即時、可靠、有效率的物聯網架構

Real-Time, Reliable, Efficient Internet-of-Things Architecture

主講人: 王超
國立臺灣師範大學資訊工程學系 網宇實體系統實驗室
Networked Cyber-Physical Systems Laboratory
NATIONAL TAIWAN NORMAL UNIVERSITY

Outline

- Introducing Cyber-Physical Systems
- Need for Edge Computing
- CPS-IoT Edge Computing Requirements
- Real-Time, Reliable, Efficient CPS-IoT Architecture
- Concluding Remarks

Cyber-physical systems (CPS)

- A reasoned integration of computing systems to physical environments
  - Accessing physical constraints to ensure the computing system achieve its goal
  - Leveraging environmental information to improve operation efficiency
  - Augmenting physical environment to circumvent computing limitations

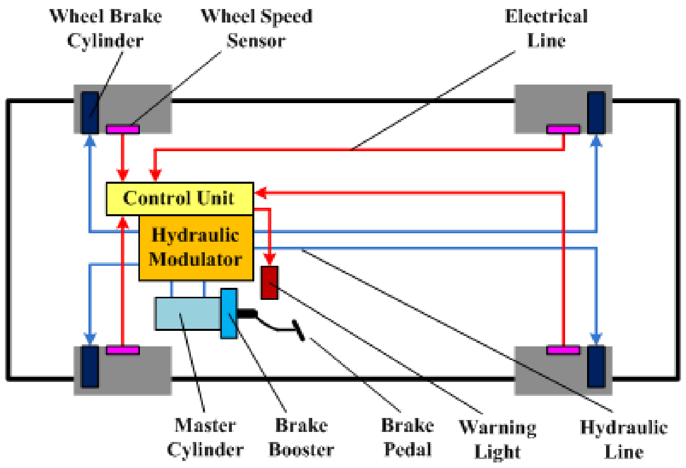
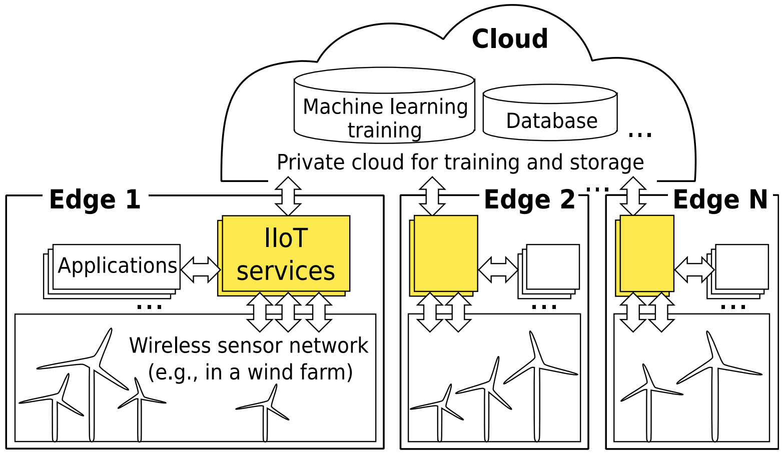


Image credit: https://cecas.clemson.edu/cvel/auto/systems/braking.html

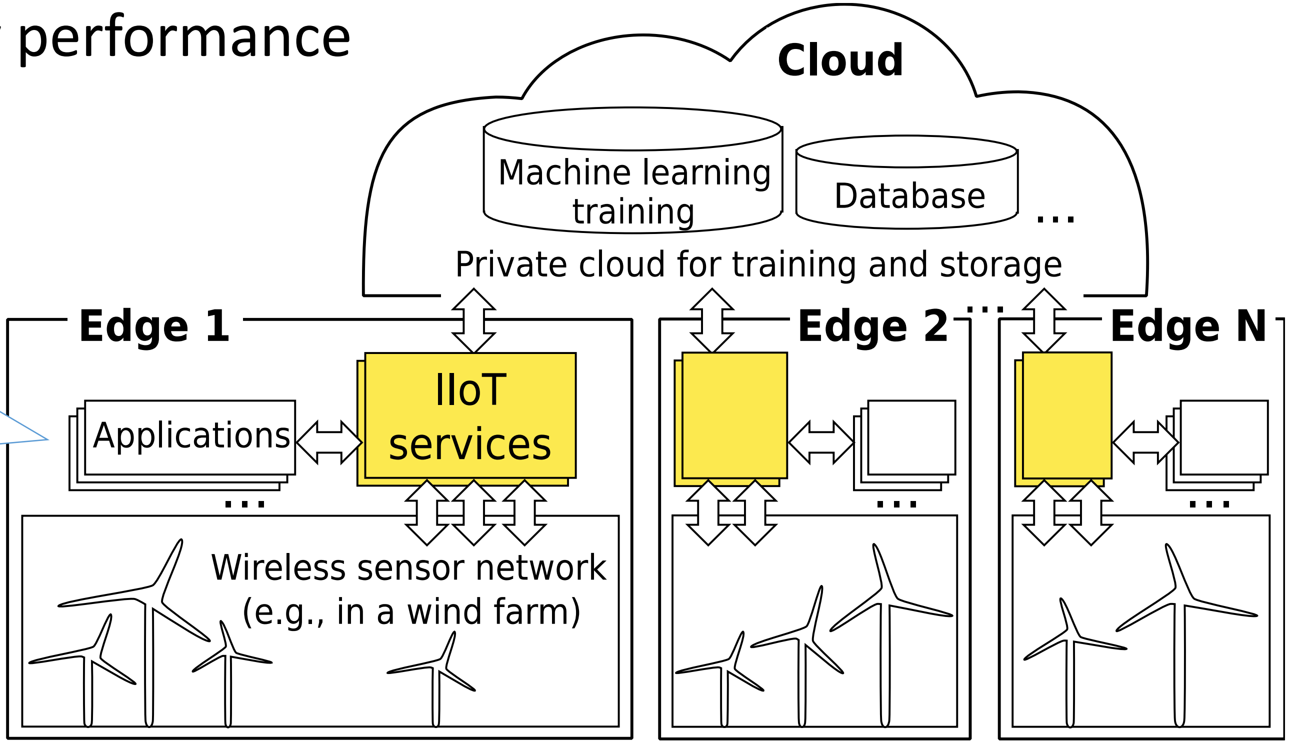


Copyright: Oleh Tokarev
https://www.123rf.com/photo_16981783_shopping-trolley-with-products.html

Industrial Internet of Things (IIoT)

- Synergizing sensing, analytics, and control
  - ✓Cloud computing for high capacity
  - ✓Edge computing for timely performance

Condition monitoring,
Emergency response,
Predictive maintenance,
…

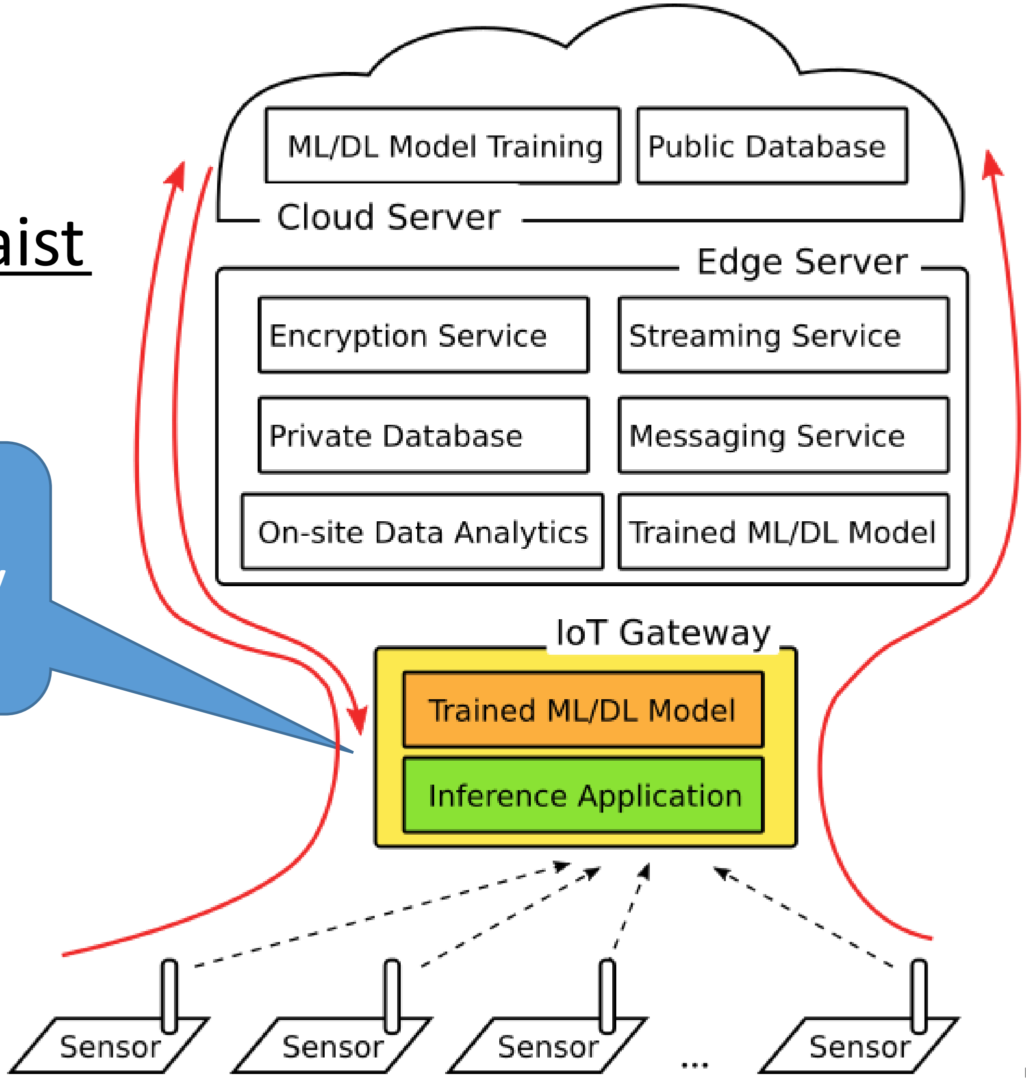


Challenges for IoT edge computing

- Reliable and timely computing
  at a resource-constraint network waist

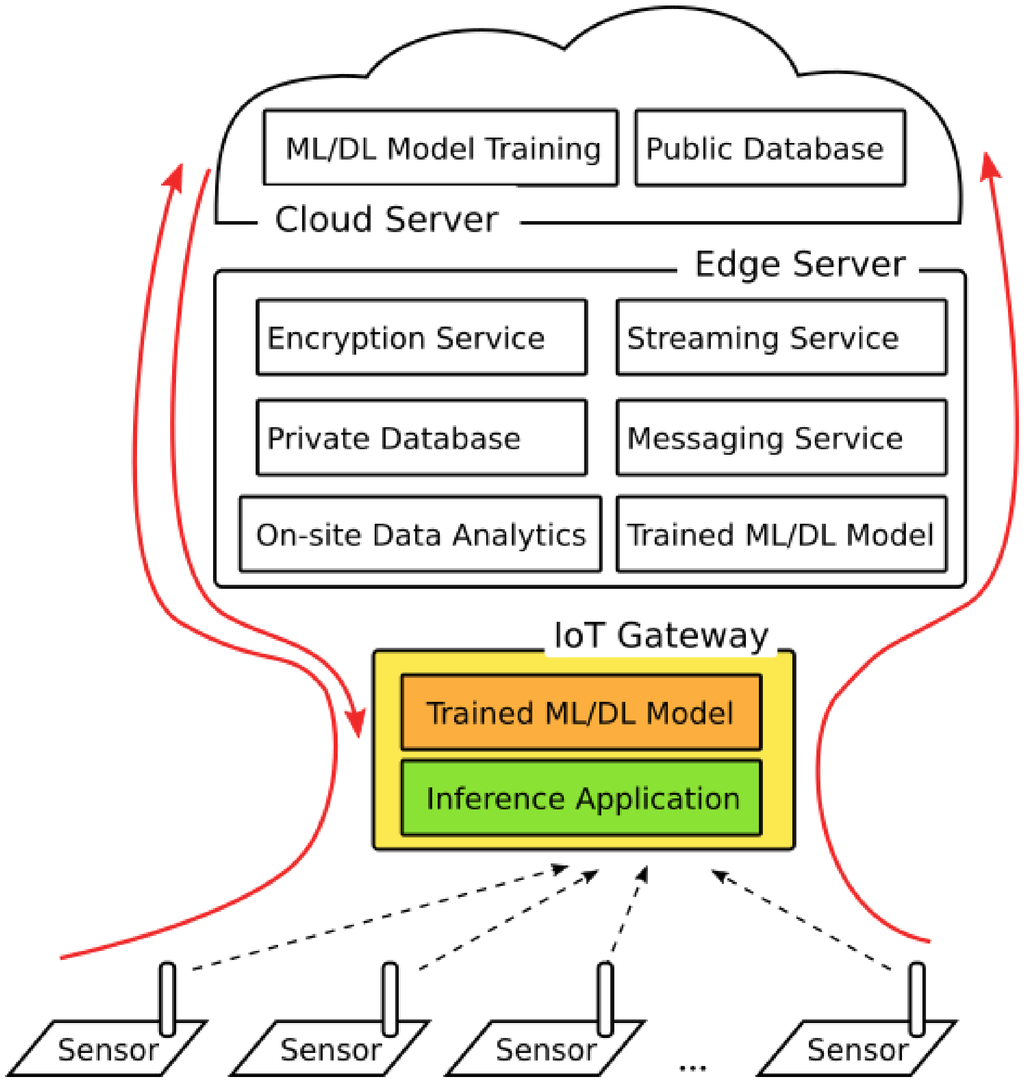
  for example, an IoT gateway

- Single point of failure
- Traffic congestion and delay
- Limited network bandwidth



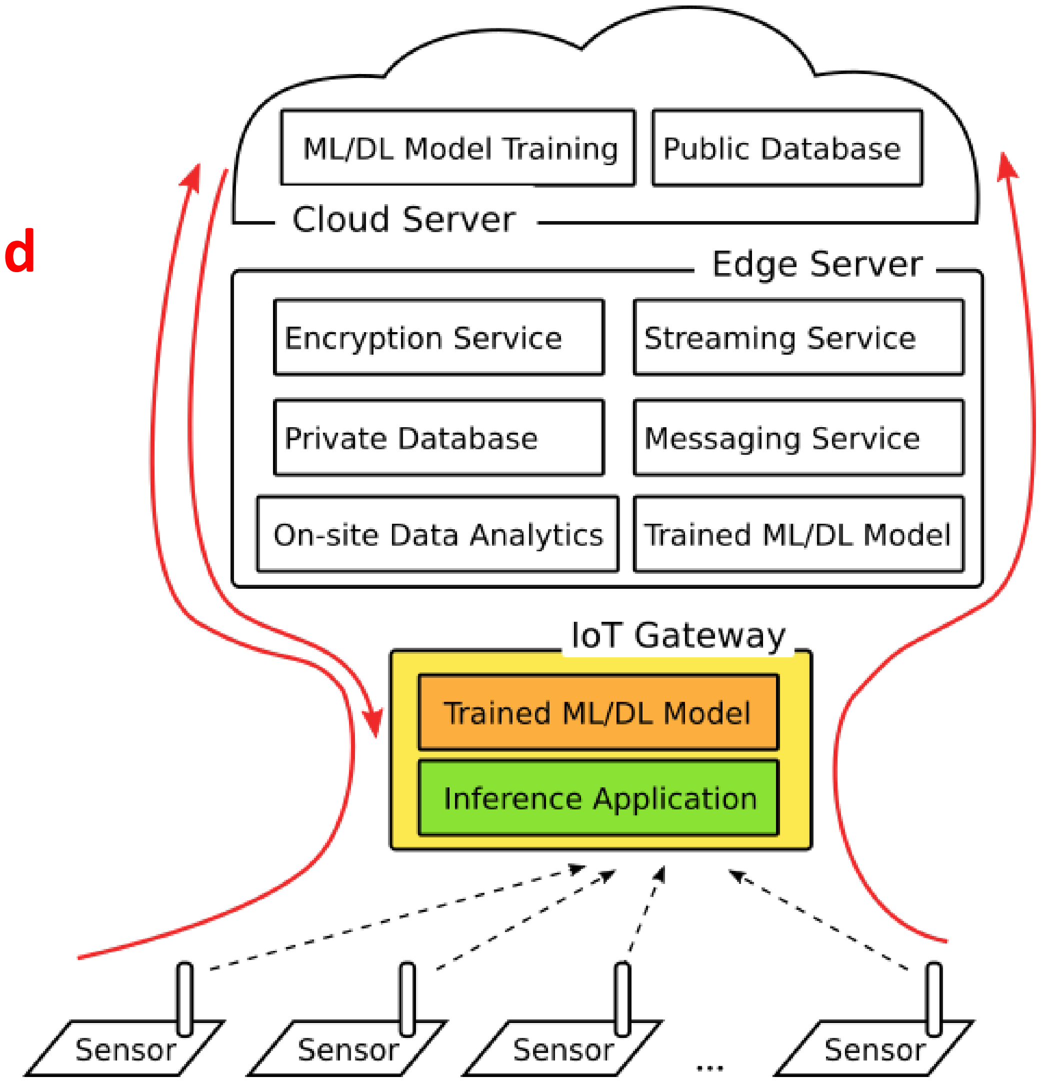
Specific IoT gateway requirements

- Real-time requirement
  - Data subscriber imposes an end-to-end soft deadline for data
- Fault-tolerance requirement
  - Data subscriber cannot accept some consecutive data losses
- Efficiency requirement
  - The gateway should not consume too much local network bandwidth



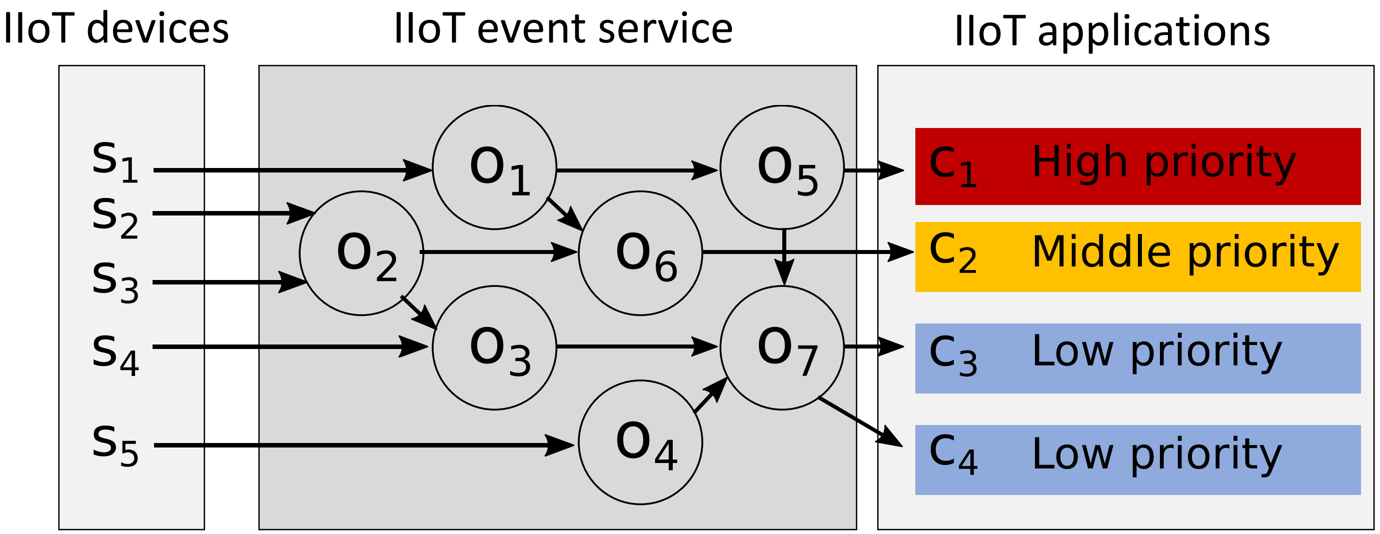
Specific IoT gateway requirements

- **Real-time requirement**
  - **Data subscriber imposes an end-to-end soft deadline for data**
- Fault-tolerance requirement
  - Data subscriber cannot accept some consecutive data losses
- Efficiency requirement
  - The gateway should not consume too much local network bandwidth



A model for CPS edge computing

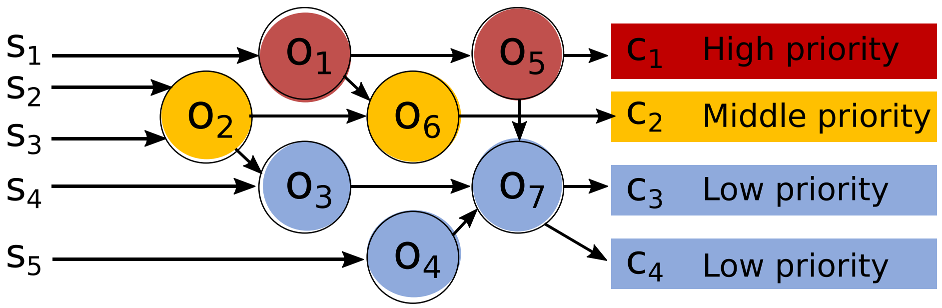
- Latency requirement: ***Absolute time consistency*** on an event's elapse time since creation



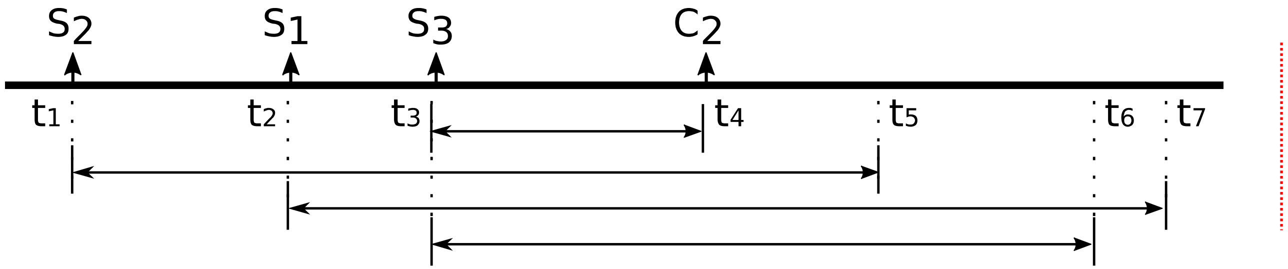
O_i: operations (filtering, transformation, encryption, …)

Real-time CPS event processing

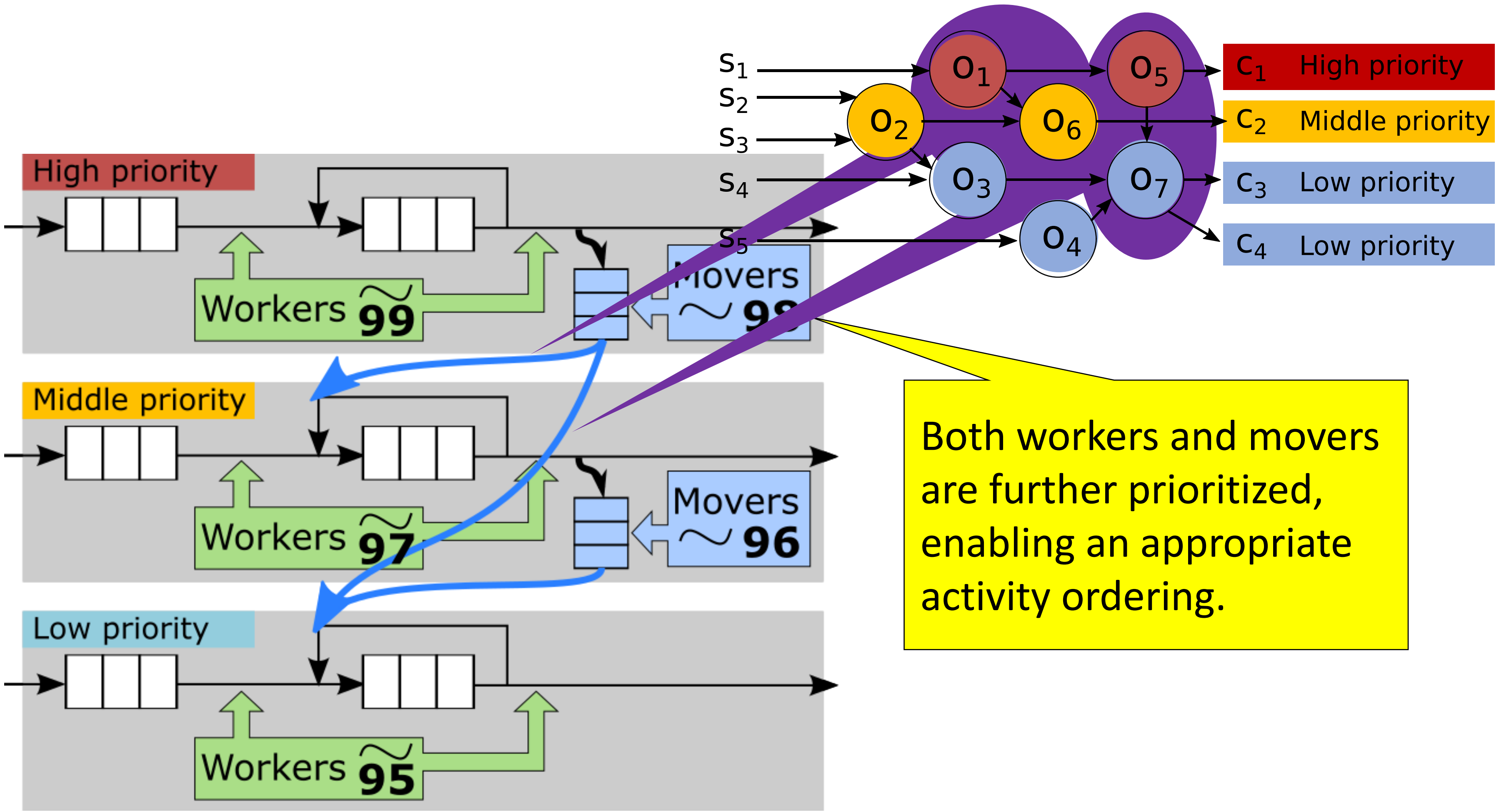
- Processing in the order of priorities propagated from application:



- Enforcing absolute time consistency



Processing architecture



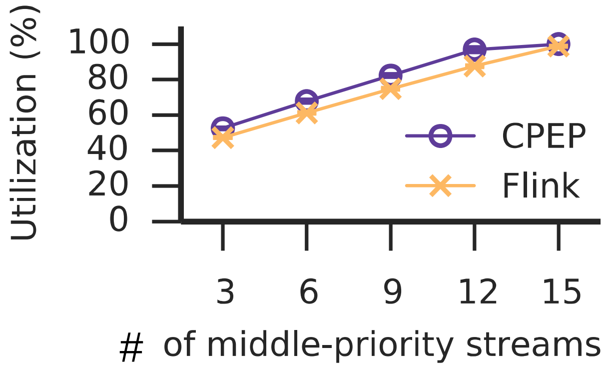
Both workers and movers are further prioritized, enabling an appropriate activity ordering.

Latency performance

99th percentile latency (unit: ms)

Priority	Service	Number of middle-priority streams				
		3	6	9	12	15
High	Flink	3.8 ± 0.1	5.9 ± 0.2	12.6 ± 0.4	52.6 ± 4.1	448.9 ± 171.7
	CPEP	0.8 ± 0.0	0.7 ± 0.0	0.7 ± 0.0	0.7 ± 0.0	0.7 ± 0.0
Middle	Flink	4.5 ± 0.1	6.4 ± 0.2	11.3 ± 0.4	28.9 ± 0.5	107.9 ± 18.1
	CPEP	1.6 ± 0.0	1.8 ± 0.0	2.2 ± 0.0	2.5 ± 0.0	3.0 ± 0.1
Low	Flink	5.2 ± 0.3	7.4 ± 0.2	15.5 ± 0.6	43.3 ± 1.3	679.8 ± 274.0
	CPEP	3.7 ± 0.3	4.8 ± 0.2	6.8 ± 0.6	10.6 ± 1.0	33.4 ± 0.2

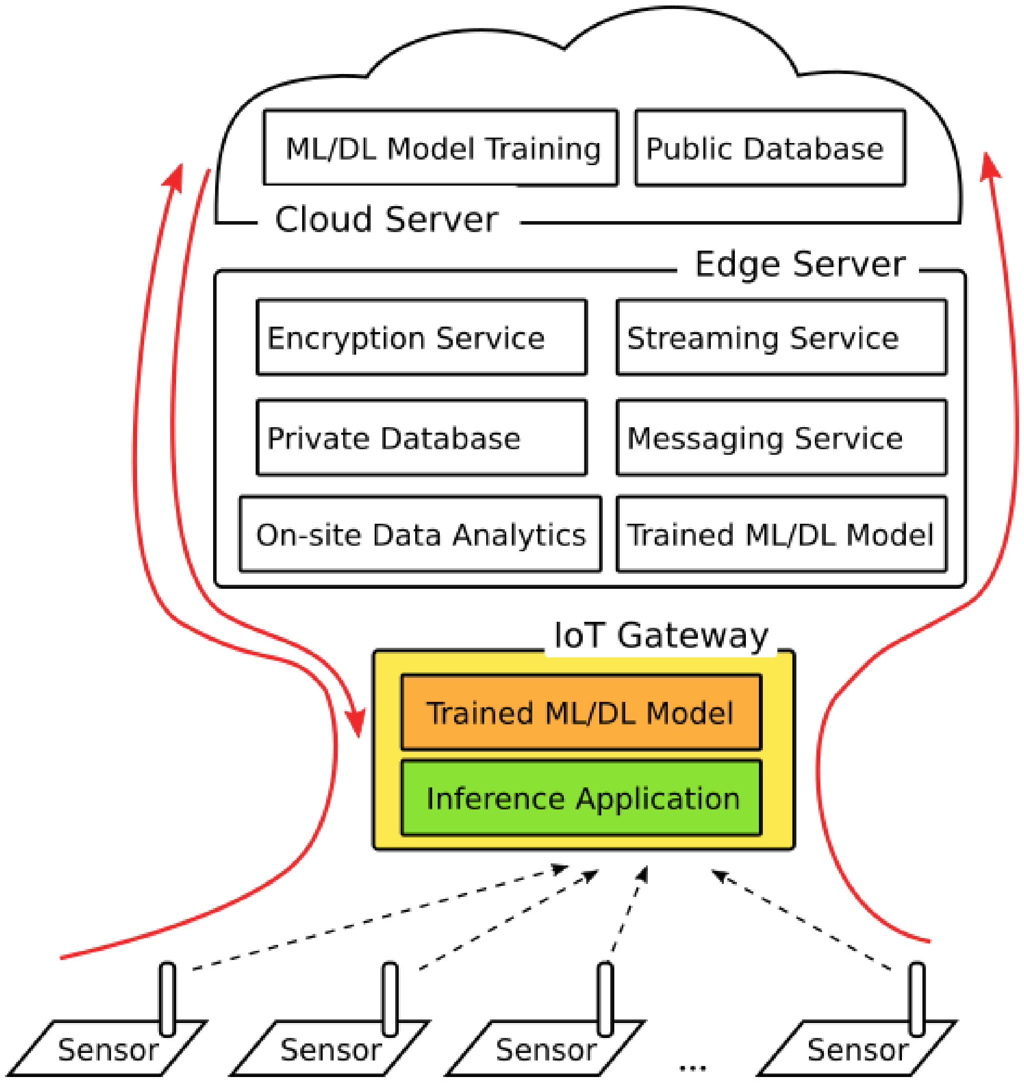
CPEP maintained high-priority latency performance as workload increased.



CPEP differentiated latency according to priority level.

Specific IoT gateway requirements

- Real-time requirement
  - Data subscriber imposes an end-to-end soft deadline for data
- **Fault-tolerance requirement**
  - **Data subscriber cannot accept some consecutive data losses**
- Efficiency requirement
  - The gateway should not consume too much local network bandwidth

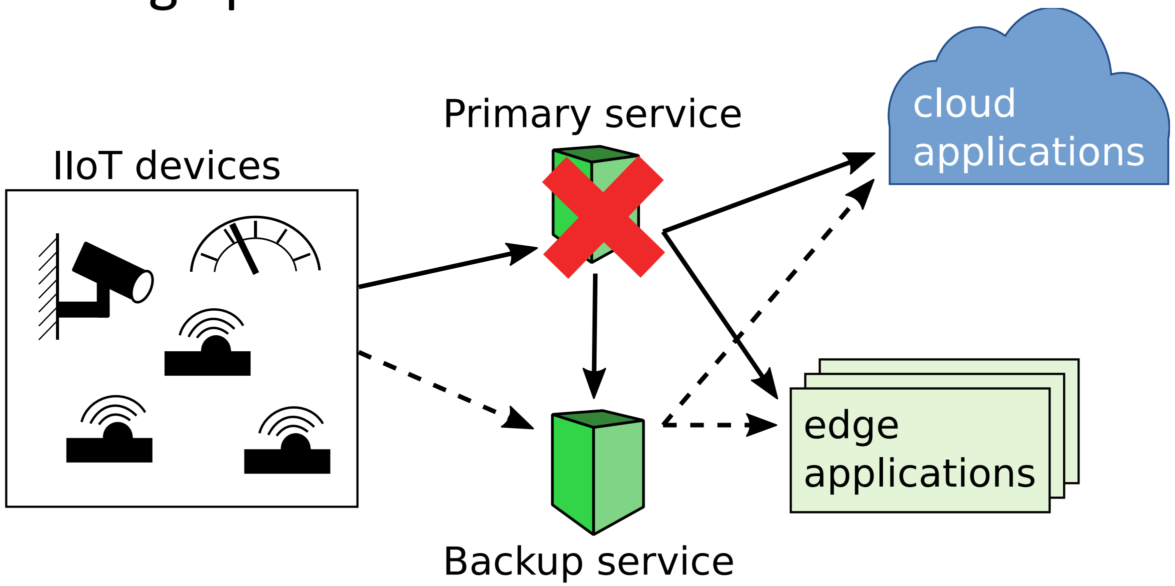


Dependability issues in computing systems

- What if some of the network components may fail to work?
- How to keep applications running properly while fixing the underlying network problems?
- Will fault-tolerance affect the performance of a data network? In which ways? How to amend it?

Fault-tolerance model

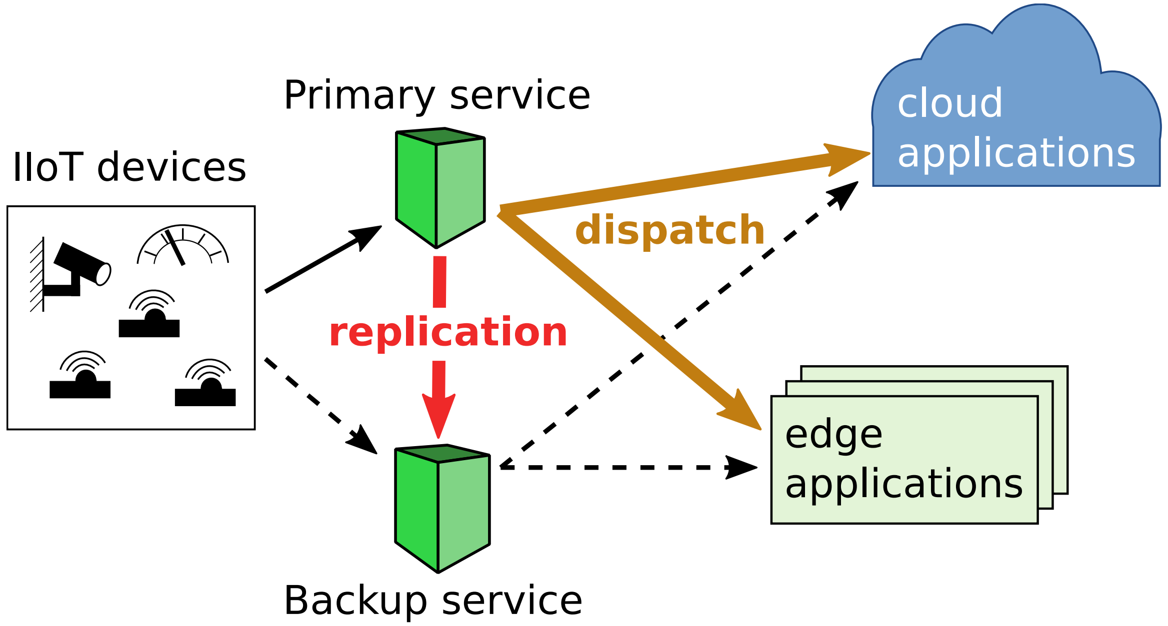
- A crash failure may happen to an IoT service host
- Lost message may be recovered
  1. via retransmissions from message publishers
  2. via a backup service [1]



[1] Budhiraja, N., Marzullo, K., Schneider, F.B. and Toueg, S., 1993. The primary-backup approach. *Distributed systems*, *2*, pp.199-216.

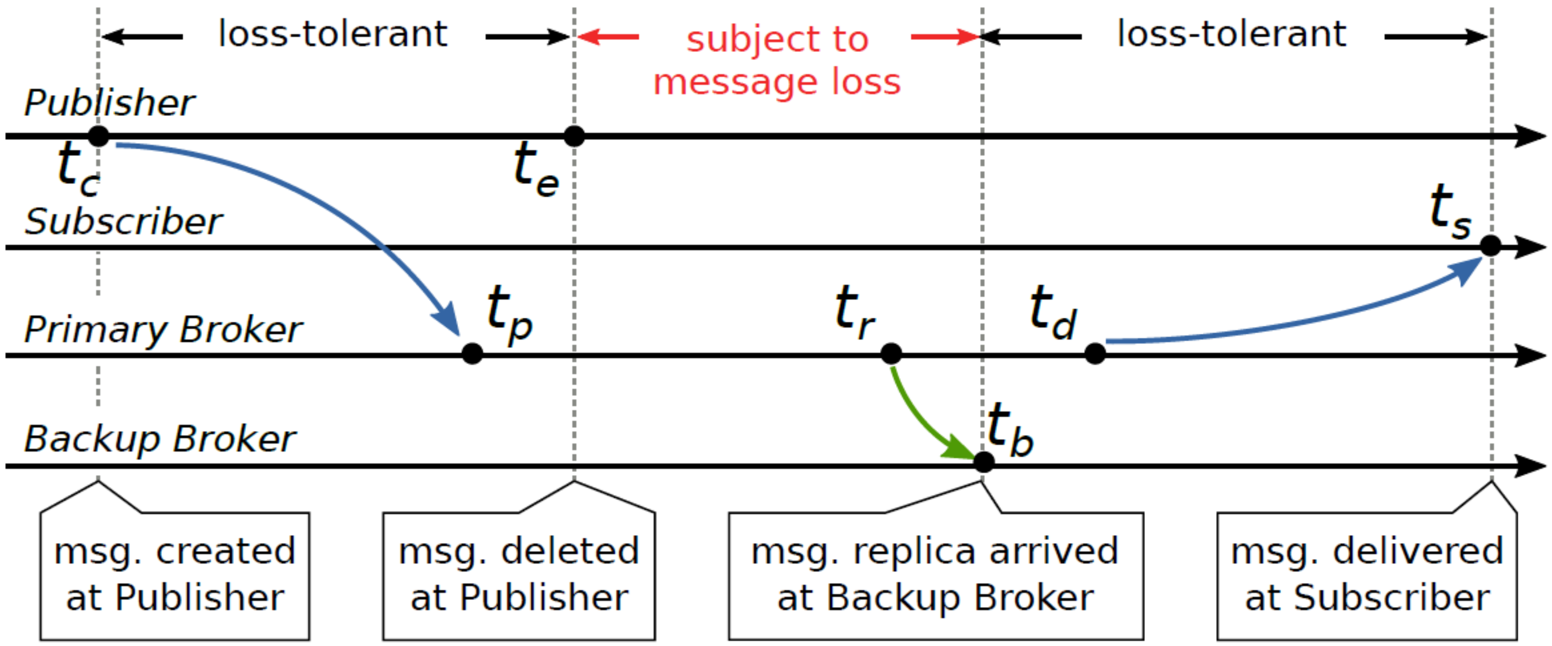
Design principles

- Fault-tolerant real-time processing:
  - Specify provable deadlines for message replication and dispatch
  - Co-schedule replication and dispatch using, e.g., the earliest deadline-first policy (EDF)



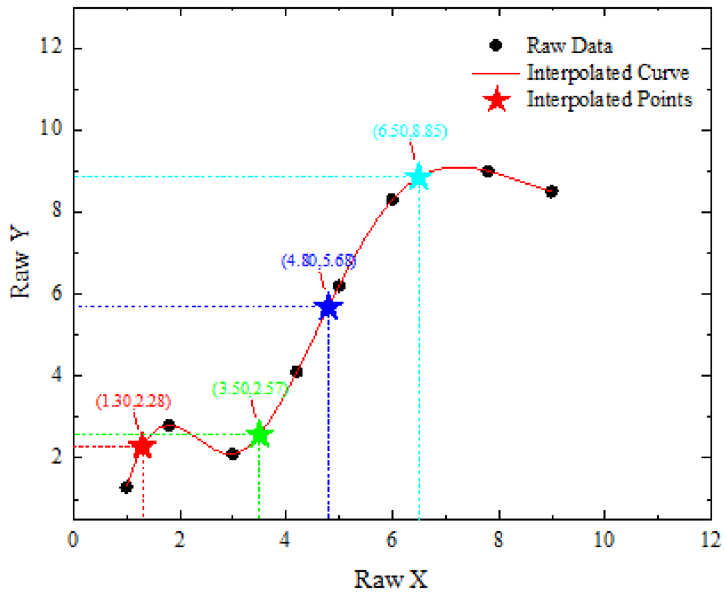
The conditions of message loss

- A message may be lost before it could arrive at a subscriber if a message replica has not been delivered at Backup Broker AND the publisher has deleted its message copy.



Message loss-tolerance requirement

- Application-specific requirements to an IIoT service
  - L_i : the tolerable number of consecutive losses for topic *i*



Value of L_i	Application examples
0	emergency response; predictive maintenance
$k > 0$	condition monitoring

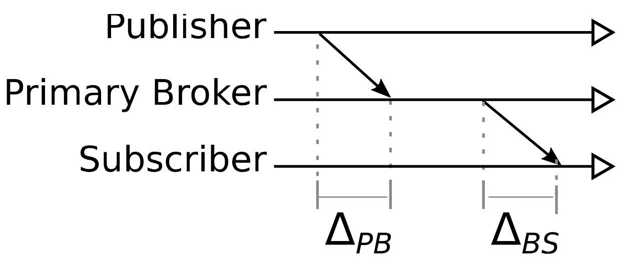
(Within the tolerable number, applications may use estimates for the missing data.)

Image source: https://www.originlab.com/doc/Origin-Help/Math-Inter-Extrapoltate-YfromX

Deadlines for data dispatch and replication

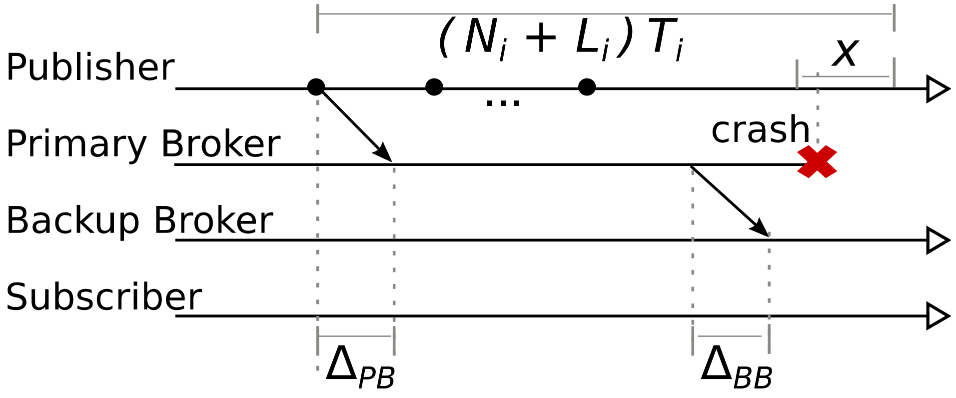
- Deadline for data dispatch:

$$\boxed{D_i^d \leq D_i - \Delta_{PB} - \Delta_{BS}}$$



- Deadline for data replication:

$$\boxed{D_i^r \leq (N_i + L_i)T_i - \Delta_{PB} - \Delta_{BB} - x}$$



T_i : topic's sending period

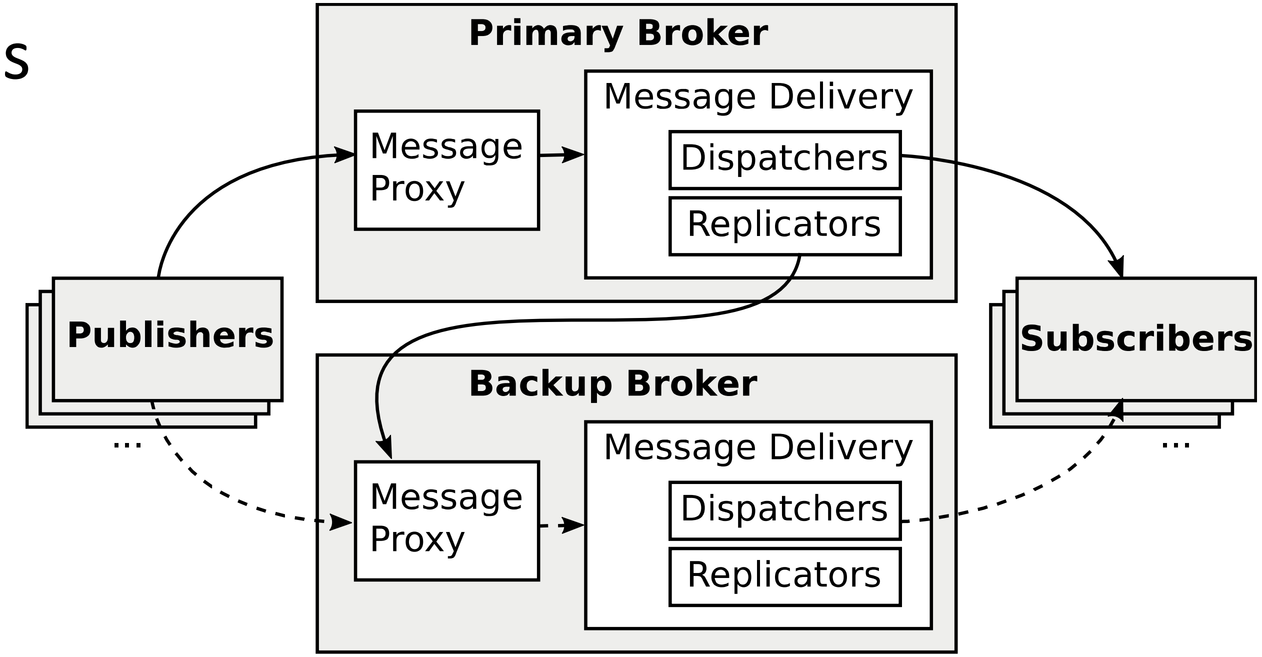
D_i : latency requirement

N_i : # of most-recent messages a publisher can retransmit

L_i : loss-tolerance requirement

A messaging architecture

- EDF scheduling to dispatch/replicate a message
- Suppress replication if the dispatch deadline is smaller
- Prune dispatched messages

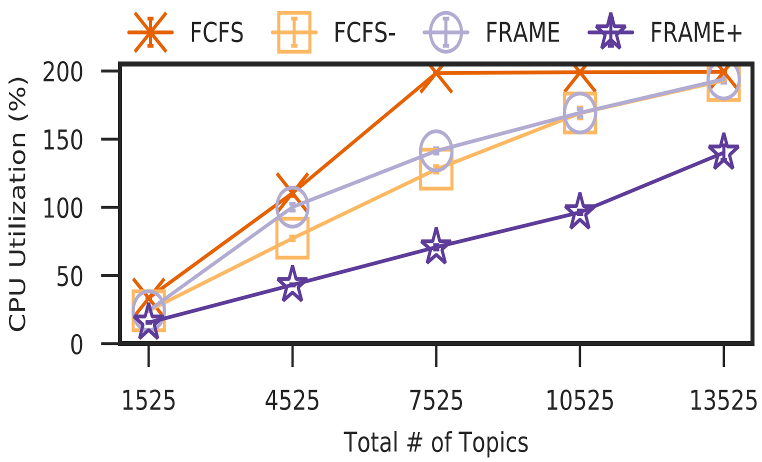


Loss tolerance performance

FRAME succeeded in assuring loss-tolerance.

A small increase in N_i can greatly improve performance (FRAME+).

$$D_i^r \leq (N_i + L_i)T_i - \Delta_{PB} - \Delta_{BB} - x$$



Success rate of meeting loss-tolerance requirements

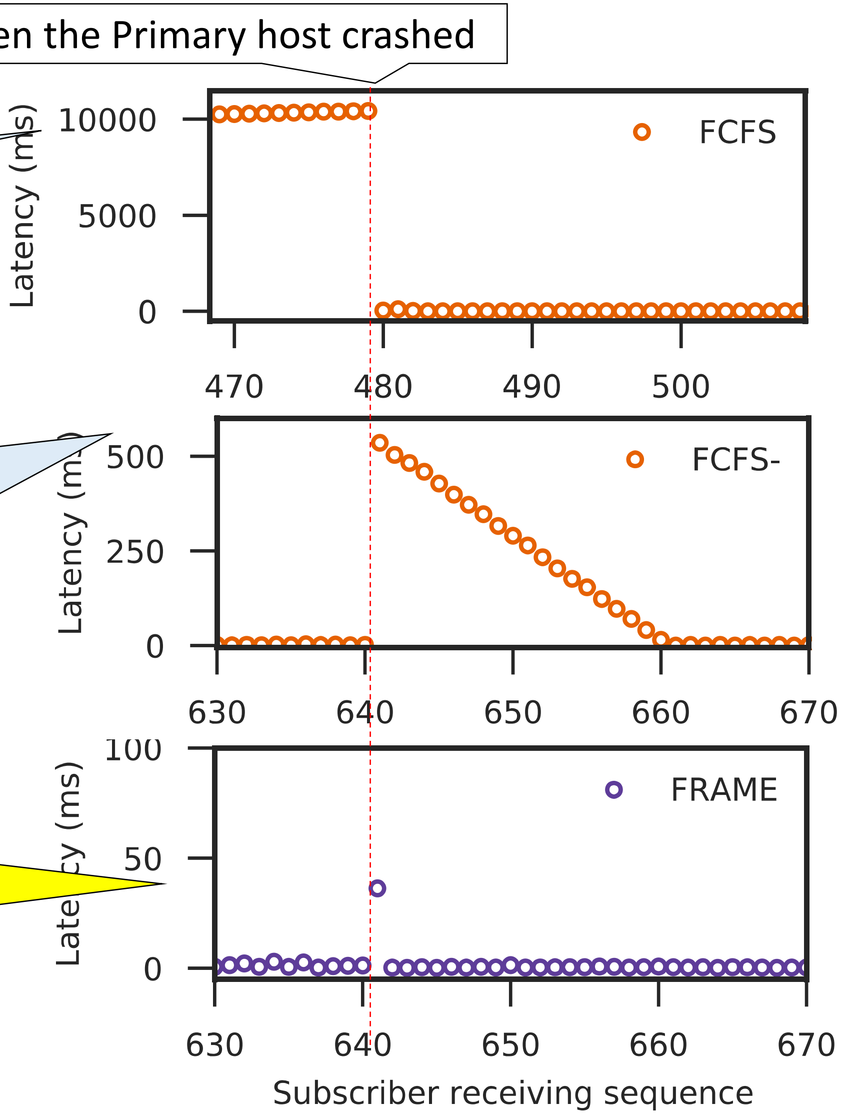
D_i	L_i	FRAME+	FRAME	FCFS	FCFS-
		Workload = 7525 Topics			
50	0	100.0	100.0	0.0	100.0
50	3	100.0	100.0	0.0	100.0
100	0	100.0	100.0	0.0	100.0
100	3	100.0	100.0	0.0	100.0
100	∞	100.0	100.0	100.0	100.0
500	0	100.0	100.0	0.0	100.0
		Workload = 10525 Topics			
50	0	100.0	100.0	0.0	100.0
50	3	100.0	100.0	0.0	100.0
100	0	100.0	100.0	0.0	100.0
100	3	100.0	100.0	0.0	100.0
100	∞	100.0	100.0	100.0	100.0
500	0	100.0	100.0	0.0	100.0
		Workload = 13525 Topics			
50	0	100.0	80.0 ± 30.1	0.0	100.0
50	3	100.0	80.0 ± 30.1	0.0	100.0
100	0	100.0	73.2 ± 30.7	0.0	78.4 ± 13.3
100	3	100.0	79.3 ± 29.9	0.0	99.3 ± 0.5
100	∞	100.0	100.0	100.0	100.0
500	0	100.0	80.0 ± 30.1	0.0	100.0

Latency performance for recovery

Without suppressing replication, pruning could overload the system.

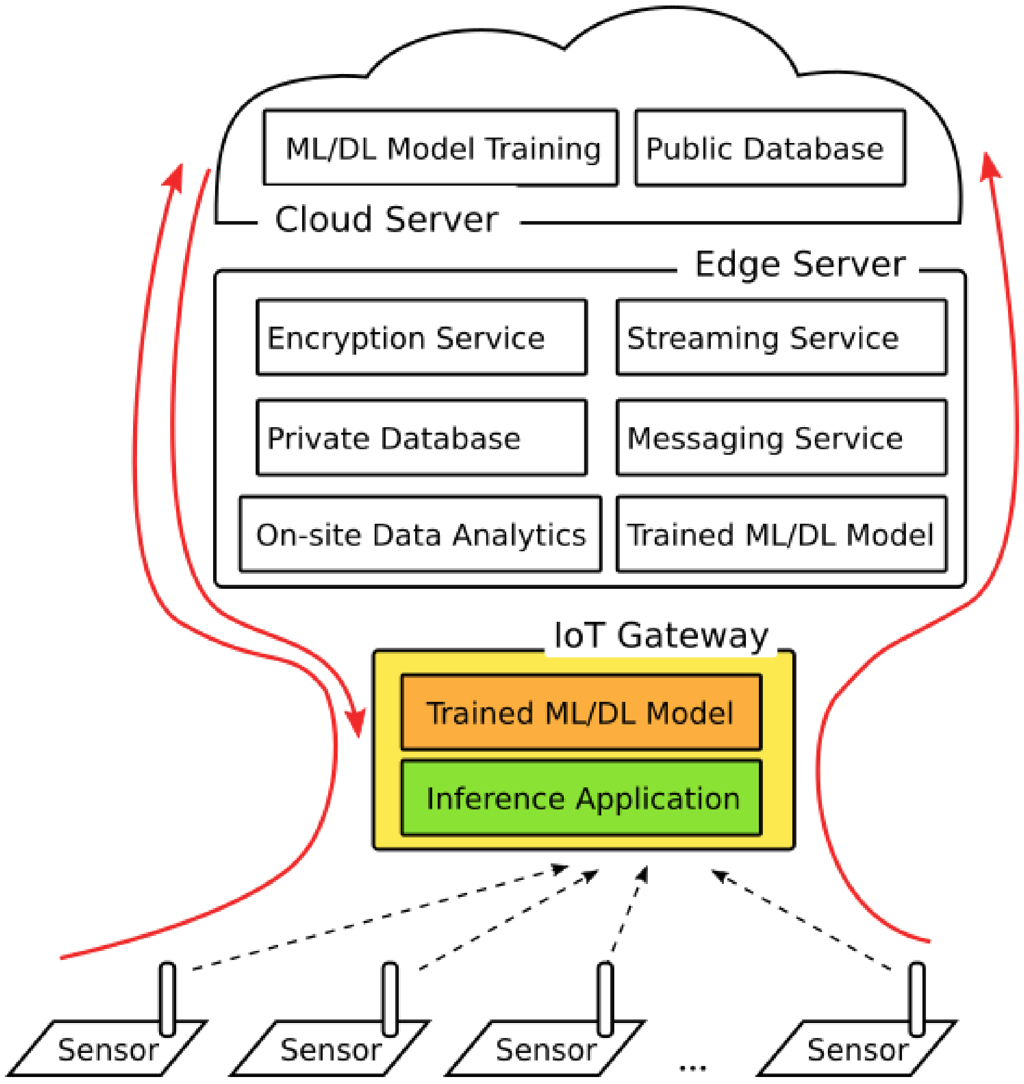
No pruning, however, could cause latency penalty by re-sending obsolete messages.

FRAME can mitigate latency penalty.



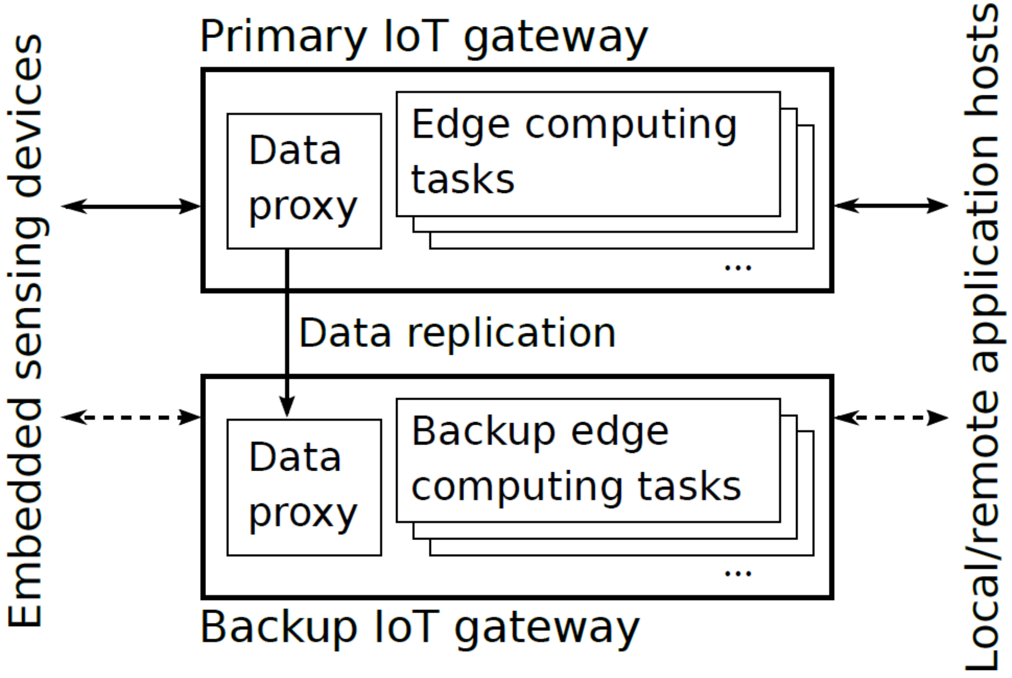
Specific IoT gateway requirements

- Real-time requirement
  - Data subscriber imposes an end-to-end soft deadline for data
- Fault-tolerance requirement
  - Data subscriber cannot accept some consecutive data losses
- **Efficiency requirement**
  - **The gateway should not consume too much local network bandwidth**



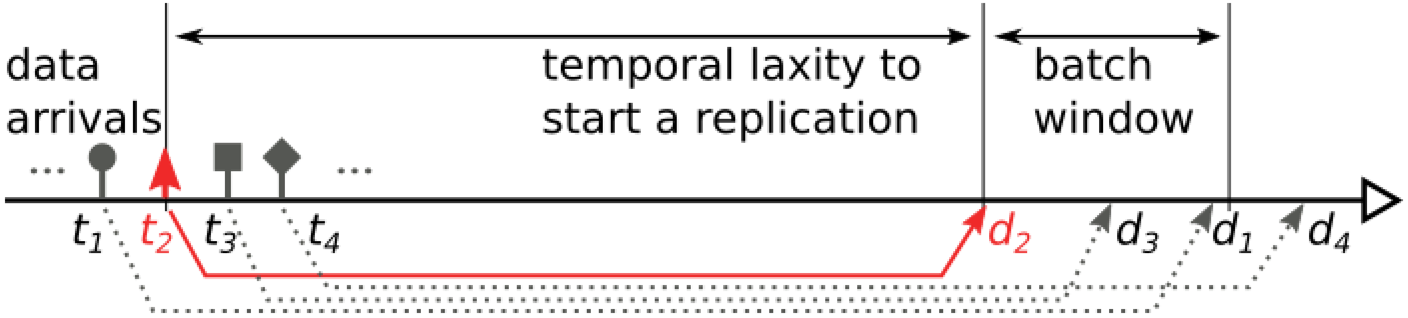
A system model for IoT edge computing

- Publish-subscribe data model
  - with a minimum inter-publishing time for each topic
- Each embedded sensing device can only keep N_i latest data elements
- Primary-backup fault tolerance
  - consider the crash failure (fail-stop)
  - data replication to backup



Key idea for adaptive data replication

- In the IoT gateway, once data is processed/delivered, it is irrelevant
- Therefore, for each data, we may postpone replication activities to reduce the need of actually performing the replication!

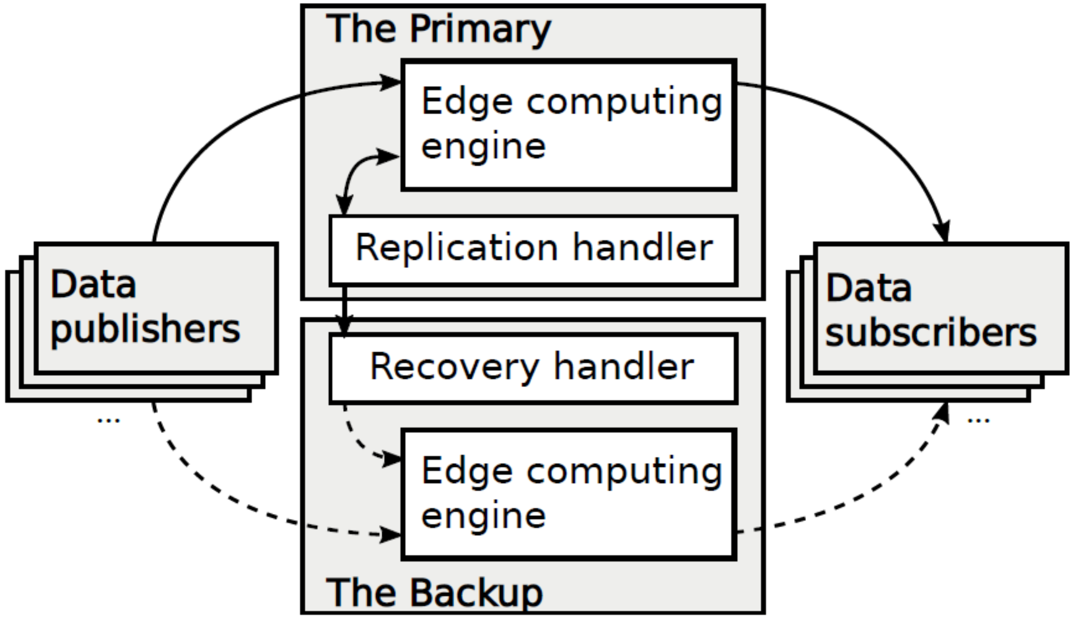


- Only start to replicate an arrival of data at the last starting time
- Once start, perform a batch of pending replications

Adaptive data replication architecture

- Edge computing engine schedules both computing tasks and replication tasks using the EDF policy

- Replication handler decides the intended rate of replication:



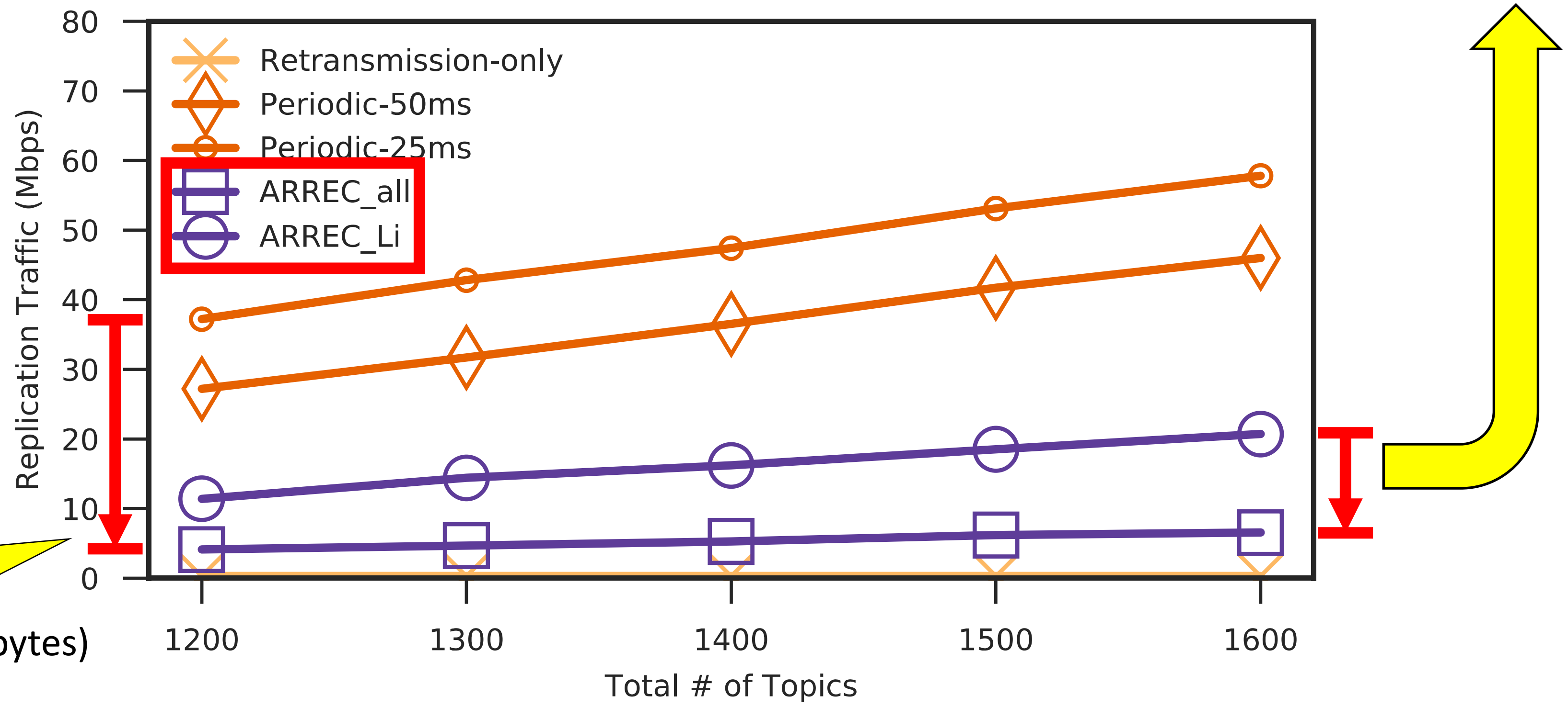
Lower intended rate -> tighter replication deadline

(see the paper for the analysis)

Chao Wang, Christopher Gill, Chenyang Lu. Adaptive Data Replication in Real-Time Reliable Edge Computing. ACM/IEEE International Conference on Internet of Things Design and Implementation (IoTDI), 2020

Empirical performance: efficiency in network bandwidth usage

A higher intended replication rate is preferred, because it permits a longer replication deadline, which in turn would allow the system to skip many more replication activities.



88% saving in bandwidth.

(Payload size = 512 bytes)

Concluding remarks

- Edge computing is key for CPS-IoT systems that shape our lives
- Performant IoT edge computing is challenging
  - Timeliness
  - Reliability
  - Resource efficiency
- In this talk we've discussed
  - Real-time CPS event processing
  - Fault-tolerant real-time messaging
  - Adaptive data replication in real-time reliable edge computing

References

- Chao Wang, Christopher Gill, and Chenyang Lu. 2019. Real-Time Middleware for Cyber-Physical Event Processing. ACM Transactions on Cyber-Physical Systems, 3, 3, Article 29, August 2019
- Chao Wang, Christopher Gill, Chenyang Lu. FRAME: Fault Tolerant and Real-Time Messaging for Edge Computing. IEEE International Conference on Distributed Computing Systems (ICDCS), 2019
- Chao Wang, Christopher Gill, Chenyang Lu. Adaptive Data Replication in Real-Time Reliable Edge Computing. ACM/IEEE International Conference on Internet of Things Design and Implementation (IoTDI), 2020