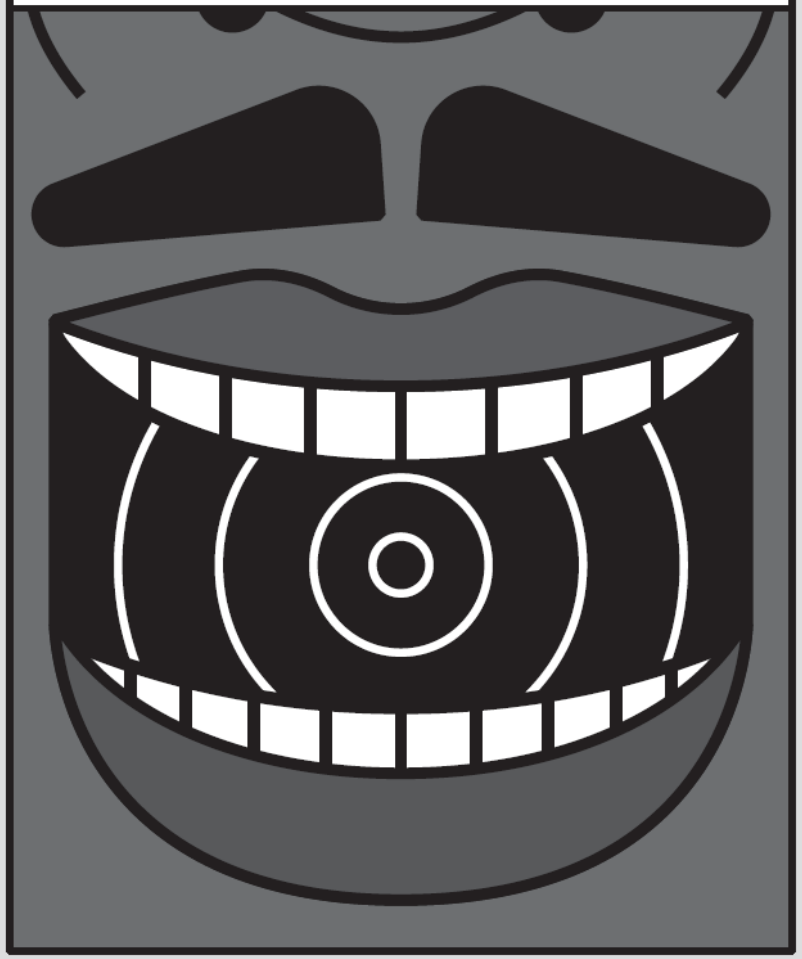
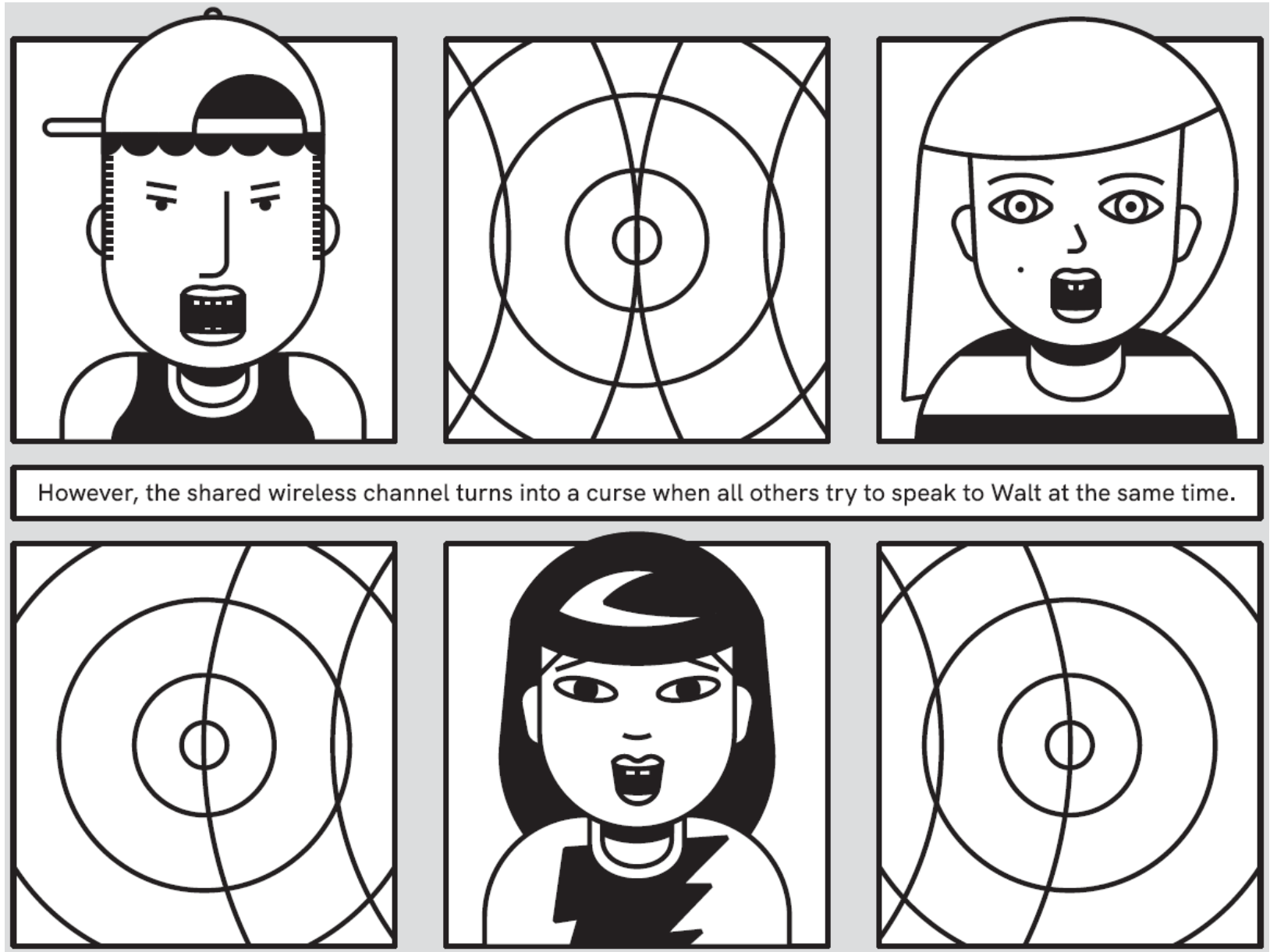
แก้ไขครั้งล่าสุดเมื่อวันที่ 31 มกราคม 2568

Chapter 3 – Medium Access Control

When the teacher Walt speaks to the students, the shared wireless channel is a blessing.

This is because it is sufficient that Walt says his thing only once, instead of repeating it for each student separately.




However, the shared wireless channel turns into a curse when all others try to speak to Walt at the same time.

Conversation Protocol

We start by describing wireless communication through an analogy with a conversation within a group of people, named Zoya, Yoshi, and Xia. We will refer to these and some other characters throughout the book; the characters will stand for wireless devices, base stations, or similar. The *data* that they want to communicate to each other is the content of their speech, which is part of the conversation. Regardless of the speech content, the conversation can only take place if the participants follow some *conversation protocol*, such that at a given time only one person speaks while the others listen. How do they agree who gets to speak and who gets to listen? One way would be, before starting the actual conversation, to have them agree upon which conversation protocol should be followed. In that case the information exchanged in that preliminary conversation cannot be regarded as useful data, but rather as *metadata*, also called *protocol information* or *control information*. The metadata is necessary in order to enable the conversation to take place. But then, how do they agree on the protocol for exchanging the metadata?

- มารยาท ท่าทาง ผ่านทางสายตาเป็น visual channel ตะหากจาก audio channel
- ใน wireless comm. ที่มี channel เดียว ต้องกำหนด protocol ให้ชัดเจน เช่น การใช้วิทยุสื่อสาร "วอ1 เรียก วอ2 วอ2 ทราบแล้วเปลี่ยน (roger that)"

Metadata, protocol info, control info

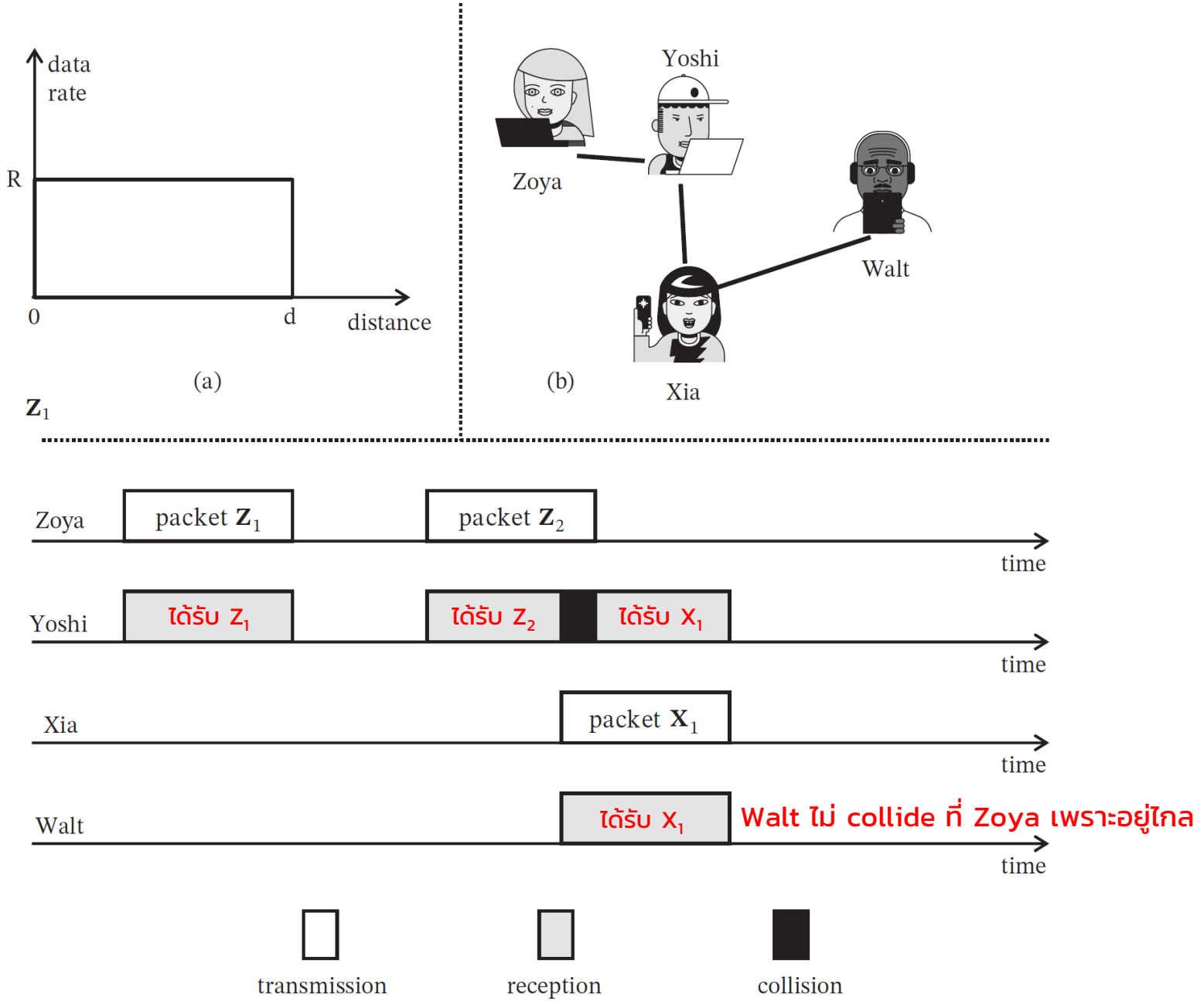
System Model (a set of assumptions)

- *Half–duplex*: A given person, e.g. Zoya, cannot speak and listen at the same time.
- *Broadcast*: If Zoya has information to convey to Yoshi and Xia, then, provided that both Yoshi and Xia are listening, Zoya needs to say her message only once, and not repeat it individually to Yoshi and to Xia.
- *Interference*: If Yoshi and Xia speak simultaneously, Zoya will not understand either of them.

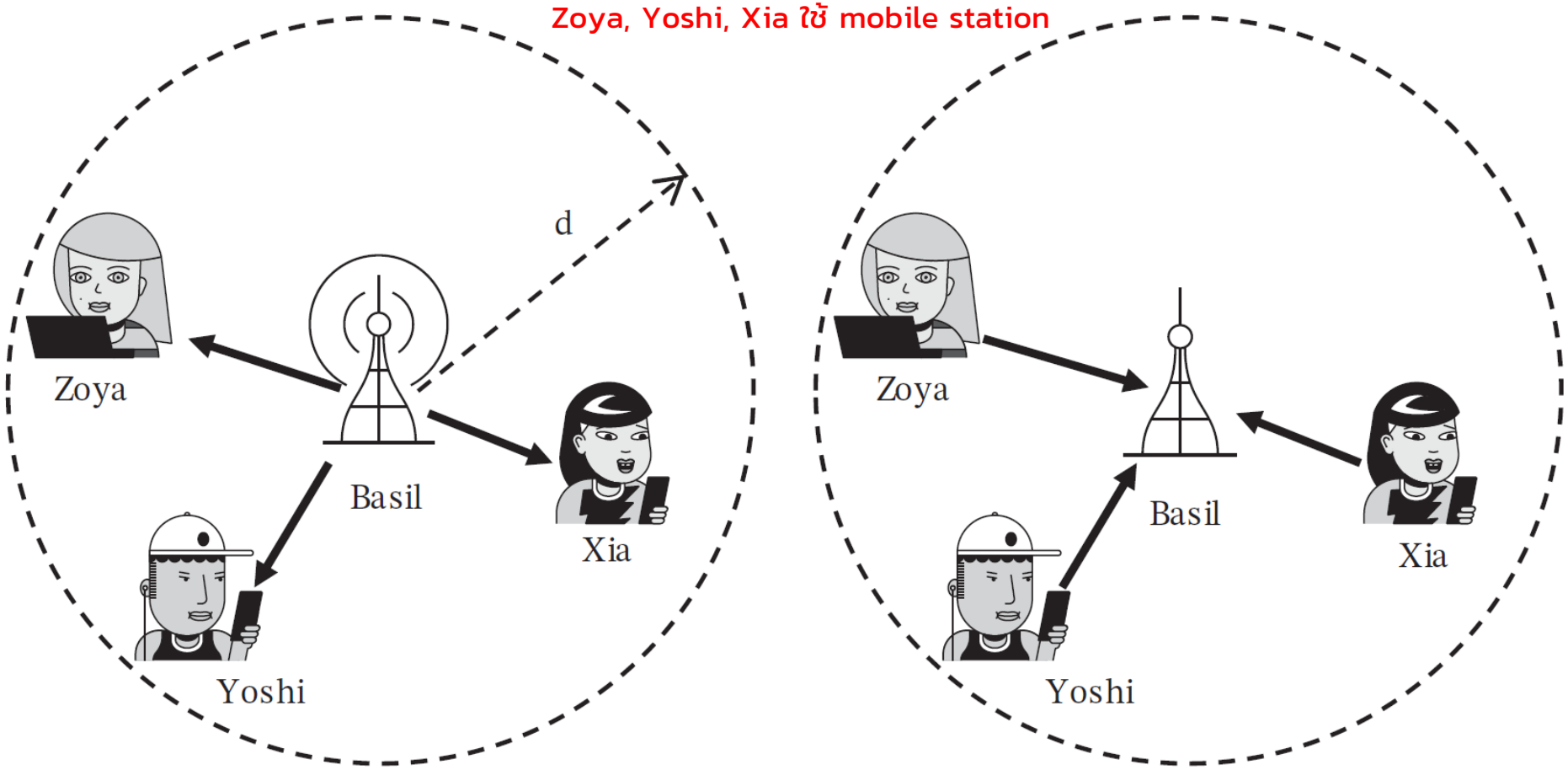
Collision Model

The communication between the wireless nodes is based on data packets. A transmitting node is capable of sending R bits per second (bps) such that a packet of duration T contains RT bits. All packets have the same duration, unless stated otherwise. In the collision model, a packet is treated as the smallest, atomic unit of information, such that either the whole packet is received correctly or it is lost. In other words, it is not possible to receive only some bits of the packet correctly. A packet sent by Zoya to Yoshi is received correctly if:

1. Yoshi is in the communication range of Zoya such that the distance between them is less than d m; สัญญาณแพร่ออกมาแบบ omnidirectional
2. No other communication node that is within d m of Yoshi transmits while Yoshi is receiving the packet from Zoya.


data rate
R
0
d
distance
(a)
Yoshi
Zoya
Walt
Xia
(b)
Z1
Zoya
packet Z1
packet Z2
time
Yoshi
ได้รับ Z1
ได้รับ Z2
ได้รับ X1
time
Xia
packet X1
time
Walt
ได้รับ X1
Walt ไม่ collide ที่ Zoya เพราะอยู่ไกล
time
transmission
reception
collision

Basil คือ base station
Zoya, Yoshi, Xia ใช้ mobile station



(a) ข้อดี (downlink)

(b) ข้อเสีย (uplink)

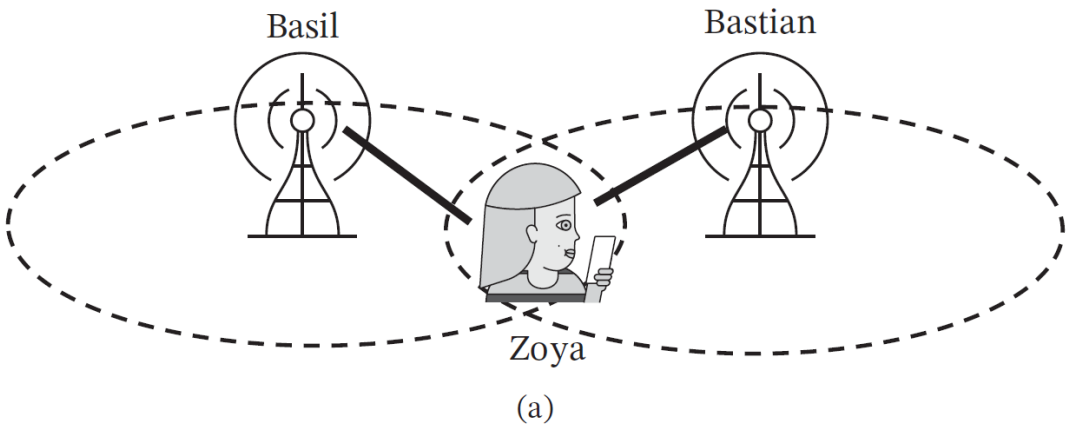
Figure 1.2 Illustration of two essential wireless features captured by the collision model. (a) Broadcast. (b) Interference.

Broadcast เหมาะสำหรับวิทยุ/โทรทัศน์

The First Contact

Back to the dark room analogy, we ask the question: how do two people, who have never met before, start to communicate when placed in a dark room? Reformulating this question in terms of wireless communication, we can ask: how do two wireless devices start to communicate? Who speaks first and who listens first? This is an important issue when the devices operate in a half-duplex manner, since a device cannot transmit and receive simultaneously. Before a packet from Zoya is sent to Yoshi, each of them needs to know that Zoya is about to transmit and Yoshi is about to receive. This may sound trivial and indeed it is, provided that we somehow let Zoya know in advance that she should take the transmitter role and Yoshi should take the role of a receiver. For example, if they have communicated in the past, then they may agree that, next time they are placed together in a dark room, Zoya takes the role of the one that starts to talk first. But, how do they know the roles if they have never communicated before? Let us explore this problem of first contact or rendezvous between two wireless nodes.

Hierarchy Helps to Establish Contact



1 channel
(half-duplex)

ไม่ collide ที่ Basil
เพราะอยู่ไกล

Figure 1.3 The problem of first contact when the mobile device Zoya is in the range of two base stations, Basil and Bastian. (a) Illustration of the scenario. (b) The invite packets of Basil and Bastian are persistently colliding. (c) Solution of the problem of collision between Basil and Bastian by using randomization.

Wireless Rendezvous without Help

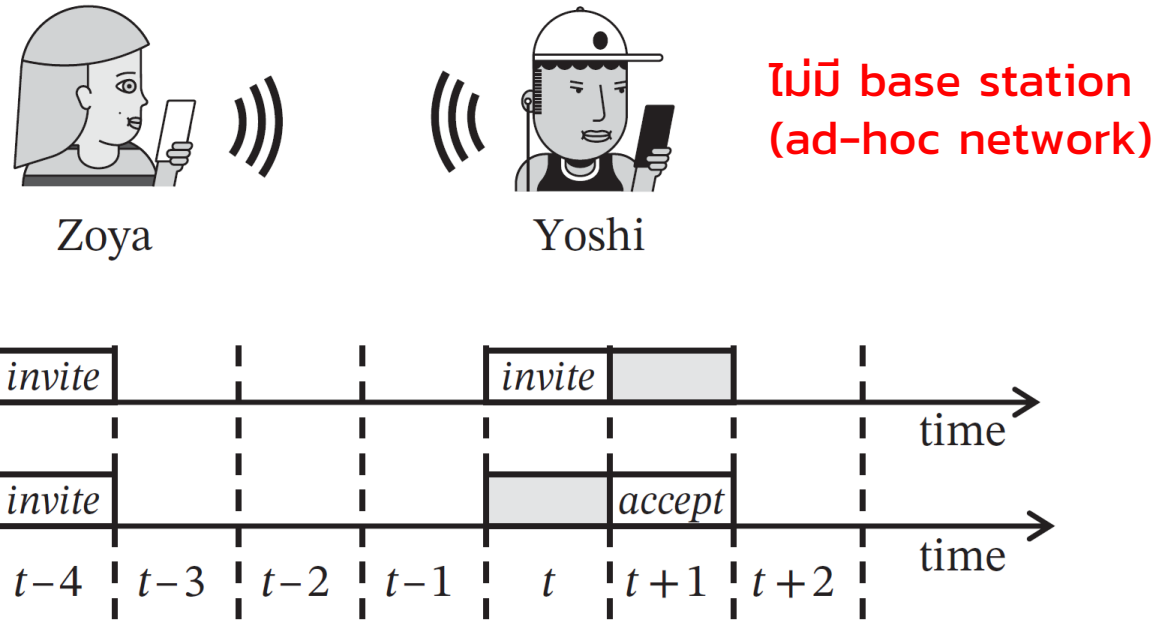


Figure 1.4 Rendezvous protocol for Zoya and Yoshi where both of them use half-duplex devices and their roles in the protocol are not predefined.

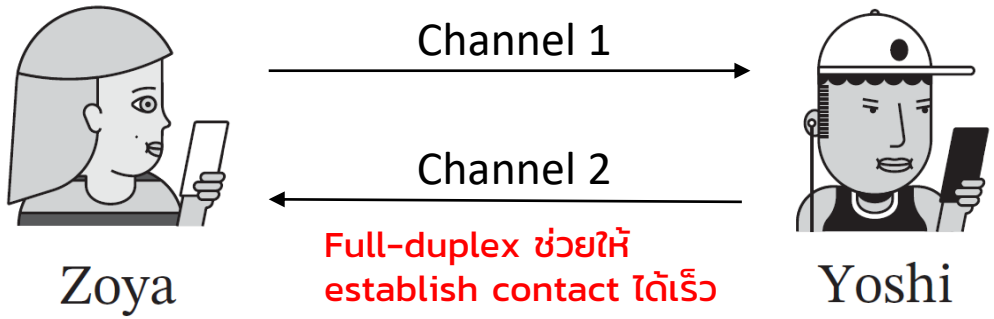
ไม่ sync ก็ได้เหมือนกัน

But what if Zoya and Yoshi are not synchronized in using the slotted channel; can that help or make things even more difficult? In fact, the synchronous case can be seen as the worst case: if both Zoya and Yoshi decide to transmit in a certain slot, then their packets are completely overlapping. On the other hand, the lack of synchronism can be helpful in breaking the symmetry between Zoya and Yoshi. For example, we can have the situation in which both Zoya and Yoshi decide to transmit, but the transmission slot of Yoshi starts slightly later than Zoya's slot. Yoshi starts to receive the *invite* packet and postpones his transmission in order to complete the reception of Zoya's packet. In this example, Yoshi adjusts its clock to Zoya and, after the *invite* packet is received, Yoshi sends back a reply using Zoya's slot timing. It can be seen that, when the devices are not in synchrony, the symmetry can be broken without using randomization, as some form of randomization is already embedded in the asynchronism. However, recalling our discussion on a proper protocol design and controlled randomness, asynchronism can facilitate the rendezvous protocol, but it should not be the definitive solution; the protocol should always have the opportunity to rely on an intentional randomized choice.

Rendezvous with Full-Duplex Devices

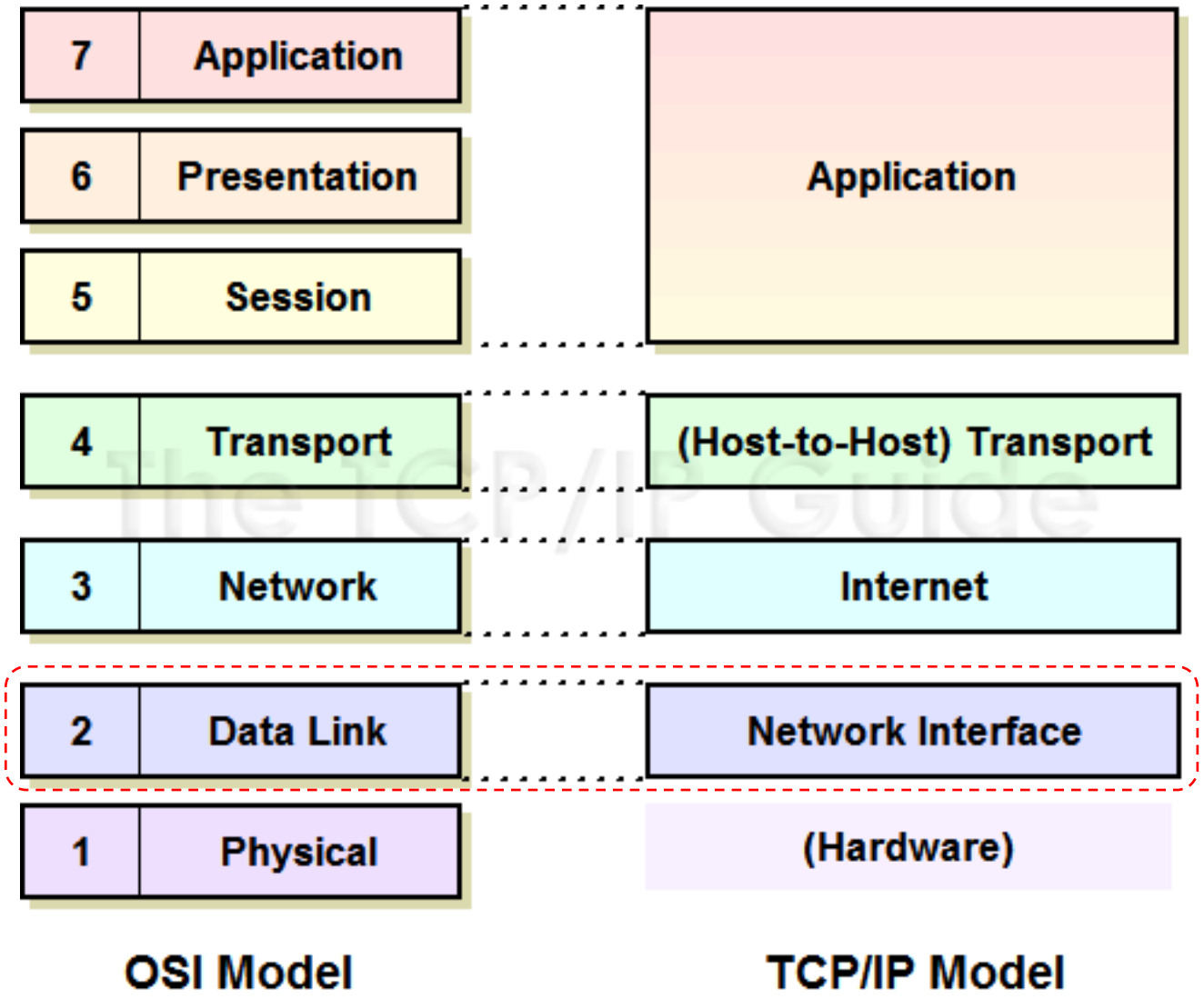
The problem of first contact becomes easier if the mobile devices are equipped with full-duplex capability. If Zoya and Yoshi both transmit simultaneously an *invite* packet to each other, then each of them will simultaneously receive the *invite* packet from the other device. Furthermore, in the next slot both Zoya and Yoshi transmit an *invitation_accepted* packet and, again, Zoya receives this packet from Yoshi and vice versa. With this, the link can be considered as being established. Therefore, full-duplex avoids the need for randomized assignment of transmit/receive roles and thus speeds up the procedure of link establishment. This may appear to be one of the most important advantages of full-duplex technology in scenarios in which fast link establishment and device discovery is of high importance.

We note that, even with full-duplex, there is still the problem of interference due to collision. Hence, full-duplex cannot help to solve the problem of colliding *invite* packets in Figure 1.3, as Zoya still does not get either of the two *invite* packets.



Full-duplex ช่วยให้
establish contact ได้เร็ว

Wired vs. Wireless



L1: Analog/digital modulation

L2: การ establish contact (ปัญหาห้องมืด)
การจัดสรร/แบ่งปัน medium ทั้ง wired และ wireless
L2 คือ local area network (LAN)

L3: มี LAN หลายวง เชื่อมกันเป็นอินเทอร์เน็ต

L4: ทำให้เสมือนมีสายสัญญาณเชื่อมคอมพิวเตอร์ 2 เครื่อง เข้าด้วยกัน เสมือนว่า packet 1, 2, 3, ... วิ่งไปตามสายนี้ตามลำดับ ทั้งที่จริงอาจใช้เส้นทางแตกต่างกัน และวิ่งแซงกันได้

Medium Access Control (MAC) อยู่ใน layer นี้

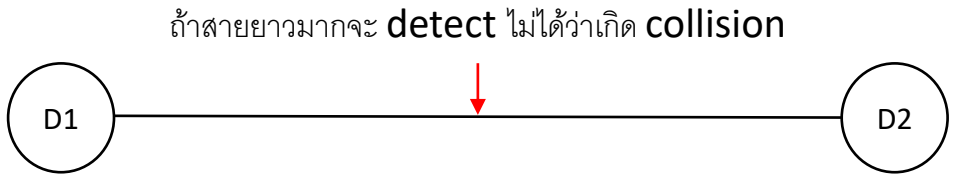
จะยืม MAC protocol ของ wired (TCP/IP) มาใช้ก็ไม่เหมาะ ต้องพัฒนา MAC สำหรับ wireless โดยเฉพาะ

Motivation for a Specialized MAC

เอา MAC ของ wired network มาใช้กับ wireless network ได้หรือไม่ ?

Local area network (LAN) ใช้ carrier sense multiple access with collision detection (CSMA/CD)

- Medium คือ สายโลหะ เช่น coaxial cable
- ผู้ส่ง sense ใน medium ก่อน ถ้าว่างค่อยส่ง ถ้า detect ได้ว่าเกิด collision จะส่ง jamming signal แล้วเริ่มส่งใหม่
- ใน wired network อุปกรณ์ทุกตัวบนสายสามารถ detect ได้ว่าเกิด collision บนสายสัญญาณ โดยดูจาก voltage ที่เพิ่มขึ้นมากผิดปกติในสาย เป็นเหตุผลที่ว่าทำไมถึงห้ามสายสัญญาณยาวเกินกี่เมตร



- ใน wireless network มีปัจจัยอื่น ๆ นอกจากระยะทาง เช่น สภาพอากาศ สิ่งกีดขวาง แบตเตอรี่ การเคลื่อนที่ ฯลฯ ทำให้ detect ไม่ได้ว่าเกิด collision

Hidden and Exposed Terminals

B is exposed to A and C. A is hidden to C and vice versa.

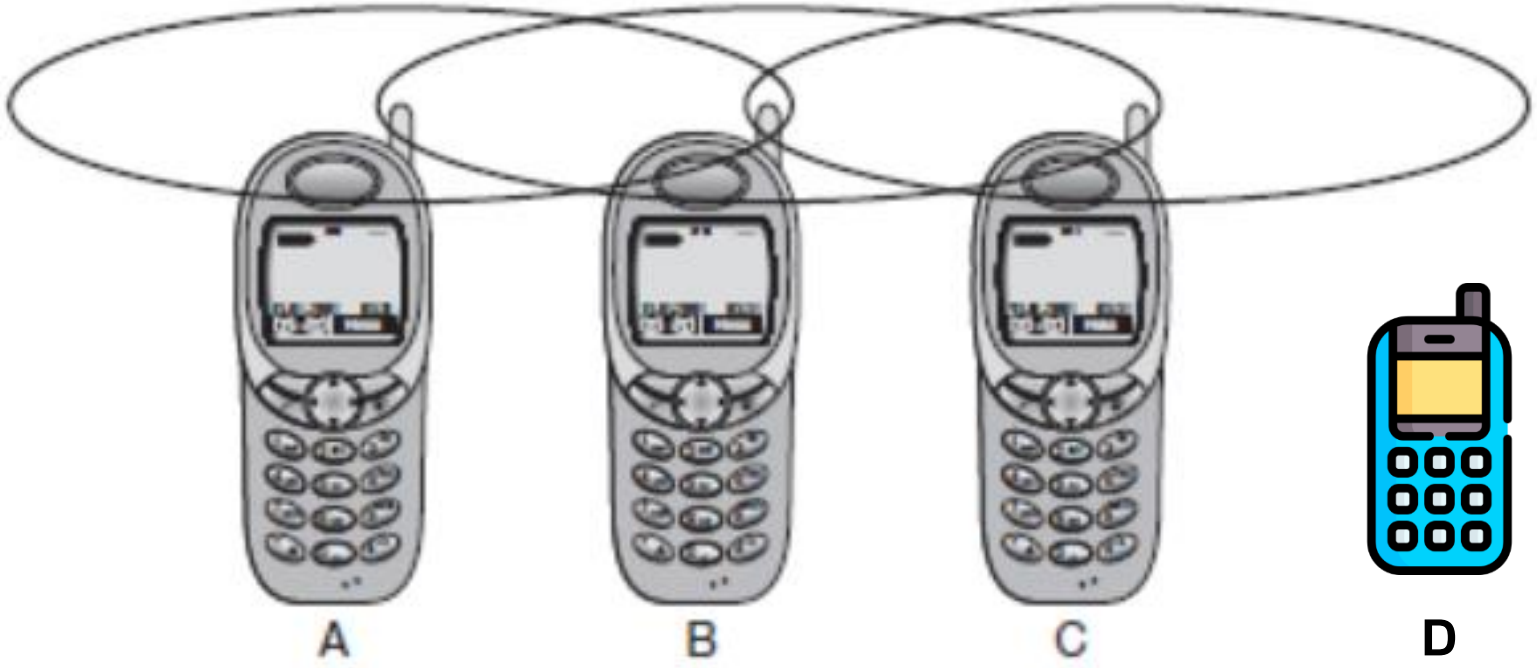


Figure 3.1
Hidden and exposed terminals

1) A กับ C ไม่รู้เลยว่า กำลังพูดพร้อมกัน ทำให้ B ฟังไม่รู้เรื่อง (ทำ collision detection ไม่ได้)
2) B ส่งให้ A และ C ส่งให้ D พร้อมกัน B คิดว่าเกิด collision เลยส่งใหม่ ทั้งที่ไม่จำเป็น C ไม่ collide ที่ A เพราะอยู่ไกล

Near and Far Terminals

- สมมติว่ากำลังส่งของ A และ B แรงเท่าๆ กัน near/far effect ทำให้ C ได้ยินแต่ B ไม่ได้ยิน A (เช่น C เป็น base station)
- Precise power control ที่ผู้ส่งสำคัญมาก เพื่อให้ผู้รับได้รับสัญญาณแรงเท่า ๆ กัน
- ระบบ CDMA/UTMS ปรับกำลังส่ง 1,500 ครั้งต่อนาที (ไม่ดี ซับซ้อน)

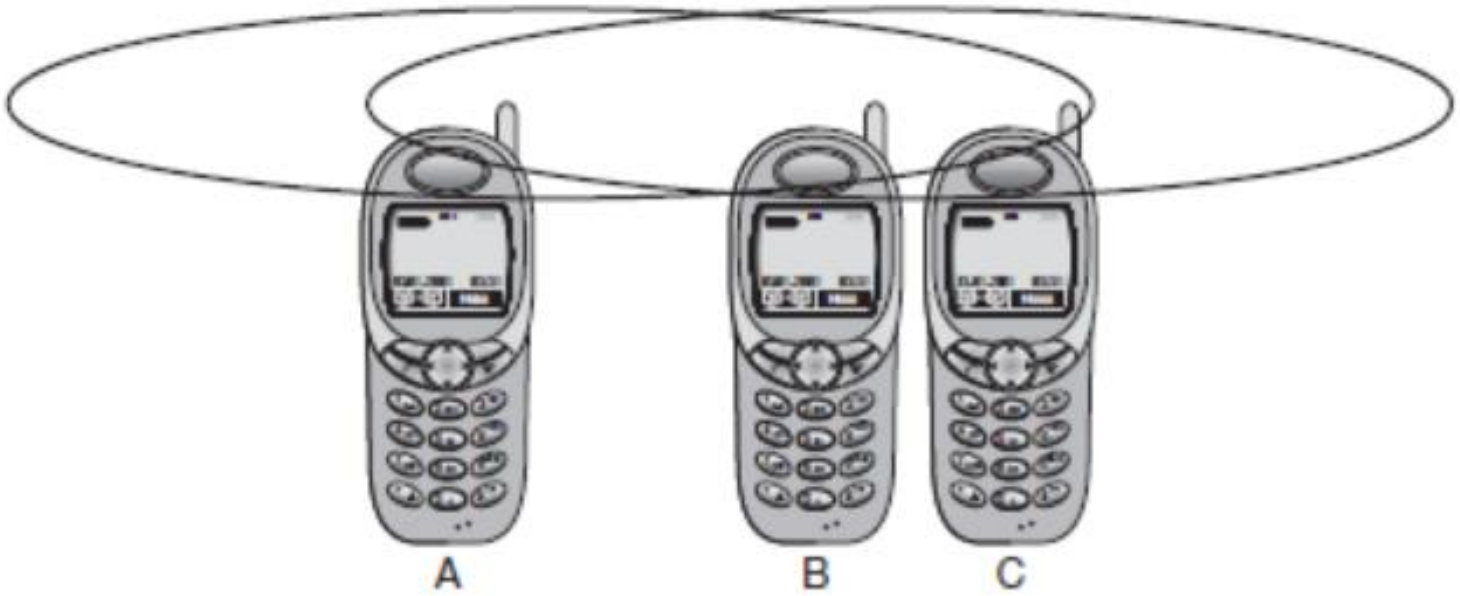


Figure 3.2
Near and far terminals

C ฟัง A ไม่ได้ยิน เพราะอยู่ไกล เสียงเบา โดนเสียง B กลบหมด

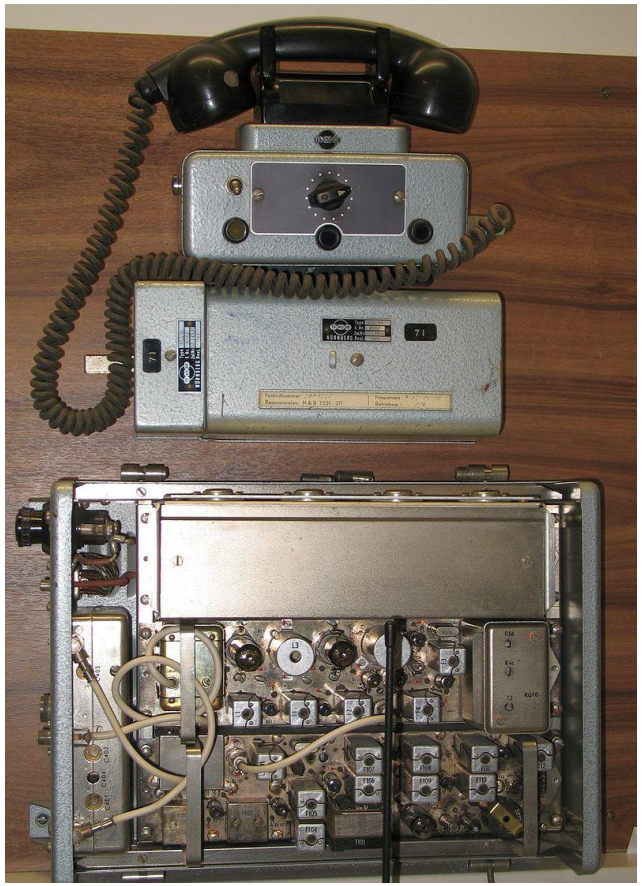
สมมติ C เป็น base station, A และ B แย่งกันขออนุญาตพูด B จะได้สิทธิ์พูดก่อน (หา fair scheme ไม่ได้)

Multiplexing

สรุปมี 4 แบบ

- Space
- Frequency (2G)
- Time (2G+) ใช้ time เพื่อรองรับผู้ใช้งานที่เพิ่มขึ้นใน cell
- Code (3G)

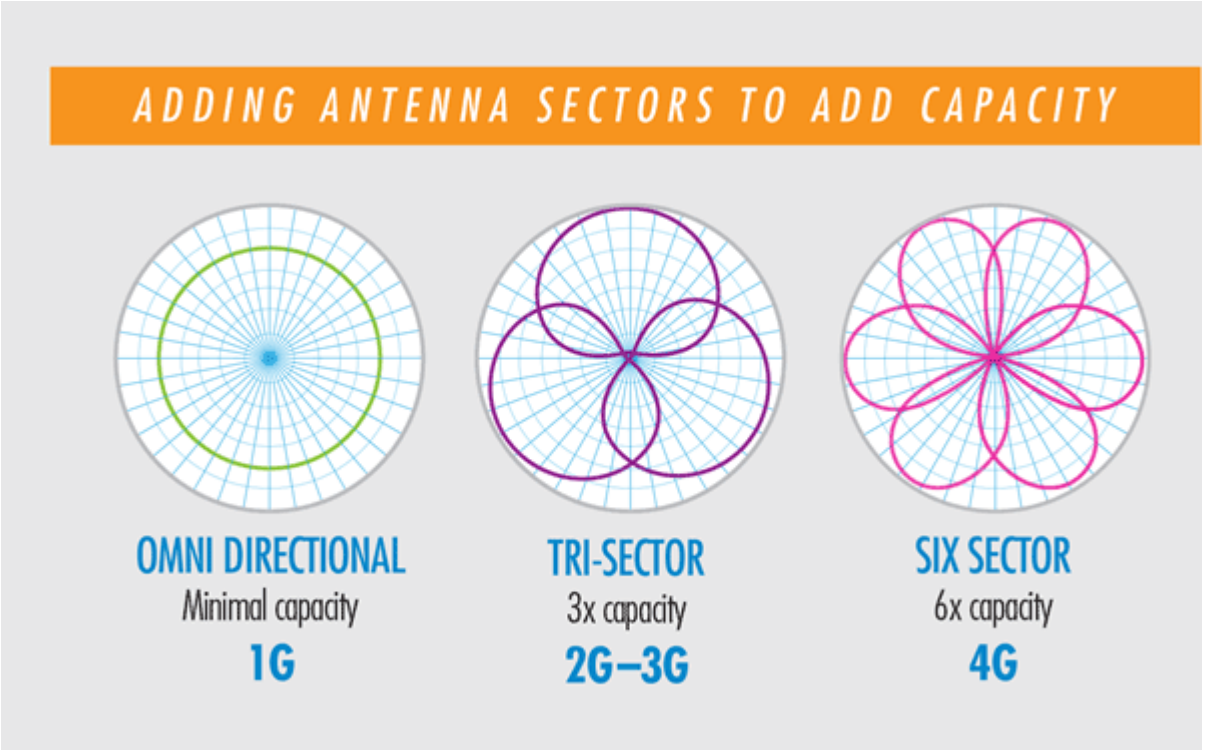
1G ใช้กับ mobile station ที่ไม่เคลื่อนที่ข้าม cell ไม่มีการทำ handover
เช่น ระบบโทรศัพท์ A-Netz ในประเทศเยอรมนี หลังสงครามโลกครั้งที่สอง



https://de.wikipedia.org/wiki/A-Netz

Space Division Multiple Access (SDMA)

- แบ่ง space ใน wireless network โทรศัพท์มือถือต้องตัดสินใจว่าจะเกาะอยู่กับ base station ใด
- SDMA มักจะใช้ร่วมกับวิธีอื่นๆ เช่น FDMA, TDMA, CDMA



Frequency Division Multiple Access (FDMA)

- Assign ความถี่แบบ fixed เช่น สถานีวิทยุ หรือแบบ dynamic ที่ assign ความถี่ให้ base station ตามความต้องการ
- ใช้แบ่ง channel เพื่อทำ duplex channel ระหว่าง mobile device กับ base station
- Frequency Division Duplex (FDD) แบ่งย่านความถี่เป็น 2 ช่วงคือ uplink และ downlink ในข้างล่างคือ GSM standard 900 MHz มี 124 channels ต่อ 1 base station ได้พร้อมๆ กัน uplink/downlink channel ถูกจับคู่กันไว้แล้ว (fixed)

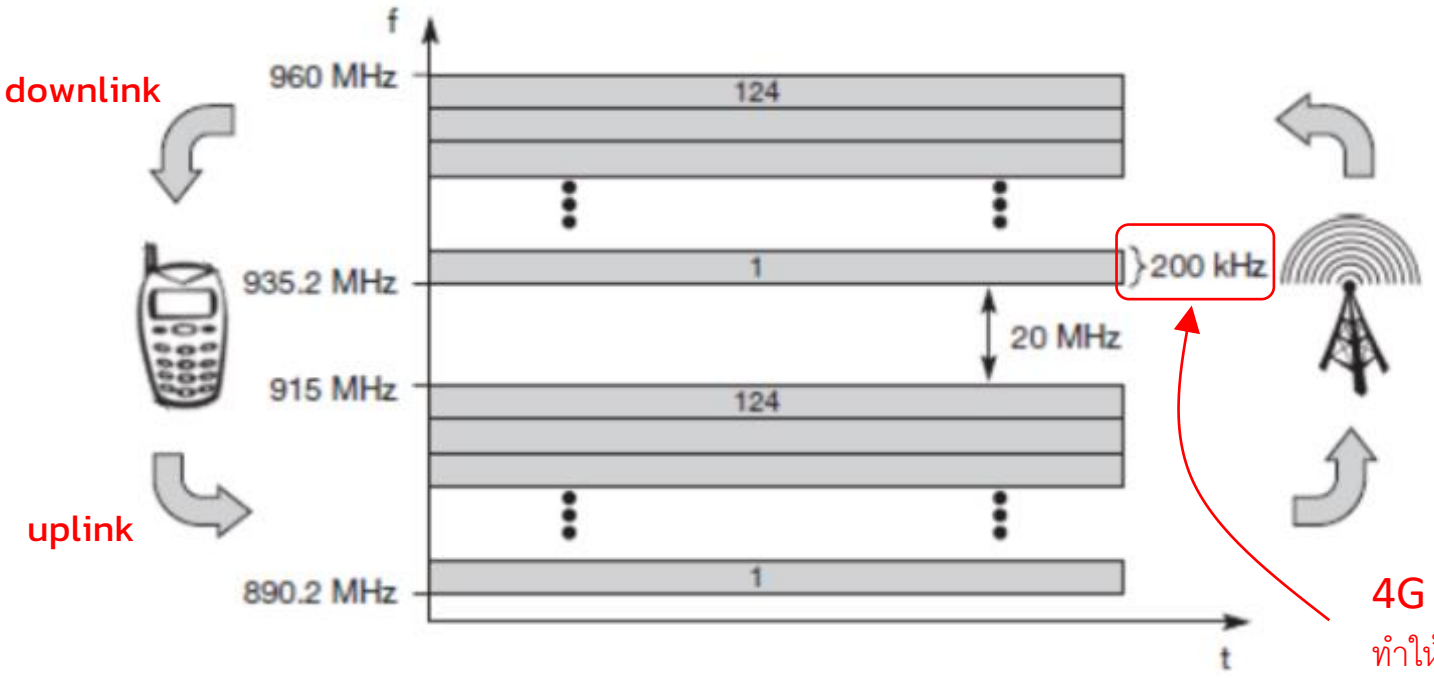


Figure 3.3
Frequency division multiplexing for multiple access and duplex

GSM Standard (2G)

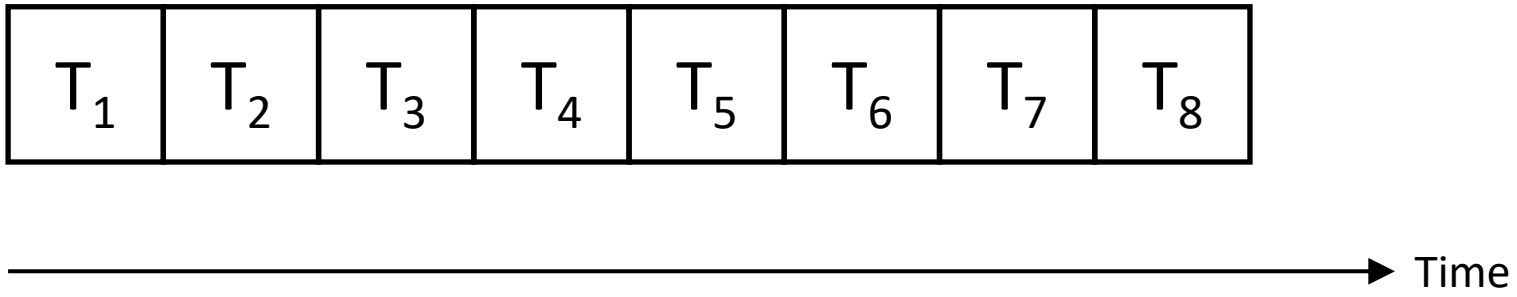
4G มีเทคนิคที่ทำให้ใช้แค่ 15 kHz
ทำให้ได้จำนวน channel มากกว่าบน bandwidth ที่เท่ากัน

Time Division Multiple Access (TDMA)

เหมือน Round Robin ในวิชา OS ที่ process ผลัดกันใช้ CPU

Process ผู้ส่งข้อมูลลงมาบน medium

CPU medium (space) ที่ทุกคนต้องใช้ร่วมกัน แต่ใช้ได้ทีละคน



Fixed TDMA

แบ่งแกนเวลาให้มี 12 channels สำหรับ uplink และ downlink

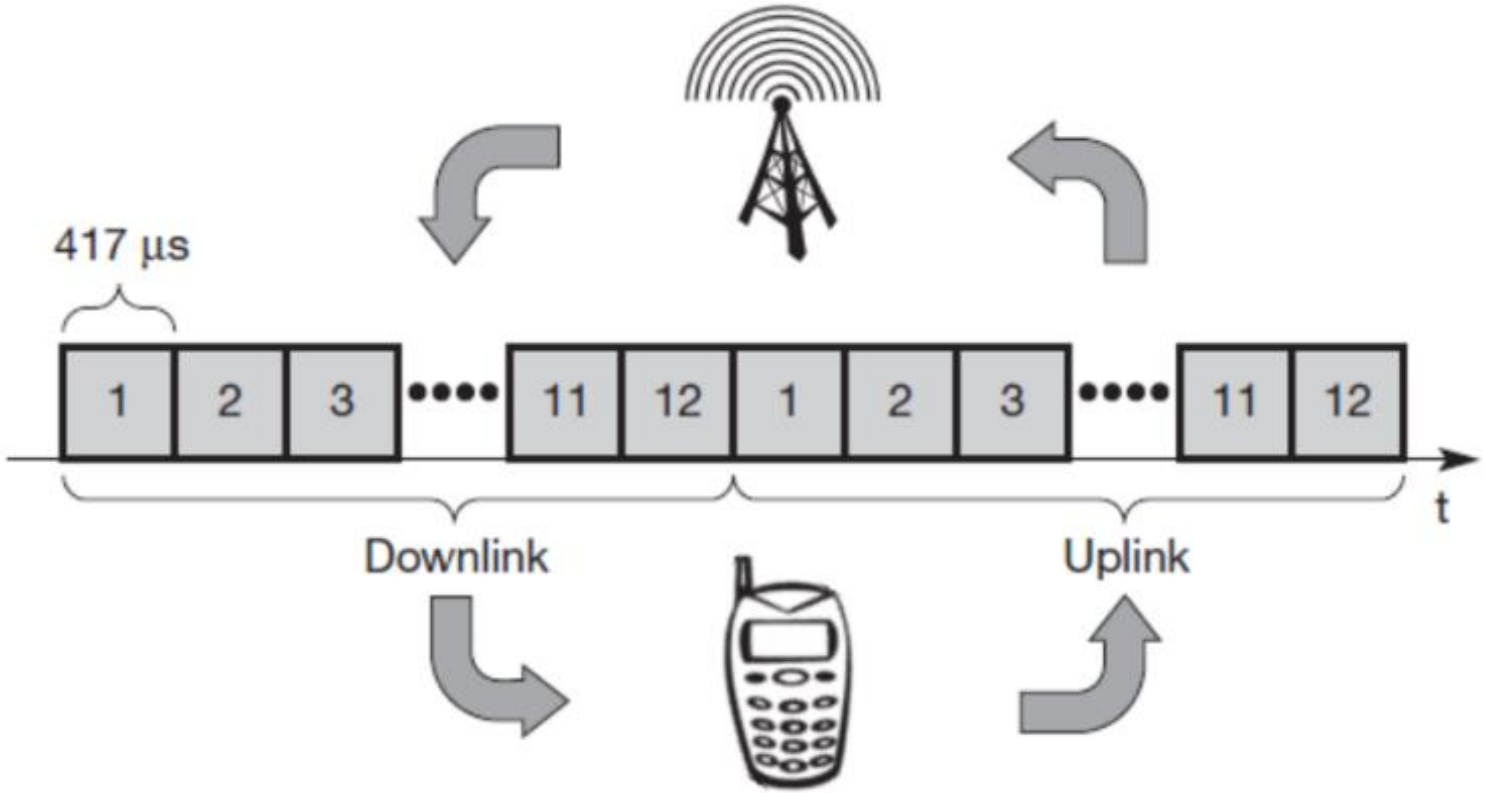
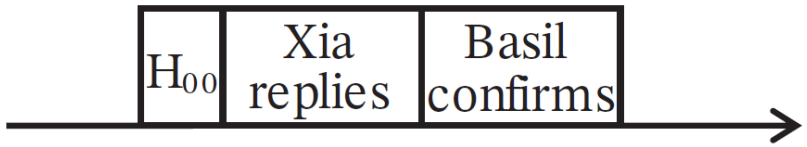


Figure 3.4
Time division multiplexing for multiple access and duplex

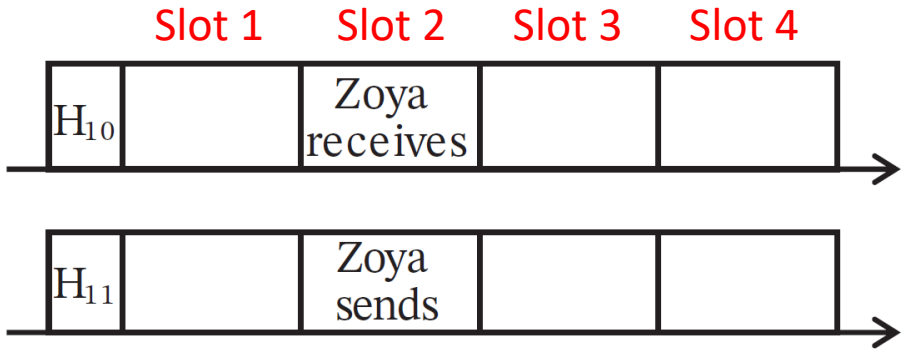
A Simple Two-Way System that Works Under the Collision Model

Basil ส่ง

H_{00} : link establishment frame
H_{01} : start of a link termination
H_{10} : this frame contains K slots for downlink transmission
H_{11} : this frame contains K slots for uplink transmission.



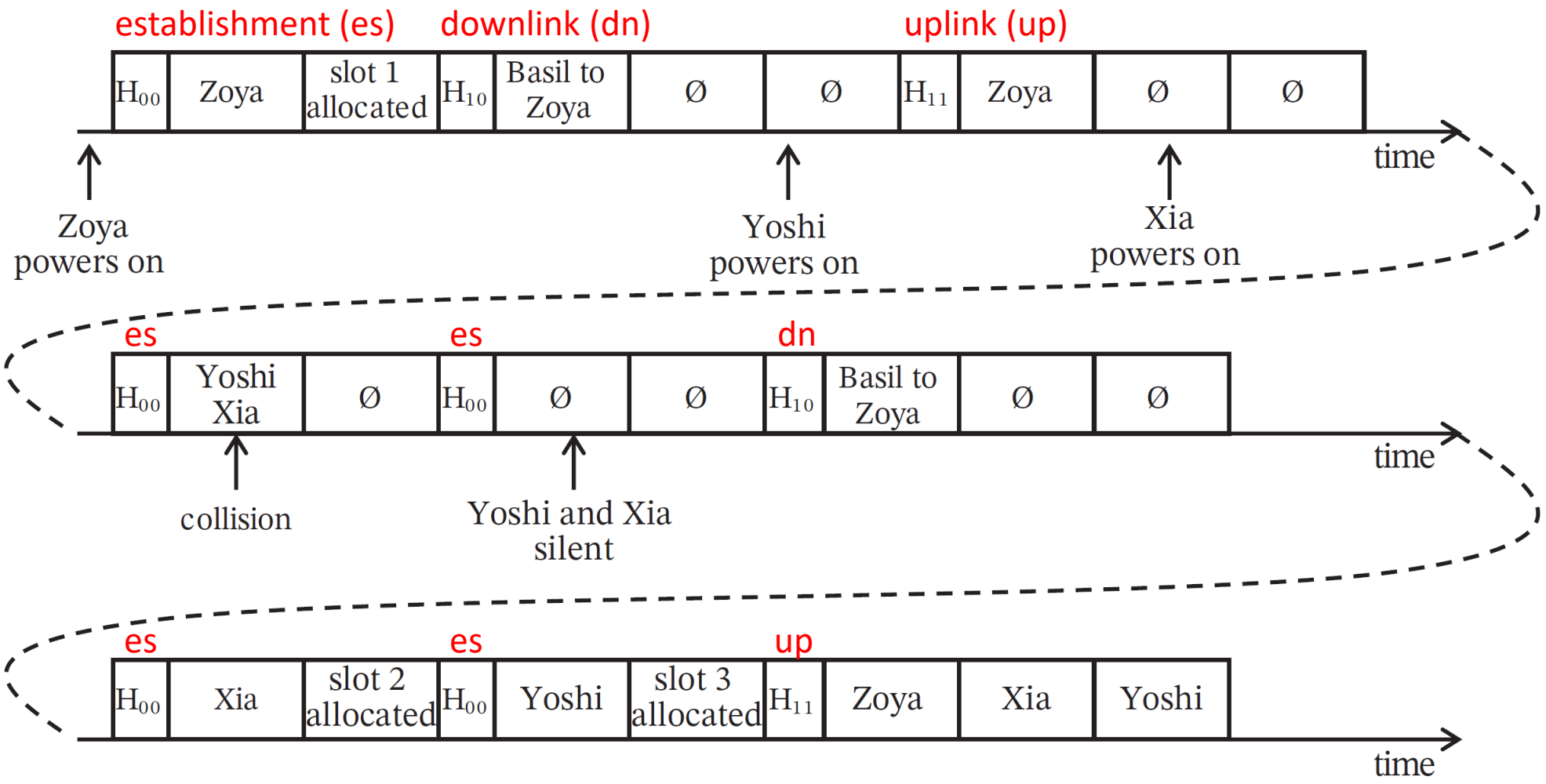
ถ้าแย่งกันตอบแล้วชนกัน Base Station (Basil) จะไม่ตอบ ทั้งคู่จะรู้ว่า collision ให้รอตอบใหม่ โดยสุ่มเวลารอ
ถ้ามี Xia ตอบคนเดียว Basil จะตอบกลับว่า Xia ได้ slot เบอร์อะไร



เมื่อสร้าง link (จอง slot ได้แล้ว) ก็จะใช้ slot เบอร์นั้น เช่น Zoya ใช้ slot เบอร์ 2
ถ้า header = H_{10} Zoya จะ download โดยใช้ slot เบอร์ 2
ถ้า header = H_{11} Zoya จะ upload โดยใช้ slot เบอร์ 2

H_{01} คือให้ตอบว่าใครจะ terminate แล้ว ตามด้วย Basil confirms (คล้าย ๆ H_{00})

A Simple Two-Way System that Works Under the Collision Model



CDMA หรือ 3G (ส่ง data bit แค่บิตเดียว)

ต้องใช้ key ที่ orthogonal กัน
(เหมือนภาษาที่ต่างกันมาก)

ถ้า fix ความยาว key ไว้
orthogonal key จะมีจำนวนจำกัด

- ผู้ส่ง A และ B ต้องการส่งข้อมูลพร้อมๆ กัน
- A ใช้ code หรือ key A_k = 010011 และ B ใช้ code หรือ key B_k = 110101
- A มี data bit A_d = 1 (แทนด้วย +1) และ B มี data bit B_d = 0 (แทนด้วย –1)
- A ส่ง signal $A_s = A_d * A_k = +1*(-1, +1, -1, -1, +1, +1) = (-1, +1, -1, -1, +1, +1)$
- B ส่ง signal $B_s = B_d * B_k = -1*(+1, +1, -1, +1, -1, +1) = (-1, -1, +1, -1, +1, -1)$
- สังเกตว่า A_k ตั้งฉากกับ B_k ($A_k * B_k = 0$) แล้วถ้า key ไม่ตั้งฉากกันจะเกิดอะไรขึ้น ?
- สัญญาณถูกส่งออกมาพร้อมกัน ที่ความถี่เดียวกัน ผสมกันในอากาศ $C = A_s + B_s = (-2, 0, 0, -2, +2, 0)$
- ถ้าผู้รับต้องการรับสัญญาณจาก A (ตั้งใจฟังภาษา A)

  $C * A_k = (-2, 0, 0, -2, +2, 0) * (-1, +1, -1, -1, +1, +1) = 2 + 0 + 0 + 2 + 2 + 0 = +6$

  ได้ positive value ดังนั้น ข้อมูลที่ส่งมาคือ 1
- ถ้าผู้รับต้องการรับสัญญาณจาก B

  $C * B_k = (-2, 0, 0, -2, +2, 0) * (+1, +1, -1, +1, -1, +1) = -2 + 0 + 0 - 2 - 2 + 0 = -6$

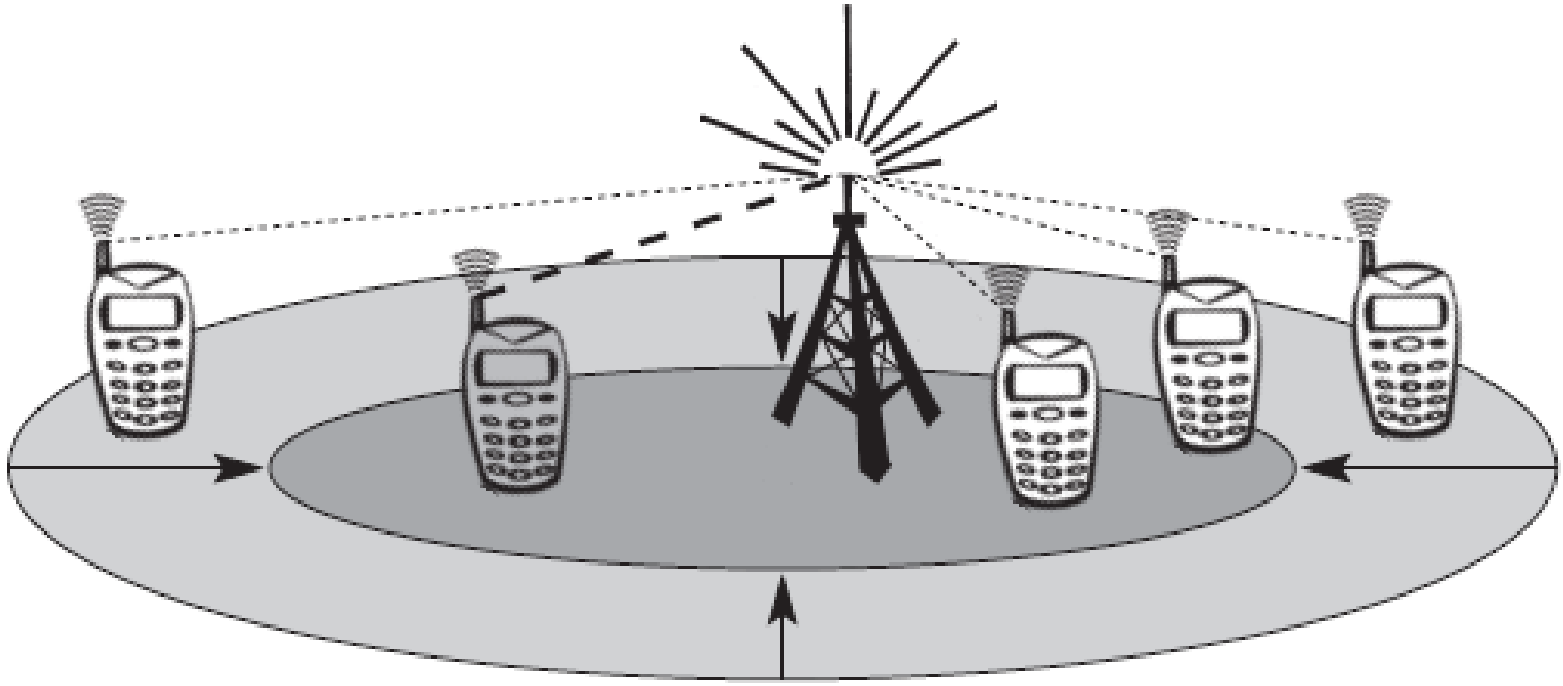
  ได้ negative value ดังนั้น ข้อมูลที่ส่งมาคือ 0
- ถ้ามีการรบกวน (noise) จะทำให้ได้ค่าลดลง ใกล้ศูนย์มากขึ้น ถ้ามี noise มากจะตัดสินยากว่าเป็น 0 หรือ 1

สมมติว่าสัญญาณ B แรงกว่า A 5 เท่า

- $C' = A_s + 5*B_s = (-1, +1, -1, -1, +1, +1) + (-5, -5, +5, -5, +5, -5) = (-6, -4, +4, -6, +6, -4)$
- ผู้รับสัญญาณ B: $C'*B_k = -6 -4 - 4 - 6 - 6 - 4 = -30$ (ลบเยอะ มองเห็นชัดว่าข้อมูลที่ส่งมาคือ 0)
- ผู้รับสัญญาณ A: $C'*A_k = 6 - 4 - 4 + 6 + 6 - 4 = 6$ (บวก ข้อมูลที่ส่งมาคือ 1 สัญญาณอ่อนกว่า B 5 เท่า และอยู่ใกล้ศูนย์มาก อาจจะทำให้ดูไม่ต่างจาก noise)
- เหมือนผู้พูด 2 คนใช้ 2 ภาษา แต่ถ้าภาษาหนึ่งพูดเสียงดังกว่ามาก ก็จะไปรบกวนอีกภาษาหนึ่ง
- เครื่องส่งในระบบ CDMA ต้องปรับกำลังส่ง (power) บ่อยมาก (>1,000 ครั้ง/นาที) ออกแบบเครื่องส่งยาก เปลืองพลังงาน

Cell Breathing / Load Balancing

Figure 2.43
Cell breathing depending on the current load



While a cell can cover a larger area under a light load, it shrinks if the load increases. The reason for this is the growing noise level if more users are in a cell. (Remember, if you do not know the code, other signals appear as noise)

https://en.wikipedia.org/wiki/Cell_breathing_(telephony)

Table 3.1 Comparison of SDMA, TDMA, FDMA, and CDMA mechanisms

Approach	SDMA	TDMA	FDMA	CDMA
Idea	Segment space into cells/sectors	Segment sending time into disjoint time-slots, demand driven or fixed patterns	Segment the frequency band into disjoint sub-bands	Spread the spectrum using orthogonal codes
Terminals	Only one terminal can be active in one cell/one sector	All terminals are active for short periods of time on the same frequency	Every terminal has its own frequency, uninterrupted	All terminals can be active at the same place at the same moment, uninterrupted
Signal separation	Cell structure directed antennas	Synchronization in the time domain	Filtering in the frequency domain	Code plus special receivers

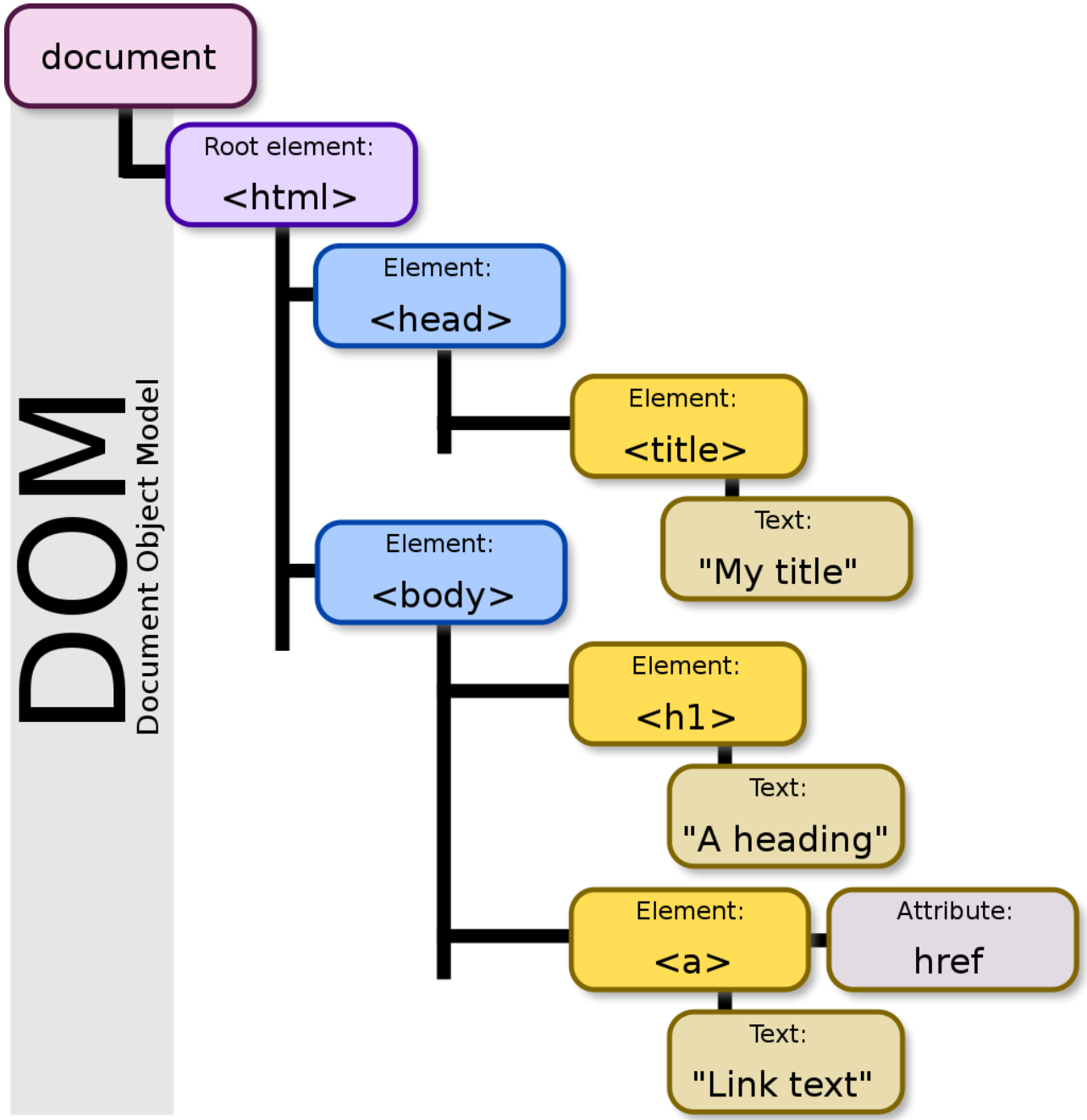
Advantages	Very simple, increases capacity per km^2	Established, fully digital, very flexible	Simple, established, robust	Flexible, less planning needed, soft handover
Disadvantages	Inflexible, antennas typically fixed	Guard space needed (multi-path propagation), synchronization difficult	Inflexible, frequencies are a scarce resource	Complex receivers, needs more complicated power control for senders
Comment	Only in combination with TDMA, FDMA or CDMA useful	Standard in fixed networks, together with FDMA/SDMA used in many mobile networks	Typically combined with TDMA (frequency hopping patterns) and SDMA (frequency reuse)	Used in many 3G systems, higher complexity, lowered expectations; integrated with TDMA/FDMA

https://en.wikipedia.org/wiki/Soft_handover

```
<!DOCTYPE html>
<html>
<body>

<h2>My First JavaScript</h2>

<button type="button"
onclick="document.getElementById('demo').innerHTML = Date()">
Click me to display Date and Time.</button>

<p id="demo"></p>

</body>
</html>
```

HTML DOM

- DOM Attribute
- DOM Console
- DOM Document
- DOM Element
- DOM Events
- DOM Event Objects
- DOM Geolocation
- DOM History
- DOM HTMLCollection
- DOM Location
- DOM Navigator
- DOM Screen
- DOM Style
- DOM Window
- WEB Storage



1. เอา code ไว้ใน head

```
<!DOCTYPE html>
<html>

<head>
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
</head>
<body>

<h1>A Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>

</body>
</html>
```

2. เอา code ไว้ท้าย body เพื่อให้แสดงผลเร็ว

```
<!DOCTYPE html>
<html>
<body>

<h1>A Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>

<script>
function myFunction() {
 document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>

</body>
</html>
```

3. แบบ external

```
<script src="myScript.js"></script>
```

JS vs jQuery

jQuery Selectors
jQuery HTML
jQuery CSS
jQuery DOM

jQuery was created in 2006 by John Resig. It was designed to handle Browser Incompatibilities and to simplify HTML DOM Manipulation, Event Handling, Animations, and Ajax.

jQuery vs. JavaScript

```
var myElement = $("#id01");
```

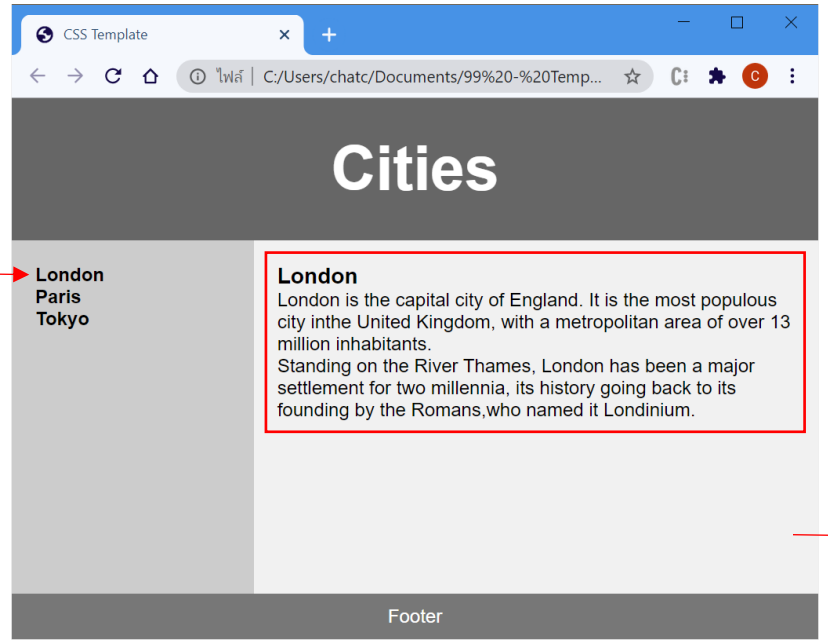
```
var myElement = document.getElementById("id01");
```

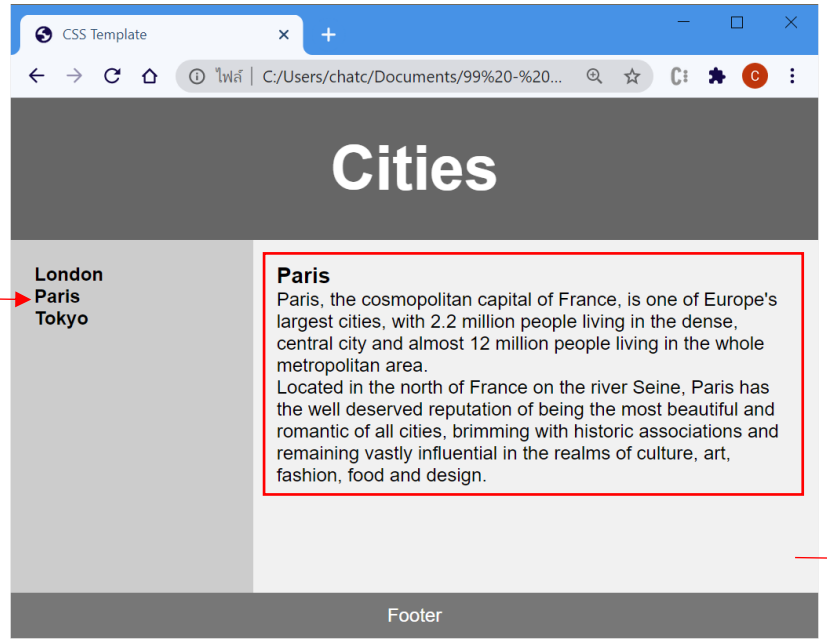
```
myElement.hide();
```

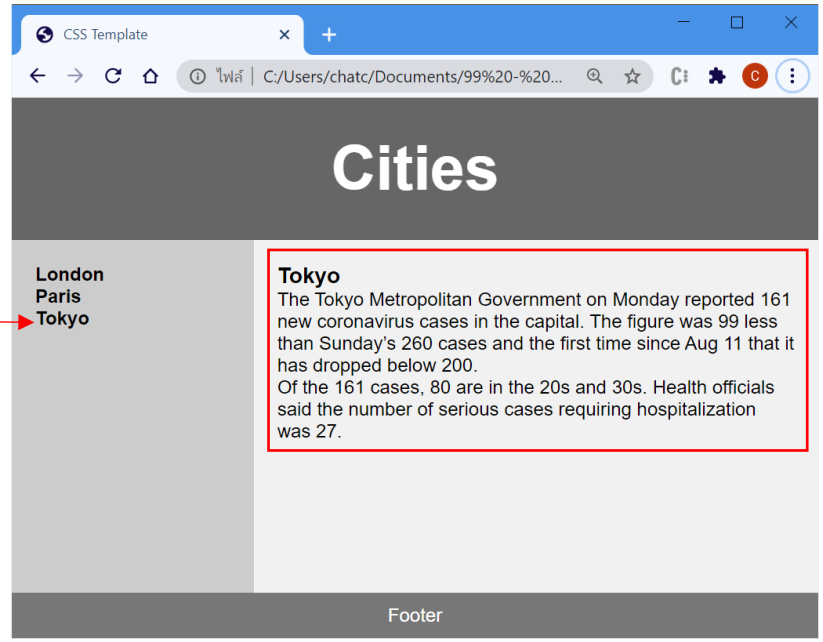
```
myElement.style.display = "none";
```

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
</head>
```

ใช้ URL ดีกว่า เพราะ web browser ส่วนใหญ่เก็บไว้ใน cache ไม่ต้องโหลดใหม่


CSS Template
ไฟล์ | C:/Users/chatc/Documents/99%20-%20Temp...
Cities
London
Paris
Tokyo
London
London is the capital city of England. It is the most populous city inthe United Kingdom, with a metropolitan area of over 13 million inhabitants.
Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans,who named it Londinium.
Footer


CSS Template
ไฟล์ | C:/Users/chatc/Documents/99%20-%20...
Cities
London
Paris
Tokyo
Paris
Paris, the cosmopolitan capital of France, is one of Europe's largest cities, with 2.2 million people living in the dense, central city and almost 12 million people living in the whole metropolitan area.
Located in the north of France on the river Seine, Paris has the well deserved reputation of being the most beautiful and romantic of all cities, brimming with historic associations and remaining vastly influential in the realms of culture, art, fashion, food and design.
Footer


CSS Template
ไฟล์ | C:/Users/chatc/Documents/99%20-%20...
Cities
London
Paris
Tokyo
Tokyo
The Tokyo Metropolitan Government on Monday reported 161 new coronavirus cases in the capital. The figure was 99 less than Sunday's 260 cases and the first time since Aug 11 that it has dropped below 200.
Of the 161 cases, 80 are in the 20s and 30s. Health officials said the number of serious cases requiring hospitalization was 27.
Footer

JavaScript

- ไม่ต้องประกาศ type ตัวแปร กำหนดตัวแปรขึ้นมาใช้ได้เลย เช่น x = 10 หรือ s = "hello"
- JavaScript จะคาดเดา type เอง มี number, string, boolean, object
- บอกแค่ว่าตัวแปรเป็น var, let, const (จะไม่บอกก็ได้)
- ไม่ต้องใส่ semicolon เมื่อจบ 1 คำสั่ง
- เนื่องจาก library เป็นแบบ camelCase เวลาเขียนโค้ดก็ควรจะใช้ camelCase
- ใช้ single/double quote ครอบ string ก็ได้ แต่นิยมใช้ single quote เพื่อให้มี double quote ใน string ได้ เช่น

```
let s = '<a href="https://www.google.com">Link to Google</a>'
```

ต้องการสงวน double quote ไว้ใช้กับ HTML

- มี if-else, for loop, while loop คล้าย ๆ ภาษา Java
- ประกาศฟังก์ชันได้ เช่น

```
function add(a, b) {
        return a + b
}
```

- ดู error ใน console (Chrome กด F12)
- JavaScript ถูก execute ที่เว็บเบราเซอร์ ใช้ cpu ของเครื่อง client

Basic

หา HTML element ที่ประกาศไว้แล้ว เช่น

```
<span id="result"></span>
```

เขียนโค้ด JavaScript ดังนี้

```
let x = document.getElementById('result')
```

เติมข้อความลงไประหว่าง tag

```
x.innerHTML = 'Hello world!'
x.innerHTML = '<a href="https://www.google.com">Link to Google</a>'
```

มีวิธีเพิ่ม HTML element ที่ดีกว่าเติม raw text

```
let e = document.createElement('p')
x.appendChild(e)
```

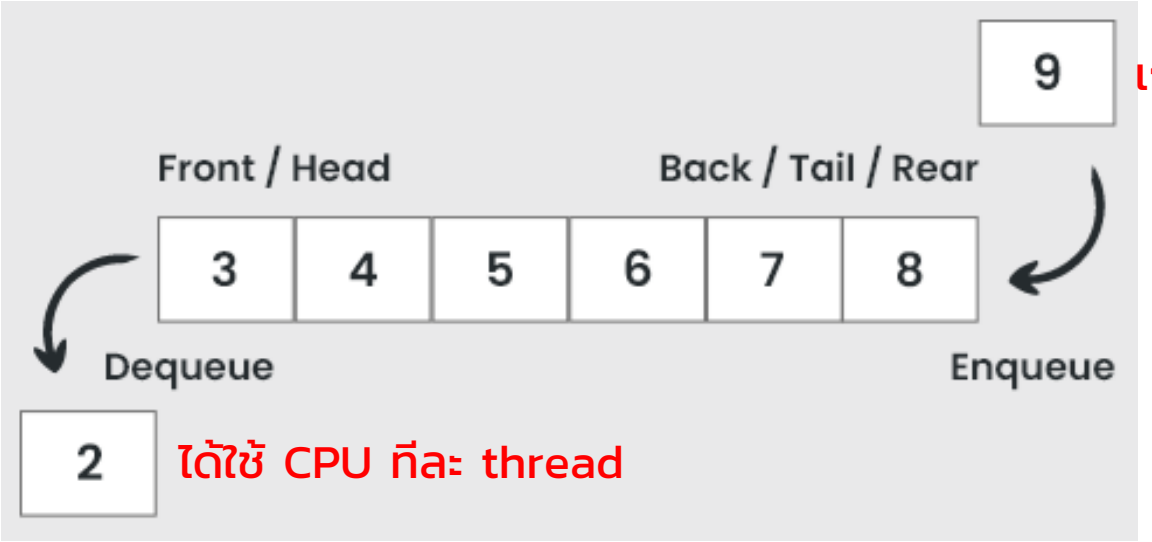
ปรับแต่ง element

```
e.innerHTML = 'I am a paragraph'
e.style.color = 'red'
```

ลบ HTML element

```
e.remove()
```

JavaScript is a single-threaded language.



เกิด event เช่น click

ได้ใช้ CPU ทีละ thread

then / catch (ต่อจาก OS บทที่ 11)

```
demo().then(
    (onResolved) => {
        // Some task on success
    },
    (onRejected) => {
        // Some task on failure
    }
)
```

Note: demo is a function that returns a promise prototype.

success failure always

เขียนแบบนี้ก็ได้

demo().then(...).catch(...).finally(...)

```
function demo() {
    console.log("Function called!!")
    return Promise.resolve("Success")
    // or
    // return Promise.reject("Failure")
}
demo().then(
    (message) => {
        console.log("Then success:" + message)
    }
)
```

```
function demo() {
    console.log("Function called!!")
    return Promise.resolve(1)
    // or
    // return Promise.reject("Failure")
}
demo().then(
    (value) => {
        console.log(value)
    return ++value
    },
    (message) => {
        console.log(message)
    }
    ).then(
    (value) => {
        console.log(value)
    },
    (message) => {
        console.log(message)
    }
)
```

https://www.geeksforgeeks.org/why-we-use-then-method-in-

```js
let demo = new Promise((resolve, reject) => {
    resolve(1)
})
let call = demo.then(
    (value) => {
        console.log(value)
        return ++value
    },
    (message) => {
        console.log(message)
    }
)
console.log(call)
setTimeout(() => {
    console.log(call)
})
```

```
Promise {status: "pending"}
1
Promise {status: "resolved", result: 2}
```

Syntax

```
JS

setTimeout(code)
setTimeout(code, delay)
```

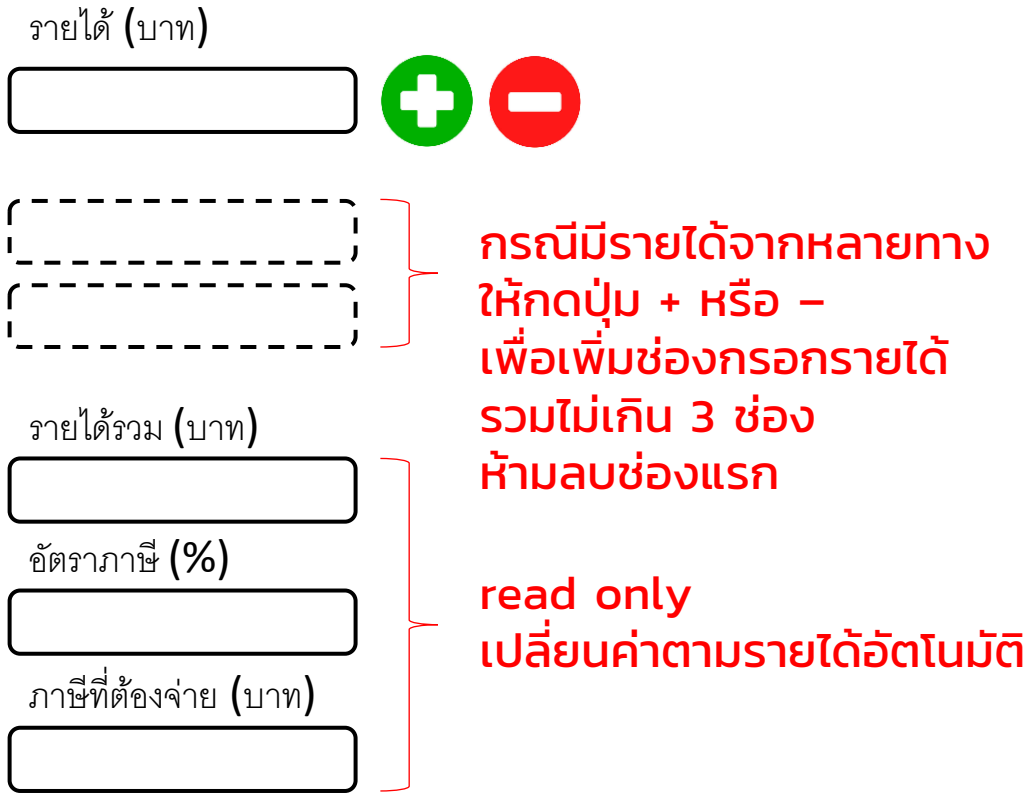
`delay` Optional

The time, in milliseconds that the timer should wait before the specified function or code is executed. If this parameter is omitted, a value of 0 is used, meaning execute "immediately", or more accurately, the next event cycle.

async / await

https://www.w3schools.com/js/js_async.asp

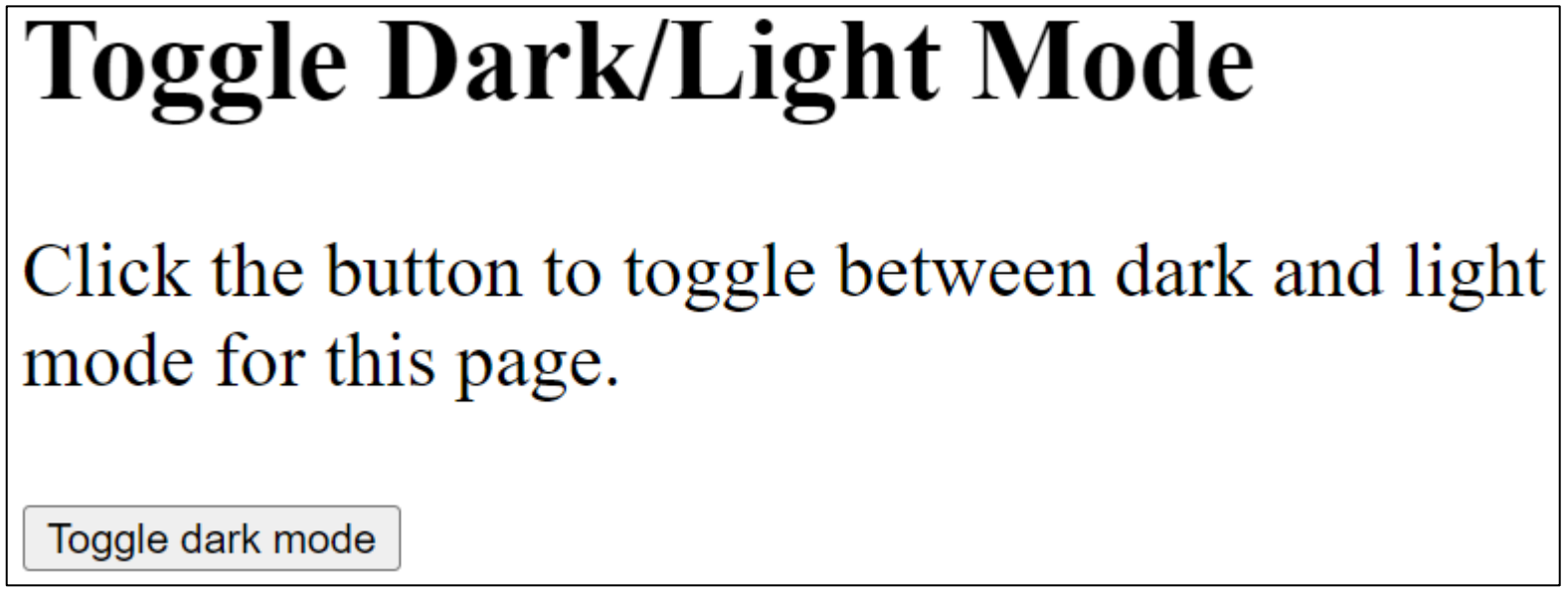
โปรแกรมคำนวณภาษีเงินได้บุคคลธรรมดา



แสดงข้อความเตือน เช่น มีรายได้บางรายการเป็นค่าลบ
หรือค่าที่เป็น non-numeric

อัตราภาษีเงินได้บุคคลธรรมดาใหม่	
เงินได้สุทธิต่อปี	อัตราภาษีเงินได้บุคคลธรรมดา
0–150,000 บาท	ได้รับการยกเว้น
150,001–300,000 บาท	5%
300,001–500,000 บาท	10%
500,001–750,000 บาท	15%
750,001–1,000,000 บาท	20%
1,000,001–2,000,000 บาท	25%
2,000,001–5,000,000 บาท	30%
5,000,001 บาทขึ้นไป	35%

ทำปุ่มสลับโหมดมืด/สว่าง (JavaScript ใช้แก้ไข style ได้ด้วย)



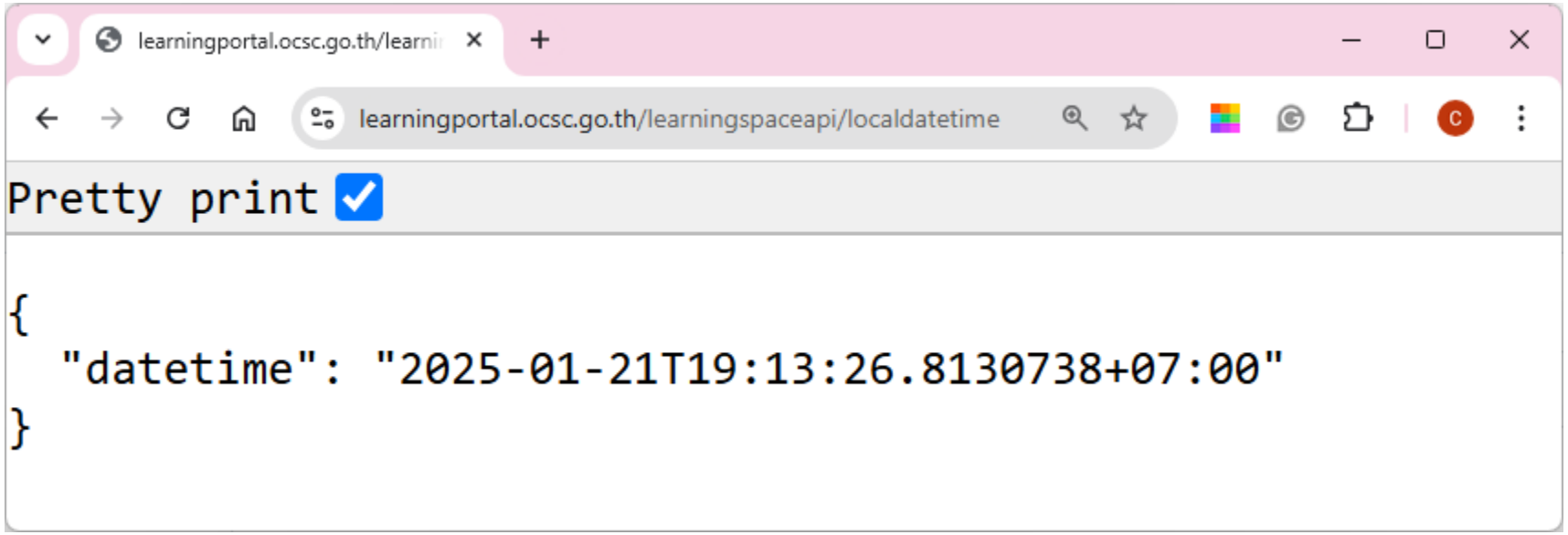
https://www.w3schools.com/howto/howto_js_toggle_dark_mode.asp

ให้ call API เพื่อแสดงวันเวลาภาษาไทย เช่น วันที่ 10 มกราคม 2568 เวลา 10:00:00 น. อัปเดตทุก 1 วินาที
ถ้า network error ให้แสดงข้อความ ระบบเครือข่ายล้มเหลว (จำลองโดยการปิดเน็ต)

fetch('https://learningportal.ocsc.go.th/learningspaceapi/localdatetime')
.then(...)
.catch(...)
.finally(...)

Default fetch() Timeout

By default, `fetch()` requests have a timeout defined by the browser. For instance, in Chrome, a request times out after 300 seconds, whereas in Firefox, it's 90 seconds. ตั้ง timeout ให้สั้น ๆ เช่น 3 วินาที



Checking response status

The promise returned by `fetch()` will reject on some errors, such as a network error or a bad scheme. However, if the server responds with an error like `404`, then `fetch()` fulfills with a `Response`, so we have to check the status before we can read the response body.