Wireless Security: A Perspective Some Detailed Solutions, Some Directions for Exploration

(aka. What We've Done Wrong, and Some of What We Can Do About It…)

Wade Trappe ***Yih-Chun Hu***

Lecture Agenda

- Tales from the dark side of security
  - Cellular
  - 802.11
  - Routing in MANETs
  - Interference
- What is the problem? And what YOU can do about it…
  - Crypto was amateur
  - Performance is still important!
  - Layering leads to “stovepipe” solutions
  - Disconnection between state-of-the-art in research and the real-world
  - Some basic tenets of security design
- Some specific solutions in the research world
  - Physical Layer Security
  - Radiometrics
  - Interference Resilience
  - Secure Routing
  - Location Privacy
- Where should we go from here? A sampling of research ideas for people with time to spare…

Tales from the Dark Side of Security: Some Exploits

Generic examples of security flaws in real systems illustrates the challenge of getting security right

- Prepayment in Electricity Meter Systems:
  - Present a (purchased) digital token to a power meter.
  - Digital token would convey an ID so it could not be duplicated or forged…
  - Problem was that the rate information was not protected
- Bank Fraud:
  - A bank would allow customers to present a bank card which had a PIN code encrypted and stored on the magnetic strip
  - Teller had a copy of the encryption key and could check the PINs.
  - Flaw in design: adversary could alter the account number on the card to someone else's, while using his own PIN number… he would check out ok… but the money would be drawn from someone else's account!
  - Flaw in design: PIN number was not connected to account #.

Wireless systems have not faired well in terms of security design

- Cellular Message Encryption Algorithm (CMEA) was deeply flawed
- 802.11 systems, when originally deployed:
  - Were shipped with security disabled
  - Offered SSID/MAC address filtering as security
  - WEP was seriously flawed
- Routing protocols are hard to get right
  - AODV is inherently insecure
  - Its secure variants (ARAN, SAODV) have not done much better
- The wireless medium is inherently more challenging
  - Eavesdropping is trivial and impossible to detect
  - Open, broadcast medium
    - *Jamming is possible*
- The wireless product space is more diverse
  - Highly programmable platforms available
  - Easy to create one's own device and use it

Cellular security algorithms were poorly designed, leading to numerous attacks

- The Telecommunications Industry Association proposed four cryptographic primitives for use in North America (1995, all are now considered weak):
  - CAVE: A mixing function used for authentication and key generation
  - XOR masking used for voice privacy
  - ORYX: an LFSR-based stream cipher
  - CMEA (Control Message Encryption Algorithm): a block cipher to encrypt control channel
- Consider CMEA:
  - CMEA is its own inverse (every key is a "weak key")
  - CMEA encrypts short blocks, but cellular telephony did not employ CFB, or random IVs→ codebook attacks are a threat (consider there are only 10 digits!)
  - LSB of plaintext is leaked
  - Internal T-box has skewed statistical distribution (reduces search space significantly)
  - Chosen-plaintext attack can succeed with 338 chosen plaintexts and very little work
  - Known plaintext attacks: 3-byte version succeeds with 80 known texts and ~2^{32} complexity; 2-byte attacks only need 4 known plaintexts (undermining IS-95)
- Compromise of control channel can lead to compromise of confidential information shared over control channel:
  - PIN numbers, credit card numbers, bank account information
  - Digits dialed by users might reveal user calling patterns

Early 802.11 used SSID/MAC address filtering, which could not achieve any security

- SSID:
  - AP periodically broadcasts SSID in a beacon.
  - End station listens to these broadcasts and chooses an AP to associate with based upon its SSID.
  - Use of SSID – weak form of security as beacon management frames on 802.11 WLAN are always sent in the clear.
  - A hacker can use analysis tools (eg. AirMagnet, Netstumbler, AiroPeek) to identify SSID.
  - Some vendors use default SSIDs which are pretty well known (eg. CISCO used tsunami)
- MAC Address Filtering: The system administrator can specify a list of MAC addresses that can communicate through an access point.
  - Increases Administrative overhead
  - Determined hackers can still break it by sniffing MAC addresses and spoofing MAC addresses

Early 802.11 proposed WEP to address security concerns, but design was inherently weak

- Designed to provide confidentiality to a wireless network similar to that of standard LANs.
- WEP is essentially the RC4 symmetric key cryptographic algorithm (same key for encrypting and decrypting).
  - Transmitting station concatenates 40 bit key with a 24 bit Initialization Vector (IV) to produce pseudorandom key stream.
  - Plaintext is XORed with the pseudorandom key stream to produce ciphertext.
  - Ciphertext is concatenated with IV and transmitted over the Wireless Medium.
  - Receiving station reads the IV, concatenates it with the secret key to produce local copy of the pseudorandom key stream.
  - Received ciphertext is XORed with the key stream generated to get back the plaintext.
- WEP has been broken! Walker (Oct 2000), Borisov et. al. (Jan 2001), Fluhrer-Mantin -Shamir (Aug 2001).
- Unsafe at any key size : Testing reveals WEP encapsulation remains insecure whether its key length is 1 bit or 1000 or any other size.

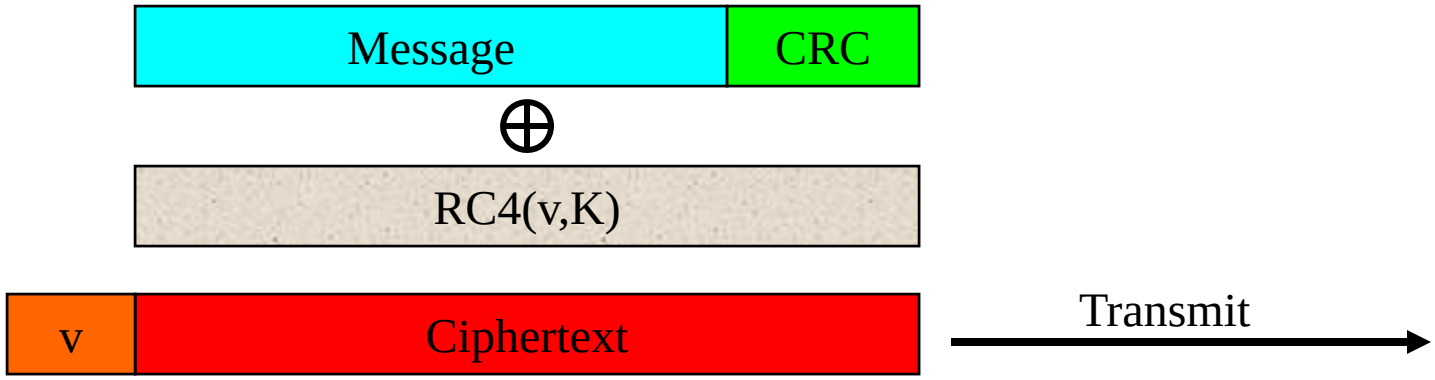
The basic WEP packet included checksums, RC4 and an IV field

- WEP relies on a shared key K between communicating parties

1. **Checksum:** For a message M, we calculate c(M). The plaintext is P={M,c(M)}
2. **Encryption:** The plaintext is encrypted using RC4. RC4 requires an initialization vector (IV) v, and the key K. Output is a stream of bits called the keystream. Encryption is XOR with P.

$$C = P \oplus RC4(v, K)$$

3. **Transmission:** The IV and the ciphertext C are transmitted.



WEP was intended to provide three main security goals so as to be "Equivalent" to wired security

- WEP had three main security goals:
  - Confidentiality: Prevent eavesdropping
  - Access Control: Prevent inappropriate use of 802.11 network, such as facilitate dropping of not-authorized packets
  - Data Integrity: Ensure that messages are not altered or tampered with in transit
- The basic WEP standard uses a 40-bit key (with 24bit IV)
- Additionally, many implementations allow for 104-bit key (with 24bit IV)
- None of the three goals are provided in WEP due to serious security design flaws and the fact that it is easy to eavesdrop on WLAN

A basic flaw in WEP was Vernam Key Stream Reuse

- Vernam-style stream ciphers are susceptible to attacks when same IV and key are reused:

$$C_1 = P_1 \oplus RC4(v, K)$$
$$C_2 = P_2 \oplus RC4(v, K)$$
$$C_1 \oplus C_2 = P_1 \oplus RC4(v, K) \oplus P_2 \oplus RC4(v, K)$$
$$= P_1 \oplus P_2$$

- Particularly weak to known plaintext attack: If P_1 is known, then P_2 is easy to find (as is RC4).
  - This might occur when contextual information gives P_1 (e.g. application-level or network-level information reveals information)
- Even so, there are techniques to recover P_1 and P_2 when just (P_1 XOR P_2) is known (frequency analysis, crib dragging)
  - Example, look for two texts that XOR to same value

Vernam key stream reuse was inadequately prevented in WEP's design

- WEP's engineers were aware (it seems??) of this weakness and required a per-packet IV strategy to vary key stream generation
- Problems:
  - Keys, K, typically stay fixed and so eventual reuse of IV means eventual repetition of keystream!!
  - IVs are transmitted in the clear, so its trivial to detect IV reuse
  - Many cards set IV to 0 at startup and increment IV sequentially from there
  - Even so, the IV is only 24 bits!
- Calculation: Suppose you send 1500 byte packets at 5Mbps, then 2^{24} possible IVs will be used up in 11.2 hours!
- Even worse: we should expect to see at least one collision after 5000 packets are sent!
- Thus, we will see the same IV again… and again…

A consequence of key stream reuse is that IV decryption dictionaries can be built

- Once a plaintext is known for an IV collision, the adversary can obtain the key stream for **that specific** IV!
- The adversary can gather the keystream for each IV collision he observes
  - As he does so, it becomes progressively easier to decrypt future messages (and he will get improved context information!)
  - The adversary can build a dictionary of (IV, keystream)
- This dictionary attack is effective regardless of keysize as it only depends on IV size!

WEP also failed to achieve its message authentication goals

- The checksum used by WEP is CRC-32, which is not a cryptographic checksum (MAC)
  - Purpose of checksum is to see if noise modified the message, not to prevent "malicious" and intelligent modifications
- Property of CRC: The checksum is a linear function of the message

$$c(x \oplus y) = c(x) \oplus c(y)$$

- This property allows one to make controlled modifications to a ciphertext without disrupting the checksum:
  - Suppose ciphertext C is:

$$C = RC4(v, K) \oplus \{M, c(M)\}$$

  - We can make a new ciphertext C' that corresponds to an M' of our choosing
  - Then we can spoof the source by: A→B: {v,C'}

WEP also failed to achieve its message authentication goals, pt. 2

- Our goal: Produce an M'=M+δ, and a corresponding checksum that will pass checksum test. (Hence, we will need to make a plaintext P'={M',c(M')} and a corresponding ciphertext C')
- Start by choosing our own δ value, and calculate checksum.
- Observe:

$$\begin{aligned} C' &= C \oplus \{\delta, c(\delta)\} \\ &= RC4(v,K) \oplus \{M, c(M)\} \oplus \{\delta, c(\delta)\} \\ &= RC4(v,K) \oplus \{M \oplus \delta, c(M) \oplus c(\delta)\} \\ &= RC4(v,K) \oplus \{M', c(M \oplus \delta)\} \\ &= RC4(v,K) \oplus \{M', c(M')\} \end{aligned}$$

- Thus, we have produced a new plaintext of our choosing and made a corresponding ciphertext C'
- Does not require knowledge of M, actually, we can choose δ to flip bits!

WEP allows for easy message injection, and does not provide any form of access control

- Property: The WEP checksum is an unkeyed function of the message.
- If attacker can obtain an entire plaintext corresponding to a frame, he will then be able to inject arbitrary traffic into the network (for same IV):

1. Get RC4(v,K)
2. For any message M' form $C' = RC4(v, K) \oplus \{M', c(M')\}$

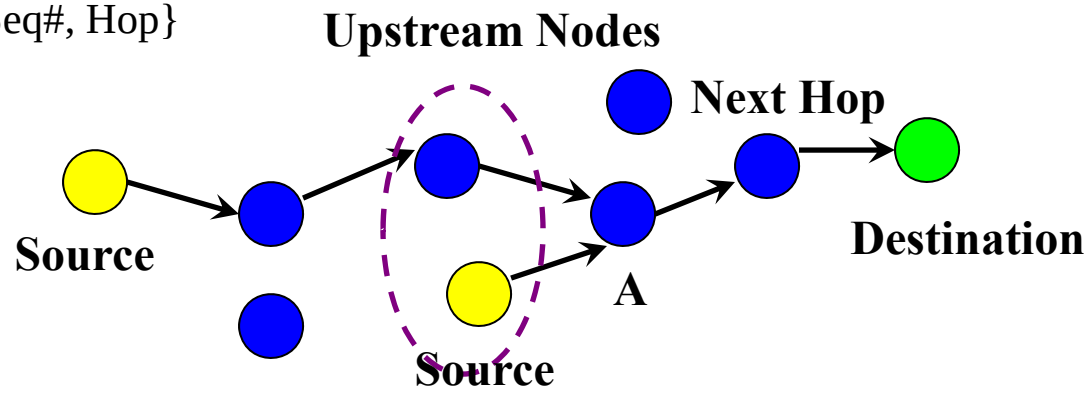
- Why did this work? c(M) only depended on M and not on any key!!!
- (Note: An adversary can easily masquerade as an AP since there are no mechanisms to prevent IV reuse at the AP-level!)

There are numerous other security flaws in 802.11, but some issues addressed in WPA2

- Another example is the Evil Twin AP:
  - An adversary installs a rogue AP near a regular AP.
  - This rogue AP may use the same SSID as the regular AP. It may or may not use the same MAC address as the regular AP. Rogue AP transmits with a stronger signal power.
    - *Clients automatically associate with rogue AP due to higher signal strength.*
    - *The rogue AP may drop all traffic from the clients that connect to it.*
  - Defenses:
    - *Perform network authentication. Requires the establishment of a known, shared key!!!*
    - *For open networks, always try both AP's and see which one provides service. Once a good AP is found, use signal strength as a consistency check.*
- Many issues can be addressed by employing WPA2
  - WPA2 addressed TKIP-legacy flaws in WPA
  - Includes pre-shared (home/office) key mode and EAP extensions for enterprise
- Major issue that remains is the protection of control frames (e.g. association frames)

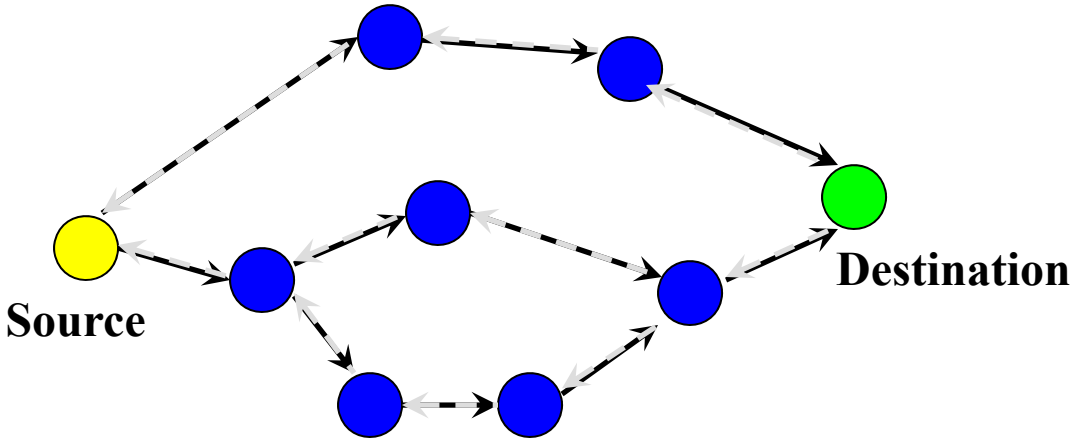
AODV is a popular routing scheme used in MANET and mesh networking

- Characteristics
  - Provides unicast, broadcast functionalities
  - Reactive: Initiate route discovery only on demand
  - Two dimensional routing metric: {Seq#, Hop}
- Routing Table
  - Destination Address, Seq Number
  - Next Hop, Hop Count
  - Lifetime
  - Upstream Nodes



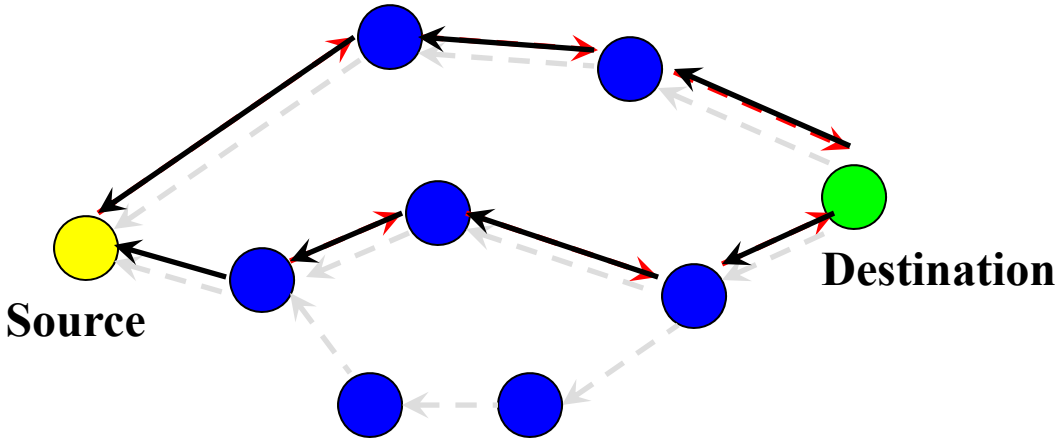
- Sequence Number
  - Created by each node to be included along with routing messages
  - Prevents routing loop, larger sequence number means a fresher route
  - A node increments its own sequence number when it
    - *Originates a route discovery*
    - *Originates a route reply in response to a route discovery request*
  - A node increments other node's sequence number by one only:
    - *In response to a lost or expired link to the next hop towards that destination*
- Hop Count
  - Smaller hop count means better route

The first stage in AODV route establishment is Route Discovery via RREQ to the destination



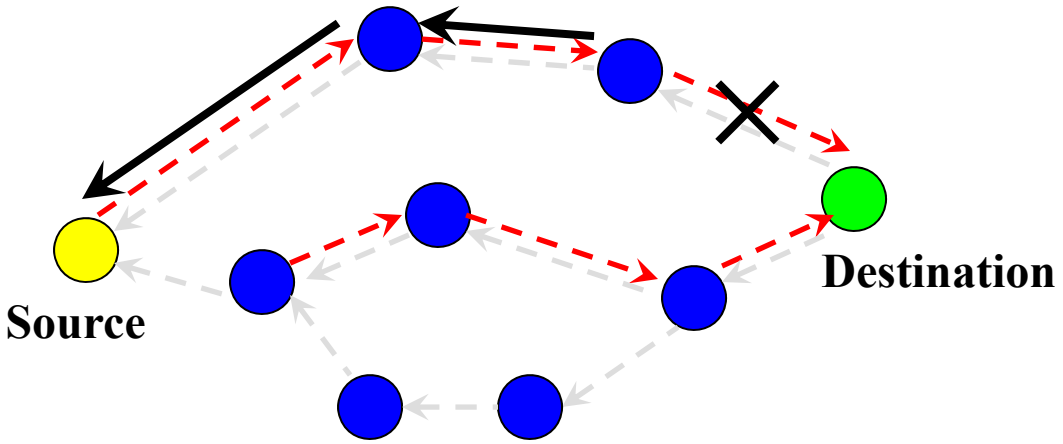
- Reverse Path Setup
  - Source broadcasts Route Request (RREQ)
  - RREQ: {S, D, ID, SrcNum, DstNum, Hop}
  - Nodes keep track of {S, ID}, discard redundant RREQs
  - Intermediate nodes update their routing entries toward Source
  - Intermediate nodes update DstNum if they have a higher copy
  - A node can reply to a RREQ if
    - *It is the destination*
    - *Has a fresh enough route to the destination*
  - Otherwise, increase the hop count and forward the RREQ

The second stage in AODV route establishment is Route Establishment via RREP to the source



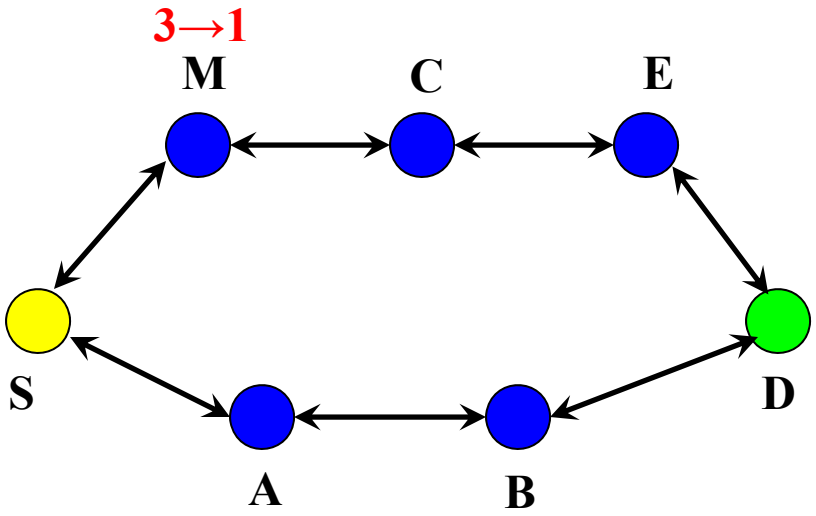
- Forward Path Setup
  - Destination or intermediate nodes with “fresh enough” route to Destination unicasts Route Reply (RREP) back to Source
  - RREP: {S, D, DstNum, Hop, Lifetime}
  - Intermediate nodes update the routing entry toward destination if it is a better or newer route, use reverse path to forward RREP
  - Intermediate nodes increase the hop count and forward the RREP

AODV route maintenance involves returning RERR messages to the source when a link breaks



- Route Maintenance
  - Node multicasts Route Error (RERR) to upstream nodes when
    - *It detects a link break for the next hop while transmitting data*
    - *It gets a data packet destined to a node for which it does not have an active route and is not repairing*
    - *It receives a RERR from a neighbor for one or more active routes.*
  - RERR: {D–List, DstNum–List}

AODV is susceptible to a variety of attacks against its basic maintenance functions



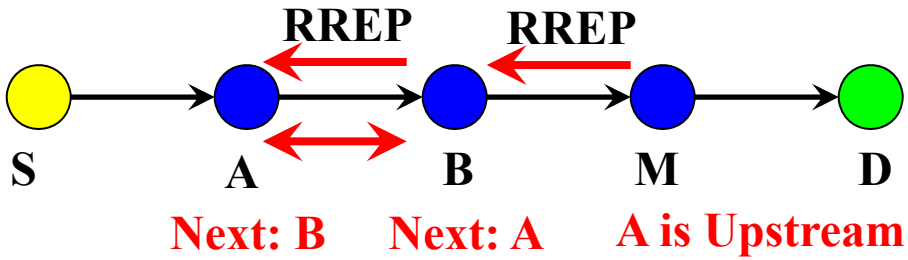
- Attacks on AODV
  - Forge RREQs/RREPs/RERRs on behalf of other nodes
  - Reduce the hop count in RREQs/RREPs
  - Increase the originator sequence number in RREQs
  - Increase the destination sequence number in RREPs
  - Selectively forward/reply RREQs, RREPs, and RERRs
  - Wormhole Attacks
- Securing AODV needs Authentication
  - End to end Authentication is not enough (how can intermediate nodes verify?)

Secure AODV (SAODV) uses public key cryptography to authenticate RREQ/RREP/RERR

- Zapata et. al. proposed SAODV in 2001
- Characteristics
  - Authenticate RREQs/RREPs/RERRs
  - Based on public key cryptography
- Assumption
  - Each node has a pair of public/private keys
- Security Extension
  - Intermediate nodes will not replace the destination sequence number in RREQ even if it has a newer copy
  - Single Signature Extension
  - Double Signature Extension
  - Hop count is protected through the use of a one-way hash function
- Details
  - Signature and hash functions are added to routing messages
  - Intermediate nodes verify signature and hashes before forwarding
  - Apply hash function to get next hop value

SAODV is nonetheless susceptible to a variety of simple attacks

- Problems remain in SAODV
  - Same hop count fraud
    - *Malicious nodes can still choose NOT to increase hop count*
  - Signature DoS
    - *Malicious nodes can flood a large amount of bogus traffic*
  - Formation of routing loops
    - *Intermediate nodes do not authenticate previous hop*

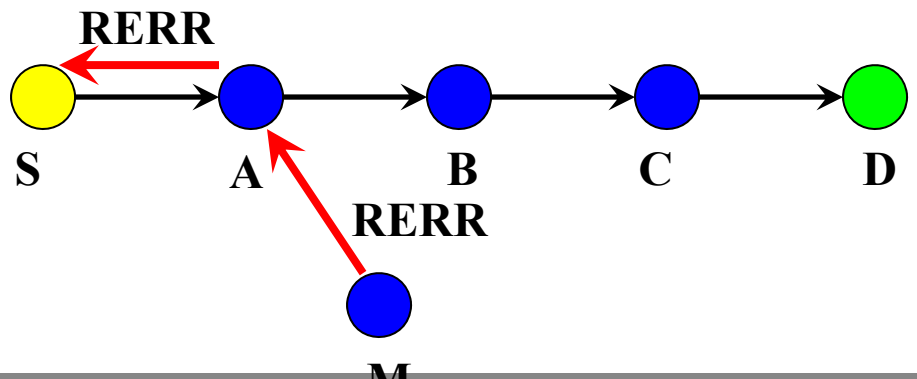


Authenticated Routing for Ad hoc Networks (ARAN) also authenticates RREQ/RREP/RERR

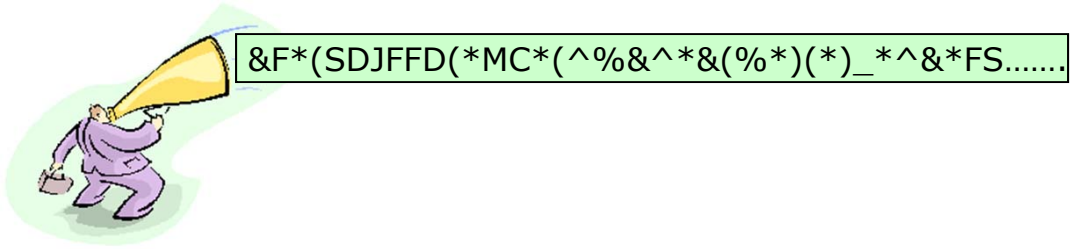
- Sanzgiri proposed ARAN in 2002
- Characteristics
  - Authenticate RREQs/RREPs/RERRs
  - Based on public key cryptography
- Assumption
  - Each node has a pair of public/private keys
- Security Extension
  - Use nonce plus timestamp to provide the freshness of a route
  - Nonce and timestamp stay unchanged during propagation
  - Does not use hop count to select optimal path, use first received RREQ
  - RERR is forwarded along the way without modification
- Details
  - Signature and certificate are added to routing messages
  - First hop nodes verify the signature and add their own signature and certificate to routing message
  - Other nodes verify the signatures and replace the previous hop signature and certificate with its own

ARAN is also susceptible to simple attacks targeted at exploiting routing functionality

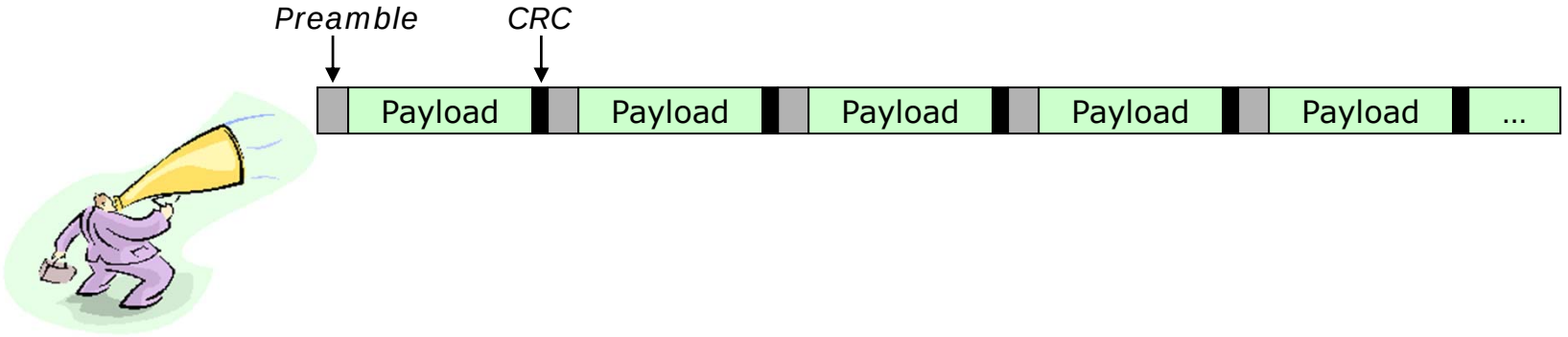
- Problems
  - Same hop count fraud
    - *Malicious nodes can still choose NOT to replace the signature and certificate of previous hop*
  - Signature DoS
    - *Malicious nodes can flood a large amount of bogus traffic*
  - Error spoofing attack
    - *Only authenticate the original source of the RERR*



Wireless networks are susceptible to interference attacks that target layer 1 and 2

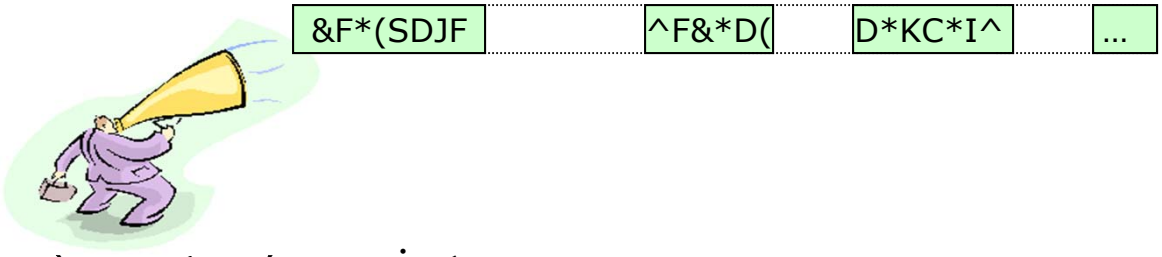


- Constant jammer:
  - Continuously emits a radio signal

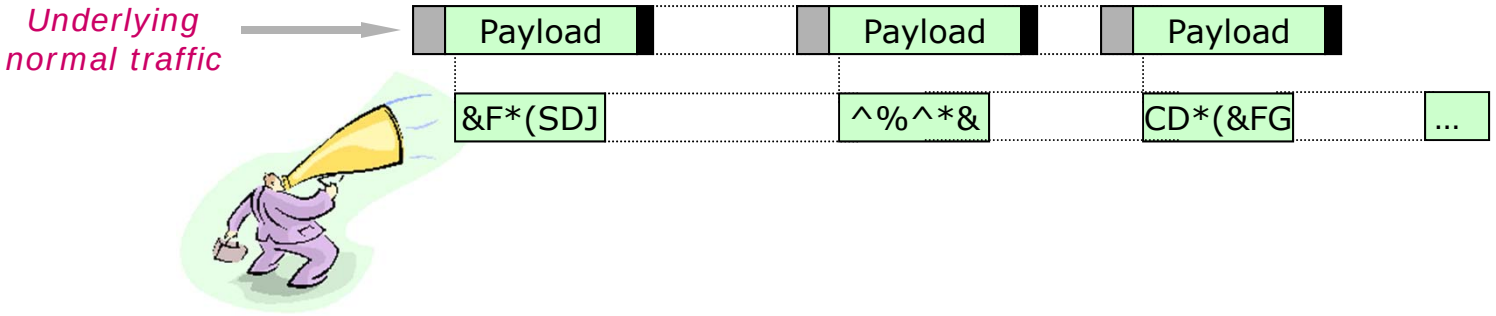


- Deceptive jammer:
  - Constantly injects regular packets to the channel without any gap between consecutive packet transmissions
  - A normal communicator will be deceived into the receive state

A variety of L2 jammer models have been proposed, including a powerful reactive jammer



- Random jammer:
  - Alternates between sleeping and jamming
    - *Sleeping period: turn off the radio*
    - *Jamming period: either a constant jammer or deceptive jammer*



- Reactive jammer:
  - Stays quiet when the channel is idle, starts transmitting a radio signal as soon as it senses activity on the channel.
  - Targets the reception of a message

Security People have Problems
and
How YOU can Fix Them…

Caveat Cryptor: Designer Beware!

- The lesson learned from these stories:
  - The adversary can be very powerful and clever!
  - Engineers make cruddy security analysts
  - There is a body of knowledge in the crypto community that never makes it to the engineering world
- We must assume that the adversary has complete control over the network…
  - Be paranoid! Alice should not blindly trust what she is getting from "Bob"! And vice-versa!
  - If we can build a system that we trust in this **Seriously Caustic** environment, then we can trust it in more general (day-to-day) computing scenarios

History has shown that security is not an easy task, and we must stand on the giants before us

- Building secure systems and protocols is not easy.
- In general, its not an easy matter to prove that some protocol is indefinitely secure.
  - Denning-Sacco protocol took 12 years for a protocol failure to be exposed
  - Needham-Schroeder survived for 17 years before a man-in-the-middle attack was found
- Attacks of today must always be considered when building systems
  - Attacks of tomorrow aren't known yet…
  - That's the challenge!
- What can we do?
  - Formal verification logics?
  - Basic design guidelines?
  - Teach people to be attackers and defenders?
  - Read Schneier and Ferguson?

We need to revisit Needham's security design guidelines

- Needham has given several guidelines for building secure systems

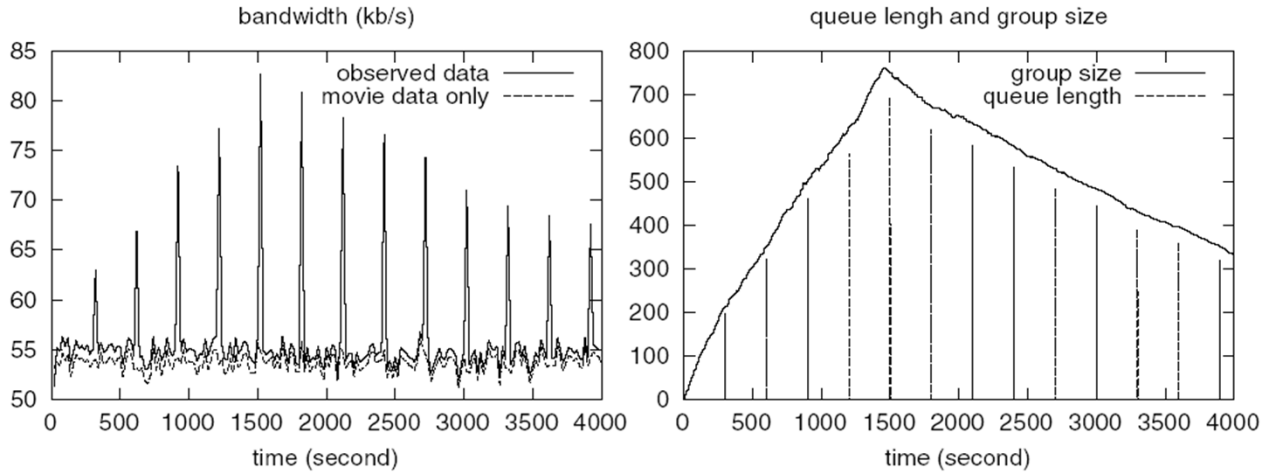
1. Be clear of security goals and assumptions
2. When using encryption, know why you are using it (secrecy? Authenticity? Binding? PRNG?) . Encryption is not security!
3. Be careful about temporal associations
4. Don't assume the identity of a participant can be excluded from a message. Generally, it should be explicitly included in a message!
5. Have redundancy in your message!
6. Know the properties and weaknesses of the cryptographic protocols you are using.
7. Signatures do not imply that the signer knows what the message is that he is signing!
8. Don't trust others to keep their secrets secret!
9. When responding to queries, be careful about encrypting, decrypting, or signing. You might be used as an oracle by an adversary!
10. Decryption is not the same as digital signatures- they have different purposes!
11. Distinguish between different runs of the protocol!

There are many other considerations that somehow must be factored into problem!!!

- KISS (Keep It Simple Stupid) is often desirable from an engineering point of view…
  - Its generally **BAD** from a security point of view!
  - Removing some data fields because they seem like they can be inferred (and thus shorten the message) can result in severe protocol failures!
- However, we still have to consider some key performance issues as we may destroy the very system we aim to protect!
- Deny by default might be desirable, but impractical
- It is not all about crypto– there are many "wireless" problems that can be exploited

3GPP tried to secure MBMS but forgot to address basic network performance issues

- 3G Multimedia Broadcast/Multicast Service provides media to a group of users
  - Qualcomm S3-030040 3GPP proposal sought to "Make the Session Key change so frequently that the cost of attacking is more expensive than the cost of subscribing to the service"
- Bandwidth: network resources will be wasted on sending out SK_RAND.
  - *SK_RAND has to be appended to each package.*
  - *For higher level of security, SK_RAND has to be large.*
- BAK update problem: at the moment that a new BAK is used, every USIM will send out a BAK request to BMSC
  - *BAK implosion problem*
  - *High peak bandwidth*



We need to train wireless security experts that understand security fundamentals and wireless

- Many mistakes arise from lack of awareness of the strengths and weaknesses of the tools they use
  - Textbook crypto is generally weak
    - *Often a weak confidentiality model, in which the enemy is a* passive *eavesdropper, is used*
    - *We should consider an active adversary– they may modify a ciphertext or calculate a plaintext and send the result to a user to get an oracle service*
  - Adversary models are generally too simplistic
    - *Adversaries are often too localized*
    - *Adversaries are often too constrained in their resources (e.g. antennas)*
    - *Adversaries are often only outsiders... security people don't trust friends!*
- There are many tools that remain to be developed or to migrate into mainstream use
  - ID-crypto is just beginning to become popular
  - Trusted platform modules are generally relegated to the DRM community and need to be employed in a broader range of scenarios
  - Many tools (e.g. formal methods) may be helpful, but are at still at a young stage of development
- Let us look at some examples things we should teach…

Dolev-Yao represents the basic "omnipresent" adversary model in distributed systems

- For distributed systems and networks, we often should assume that there are adversaries
  - Everywhere in the network
  - Adversary may: eavesdrop, manipulate, inject, alter, duplicate, reroute, etc…
  - Adversary may control a large number of network nodes that are geographically separated
- Dolev-Yao Threat Model:
  - A very powerful adversarial model that is widely accepted as the standard by which cryptographic protocols should be evaluated
  - Eve, the adversary, can:
    - *Obtain <u>any</u> message passing through the network*
    - *Act as a legitimate user of the network (i.e. can initiate a conversation with any other user)*
    - *Can become the receiver to any sender*
    - *Can send messages to any entity by impersonating any other entity*

The Dolev-Yao model is not all-powerful, but assumes the existence of good crypto

- This seems very powerful, but not entirely so…
- Under Dolev-Yao:
  - Any message sent via the network is considered to have been sent by Eve
  - Thus, any message received "might" have been manipulated by Eve
  - Eve can control how things are sent
- What is not possible:
  - Eve cannot guess a random number which is chosen as part of a security protocol
  - Without knowledge of a key, Eve cannot figure out a plaintext from a ciphertext, nor can she create ciphertexts from a plaintext.
  - Eve can't solve the private-key pairing of a public key
  - Eve cannot control the "memory" of a computing device of a legitimate user (i.e. Eve can only play with the communication)

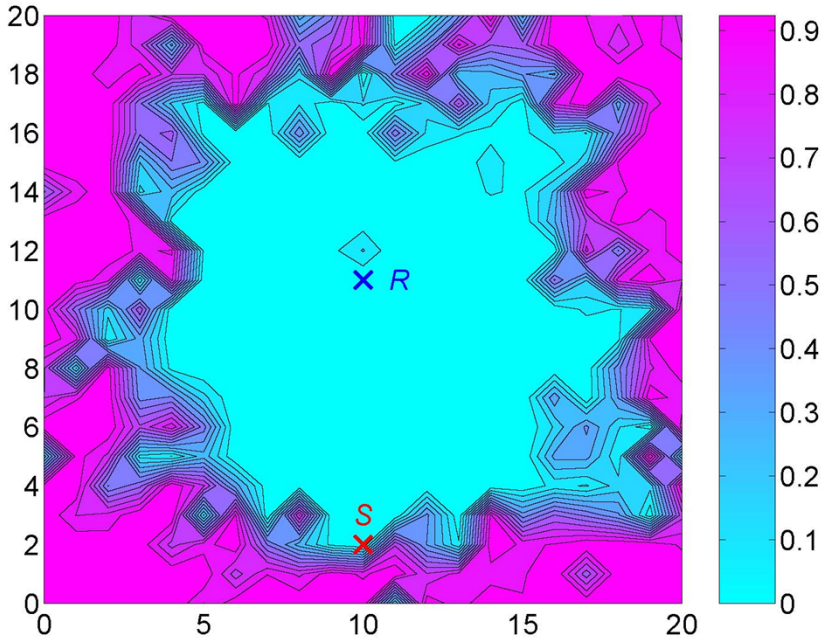
Non-Malleability is a cryptographic attack model that has caused problems in many systems

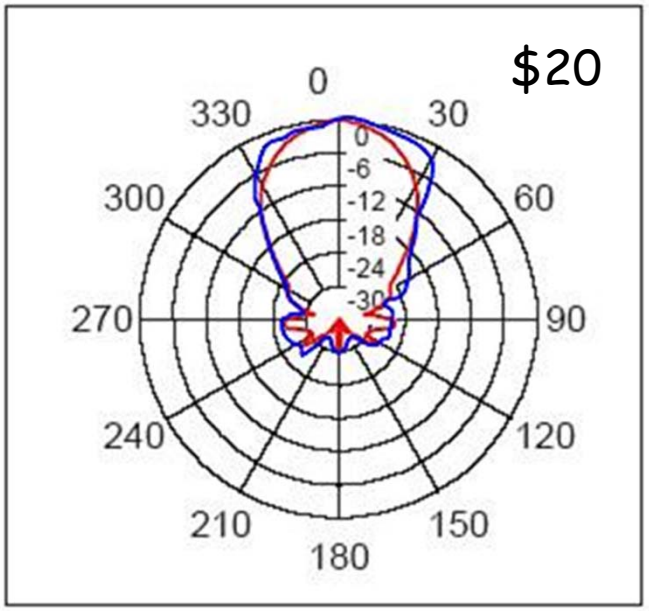
- Non-malleable cryptography: it should not be possible for Eve to modify a plaintext in a meaningfully controllable manner via modifying the ciphertext
- We have seen such a problem before:
  - One-time pads and Vernam ciphers: it is possible to modify select bits
  - We saw this type of weakness in WEP
- In a malleability attack, Eve's objective is, given a ciphertext C, not to learn something about the plaintext M, but instead to wreak havoc upon the eventual decoding
  - Eve needs to create a relationship C→ C' that results in a meaningful relationship M→ M'
- Problem: Most conventional cryptographic algorithms are the result of trapdoor functions
  - Partial information oracles exist for these public key schemes (e.g. the math that lets one learn parity can be the basis for conducting a malleability attack)
- Example: How to double a plaintext in RSA encryption
  - Take $C=M^e \bmod N$ and then produce $C' = C2^e \bmod N$

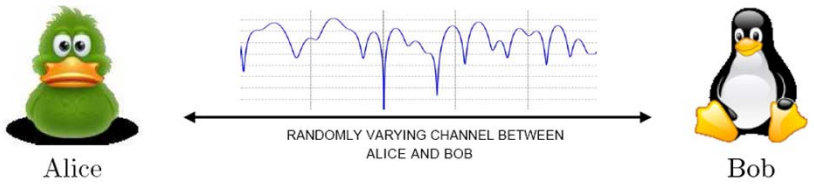
Byzantine threats involve weaknesses from the inside, which capture a key source of threats

- The Dolev-Yao model is the foundation of security analysis for active adversary scenarios, but does not capture everything that an adversary can do:
  - It does not involve entity compromise
- In situations involving many participants (e.g. distributed computing or peer-to-peer), it is natural to ask what can happen if a legitimate entity becomes compromised
- A Byzantine failure is one where a node/entity fails to operate properly, but continues to operate (as opposed to fail-stop failures)
- Example Byzantine Failures:
  - A node may lie about connectivity
  - Flood network with false traffic
  - Falsely describe opinions of another node (e.g. P2P)
  - Capture a strategic subset of devices and collude

On the wireless front, we need to be aware of radio irregularities, antennas and propagation



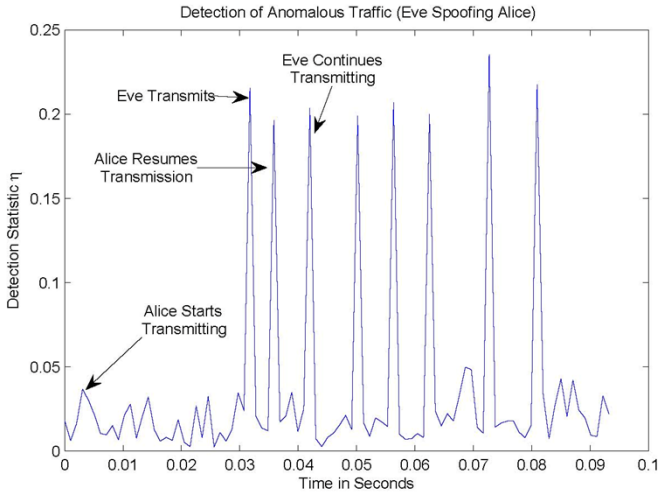




Alice

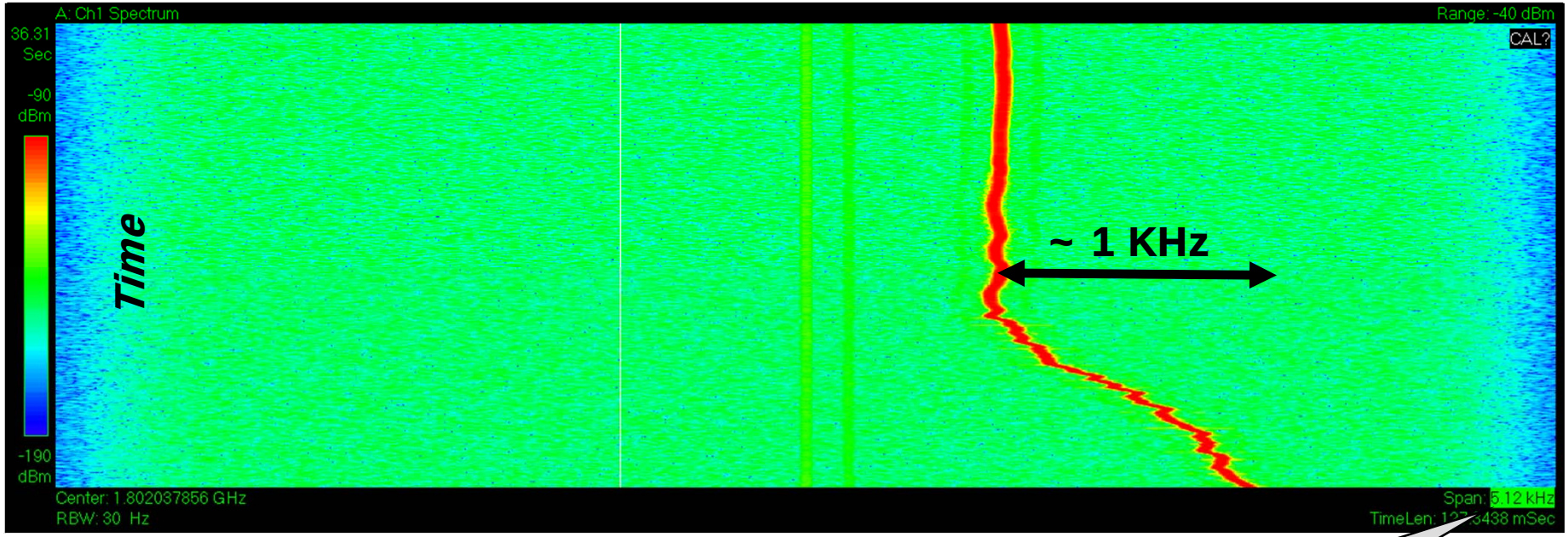
Bob

Eve



Wireless is not as easy as people think…

- LO Synchronization is tough
  - Cooperative protocols must face the fact that units are spatially diverse
  - Even so, synchronization at the senders does not mean synchronization at the receiver!
  - Then there is frequency drift… caused by the slightest of things…



The Sampler Platter: Research State-of-the-Art

Alice and Bob Get Physical: Insights into Physical Layer Security

Wade Trappe

Collaborators: Narayan Mandayam, Larry Greenstein, Roy Yates, InterDigital

Students:

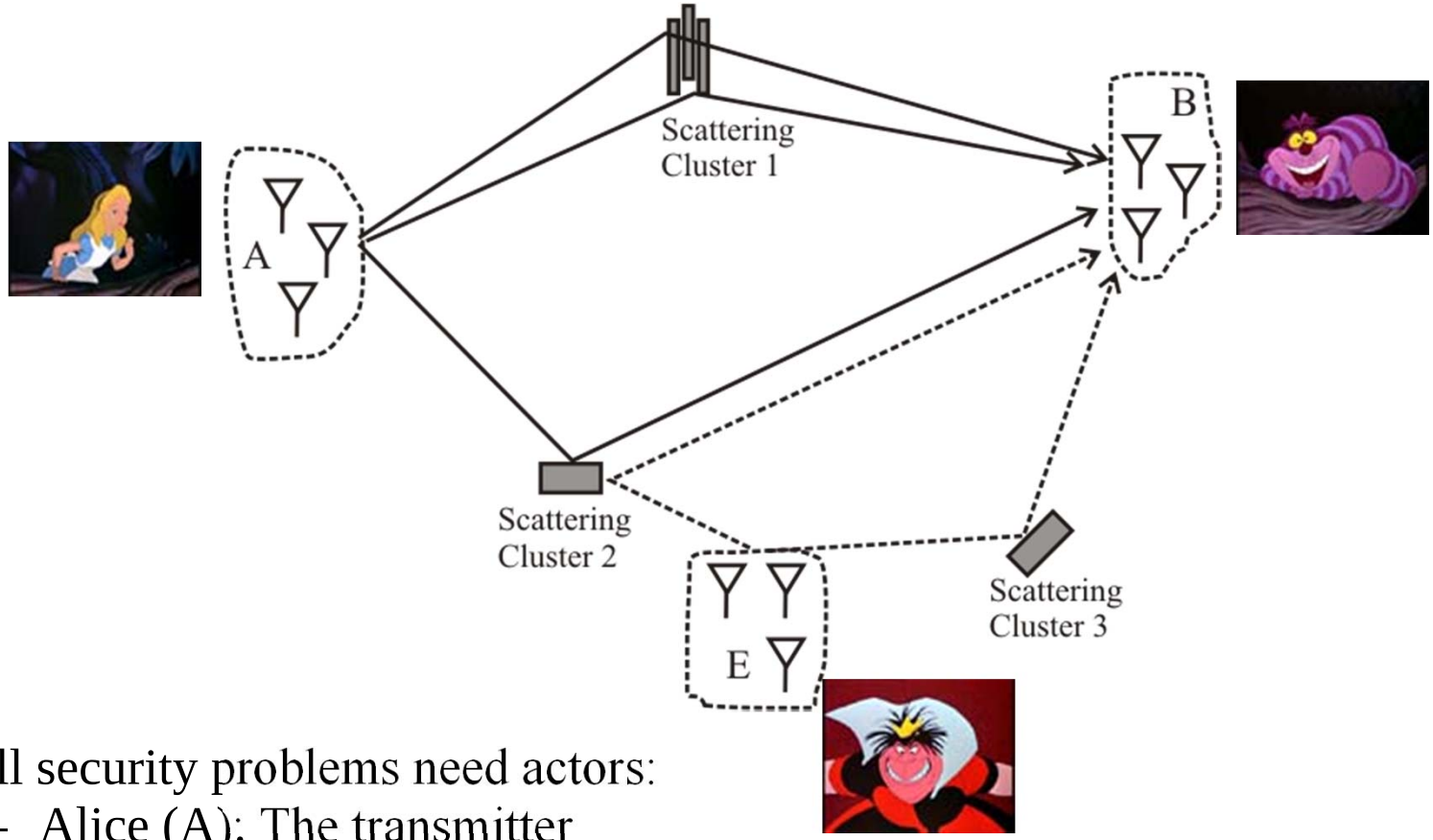
Liang Xiao
Zang Li
Suhas Mathur
Wenyuan Xu
Rob Miller
Ozge Aliye Kaya

RUTGERS Research supported by: NSF, DARPA, Army, and Wade's pizza-fund

A 10000 Ft. Overview of Security Via Lower Layer Enforcements (SEVILLE)

- Although conventional cryptographic and network security techniques are essential to securing wireless networks, they are not a complete solution
- We believe lower-layer information associated with the wireless channel can be used to enhance wireless security
  - The typical wireless multipath transmit-receive channel is *frequency-selective* (or in the time domain, *dispersive*) in a way that is *location-specific* with rapid *decorrelation* properties
  - The channel response between a transmitter and a receiver can be a unique, shared, non-predictable source of *secret* information
- This secret information is a "fingerprint in the ether" we propose to use to develop cross-layer *Authentication Services* and *Confidentiality Services*
- We are encouraged by two notable parallel paradigm shifts in wireless systems:
  - (1) code division multiple access (CDMA) systems, where the use of Rake processing transforms multipath into a diversity-enhancing benefit
  - (2) multiple-input multiple-output (MIMO) antenna techniques, which transform scatter-induced Rayleigh fading into a capacity-enhancing benefit

Alice, Bob and Eve get Physical !!!



- All security problems need actors:
  - Alice (A): The transmitter
  - Bob (B): The receiver
  - Eve (E): The evil adversary
- Their roles depend on the type of security objective we have

PHY-101

- RF Signals transmitted from Alice to Bob are affected by a variety of different factors: attenuation, large-scale and small-scale fading
- Fading arises as a signal's multipaths constructively & destructively combine at the receiver
- System Model: For input u(t), the received signal is

$$r(t) = \int_{-\infty}^{\infty} h(t,\tau)u(t-\tau)d\tau$$

- Under the wide-sense stationary uncorrelated scatter (WSSUS) model, the channel response becomes a tapped-delay line:

$$h(t,\tau) = \sum_{i=1}^{N} h_i(t)\delta(t-\tau_i)$$

- Under Rayleigh Fading assumptions $h_i(t)$ are zero-mean complex Gaussian

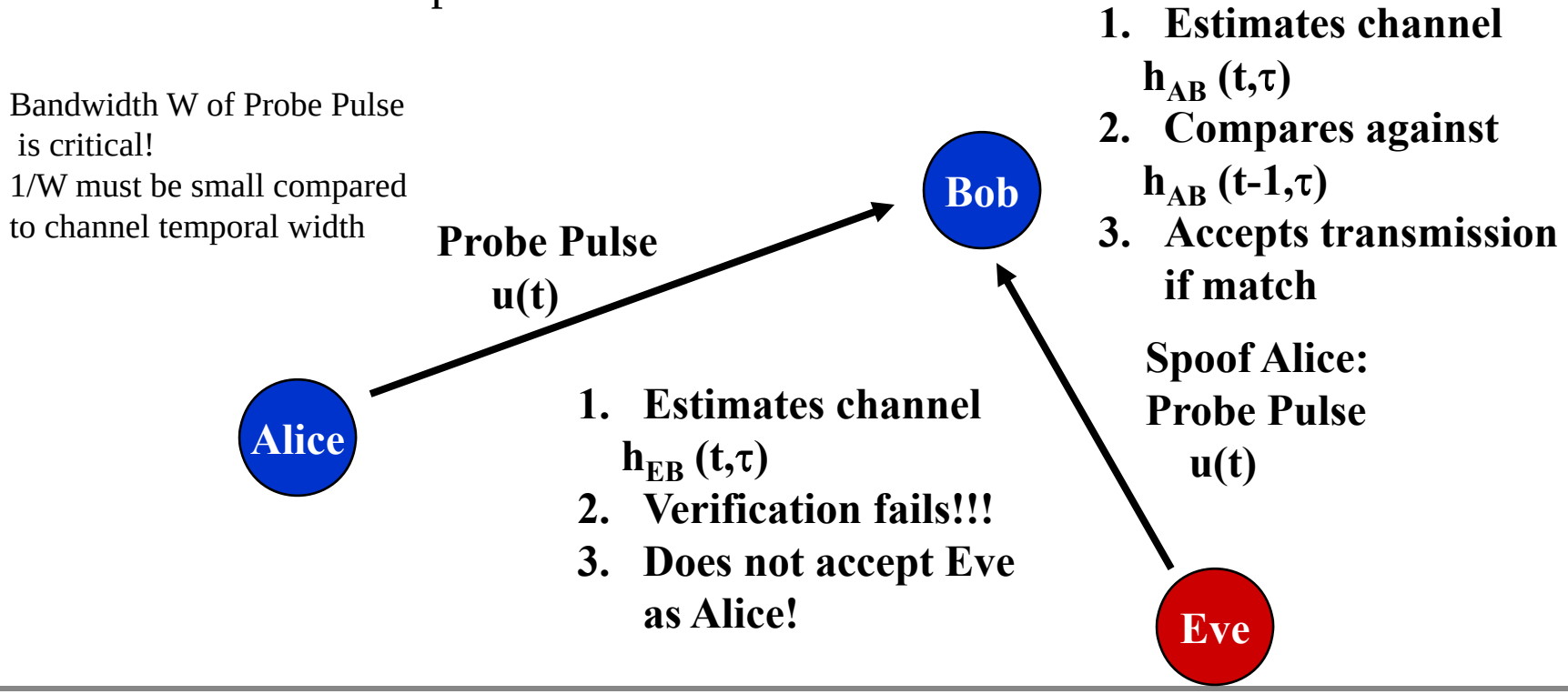
- The channel response is itself time-varying and stochastic
  - There is temporal, spectral and spatial variability of the channel response
  - *Coherence Time:* Difference in time needed for fading correlation to drop below a threshold
  - *Coherence Bandwidth:* Separation in frequency needed for fading correlation to drop below a threshold
- Additionally, we may examine the instantaneous fading correlation between locations
- Jakes showed under uniform scattering that the fading correlation (amplitude correlation in received signal) drops off rapidly over a distance of half a wavelength

$$C(d) \approx J_0^2(2\pi d/\lambda)$$

- Separate by a wavelength and independence is a reasonable assumption (under Rayleigh WSSUS)

Authentication: A Cartoon Version

- Authentication in the PHY-sense is about verifying a transmission came from a particular transmitter– useful for spoofing detection!!!
- Wireless devices can authenticate themselves based upon
  - Ability to produce an appropriate received signal/channel estimate at the recipient
  - Location information can be extracted to authenticate a transmitter relative to its previous location

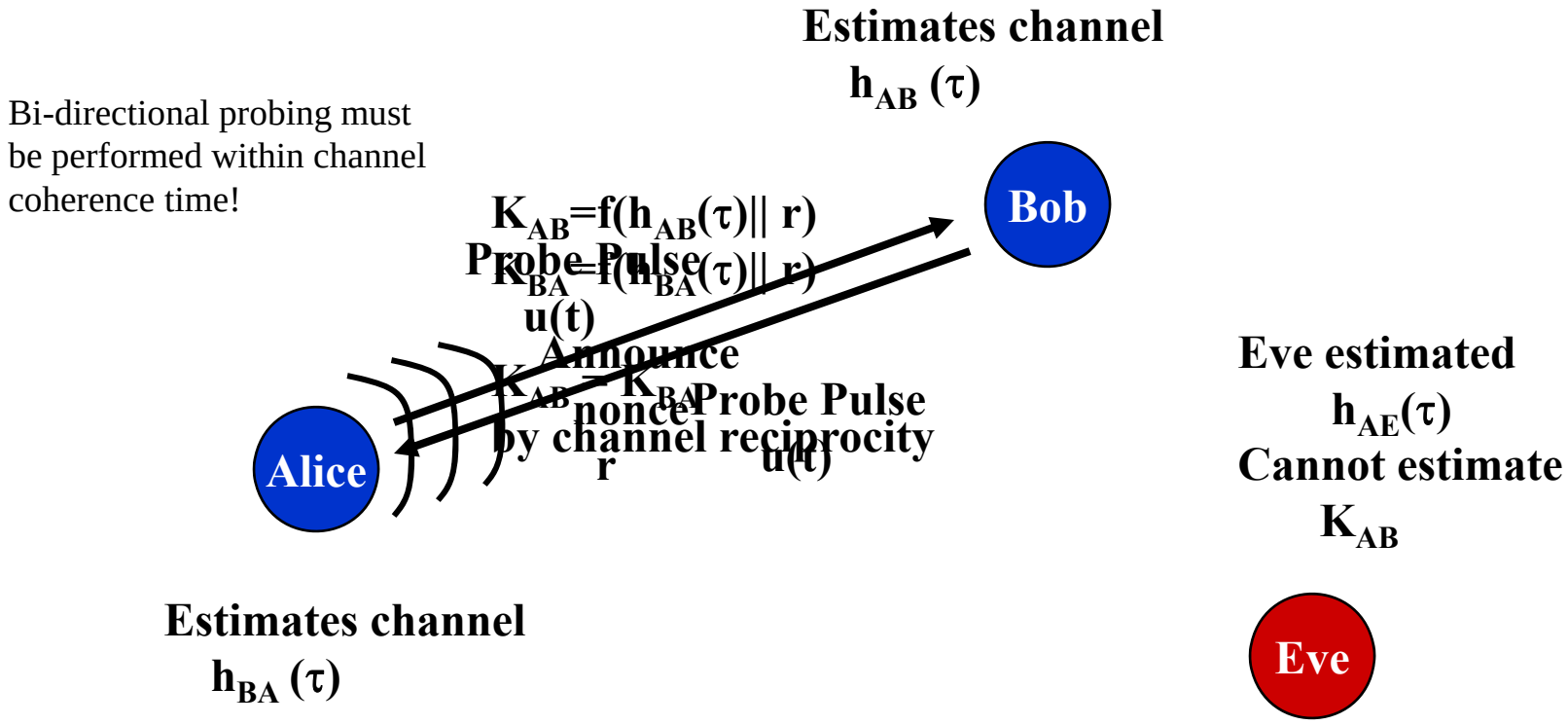


Confidentiality: Different Means to an End

- We also would like to use the PHY-Layer to support confidential communications
  - For higher-rate secret communications, we suggest that the PHY-layer be used to form higher-layer cryptographic keys
- There are two types of PHY-Layer Confidentiality Services:
  - *Extraction*: Use the channel estimate itself to form key bits
  - *Dissemination*: Use channel variations to opportunistically, and secretly convey communications/key bits…
- Note: There is a distinction between secret communication and LPI/LPD communications!

Extraction: A Cartoon Version

- The uniqueness and non-predictability of the channel can be used to establish a shared secret key for encryption services

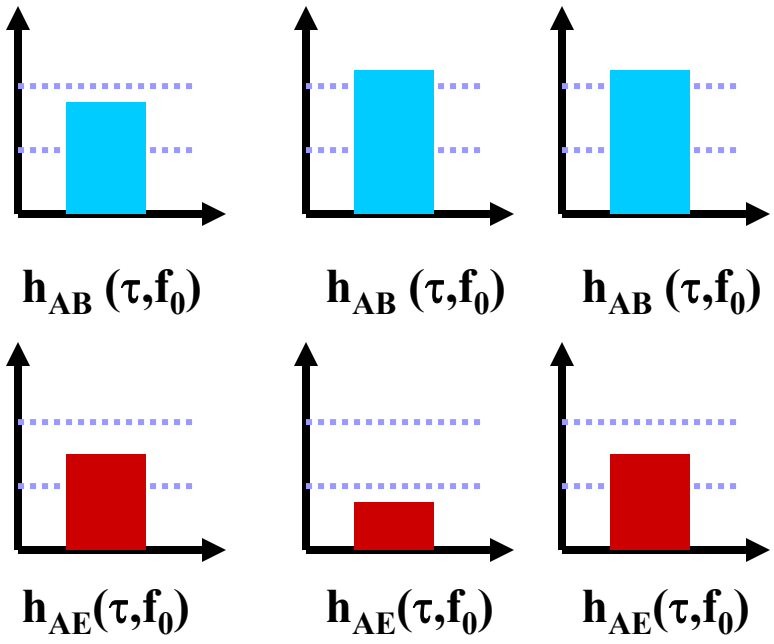


- Practical issues arise: quantization of channel estimates, channel reciprocity, temporal coherence, fast channel estimation.

Dissemination: A Cartoon Version

- Idea: When Alice→Bob channel is good, and Alice→Eve channel is bad… transmit!!!

Assume everyone's channel conditions are known by Alice



Alice

Don't Transmit!

Gain Difference Large Enough… Transmit!!!

Gain Difference Not Large Enough… Don't Transmit!!!

Bob

Eve

- Question: Why would Alice know Eve's channel?

Physical Layer Authentication

PHY-Authentication Via Significance Testing

- Sample frequency response at M frequencies
- Two complex frequency response vectors

$$\underline{\hat{H}}_{AB} = [\hat{H}_{AB}(0, f_1), \hat{H}_{AB}(0, f_2), ..., \hat{H}_{AB}(0, f_M)]^T$$

$$\underline{\hat{H}}_t = [\hat{H}_?(t, f_1), \hat{H}_?(t, f_2), ..., \hat{H}_?(t, f_M)]^T$$

- Simple Hypothesis:

$$\mathcal{H}_0: \ \underline{H}_t = \underline{H}_{AB}$$

$$\mathcal{H}_1: \ \underline{H}_t \neq \underline{H}_{AB}$$

- Test Statistic: $$Z = \min_{\theta} \frac{1}{\sigma^2} \| \underline{\hat{H}}_A - \underline{\hat{H}}_t e^{j\theta} \|^2$$
  - Phase measurement error due to changes of receiver local oscillator
- Channel measurement assumed to be noisy

Hypothesis Analysis

- Null Hypothesis $\boldsymbol{H_0}$

iid $N(0,1)$

$$Z = \frac{1}{\sigma^2}\left(\sum_{m=1}^{M} n_{1,m}^2 + \sum_{n=1}^{M} n_{2,n}^2\right) \sim \chi_{2M}^2$$

- Alternative Hypothesis $\boldsymbol{H_1}$

$$Z = \frac{1}{\sigma^2}\left(\sum_{m=1}^{M} (\Delta h_{1,m} + n_{1,m})^2 + \sum_{n=1}^{M} (\Delta h_{2,n} + n_{2,n})^2\right) \sim \chi_{2M,\mu}^2$$

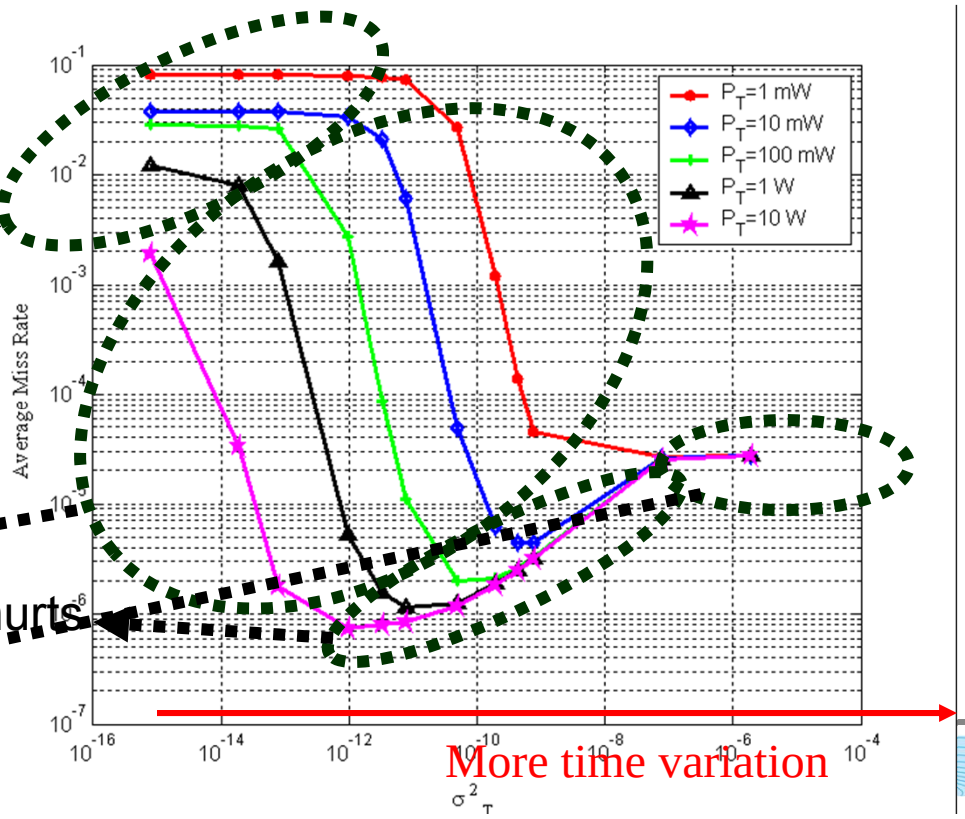
Real & Imaginary part of

$$\hat{H}_{AB,m} - \hat{H}_{t,m} e^{jArg\left(\sum_{m=1}^{M} \hat{H}_{AB,m}^{*} \hat{H}_{t,m}\right)}$$

$$\mu = \frac{1}{\sigma^2} \left\| \underline{\hat{H}}_{AB} - \underline{\hat{H}}_{t} e^{jArg\left(\sum_{m=1}^{M} \hat{H}_{AB,m}^{*} \hat{H}_{t,m}\right)} \right\|^2$$

Example Results: Time-Variant Channel

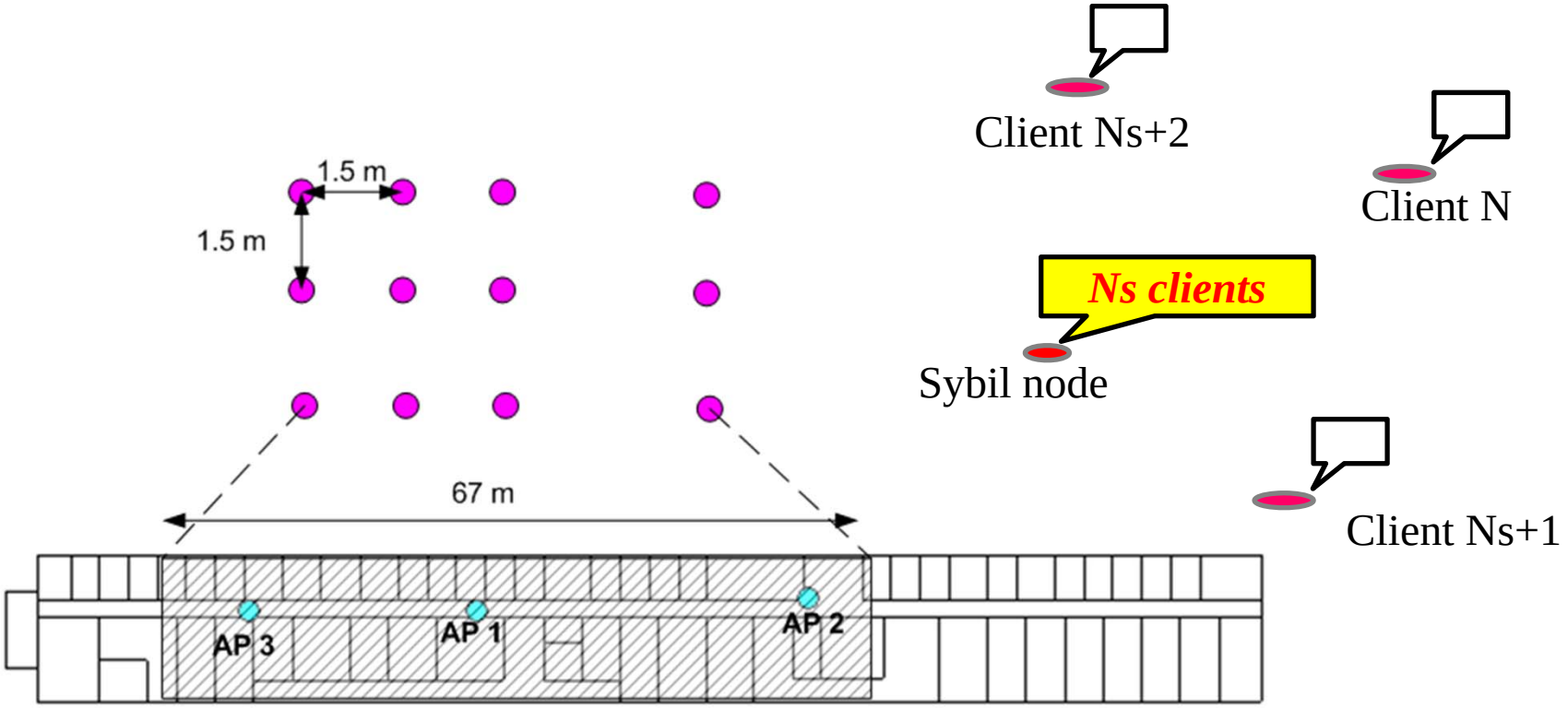
- Channel response $H_{AB}(t, f) = \bar{H}_{AB}(f) + \varepsilon_{AB}(t, f)$
  - Tap-delay model for the inverse Fourier transform of $\varepsilon_{AB}(t, f)$
  - Single-sided exponential model as power delay profile
  - AR-1 Model for the time correlation
- W=10 MHz, M=10



Time variation is negligible
Time variation helps
Time variation is so big that it hurts
Thermal noise is negligible

Sybil Detection

- The Sybil Attack: A node claims multiple identities
- The channel response serves as a fingerprint to detect multiple claimed identities
- Clever Adversary: Adapt power across subcarriers for each identity (but don't change "shape" or else decoding failure!)
- Issues to consider: System bandwidth, Number of APs and their synchronization



L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe, "Channel-based detection of Sybil attacks in wireless networks," *IEEE Transactions on Information Forensics & Security*

Channel-based Sybil Detection

- Channel sample obtained by the *n*-th client:

$$\underline{\hat{H}}_n = \underline{H}_n a_n e^{j\phi_n} + N_n, 1 \le n \le N$$

- Pair-wise Sybil detection between the *m*-th and *n*-th clients
  - Constant power allocation by the Sybil node

$$L(m,n) = \| \underline{\hat{H}}_m - e^{jArg(\underline{\hat{H}}_m \underline{\hat{H}}_n^H)} \underline{\hat{H}}_n \|^2 / \sigma^2$$

  - Adaptive power allocation by the Sybil node

$$L(m,n) = 2\| \underline{\hat{H}}_m - w\underline{\hat{H}}_n \|^2 / (1+|w|^2)\sigma^2 \ , \quad w = \underline{\hat{H}}_m \underline{\hat{H}}_n^H / \| \underline{\hat{H}}_n \|^2$$

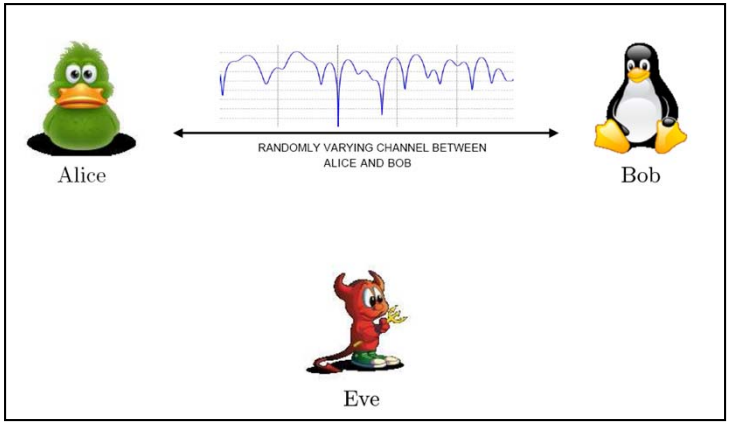
- Sybil detection with multiple clients:
  - Claim a Sybil client, if it has similar channel samples to at least one other client.
  - The decision function

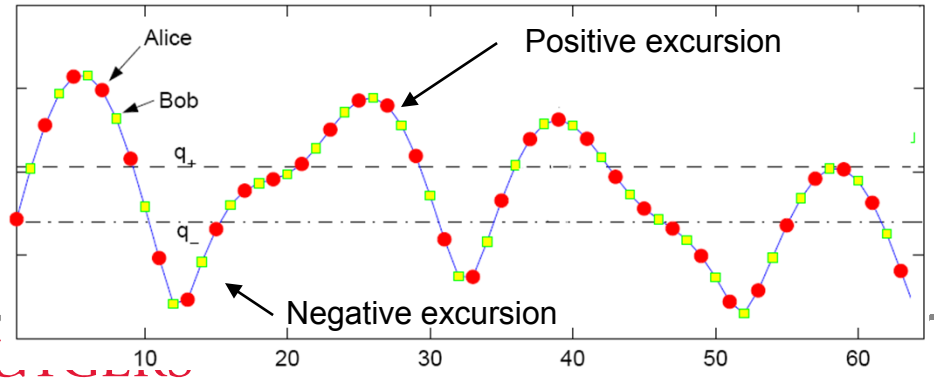
$$I(m) = \begin{cases} 1, \exists n \ne m, 1 \le n \le N, s.t.\ L(m,n) \le K \\ 0, o.w. \end{cases}$$

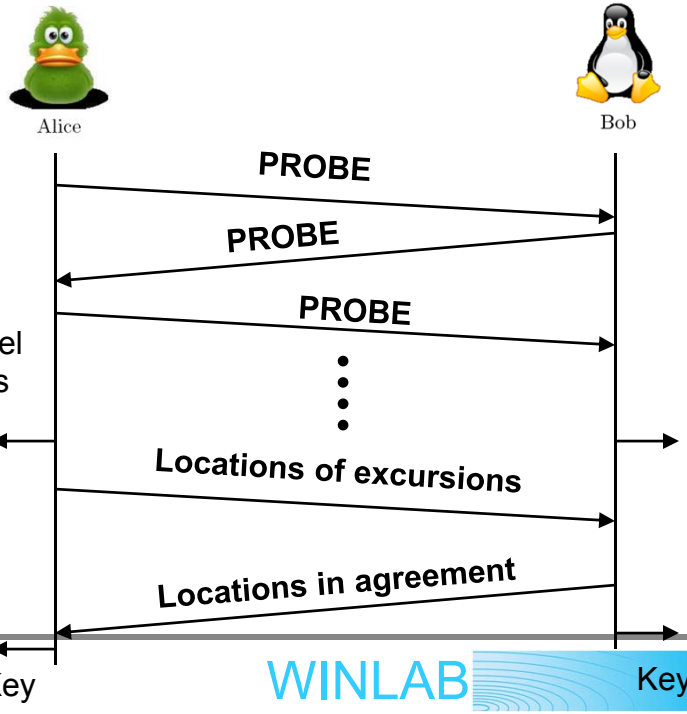
Confidentiality: Secrecy Extraction

Secret key extraction from a wireless channel

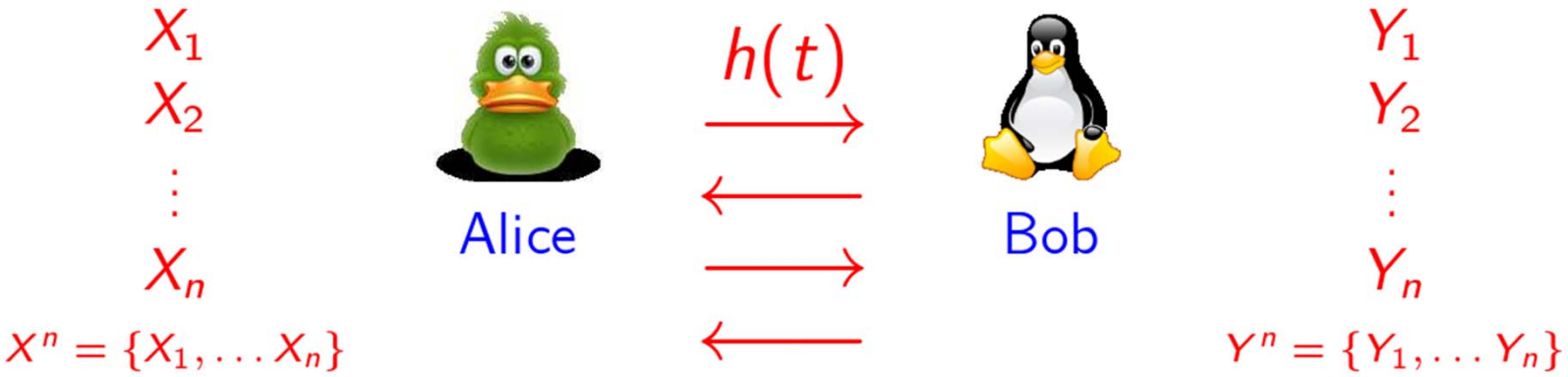
- Use channel reciprocity to build highly correlated data sets
  - Probe the channel in each direction
  - Estimate channel using recd. probe
- Eve receives only uncorrelated information as she is more than $\lambda/2$ away
- Level crossings are used to generate bits
- Alice and Bob must exchange msgs over public channel to create identical bits
- What if channel is not already authenticated?
  - Requires additional sophistry to prevent man-in-the-middle attack.
  - It is possible using the correlated data collected from received probes.

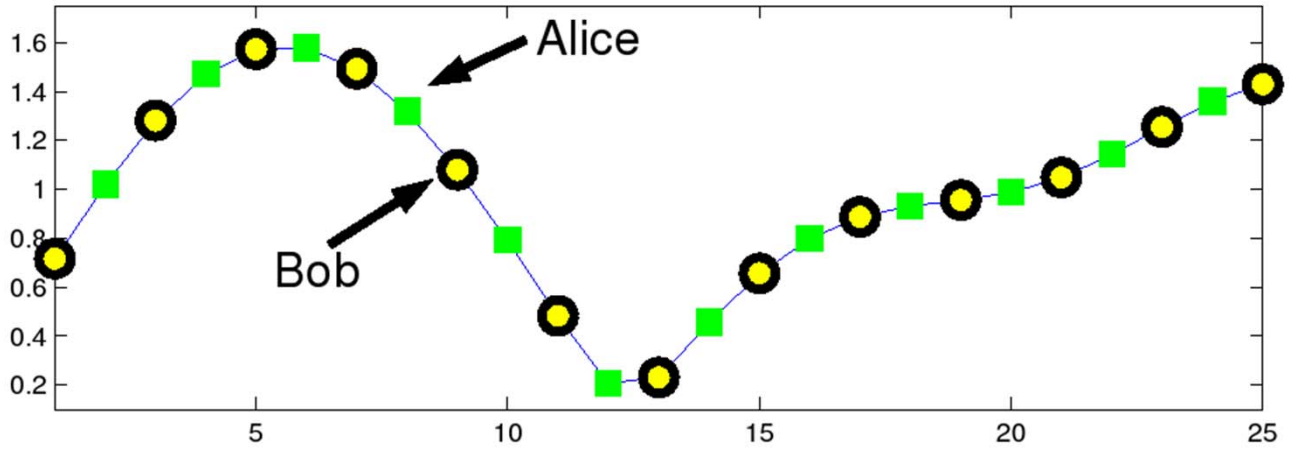




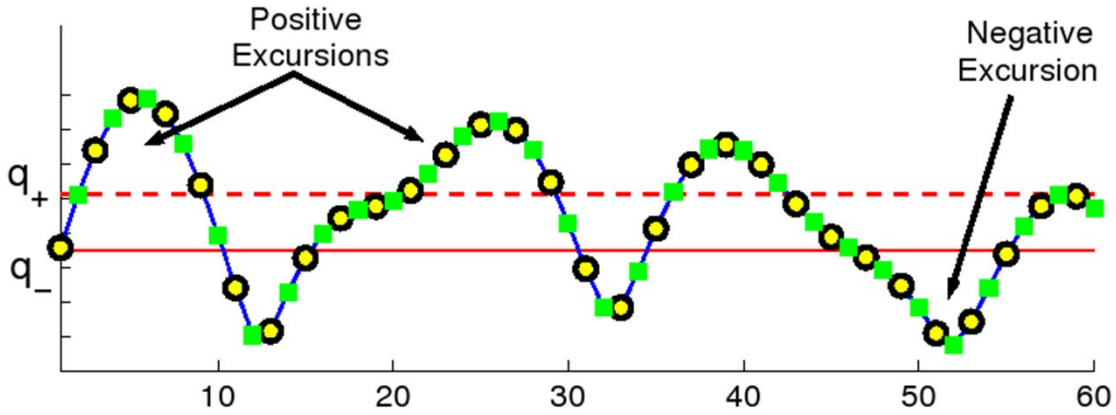


Radio Telepathy: Start





Radio Telepathy: The Protocol



Find locations of excursions in X^n of size $\geq m$. e.g. $\{6, 27, 42, 52 \ldots\}$
Send a random subset to Bob
$L = \{6, 42, 52 \ldots\}$

L →

Find the set of indices $\tilde{L} \subseteq L$ where Y^n has excursions of size $\geq m - 1$

If $\frac{|\tilde{L}|}{|L|} < \frac{1}{2} + \epsilon$ for some $0 < \epsilon < \frac{1}{2}$, declare message not sent by Alice

Compute $Q(Y^n(\tilde{L}))$.
The first N_{au} bits = auth-key K_{au}.
Remaining $N - N_{au}$ bits = secret key.
Send $\left\{\tilde{L}, MAC\left(K_{au}, \tilde{L}\right)\right\}$ to Alice.

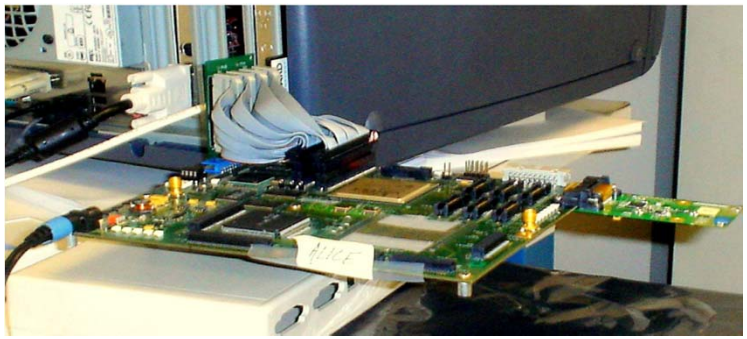
← $\tilde{L}, mac$

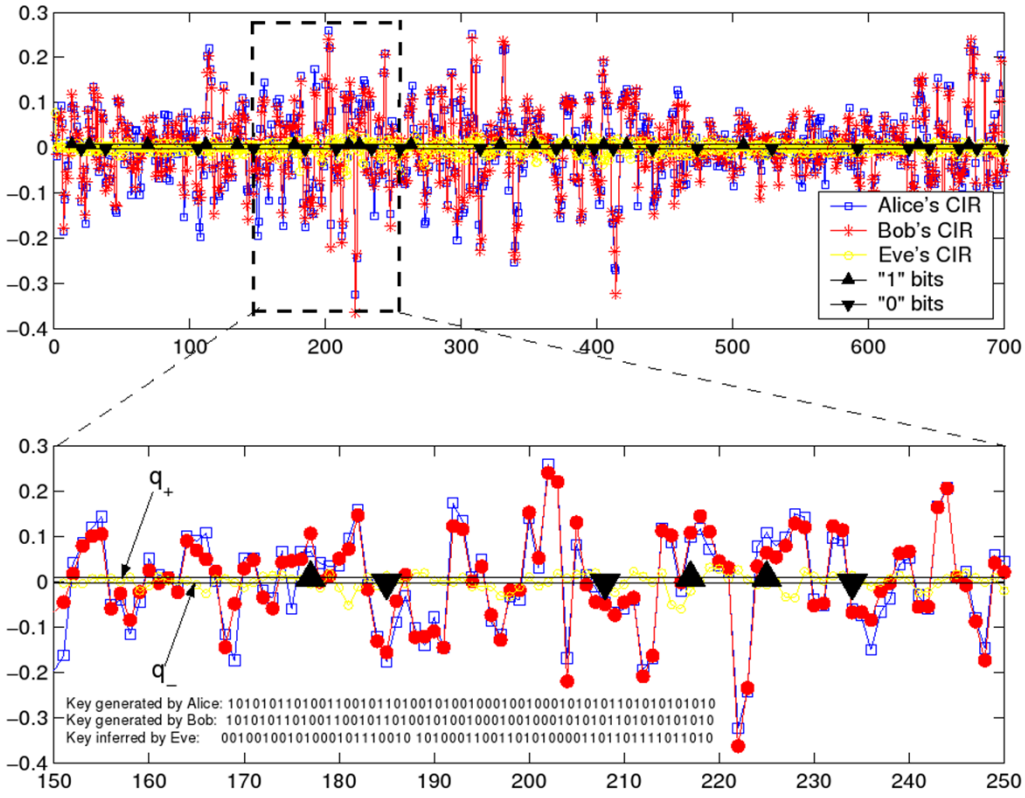
Compute $Q(X^n(\tilde{L}))$. The first N_{au} bits = auth-key K_{au} to verify the MAC.

Attacks on Radio Telepathy

- The integrity of $\tilde{L}$ is protected by the MAC
  - Eve doesn't have K_{au}, which is required to compute the MAC
  - Alice can compute K_{au} using X_n and $\tilde{L}$
- Modification of L will either
  - Reduce rate, by at most (1/2 + ε) of the expected value
  - Reveal Eve's presence by causing $\tilde{L} \not\subset L$
- Eve can insert her own PROBES
  - Defend using PHY-authentication
  - Apply one-way hash chain methods
- Man-in-the-Middle
  - Can't be protected against without mutual authentication!
  - Don't blame us… same problem exists for Diffie-Hellman!

Radio Telepathy Prototyping

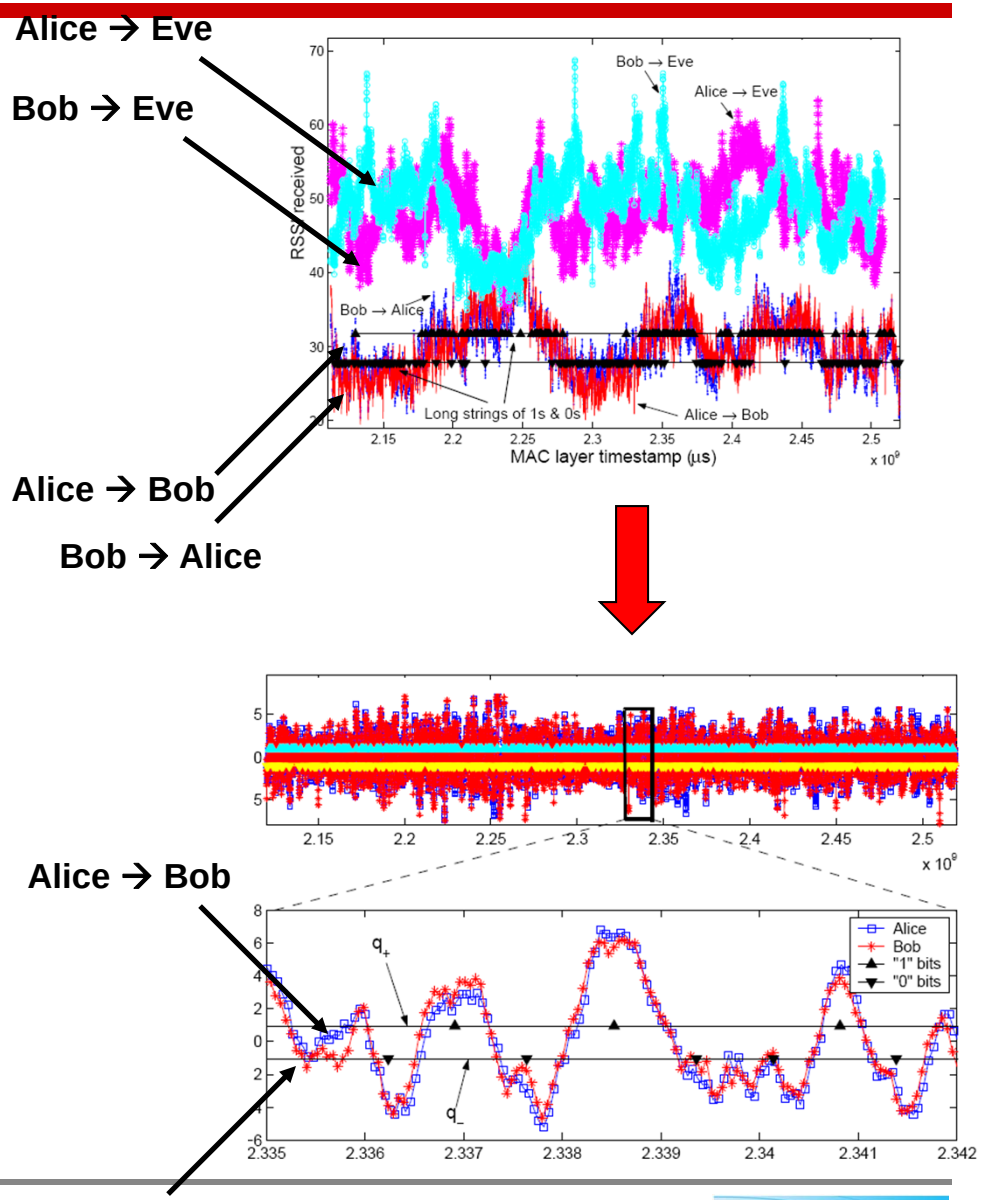




- 64 Point Channel Impulse Response from 802.11a Preamble
- Tallest Peak in CIR Extracted
- STA= Bob, AP =Alice
- Probing of channel: PROBE request and PROBE response
- New PROBE every 110msec

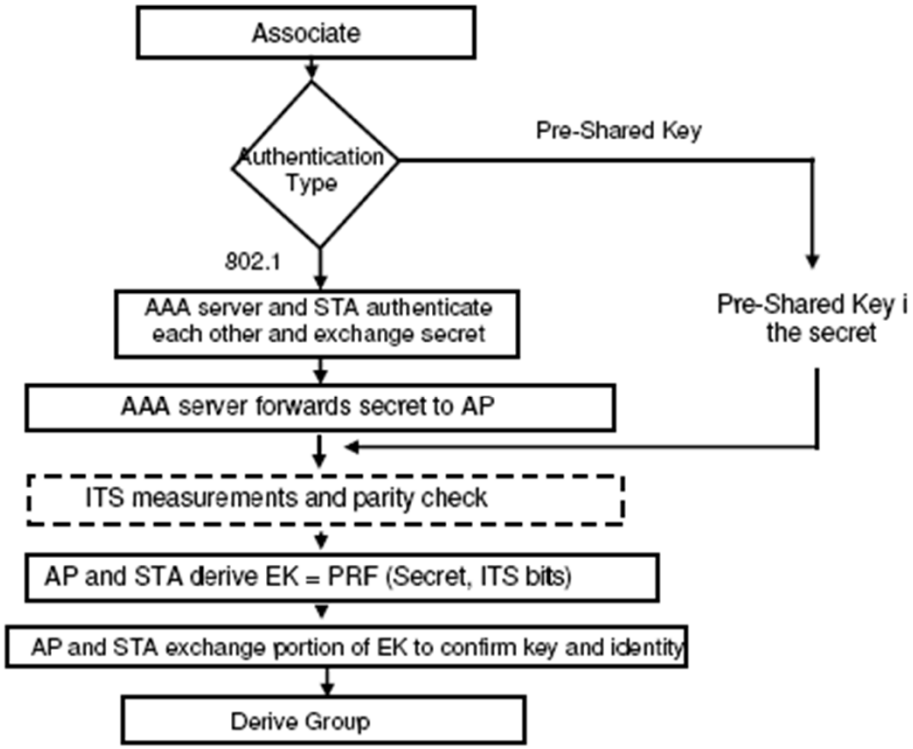
System Validation using 802.11 RSSI

- Experimental setup:
  - Alice = AP
  - Bob = Client
  - Eve = Client on same channel
- Alice → Bob: **PING REQUEST** Bob → Alice: **PING REPLY**
- 20 packets per second
- Eve overhears packets from both legitimate users
- (RSSI, timestamp) from recd. packet headers are pulled out by each user
- Mesg. exchange protocol uses the locations of excursions to distil identical bits
- ~1 bit/sec in typical indoor environments with no errors.

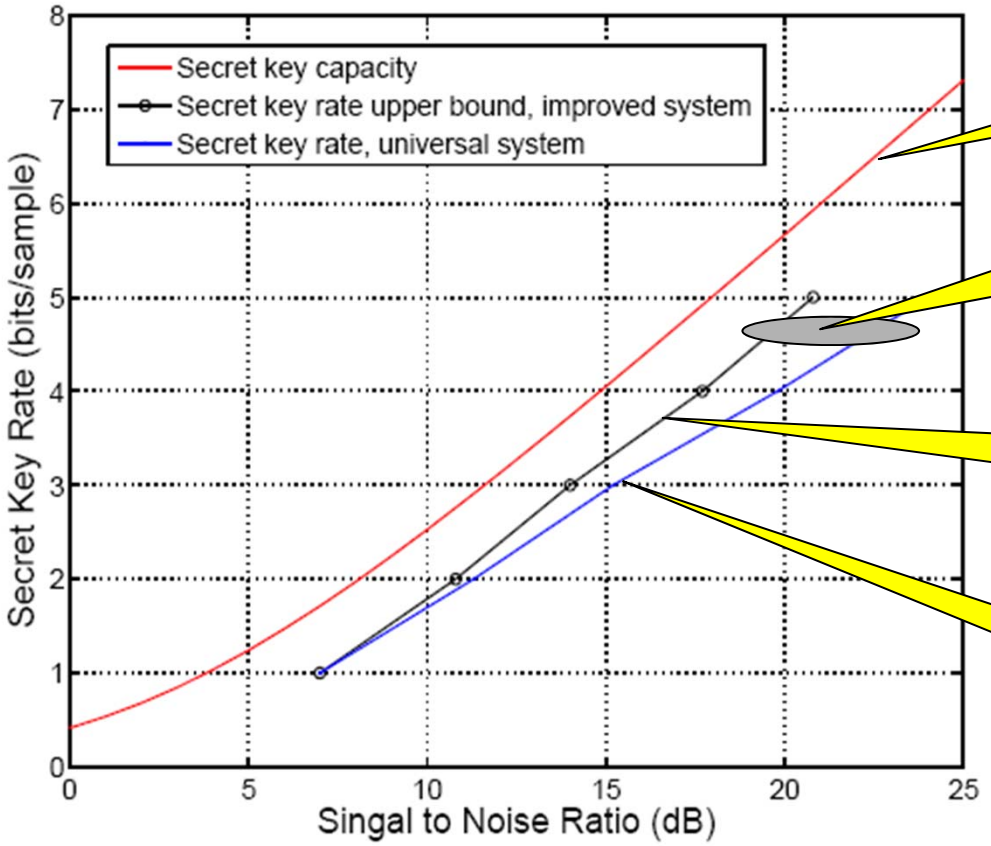


OK, great, but what do I do with 1 bit per second?

- The system-level perspective
  - Can't use in a one-time pad
  - Rates are too-low
- However, remains a valuable resource
  - For example, modified key-refresh of 802.11i (on right)
    - *[Mathur/R./Trappe/Mandayam /Mukherjee/Rahman, sub. to Comm. Mag., 08]*
  - Provides a new 128-bit key about every 2 minutes
  - This is fast enough to effectively prevent some known attacks on WPA
- Other application in real systems – we are working on them



Performance of InterDigital's AirKey System



M.I. for Rayleigh Distribution

As SNR grows, need to quantize ever finely, otherwise the error propagation is not sufficient

What we can do with infinite quantization (error value is real valued)

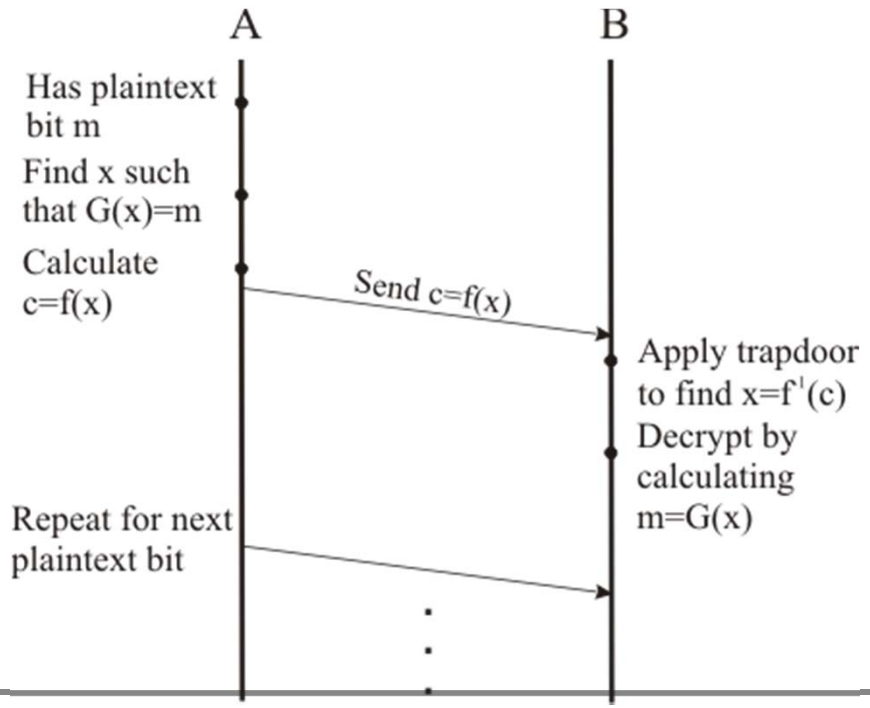
A fixed number of total bits (8)

- For details, see [Mathur/Ye/R./Shah/Trappe/Mandayam, sub. to. Tr. Inf. For. Sec., 2009]

Confidentiality: Dissemination

Dissemination: Probabilistic Encodings

- A powerful approach to secret dissemination borrows from probabilistic encryption and Wyner codes
- Let f(x) be a trapdoor function and G(x) a hardcore predicate for f(x) (i.e. hard to calculate G(x) from just f(x))
- A probabilistic encryption procedure is:



Dissemination: Probabilistic Encodings

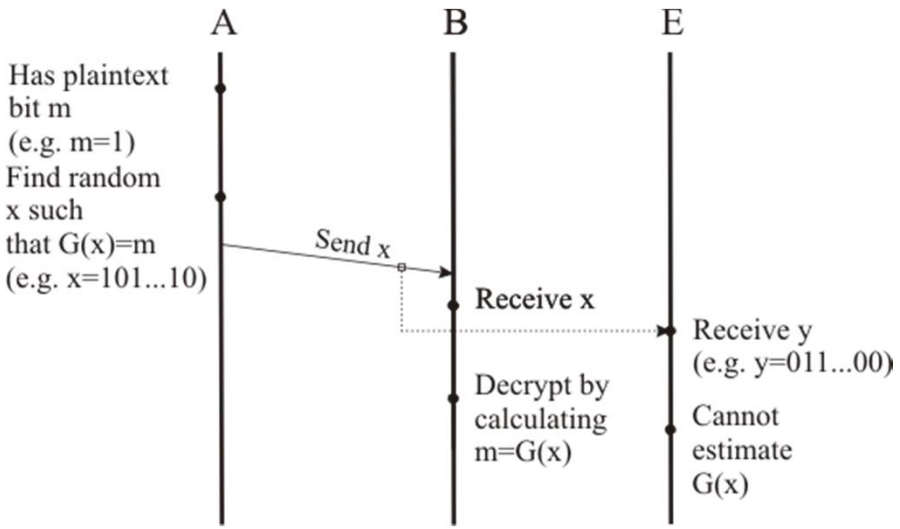
- Applying this idea:
  - Take G(x)=m to be the parity function of x

1. Alice wants to send m=0 or 1 to Bob
2. She chooses a random x of length N such that G(x)=m
3. Transmits x to Bob

- Assuming $p_{AB}=0$, then Bob recovers m by calculating G(x)
- For large N, the probability of an odd amount of bit errors is

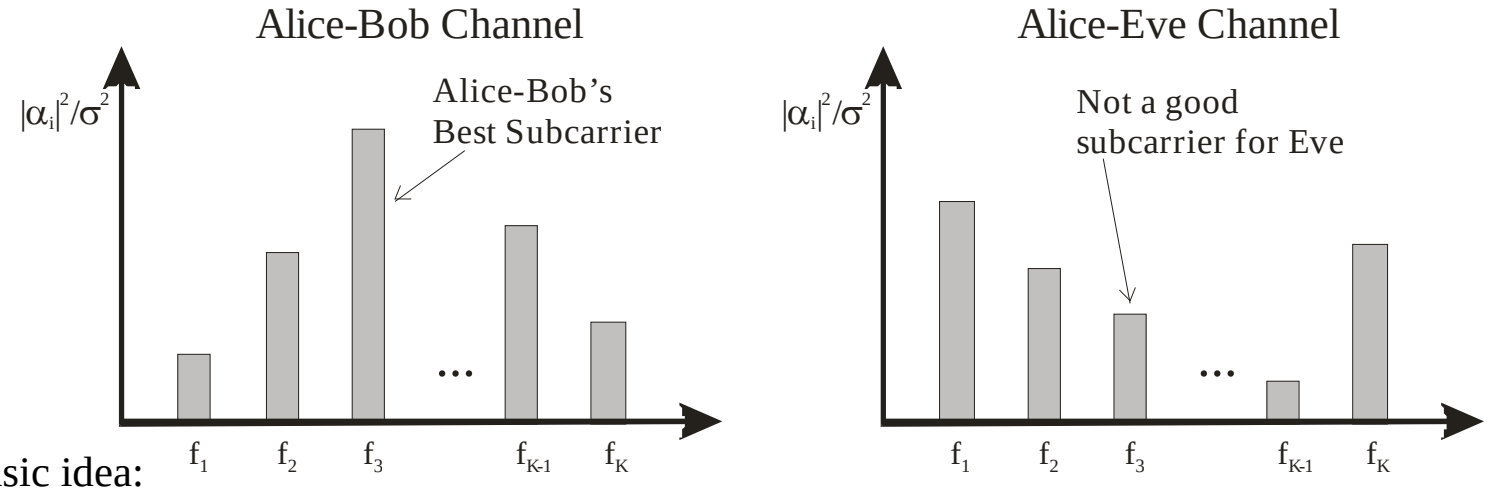
$$0.5\left[1+\left(1-2p_E\right)^N\right]\rightarrow 0.5$$

- More generally, apply ECC across m's to handle p_{AB} not 0
- This method needs $p_{AB} < p_E$!



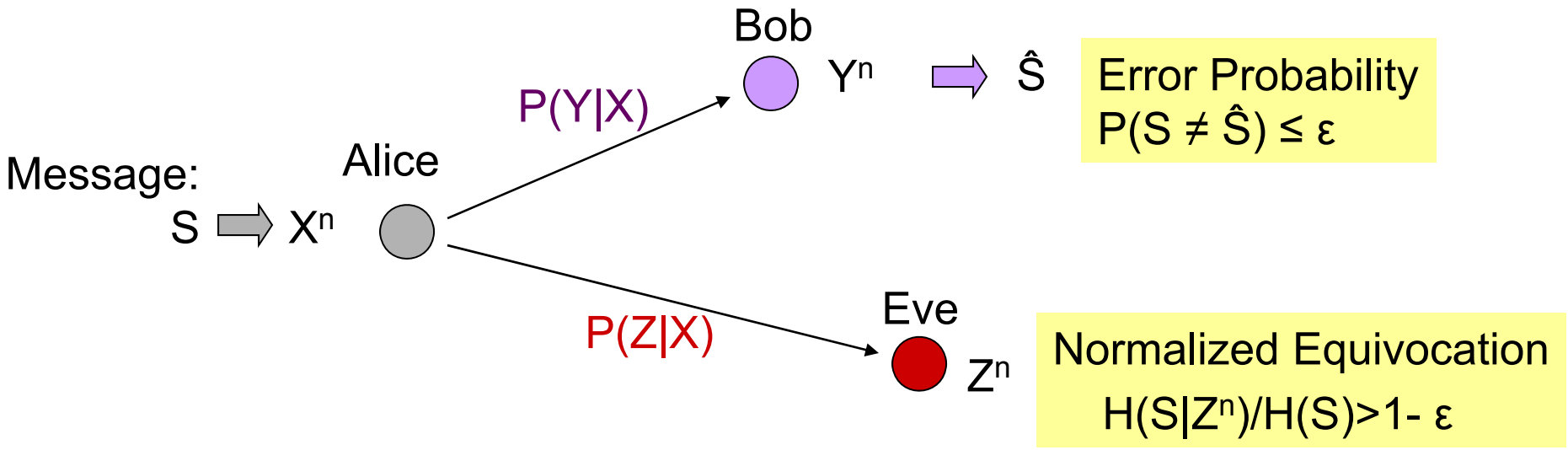
Waterfilling-esque Dissemination

- In order to make this procedure work, we need Alice-Bob's channel to have higher "quality" than Alice-Eve's channel
- Exploiting the independence of carriers for a multipath environment, we may devise a "waterfilling"-style strategy to synthetically degrade Alice-Eve's channel



- Basic idea:
  - Transmit on Alice-Bob's best channel
  - There's a good chance that that Eve's corresponding channel will not be as good!
  - Issue of max-statistics for Alice→Bob versus average statistics for Alice→Eve
  - Idea can be generalized to code across more than one carrier
  - Allerton Paper Result: We show waterfilling achieves secrecy capacity for parallel Gaussian channels
  - Similar ideas apply to MIMO systems!

Secret Communication: Info-Theory



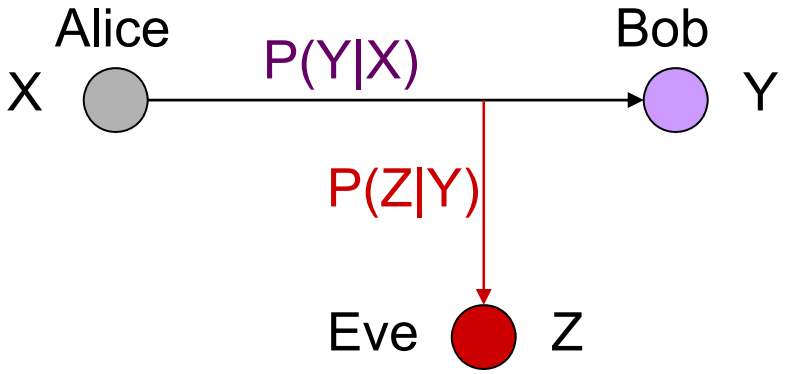
- Reliable transmission requirement
- Perfect secrecy requirement
- Passive Eavesdropper
- **Secrecy Capacity:** maximum reliable rate with perfect secrecy

Wiretap and Broadcast Channels

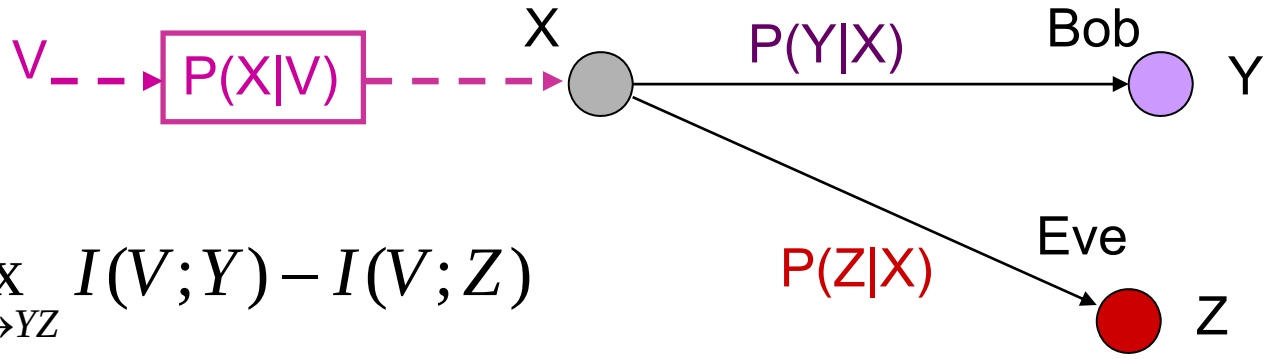
- Wiretap channel (Wyner75)

  Eve has a degraded channel
  $X \rightarrow Y \rightarrow Z$

$$C_{\text{sec}} = \max_{P(x)} I(X;Y) - I(X;Z)$$



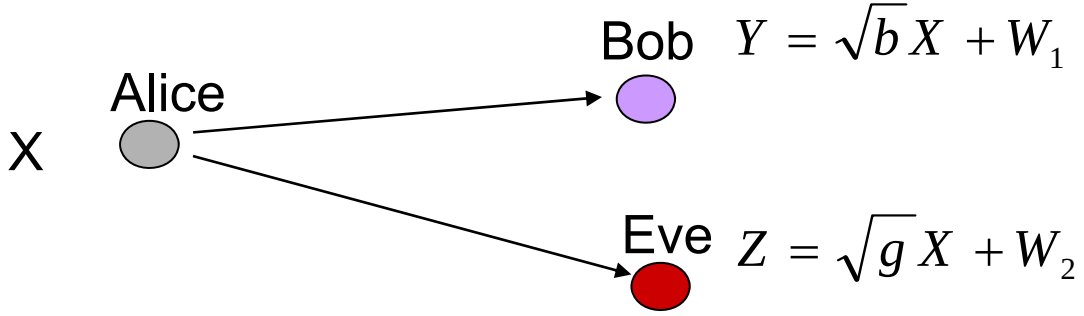
- Broadcast channel (Csiszar & Korner 78)



$$C_{\text{sec}} = \max_{V \rightarrow X \rightarrow YZ} I(V;Y) - I(V;Z)$$

Motivation

Scalar AWGN Broadcast Channel



$$C_{AWGN} = \max_{P(x)} I(X;Y) - I(X;Z) = \frac{1}{2}\left(\log(1+bP) - \log(1+gP)\right)^{+}$$

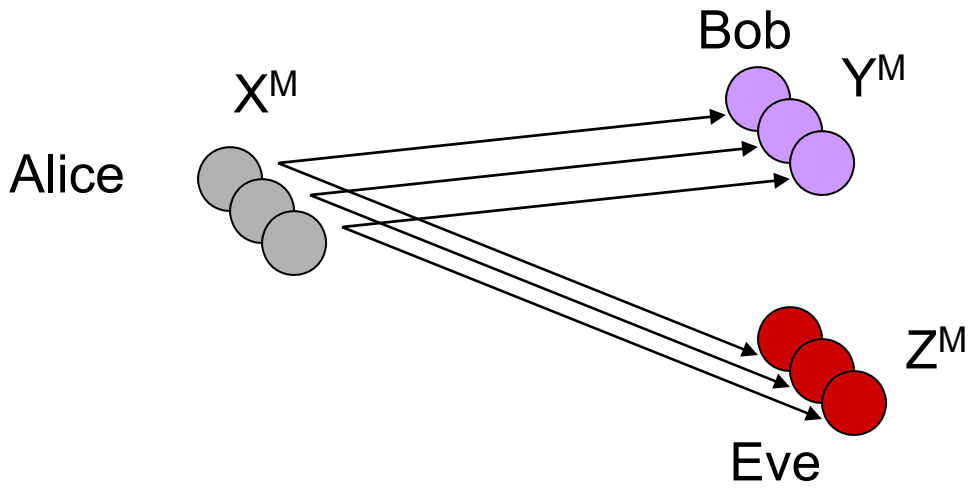
(Leung-Yan-Cheong & Hellman 78, Van Dijk 97)

$\lim_{P\to\infty} C_{AWGN} = \{\frac{1}{2}\log(b/g)\}^{+}$

Ways to improve?

OFDM, Fading channels, Multi-antenna

Independent Parallel Channels



$$C_M = \max_{V \to X^M \to Y^M Z^M} I(V;Y^M) - I(V;Z^M) \quad \textbf{= ?}$$

- More capable condition is not satisfied
  - Some subchannels are better but some may be worse

Secrecy Capacity
Independent Parallel Channels

$$C_M = \max_{V \to X^M \to Y^M Z^M} I(V;Y^M) - I(V;Z^M)$$

$$= \sum_{m=1}^{M} \max_{V_m \to X_m \to Y_m Z_m} I(V_m;Y_m) - I(V_m;Z_m)$$

- The secrecy capacity of the system is the sum of the secrecy capacities of each individual channels
- Channels with zero secrecy capacity should not be used

Independent Parallel AWGN Channels

- When the channels are AWGN, and have a total power constraint P_{tot}

$$C_M = \sum_{m=1}^{M} C_{AWGN}(b_m, g_m, P_m)$$

$$\frac{1}{2}\left(\log(1+b_m P_m) - \log(1+g_m P_m)\right)^+$$

Concave w.r.t P_m when $b_m > g_m$

0 when $b_m \le g_m$

Independent Parallel AWGN Channels

Maximize $$\sum_{m=1}^{M'} \frac{1}{2}\left(\log(1+b_m P_m) - \log(1+g_m P_m)\right)$$

Subject to $$\sum_{i=1}^{M'} P_m \le P_{tot}$$

- Only consider M' subchannels that Bob is better than Eve ($b_m > g_m$)
- Convex optimization problem
- Standard Lagrangian method to find the optimal power allocation

Optimal Power Allocation

- Form the Lagrangian $L(\lambda, P) = \sum_{m=1}^{M'} \left(\log(1 + b_m P_m) - \log(1 + g_m P_m) - \lambda P_m \right)$

Maximize $L(\lambda,P)$ ➡ $\left(P_m + \frac{1}{g_m} \right)\left(P_m + \frac{1}{b_m} \right) - \frac{1}{\lambda}\left(\frac{1}{g_m} - \frac{1}{b_m} \right) = 0$

Solve for non-negative P_m

$$P_{AWGN}(b, g, \lambda) = \frac{1}{2}\left(\sqrt{\left(\frac{1}{b} + \frac{1}{g} \right)^2 + 4\left[\frac{1}{\lambda}\left(\frac{1}{g} - \frac{1}{b} \right) - \frac{1}{gb} \right]} - \left(\frac{1}{b} + \frac{1}{g} \right) \right)^+$$

λ is chosen such that

$$P_{AWGN}(b, g, \lambda) > 0 \quad \Leftrightarrow \quad \frac{1}{\lambda}\left(\frac{1}{g} - \frac{1}{b} \right) - \frac{1}{gb} > 0 \quad \Leftrightarrow \quad b - g > \lambda$$

Similar to waterfilling!

AWGN Fading Channel

- Consider a fading channel with known channel states
- Normalized stationary and ergodic time-varying gains
- Additive white Gaussian noise

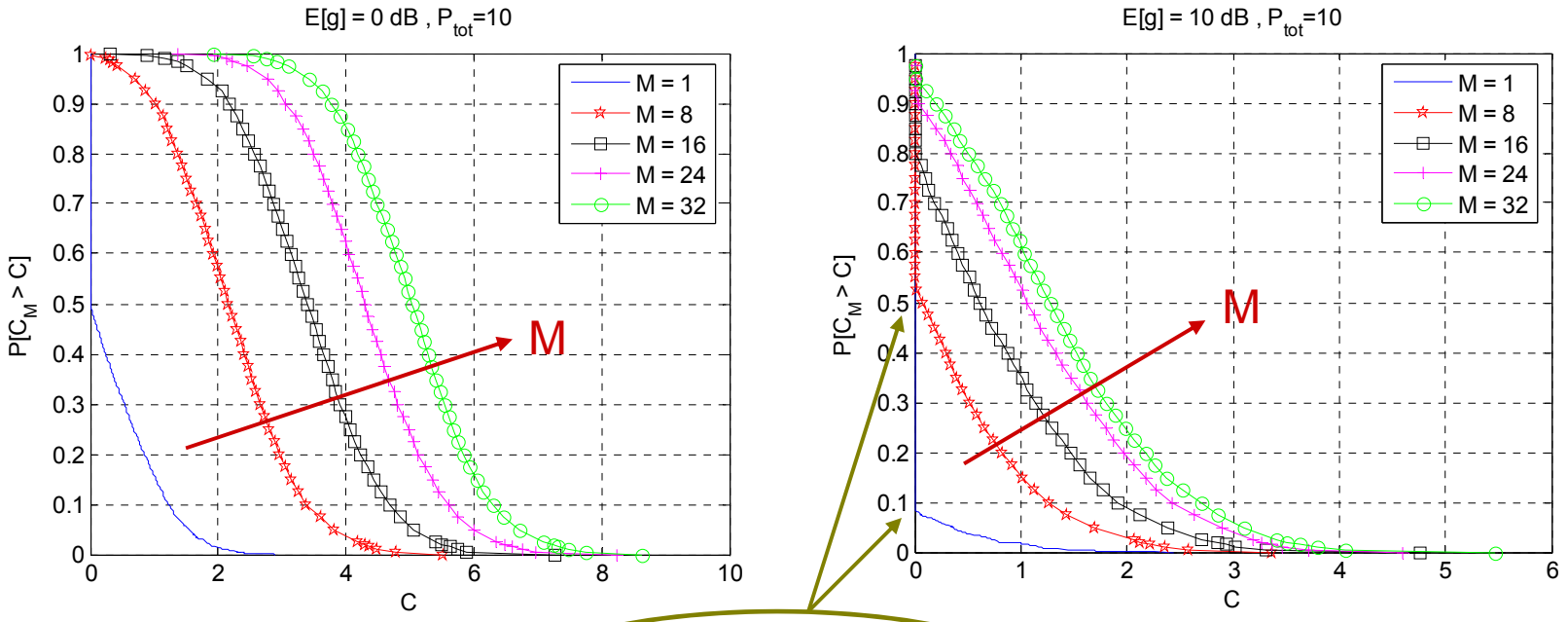
$$Y_t = \sqrt{b_t} X_t + W_{1,t} \qquad Z_t = \sqrt{g_t} X_t + W_{2,t}$$

- It is a special case of independent parallel channels
  - Similar power allocation results
- Multiplexing codebook can achieve the secrecy capacity but is unnecessary
  - A single codebook with adapted transmission power is enough

(Non-Ergodic) Numerical Evaluation

- OFDM system, Rayleigh fading channel gains
- Factors affecting the secrecy capacity
  - M : total number of channels (bandwidth)
  - b & g : Bob & Eve channel gains
  - P_{tot} : Total power constraint
- Examine the variation of secrecy capacity
  - b & g : exponential distribution. Fix $E[b]=1$ and vary $E[g]$
    - ***E[g]=1****: Equal quality for Bob and Eve*
    - ***E[g]=10****: Eve is 10dB better then Bob*
  - C_{sec} is a random variable depending on the channel realization
  - Compute the Complementary CDF $P(C_{sec}>C)$ numerically

Secrecy Capacity Improves with M



Pr[$b_i > g_i$ for some i] = $1-\left(\Pr(b \le g)\right)^M = 1-\left(\frac{E[g]}{E[b]+E[g]}\right)^M$

Diversity helps !

(Barros & Rodrigues ISIT06 did similar outage formulation)

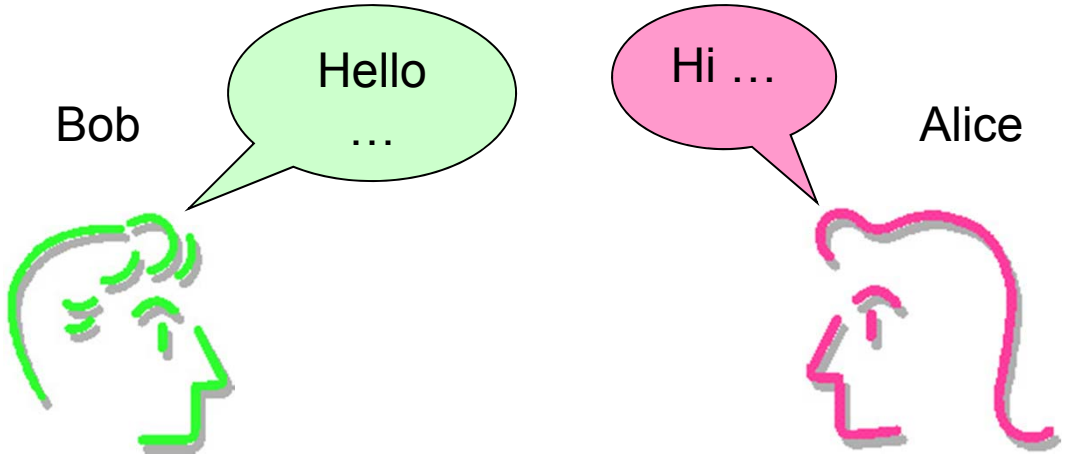
References for the Weary

"I think I should understand that better, if I had it written down: but I can't quite follow it as you say it…" (Alice)

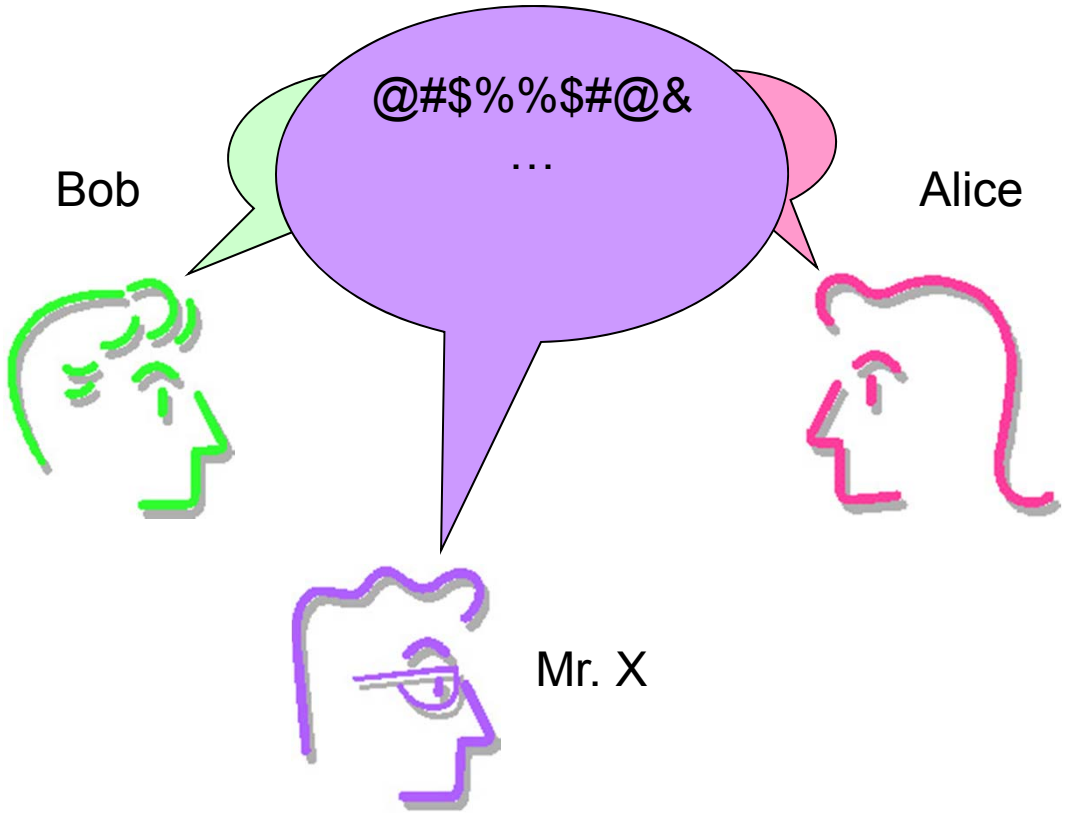
- U. Maurer, "Secret key agreement by public discussion from common information," IEEE Transactions on Information Theory, vol. 39, no. 4, pp. 733–742, 1993.
- L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe, "Fingerprints in the ether: Using the physical layer for wireless authentication," in Proceedings of the IEEE Int.. Conf. on Comm., pp. 4646 – 4651.
- L. Xiao, L. J. Greenstein, N. B. Mandayam, and W. Trappe, "A physical-layer technique to enhance authentication for mobile terminals," *IEEE International Conference on Communications* (ICC), 2008
- L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe, "Using the physical layer for wireless authentication under time-variant channels," *IEEE Transactions on Wireless Communications*.
- B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener, "Robust key generation from signal envelopes in wireless networks," in CCS '07: Proceedings of the 14th ACM Conference on Computer and communications security, 2007, pp. 401–410.
- A. Hassan, W. Stark, J. Hershey, and S. Chennakeshu, "Cryptographic key agreement for mobile radio," Digital Signal Processing, vol. 6, pp. 207–212, 1996.
- R. Wilson, D. Tse, and R. Scholtz, "Channel identification: Secret sharing using reciprocity in UWB channels," IEEE Transactions on Information Forensics and Security, vol. 2,no. 3, pp. 364–375, 2007.
- T. Aono, K. Higuchi, T. Ohira, B. Komiyama, and H. Sasaoka, "Wireless secret key generation exploiting reactance-domain scalar response of multipath fading channels," IEEE Transactions on Antennas and Propagation, vol. 53, no. 11, pp. 3776–3784, Nov 2005.
- Suhas Mathur , Wade Trappe, Narayan Mandayam, Chunxuan Ye and Alex Reznik, "Radio-telepathy: Extracting a Cryptographic Key from an Un-authenticated Wireless Channel," The 14th Annual International Conference on Mobile Computing and Networking, (ACM MobiCom 2008).

Detection and Defense against Jamming Attacks in Wireless Networks

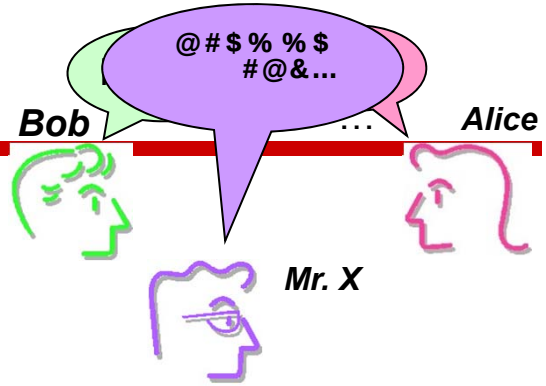
Jamming Style DoS



Jamming Style DoS



Jamming Attacks



- *Jamming Attacks:*
  - *Behavior that prevents other nodes from using the channel to communicate by interfering with the physical transmission and reception of wireless communications*

- *Unintentional jamming:*
  - Co-existing devices: 802.11b/g interferes with cordless phone, Bluetooth, Microwave oven...
  - Equipment accidentally emits a signal on an frequency band that does not belong to it.

- *Intentional jamming:*
  - A transmitter, tuned to the same frequency as the receiving equipment, can override any signal with enough power

The history of jamming

- World War II – Radio jamming
  - Jamming radar that is used to guide an enemy's aircraft
    - *Mechanical jamming*.
      - Chaff, corner reflectors, decoys
    - *Electrical jamming*
      - Spot jamming, sweep jamming…
  - Jamming foreign radio broadcast stations
    - *Prevent or deter citizens from listening to broadcasts from enemy countries.*

- Countermeasure:
  - Frequency hopping over a broad-spectrum
    - *The more random the frequency change, the more likely the jammer*

Jamming in the civilian world

- Cell phone jammer unit:
  - Intended for blocking all mobile phone types within designated indoor areas
  - 'plug and play' unit
  - $1,950-$11,800+

- Radar/speed gun jammers (Illegal!)
  - $100 - $2,000+

- Radio Jammers (Illegal!)
  - Your neighbor plays loud radio while you are preparing for your exam
  - Prevent nearby cars from playing loud music by broadcasting your own signal

Jamming wireless networks

- Waveform Generator
  - Tune frequency to whatever you want
  - $1,500 - $50,000+
  - Require external power supply

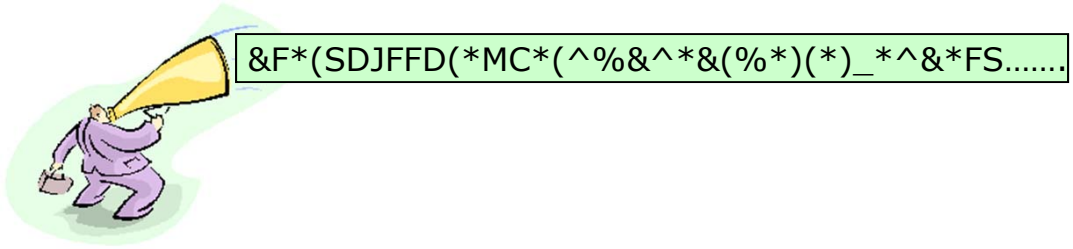
- MAC-layer Jammers
  - 802.11 laptop
  - Mica2 Motes (UC Berkeley)
    - *8-bit CPU at 4MHz*
    - *128KB flash, 4KB RAM*
    - *916.7MHz radio*
    - *OS: TinyOS*
    - *Language: NesC*
  - Disable the CSMA
  - Keep sending out the preamble

What has been done?

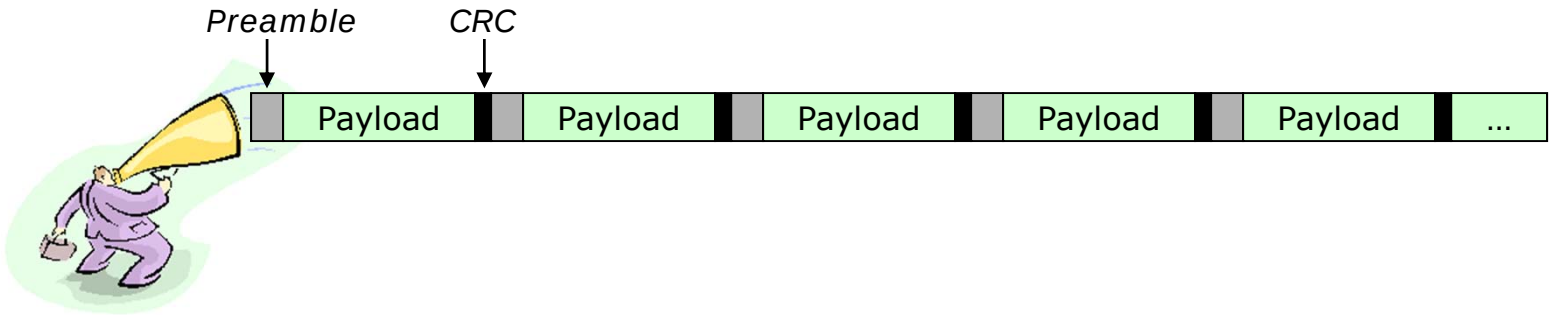
- Somewhat related work on jamming:
  - Greedy user behaviors
    - *DOMINO: system for detection of greedy behavior in the MAC layer of IEEE 802.11 public Networks [Hubaux04]*
  - 802.11 DoS attacks
    - *802.11 Denial of Service attacks [Savage03]*
    - *Attacks that jam RTS, and floods RTS [Perrig03]*

- Work on jamming attacks:
  - Mapping a jamming-area for sensor networks
    - *Brief discussion on jamming detection [Stankovic03]*
  - Countermeasure against jamming attacks
    - *Traditional physical layer technologies – Spread Spectrum [DigComm00], [WarFare99]*
    - *Low density parity check codes (LDPC) [Noubir03]*
  - Channel capacity of jamming channels
    - *The capacity of Correlated Jamming Channels [Medard97]*

Jammer Attack Models

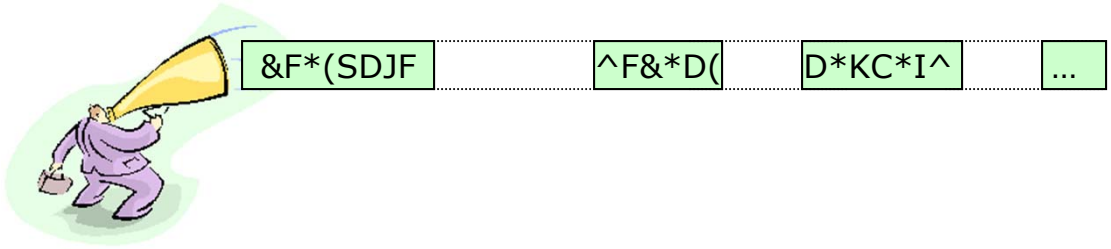


- Constant jammer:
  - Continuously emits a radio signal

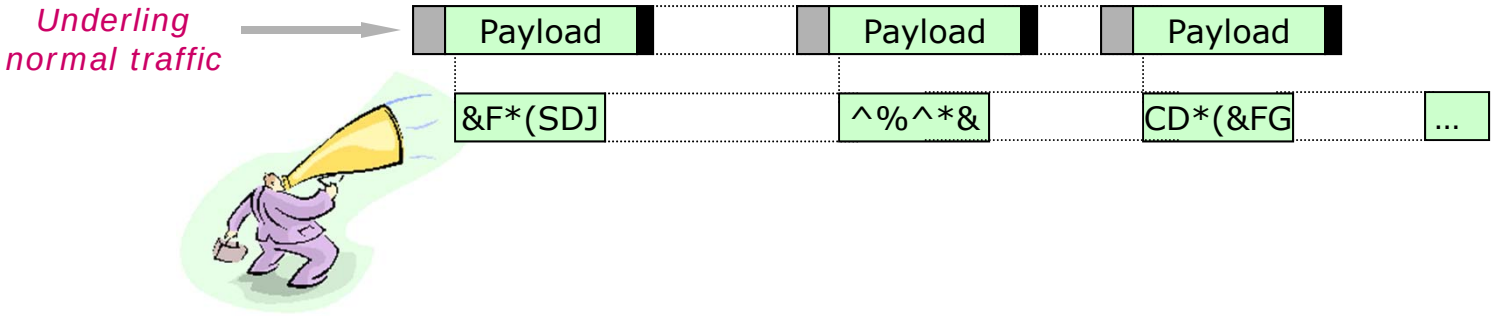


- Deceptive jammer:
  - Constantly injects regular packets to the channel without any gap between consecutive packet transmissions
  - A normal communicator will be deceived into the receive state

Jammer Attack Models



- Random jammer:
  - Alternates between sleeping and jamming
    - *Sleeping period: turn off the radio*
    - *Jamming period: either a constant jammer or deceptive jammer*



- Reactive jammer:
  - Stays quiet when the channel is idle, starts transmitting a radio signal as soon as it senses activity on the channel.
  - Targets the reception of a message

Metrics & Implementation

- Goals of the jammer:
  - Interfere with legitimate wireless communications
  - Prevent a sender from sending out packets
  - Prevent a receiver from receiving a legitimate packets
- Packet Send Ratio (PSR)
  - The ratio of packets that are successfully *sent out* by a legitimate traffic source compared to the number of packets it intends to send out in the *MAC layer*
- Packet Delivery Ratio (PDR)
  - The ratio of packets that are successfully *delivered* to a destination compared to the number of packets that have been sent out by the sender
- Implementation platform:
  - Mica2 Motes
  - Disabled channel sensing and backoff operation in TinyOS MAC protocol

Experimental Results

- Involved three parties:
  - Normal nodes:
    - *Sender A*
    - *Receiver B*
  - Jammer X

- Parameters
  - Four jammer models
  - Distance
    - *Let* $d_{XB} = d_{XA}$
    - *Fix* d_{AB} *at 30 inches*
  - Power
    - $P_A = P_B = P_X = -4dBm$
  - MAC
    - *Fix MAC threshold*
    - *Adaptive MAC threshold (BMAC)*

Deceptive Jammer		
d_{xa} (inch)	PSR(%)	PDR(%)
38.6	0.00	0.00
54.0	0.00	0.00
72.0	0.00	0.00

Reactive Jammer			
d_{xa} (inch)		PSR(%)	PDR(%)
m = *7bytes*	38.6	99.00	0.00
	54.0	100.0	99.24
m = *33bytes*	38.6	99.00	0.00
	54.0	99.25	98.00

Experimental Results

- Involved three parties:
  - Normal nodes:
    - *Sender A*
    - *Receiver B*
  - Jammer X

- Parameters
  - Four jammer models
  - Distance
    - *Let* $d_{XB} = d_{XA}$
    - *Fix* d_{AB} *at 30 inches*
  - Power
    - $P_A = P_B = P_X = -4dBm$
  - MAC
    - *Fix MAC threshold*
    - *Adaptive MAC threshold (BMAC)*

Deceptive Jammer		
d_{xa} (inch)	PSR(%)	PDR(%)
38.6	0.00	0.00
54.0	0.00	0.00
72.0	0.00	0.00

Reactive Jammer			
d_{xa} (inch)		PSR(%)	PDR(%)
m = *7bytes*	38.6	99.00	0.00
	54.0	100.0	99.24
m = *33bytes*	38.6	99.00	0.00
	54.0	99.25	98.00

Experimental Results

- Involved three parties:
  - Normal nodes:
    - *Sender A*
    - *Receiver B*
  - Jammer X

- Parameters
  - Four jammer models
  - Distance
    - *Let* $d_{XB} = d_{XA}$
    - *Fix* d_{AB} *at 30 inches*
  - Power
    - $P_A = P_B = P_X = -4dBm$
  - MAC
    - *Fix MAC threshold*
    - *Adaptive MAC threshold (BMAC)*

Deceptive Jammer		
d_{xa} (inch)	PSR(%)	PDR(%)
38.6	0.00	0.00
54.0	0.00	0.00
72.0	0.00	0.00

Reactive Jammer			
d_{xa} (inch)		PSR(%)	PDR(%)
m = *7bytes*	38.6	99.00	0.00
	54.0	100.0	99.24
m = *33bytes*	38.6	99.00	0.00
	54.0	99.25	98.00

Experimental Results

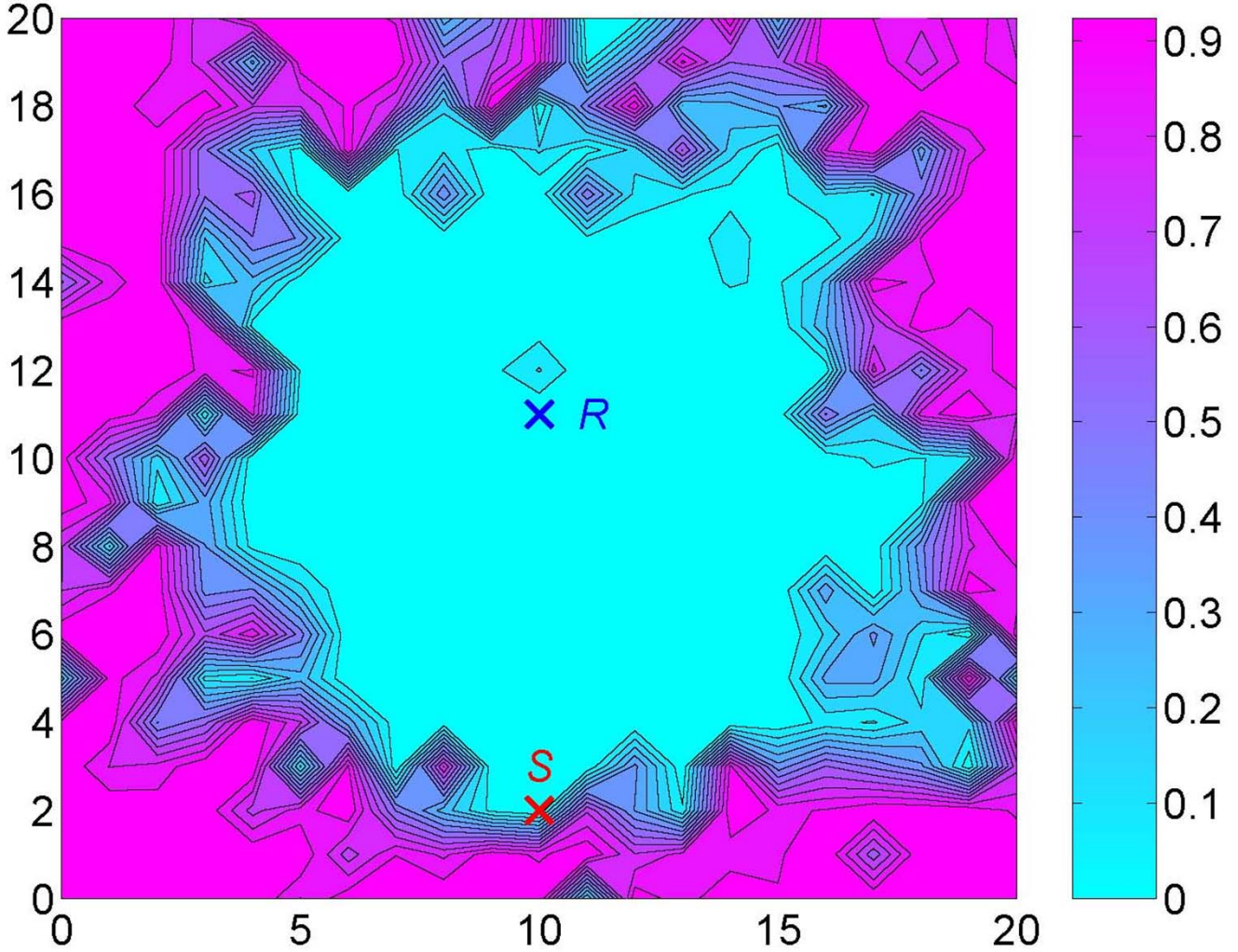
- Involved three parties:
  - Normal nodes:
    - *Sender A*
    - *Receiver B*
  - Jammer X

- Parameters
  - Four jammer models
  - Distance
    - *Let* $d_{XB} = d_{XA}$
    - *Fix* d_{AB} *at 30 inches*
  - Power
    - $P_A = P_B = P_X = -4dBm$
  - MAC
    - *Fix MAC threshold*
    - *Adaptive MAC threshold (BMAC)*

Deceptive Jammer		
d_{xa} (inch)	PSR(%)	PDR(%)
38.6	0.00	0.00
54.0	0.00	0.00
72.0	0.00	0.00

Reactive Jammer			
d_{xa} (inch)		PSR(%)	PDR(%)
m = *7bytes*	38.6	99.00	0.00
	54.0	100.0	99.24
m = *33bytes*	38.6	99.00	0.00
	54.0	99.25	98.00

Radio irregularity- PDR Contour

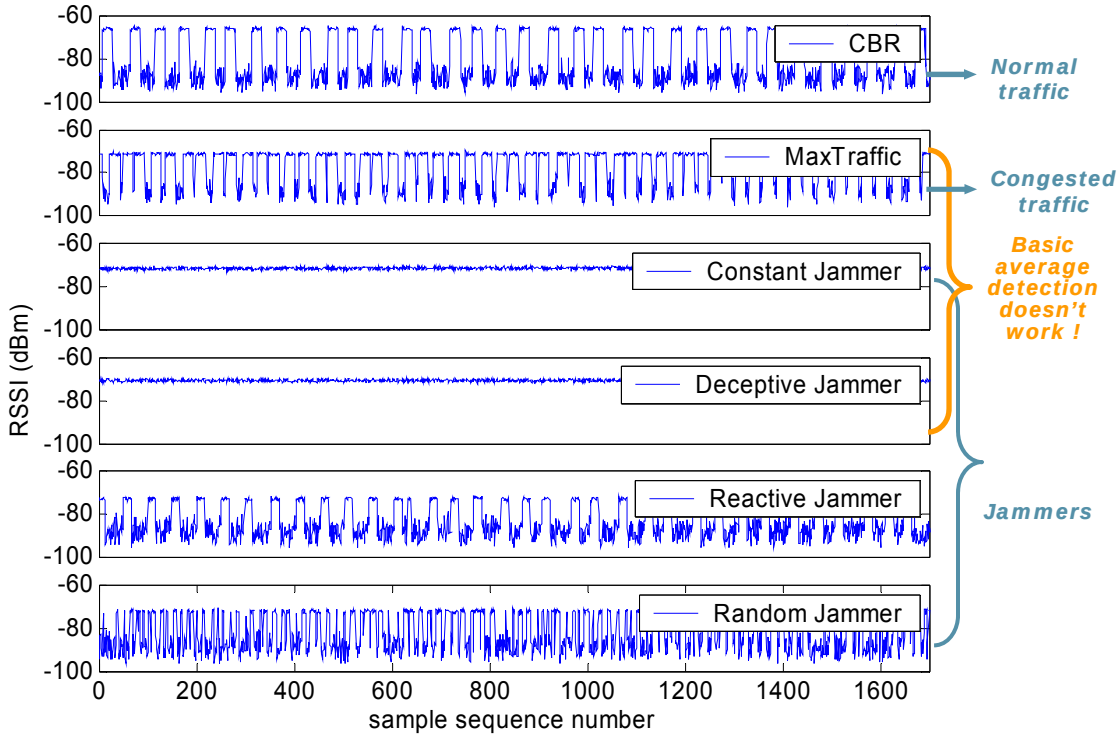


Basic Detection Statistics

- Idea:
  - Many measurements will be affected by the presence of a jammer
  - Network devices can gather measurements during a time period prior to jamming and build a statistical model describing basic measurements in the network

- Measurements
  - Signal strength
    - *Moving average*
    - *Spectral discrimination*
  - Carrier sensing time
  - Packet delivery ratio

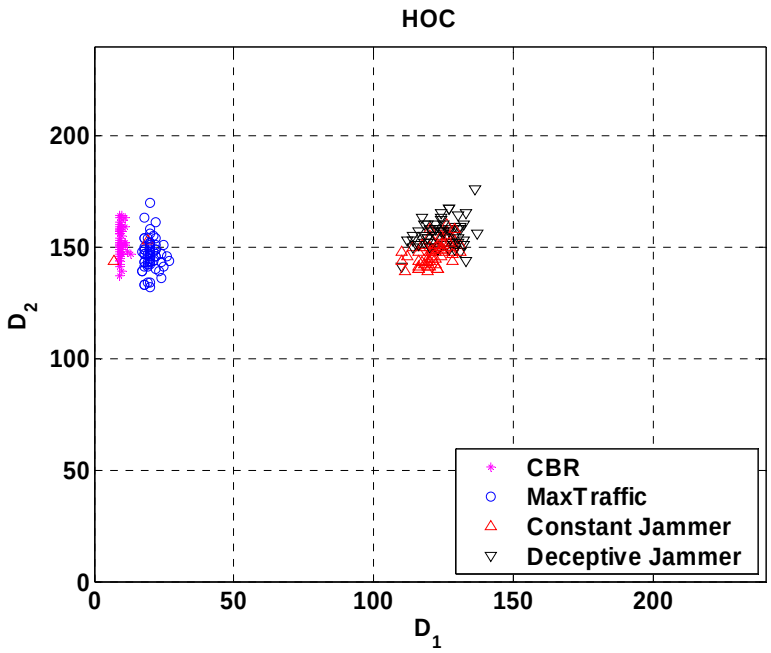
- Experiment platform:
  - Mica2 Motes
  - Use RSSI ADC to measure the signal strength

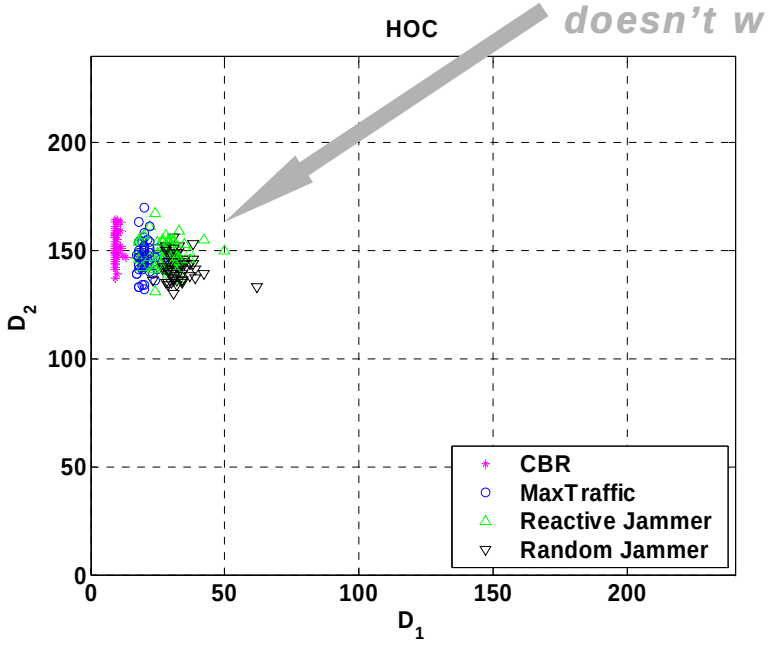


Signal Strength Detection

- Basic Average and Energy Detection don't work!
- How about spectral discrimination mechanism?
  - Higher Order Crossing (HOC)
    - *Combine zero-crossing counts in stationary time series with linear filters.*
    - *Calculate the first two higher order crossings for the time series.*
    - *Window size: 240 samples*

SS spectral discrimination doesn't work !





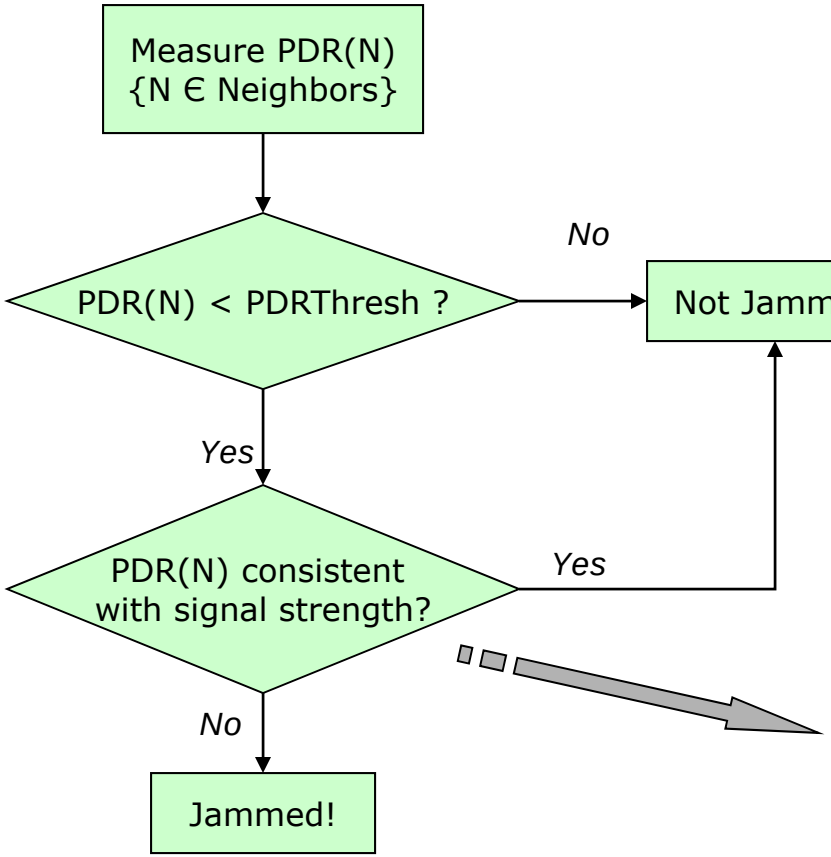
Basic Detection Statistics

- Can basic statistics differentiate between *jamming scenarios* and normal scenarios including *congested scenarios*?

	Signal strength		Carrier sensing time	Packet delivery ratio
	Average	Spectral Discrimination		
Constant Jammer	✗	✓	✓	✓
Deceptive Jammer	✗	✓	✓	✓
Random Jammer	✗	✗	✗	✓
Reactive Jammer	✗	✗	✗	✓

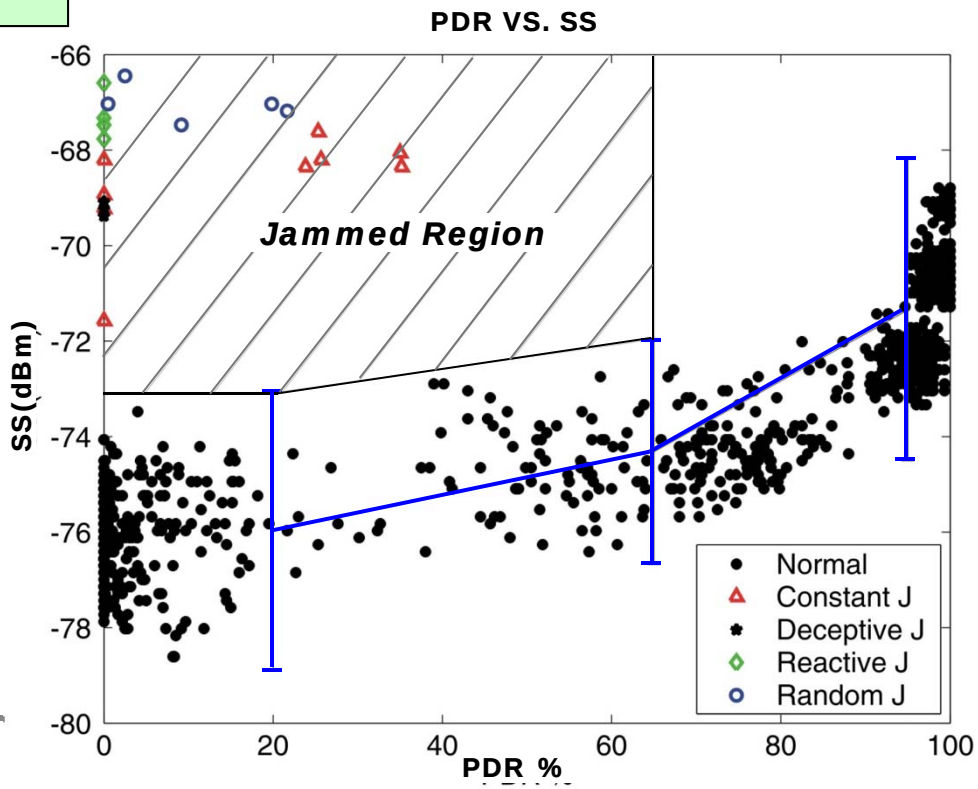
- Differentiate jamming scenario from all network dynamics, e.g. congestion, *hardware failure*
  - PDR is a relatively good statistic, but cannot handle hardware failure
  - Consistency checks --- using Signal strength
    - *Normal scenarios:*
      - High signal strength → a high PDR
      - Low signal strength → a low PDR
    - *Low PDR:*
      - Hardware failure or poor link quality → low signal strength
      - Jamming attack → high signal strength

Jamming Detection with Consistency Checks



Build a (PDR,SS) look-up table empirically

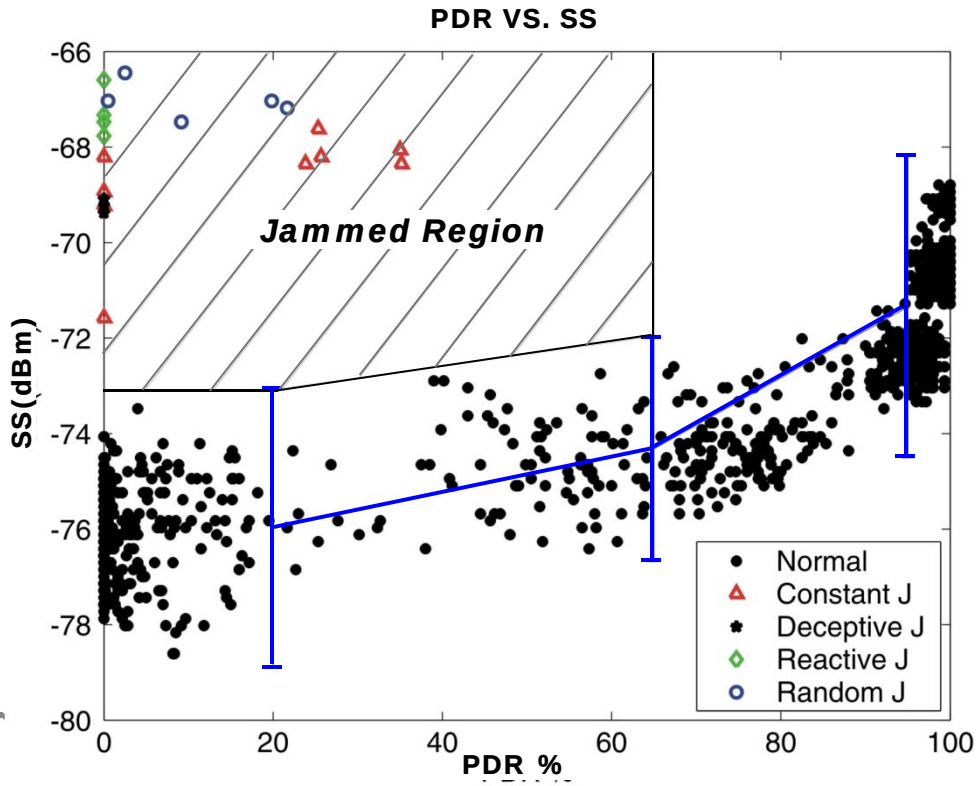
- Measure (PDR, SS) during a guaranteed time of non-interfered network operation
- Divide the data into PDR bins, calculate the mean and variance for the data within each bin.
- Get the upper bound for the maximum SS that world have produced a particular PDR value during a normal case.
- Partition the (PDR, SS) plane into a jammed-region and a non-jammed region.



Jamming Detection with Consistency Checks

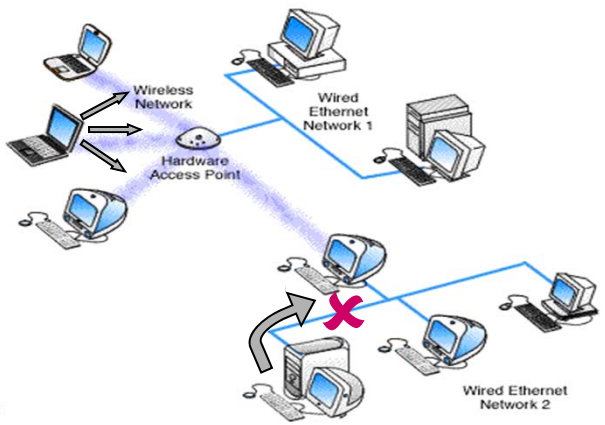
- Jammer setup:
  - Transmission power: -4dBm
  - The reactive jammer injects 20-byte long packets
  - The random jammer turns on for t_j = U[0,31] and turns off for t_s = U[0,31]
- The (PDR, SS) values for all jammers distinctively fall within the jammed-region

- The more aggressive the jammer is, the more likely it will be detected.
- The less aggressive the jammer is, the less damage it causes to the network.
- Similarly, we can deploy a *location information* based consistency check to achieve an enhanced jamming detection.



Handling Jamming: Strategies

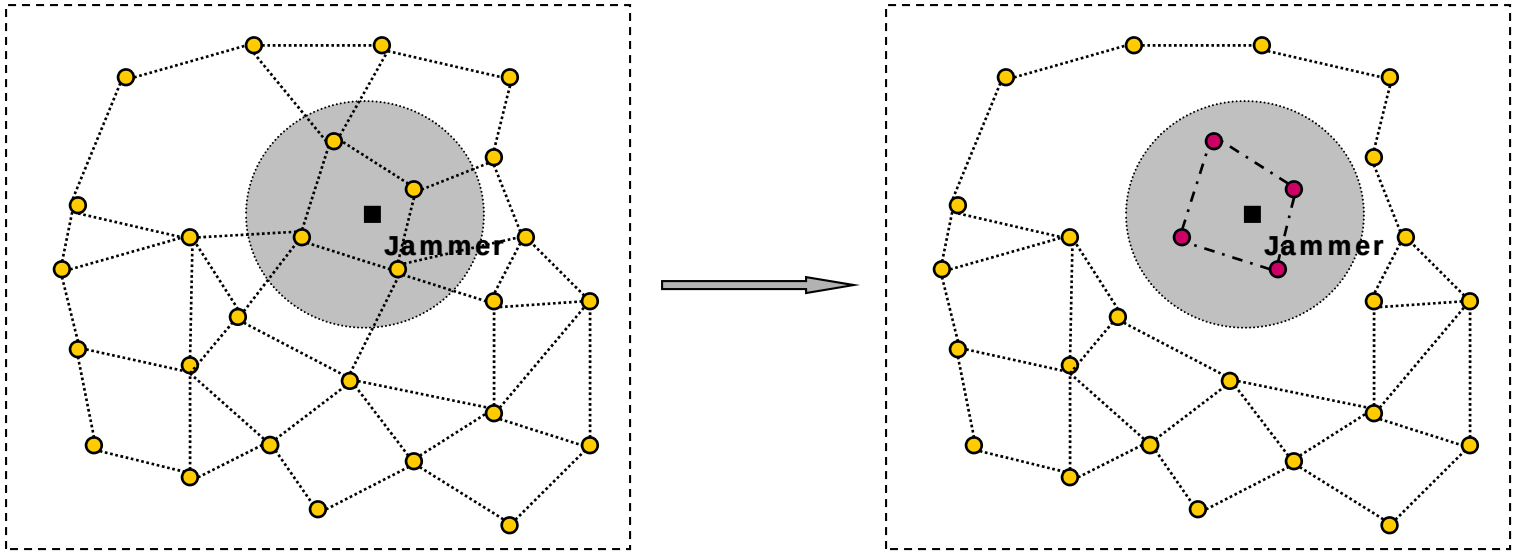
- *What can you do when your channel is occupied?*
  - In wired networks you can cut the link that causes the problem, but in wireless…
  - Make the building as resistant as possible to incoming radio signals?
  - Find the jamming source and shoot it down?
  - Battery drain defenses/attacks are not realistic!
- *Protecting networks is a constant battle between the security expert and the clever adversary.*
- One approach: ***He who cannot defeat his enemy should retreat*** ("Thirty-Six Stratagems").
- Retreat Strategies:
  - Channel surfing
  - Spatial retreat
  - Routing



Channel Surfing

- Idea:
  - If we are blocked at a particular channel, we can resume our communication by switching to a "safe" channel
  - Inspired by frequency hopping techniques, but operates at the link layer in an on-demand fashion.

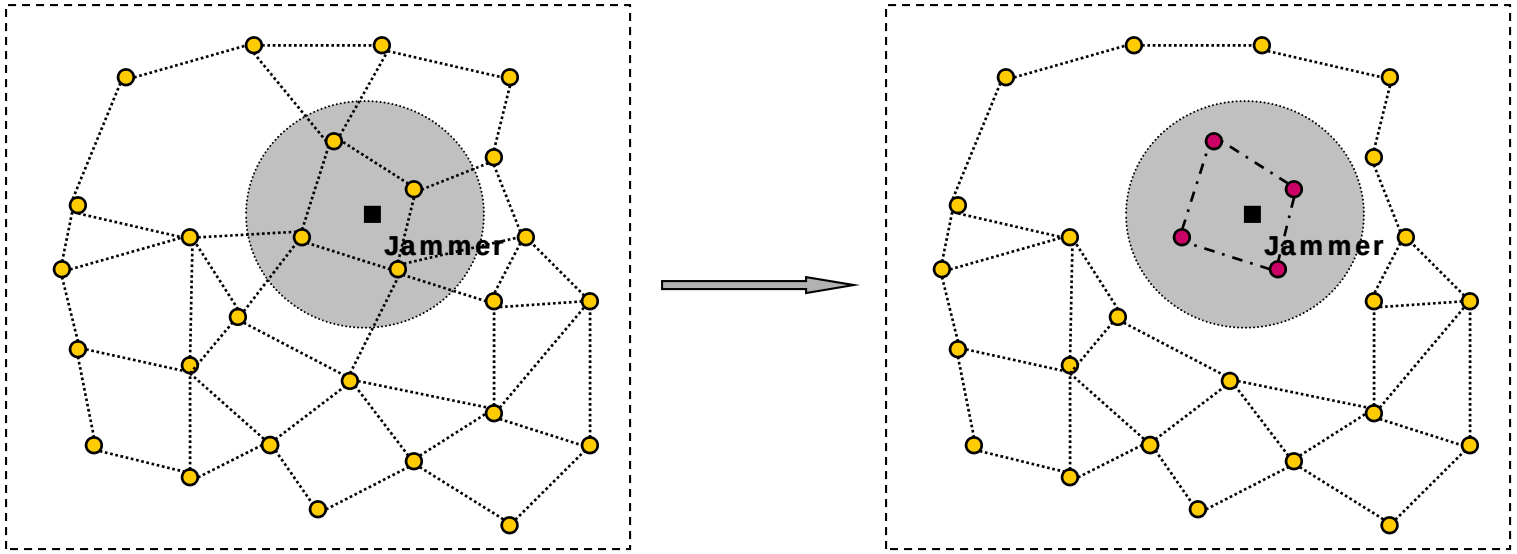
- Challenge
  - Distributed computing
  - Asynchrony, latency and scalability



Channel Surfing Framework

- Channel Surfing Algorithm:

```
While (1) do
    if NeighborsLost() == True then
        working_channel = next_channel;
        if FindNeighbor() == False then
            working_channel = original_channel
        else
            Use a Channel Surfing Strategy
        end
    end
end
```



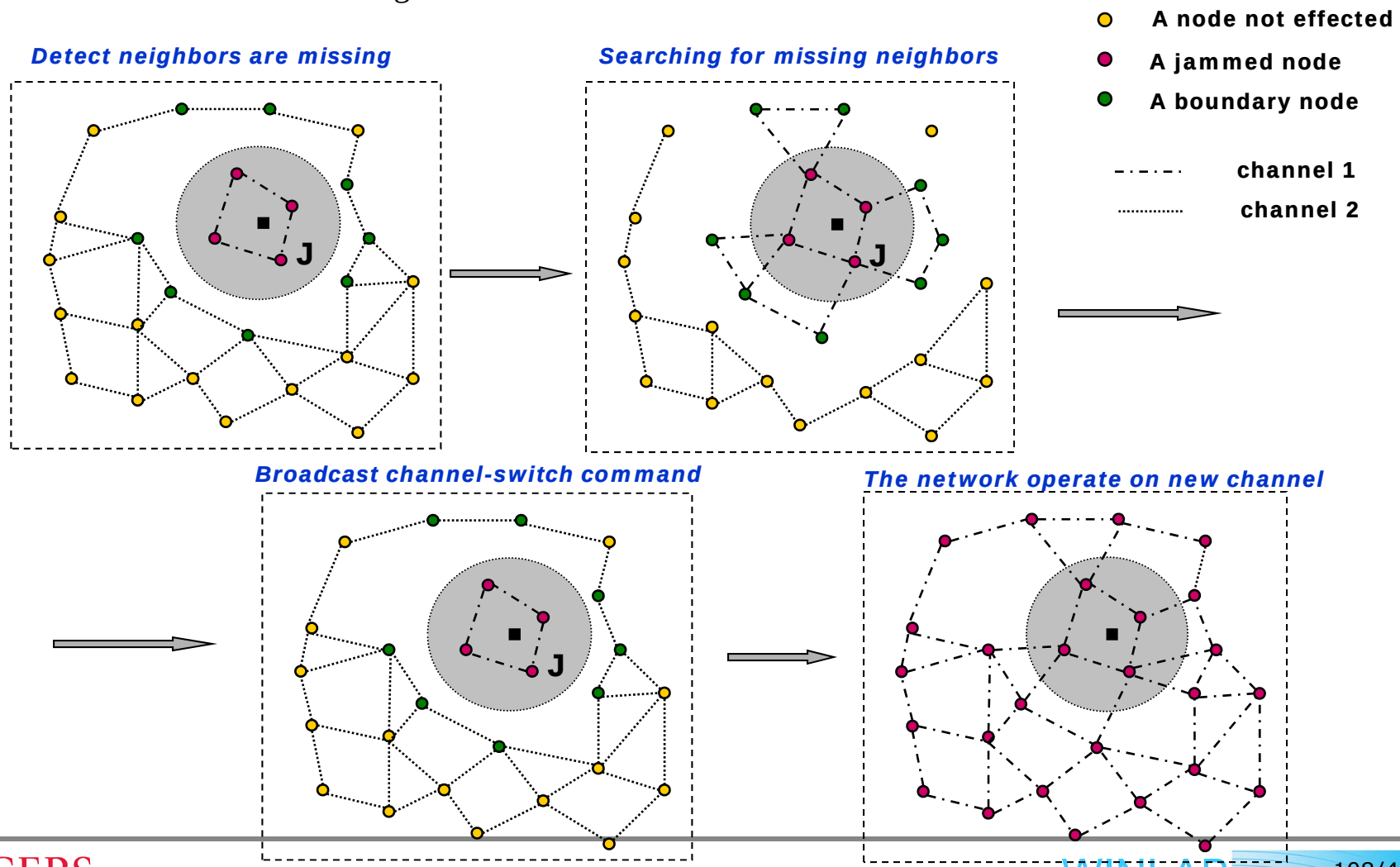
Channel Surfing Framework

- Issues
  - How does a node detect that its neighbor is missing?
    - *Link quality*
  - How to ensure the boundary nodes find their missing neighbors in the new channel?
    - *It takes less time for a node to detect the absence of a neighbor than it does for a node to decide it is jammed.*
  - How to choose the new channel?
    - *Make it harder for the adversary to predict*
    - *Keyed pseudo-random generator*
    - $C(n+1) = E_k(C(n))$
  - How to resume the network connectivity?

```
While (1) do
    if NeighborsLost() == True then
                working_channel = next_channel;
                if FindNeighbor() == False then
                    working_channel = original_channel
                else
                    Use a Channel Surfing Strategy
                end
    end
end
```

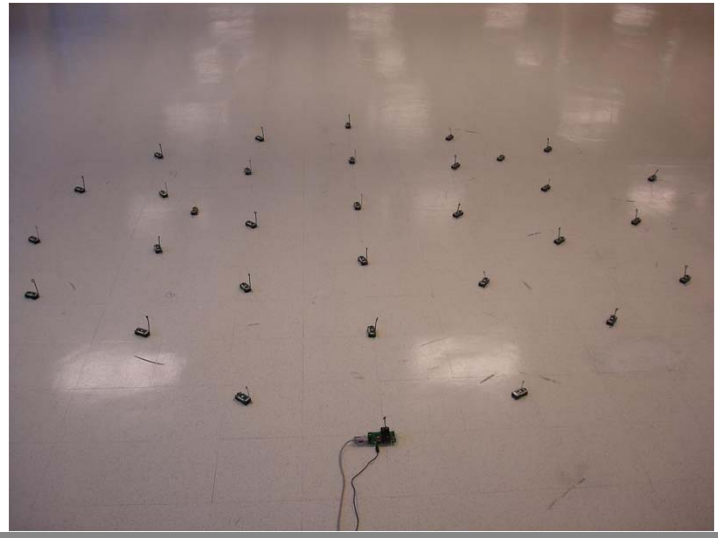
Coordinated Channel Surfing

- Coordinated Channel Surfing
  - The entire network changes its channel to a new channel



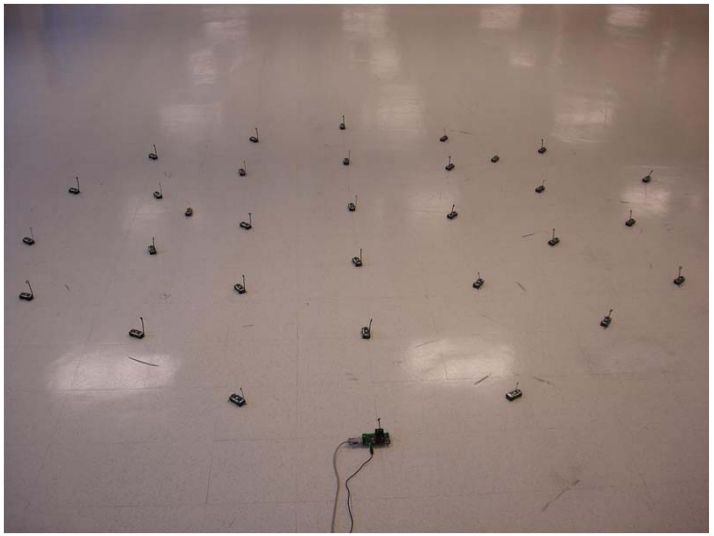
Strategy validation

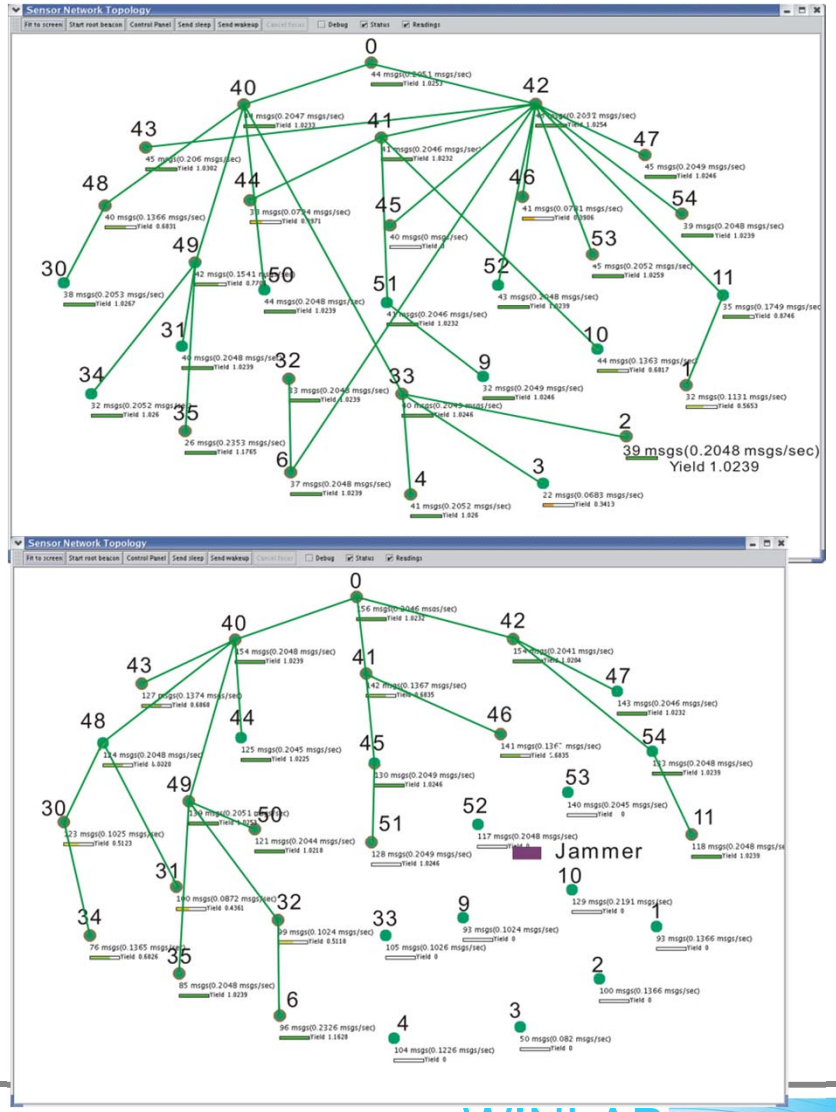
- Mica2 Motes
  - *8-bit CPU at 4MHz,*
  - *128KB flash, 4KB RAM*
  - *916.7MHz radio*
  - *OS: TinyOS*
- Debugging facilities:
  - JTag: not compatible with TinyOS 1.1.7
  - TOSSIM: poor PHY-layer support
    - *Example: no multi-channel support*
  - "Most effective" debugging interface: 3 LEDs
- Upload code:
  - Wireless code propagation (Deluge):
    - *Periodically broadcast code summary, which interferes with measurements.*
  - Most "reliable" way: manually plug Motes onto the MIB510 programming board
- Hardware failure
  - Need to solder wires from time to time



Strategy validation

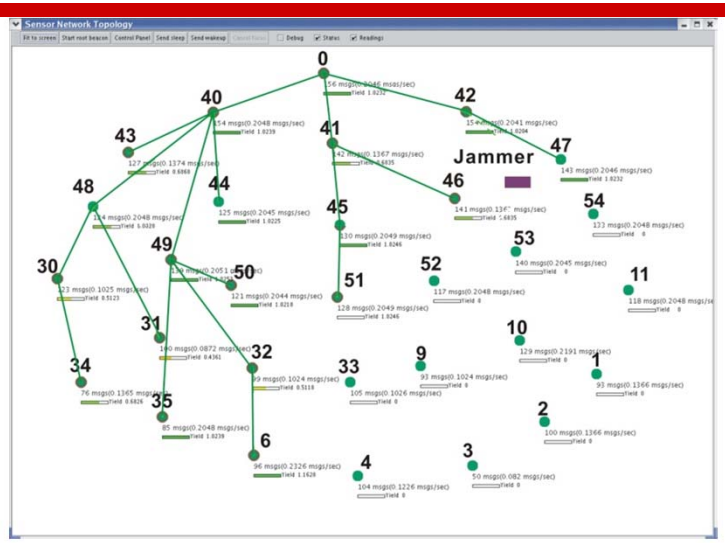
- Testbed
  - 30 Mica2 motes
  - 2.5 feet spacing
  - Tree-based routing
  - Surge
- Performance Metrics:
  - Network recovery
  - Protocol overhead

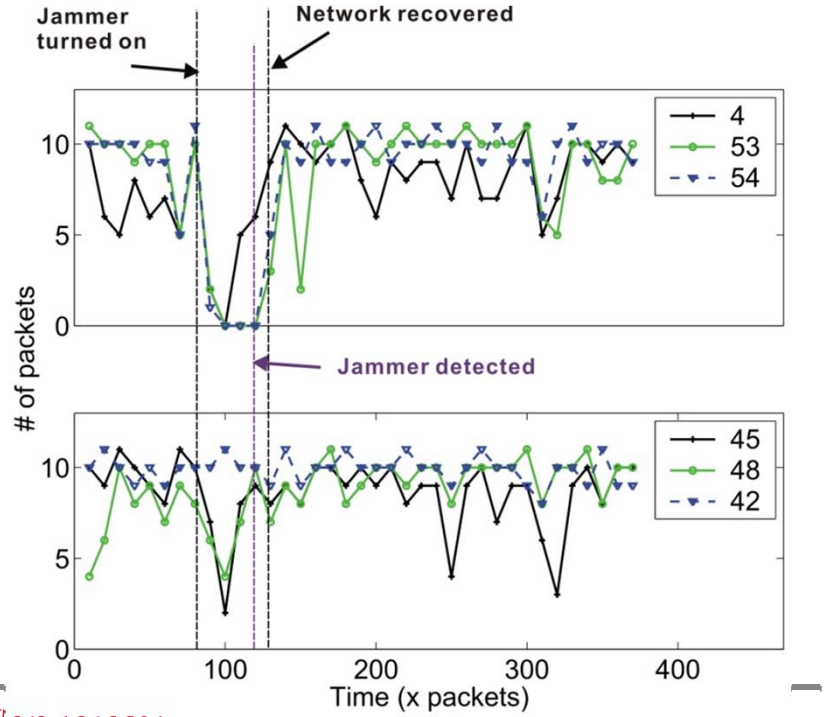


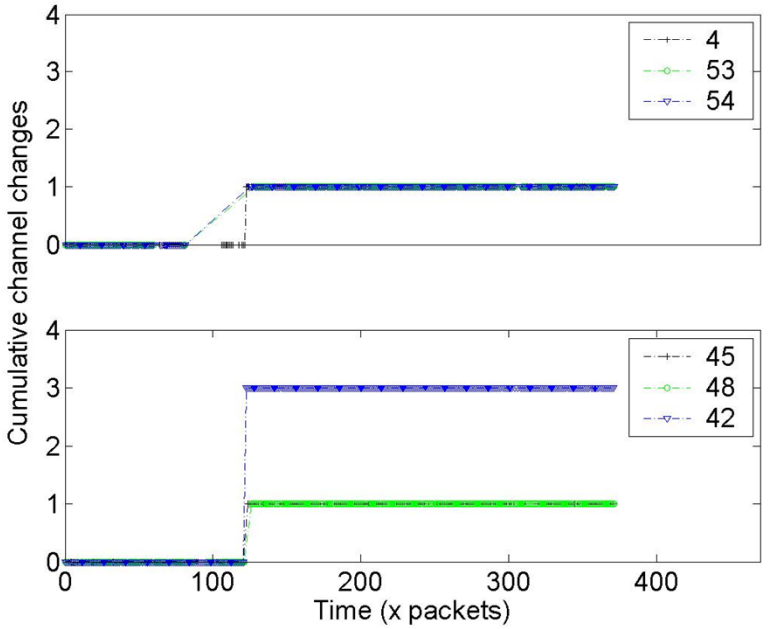


Experimental results

- Performance Metrics:
  - Network recovery
  - Protocol overhead

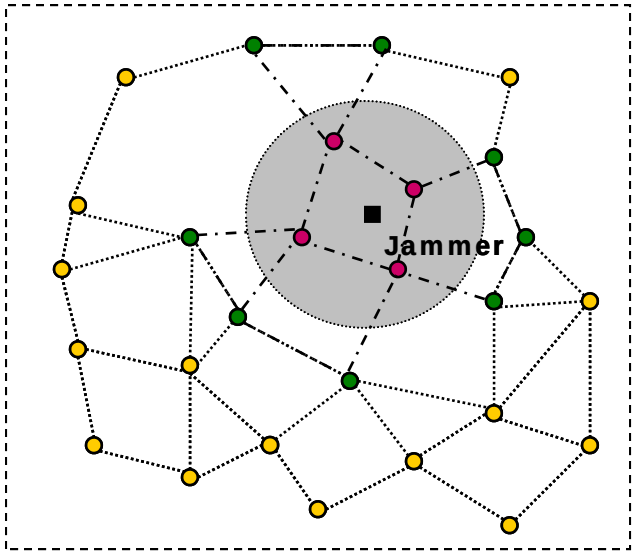






Spectral Multiplexing

- Spectral Multiplexing
  - Jammed nodes switch channel
  - Nodes on the boundary of a jammed region serve as relay nodes between different spectral zones
- Challenge
  - Sender-receiver frequency mis-matching
  - Synchronization
  - Initiation
  - Slot duration
- Algorithms
  - Synchronous Spectral Multiplexing
  - Asynchronous Spectral Multiplexing



Node working in channel 1

Node working in channel 2

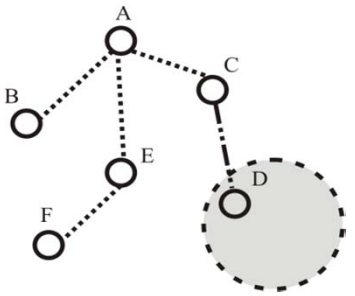
Node working in both channel 1 & 2

channel 1

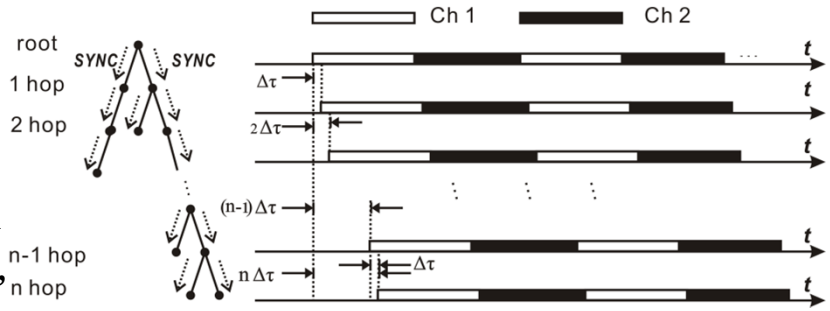
channel 2

Synchronous Spectral Multiplexing

- Idea:
  - One global clock, divided into slots
  - Each slot is assigned to a single channel. The network may only use the assigned channel – regardless of whether nodes are jammed.
- Challenges:
  - How to synchronize the global time efficiently when nodes may work in different channels?
  - Initiation
  - Slot duration
- Solution:
  - The root sends out SYNC to its children, and the children send out SYNC to their children, and so on …
  - Boundary nodes send SYNC in rapid succession across both channels.

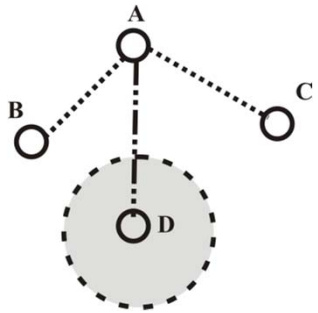


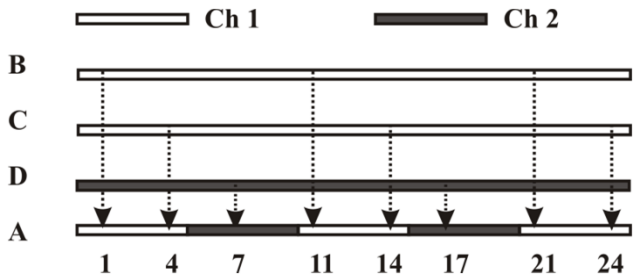
slot 1	A, B, C, E, F
slot 2	C, D



Asynchronous Spectral Multiplexing

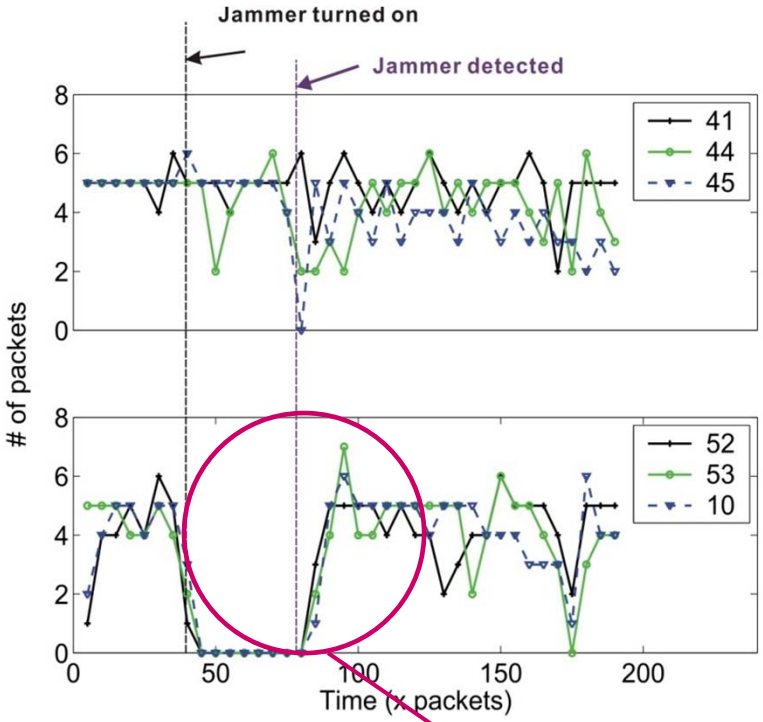
- Idea:
  - Nodes operate on local schedules. The boundary nodes make local decisions on when to switch channel
- Challenges:
  - How to coordinate the schedules among neighbors?
  - How long a node should stay on each channel?
  - Initiation
  - Slot duration
- Solution:
  - The boundary node notifies its children its change of channel
  - Stay in each channel long enough to offset the switching overhead, short enough to avoid buffer overflow.



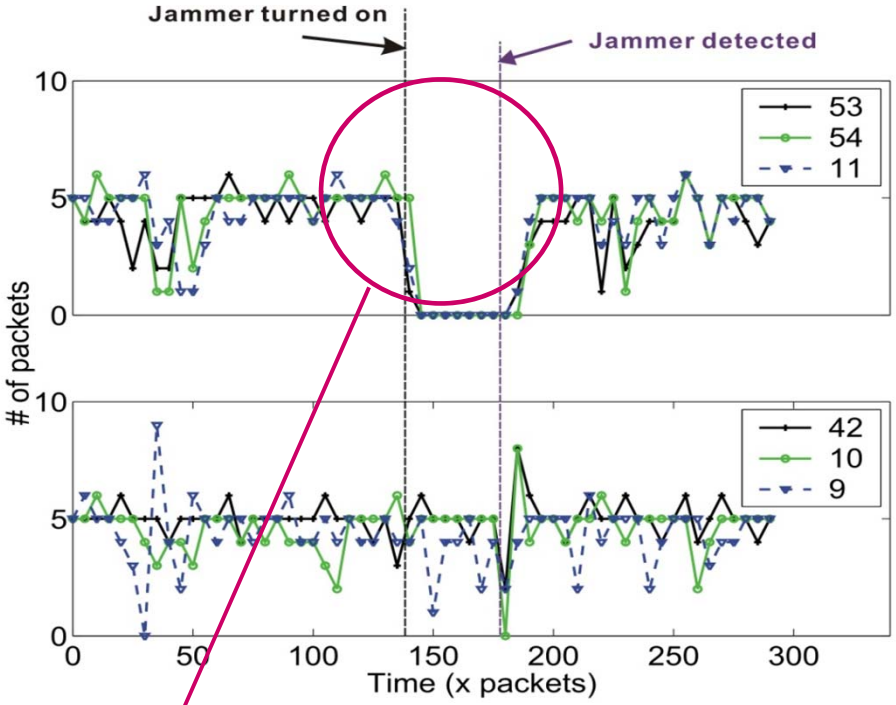


Experimental results:

Synchronous
Spectrum Multiplexing



Asynchronous
Spectrum Multiplexing



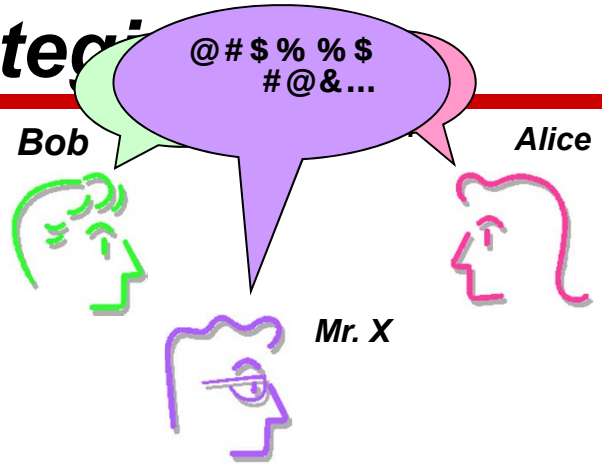
Down time due
to jamming

Channel Surfing Algorithm

- Coordinated Channel Surfing
  - Pros:
    - *Simple*
  - Cons:
    - *Even if a small portion of the network is jammed, the whole network has to pay for the price of channel surfing.*
      - Synchronous Spectral Multiplexing
  - Pros:
    - *The deterministic and synchronous nature of this algorithm guarantees that it can work well even under complex scenarios where multiple nodes need to work on multiple channels and these nodes are neighbors of each other.*
  - Cons:
    - *Extra overhead to maintain synchrony among nodes*
- Asynchronous Spectral Multiplexing
  - Pros:
    - *Small synchronization overheard when jammed region is small*
    - *Able to adapt to local traffic and buffer conditions*
  - Cons:
    - *Complicated, advantage less pronounced when jammed region is large.*

	Coordinated Channel Surfing	Spectral Multiplexing	
		Synchronous	Asynchronous
ROM usage (bytes)	28186	32634	30070
RAM usage (bytes)	3511	3557	3495

Other Defense Strategi



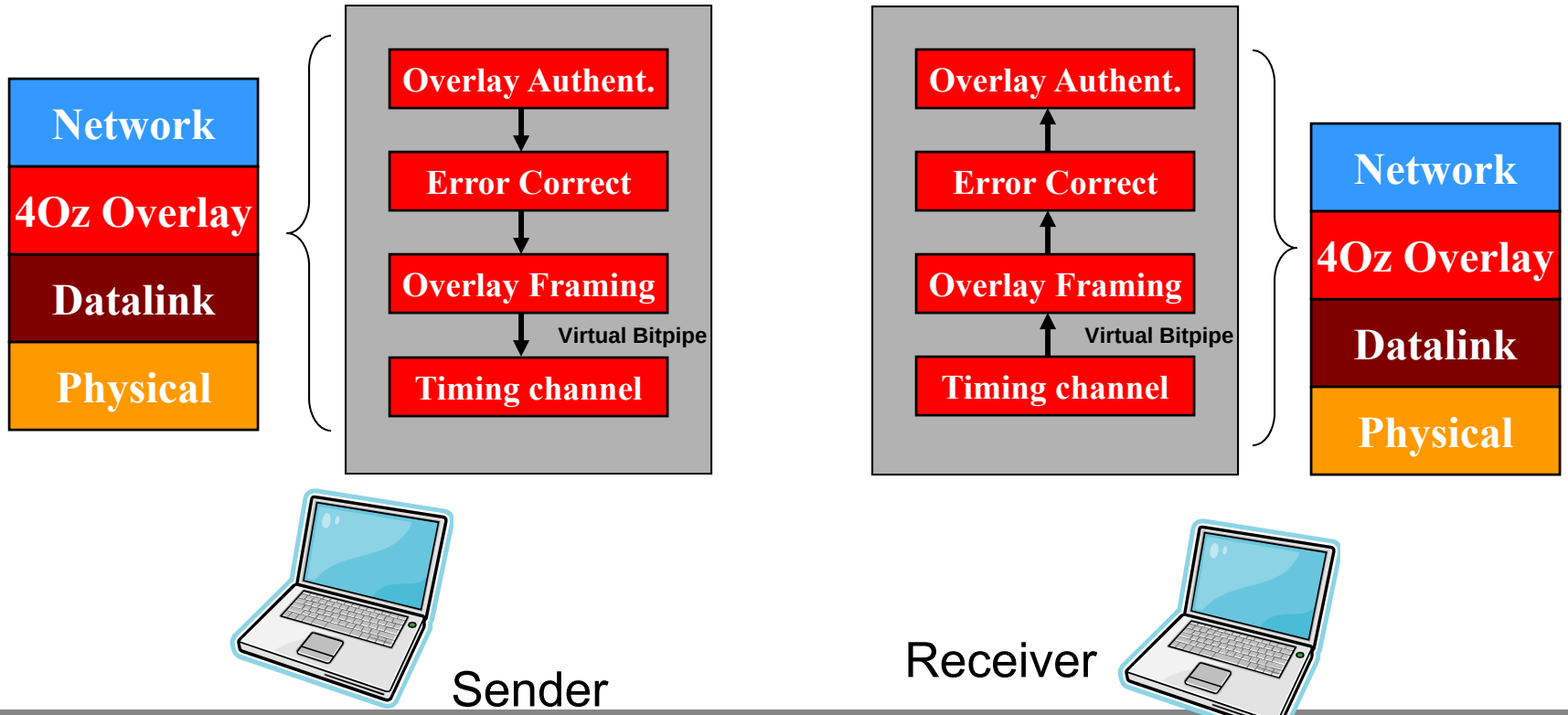
- *Goal:*
  - Convey information between Bob and Alice in the presence of Mr. X, the interferer.
  - Using existing wireless platforms (CSMA)

- Possible Strategies:
  - Channel Surfing -> interference-free channels available
  - Spatial retreat -> mobile wireless nodes
  - Power control -> increase transmission power

- What can you do if:
  - No interference-free channels are available.
  - Mobility is not an option.
  - You cannot over-power the jammer
  - You have a *short emergency* packet to send

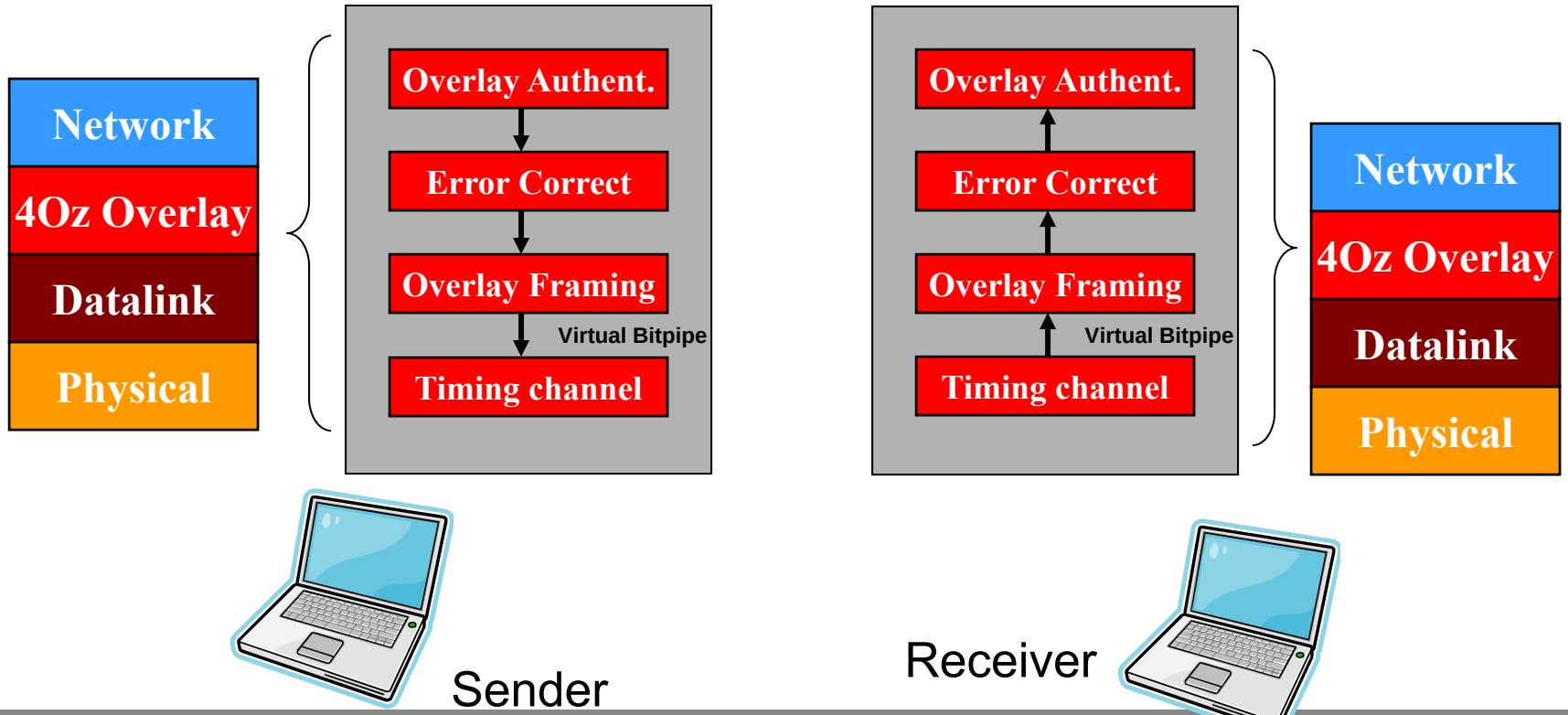
Timing Channel for Anti-Jamming

- Idea:
  - Alice and Bob: Alice did not know exactly what Bob was saying, but she knew that Bob said something by looking at *his lip movements*.
  - Wireless network: exploit the fact that there was an attempted, incoming packet to convey information.

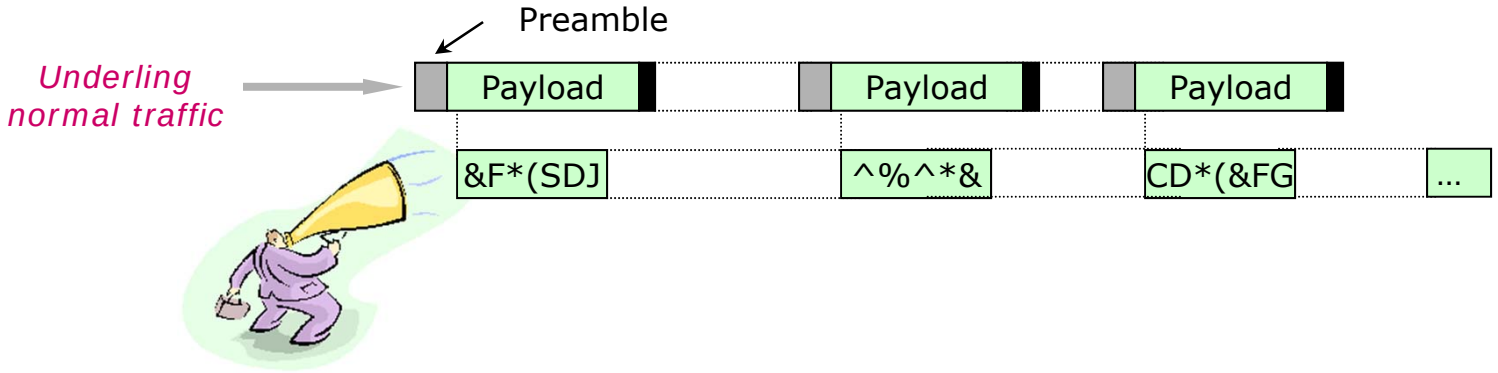


Timing Channels for Anti-jamming

- Sub-problems:
  - Can you detect a packet in spite of the radio interference?
  - Can you modulate the event of incoming packet?
  - Can you implement such a strategy in a real system?



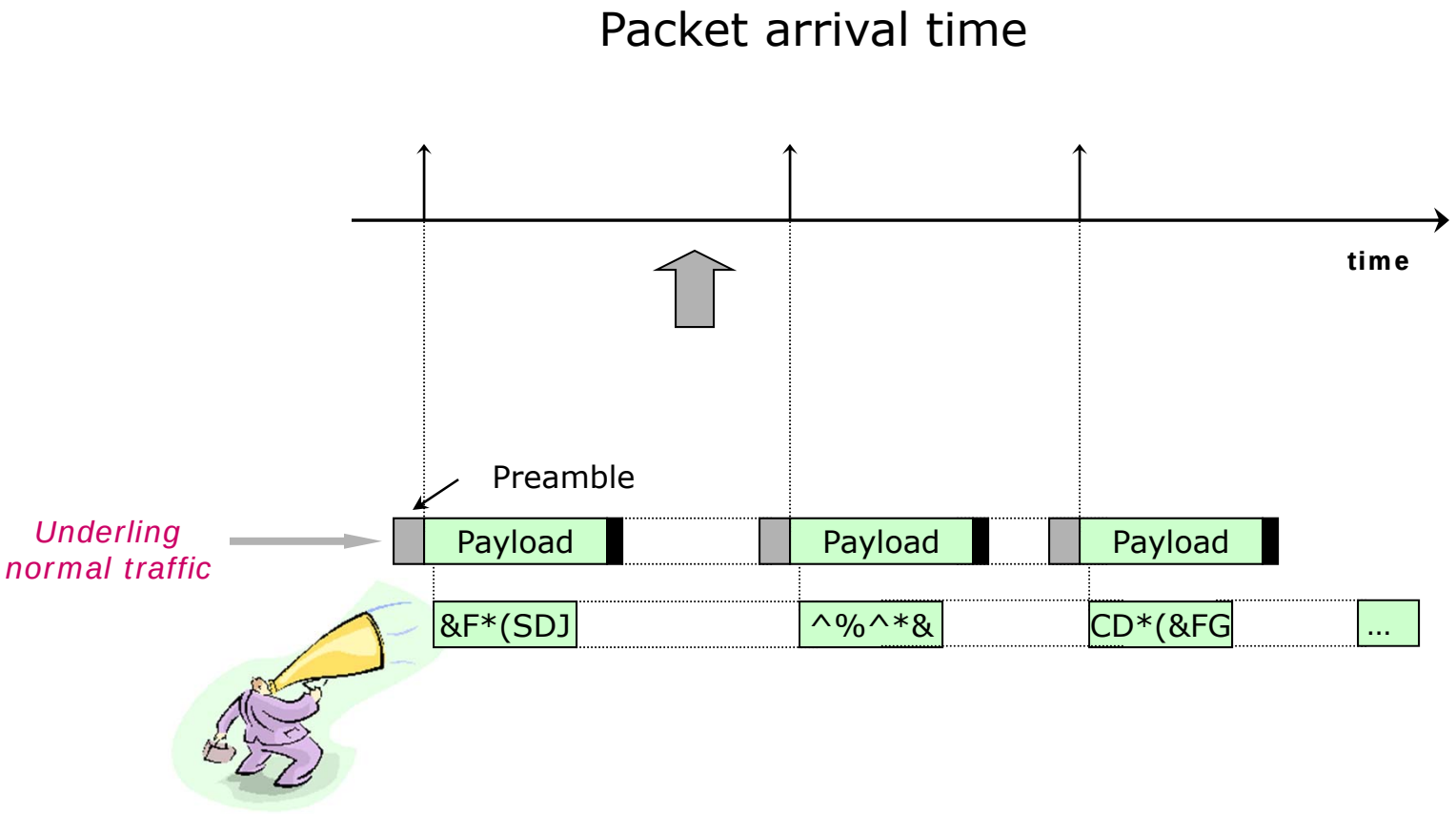
Malicious Jammer



- Reactive jammer:
  - Stays quiet when the channel is idle, but starts jamming as soon as it senses activity on the channel.

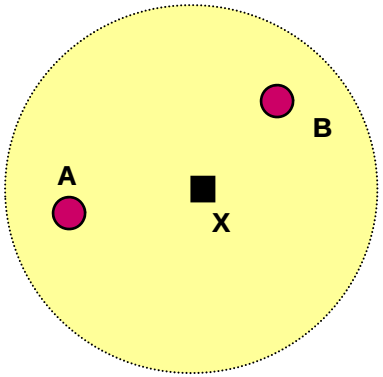
- Observation
  - CANNOT decode packets correctly
  - Preamble CAN be detected correctly by the receiver

Packet Arrival Times

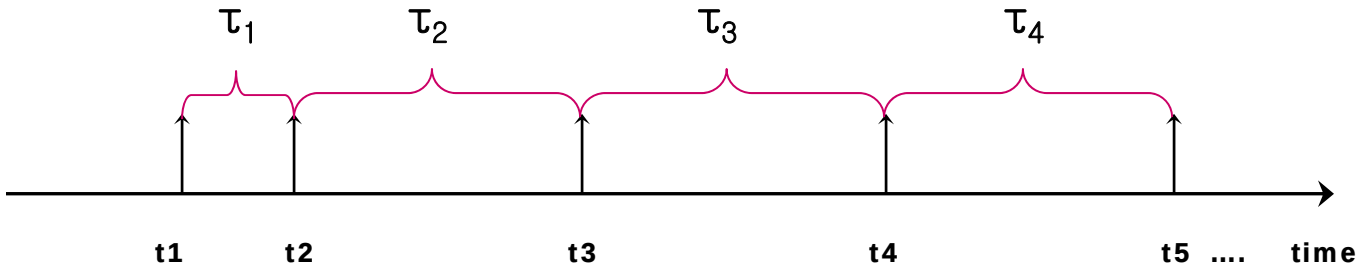


Conveying Information in Timing: A Two Party Prototype

- Two party scenario
  - A: Sender
  - B: Receiver
  - X: Interferer

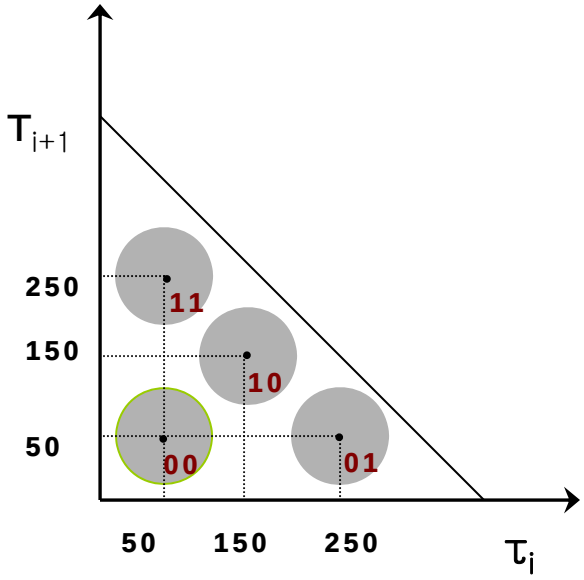


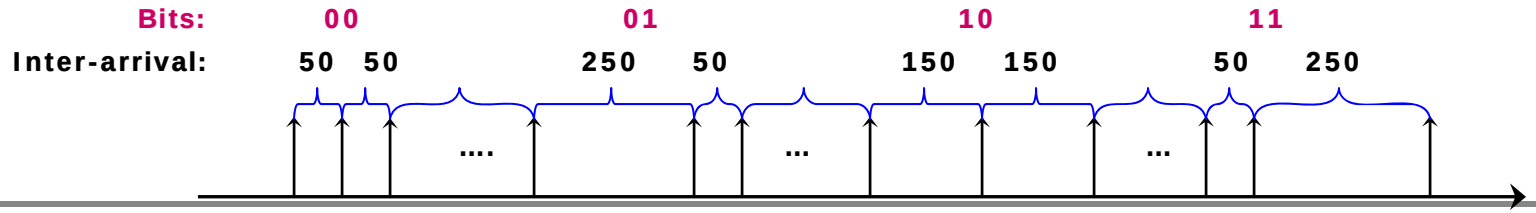
- What B observes:
  - t_i: the arrival time of the i^{th} packet
  - τ_i: the inter-arrival time, $\tau_i = t_{i+1} - t_i$



Example of Inter-arrival Time Coding

- Use (τ_i, τ_{i+1}) to code 2 bits
  - Triangular simplex
  - Each circle represents 2 bits
    - *(50,50) -> 00*
    - *(250,50) -> 01*
    - *(150,150) -> 10*
    - *(50,250) -> 11*
  - Sender *encodes*:
    - *00->(50,50)*
    - *Send pkts at time 0, 50 & 100*
  - Receiver *decodes* (nearest neighbor):
    - *Receive packets at 0, 50 and 100*
    - *Get inter-arrival time pair (τ_i, τ_{i+1})=(50,50)*
    - *(τ_i, τ_{i+1}) -> Calculate the Euclidean distance*





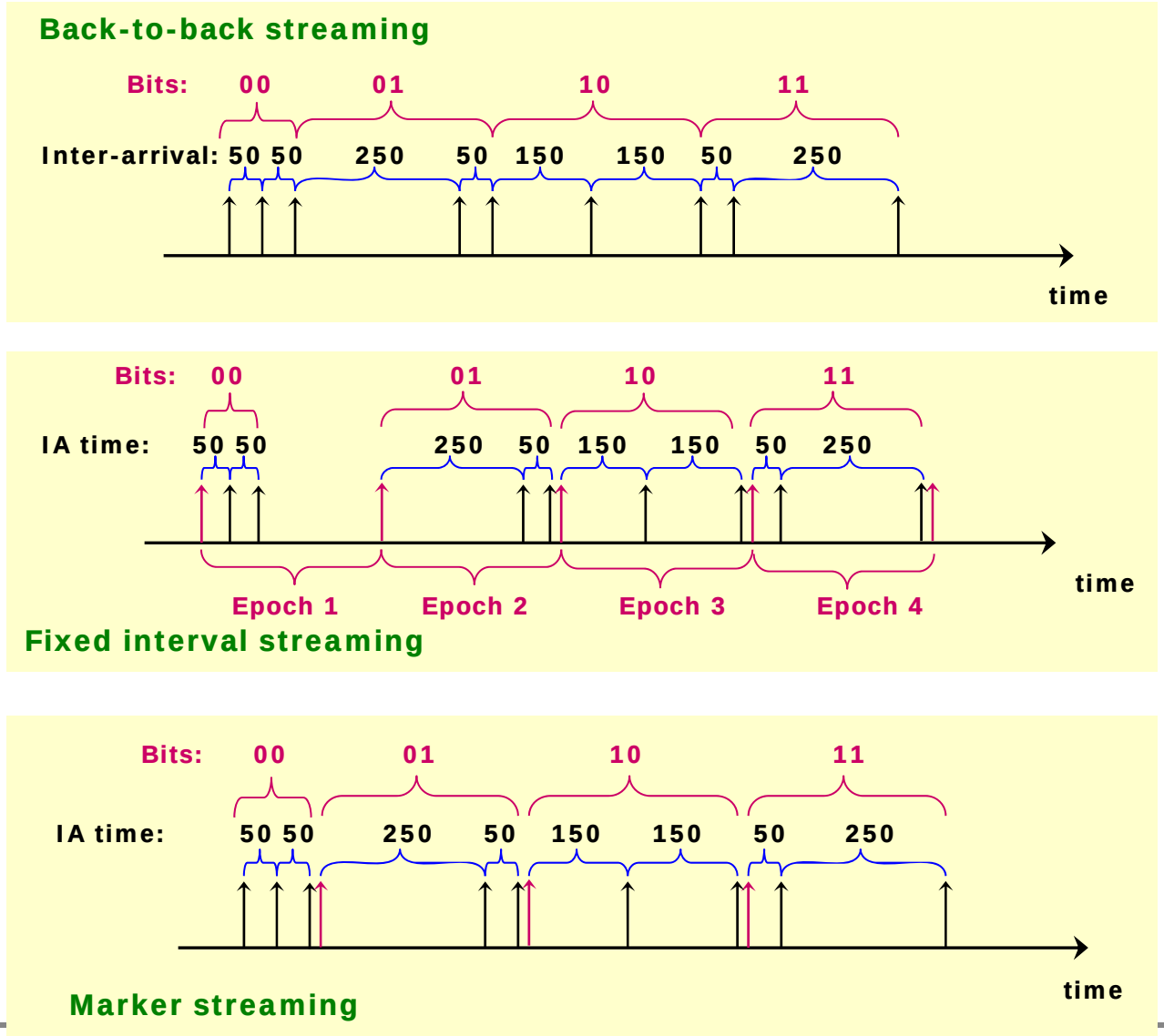
Experimental Validation

- Mica2 Motes
  - *8-bit CPU at 4MHz,*
  - *128KB flash, 4KB RAM*
  - *916.7MHz radio*
  - *OS: TinyOS*

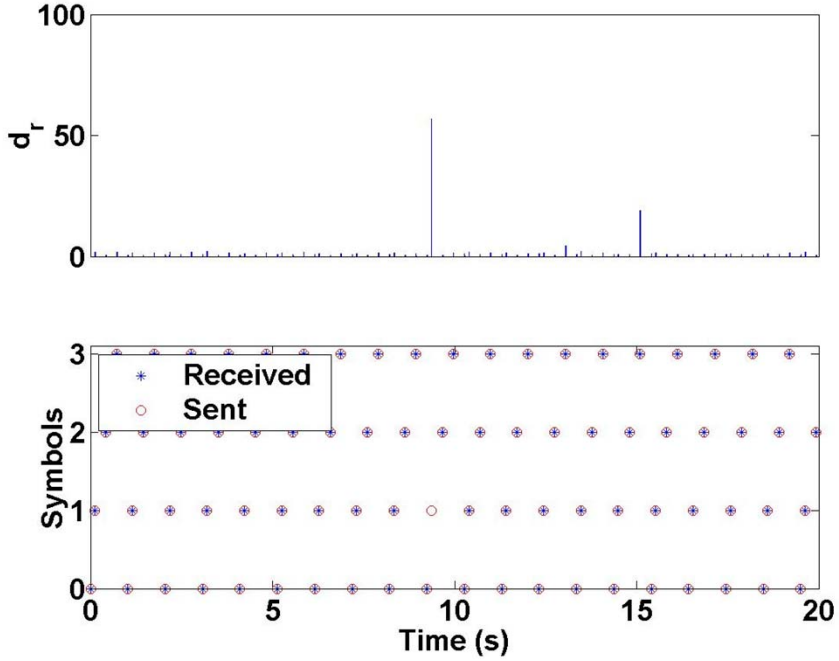
- Three nodes:
  - Sender:
    - *Send 00, 01, 10, 11, 00…*
  - Receiver
  - Interferer: Reactive jammer

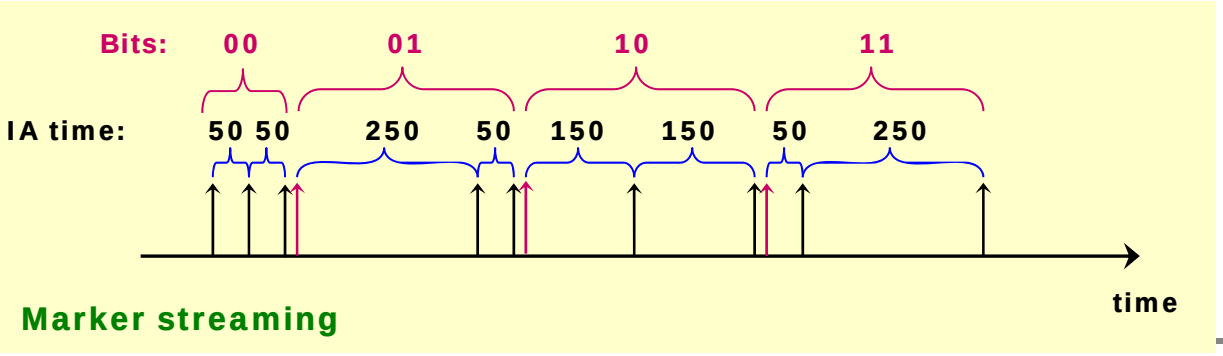
- *Challenges:*
  - Jitter
    - disable carrier sensing and back-off
    - code design
  - Detect the presence of a packet
  - Clock skew
  - *How to connect consecutive codewords together?*

Symbol streaming strategies



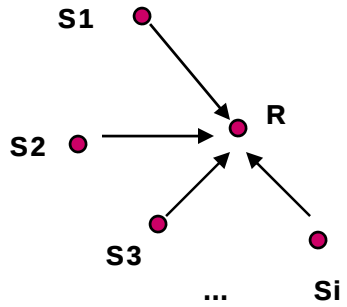
Symbol streaming strategies

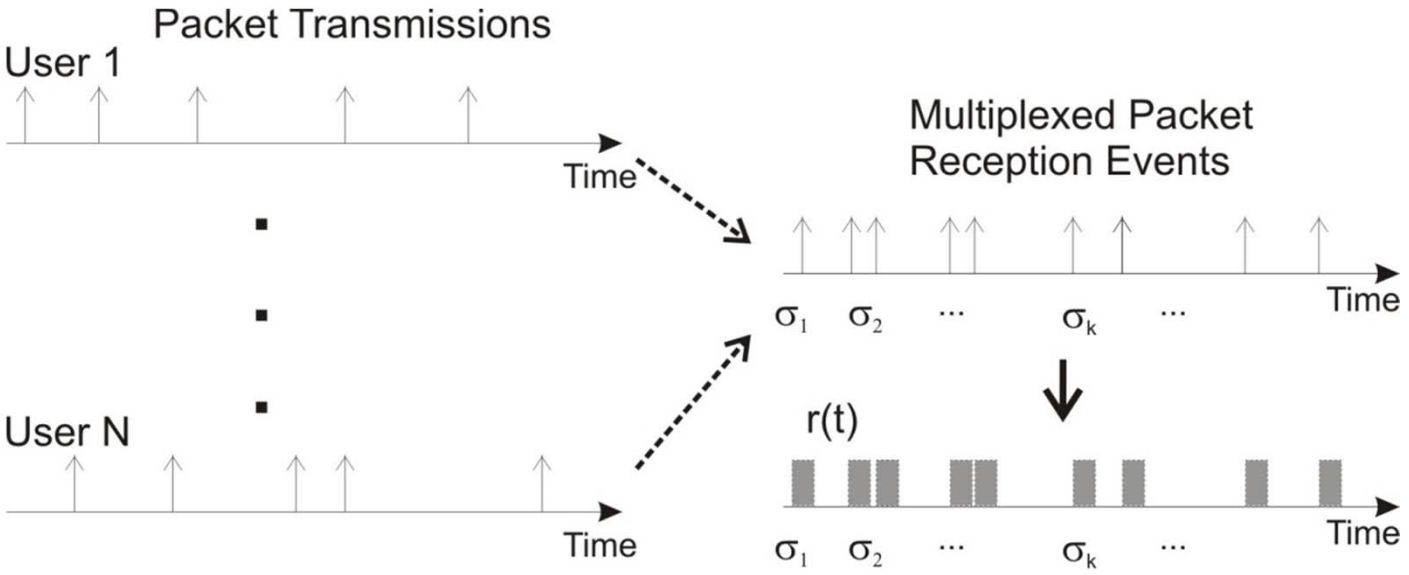




Multiple Senders

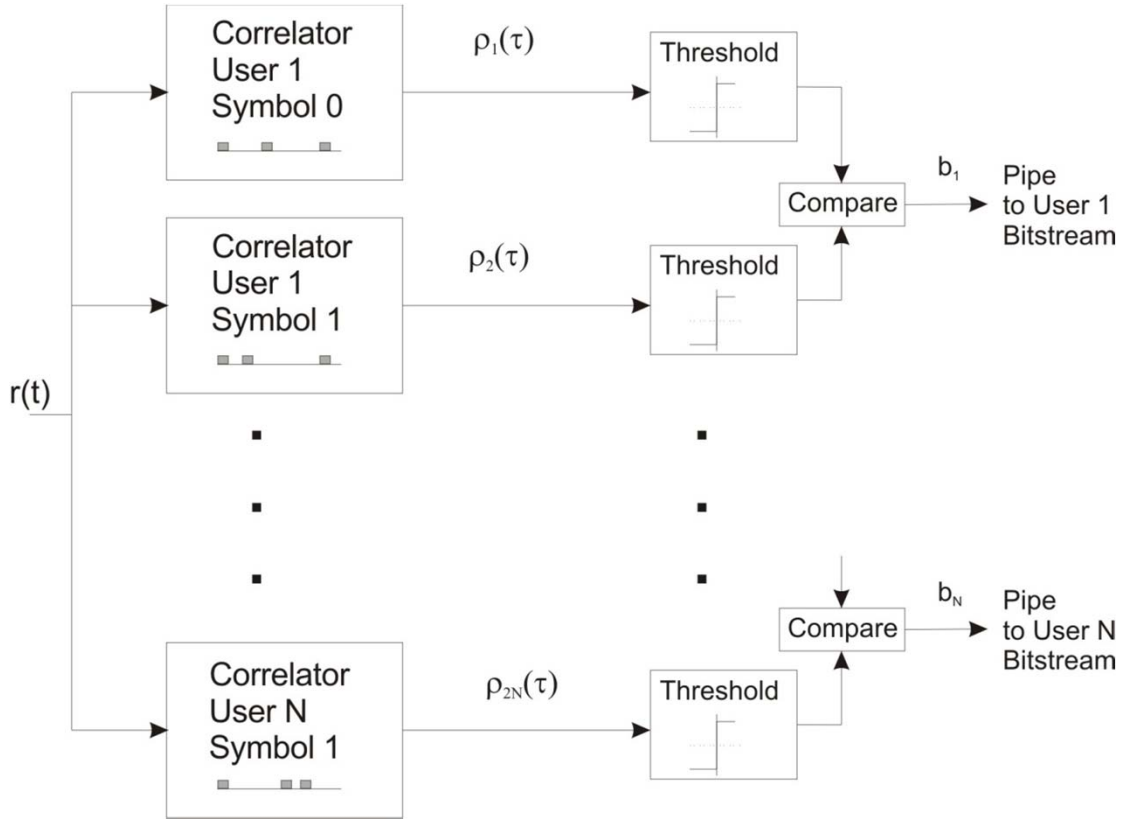
- Challenges:
  - Packets from various receivers interleave with each other
  - Extract individual sender's communications from the mixed packet arrivals at the reciever





Multiple Senders

- Idea: design transmission sequences based on optical orthogonal codes (OOC)
  - The cross-correlation between different code words is very small
  - The auto-correlation is high



Where to go from here?
(aka, ideas for the next army of graduate students)

We need a holistic approach to addressing security issues in emerging wireless systems

Confidentiality	**Wireless is easy to sniff. We still need encryption services and key management. Key freshness is an issue.**
Integrity	**Wireless hardware/equipment need to be safe from modification. Data/control info should not be modified before or during transit.**
Forensics	**Wireless networks will be the platform of choice for attacks. Should the network keep track of forensic evidence?**
Privacy	**Perpetual connectivity can mean constant surveillance! With snooping one can monitor mobility and handoffs between networks.**
Location	**Location is a new form of information provided by wireless systems that will facilitate new services. Location information needs to be trusted.**
Intrusion	**The pervasiveness of the wireless networks should not mean that just anyone can participate! Example: Rogue APs**
Availability	**The value of a wireless network is its promise of ubiquitous connectivity. Unfortunately, wireless networks are easy to "break" (e.g. jam, denial of service)**
Non-repudiation	**RF energy radiates, and wireless entities within the radio coverage pattern may serve as witnesses for the actions of the transmitter.**