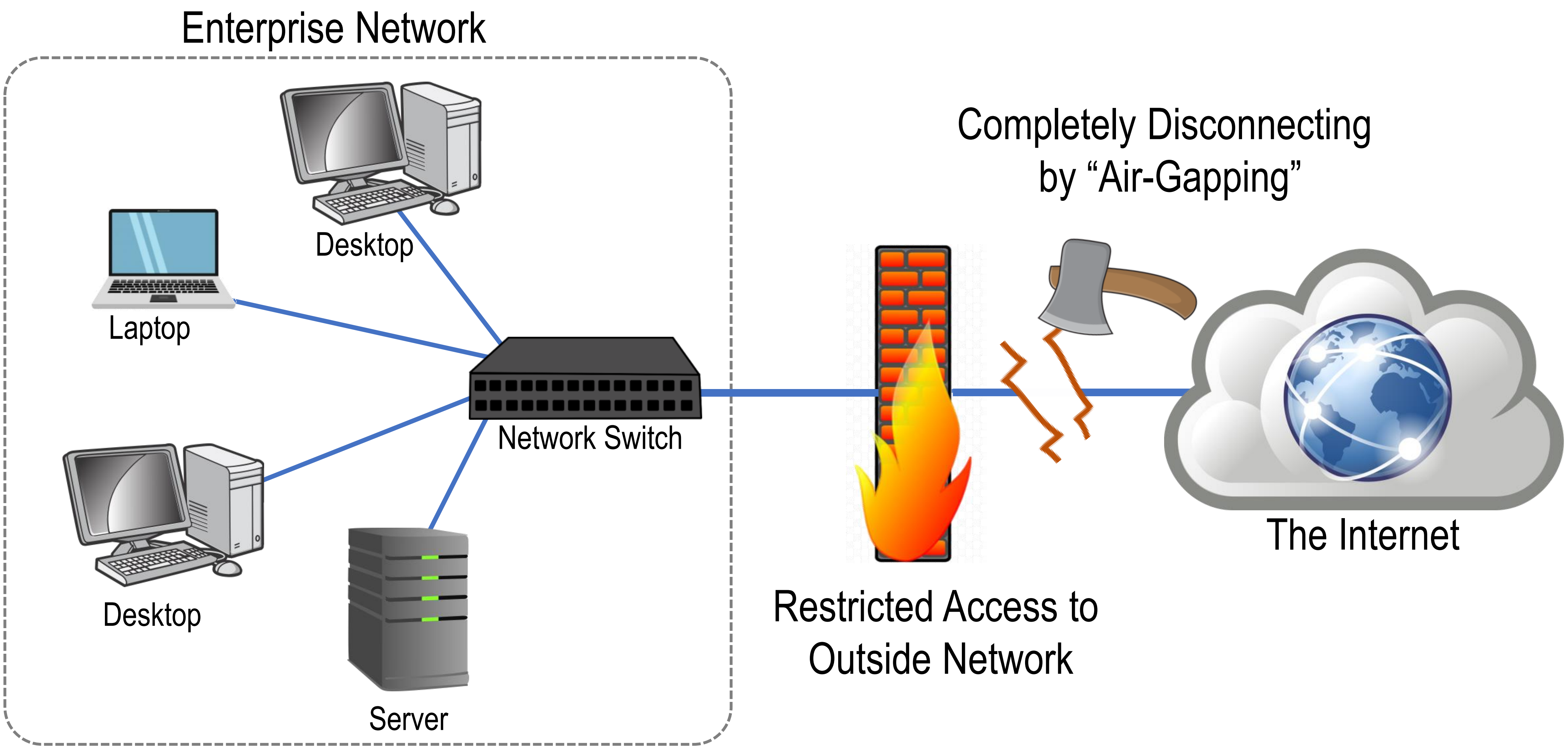
Your Noise, My Signal: Exploiting Switching Noise for Stealthy Data Exfiltration from Desktop Computers

Zhihui Shao[1], **Mohammad A. Islam**[2], and Shaolei Ren[1]

[1]UC Riverside, [2]UT Arlington

Acknowledgement: This work was supported by the U.S. NSF under grants CNS-1551661, ECCS-1610471, and CNS-1910208.



Enterprise Network

Desktop

Laptop

Network Switch

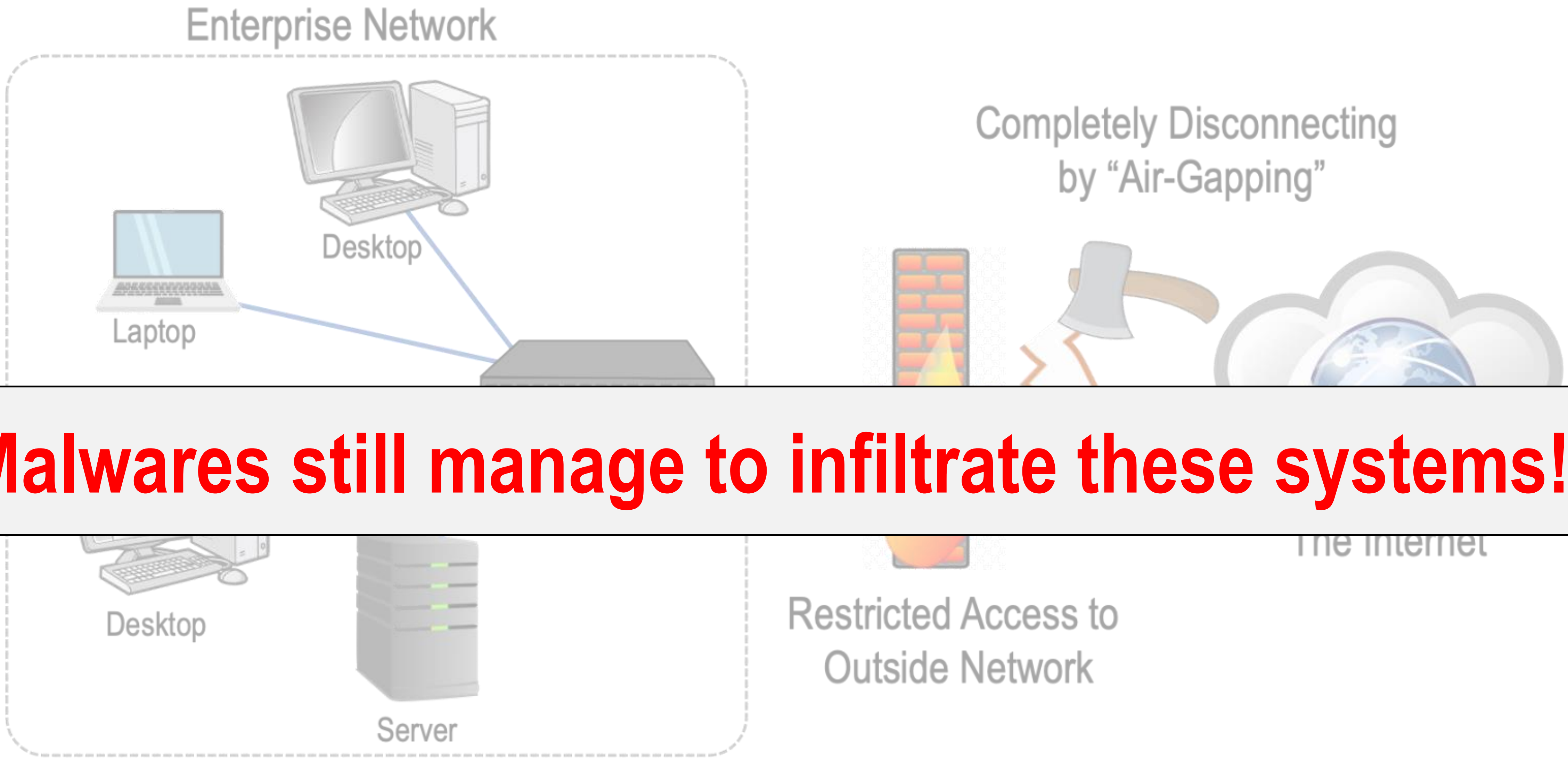
Desktop

Server

Completely Disconnecting by "Air-Gapping"

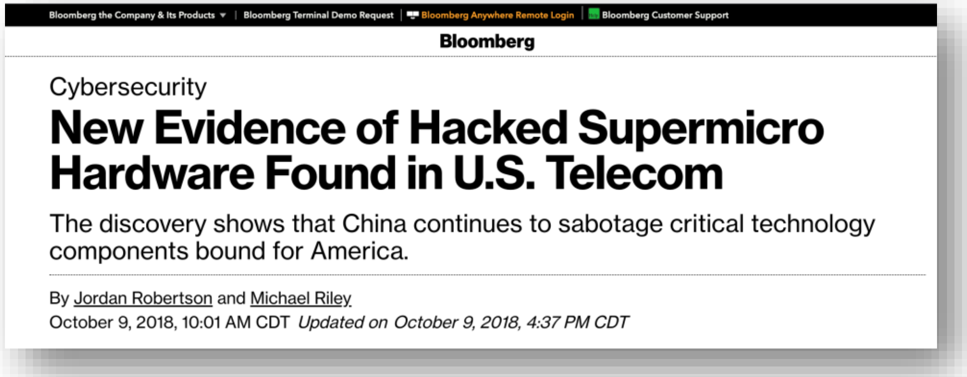
Restricted Access to Outside Network

The Internet


Enterprise Network
Desktop
Laptop
Desktop
Server
Completely Disconnecting by “Air-Gapping”
Restricted Access to Outside Network
The Internet

Malwares still manage to infiltrate these systems!

- Supply chain attacks


Bloomberg the Company & Its Products ▾ | Bloomberg Terminal Demo Request | Bloomberg Anywhere Remote Login | Bloomberg Customer Support
Bloomberg

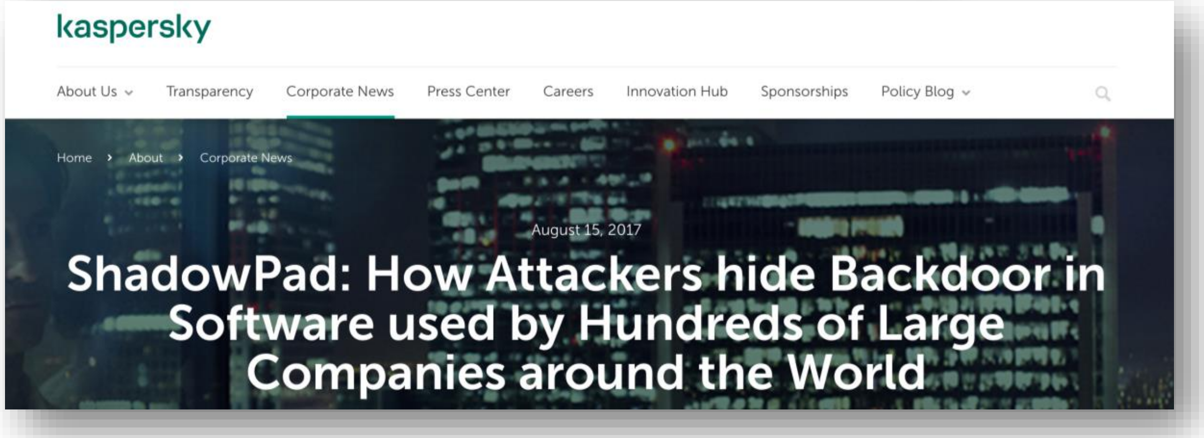
Cybersecurity

New Evidence of Hacked Supermicro Hardware Found in U.S. Telecom

The discovery shows that China continues to sabotage critical technology components bound for America.

By Jordan Robertson and Michael Riley
October 9, 2018, 10:01 AM CDT *Updated on October 9, 2018, 4:37 PM CDT*

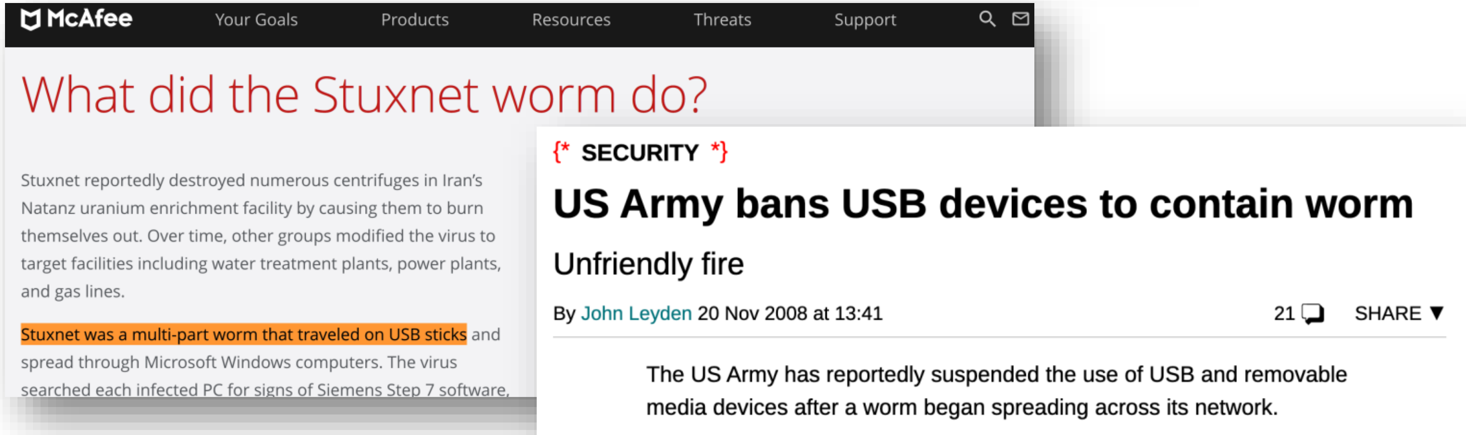
- HW/SW backdoors



August 15, 2017

ShadowPad: How Attackers hide Backdoor in Software used by Hundreds of Large Companies around the World

- Portable drives



What did the Stuxnet worm do?

Stuxnet reportedly destroyed numerous centrifuges in Iran's Natanz uranium enrichment facility by causing them to burn themselves out. Over time, other groups modified the virus to target facilities including water treatment plants, power plants, and gas lines.

Stuxnet was a multi-part worm that traveled on USB sticks and spread through Microsoft Windows computers. The virus searched each infected PC for signs of Siemens Step 7 software,

{* SECURITY *}

US Army bans USB devices to contain worm

Unfriendly fire

By John Leyden 20 Nov 2008 at 13:41

The US Army has reportedly suspended the use of USB and removable media devices after a worm began spreading across its network.

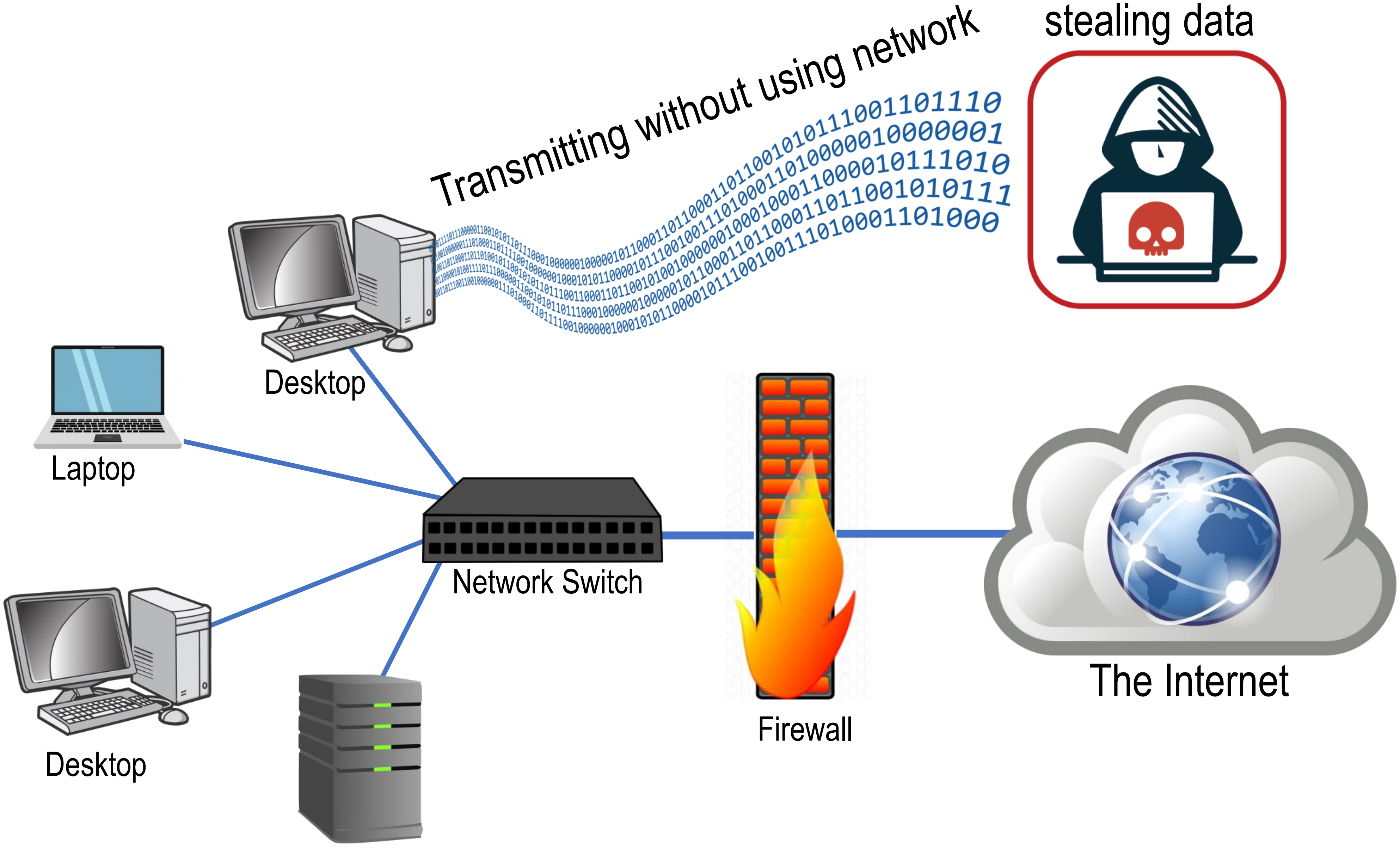
- And many other ways…

Data exfiltration remains a challenge!

- Getting in, the **infiltration**, can be a “one time” incident
- Getting stolen data out, the **exfiltration**, is long-term
  - Infiltration methods are not suitable for exfiltration
  - Cannot use the network

How to send data without using the network?

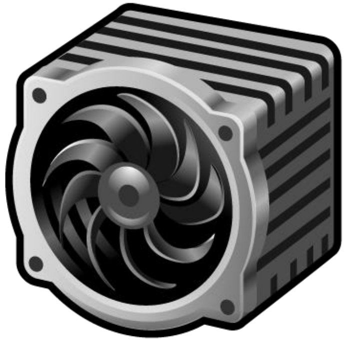
Focus of our work!



Stealthiness

Fast Data Rate

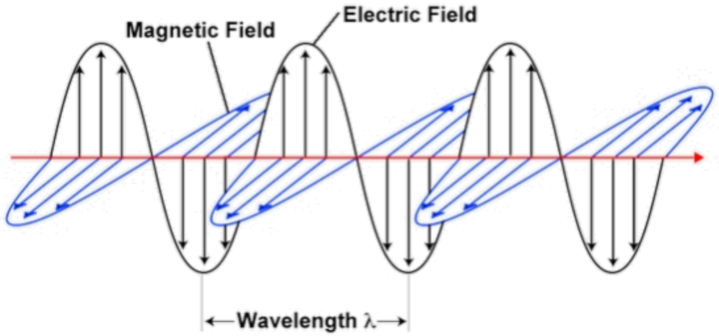
Data transmission without using network



Fan
Noise

Generated
Heat

Status
LED

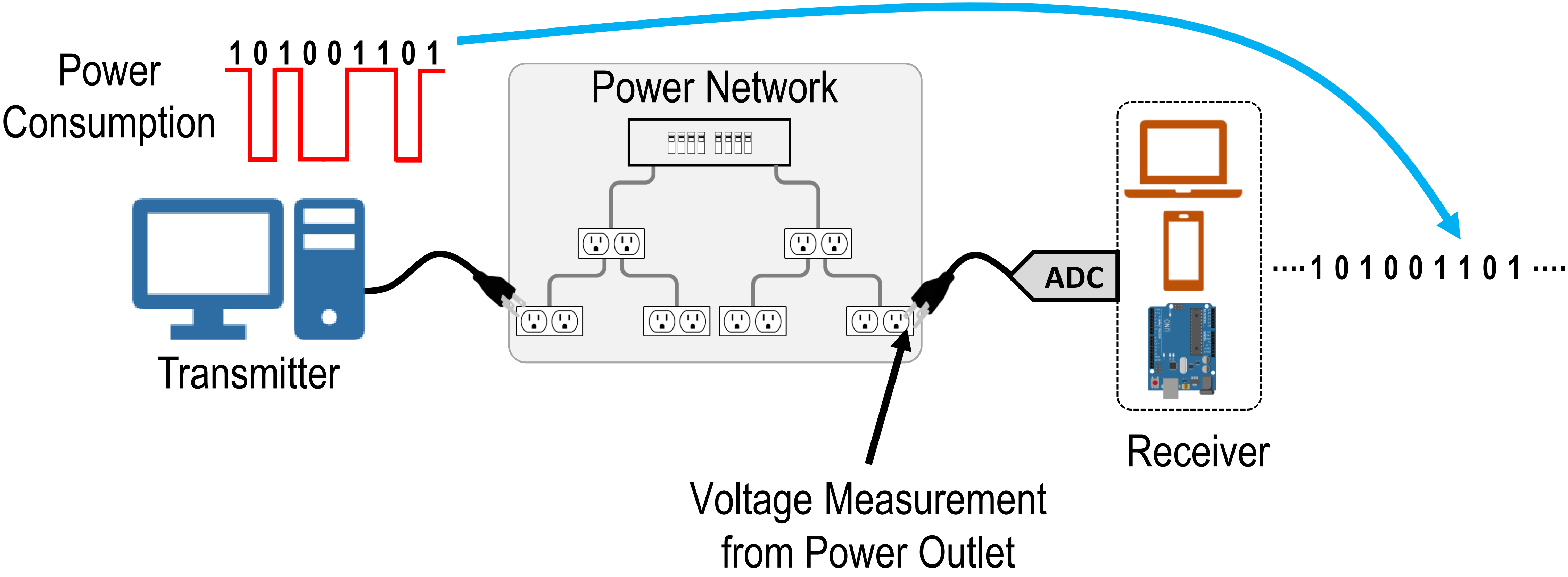


Electromagnetic
Emanation

....

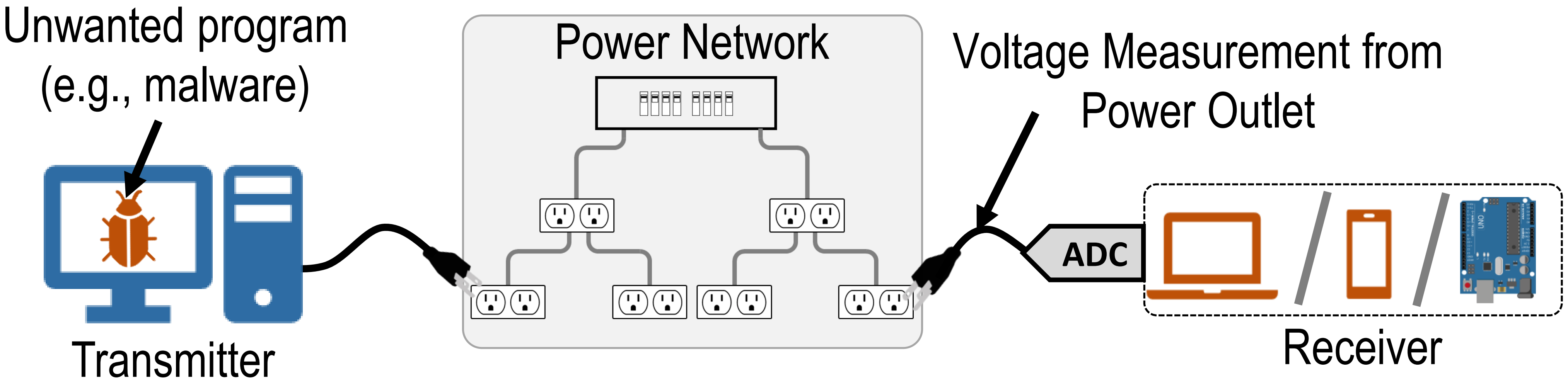
Our approach

- We vary computer power consumption to send data over the power network
- We extract data from voltage measurements at other outlet



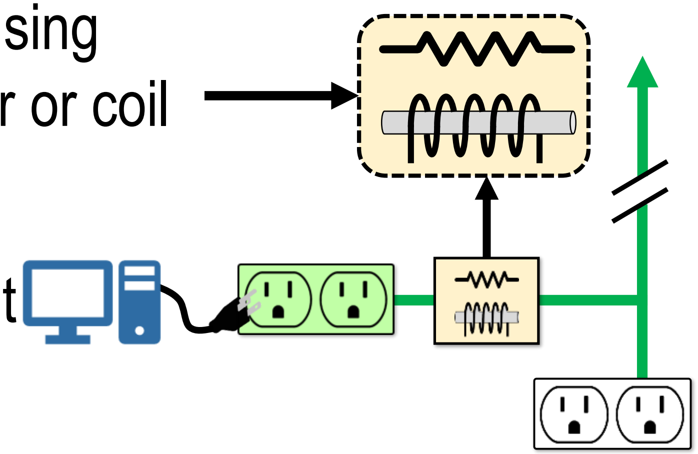
Threat model

- Transmitter
  - Target is infected with malware that can steal sensitive data
  - Malware modulates the power by running CPU intensive instructions
- Receiver
  - Connected to a power outlet within the same power network as the transmitter
  - Equipped with an ADC to collect voltage measurements



Why use voltage measurement?

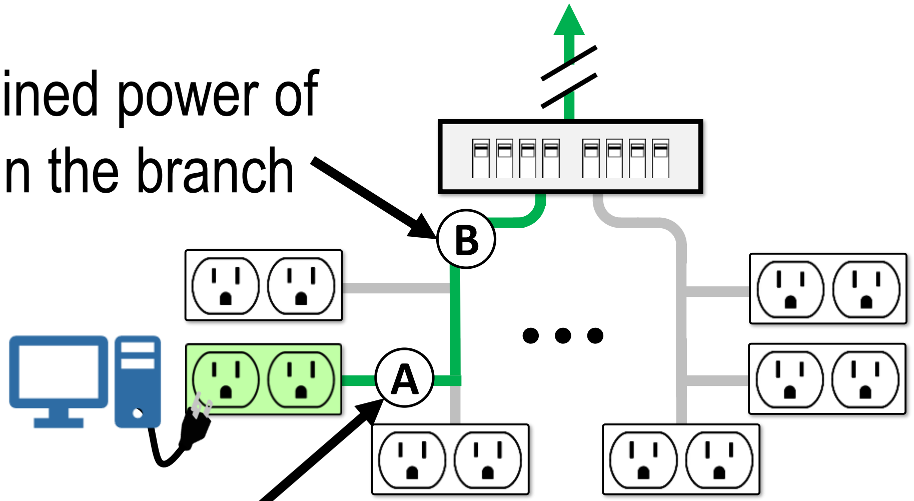
- Limitations of prior works that use *traditional* power measurement



Sensing register or coil

Target

Requires physically tempering the power outlet/cable



Senses combined power of every outlet in the branch

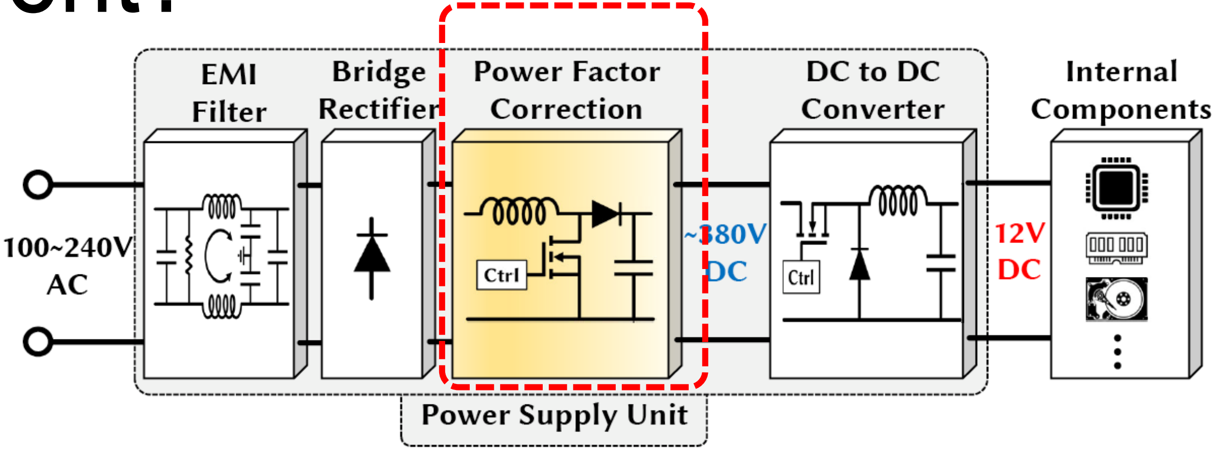
Target

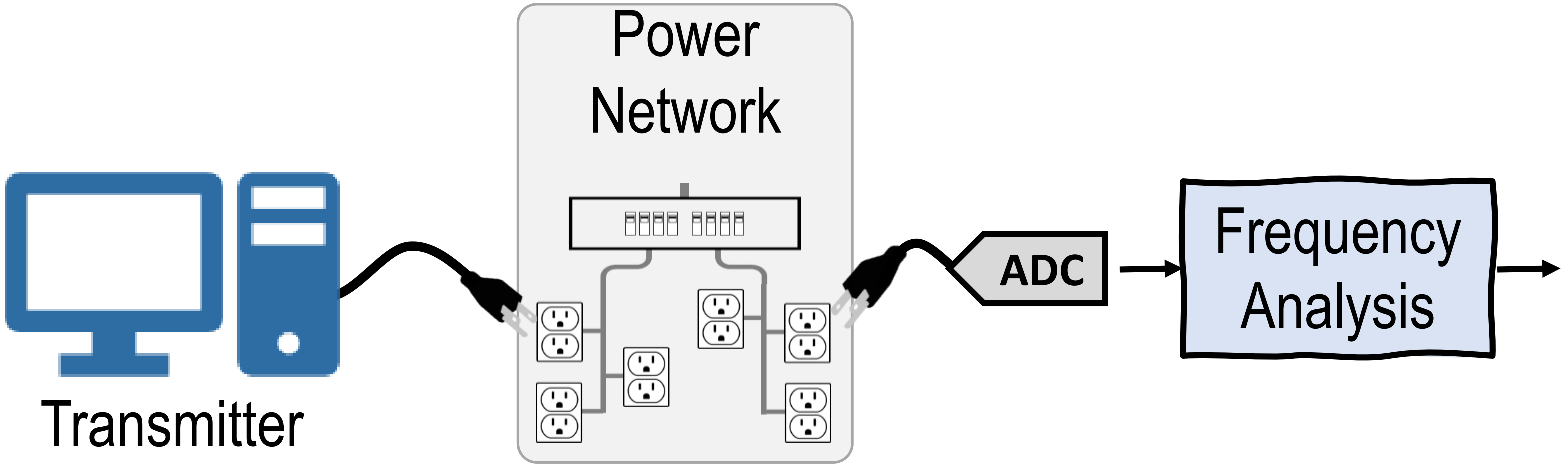
Only suitable location for sensing

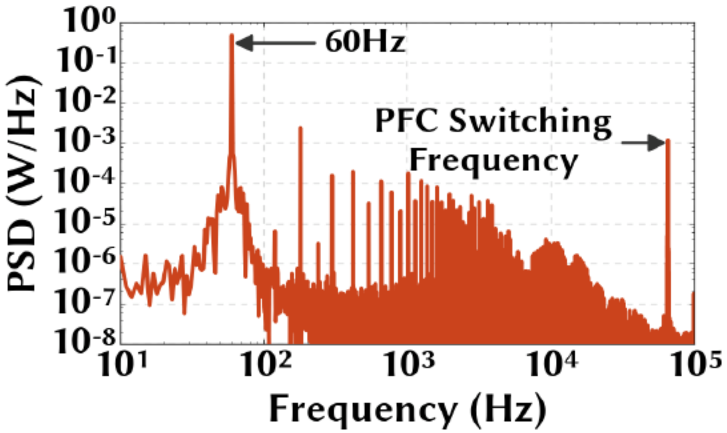
Requires targeted sensor placement

How to use voltage measurement?

- Power factor correction (PFC) circuits is ubiquitously available in desktop computer power supply unit
- PFC creates high-frequency voltage ripples due to rapid switching



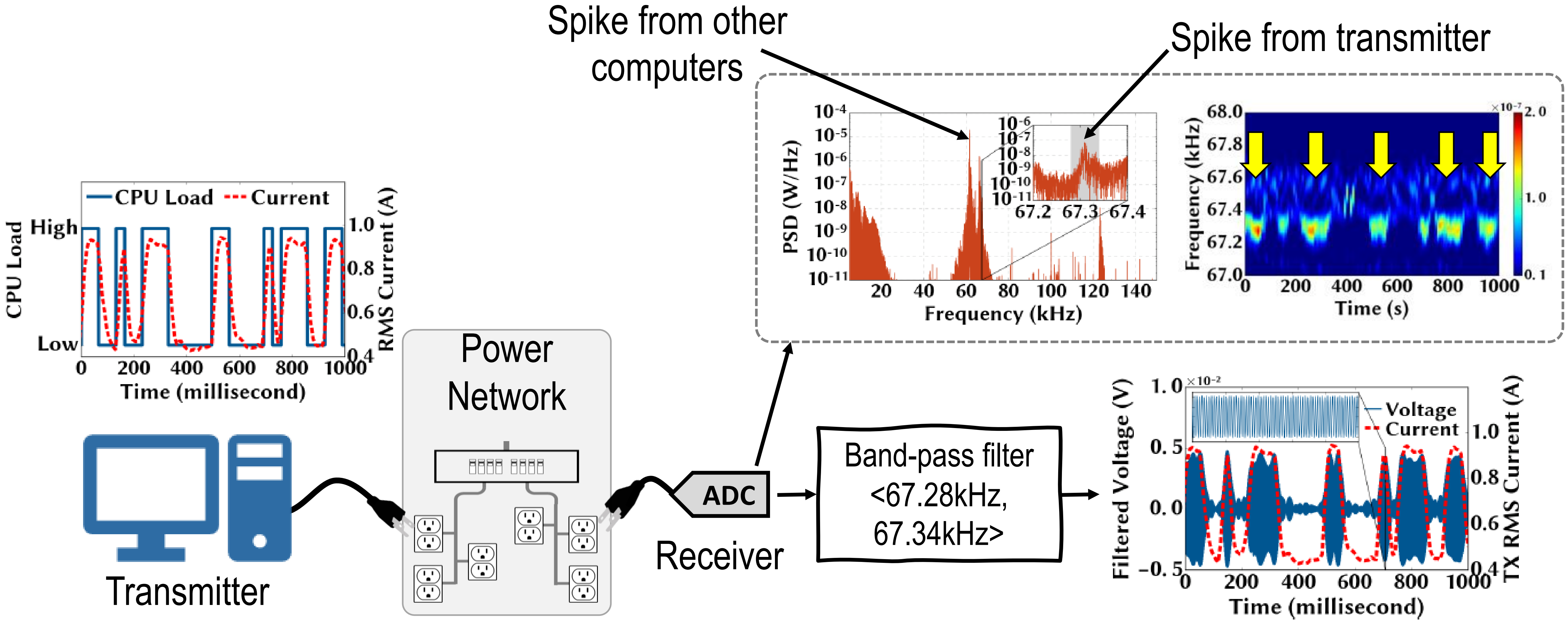




- PFC switching frequency varies with power supplies

Sending data using voltage measurement

- Transmitter and receiver are in a lab, ~55 feet away from each other



Spike from other computers

Spike from transmitter

Power Network

Transmitter

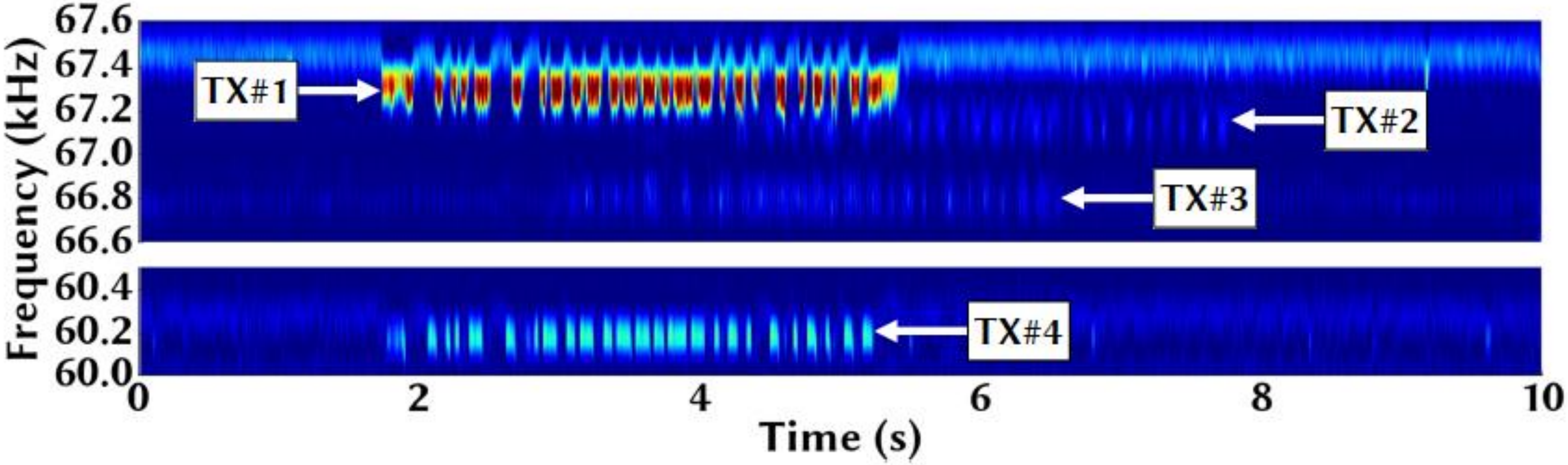
Receiver

ADC

Band-pass filter <67.28kHz, 67.34kHz>

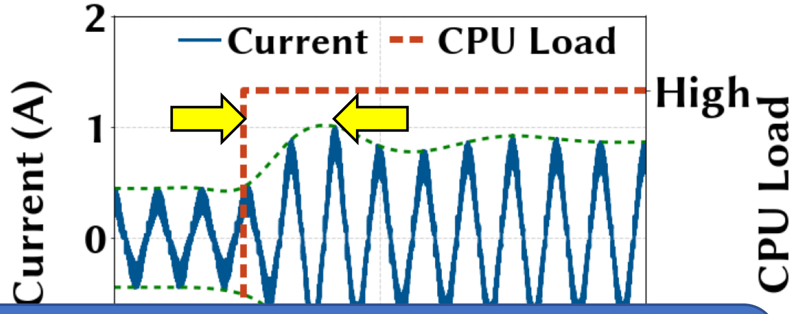
Simultaneous transmission

- 4 transmitters sending data to a single receiver



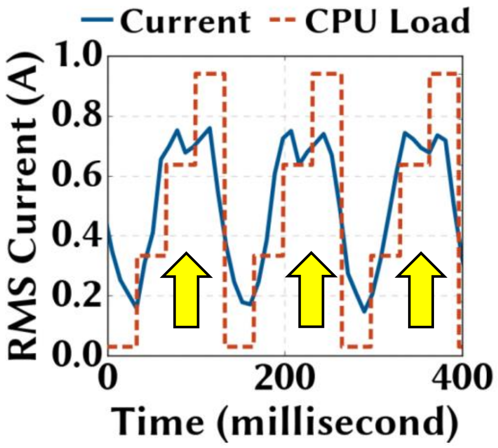
Bit rate

- Symbol rate
  - Limited by lag in response to CPU load change
  - Maximum symbol rate is ~30 symbols/second

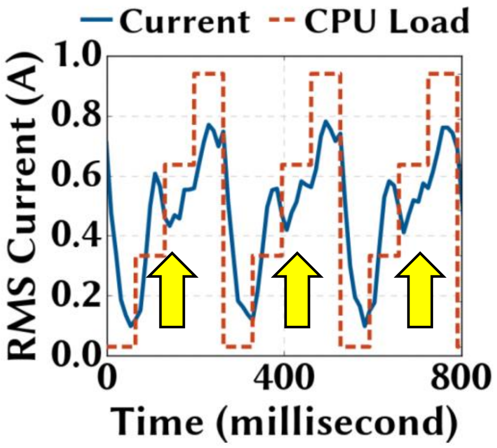


Maximum bit rate ~30 bits/s

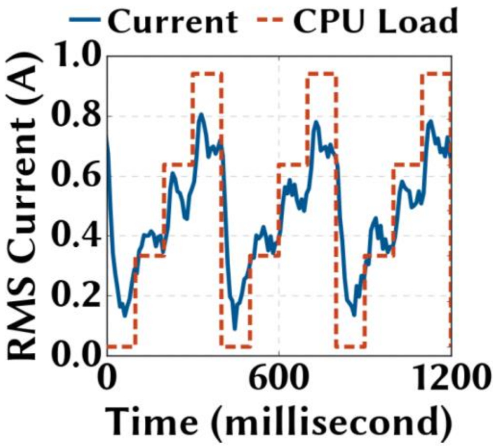
- Bits per symbol
  - Current needs time to settle
  - One bit/symbol



Symbol length
33 millisecond

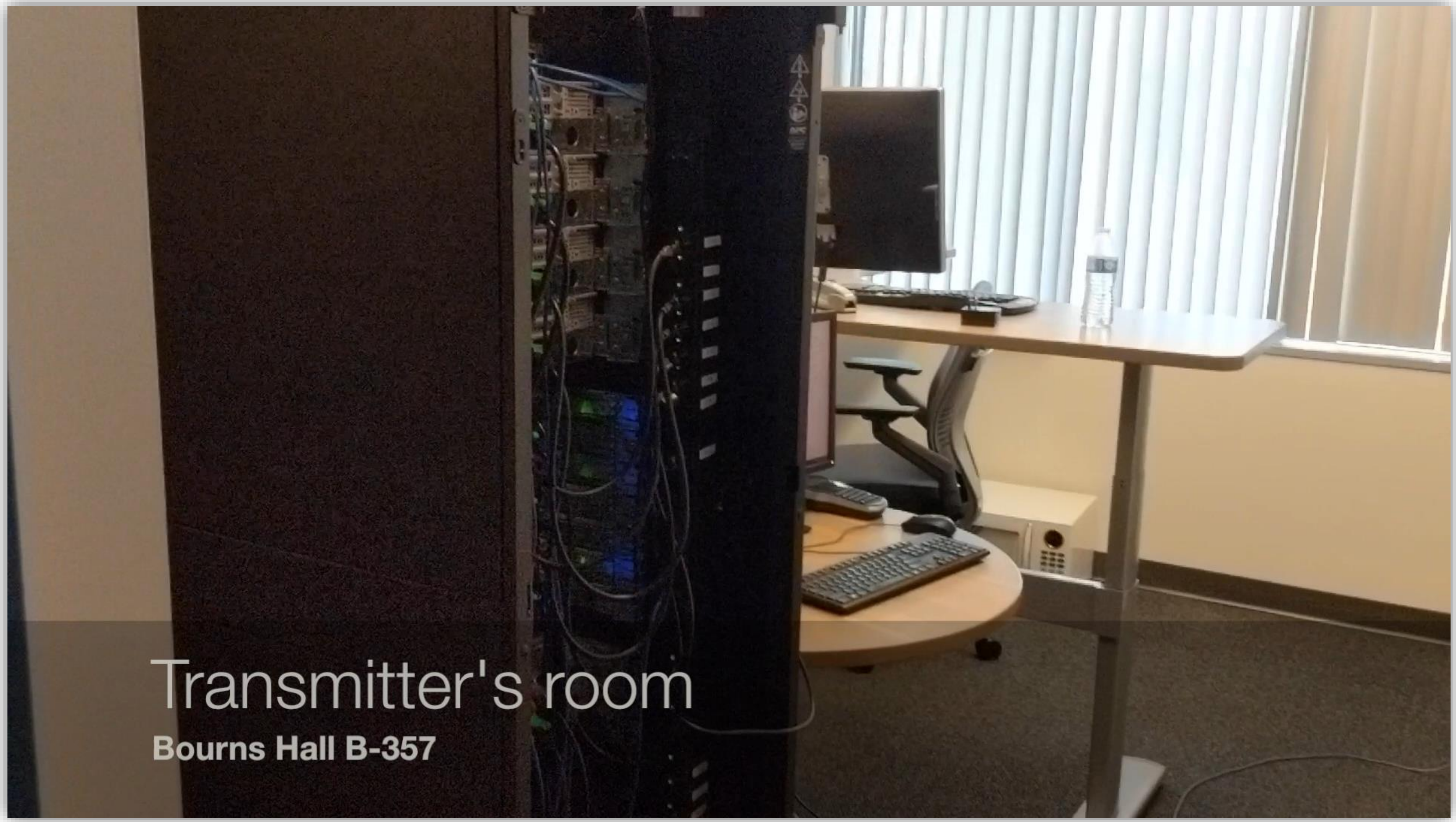


Symbol length
66 millisecond



Symbol length
100 millisecond

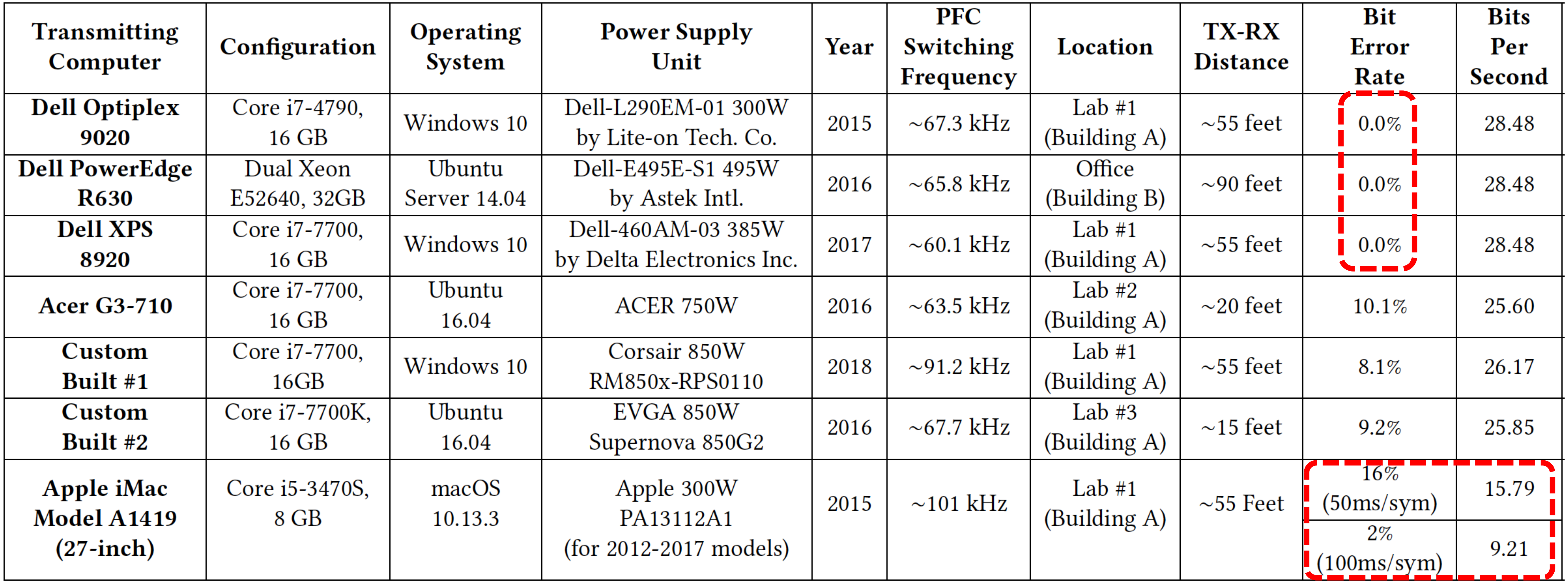
Demo



Transmitter's room

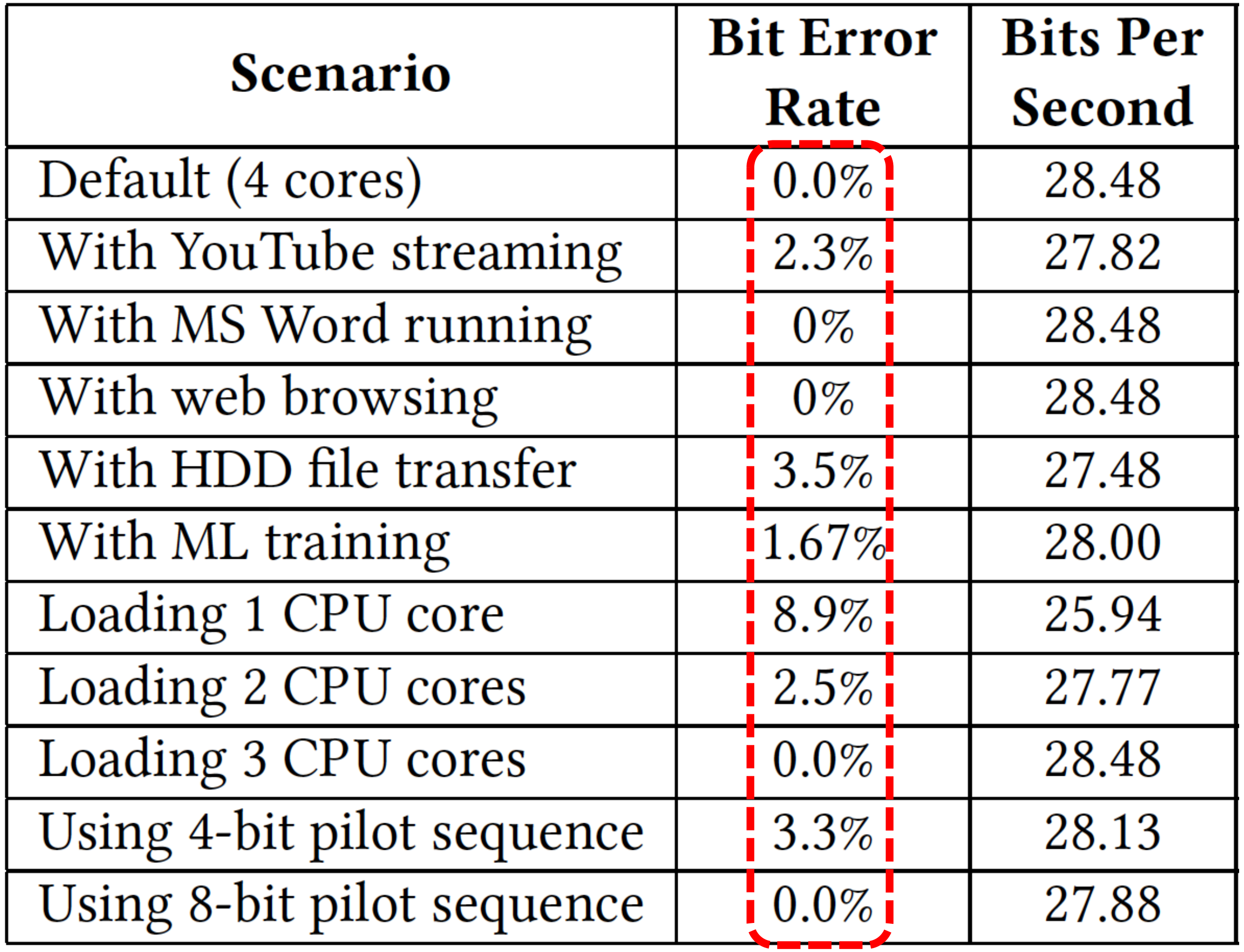
Bourns Hall B-357

Experiments with different computers and locations



Transmitting Computer	Configuration	Operating System	Power Supply Unit	Year	PFC Switching Frequency	Location	TX-RX Distance	Bit Error Rate	Bits Per Second
Dell Optiplex 9020	Core i7-4790, 16 GB	Windows 10	Dell-L290EM-01 300W by Lite-on Tech. Co.	2015	~67.3 kHz	Lab #1 (Building A)	~55 feet	0.0%	28.48
Dell PowerEdge R630	Dual Xeon E52640, 32GB	Ubuntu Server 14.04	Dell-E495E-S1 495W by Astek Intl.	2016	~65.8 kHz	Office (Building B)	~90 feet	0.0%	28.48
Dell XPS 8920	Core i7-7700, 16 GB	Windows 10	Dell-460AM-03 385W by Delta Electronics Inc.	2017	~60.1 kHz	Lab #1 (Building A)	~55 feet	0.0%	28.48
Acer G3-710	Core i7-7700, 16 GB	Ubuntu 16.04	ACER 750W	2016	~63.5 kHz	Lab #2 (Building A)	~20 feet	10.1%	25.60
Custom Built #1	Core i7-7700, 16GB	Windows 10	Corsair 850W RM850x-RPS0110	2018	~91.2 kHz	Lab #1 (Building A)	~55 feet	8.1%	26.17
Custom Built #2	Core i7-7700K, 16 GB	Ubuntu 16.04	EVGA 850W Supernova 850G2	2016	~67.7 kHz	Lab #3 (Building A)	~15 feet	9.2%	25.85
Apple iMac Model A1419 (27-inch)	Core i5-3470S, 8 GB	macOS 10.13.3	Apple 300W PA13112A1 (for 2012-2017 models)	2015	~101 kHz	Lab #1 (Building A)	~55 Feet	16% (50ms/sym)	15.79
								2% (100ms/sym)	9.21

Experiments under different scenarios



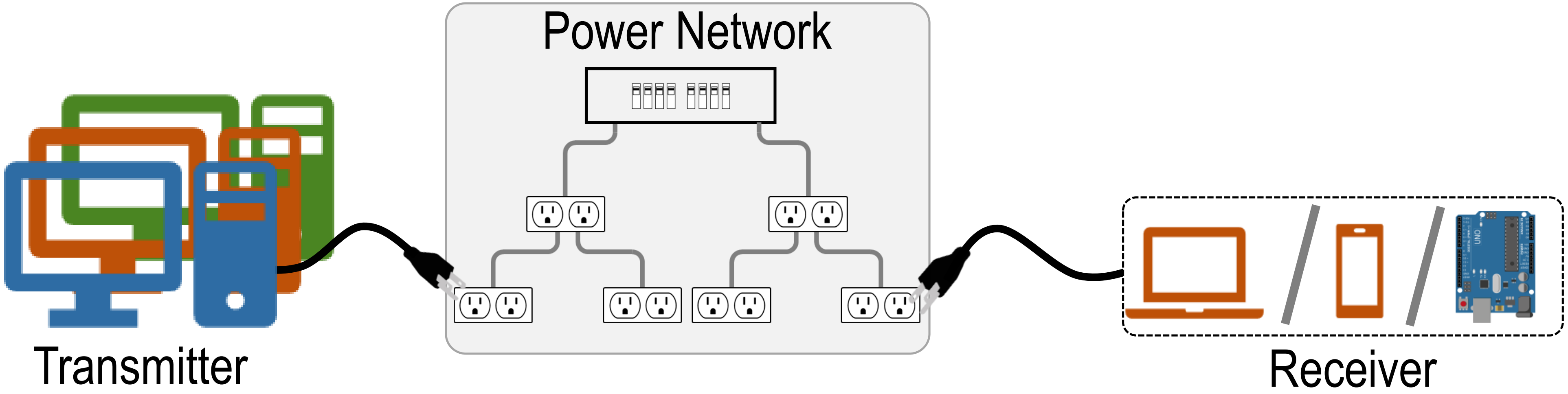
Scenario	Bit Error Rate	Bits Per Second
Default (4 cores)	0.0%	28.48
With YouTube streaming	2.3%	27.82
With MS Word running	0%	28.48
With web browsing	0%	28.48
With HDD file transfer	3.5%	27.48
With ML training	1.67%	28.00
Loading 1 CPU core	8.9%	25.94
Loading 2 CPU cores	2.5%	27.77
Loading 3 CPU cores	0.0%	28.48
Using 4-bit pilot sequence	3.3%	28.13
Using 8-bit pilot sequence	0.0%	27.88

Possible defense strategies

- Eliminate PFC-induced switching noise
    - Require change in a mature power electronics design
- Preventing switching noise from entering the power network
    - Use UPS or power-line filters
- Suppressing Malware Activities
    - Randomize power consumption of a computer

Key take away!

- **Your Noise** is **My Signal**



Thank you!

- Please contact us with questions and comments.
    - Zhihui Shao (zshao006@ucr.edu)
    - Mohammad A. Islam (mislam@uta.edu)
    - Shaolei Ren (sren@ece.ucr.edu)