Real-Time Communications Infrastructures for Cyber-Physical Systems

Qixin Wang

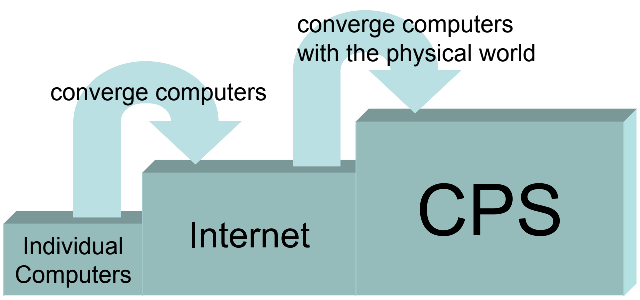
Dept. of Computing

The Hong Kong Polytechnic Univ.

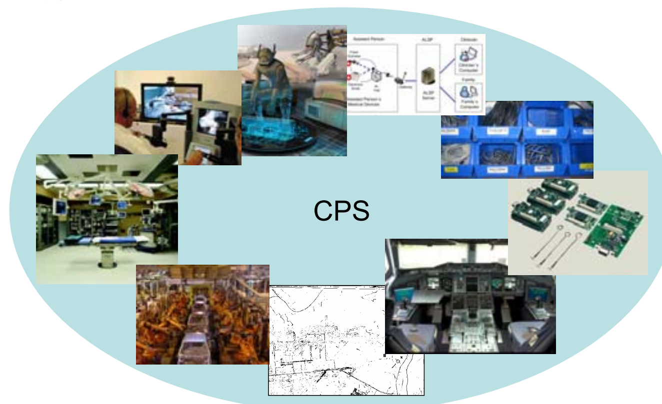
23/6/2009

Cyber-Physical Systems (CPS) is expected to be a theme topic in CS after the Internet

Synonym to Pervasive Computing



CPS covers a large variety of applications with crucial social and economic impacts



MDPnP integrates tens of thousands of medical devices for hospitals & assisted living

Cables for connecting various monitors to anesthesia EMR

Operation Room Automation

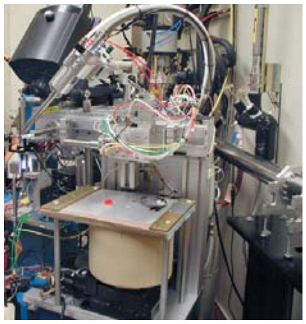
Reduce the risk of tripping over wires

Future

Today

Next Generation Industrial Control

- More embedded systems
- More interaction/complexity
- Wired → Wired + Wireless



Next Generation Aircraft

Concord

1976

Next Generation Aircraft

A380

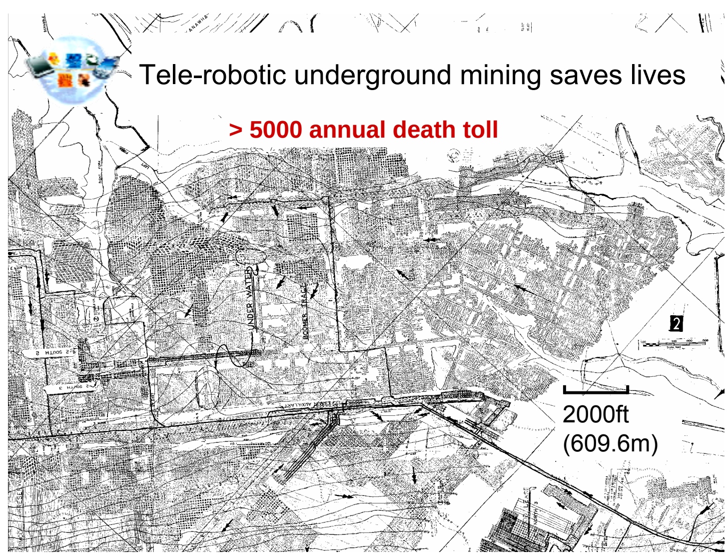
2007

Telepresence lets people collaborate without moving to the same geographical location

Telemedicine, Tele-Surgery save more lives

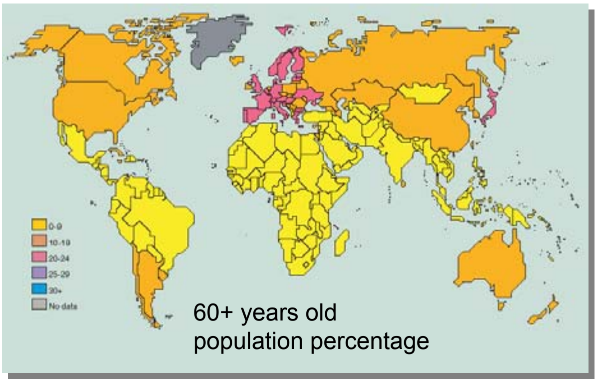
Tele-robotic underground mining saves lives

> 5000 annual death toll



Assisted Living helps saving our economy and society from the aging crisis

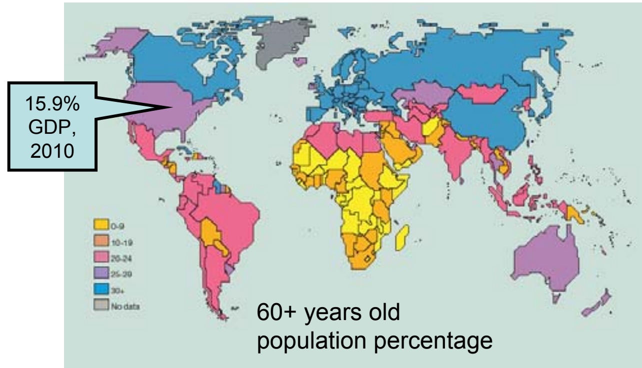
2002



SOURCE: United Nations • "Population Aging • 2002"

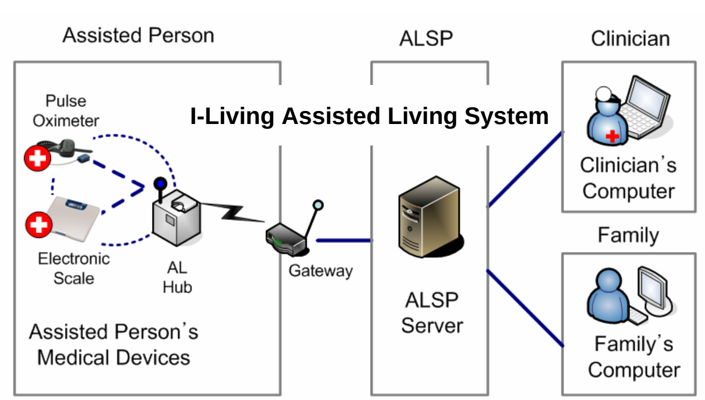
Assisted Living helps saving our economy and society from the aging crisis

2050

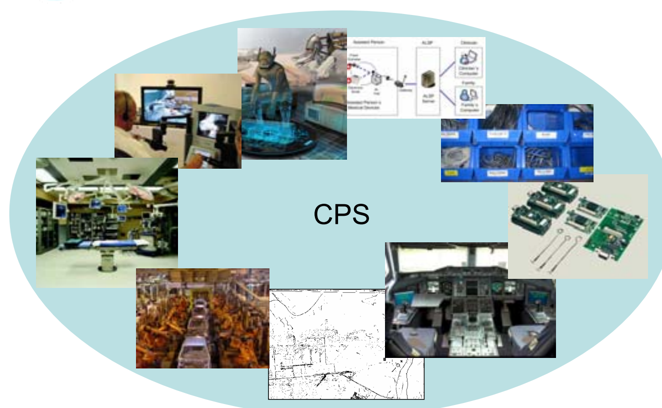


SOURCE: United Nations • "Population Aging • 2002"

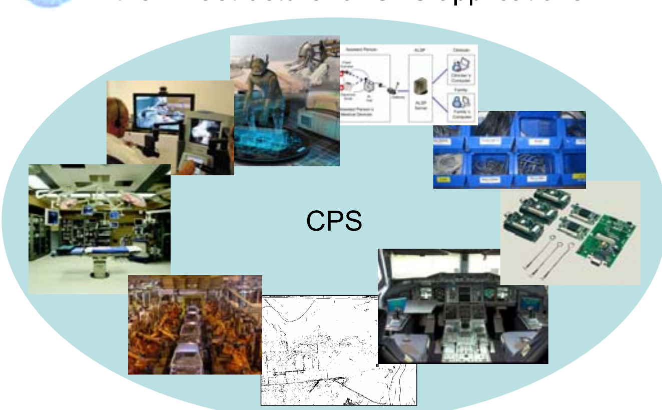
Assisted Living helps saving our economy and society from the aging crisis



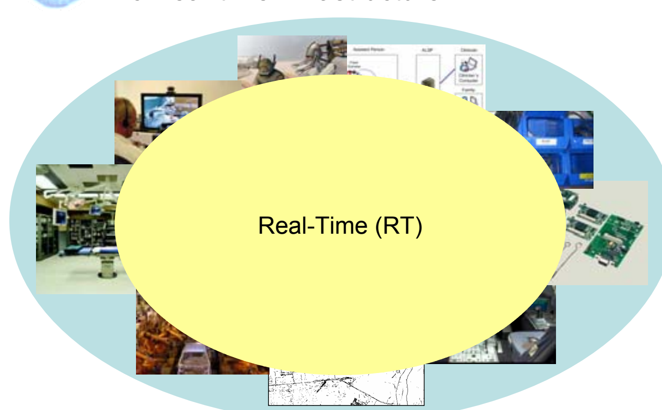
Many Other CPS Applications



Need many building block components to lay the infrastructure for CPS applications



One important category is the building blocks for real-time infrastructure



Robotic Surgery: each task is a continuous loop of sensing (or actuating) jobs

Each job:

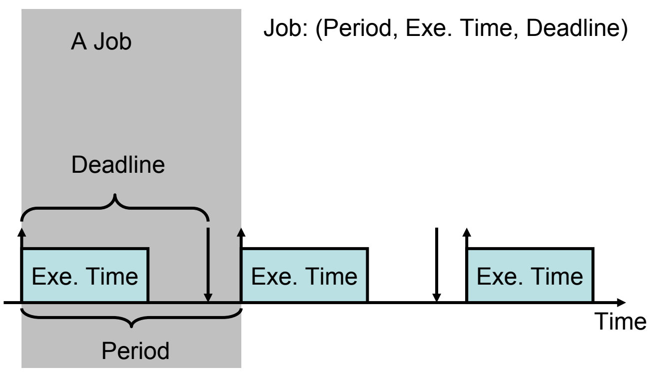
1. Must catch deadline
2. Does not have to be fast

Aviation and Industrial Control: each task is a continuous loop of sensing (or actuating) jobs

Each job:

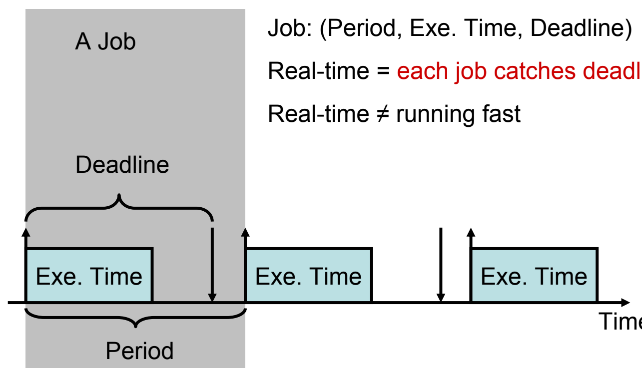
1. Must catch deadline
2. Does not have to be fast

A typical real-time task is a continuous loop of periodic jobs.



Job: (Period, Exe. Time, Deadline)

A typical real-time task is a continuous loop of periodic jobs.

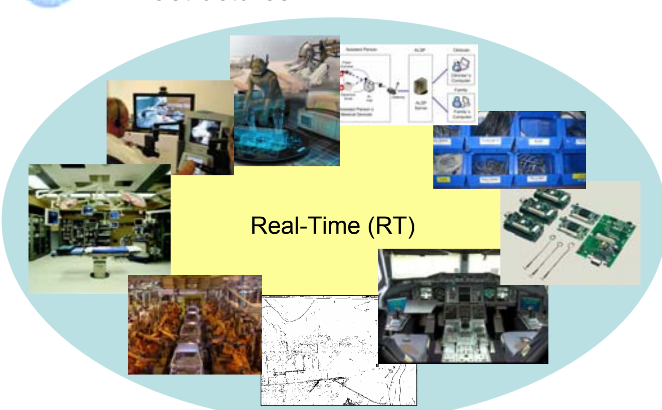


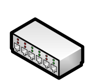
Job: (Period, Exe. Time, Deadline)

Real-time = each job catches deadline

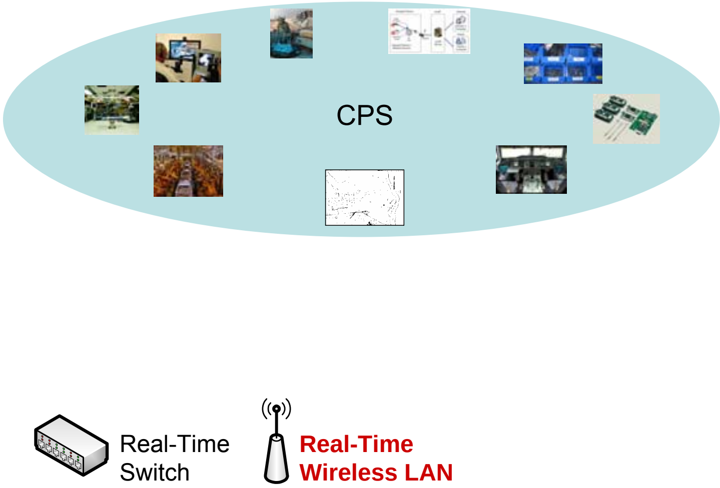
Real-time ≠ running fast

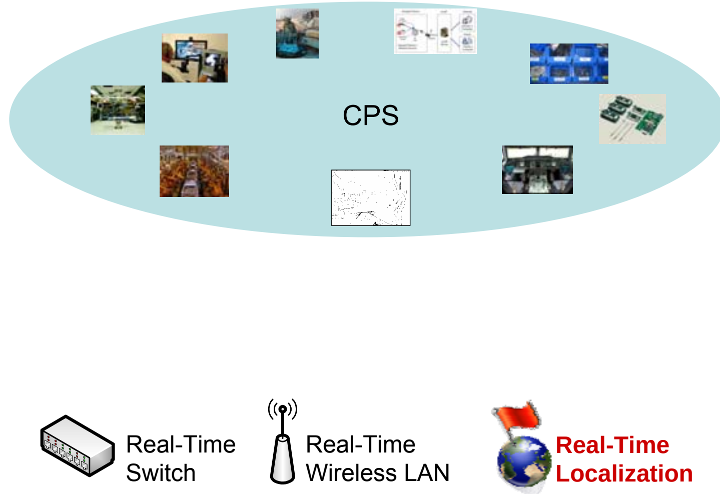
CPS need real-time communication infrastructures

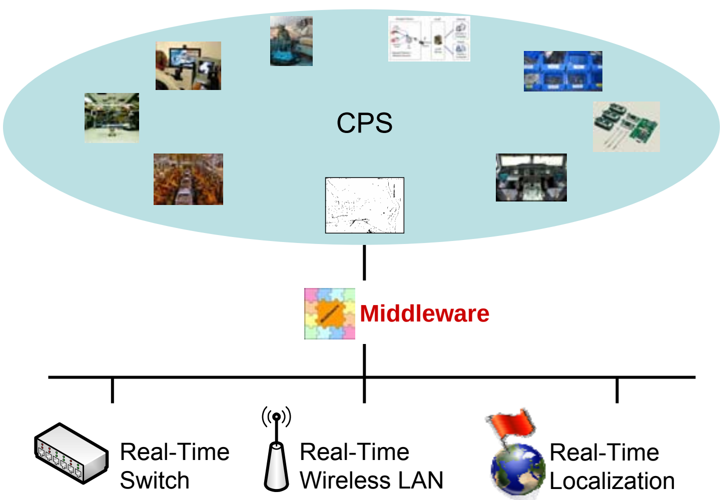


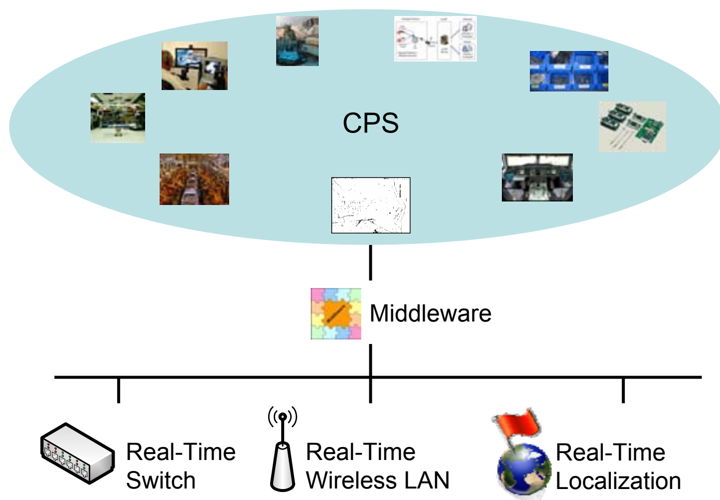


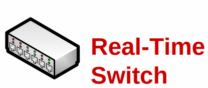
Real-Time Switch











Real-Time Switch is the key component for many infrastructures

Real-Time WAN, Real-Time Internet Subnet

Real-Time Fieldbus

RT High Performance Computing Architecture, e.g., InfiniBand

Real-Time Switch is the key component for many infrastructures

Real-Time WAN, Real-Time Internet Subnet

Real-Time Fieldbus

RT High Performance Computing Architecture, e.g., InfiniBand



Real-Time Switch is the key component for many infrastructures

Real-Time WAN, Real-Time Internet Subnet

Real-Time Fieldbus

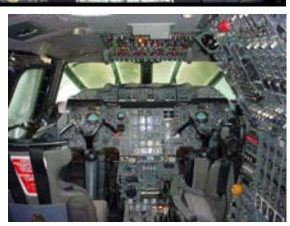
RT High Performance Computing Architecture, e.g., InfiniBand


Real-Time Switch is the key component for many infrastructures

Real-Time WAN, Real-Time Internet Subnet

Real-Time Fieldbus

RT High Performance Computing Architecture, e.g., InfiniBand

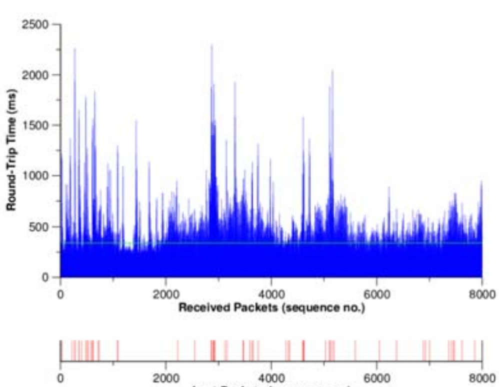



Designing real-time switch is not easy

1. A. Demers, S. Keshav, and S. Shenker, “Analysis and simulation of a fair queueing algorithm,” in SIGCOMM’89, 1989. pp. 1-12.
2. D. Ferari, and D. Verma, “A scheme for real-time channel establishment in wide-area networks,” in IEEE JSAC, v8, n3, April, 1990
3. D. C. Verma, H. Zhang, and D. Ferrari, “Delay jitter control for real-time communication in packet switching network,” in Proceedings of Tricomm’91, pp. 35-46, April 1991.
4. S. Golestani, “A framing strategy for congestion management,” in IEEE JSAC, v9, n7, pp. 1064-1077, Sep., 1991
5. L. Z ... y 199
6. A. ... de Ca
7. A. K. Parekh, R. G. Gallager, “A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple Node Case,” in IEEE/ACM ToN, v2, n2, April, 1994.
8. S. Jamaloddin Golestani, ``A self-clocked fair queuing scheme for broadband applications," Proc. IEEE INFOCOM'94, pages 636--646, IEEE, 1994.
9. H. Zhang, "Service Disciplines For Guaranteed Performance Service in Packet-Switching Networks," Proceedings of the IEEE, 83(10), Oct 1995
10. Jon C. R. Bennett and H. Zhang, ``WF2Q: worst-case fair weighted fair queueing," IEEE INFOCOM'96, March, 1996
11. Pawan Goyal, Harrick M. Vin, and Haichen Cheng, "Start-time fair queueing: a scheduling algorithm for integrated services packet switching networks," ACM SIGCOMM'96, pages 157--168, ACM press, August, 1996
12. I. R. Philp, “Scheduling real-time messages in packet-switched networks,” Ph.D. Thesis, Technical Report No. UIUCDCS-R-96-1977, CS Dept., UIUC, Oct., 1996.
13. Subhash Suri, George Varghese, and Girish Chandranmenon, ``Leap forward virtual clock: A new fair queuing scheme with guaranteed delays and throughput fairness," IEEE Proc. INFOCOM'97, 1997
14. I. Stoica, S. Shenker, and H. Zhang, ``Core-stateless fair queuing: achieving approximation fair bandwidth allocations in high speed networks," Proc. of ACM SIGCOMM'98, October 1998, pp. 118-130.
15. C. R. Kalmanek, H. Kanakia, and S. Keshav, “Rate controlled servers for very high speed networks,” in IEEE Global Telecommunications Conference, Dec., 1999.
16. W. Sun, and K. G. Shin, “End-to-end delay bounds for traffic aggregates under guaranteed-rate scheduling algorithms,” in IEEE ToN, 13(5): 1188-1201, Oct., 2005.

Huge Volume of Literature since 1989

Today’s Internet is still NOT real-time.

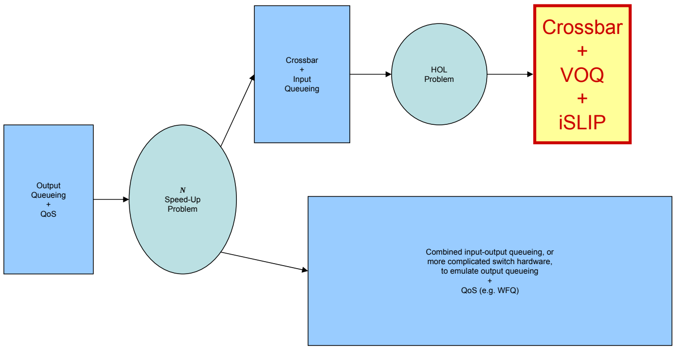


Measured delay btw UK and Spain [Marsh06]

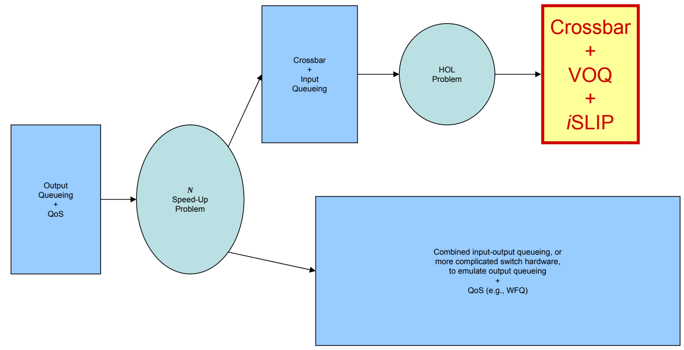
Today’s Internet is still NOT real-time.

Why?

Industry acceptance has the final say. Two things rules: simplicity and switch hardware features



Past lessons tell us to design the real-time switch compatible to "crossbar + VOQ + *i*SLIP"



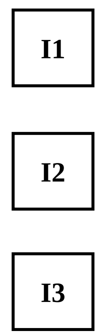
What is "Crossbar + VOQ + *i*SLIP"?

A widely implemented switch architecture

Advantages
- Simple
- High switch utilization
- Fast adaptation to random traffic (Internet traffic)

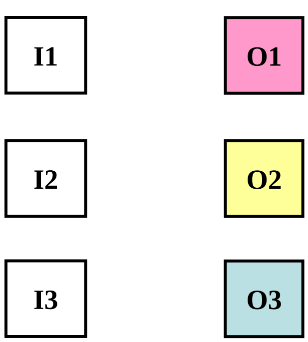
What is "Crossbar + VOQ + *i*SLIP"?

Input Ports



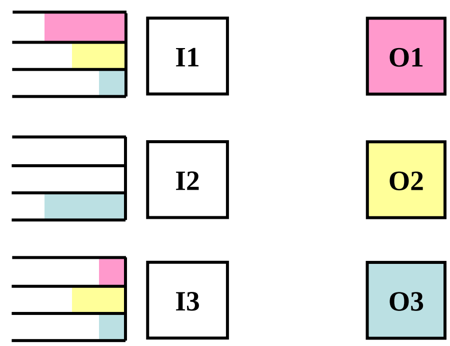
What is "Crossbar + VOQ + *i*SLIP"?

Output Ports

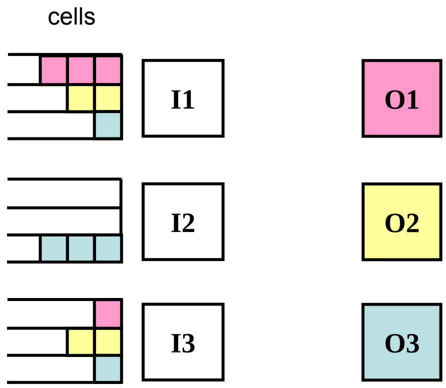


What is "Crossbar + VOQ + *i*SLIP"?

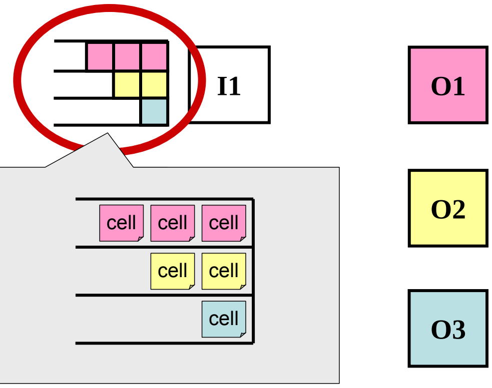
Virtual Output Queue (VOQ)



What is “Crossbar + VOQ + *i*SLIP”?

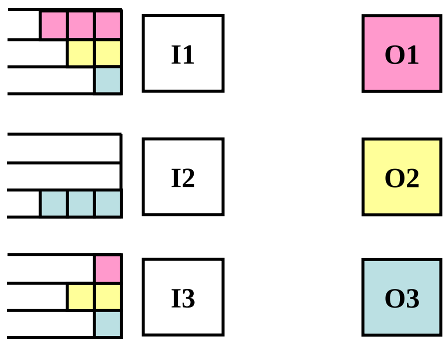


What is “Crossbar + VOQ + *i*SLIP”?



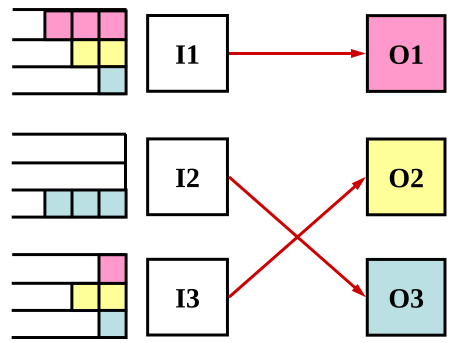
What is “Crossbar + VOQ + *i*SLIP”?

Synchronous periodic cell forwarding
Cell-Time



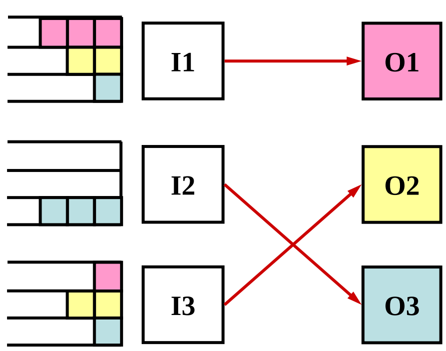
What is “Crossbar + VOQ + *i*SLIP”?

Matching

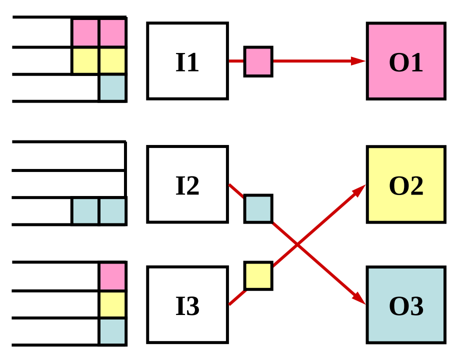


What is “Crossbar + VOQ + *i*SLIP”?

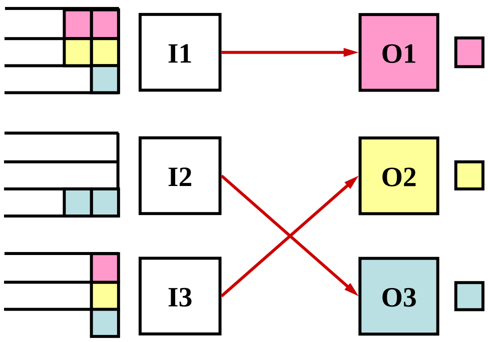
Why Matching? An input/output can only send/receive one cell per cell-time



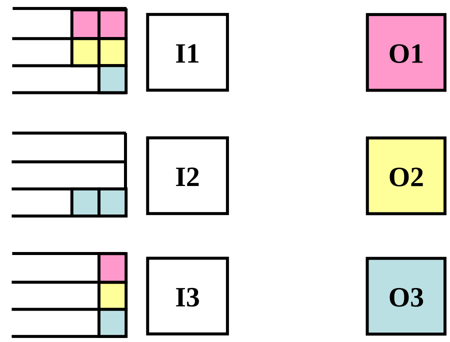
What is “Crossbar + VOQ + *i*SLIP”?



What is “Crossbar + VOQ + *i*SLIP”?

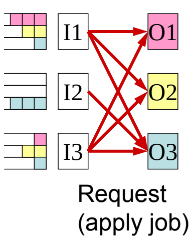


What is “Crossbar + VOQ + *i*SLIP”?



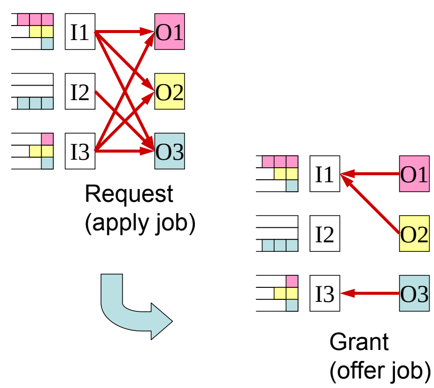
What is “Crossbar + VOQ + *i*SLIP”?

*i*SLIP is like faculty job hunting ☺



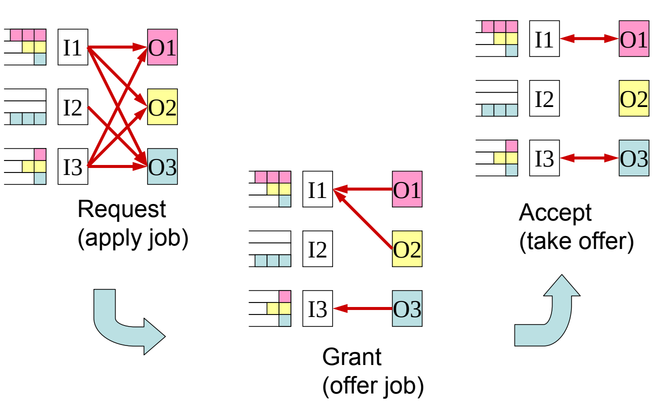
What is “Crossbar + VOQ + *i*SLIP”?

*i*SLIP is like faculty job hunting ☺



What is “Crossbar + VOQ + *i*SLIP”?

*i*SLIP is like faculty job hunting ☺



But we cannot directly adopt “crossbar + VOQ + *i*SLIP”

Huge per hop delay bound because of poor isolation;

E2E delay bound is an open problem [Gopalakrishnan06].

But we cannot directly adopt “crossbar + VOQ + *i*SLIP”

Huge per hop delay bound because of poor isolation;

E2E delay bound is an open problem [Gopalakrishnan06].

What to do?

But we cannot directly adopt “crossbar + VOQ + *i*SLIP”

Huge per hop delay bound because of poor isolation;

E2E delay bound is an open problem [Gopalakrishnan06].

What to do?

Look at changed assumptions and new features.

Changed assumption: traffic predictability

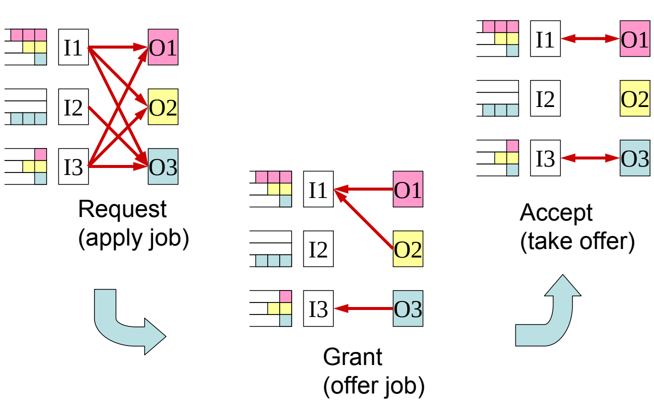
*i*SLIP is for non-real-time traffic: Unpredictable

Changed assumption: traffic predictability

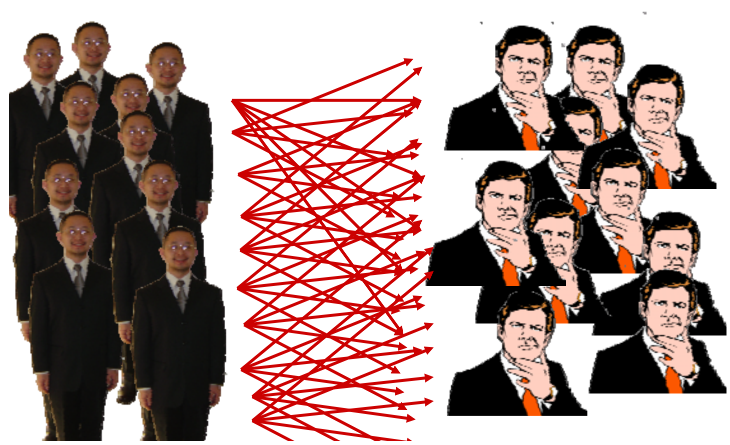
*i*SLIP is for non-real-time traffic: Unpredictable

- Flows change rapidly
- Unpredictable message size and arrival

Unpredictable traffic forces *i*SLIP to negotiate for each cell-time



Just like faculty job application, *i*SLIP’s negotiate is dynamic, hard to give tight delay upper bound.



Changed assumption: traffic predictability

*i*SLIP is for non-real-time traffic: Unpredictable
- Flows change rapidly
- Unpredictable message size and arrival

RT-Switch for real-time traffic: Deterministic

Changed assumption: traffic predictability

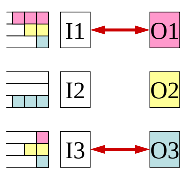
*i*SLIP is for non-real-time traffic: Unpredictable
- Flows change rapidly
- Unpredictable message size and arrival

RT-Switch for real-time traffic: Deterministic
- Flows rarely change
- Regular message size and arrival
- Non-real-time traffic → real-time traffic

Deterministic traffic allows RT-Switch to forward cells like clockwork, no need to negotiate!

Deterministic Grant is enough

No need for Request and Accept



Simplifies instead of extends *i*SLIP

New feature: Large message arrival period allows clock-driven time slicing

Real-time traffic

New feature: Large message arrival period allows clock-driven time slicing

Real-time traffic
- Small msg size (sensing: 1cell/msg, TV quality video: 480cell/msg)

New feature: Large message arrival period allows clock-driven time slicing

Real-time traffic
- Small msg size (sensing: 1cell/msg, TV quality video: 480cell/msg)
- Short per msg transmission time (s: 0.5us, v: 240us)

New feature: Large message arrival period allows clock-driven time slicing

Real-time traffic

- Small msg size (sensing: 1cell/msg, TV quality video: 480cell/msg)
- Short per msg transmission time (s: 0.5us, v: 240us)
- Large msg arrival period (s: 10ms, v: 30ms)

New feature: Large message arrival period allows clock-driven time slicing

Real-time traffic

- Small msg size (sensing: 1cell/msg, TV quality video: 480cell/msg)
- Short per msg transmission time (s: 0.5us, v: 240us)
- Large msg arrival period (s: 10ms, v: 30ms)

Reminds us of clock-driven time slicing OS

New feature: Large message arrival period allows clock-driven time slicing

Real-time traffic

- Small msg size (sensing: 1cell/msg, TV quality video: 480cell/msg)
- Short per msg transmission time (s: 0.5us, v: 240us)
- Large msg arrival period (s: 10ms, v: 30ms)

Reminds us of clock-driven time slicing OS

RT-Switch runs fine time grain clock period (1ms), serving time slices (cell-time) to coarse time grain periodic RT flows

Solution: crossbar RT-Switch runs deterministic clock-driven time slicing

Solution: crossbar RT-Switch runs deterministic clock-driven time slicing

Clock period of M cell-time, e.g., M = 5

Cell time:	1	2	3	4	5
$I1$:					
$I2$:					
$I3$:					
$I4$:					

Solution: crossbar RT-Switch runs deterministic clock-driven time slicing

Clock period of M cell-time, e.g., M = 5

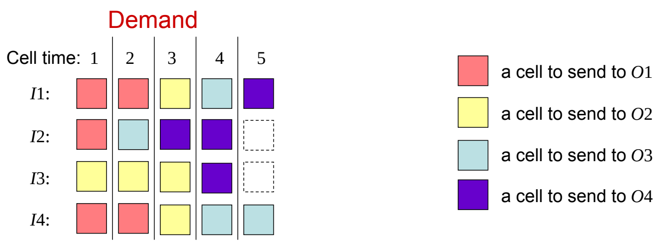
Fit original real-time flow-forwarding tasks into clock period, e.g., (11, 3) → (5, 2), i.e., (10, 4)

Cell time:	1	2	3	4	5
$I1$:					
$I2$:					
$I3$:					
$I4$:					

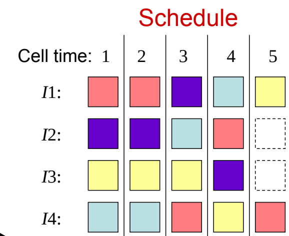
Solution: crossbar RT-Switch runs deterministic clock-driven time slicing

Clock period of M cell-time, e.g., $M = 5$

Fit original real-time tasks into clock period, e.g., (11, 3) → (5, 2), i.e., (10, 4)



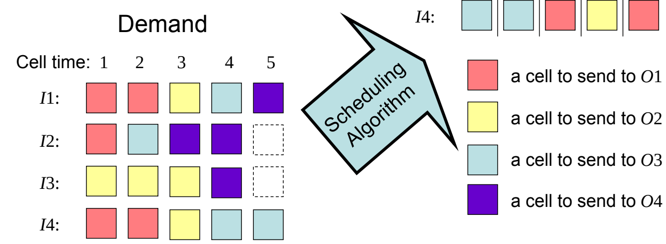
Solution: crossbar RT-Switch runs deterministic clock-driven time slicing



Clock period of M cell-time

Fit original RT tasks into clk period

Config. time scheduling ($O(N^4)$)



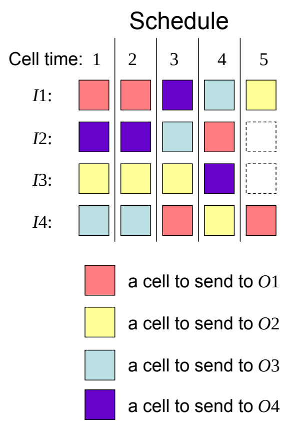
Solution: crossbar RT-Switch runs deterministic clock-driven time slicing

Clock period of M cell-time

Fit original RT tasks into clk period

Config. time scheduling ($O(N^4)$)

Forward cells according to the schedule during runtime



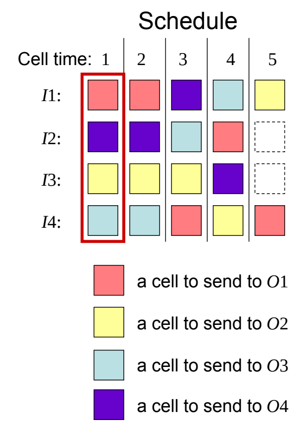
Solution: crossbar RT-Switch runs deterministic clock-driven time slicing

Clock period of M cell-time

Fit original RT tasks into clk period

Config. time scheduling ($O(N^4)$)

Forward cells according to the schedule during runtime



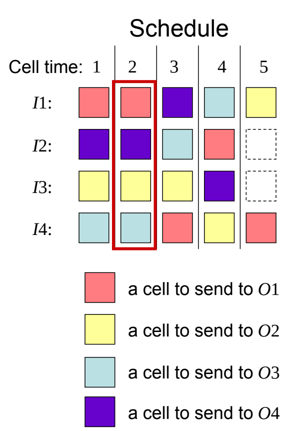
Solution: crossbar RT-Switch runs deterministic clock-driven time slicing

Clock period of M cell-time

Fit original RT tasks into clk period

Config. time scheduling ($O(N^4)$)

Forward cells according to the schedule during runtime



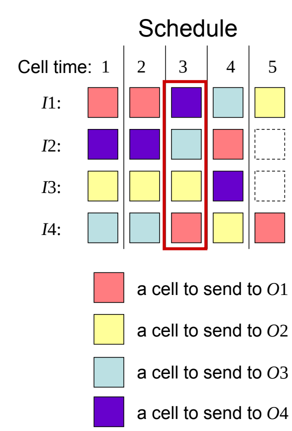
Solution: crossbar RT-Switch runs deterministic clock-driven time slicing

Clock period of M cell-time

Fit original RT tasks into clk period

Config. time scheduling ($O(N^4)$)

Forward cells according to the schedule during runtime



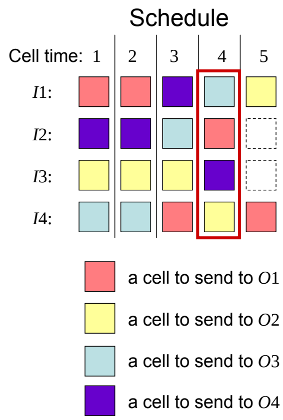
Solution: crossbar RT-Switch runs deterministic clock-driven time slicing

Clock period of M cell-time

Fit original RT tasks into clk period

Config. time scheduling ($O(N^4)$)

Forward cells according to the schedule during runtime



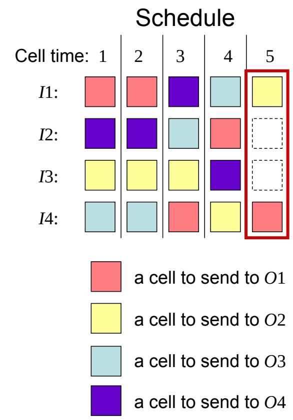
Solution: crossbar RT-Switch runs deterministic clock-driven time slicing

Clock period of M cell-time

Fit original RT tasks into clk period

Config. time scheduling ($O(N^4)$)

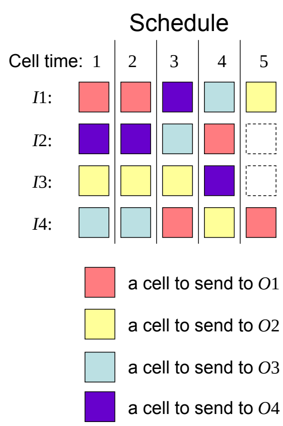
Forward cells according to the schedule during runtime



Solution: crossbar RT-Switch runs deterministic clock-driven time slicing

Satisfies demands:

- Real-Time Guarantee
- Backward Compatibility
- Simplicity
- Performance
- Isolation



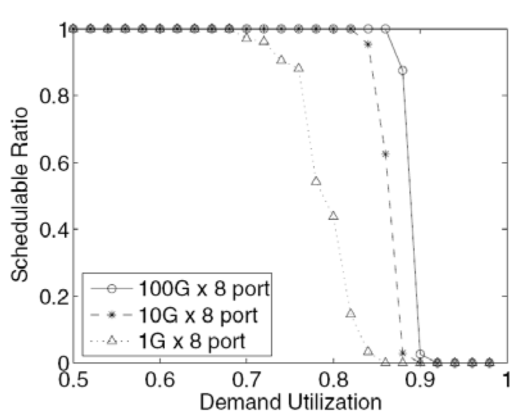
Evaluation: Schedulabililty

Randomized traffic demand based on typical real-time industrial/medical application models

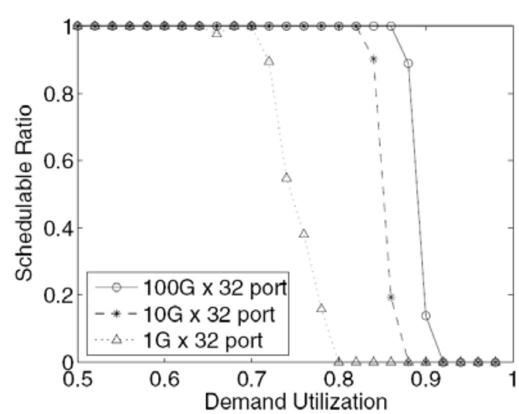
Total demanded switch utilization is U

Given U, what is the schedulable ratio?

Evaluation: Schedulability



Evaluation: Schedulability



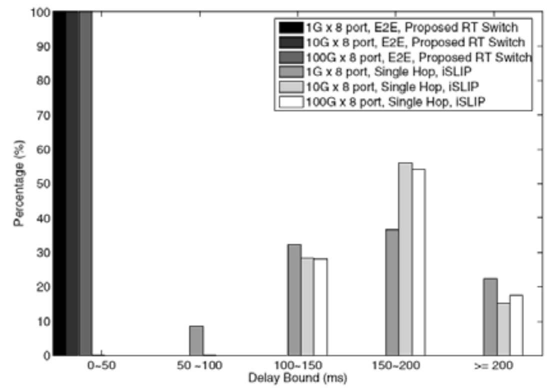
Evaluation: E2E Delay Bound

Randomized traffic demand based on typical real-time industrial/medical application models

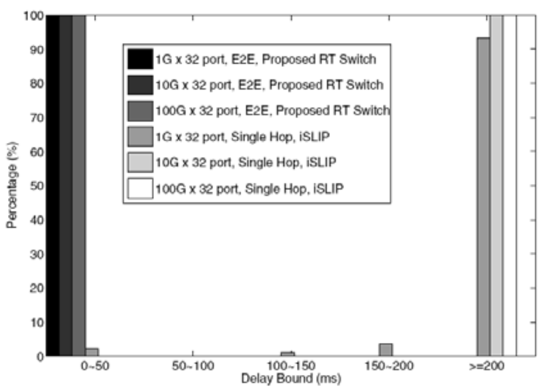
Max hop count is 15

Compare E2E Delay Bound provided by Real-Time Switch and *i*SLIP

Evaluation: E2E Delay Bound



Evaluation: E2E Delay Bound



Conclusion

Short E2E delay guarantee

Good schedulability

Compatible to, and simpler than the widely implemented *i*SLIP crossbar switch.

$O(1)$ runtime computation

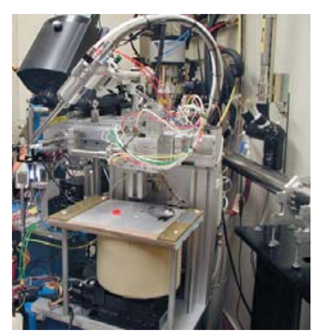
Polynomial configuration time computation

Real-Time Wireless LAN

The demand for real-time wireless communication is increasing.

Mechanical Freedom / Mobility

Ease of Deployment / Flexibility



The demand for real-time wireless communication is increasing.

Cables for connecting various monitors to anesthesia EMR

The demand for real-time wireless communication is increasing.

Reduce the risk of tripping over wires

Today

Future

Reliability and Robustness is the top concern for real-time wireless communication.

Cannot back off under adverse wireless channel conditions

Reliability and Robustness is the top concern for real-time wireless communication.

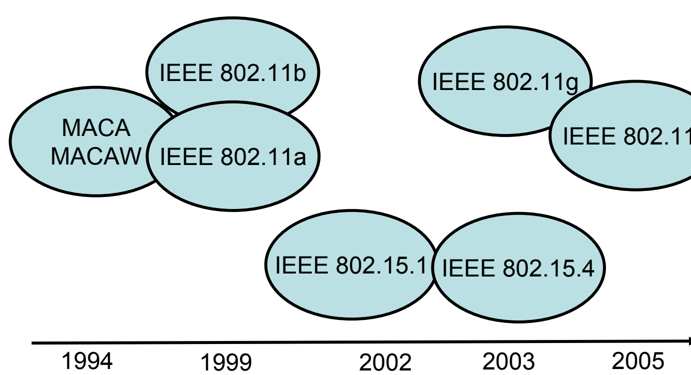
Cannot back off under adverse wireless channel conditions

Reliability and Robustness is the top concern for real-time wireless communication.

Adverse wireless medium

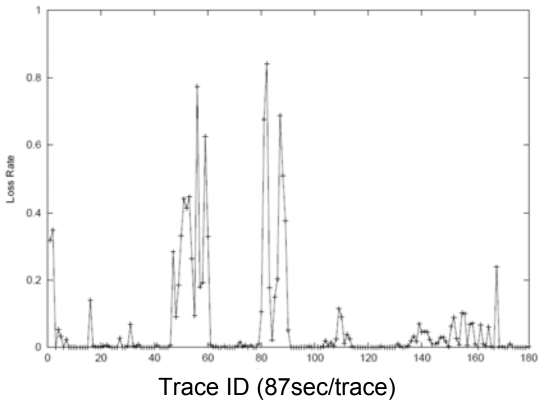
- Large scale path-loss
- Multipath
- Persistent electric-magnetic interference
- Same-band / adjacent-band RF devices

Nowadays wireless LANs are NOT real-time.



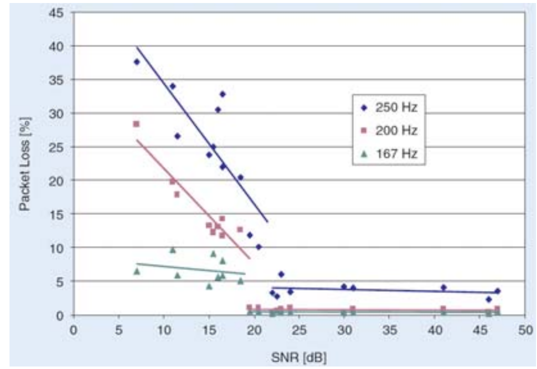
Nowadays wireless LANs are NOT real-time.

IEEE 802.11 packet loss rate in an industrial environment [Willig02]



Nowadays wireless LANs are NOT real-time.

IEEE 802.11 packet loss rate in an office environment [Ploplys04]

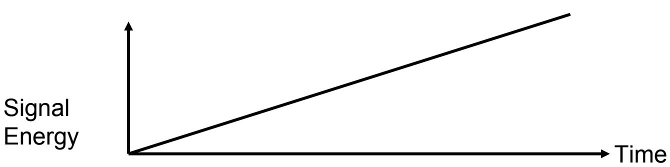


Nowadays wireless LANs are NOT real-time.

Why?

Design philosophy mismatch: pursuing large data throughput & short delay

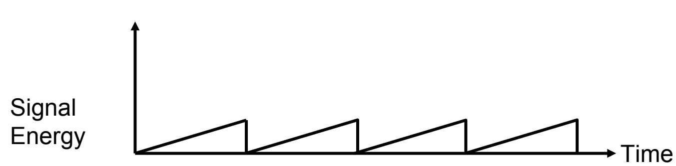
Design philosophy mismatch: pursuing large data throughput & short delay



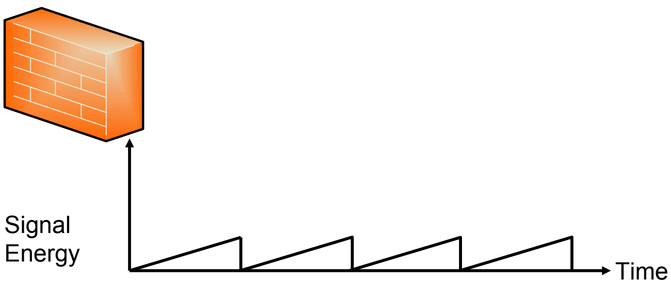
Design philosophy mismatch: pursuing large data throughput & short delay

Send packet fast

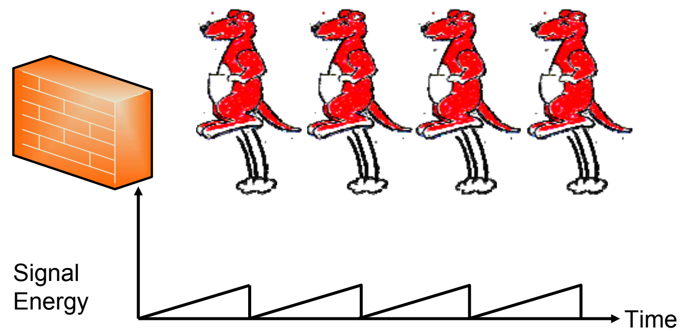
Do not spend much time accumulate strength



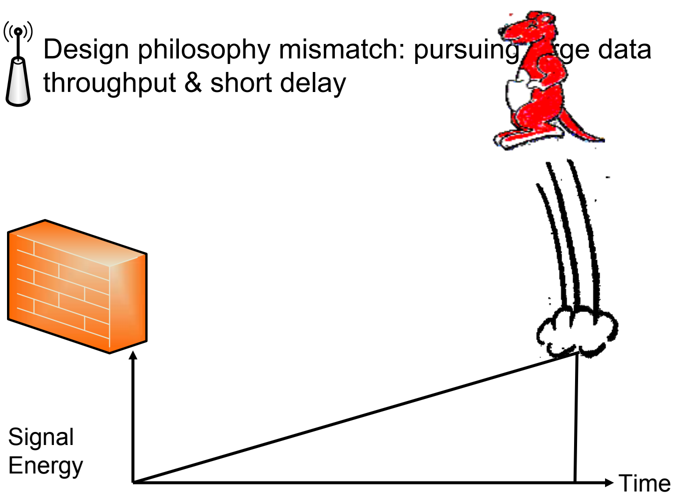
Design philosophy mismatch: pursuing large data throughput & short delay



Design philosophy mismatch: pursuing large data throughput & short delay



Design philosophy mismatch: pursuing large data throughput & short delay



Observation: Real-time communications are usually persistent connections with low data rate

Typical inter-node traffic:

100~200 bit/pkt, 10~1 pkt/sec per connection.

Observation: Real-time communications are usually persistent connections with low data rate

Typical inter-node traffic:

100~200 bit/pkt, 10~1 pkt/sec per connection.

Information Theory:

Lower data rate → higher robustness.

Observation: Real-time communications are usually persistent connections with low data rate

Typical inter-node traffic:

100~200 bit/pkt, 10~1 pkt/sec per connection.

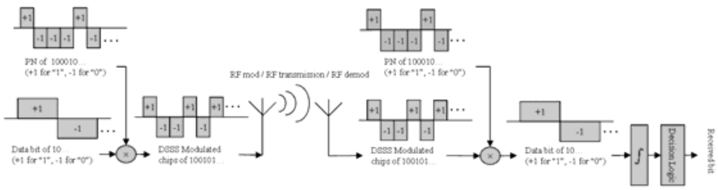
Information Theory:

Lower data rate → higher robustness.

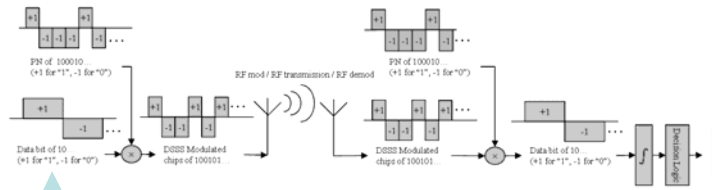
Direct Sequence Spread Spectrum (DSSS) Technology:

Lower data rate ↔ Higher robustness

Tutorial on DSSS



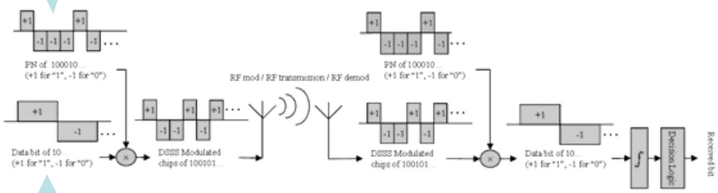
Tutorial on DSSS



Data stream, a.k.a bit stream. Bit rate: r_b .

Tutorial on DSSS

Pseudo Noise Sequence (PN) Stream, a.k.a chip stream. Chip rate: R_c.

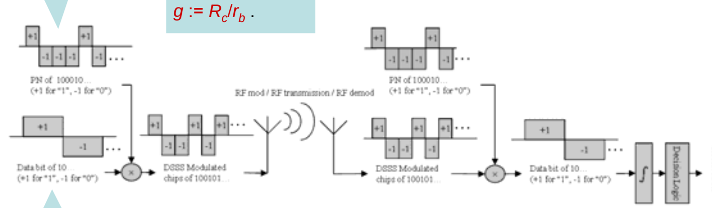


Data stream, a.k.a bit stream. Bit rate: r_b .

Tutorial on DSSS

Pseudo Noise Sequence (PN) Stream, a.k.a chip stream. Chip rate: R_c.

Definition: Processing Gain $g := R_c/r_b$.

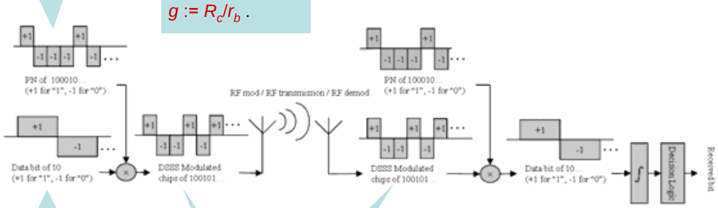


Data stream, a.k.a bit stream. Bit rate: r_b .

Tutorial on DSSS

Pseudo Noise Sequence (PN) Stream, a.k.a chip stream. Chip rate: R_c.

Definition: Processing Gain $g := R_c/r_b$.



Data stream, a.k.a bit stream. Bit rate: r_b .

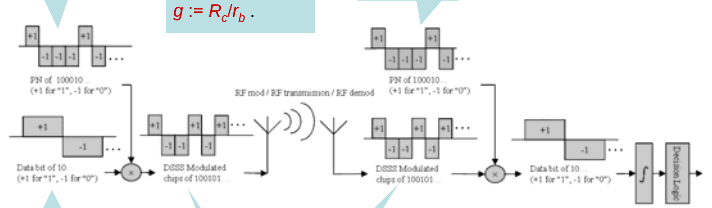
DSSS Modulated Stream, a.k.a Scrambled Stream

Tutorial on DSSS

Pseudo Noise Sequence (PN) Stream, a.k.a chip stream. Chip rate: R_c.

Definition: Processing Gain $g := R_c/r_b$.

Same PN Sequence



Data stream, a.k.a bit stream. Bit rate: r_b .

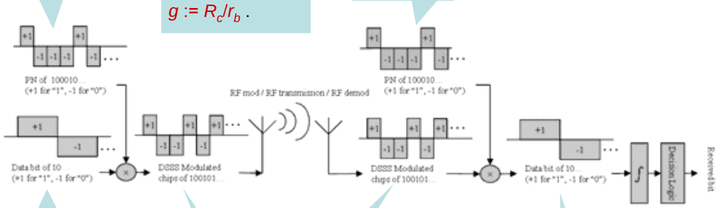
DSSS Modulated Stream, a.k.a Scrambled Stream

Tutorial on DSSS

Pseudo Noise Sequence (PN) Stream, a.k.a chip stream. Chip rate: R_c.

Definition: Processing Gain $g := R_c/r_b$.

Same PN Sequence



Data stream, a.k.a bit stream. Bit rate: r_b .

DSSS Modulated Stream, a.k.a Scrambled Stream

Original Data

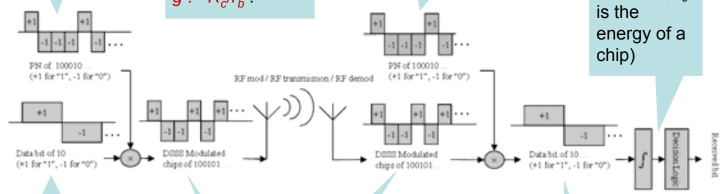
Tutorial on DSSS

Pseudo Noise Sequence (PN) Stream, a.k.a chip stream. Chip rate: R_c.

Definition: Processing Gain $g := R_c/r_b$.

Same PN Sequence

Integration $= gE_c$ for each bit (E_c is the energy of a chip)



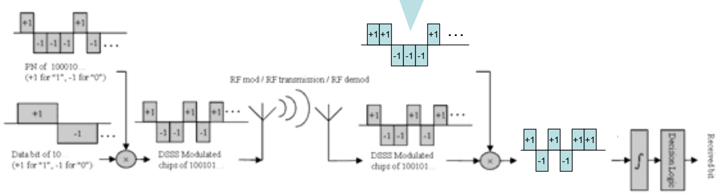
Data stream, a.k.a bit stream. Bit rate: r_b .

DSSS Modulated Stream, a.k.a Scrambled Stream

Original Data

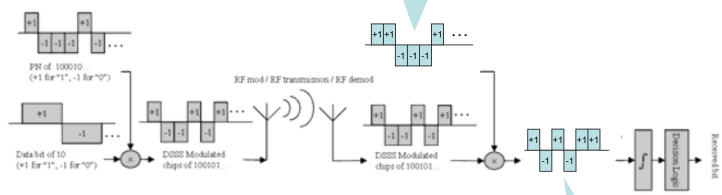
Tutorial on DSSS

If a different PN Sequence is applied



Tutorial on DSSS

If a different PN Sequence is applied

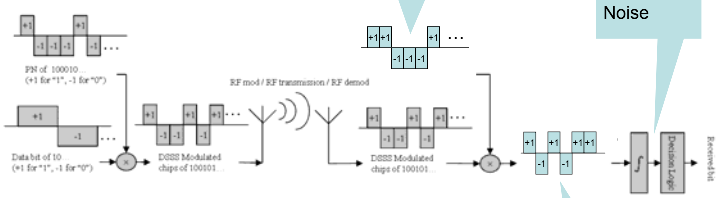


Another scrambled sequence

Tutorial on DSSS

If a different PN Sequence is applied

Integration = Gaussian Noise



Another scrambled sequence

Observation

Bit Error Rate

Processing Gain

$$\mathcal{P}_{ber} \leq \exp\left(-\frac{gP_u}{J+\sum_{i=1,i\neq u}^{\Xi} P_i + \sum_{h=1}^{H} A_h + P_u}\right)$$

- DSSS Technology:

  Larger Processing Gain g ⇔ Lower data rate ⇔ Lower Bit Error Rate (Higher robustness)

Observation: DSSS can exploit low data rate to achieve higher robustness

$$P_{BER} \leq \exp(-gK) = \exp\left(-\frac{K}{r_b}\right)$$

DSSS BER Upper Bound

Observation: DSSS can exploit low data rate to achieve higher robustness

Bit Error Rate

$$P_{BER} \leq \exp(-gK) = \exp\left(-\frac{K}{r_b}\right)$$

DSSS BER Upper Bound

Observation: DSSS can exploit low data rate to achieve higher robustness

Bit Error Rate

$$P_{BER} \leq \exp(-gK) = \exp\left(-\frac{K}{r_b}\right)$$

Constant

DSSS BER Upper Bound

Observation: DSSS can exploit low data rate to achieve higher robustness

Bit Error Rate

Processing Gain

$$P_{BER} \leq \exp(-gK) = \exp\left(-\frac{K}{r_b}\right)$$

Constant

DSSS BER Upper Bound

Observation: DSSS can exploit low data rate to achieve higher robustness

Bit Error Rate

Processing Gain

$$P_{BER} \leq \exp(-gK) = \exp\left(-\frac{K}{r_b}\right)$$

Constant

Bit Rate

DSSS BER Upper Bound

Observation: DSSS can exploit low data rate to achieve higher robustness

Bit Error Rate

Processing Gain

$$P_{BER} \leq \exp(-gK) = \exp\left(-\frac{K}{r_b}\right)$$

Constant

Bit Rate

DSSS BER Upper Bound

Lower data rate $\boldsymbol{r_b}$

Observation: DSSS can exploit low data rate to achieve higher robustness

Bit Error Rate

Processing Gain

$$P_{BER} \leq \exp(-gK) = \exp\left(-\frac{K}{r_b}\right)$$

Constant

Bit Rate

DSSS BER Upper Bound

Lower data rate r_b ←→ Larger Processing Gain g

Observation: DSSS can exploit low data rate to achieve higher robustness

Bit Error Rate

Processing Gain

$$P_{BER} \leq \exp(-gK) = \exp\left(-\frac{K}{r_b}\right)$$

Constant

Bit Rate

DSSS BER Upper Bound

Lower data rate r_b ←→ Larger Processing Gain g ←→ Lower Bit Error Rate P_{BER} (higher robustness)

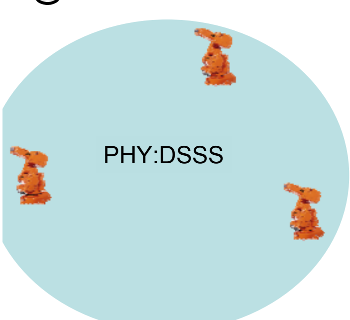
Key Idea: How to configure for max robustness for adverse wireless medium?

Key Idea: How to configure for max robustness for adverse wireless medium?

Answer: Use DSSS, deploy as slow data rate r_b (i.e., as large processing gain g) as the application allows.

$$P_{BER} \leq \exp(-gK) = \exp\left(-\frac{K}{r_b}\right)$$

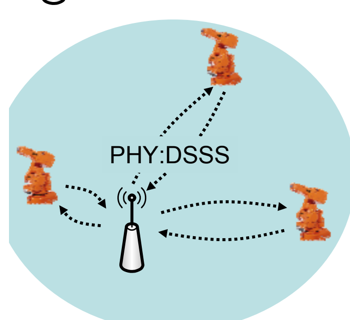
Solution Heuristics



DSSS with low data rate for high robustness

Observation: Centralized, last-hop wireless scheme is preferred

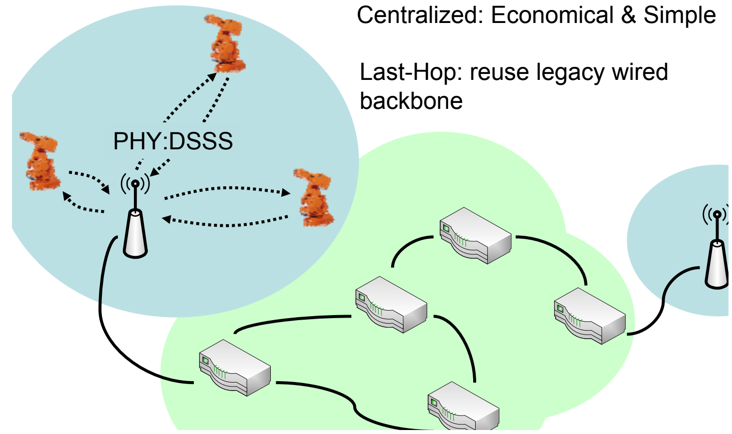
Centralized: Economical & Simple



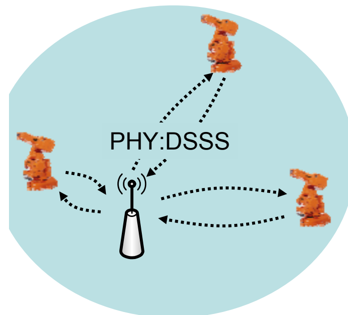
Observation: Centralized, last-hop wireless scheme is preferred

Centralized: Economical & Simple

Last-Hop: reuse legacy wired backbone



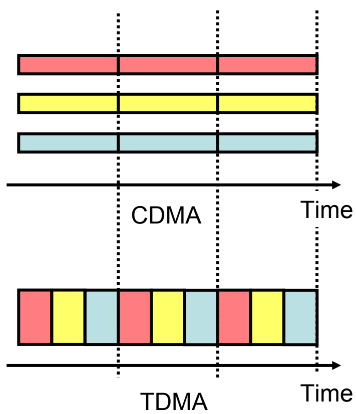
Solution Heuristics



DSSS with low data rate for high robustness

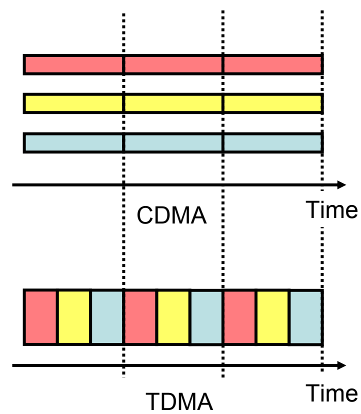
Centralized WLAN paradigm

Observation: CDMA is better than TDMA (e.g., IEEE 802.11 PCF).



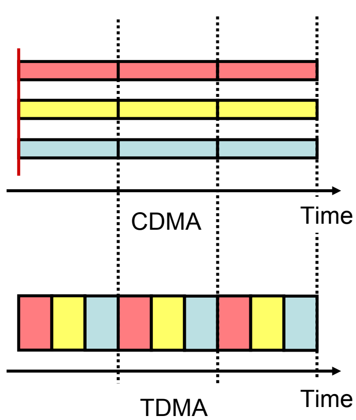
Observation: CDMA is better than TDMA (e.g., IEEE 802.11 PCF).

Smaller overhead under adverse channel conditions



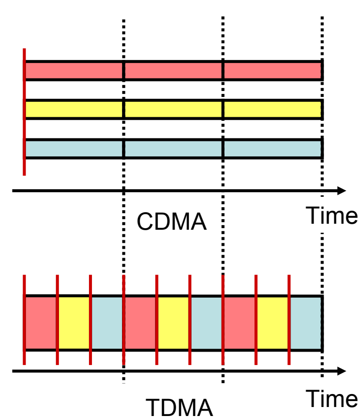
Observation: CDMA is better than TDMA (e.g., IEEE 802.11 PCF).

Smaller overhead under adverse channel conditions



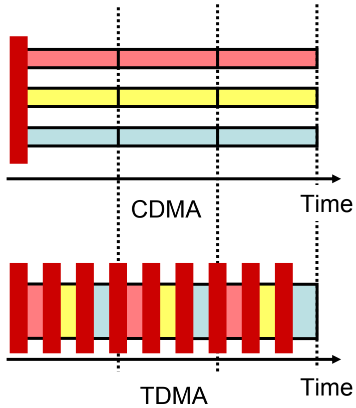
Observation: CDMA is better than TDMA (e.g., IEEE 802.11 PCF).

Smaller overhead under adverse channel conditions



Observation: CDMA is better than TDMA (e.g., IEEE 802.11 PCF).

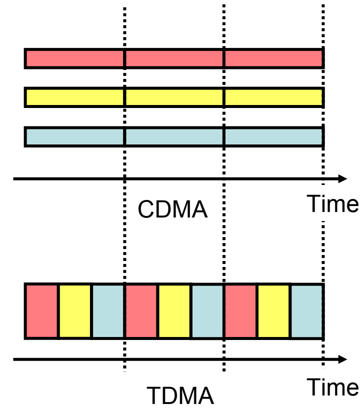
Smaller overhead under adverse channel conditions



Observation: CDMA is better than TDMA (e.g., IEEE 802.11 PCF).

Smaller overhead under adverse channel conditions

Easier to schedule

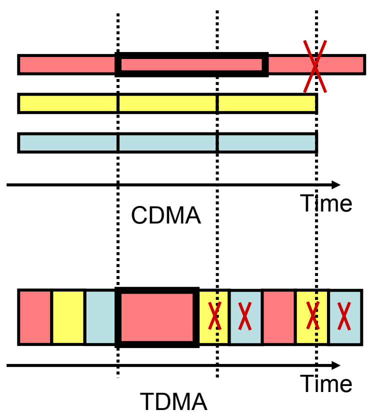


Observation: CDMA is better than TDMA (e.g., IEEE 802.11 PCF).

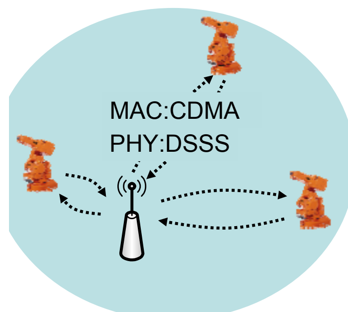
Smaller overhead under adverse channel conditions

Easier to schedule

Better overrun isolation



Solution Heuristics

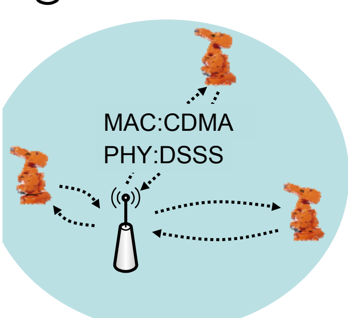


Centralized WLAN paradigm

DSSS with low data rate for high robustness

CDMA instead of TDMA

Solution Heuristics ➔ Choose DSSS-CDMA cell phone network paradigm!



DSSS with low data rate for high robustness

Centralized WLAN paradigm

CDMA instead of TDMA

Simulation and Comparisons

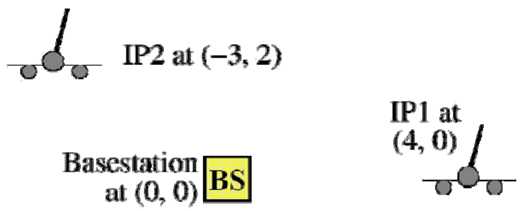
Wireless medium model complies with typical settings for industrial environments [Rappaport02]:

Table 1. Wireless Medium Model

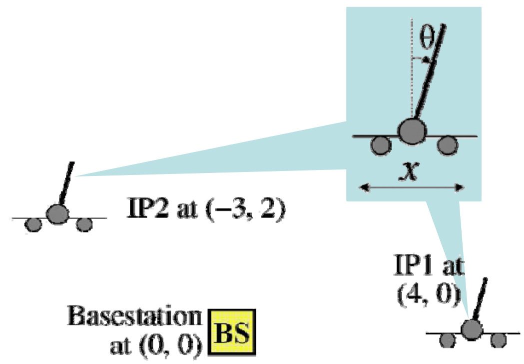
Large-scale path loss model	Log-normal shadowing model with $\beta = 4 \sim 6$, $\sigma = 6.8$dB *
Small-scale fading model	Rayleigh
Multipath max excess delay	90.909nsec
Additive White Gaussian Noise†	Spectral density $= -174$dBm/Hz

∗ β is the path loss exponent, σ is the log-normal standard deviation.
† Typically refers to thermal noise.

A simulated demo showing DSSS-CDMA tolerates RF jamming, while IEEE 802.11b cannot

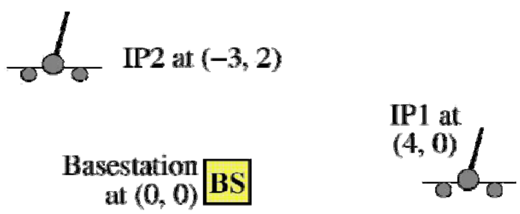


A simulated demo showing DSSS-CDMA tolerates RF jamming, while IEEE 802.11b cannot



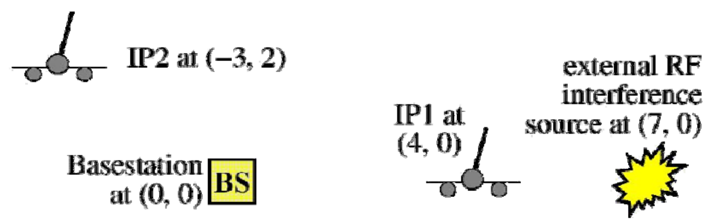
A simulated demo showing DSSS-CDMA tolerates RF jamming, while IEEE 802.11b cannot

Typical industrial environment wireless medium model



A simulated demo showing DSSS-CDMA tolerates RF jamming, while IEEE 802.11b cannot

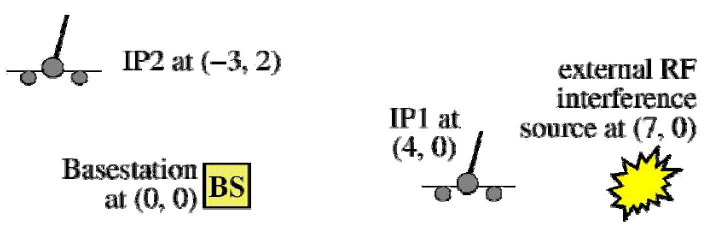
Typical industrial environment wireless medium model



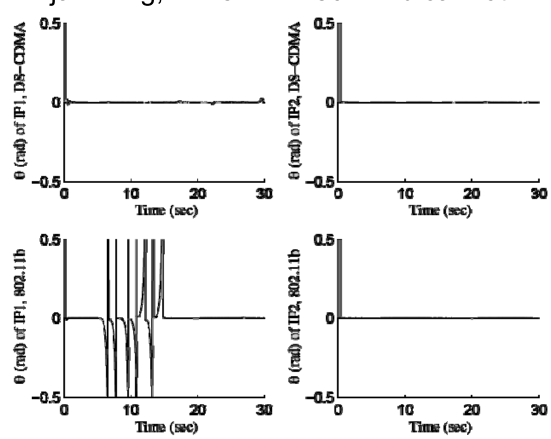
A simulated demo showing DSSS-CDMA tolerates RF jamming, while IEEE 802.11b cannot

Comparison:

- DSSS-CDMA: lowest data rate
- IEEE 802.11b: keep retransmitting



A simulated demo showing DSSS-CDMA tolerates RF jamming, while IEEE 802.11b cannot



A Monte-Carlo simulation showing DSSS-CDMA is more robust than IEEE 802.11a/b

Monte-Carlo simulation setup

- 20m x 20m room, base station at the center
- n (n = 1, …, 100) remote stations, random layout
- 200 trails for each n
- Typical industrial environment wireless medium model

Robustness Method:

- DSSS-CDMA: lowest data rate
- IEEE 802.11a/b: keep retransmitting

A Monte-Carlo simulation showing DSSS-CDMA is more robust than IEEE 802.11a/b

802.11:

- Use the most robust mode:
  - 802.11b (DSSS): 1, 2, 5.5, 11Mbps
  - 802.11a (OFDM): 6, 9, 12, 18, 24, 36, 48, 54Mbps
- Under adverse channel conditions, 802.11 keeps retransmitting (PCF).

DSSS-CDMA

- Deploy as slow data rate as (i.e., as large processing gain g as) the application allows (proposition 1).
- Keep transmitting even under adverse channel conditions.

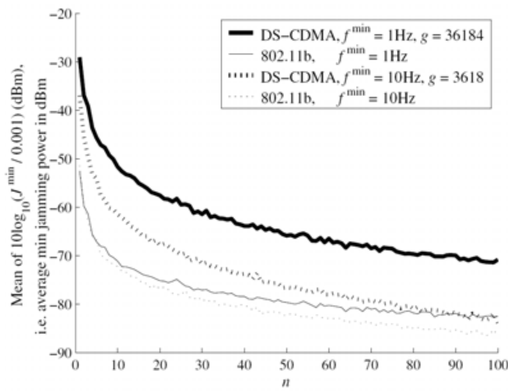
Simulation and Comparisons

Table 2. Physical Layer Settings for Comparisons

	Max trans power*	RF band†
DSSS-CDMA vs. IEEE 802.11b comparison	1watt	2.425 ~ 2.449GHz
DSSS-CDMA vs. IEEE 802.11a comparison	800mw	5.735 ~ 5.795GHz

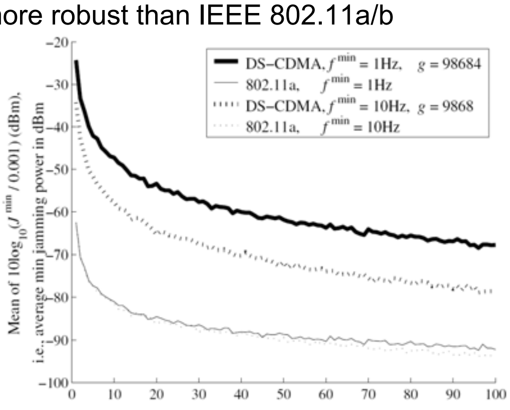
∗ According to FCC regulation.
† According to IEEE 802.11 specification. Note RF bandwidth also decides baseband bandwidth (i.e. chip rate for DSSS and bit rate for OFDM).

A Monte-Carlo simulation showing DSSS-CDMA is more robust than IEEE 802.11a/b



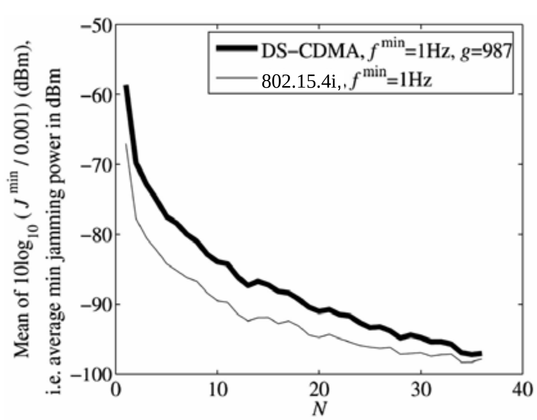
(a) Comparison with IEEE 802.11b

A Monte-Carlo simulation showing DSSS-CDMA is more robust than IEEE 802.11a/b

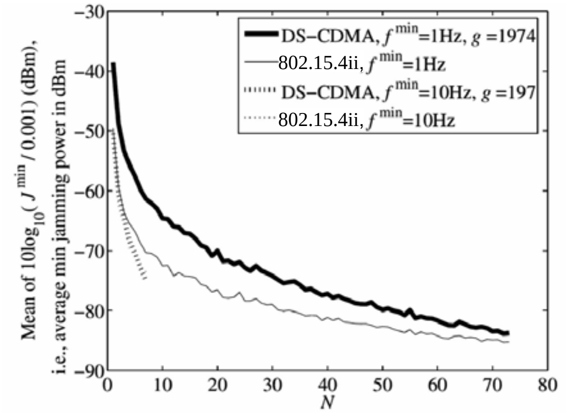


(b) Comparison with IEEE 802.11a

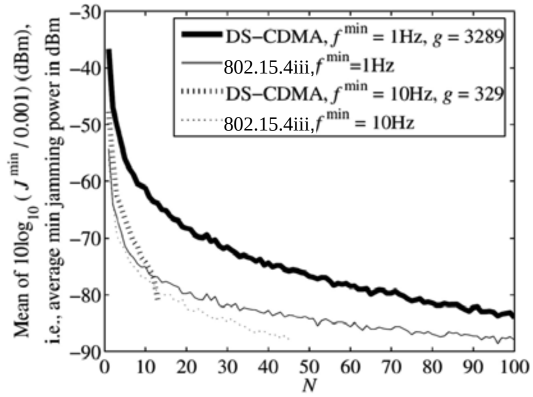
Monte Carlo comparison with IEEE 802.15.4



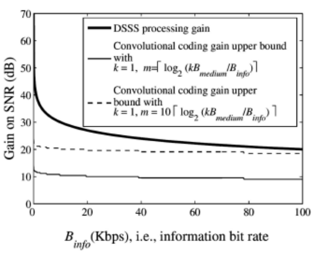
Monte Carlo comparison with IEEE 802.15.4

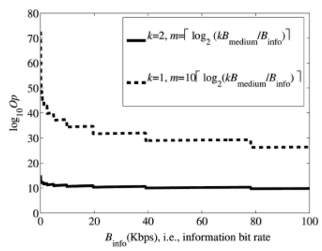


Monte Carlo comparison with IEEE 802.15.4



Feasibility of Convolutional Coding





k input bits, ***m*** shift registers

Conclusion

"DSSS-CDMA Cell Phone Paradigm + Slowest Data Rate" is more robust than "IEEE 802.11 + Retransmission".

Conclusion

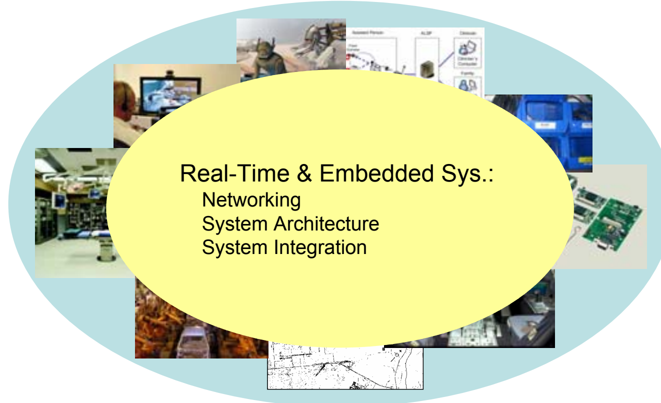
"DSSS-CDMA Cell Phone Paradigm + Slowest Data Rate" is more robust than "IEEE 802.11 + Retransmission".

For real-time wireless LAN, change philosophy from pursuing throughput/delay to pursuing reliability/robustness.

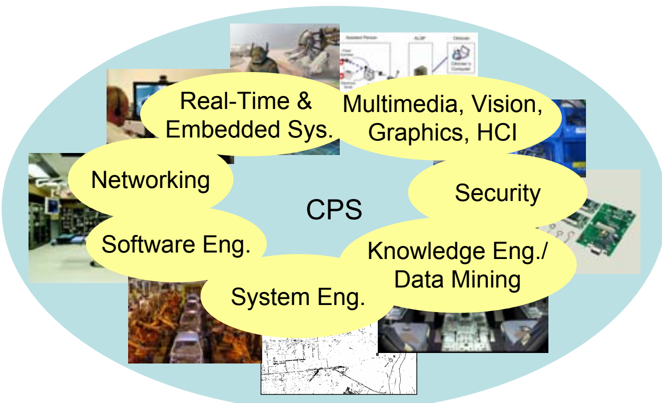
Future Work and Vision

CPS covers a large variety of applications with crucial social and economic impacts

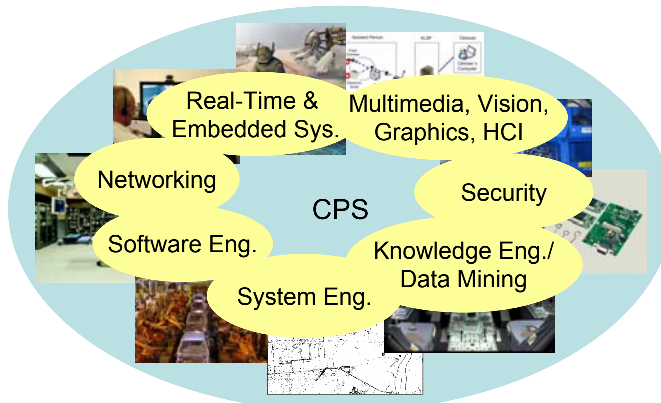
Real-Time and Embedded Systems Infrastructure is one of the CPS challenges



But Real-Time is far from all; CPS' challenges involve nearly every aspects of CS.



CPS and Pervasive Computing will bring Computer Science a promising future.



Thank You!

Publications

Journal Publications:

1. [TMC] **Qixin Wang**, Rong Zheng, Ajay Tirumala, Xue Liu, and Lui Sha, "Lightning: A Hard Real-Time, Fast, and Lightweight Low-End Wireless Sensor Election Protocol for Acoustic Event Localization", (accepted for publication) in IEEE Transactions on Mobile Computing.
2. [TMC07] **Qixin Wang**, Xue Liu, Weiqun Chen, Marco Caccamo, and Lui Sha, "Building Robust Wireless LAN for Industrial Control with the DSSS-CDMA Cell Phone Network Paradigm", in IEEE Transactions on Mobile Computing, vol 6, number 6, June, 2007.
3. [TOSN06] Xue Liu, **Qixin Wang**, Wenbo He, Marco Caccamo, and Lui Sha, "Optimal Real-Time Sampling Rate Assignment for Wireless Sensor Networks", in ACM Transactions on Sensor Networks, vol 2, issue 2, May, 2006.

Conference, Workshop and Other Publications:

4. [RTAS08] **Qixin Wang**, Sathish Gopalakrishnan, Xue Liu, and Lui Sha, "A Switch Design for Real-Time Industrial Networks", (full paper accepted for publication) in Proceedings of the 14th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2008), 2008.
5. [RTSS07] **Qixin Wang**, Xue Liu, Jennifer Hou, and Lui Sha, "GD-Aggregate: A WAN Virtual Topology Building Tool for Hard Real-Time and Embedded Applications", in Proceedings of the 28th IEEE Real-Time Systems Symposium (RTSS 2007), pp. 379-388, Tucson, Arizona, Dec. 3-6, 2007.
6. [HCMDSS07a] Mu Sun, **Qixin Wang**, and Lui Sha, "Building Safe and Reliable MD PnP Systems", in Joint Workshop on High Confidence Medical Devices, Software, and Systems (HCMDSS) and Medical Devices Plug-and-Play (MD PnP), June, 2007.
7. [HCMDSS07b] Jennifer C. Hou, **Qixin Wang**, et. al., "PAS: A Wireless-enabled, Sensor-integrated Personal Assistance System for Independent and Assisted Living", in Joint Workshop on High Confidence Medical Devices, Software, and Systems (HCMDSS) and Medical Devices Plug-and-Play (MD PnP), June, 2007.
8. [ICSMC06] **Qixin Wang**, Wook Shin, Xue Liu, et. al., "I-Living: An Open System Architecture for Assisted Living", (invited paper) in Proc. of IEEE International Conference on Systems, Man, and Cybernetics 2006.
9. [RTSS05] **Qixin Wang**, Xue Liu, Weiqun Chen, Wenbo He, and Marco Caccamo, "Building Robust Wireless LAN for Industrial Control with DSSS-CDMA Cellphone Network Paradigm", in Proc. of the 26th IEEE International Real-Time Systems Symposium (RTSS 2005), Miami, USA, December, 2005. (Power Point)(Poster-Sized Power Point)
10. [ICAS05] Xue Liu, Rong Zheng, Jin Heo, **Qixin Wang**, and Lui Sha, "Timing Control for Web Server Systems Using Internal State Information", in Proc. of Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services (ICAS-ICNS 2005), 2005.
11. [RTSS04] **Qixin Wang**, Rong Zheng, Ajay Tirumala, Xue Liu, and Lui Sha, "Lightning: A Fast and Lightweight Acoustic Localization Protocol Using Low-End Wireless Micro-Sensors", in Proc. of the 25th IEEE International Real-Time Systems Symposium (RTSS 2004), Lisbon, Portugal, December, 2004. (Power Point) (Demo Video)
12. [RTSS03] Xue Liu, **Qixin Wang**, Lui Sha and Wenbo He, "Optimal QoS Sampling Frequency Assignment for Real-Time Wireless Sensor Networks", in Proc. of the 24th IEEE International Real-Time Systems Symposium (RTSS 2003), Cancun, Mexico, December, 2003.
13. [IPSN03] **Qixin Wang**, Wei-Peng Chen, Rong Zheng, Kihwal Lee, and Lui Sha, "Acoustic Target Tracking Using Tiny Wireless Sensor Devices", in Proc. of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN'03), Lecture Notes in Computer Science 2634, Springer, 2003.