Module: Jamming

Panos Papadimitratos

2015-11-05

Contents

1 Introduction (slides 1–12)

Consider communications that share a wireless medium, as illustrated in Slide 1 for device A, B and C. Any transmissions that overlap in time and frequency (i.e., over the same channel) can result in no useful reception. This, termed frequently a collision, is handled by medium access control layer protocols (as discussed in the intro/revision lectures). Moreover, transmissions over adjacent channels can also affect the ability to decode correctly at the receiver. There can unintentional interference, transmissions by other devices belonging to other systems, as the example of the micorwave oven shows.

The usable frequencies available are limited. E.g. the Industrial, scientific and medical (ISM)[1] are crowded: all sorts of applications, from microwave ovens to Wireless Local Area Networks and Personal Area Networks (e.g., Bluetooth). The International Telecommunications Union (ITU) states that these *"Radiocommunication services operating within these bands must accept harmful interference, which may be caused by these applications."*[2] More on those techniques later in Section 2.

The picture on slide 9[3], with the United States frequency allocation, illustrates this point: the airwaves are crowded. Being a resource in shortage implies that access is regulated, allowing access to and use of the spectrum only by the authorized users and/or its owner(s). Nonetheless, policy alone cannot be effective at all times.

Ignoring interference can be a costly affair, see e.g. the IEEE Spectrum article[4] about LightSquared.

Of course, desingers of communication systems have always accounted for noise and interference and designed their systems to withstand such impairments, along with those of the signal propagation environment. One way to handle transmission errors is to introduce some redundancy, to detect and possibly detect and correct some limited number of bits in error. This is discussed in the next few slides.

The challenge lies in that deliberate interference (jamming) can strong(er) and prevent the correction of any transmission at the receiver, or even the reception of anything meaningful.

1.1 ECC

This section contains an overview of ECC. For another introduction and further pointers, see e.g. the Wikipedia page[5]. For those who seek a a rigorous and mathematical description, Lin and Costello's book [1] is a comprehensive textbook, covering material well beyond the scope of this course.

1.1.1 Basic schemes

One way to increase resilience to interference, or transmission errors, is to use Error Control Coding (ECC). The naive way is to use a repetition code, i.e., to send the same symbol multiple times. If you were to send `1010`, you now send `1010 1010 1010`. This, of course, can be a waste of bandwidth.

The next step, illustrated on slide 12, is to use a parity bit. This allows the receiver to detect one bit in error per block of data. To put it simply, within a block of bits the transmitter counts the number of ones the block contains. If this is even, the parity bit is set to `0`, otherwise `1`. E.g. `1010` will become `1010`$\underline{\texttt{0}}$. This is usually implemented as an XOR ($\otimes$) over all bits. If *at most* one bit is flipped during the transmission, the receiver counts can tell that something went wrong and request a retransmission.

[1] http://www.itu.int/ITU-R/terrestrial/faq/index.html#g013

[2] Ibid.

[3] Available at http://www.ntia.doc.gov/files/ntia/publications/spectrum_wall_chart_aug2011.pdf

[4] http://spectrum.ieee.org/tech-talk/telecom/wireless/final-blow-for-lightsquared

[5] http://en.wikipedia.org/wiki/Error_detection_and_correction

1.1.2 CRC

By seeing messages as polynomials in GF(2) (a finite field[6] with two elements), a Cyclic Redundancy Check (CRC) can be made on messages of arbitrary length. Let the message be called $m(x)$, and take a so called generator polynomial $g(x)$ of degree n. An example could be the CRC-CCITT[7] $g_{\text{CCITT}}(x) = x^{16} + x^{12} + x^5 + 1$ used by Bluetooth, or $g(x) = x + 1$ which is the parity bit scheme described before. Then using the division algorithm,

$$m(x)x^n = q(x)g(x) + r(x) \tag{1}$$

we can obtain a $r(x)$ with degree lower than n. This is your CRC value, which should be sent with the message. E.g., with the message `1010`, code it as $m(x) = x^3 + x$, and use the parity bit polynomial $g(x) = x + 1$. In the field of real numbers we would have

$$(x^3 + x)x = (x^4 + x^2) = (x^3 - x^2 + x - 2)(x + 1) + 2 \tag{2}$$

but in GF(2) we take everything modulo 2,

$$x^4 + x^2 = (x^3 + x^2 + x)(x + 1) + 0 \tag{3}$$

giving us $r(x) = 0$, just as before, $\texttt{1010}\underline{\texttt{0}}$.

1.1.3 Hamming codes

There are codes that can not only detect errors, but also repair them. One example is Hamming codes. They are prefect codes, in the sense that they achieve maximum rate for the given correcting ability, here 1 error.

For any positive integer $m \geq 3$, there is a Hamming code with code length $n = 2^m - 1$, with $k = 2^m - m - 1$ information symbols (and thus $n - k = m$ parity check symbols) that can correctly correct 1 error. Such a code is called a Hamming(n,k) code.

Slide 15 gives the example of $m = 3$, the Hamming(7,4) code, which expands every four information bits to seven bit code words. Please refer to the illustrations the lecture slide provides[8]. Call the information bits d_1, d_2, d_3, d_4 and the parity bits p_1, p_2, p_3. Hamming(7,4) can described as

$$\begin{aligned} p_1 &= d_1 \otimes d_2 \otimes d_4 \\ p_2 &= d_1 \otimes d_3 \otimes d_4 \\ p_3 &= d_2 \otimes d_3 \otimes d_4 \end{aligned} \tag{4}$$

With the message `1010`, say, $d_1 = 1$, $d_2 = 0$, $d_3 = 1$, $d_4 = 0$, we have

$$\begin{aligned} p_1 &= 1 \otimes 0 \otimes 0 = 1 \\ p_2 &= 1 \otimes 1 \otimes 0 = 0 \\ p_3 &= 0 \otimes 1 \otimes 0 = 1 \end{aligned}$$

and our sent message is $\texttt{1010}\underline{\texttt{101}}$.

Say now, that under the transmission, d_3 gets flipped, so $\texttt{1000}\underline{\texttt{101}}$. The receiver does the same parity calculations,

$$\begin{aligned} p_1 &= 1 \otimes 0 \otimes 0 = 1 \\ p_2 &= 1 \otimes 0 \otimes 0 = 1 \\ p_3 &= 0 \otimes 0 \otimes 0 = 0 \end{aligned}$$

[6] `http://mathworld.wolfram.com/FiniteField.html`

[7] International Telecommunication Union (ITU)

[8] Or see them at `http://en.wikipedia.org/wiki/Hamming(7,4)`

and sees that p_2 and p_3 is wrong. Checking (4), we see that the information bit involved in those two parity calculations (but not in p_1) is d_3, and we can correct this. The scheme works equally well if one of the parity bits is flipped, say p_1 (so we receive 1010001),

$$\begin{aligned}p_1 &= 1 \otimes 0 \otimes 0 = 1\\ p_2 &= 1 \otimes 1 \otimes 0 = 0\\ p_3 &= 0 \otimes 1 \otimes 0 = 1\end{aligned}$$

and we have a miss-match on only p_1, and can therefore flip it back.

If two bits are flipped, say d_3 and d_4,

$$\begin{aligned}p_1 &= 1 \otimes 0 \otimes 1 = 0\\ p_2 &= 1 \otimes 0 \otimes 1 = 0\\ p_3 &= 0 \otimes 0 \otimes 1 = 1\end{aligned}$$

we can now see that p_1 is erroneous, but this error pattern will be corrected to the wrong code word (by assuming that p_1 is the only error) since we can no longer determine which bit is the source of the error. With a Hamming distance of 3, there is always a closest valid code word.

It is possible to extend this to a code able to detect 2 errors, by extending the code word with an overall parity bit. This scheme will have a Hamming distance of 4, making room for code worlds exactly between two valid code words.

1.1.4 Erasure codes

Erasure codes is a set of codes to handle dropped packets, most famously implemented as Reed–Solomon (RS) codes. RS codes are used to protect a vast set of data types[9], e.g., CD, DVD and Blue-Ray. If you scratch your CD, thank Reed and Solomon that you can still play it. Or if your QR code gets smudge (or someone placed a logo over it). Some RAID6 schemes use RS codes too.

2 Jamming and anti-jamming actions/defenses (slides 10 - 40)

We are concerned with adversaries that deliberately disrupt communications, beyond the point communication and networking protocols can cope with. Communication systems can cope with symbol losses through error correction codes, resilient modulation, and perhaps increased transmission power. Nonetheless, none of these can really address jamming - deliberate interference. Moreover, at upper layers it is likely to remedy the case of a lost/jammed packet by requesting a retransmission. But again if the communication link is jammed, the re-transmission will not be successful either.

Jamming essentially lowers your opponent's *Signal to Noise Ratio (SNR)*, so that communication is no longer possible. This is done by adding noise within the signal spectrum. Jamming is part of the broad set of *Denial of Service (DoS)* attacks. It was to a certain extent a military affair. However, there is significant relevance to civilian applications. Law enforcement uses include mobile phone jamming (e.g. prisons in Sweden). Moreover, jamming of *Global Positioning System (GPS)* signals - see material of the *Advanced Networked Systems Security (ANSS)* course.

[9] http://www.eccpage.com/reed_solomon_codes.html is a nice read about the impact of RS codes

It is possible, and in some cases relatively easy to locate jammers. This may make jamming a powerful yet a short-lived attack. There are however cases when outages caused by jamming (unintentional) can take day(s) to resolve.

The adversary can jam in different ways, perhaps not all the time (see slide 16). It may be that the jammer affects all available channels. But, in principle, it is reasonable to assume that the strength of the jammer is limited. In other words, it can only jam a certain part frequency spectrum with a certain signal transmission power. There are different ways to modulate and code signals to be more resilient to interference, from jammers or not.

The essence is that if the adversary overwhelms a channel, then legitimate users/devices should switch to a jammer-free channel. This is an option for many of the popular nowadays technologies. The challenge is to make it hard for the jammer to follow or predict where to jam. If this is eventually impossible, i.e., the jammer is too powerful, then the solution is eventually to localize the jammer and remove the jamming devices (sources of offending signals) physically.

2.1 Antennas (slide 19)

The picture is from Aerospace[10]. The main point here is to use beamforming to improve antenna gain in the direction of the sender; and try to filter out signals in the direction of the jammer.

2.2 DSSS (slides 20–22)

Direct-sequence spread spectrum, used by 802.11b, uses a pseudo-random sequence to modulate the signal to a wider spectrum, increasing the energy requirements of a jammer. The slides illustrates how this works, with pictures from [2] and [3]. Furthermore, DSSS signals are harder to detect, as they are closer to the background noise in the air.

2.2.1 OFDM (slide 23)

The Orthogonal frequency-division multiplexing (OFDM) used by, among others, IEEE 802.11a, g, n and 4G. With orthogonal signals, several senders can coexist on the same channel, achieving an efficient use of frequency spectrum. IEEE has a tutorial[11] on OFDM, where the slide picture is taken from. It goes through OFDM in detail.

2.3 Frequency hopping (slides 24–29)

Frequency-hopping spread spectrum makes it harder to intercept or jam a signal by chaining frequency/channel in a pseudo-random way. Used in Bluetooth and is very common in military radio. The illustration on slide 25 highlights this, that a given frequency is only used for a certain time. This eludes a jammer, which probably only can jam a few of the channels available.

Generation of a hopping sequence: based on a secret. Recall the analogy of a one-time-pad and the availability of a predefined known sequence. Moreover, consider how easy/hard it is for the adversary to guess a hopping sequence.

[10] http://www.aero.org/publications/crosslink/summer2002/06.html

[11] http://www.ieee802.org/22/Meeting_documents/2005_Jan/22-05-0005-00-0000_OFDMA_Tutorial_IEEE802-22_Jan%2005.ppt

2.3.1 Bluetooth (slide 28)

Bluetooth uses frequency hopping across 79 communication channels, in a way only known to the receiver[12]. But since it is designed to resist WLAN heavy environments, it uses Adaptive Frequency Hopping (AFH)[13], to avoid channels with high levels of interference.

> ***Q:*** *Will Bluetooth and Wireless LAN (WLAN) interfere with each other?*
> ***A:*** *No, both Bluetooth and WLAN can co-exist. Since Bluetooth devices use Frequency Hopping and most WLANs use Direct Sequence Spreading techniques they each appear as background noise to the other and should not cause any perceivable performance issues."* [14]

Point of caution: implementations can be tricky - there are machines (laptops) from popular manufacturers that get their WiFi on the 2GHz area severely impaired when BlueTooth is on. The problem disappears when WiFi operates in the 5GHz area.

2.3.2 What if there is no shared information? (slide 29)

Then, nodes communicate by randomly choosing channels and exchange data only when they hop to the same channel. This would result is very low throughput. Thus, it would be useful e.g. to bootstrap a hopping sequence; i.e., for limited use. Another approach would be use a rendez-vous scheme, with nodes follows their peers. The downside would be loss of communication or in a sense synchronization. One can combine this with some pre-agreed (possible to generate) sequence if needed, returning to previous, pre-shared-based mode of operation.

3 Localization (slides 30–37)

One efficient way to defend against jamming is to locate the jammer and physically remove it.

3.1 Time Difference of Arrival (TDOA)(slides 33–36)

The TDOA method relies on the finite propagation speed of the measured signals, resulting in different times of arrival at different receiver locations.

Given two sensors and one transmitted signal, the two received signals can be modeled[15] as:

$$x_1(t) = s(t) \quad \text{and} \quad x_2(t) = s(t + \Delta), \tag{5}$$

with the the time delay Δ. From two real-valued signals, the cross correlation function can be calculated as the expected value

$$r(\tau) = E[x_1(t)x_2(t + \tau)] \tag{6}$$

to find the time delay estimator as the τ that maximizes $r(\tau)$.

This estimator will give us hyperbola of possible locations of the sender, as seen in slide 35. It is not sufficient using only two receivers to locate a signal source on a two dimensional plane using TDOA, we need at least three. To illustrate this, we add

[12] `http://www.althos.com/tutorial/Bluetooth-tutorial-frequency-hopping-FHSS.html`

[13] Ibid, next page.

[14] `http://www.mobileinfo.com/Bluetooth/FAQ.htm#t4`

[15] For simplicity, we here ignore noise and attenuation

another receiver on slide 36. This setup is very inefficient and unlikely (and gives us a false mirror point), so the setup at slide 37 is much more apt. Here we have a very good possibility of locating the sender. Slide 38 illustrates a steered response power over the same setup, where the sender is most likely located where we have a local maximum.

3.2 Frequency Difference of Arrival (FDOA)(slide 37)

In addition to TDOA, a localization system can use FDOA (sometimes called differential Doppler) to better track fast-moving senders. If there is no relative speed between the sender and the transmitters, FDOA does not work. But if your localization system moves, say with aircrafts, you can track stationary senders.

4 Impact and physical layer attacks (slides 39–48)

The purpose here is to show that the physical layer matters. A jammer can disrupt links and control our environment. In fact, jamming, especially when it intelligently applied, can be the stepping stone for more sophisticated, in some cases, or upper layer attacks. Recall and connect to network-level attacks (in upcoming modules): jamming is very effective or even devastating, yet relatively localized in nature (and up to a point in impact).

4.1 IMSI catcher and SSID overtake (slides 42–43)

The International Mobile Subscriber Identity (IMSI) is the way GSM mobiles are identified in the network. By adding a fake base station (the IMSI catcher) to the network, that is easier to connect to than the legitimate base stations, the IMSI catcher can collect these. Furthermore, such an adversarial node can launch an effective man in the middle attack by suggesting to the mobile to turn encryption off[16]. GSM supports this legacy scheme. Funny enough, Rohde & Schwarz holds a European patent on this scheme, *Method for identifying a mobile phone user or for eavesdropping on outgoing calls* [4]. If you are going to engage in unlawful wire-tapping, will you care about patent infringements?

SSID overpower: misleading mobile hosts to connect to the attackers rogue Access Point. This would not be significant if strong access point authentication is in place.

4.2 Packet injection (slides 45–48)

Not jamming per se, but Goodspeed showed that by including whole packets (header and all) inside the payload of unencrypted wireless packets, other devices on that network might see the payload packets as real packets. This can be even more effective in the presence of (some) jamming since this increases the probability that the first, real, header is missed.

As the slides note, the paper is titled after the 1938 radio incident[17] where a theatrical version of Orson Welles' *War of the Worlds* were broadcast in the USA. Many of those that missed the introduction, clearly stating that the next few hours would be

[16] For details, see the paper about IMSI catchers at `http://www.emsec.rub.de/teaching/seminars/seminar_ss07`

[17] `http://en.wikipedia.org/wiki/The_War_of_the_Worlds_(radio_drama)`

a reading of this book, thought it was a news cast about an ongoing alien invasion. Without much other real-time media, panic ensued.

In much the same way, if a receiver misses the first packet header (due to jamming, interference otherwise) it can falsely assume that the packet embedded in the payload part of the original packet *is* the packet.

5 Connection to other lectures (slides 49–50)

Jamming can block links and cut networks in half, more on this in the lecture about secure routing. In a network with carrier sense multiple access with collision avoidance (CSMA/CA) like IEEE 802.11, a sender request permission by a Request to Send (RTS) message, hopefully waiting for a Clear to Send (CTS) packet. Flooding the network with these RTS messages can severely clog this link. More of these type of attacks in the lecture about Distributed Denial of Service.

References

[1] S. Lin and D. Costello, *Error control coding: fundamentals and applications.* Pearson-Prentice Hall, 2004.

[2] A. Carlson, P. Crilly, and P. Crilly, *Communication Systems.* McGraw-Hill Higher Education, 2009.

[3] R. Poisel, *Modern Communications Jamming Principles and Techniques*, ser. Artech House intelligence and information operations series. Artech House, 2011.

[4] J. Frick, "Method for identifying a mobile phone user or for eavesdropping on outgoing calls," Patent EP1 051 053, July, 2003. [Online]. Available: http://www.freepatentsonline.com/EP1051053B1.html