new TCP-friendly high-speed TCP variant,” SIGOPS Operating System Review, vol. 42, pp. 64–74, July 2008.
- M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, “Data center TCP (DCTCP),” in SIGCOMM ’10.
- K. Hong, S. Yang, Z. Ma, and L. Gu, “A Synergy of the Wireless Sensor Network and the Data Center System,” in IEEE MASS 2013.
- K. Hong, Z. Ma, and L. Gu, “Wirelessly Assisted TCP for High-Bandwidth Data Center Networks”, submitted to ACM MobiSys ’14.

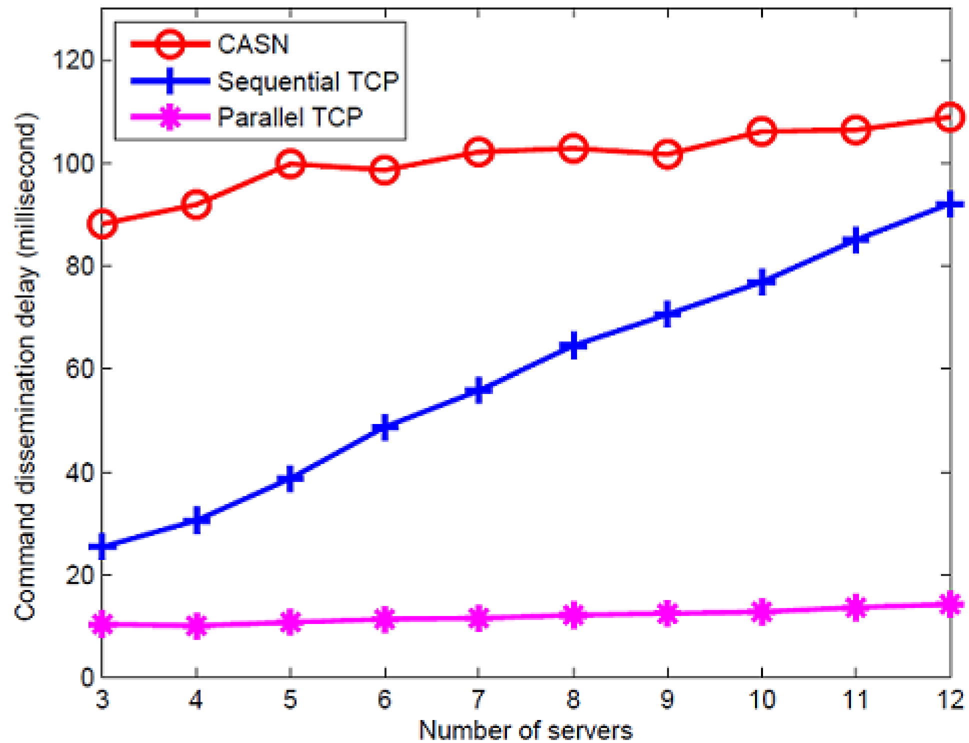
Q&A

Thank You!

Ke Hong
khongaa@ust.hk

Command Dissemination Delay

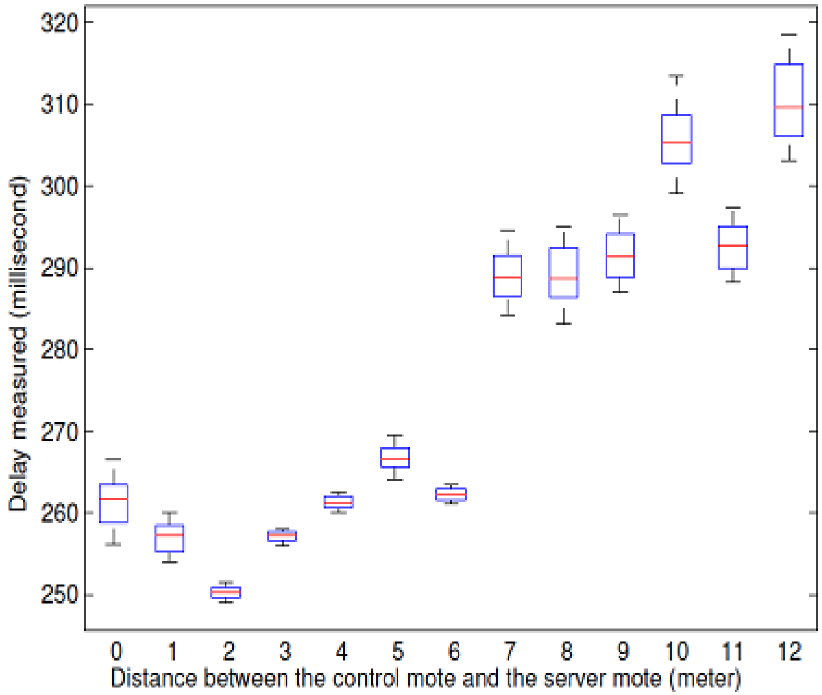
- To evaluate the round-trip delay of command dissemination to a number of servers across three racks
- Results
  - Scalable broadcast via sensornet
  - Stable delay



Reprogramming Delay of CASN

- Reprogramming delay: command dissemination delay + physical verification delay

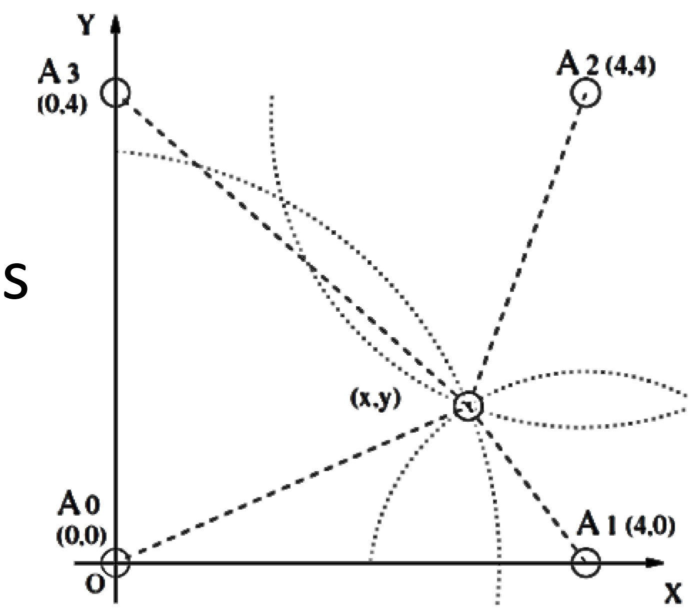
- Results
  - Less than 300 milliseconds with distance closer than 10 meters
  - Low enough for effective command dissemination



Reduce Computation Cost

- Computationally costly for all possible cases
  - In total K^H cases for *H* RSSI measurements per transmission, give that each maps to *K* **R**s

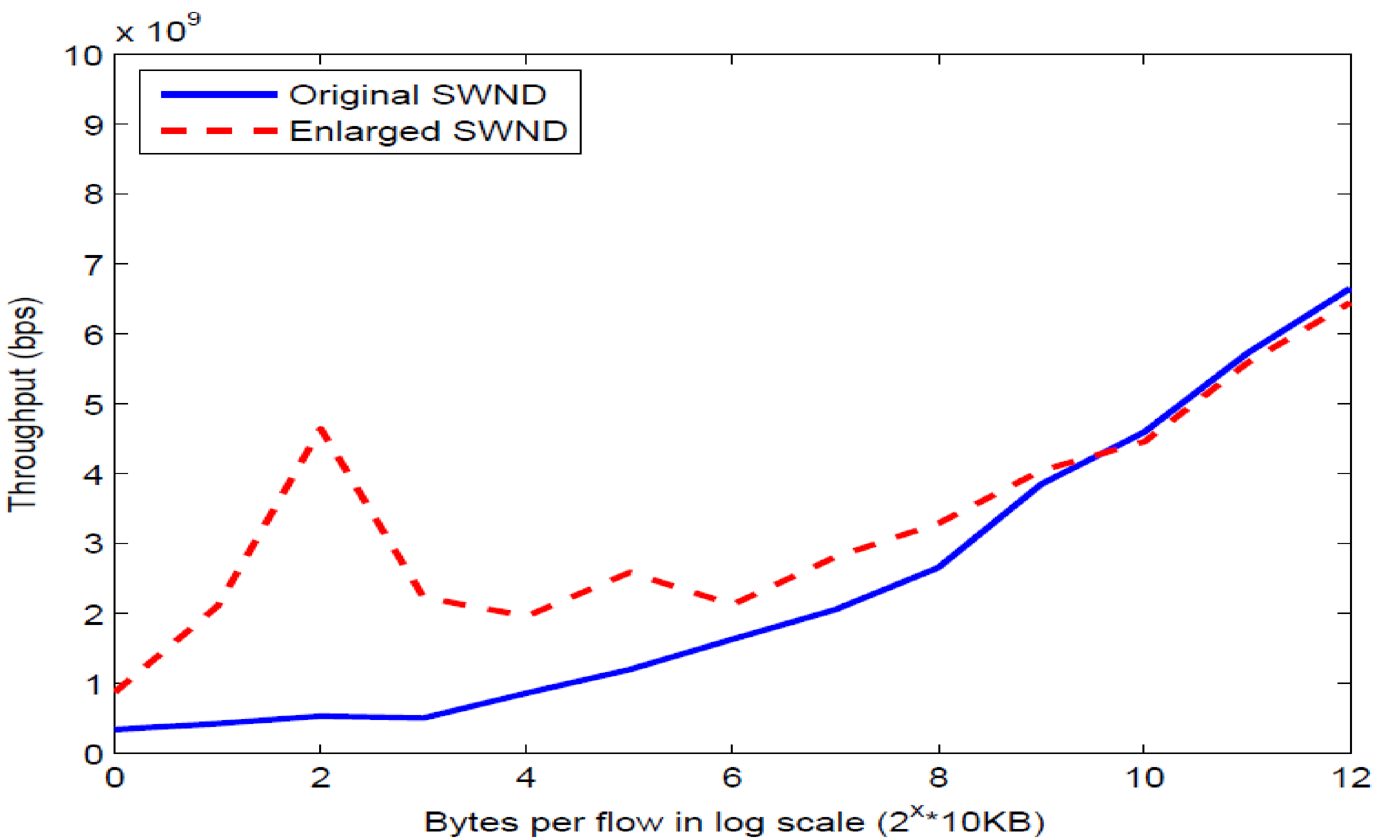
- Reduce computation cost by
  - Narrowing down distances by applying geometric constraints
  - Utilizing the known distances between anchors



Flow Control

Validating new flow control

– One single RTT for flows of size < 64 KB

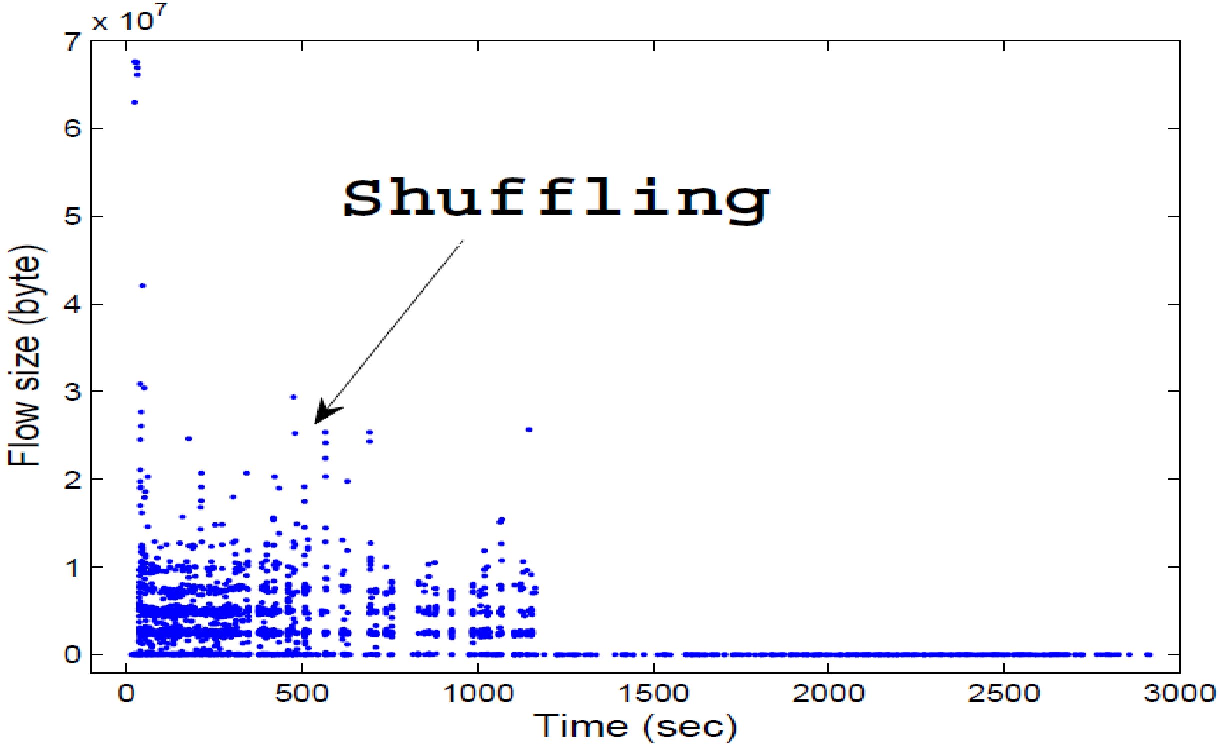


Hadoop Trace Experiment

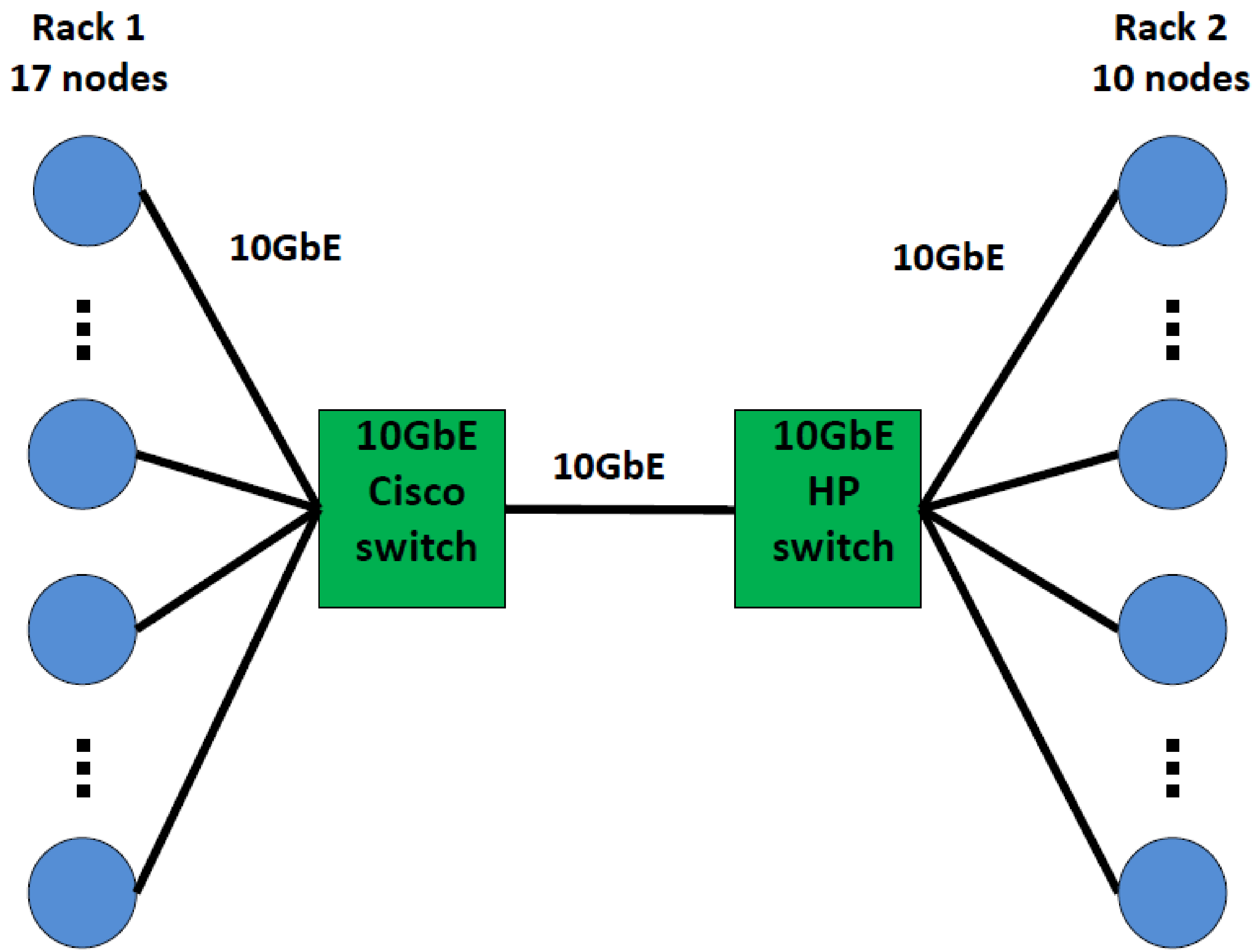
Hadoop sorting trace of 8, 16, 27 servers

– Extracted by *tcpdump*

– Replay in memory to avoid slow disk I/O



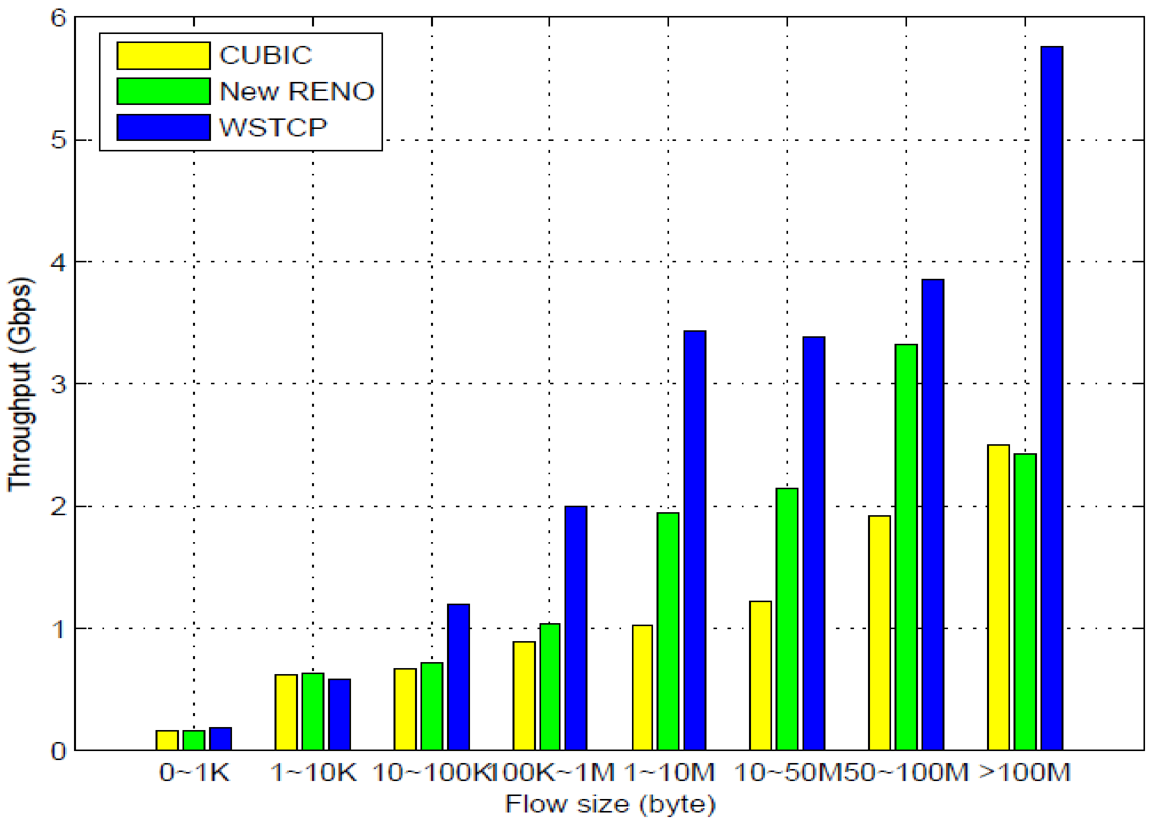
Testbed Topology



Hadoop Trace Experiment

Hadoop trace experiment of 16 servers

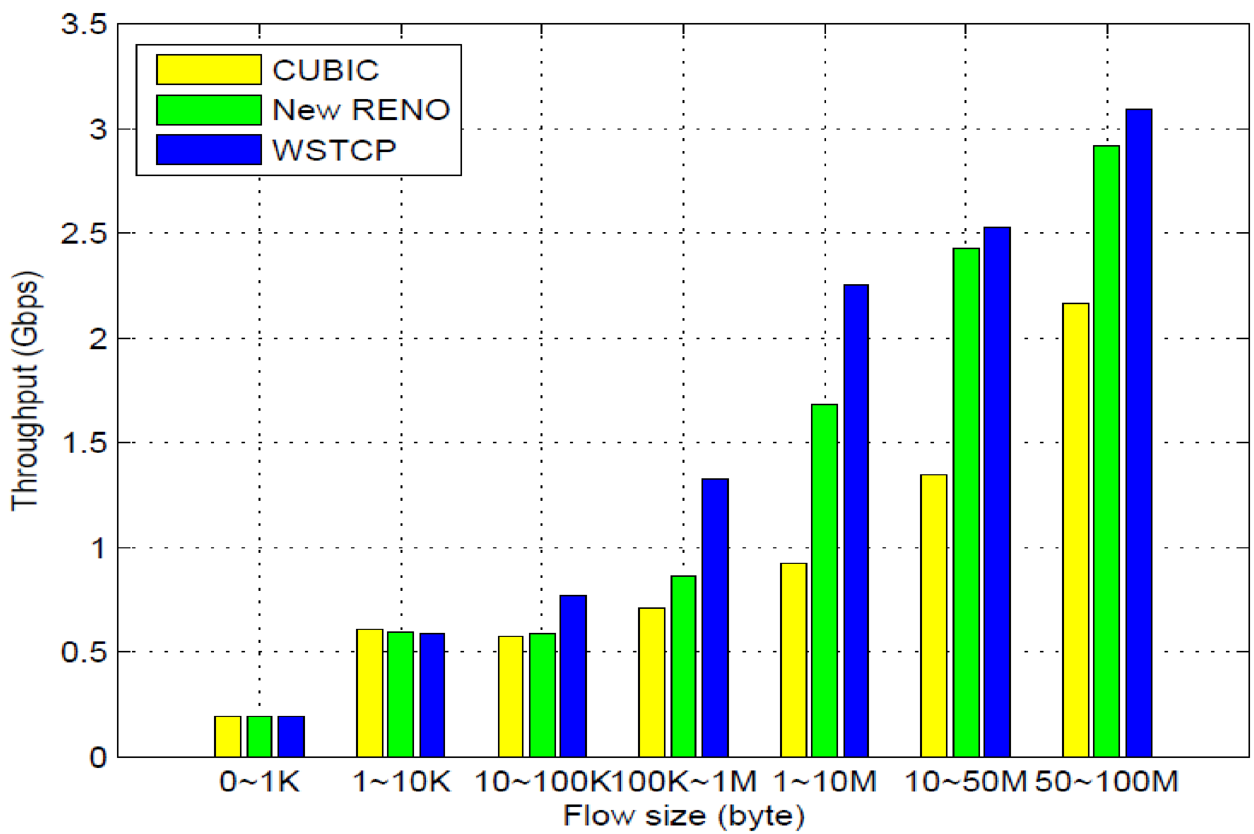
– Average throughput: 0.859, 1.448, 2.453Gbps



Hadoop Trace Experiment

Hadoop sorting trace of 27 servers

– Average throughput: 0.819, 1.370, 1.794Gbps



Algorithm 1 Update active peer flows

Input: `DC_TRAFFIC_MSG` $rPkt$

id = $rPkt$.ID

F_{active} = set of active flows in $rPkt$.`DC_TRAFFIC_MSG`

for all active flows f in F_{active} **do**

  active_src_dest_pkt_cnt[id][f.dest]++

  if active_tput[id][f.dest] > 0 **then**

    act_tput[id][f.dest] = $\frac{act\_tput[id][f.dest]+f.tput}{2}$

  else

    act_tput[id][f.dest] = f.tput

  end if

  if active_src_dest_pkt_cnt[id][f.dest] $> \sigma$ **then**

    active_flow[id][f.dest] = 1

  end if

end for

Algorithm 2 Aggregate active peer flows

```
Input: set of servers S_node
Initialize AggDCMsg with empty active peer flows
for all i in S_node do
    for all j in S_node do
        if active_flow[i][j] > 0 then
            Initialize an active peer flow f
            f.src = i
            f.dest = j
            f.tput = active_tput[i][j]
            Add f into AggDCMsg
        end if
    end for
end for
for all i in S_node do
    for all j in S_node do
        active_flow[i][j] = 0
        act_tput[i][j] = 0
        active_src_dest_pkt_cnt[i][j] = 0
    end for
end for
Output AggDCMsg
```