Why build computer networks ?

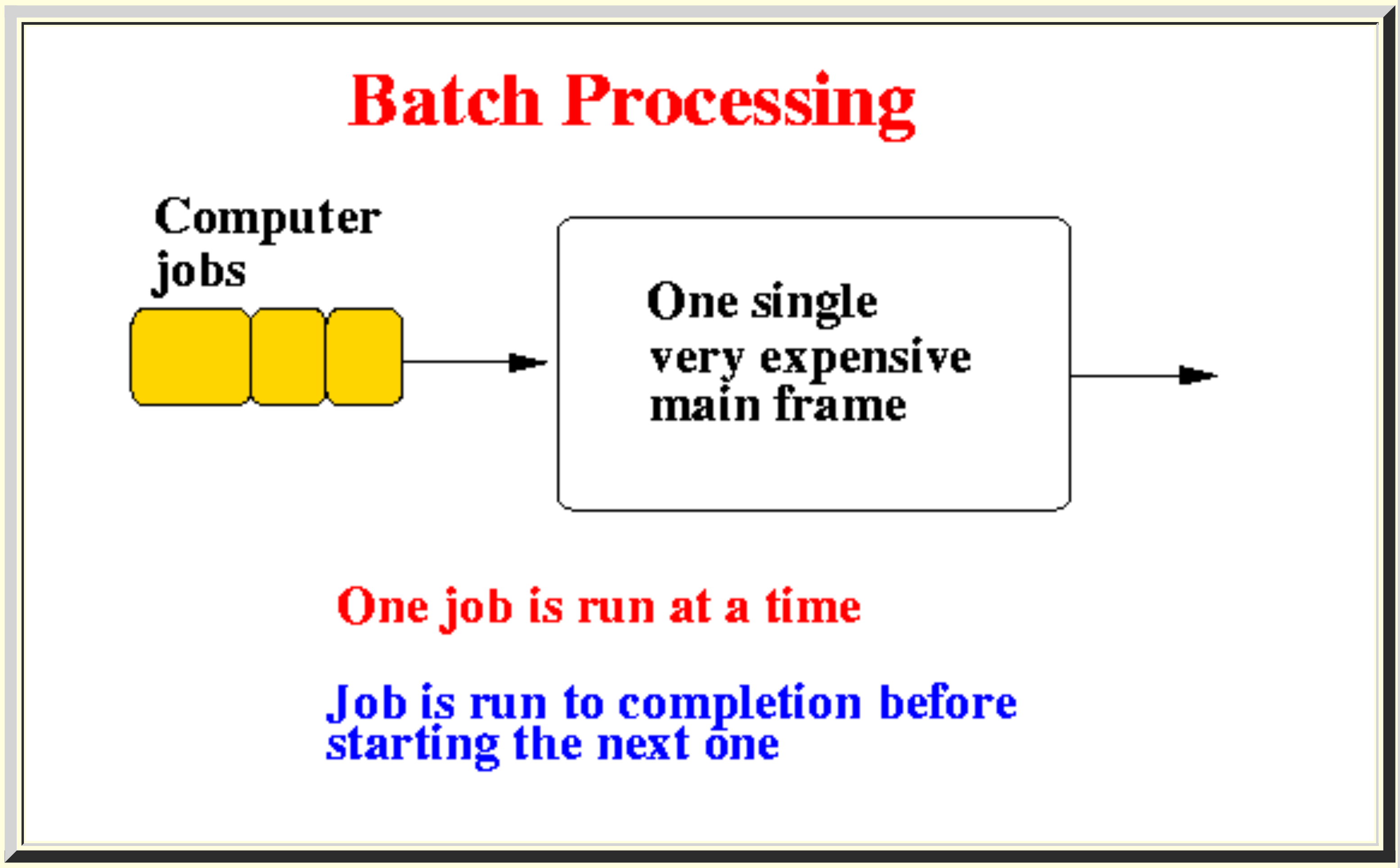
- **Trend of Computer Usage:**
  - **1950 - 1965**: **"Batch" processing**
    - Institutes have **one** (or a few) **powerful** and **expensive *stand-alone* main frame computer**:

      (The **IBM 360** --- with **tape drives** as storage)

    - A **computer** was **very expensive** (millions of dollars !!!)

    - **Jobs (programs)** are processed ***one* at a time** (batch process):

    - ***Very few*** **universities** and **corporations** coulg **afford** to **buy** a **computer**

- **Further information:**

  - IBM **main frame** history page: click here

- **1965 - 1978: time shared computing**

  - Computers still **very expensive** (**millions** for a **main frame** and **hundreds of thousands** for **mini computers**)

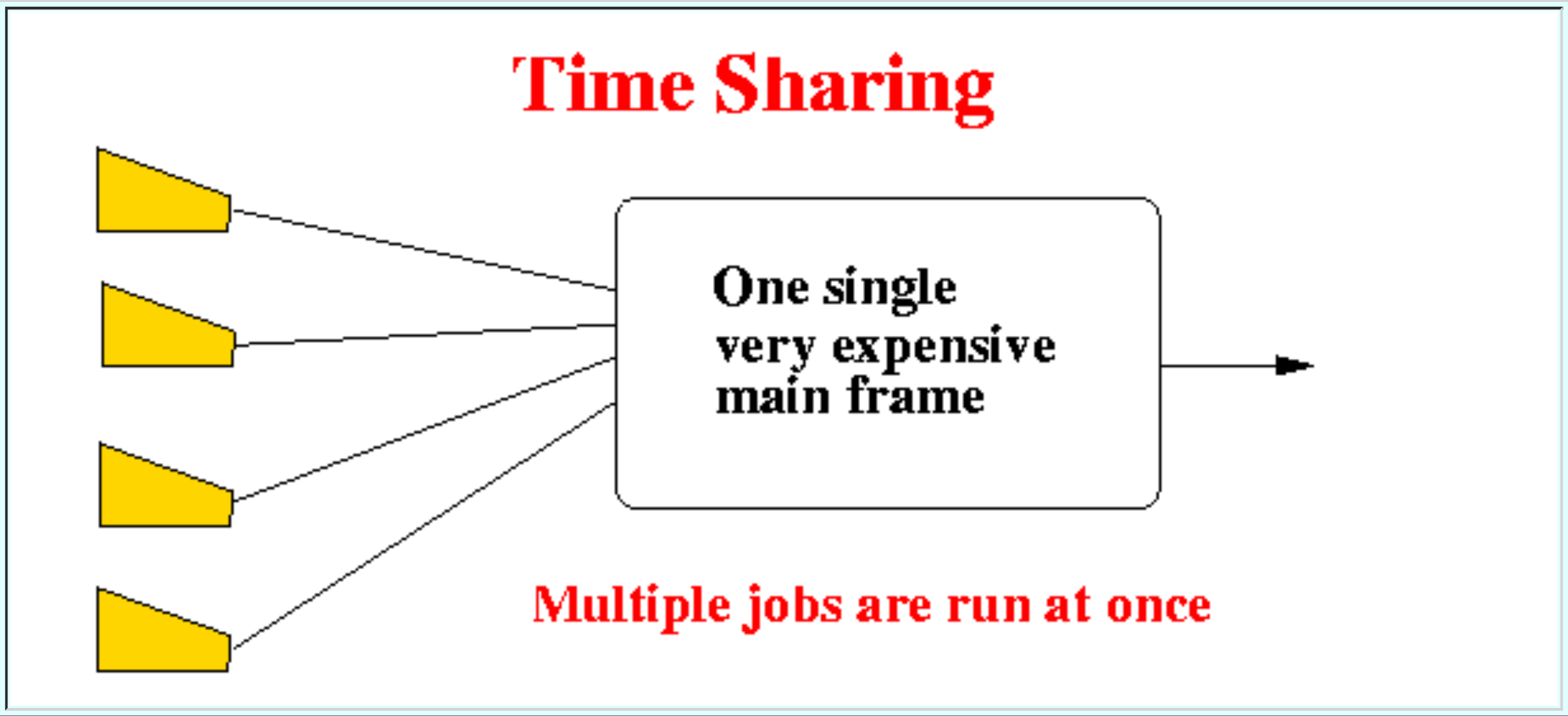
  - The **computer** was **accessed** by **many users** at the ***same* time** through **dumb terminals**

    (**I** have used this "classic" **VT100** terminal)

  - The **computer** executes ***multiple* programs** at the **"same"** time using the ***time sharing*** technique

    (The **time sharing** technique **switches** programs **rapidly** --- take the course **CS355** if you want to learn more...)

  - **More** universities and corporations can now afford a **mini computer** !!!

- **1978 - 1982: *personal* computing** (**8080 Intel chip** !)

  - **1978:** The ***first*** **general purpose micro-chip CPU Intel 8080** (and later the **Motorola 6502**) made **personal computers** a reality !

- A **personal computer** was ***relatively* cheap** ($5000 for an Apple II)

  (The **popular Apple II** personal computer - 1980)

- **IBM PC** introduced in **1982**:

- **Computers** are more **proliferate**.

  - **People** have **data files** that they like to ***share***

  - **Data (files)** must be **transferred between computers** to be **shared**

- ***Initial* attempt** of **file sharing**:

  - Through **copying** files using **floppy disks**

  The **floppy disk** through the **history**:

  (8 inch, 5 1/4 inch and the 3 inch floppy that does **not** "flop")

- **1980's - 1990's:**

    - **Very cheap** (personal) **computers**

    - **proliferation** of **workstations** and **personal computers** !

        - **Tons** of people want to **share** files...

    - There is a **need** to **share** information (files) **more *easily*** then **copying** to **floppy disks** !!!

        - **Connected** computers and develop **software** to allow **computers** to **exchange *messages*** !!!

        - This is the **start** of **computer networks** !!!!

- **Aptly saying:**

    - **Neccesity** is the **mother** of all **inventions**.....

- **The first computer network**

    - The ***first* computer network** was:

        - the **ARPA net**

          (Advanced Research Project Agency)

      The **ARPANet** built around **1969**.

    - **Motivation:**

        - **ARPA** has sponsored a **number** of **research centers** with **large computing centers (super-computers**

          Problem:

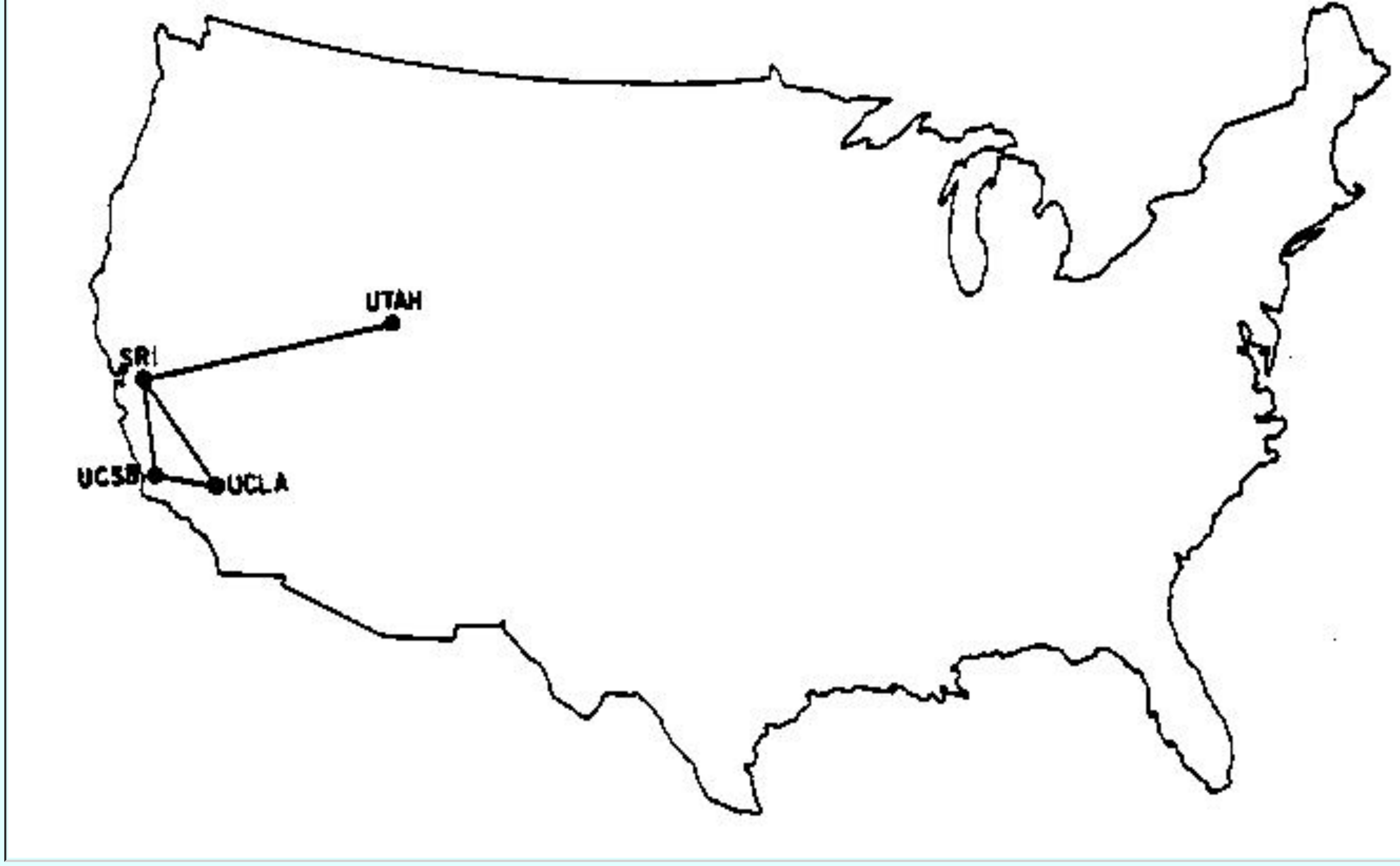
            - How to let **researchers** at **other sites** gain **access** to these **super-computers** ???

        - **ARPA** wanted to create:

            - a **communication network** that allows ***remote* users** to use the **computers** at the **ARPA-sponsored computing centers**

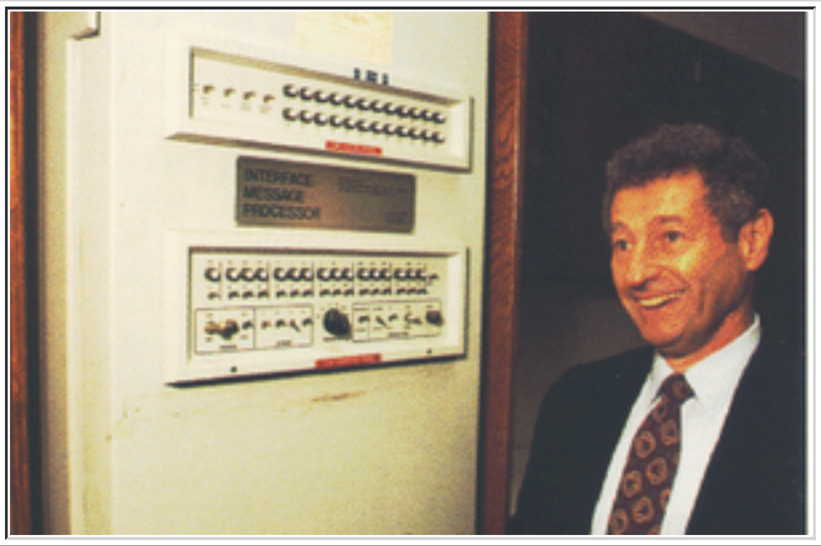
          (**Super-computers** were ***extremely* expensive**)

    - The **initial ARPANET** consisted of **four** nodes:



- **UCLA** (Leonard Kleinrock)
- **Stanford Research Institute**'s Augmentation Research Center (Douglas Engelbart)
- **UC Santa Barbara**
- **University of Utah**'s Computer Science Department

- **Kleinrock** (UCLA) and the first **Interface Message Processor** (**router**) at UCLA:



The Internet

- **Why was the Internet created ???**
    - You **all** know the **Internet**....

    - **Question:**
        - ***Why*** was the **Internet built/created** ????

- **Background info: A *very* important characteristic of *communication networks***
    - **Fact:**
        - A **communication network** is **determined** by the **communication *protocol*** used (by that network)
            - The **protocol** specifies ***how*** to **interpret** certain **information fields** in a **message**

    - **Analogy:**
        - What's the **difference** between **FedEx** and **UPS** ?

Difference:

- Each **carrier** uses its own **way** to **identify** a **package** (i.e., **different *addressing* (= tracking numbers !!)**)
- Each **carrier** uses its own **code** to **identify** the **cities**
- Each **carrier** uses its own **hubs** to **forward** the **packets**
- And so on...

Result:

- A **FedEx** package can **not** be **processed** at a **UPS processing hub**
- And **vice versa** !!!

- What is the ***Internet***:

    - **Internet** = a **network** that uses the **IP protocol** to transmit **messages**
    - **IP protocol** = the ***Internet Protocol*** (click here)

- **Life *before* the Internet**

  - **Fact:**

    - There were **various *different* computer networks** before the **Internet**

      Note:

      - ***Each* type of computer network** uses ***its own* protocol**

      Consequence:

      - **Messages** sent on **one *type* of network** cannot be sent on ***another* type of network** **without** changing the **message format**

  - Some **mile stones** in **computer communication *before*** the creation of the **Internet**:

    - **1969: ARPAnet** (the **first computer network**)

      - ARPAnet connecting: UCLA, UCSB, Unib. of Utah and Stanford Research Institute.
      - ARPAnet does **not** use the **Internet Protocol (IP)** !!!

    - **1971:** first **email** sent on ARPAnet !
      (Yes, there can be **email without** using the **Internet** !)
    - **1971: File Transfer** Protocol (**ftp**) introduced

      Note:

      - **Networking** is ***clearly* useful** (can you live without **email** ??!!)

      - ***Other* people/institutes** now want to be **connected** to **each other**....

      - I think if they have **expanded** the **ARPANet** to connect ***other* computers** in the **world**, you do **not** need to **invent** the ***Internet*****.....**

      - Anyway, ***other* people/institute** now **starts** to make **their *own*** communication **networks**....

        (They do **not** get on the **ARPANet** because they were **not** sponsored by **ARPA**.....)

    - **1978: UUCP** (Unix to Unix copy) **communication protocol** -- mainly used for **software distribution** between UNIX systems.
    - **1980 and beyond:** Other networks are starting to be developed and employed

      - It was the start of **chaos**...
      - I can **remember** I needed to use ***several* types** of **email addresses** to send my **emails**; depending on **where** I wanted to send the **email**....

  - Some **well-known computer networks** created **after** the creation of **ARPANet**:

- **CSNet** (1981) was established to connect computers from various **computer science department** of (poorer) universities - See: click here

    - The **Computer Science Network (CSNET)** was a **computer** network that began operation in 1981 in the United States.[1] Its purpose was to extend networking benefits, for **computer science departments** at academic and research institutions that could **not** be directly **connected to ARPANET**, due to funding or **authorization limitations**.

  So the **reason** why ***other* institutes** start their **own networks** is because they **cannot** get on **ARPANet**.....

- **BITNet** was another alternative to **ARPAnet** (1981) - See: click here

    - **BITNET** was a cooperative **USA university computer network** founded in 1981 by Ira Fuchs at the City University of New York (CUNY) and Greydon Freeman at Yale University.

- **FidoNet** (1984) mostly for Bulletin boards - See: click here

    - **FidoNet** is a worldwide computer network that is used for communication between **bulletin board systems** (BBSs). It uses a store and forward system to exchange private (email) and public (forum) messages between the BBSs in the network, as well as other files and protocols in some cases.

- etc, etc

- **The *problem* of having *many* types of communication networks**

    - **Emerging problem:**

        - Each **type of network** uses a **different communication protocol** - it is like people from different countries speaking different languages.

        - **Computers on different types of networks *cannot* communicate** with one another **without** using some kind of **communication protocol *translation*** software.

            - **Addressing** (identifying a user) was a **pain**: you had **many different** address **formats** !!!
            - **Relaying** a **message** from **one network** to **another network** was a **nightmare** (the more networks there were, the more **translation routines** needed !)

    The **solution**:

    - Get rid of **all** the **networks** that have been **created**

    - **Connect all computers** on ***one* network**:

        - The ***Internet*.....**

      The **new Internet** uses the the **Internet Protocol (IP) created**

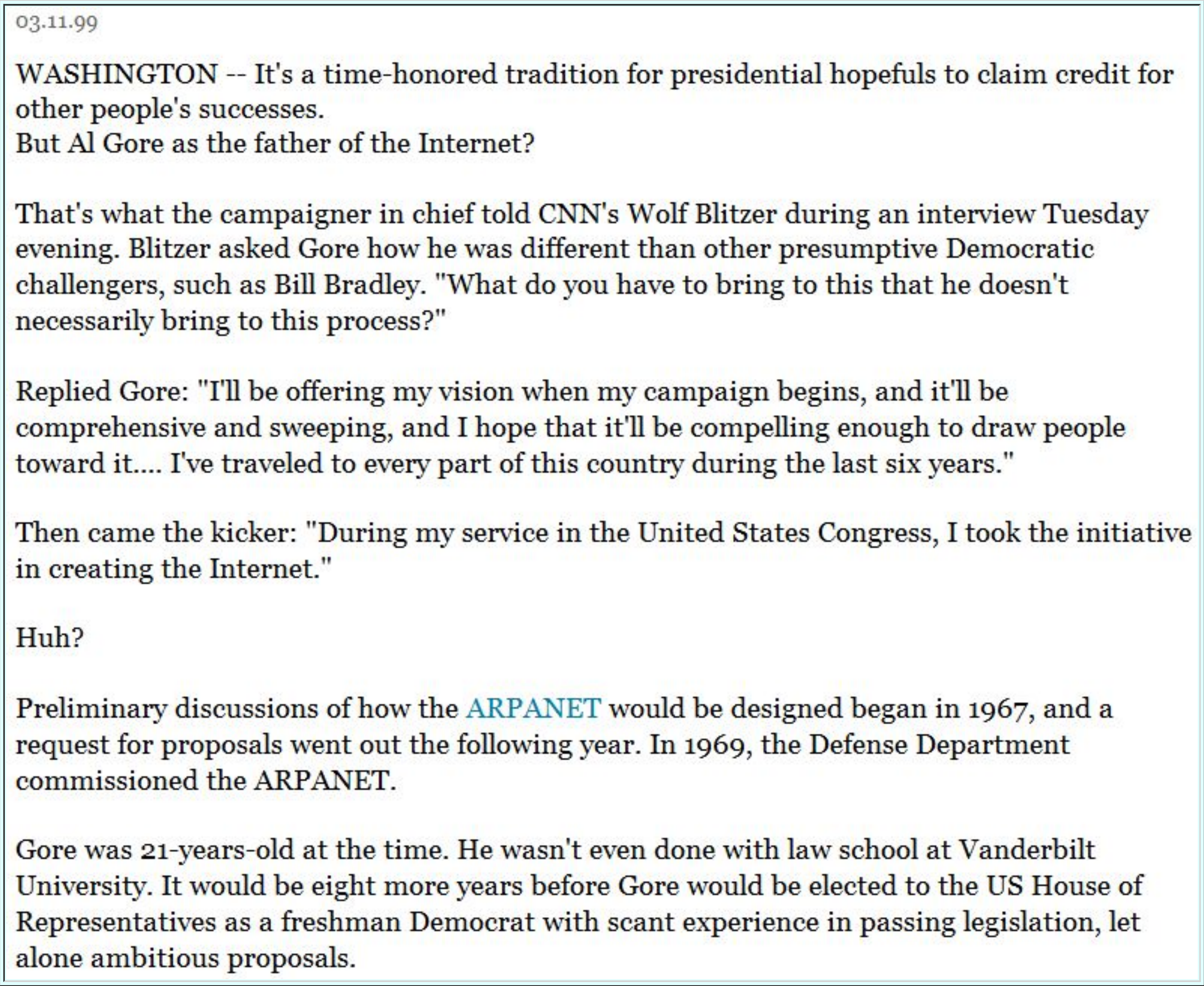
- **Who created the Internet ???**

    - ***Self-proclaimed* father of the Internet: Al Gore** :)

- **Al Gore ???** (LOL :))

Al Gore: "During my service in the United State Congress, **I took the initiative in *creating* the Internet**

Yeah, right:



03.11.99

WASHINGTON -- It's a time-honored tradition for presidential hopefuls to claim credit for other people's successes.
But Al Gore as the father of the Internet?

That's what the campaigner in chief told CNN's Wolf Blitzer during an interview Tuesday evening. Blitzer asked Gore how he was different than other presumptive Democratic challengers, such as Bill Bradley. "What do you have to bring to this that he doesn't necessarily bring to this process?"

Replied Gore: "I'll be offering my vision when my campaign begins, and it'll be comprehensive and sweeping, and I hope that it'll be compelling enough to draw people toward it.... I've traveled to every part of this country during the last six years."

Then came the kicker: "During my service in the United States Congress, I took the initiative in creating the Internet."

Huh?

Preliminary discussions of how the ARPANET would be designed began in 1967, and a request for proposals went out the following year. In 1969, the Defense Department commissioned the ARPANET.

Gore was 21-years-old at the time. He wasn't even done with law school at Vanderbilt University. It would be eight more years before Gore would be elected to the US House of Representatives as a freshman Democrat with scant experience in passing legislation, let alone ambitious proposals.

Full article: click here

(He's the perfect Politician for me to teach **recursion**: do very little work and claim all the glory....)

- The **real creators** of the **Internet**:

  - **Robert Kahn** (click here) and **Vinton Cerf** (click here)

- **Evolution of the Internet (** click here**)**
  - **Different *versions*** of the **Internet**
    - There is ***no* one version** of the **Internet** !!!
      - There are ***many* different versions** of the **Internet** !!!!

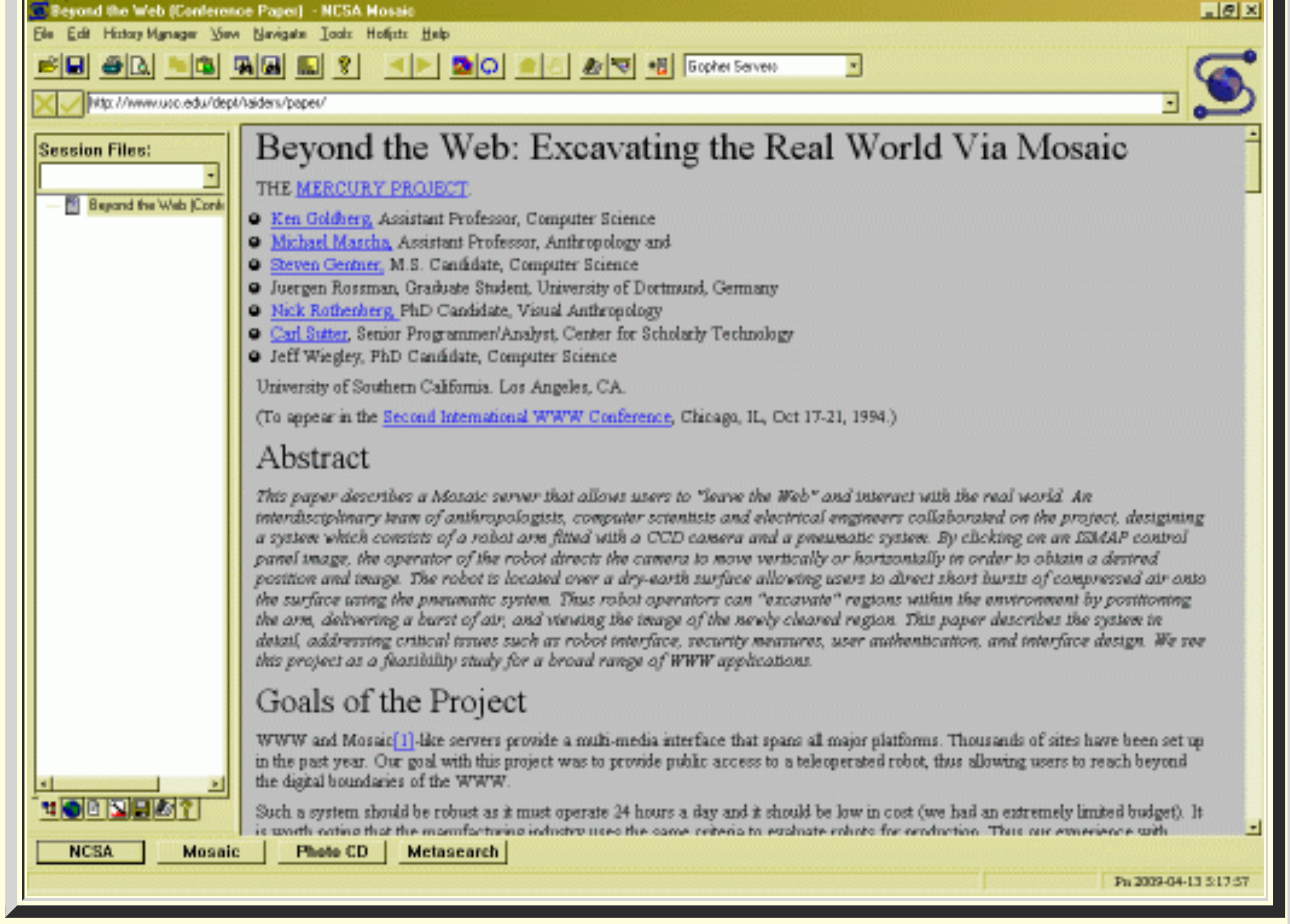
    - The **most popular version** of the **Internet Procotol** used is: **verions #4 (IPv4)** !!!

  - **Development** of **IPv4**:
    - **1974**: document **Specification of Internet "Transmission Control Program"**, by Vinton Cerf, Yogen Dalal and Carl Sunshine, Network Working Group (TCP v1)
    - ??? --- TCP v2,
    - **Spring 1978:** TCP/IP version 3 - splits the original protocol into **2 protocols**: **TCP v3** and **IP v3**
    - **September 1981:** the **IP standard** (**IPv4** - **Internet Protocol version 4**)
    - **January 1, 1983**: **flag day** --- all protocols replaced by **IPv4** !!!

  - The **calm** before the **storm**:
    - For a **long time**, the **Internet** was the **best kept secret** of the academia...
    - The **Internet** was mainly used for:
      - **email**
      - **news** (USENET)

      It was **quiet** and **peaceful** on the **Internet** - the **good old days :-)**...

  - This **killer application** put the **Internet** in the public spot light....
    - **1993: Mosaic**



What is **Mosaic**:

- **Mosaic** was the **grand-daddy** of **Netscape** (or the present day **Mozilla firfox**)...

  See: click here

- With Mosaic, a user can share the information stored in his website easily with anyone on the Internet

- The **World Wide Web (WWW)** was born....

- **Mosaic companion:**

  - But there is still a **problem**: unless you **tell** someone about your website, **nobody can find it**...
  - **1994: Webcrawler** - See: click here
    - **Webcrawler** was the ***first* Web search engine** - you can say it's the grand-grand-grand-daddy of **Yahoo** and **Google** and other web page search engines...
    - With **Webcrawler**, a user can **find** the relevant information !

The rest is **history**....

Soon ***everybody*** wanted a **website**....

Problem:

- The **Internet** start to **run out** of **Internet *Addresses***

  (It's like the **telephone company** running out of **xxx-xxx-xxxx telephone numbers** !!!)

- **1994: IPv6** - **Internet Protocol version 6** (See: click here)

  - As with any **good thing**, the **Internet** was a **victim** of its own **success**
  - **Each computer** on the **Internet** requires a **unique IP address** (like your own **telephone number**)
  - **IPv4** has 2^{32} **= 4294967296** numbers...

    The world is running out of **IPv4 addresses**

  - **New Internet Protocol: IPv6** supports 2^{128} **= 340282366920938463463374607431768211456** addresses... Oh, yeah, baby :-)

Unfortunately, changing over to **IPv6** is **not easy** and **still has not happened** in **2014**....

- We found many ***ad hoc* fixes** to **limit** the **number of IPv4** addresses used
- The **US** still uses **IPv4**
- **Many other countries** have switched to **IPv6**

- **IPv4** and **IPv6**:

  - Today, the **Internet** uses a **mix** of **IPv4** and **IPv6** protocols.

  - There are **many ways** to allow **IPv4** and **IPv6** to **"co-exist"**

    Article: click here

    (The **techniques** mentioned in the **article** will be discussed in this **course** - but **not** specifically in relationship with **IPv4** and **IPv6**)

Stroing **information**

- ***Storing*** **information**
  - **Question:**
    - **How** do you ***store*** **information** ?????

  - As you **recall** from your **high school *Physics*** class:
    - There are **2 kinds** of **stuff** in the **universe**:
      1. **Matter**
      2. ***Energy***

    So **how** can you ***store*** **information** using these ***stuff*** ?????

- **Storing information**
  - **Information** can be **stored**:
    - Using **matter**

      Example:

      Explanation:

      - When you ***write*** some **information** on **paper** (using a **pen** or pencil), the **information** is stored in **matter**

- Using **energy**

  Example:

  Explanation:

  - When you **"write"** some **information** by **tapping/flashing** the **encoding** in **Morse code**, the **information** is stored in the **sound/light wave** (which is a form of **energy**)

- **Interesting question: is *information* itself made of by matter or energy ?**
  - As you have been **thought** in **High School** in **Physics**:
    - ***All* things** in this world is either **matter** or **energy**
      - (**Matter** can be **coverted** into **energy** and **vice versa**)
  - **$64,000 question:**
    - Is **information**

- a **form** of **matter** **or**
- a **form** of **energy** ???

Or **both** ???

Answer: ***neither***

- **This** is **information**:

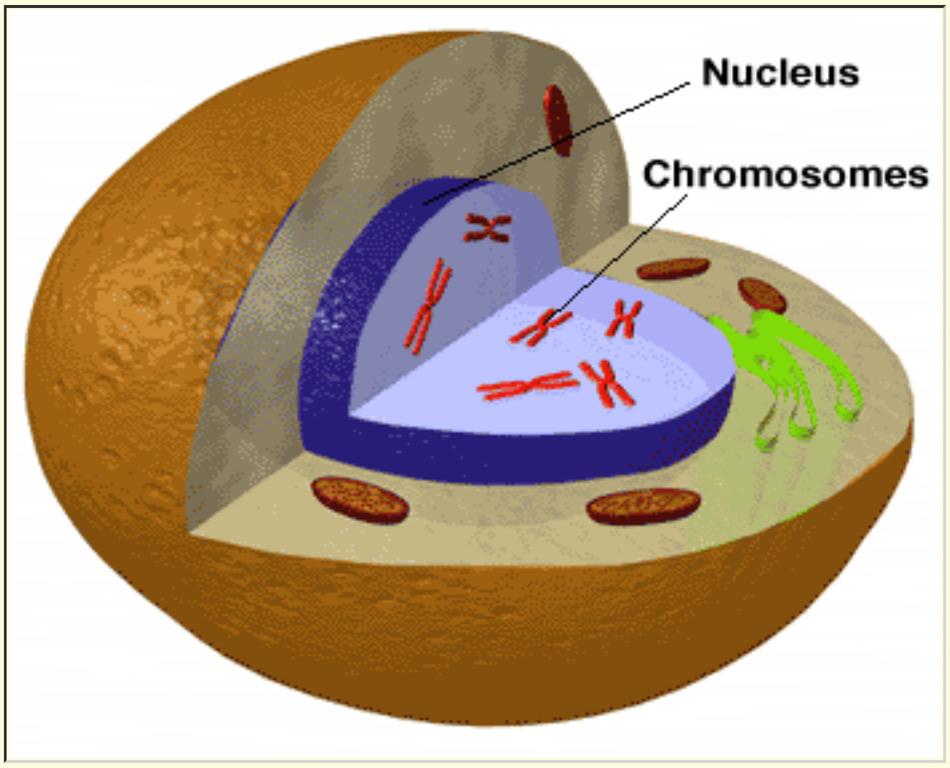
- **But** ***this*** is ***not*** **information**: (it's ***gibberish*** click here)

Observation:

- **Information** is ***not*** the **matter** (or **energy**) ***itself*** !!!
- **Information** is **represented** by the ***arrangment*** of **matter** (or **energy**) !!!!

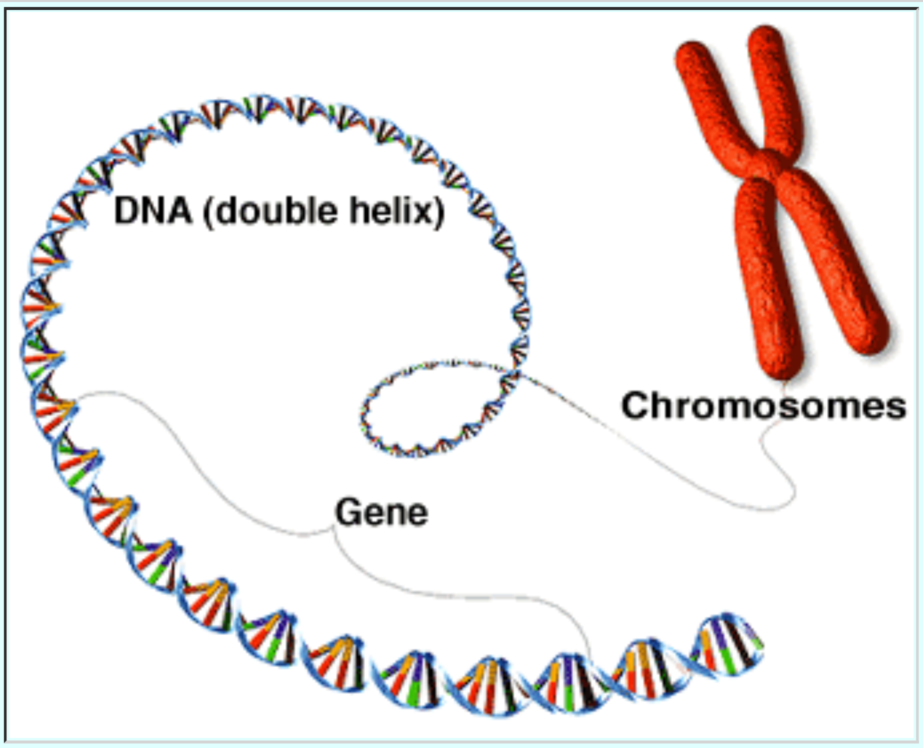
Information stored in DNA

- **Genetic *Information***
  - **Cell:**
    - A **cell** contains (among other things) a **nucleus**:



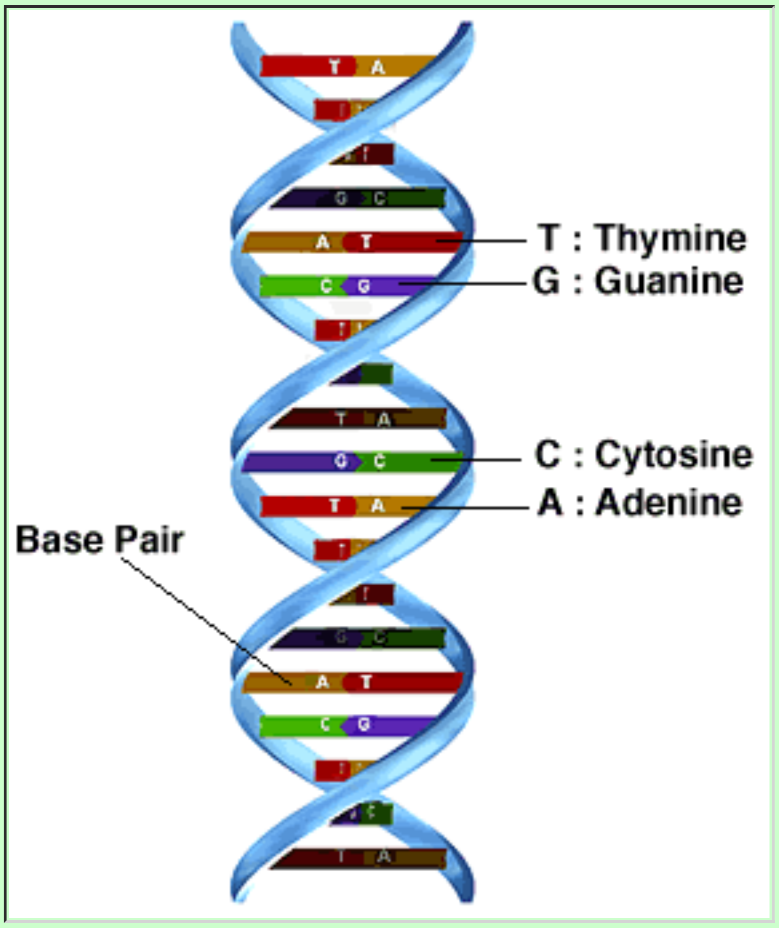
Inside the **nucleus**, there are a number of ***chromosomes***

  - **Chromosomes**:
    - Each **chromosome** consists of a (long strand) **DNA** molecule:



- **DNA**:

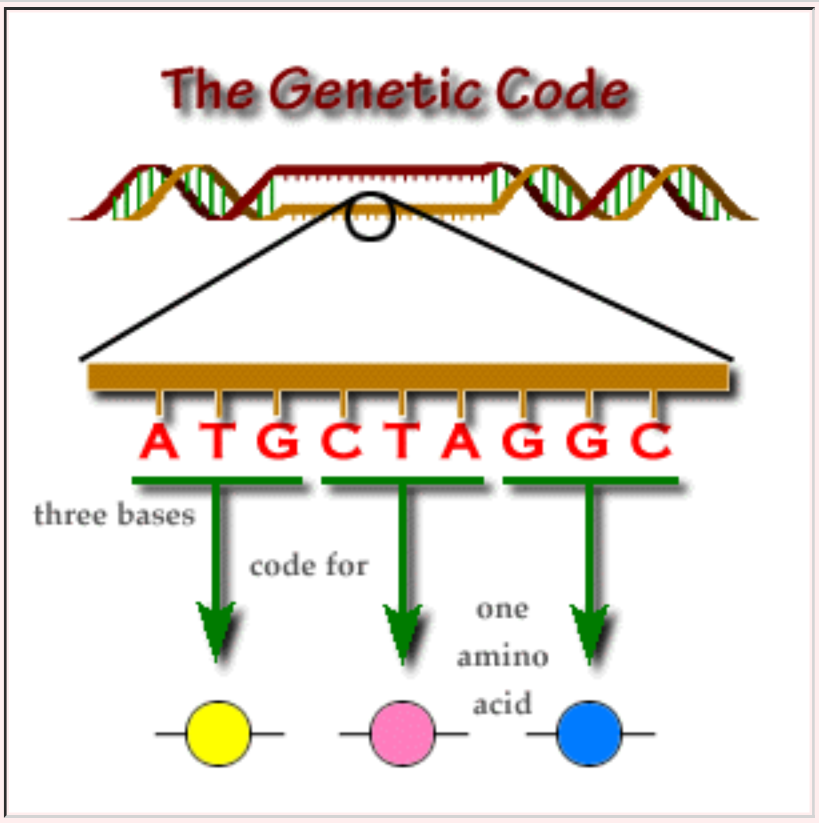
  - Each **DNA** molecule is **constructed** with **4 *different*** molecules:

    These **molecules** are called ***bases***

- **Encoding genetic information**:

  - We have **discovered** that **each *combination*** of ***3* bases**, **identifies** a ***unique*** **amino acid molecule**:

- The **genetic** ***code*** of **all living being**:

Second letter

First letter	U	C	A	G	
U	UUU UUC Phenyl-alanine UUA UUG Leucine	UCU UCC UCA UCG Serine	UAU UAC Tyrosine UAA Stop codon UAG Stop codon	UGU UGC Cysteine UGA Stop codon UGG Tryptophan	U C A G
C	CUU CUC CUA CUG Leucine	CCU CCC CCA CCG Proline	CAU CAC Histidine CAA CAG Glutamine	CGU CGC CGA CGG Arginine	U C A G
A	AUU AUC AUA Isoleucine AUG Methionine; initiation codon	ACU ACC ACA ACG Threonine	AAU AAC Asparagine AAA AAG Lysine	AGU AGC Serine AGA AGG Arginine	U C A G
G	GUU GUC GUA GUG Valine	GCU GCC GCA GCG Alanine	GAU GAC Aspartic acid GAA GAG Glutamic acid	GGU GGC GGA GGG Glycine	U C A G

- **How is the genetic information *used* ?**
  - **Fact:**

    - The **human body** is **made up** of ***protein*** and ***other*** **moluecules**

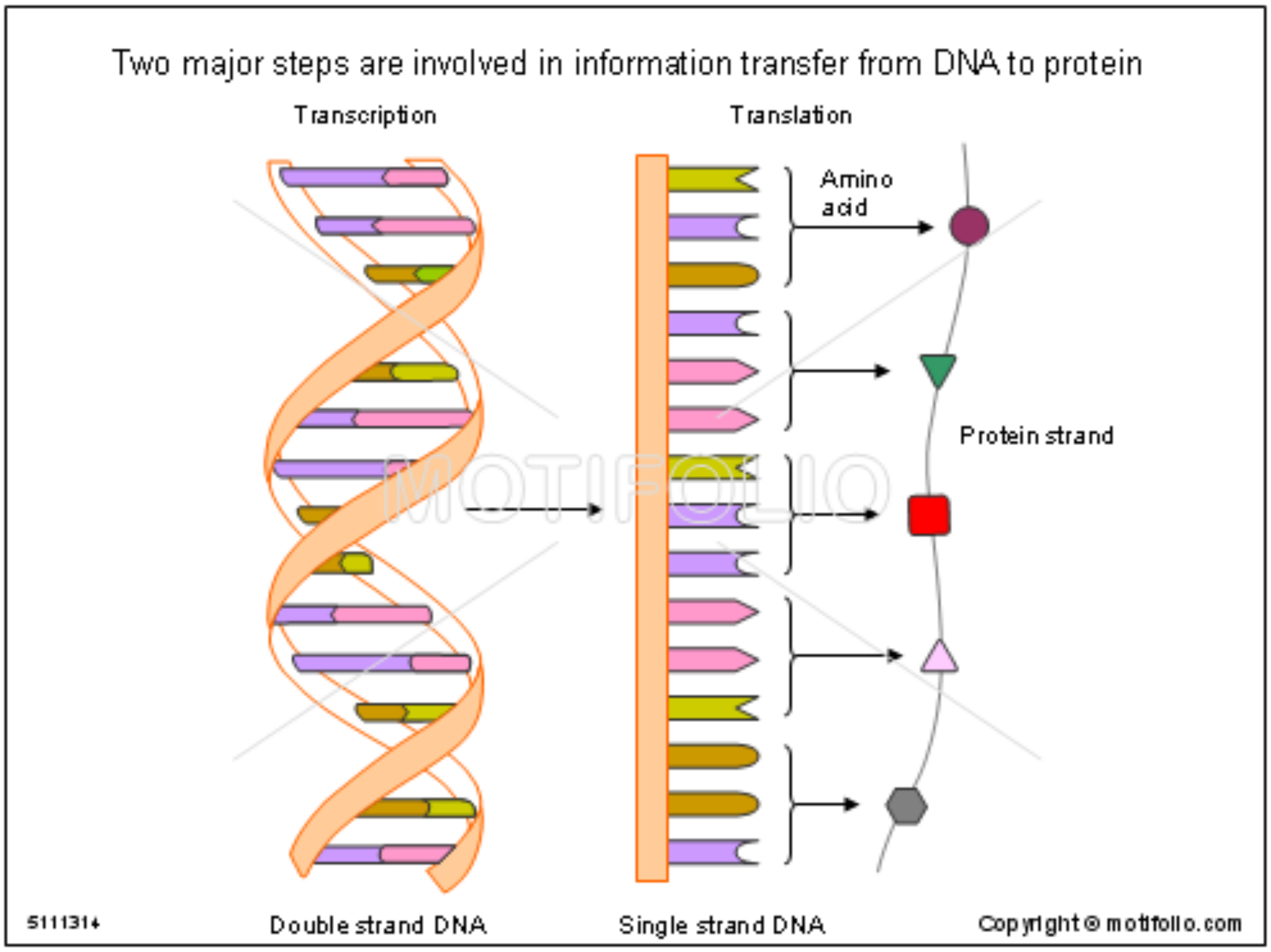
    Examples:

    - You **eyes** is ***completely*** made up by ***protein*** **molecules**

      (So are **most** of your ***organs***)

  - **Use** of your **genetic information** by you **body**:

    - Your **DNA** determines the **proteins** that your **body** will **make**:



Example:

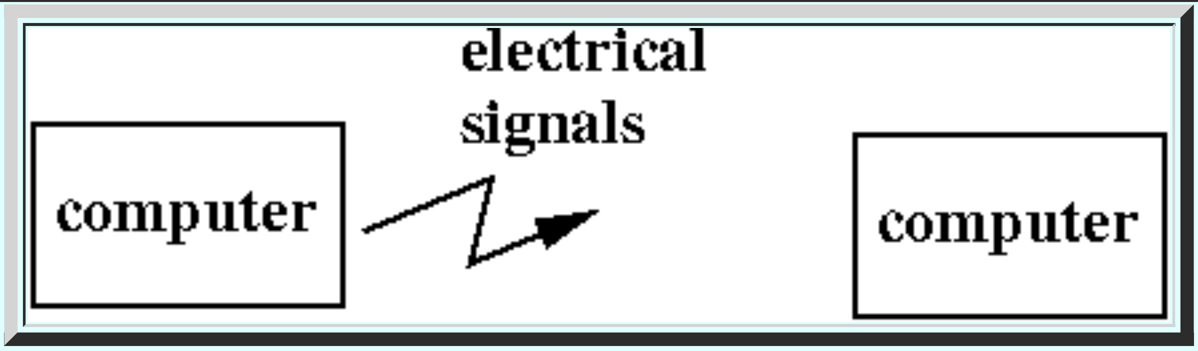
- So **if** you have ***blue* eyes**, then ***your* gene** will **contain genetic *information*** on **how to make "blue" pigmented eyes** !!!!

- So **if** you have ***brown* eyes**, then ***your* gene** will **contain genetic *information*** on **how to make "brown" pigmented eyes** !!!!

- **And so on...**

Infra-structure to relay electrical signals

- **How computers communicate**
  - **Communication** between **computers**:



- The **sender** ***transmits*** **electrical signals** that form **"words"** in some **"language"**....
- The **receiver** ***receives*** the **electrical signals** and **interprete** the **"words"**....

- **Infra-structure for *electrical* communication (What is needed to make communication possible)**
  - **Communication** **infra-structure** for **electrical** communication:

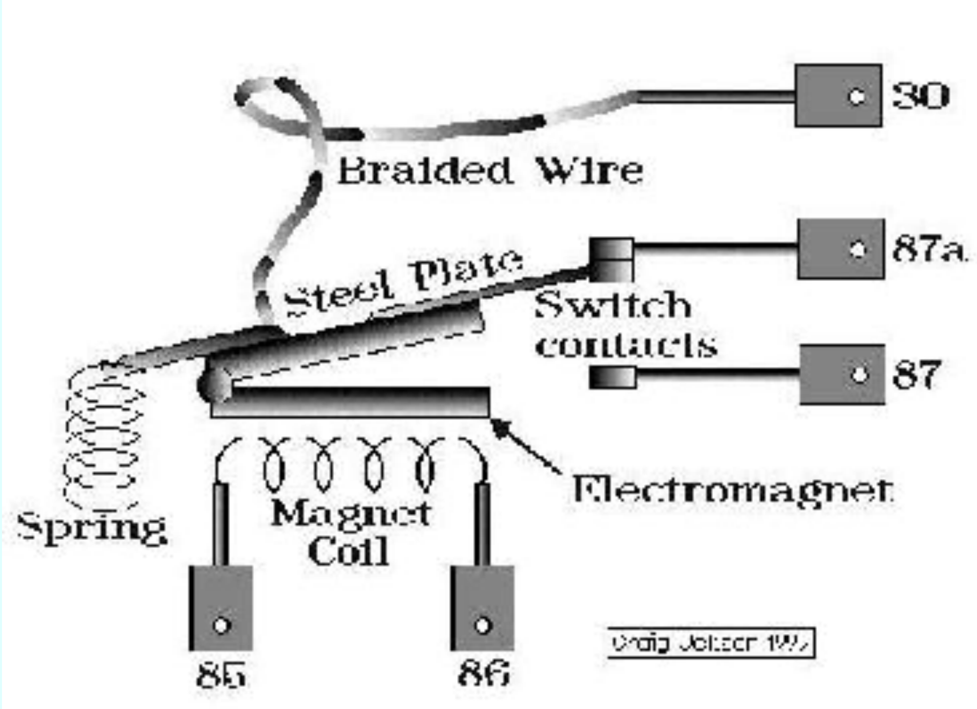
1. **Transmission scheme**:

   - We need a **mechanism** to **convert** the **message** into ***electrical* signals**

     (So that we can **transmit** the ***electrical* signals**

   - We **also** need a **mechanism** to **convert** the ***electrical* signal** **back** into something **humans** can **interpret**

   Example: a **simple** **touch** circuit transmitter



---

2. **Communication link**:

   - We need a **medium** to ***carry*** the ***electrical* signal** to the **destination**

   Example: copper wires

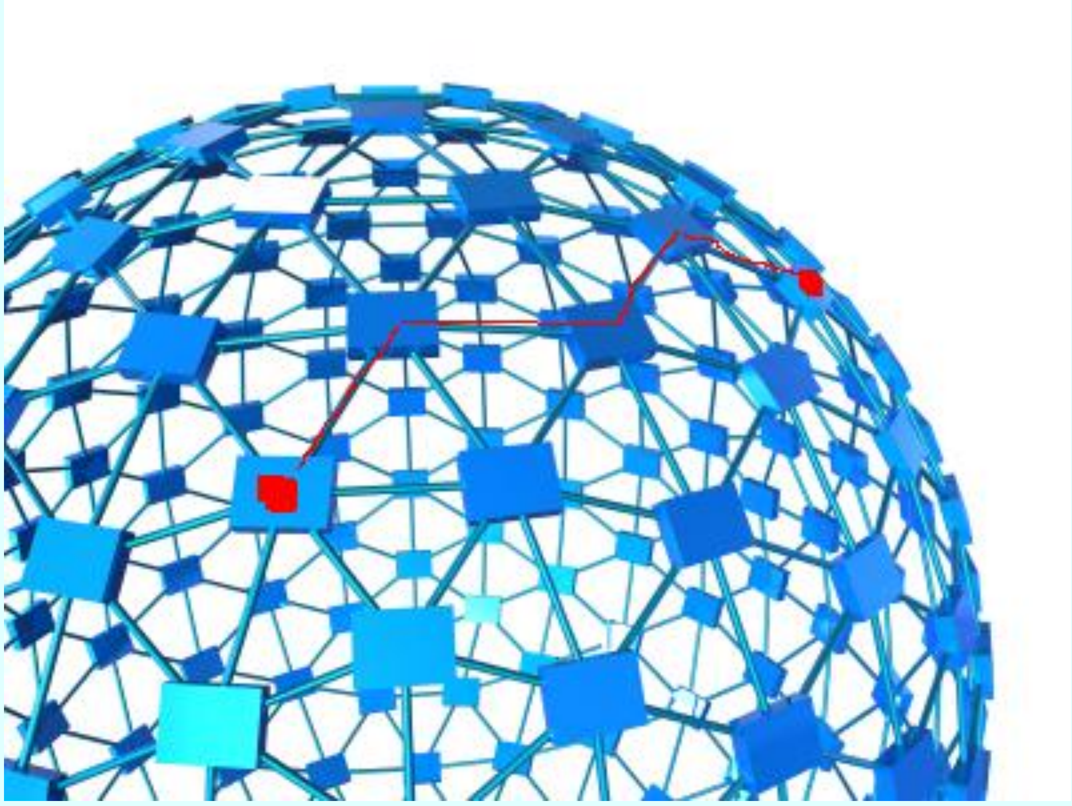
   

---

3. **Routing scheme**:

   - Because the **commincation network** is like a ***maze***, we need a **system** to **find** a ***route*** through the **communication network**

   Graphically:



(Two devices have been developed that can perform routing: **routers** (high level) and **switches** (low level))

- **Terminology**
  - **Host:**

    - **Host** = a **sender** or **receiver**

      I.e.: host = a **source** or a **destination** of **information**

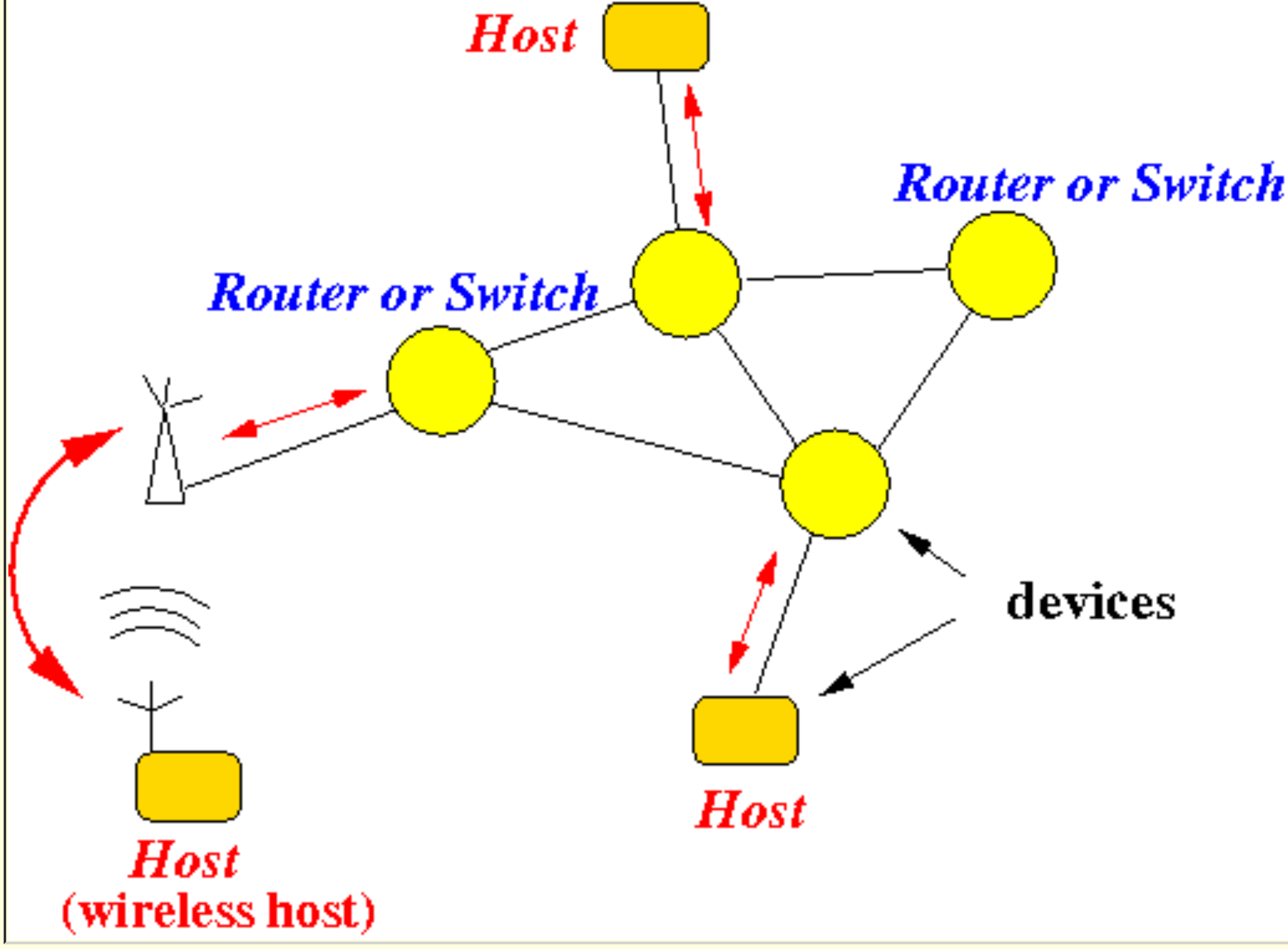
  - **Router/switch:**

    - **Router/switch** = a **device** that **relay** electrical signal (**towards** the **destination host**)

      (A **riuter/switch** is **not** a **source/destination** of **information**

- ***Infra-structure* to *relay* the electrical signals**
  - In order to **communicate** over ***large* distances**, we **build** a **communication *relay* infra-structure**:

    - The **communication *relay* infra-structure** (= **communication network**) **relays messages** from one **(source) host** to ***another* (destination) host**:



Explanation:

- There are **2 kinds** of nodes in a **communication network**:
  - **Hosts**: the **sending/receiving devices (nodes)**
  - **Routers/switches**: **devices** that ***relay*** the **messages** towards **destination nodes**

- **Router/switches** --- **Internal/intermediate devices**
  - **Routers/switches *relay* electrical signals**
  - The **routers/switches** do ***not*** need to **decode** the **messages**

    Result:

    - **Routers/switches** do **not** need to be very **"*intelligent*"**

      (I.e., **routers/switches** do **not** have a lot of different kinds of **software**)

- **Hosts** --- **end/user devices**

  - **Hosts** are **sources** and **destinations** of **messages**.

  - A **destination node** is the **final node** in the **transmission path**

  - The **message** received is ***used*** by a **destination node**

    Consequently:

    - A **destination** must **decode** a **messages**

      (**I.e.,** a **destination node** has a **high degree** of **intelligence** (= or ***more*** **software**)

- **More info on hosts and routers/switches**

  - **Routers/switches**:

    - **Routers/switches** are **specialized devices**:

      - **Routers/switches** can ***only*** perform the **routing operation** (= a **specilization**)

    - **Routers/switches** are **simple devices** (= **less complex**)

      - **Because** they ***only*** perform ***one*** **task** !!!

      Example routers/switches:

      - The **Ethernet switch** you buy for **$50 or so**

        (More **powerful switches** can cost **a lot of money**...)

  - **Hosts**:

    - **Host nodes** are **more *complex*** (= more **expensive**) because:

      - **Hosts** perform **compututaion** for the **user** !!!

Example of host nodes:

- Your **laptop** (**end user** equipment !!!)
- Your **cell phone**
- MathCS department's **servers**
- And so on....

Comment:

- Since **hosts** are **general purpose computers**, a **host** can ***also*** be **programmed** to **perform *routing* function** !!!

Encoding information in a computer (using *binary* numbers)

- **Representing *information* in a computer**
  - **Information** stored ***inside*** a **computer** must be ***represented*** using:

    - **Binary numbers** !!! (or "binary signals")

      (That's **because** the **memory** of a **computer** can ***only*** store ***binary* digits**)

  - **Example** of ***encoding* methods** that uses **Binary numbers**:

    - The **2's complement *encoding*** method (which you learned in **CS255**):

      ```
         Code          Value represented by code
      ============================================
       11111111                    -1
       00000000                     0
       00000001                     1
       ....
      ```

    - **ASCII code**:

      ```
         Code          Character represented by code
      ==============================================
       01100001                    'a'
       01100010                    'b'
       ....
      ```

      Alpha-numerical information inside the computer is **represented** using the **ASCII code**

  - **Very important condition:**

    - To **facility exchange** of ***information***:

      - ***Every* computer** must **use** the ***same* encoding** !!!

      Example:

      - `01100001` **must** represent the **character** `a` in ***every* computer** !!!

- **International encoding standards**
  - **(International) encoding standard** for **languages** using the ***English* alphabet**:

    - **American Standard Code for Information Interchange** (**ASCII**) code

    The **ASCII code** table:

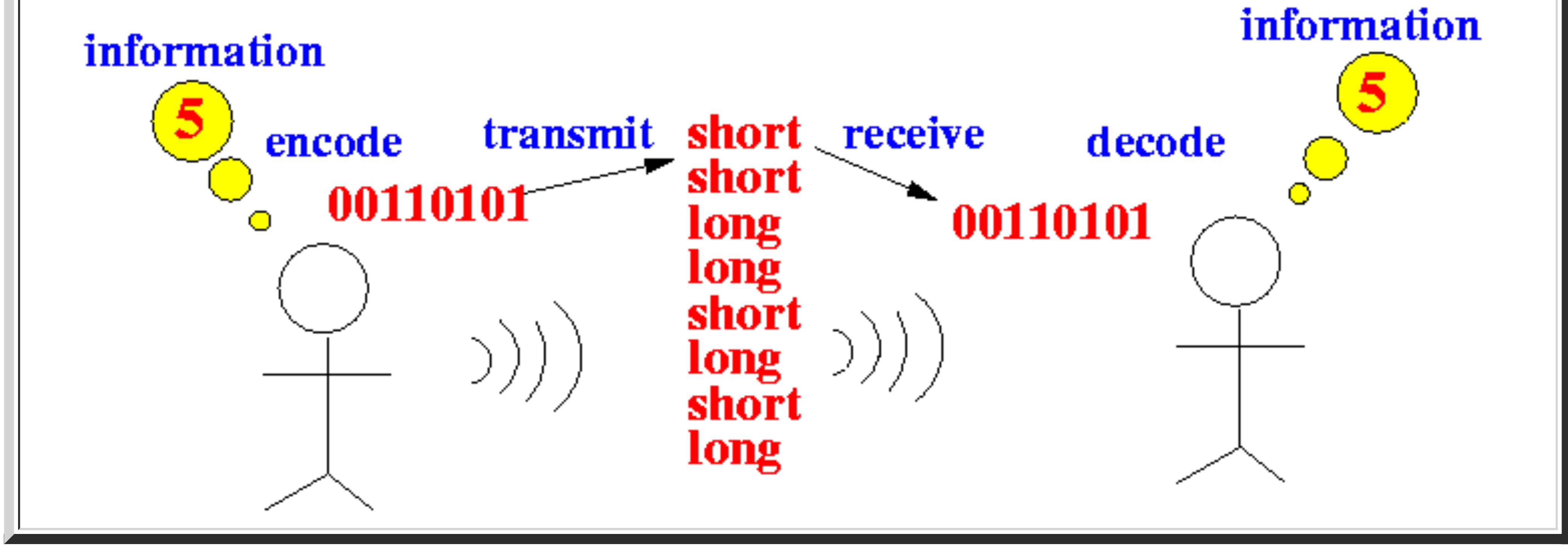
Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

- **International code** for ***all* languages**:

  - **Unicode** !!!

    More on the **Unicode**: click here

- **Summary: how *computers* communicate**
  - **Graphically:**


information
5
encode
00110101
transmit
short
short
long
long
short
long
short
long
receive
00110101
decode
information
5

Introduction: Types of computer networks

- **Types of *computer* networks**
  - **Broadcast networks:**
    - **signal transmitted** by an end-user equipment (**host**) is **received** by ***all* hosts**

    Examples:

    - **Radio, TV, Walkie-talkie...**
    - **Wireless networks**
    - **Ethernet** (with Ethernet hubs)

    Strengths and weaknesses:

    - **Cheap**
    - **Cannot** accommodate a ***large* number** of **senders** (***interference*** !!!)

  - **Switched networks**
    - **signal transmitted** by an **end-user equipment** (**host**) is **received** by ***only*** by the ***intended* destination**

    Examples:

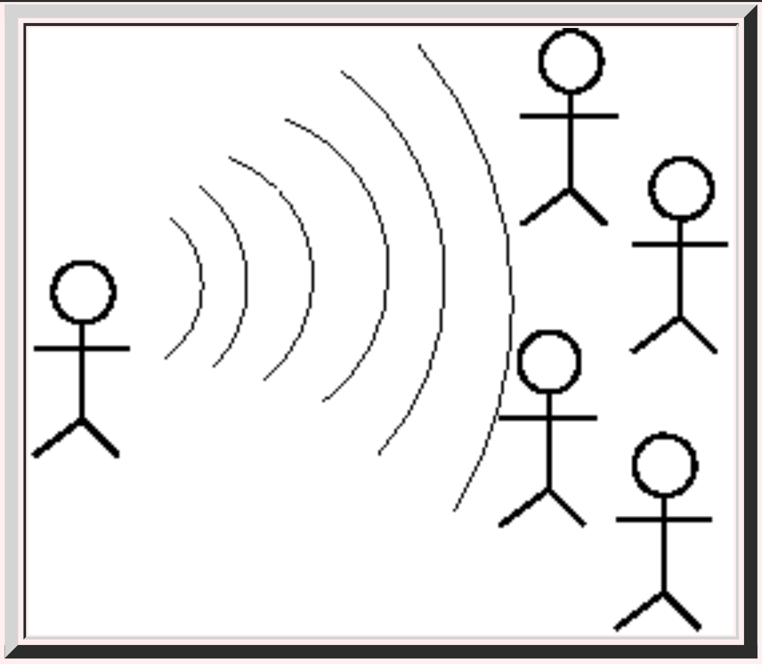
    - **Telephone**

    Strengths and weaknesses:

    - **Expensive** (telephone network is very expensive to build !)
    - Is **capable** to accommodate a ***large* number** of **senders** (millions of people can use the telephone network **simultaneously**)

Broadcast type of computer networks

- **Broadcasting**
  - **Broadcasting:**



  - **Broadcasting** = **transmission** is **received** by ***all*** **receivers**

- **Types of *broadcast* (computer) networks**
  - There are **2 types** of **broadcast networks**:
    - ***Wireless* (broadcast)** networks

      Examples:

      - **Aloha** (first wireless network)
      - **IEEE 802.11** networks (popularly knowns as **"Wifi"** networks)
      - **BlueTooth**

    - ***Wired* broadcast** networks

      Example:

      - **Ethernet** (popular in the **US**)
      - **Token ring** (popular in **Europe**)
      - **Token *bus***

- **Broadcast Network Example: the Ethernet**
  - **Different *varieties*** of **Ethernets**: (they all use the ***same* protocol**)
    - **10-Base 2** (10 Mb/s --- a.k.a. "Thin" Ethernet --- uses a **coaxial cable**)

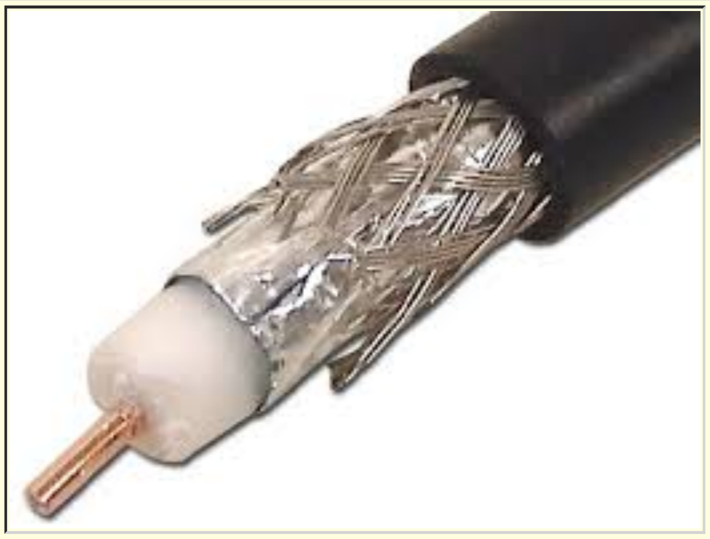
- **10-Base T** (10 Mb/s Ethernet --- uses a **CAT5 cable**)
- **100-Base T** (100 Mb/s Ethernet or "Fast Ethernet" --- click here))
- **1000-Base T** (1 GBs Ethernet or "Gigabit Ethernet" --- click here)

- **Connectors** used by **Ethernet**:

The **left connector** is a **CAT5 cable** connector.
The **right connector** is a **Coaxial cable** connector.

- **Cables** used in **Ethernet**:

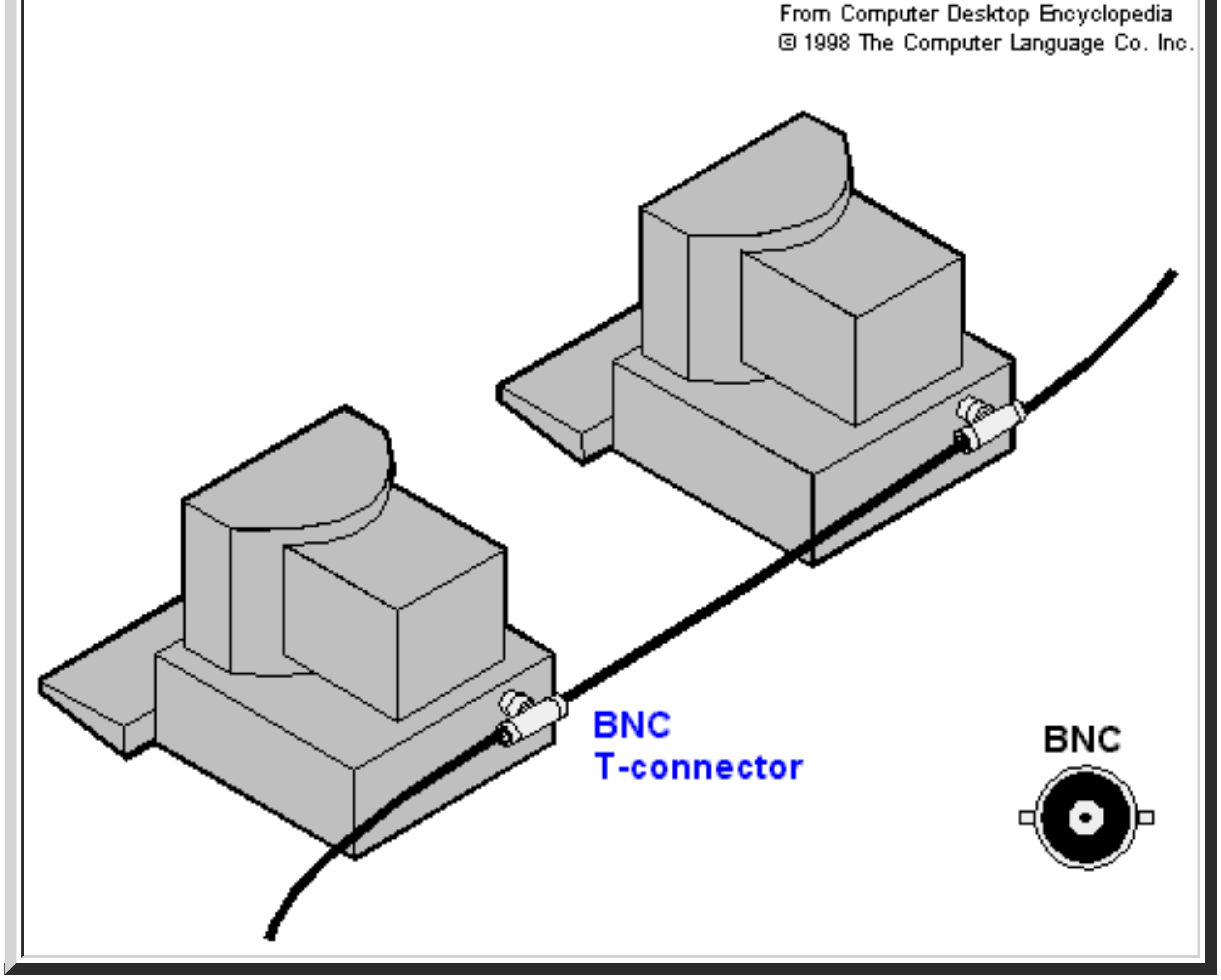
  - **Coaxial** cable:



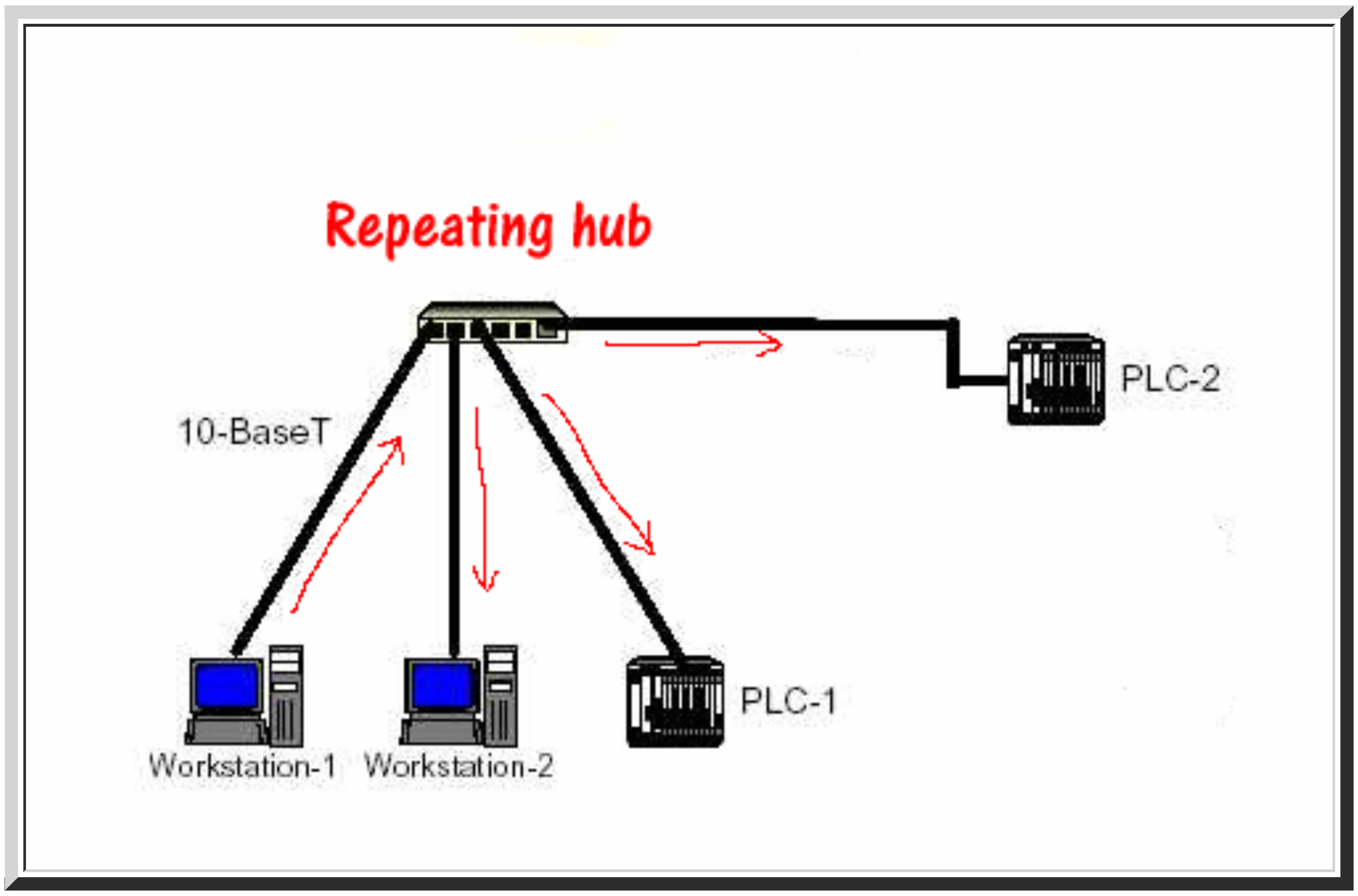
  - **CAT5 (Category 5)** cable:

Wikipedia page: click here

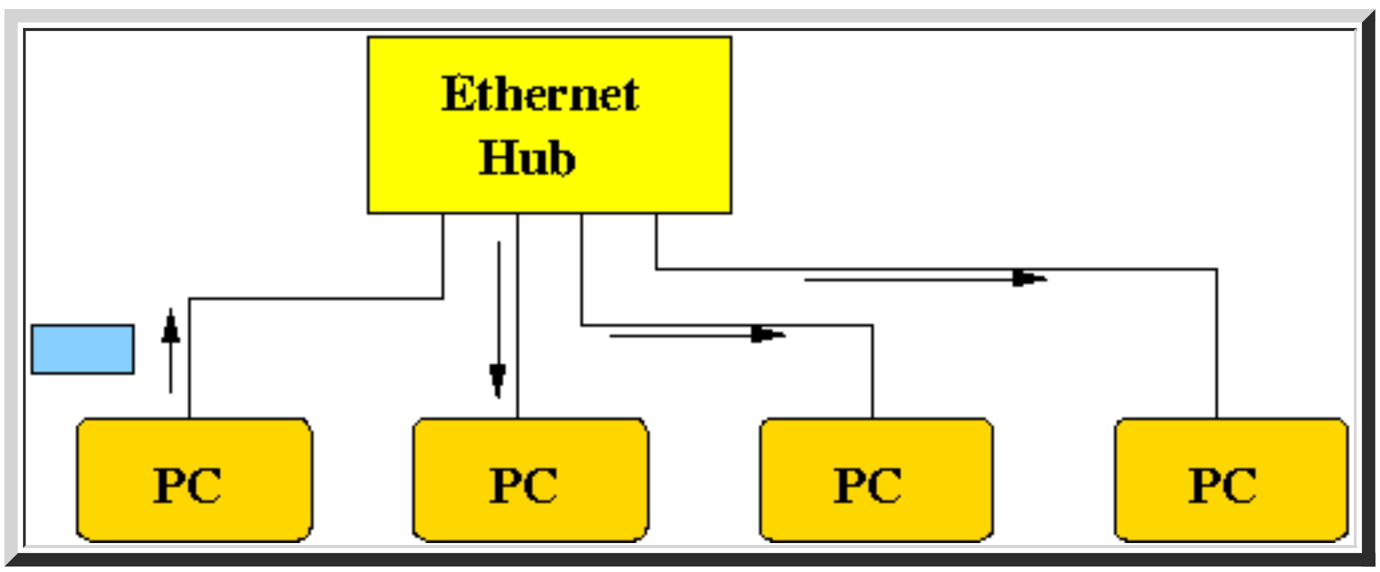
- **Ancient way** to connect **Ethernet** --- using **"Thin" Ethernet:** (**coaxial cable**, a.k.a.: **10Base2 Ethernet**)



- **Modern Architecture of an Ethernet:** (uses a **Ethernet hub**)



The **Ethernet (repeating) hub** will **repeat** an ***incoming*** **message** on some port on ***all* other ports**:



(The **arrows** represent the **transmission forwarding**)

- **Broadcast Network Advantages/Disadvantages**
  - ***Advantage*:**
    - **Simple (cheap) hardware**
      - The **"Thin" Ethernet** uses a **coaxial *cable*** to **connect** the **computers** !!!

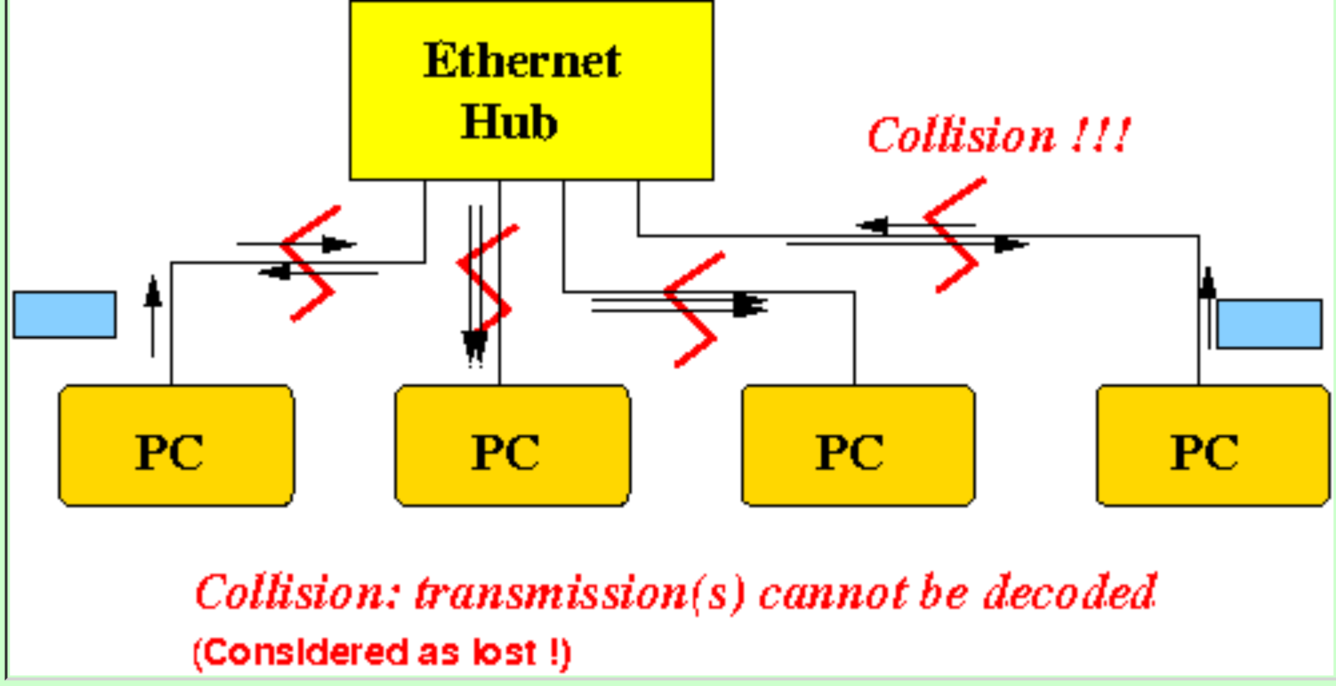
    - **No routing** (= **finding** a **route/path** to the **destination**)
      - **Because** every **message transmissions** will reach ***all* computers** !!!
  - ***Disvantages*:**
    - **Excessive *processing* overhead**:
      - A **message transmission** is **received** by ***every* Ethernet hosts (computer)**
        - **Including** the **hosts** that are **not** the **destination** of the **message** !!!!

      - ***Every* host** must **determine** whether the **message** was **destined** for the **host** (or not)

        Result:

        - **excessive *processing* overhead**

    - ***Not* Scalable**
      - When **2 or more** computers **transmit** at the ***same* time**, the **transmissions** will be ***corrupted***:



- **Result:**

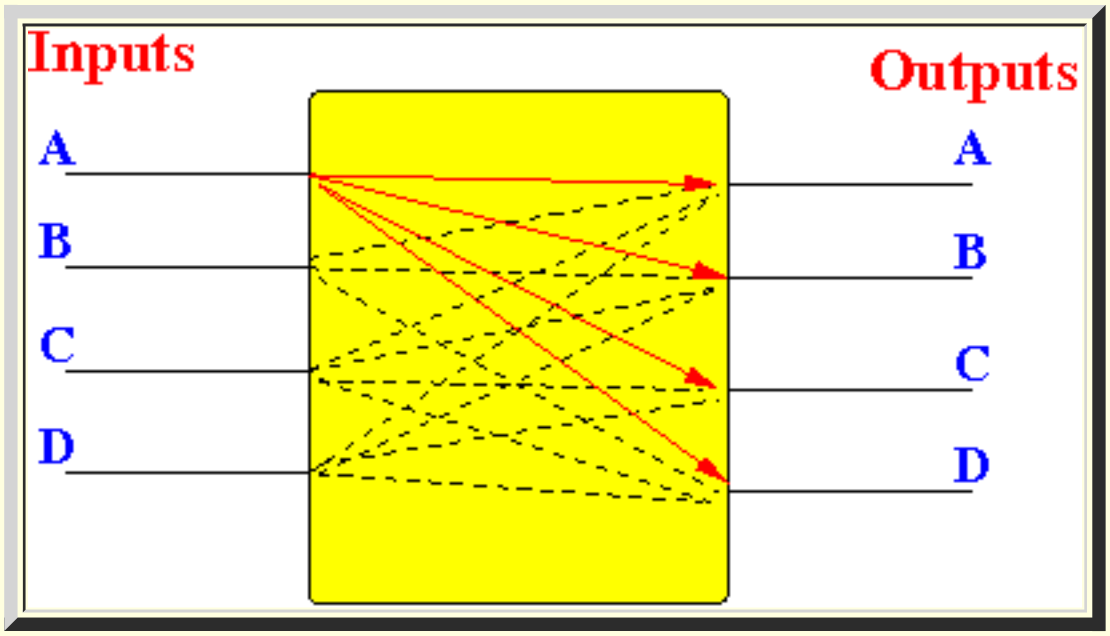
  - **Broadcast networks** (such as **Ethernet**) can ***only*** support a ***limited* number** of **computers**

    - **Because** if you have a **large number** of computers, the probability that **2 or more** of them will **transmit** at the ***same* time** is **increased**.

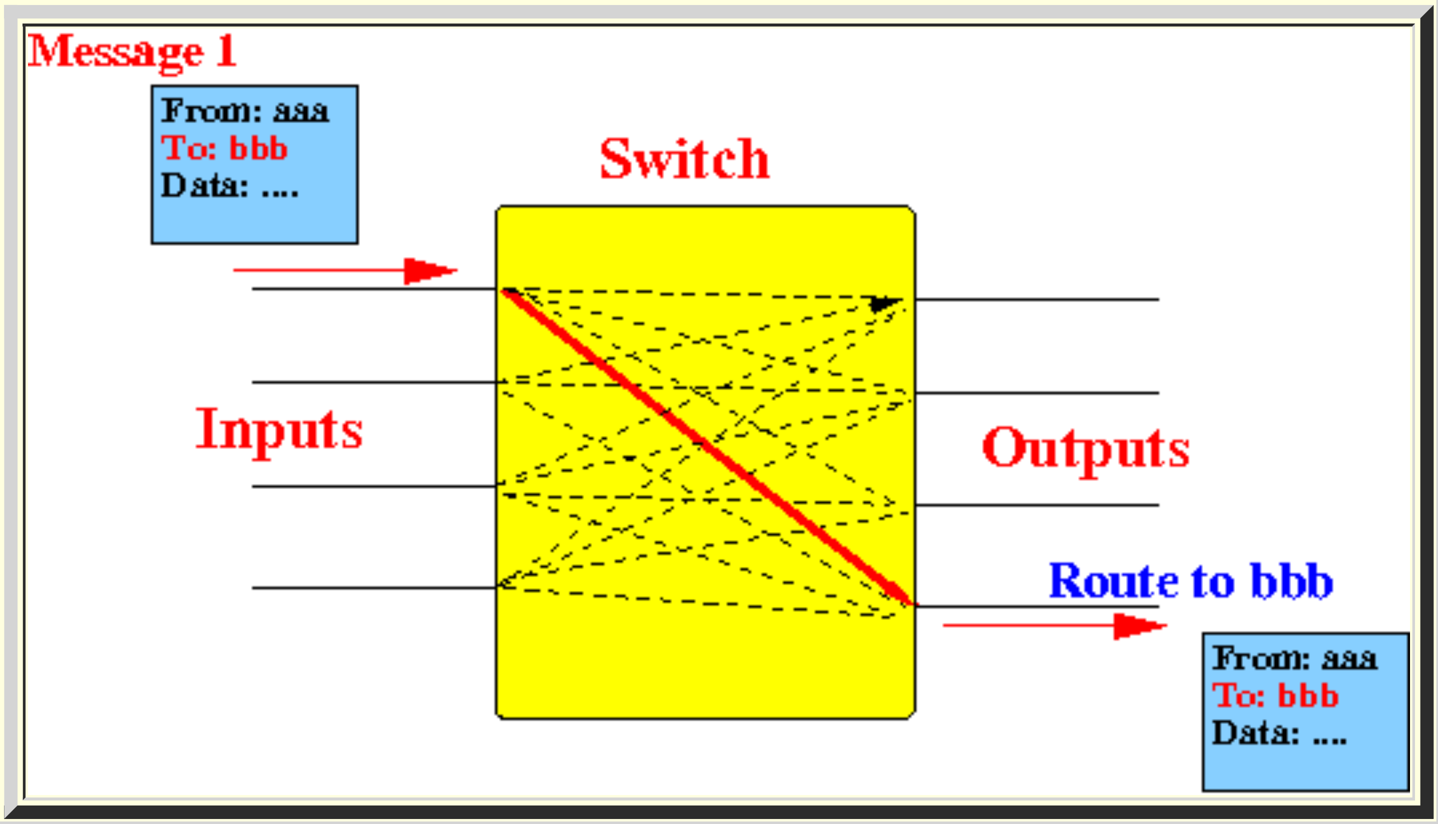
  (Max **~500 hosts** on one **Ethernet** network)

Switched type computer networks

- **Switch**
  - **Switch:**
    - A **switch** has ***N*** **inputs** and ***N*** **outputs**:



The **data** coming into an ***input*** can be **forwarded** onto ***one*** **(or more)** **outputs**:

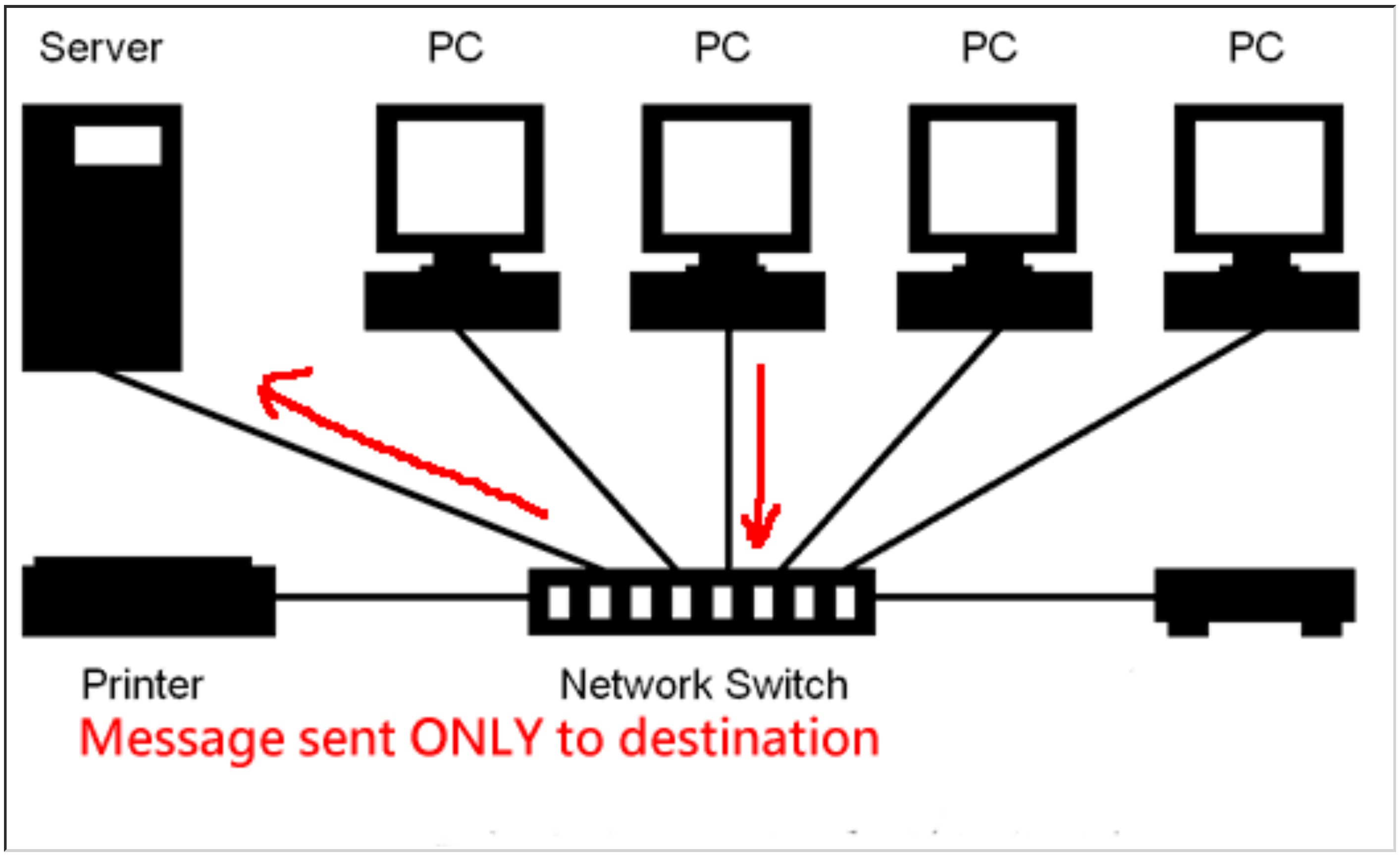


- **Switched Networks**
  - **Fact:**

- **Switched network** is always ***wired***

  (because wireless transmissions cannot support switching)

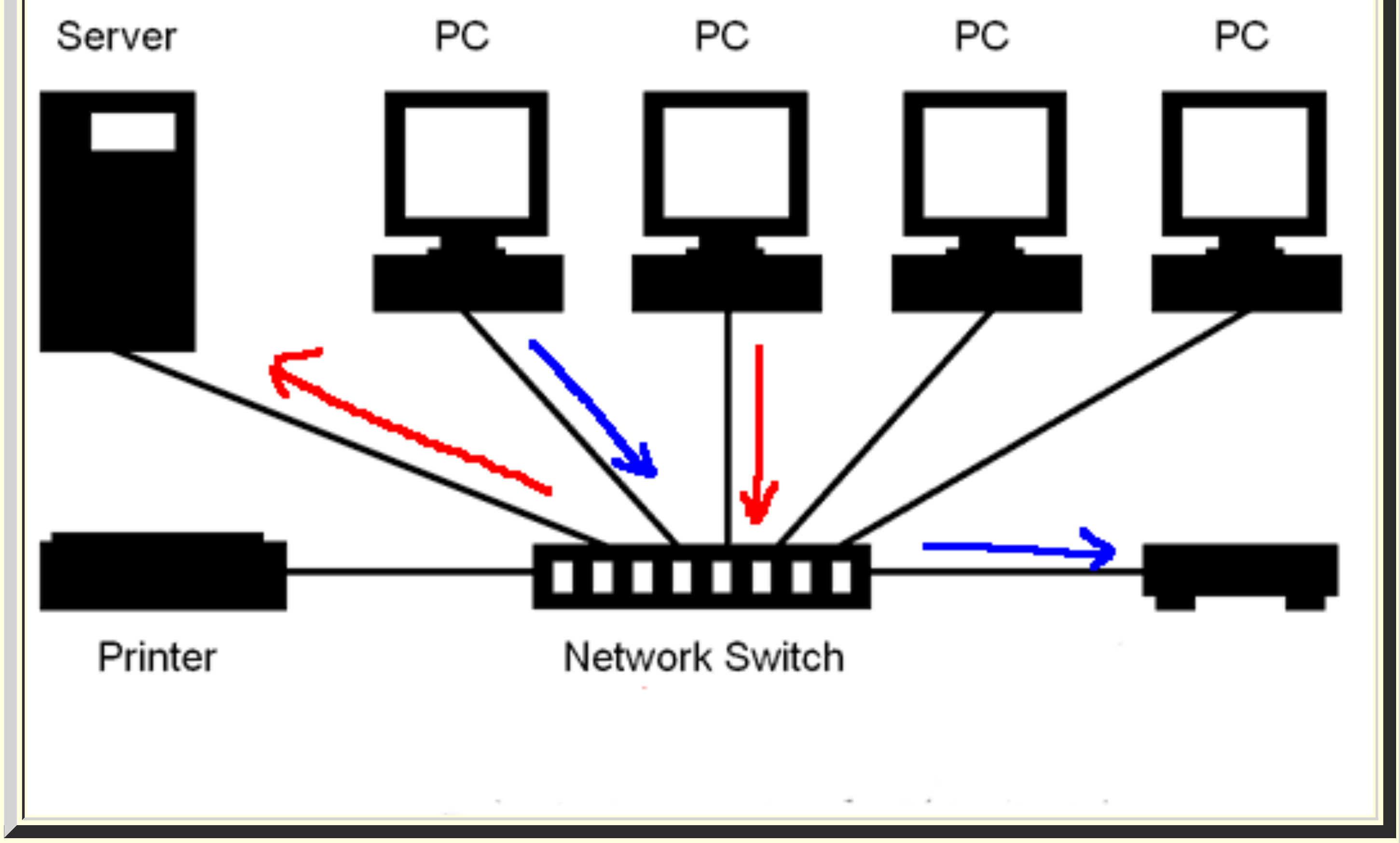
- **Simplest** switched network: use **one switch**



Notice that:

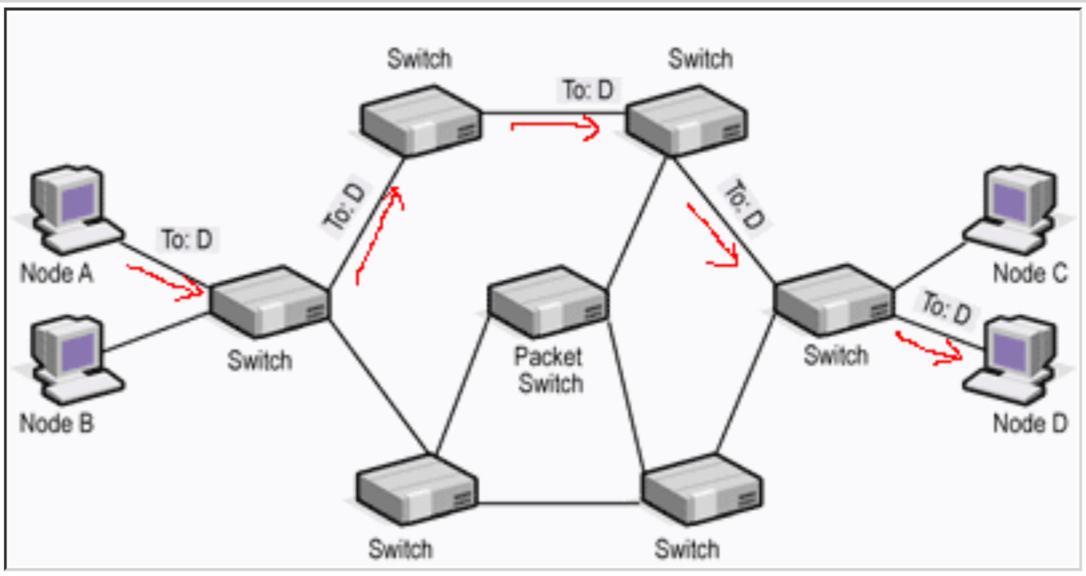
- **Messages** from a **sender** is **forwarded** ***only*** to the intended **receiver** in **switched networks**

- **Simultaneous transmission in a switch**
  - **Important fact:**
    - A **switch** supports ***simultaneous*** **transmissions** as long as the **transmissions** are ***not*** to the ***same*** **output**:



The red **transmission** and the blue **transmission** can occur ***simultaneously*** !!!

- **Switched *network***
  - **Switched network**: uses ***multiple*** **switched**

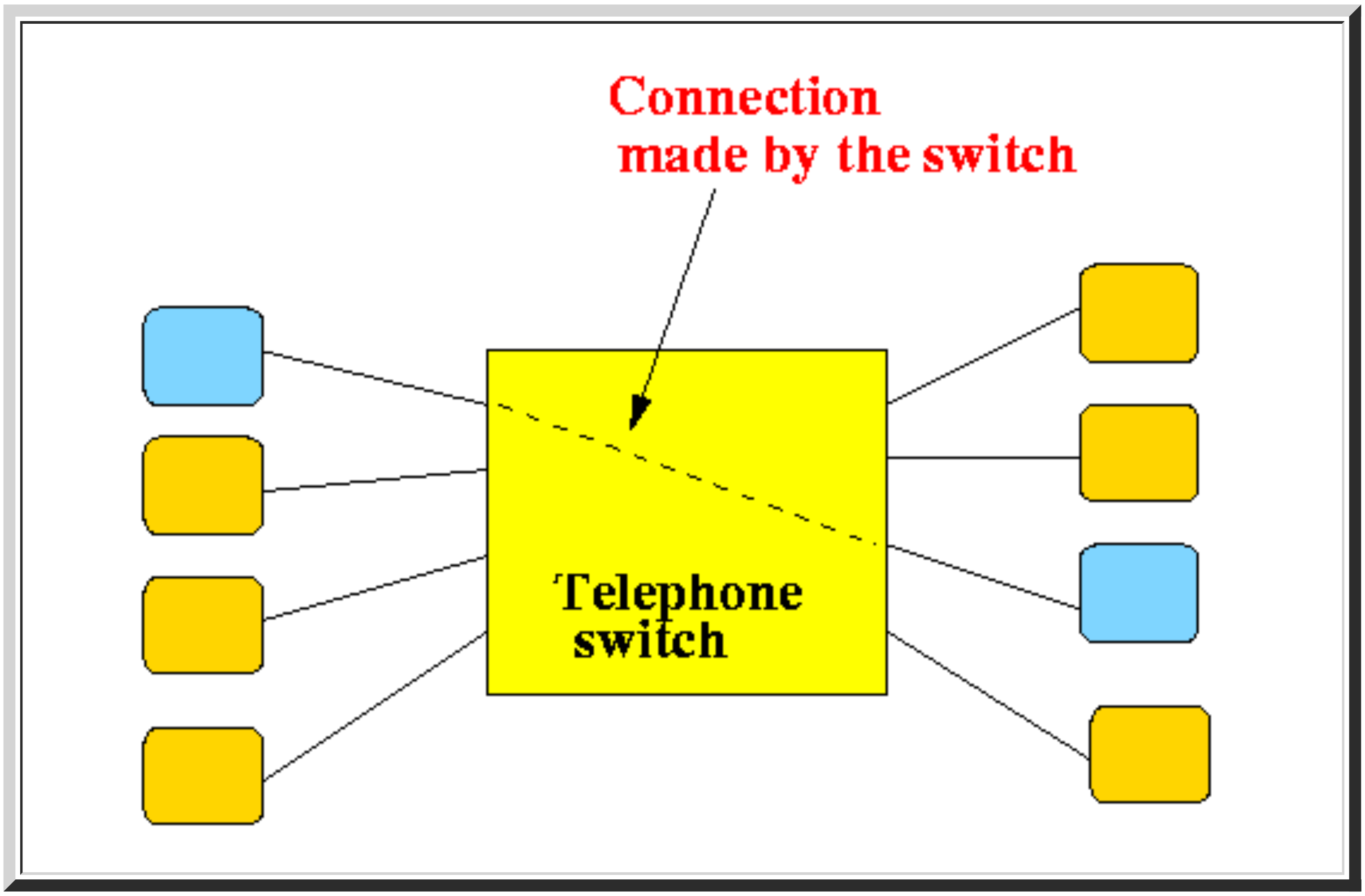


Noticed that:

- ***Only*** the **destination (receiver) node** will receive the **message**
- **Message must** follow a ***correct* path** in order to **reach** the intended **destination** !!!

- **Switched Network demo**

- **Circuit demo**:



- **Demo (Switch):** (Demo above code)

Example

  - Prog file: click here
  - **How to run the demo:** `/home/cs455001/demo/Logic-Sim/mux-demux`

- **Observe** the **property** of a **switch**:

  - **Signal (= message)** is sent to a ***specific*** **output**

  - The ***other*** **outputs** do **not** receive the **signal** !!

- **Switched Network: Advantages/Disadvantages**

  - ***Advantage***:

    - **Non-intrusive transmissions**:

      - ***Only*** the ***destination*** **computers** on the network will **receive** the **transmission**

        (***Other*** **computers** will ***not*** be ***intruded*** **upon** by the **message transmission**)

Result:

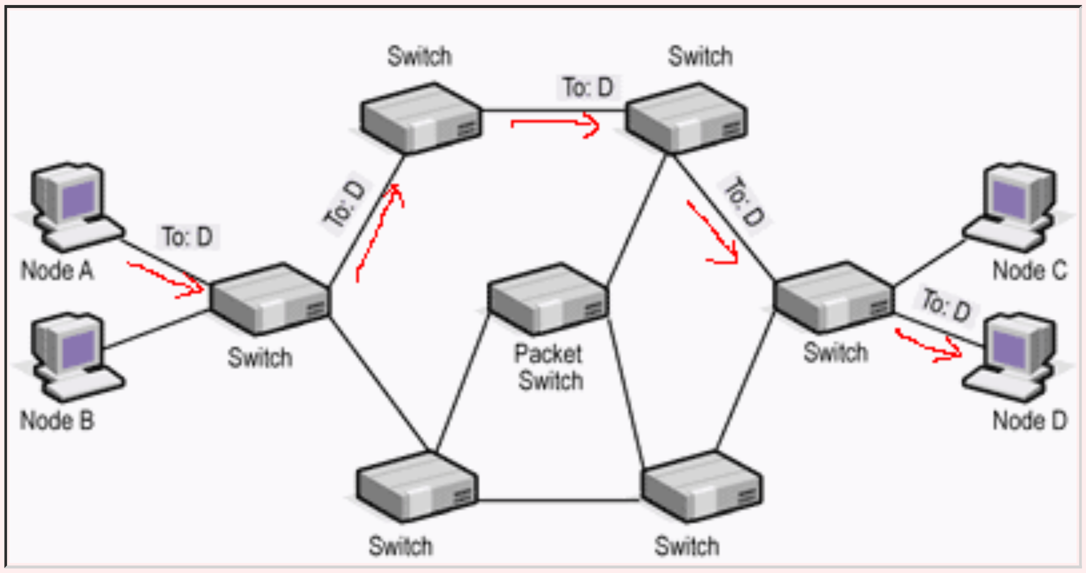
- **Switched networks** can support a ***very larger*** number of **computers** in the **network**

- ***Disadvantage***:

  - ***Expensive* Hardware**:

    - **Switching circuits** is ***very* complex** and **expensive** to **manufacture**

  - Requires **complex software** to find ***routes*** in the **network**:



Route discovery is ***not*** a **trivial problem** to solve...

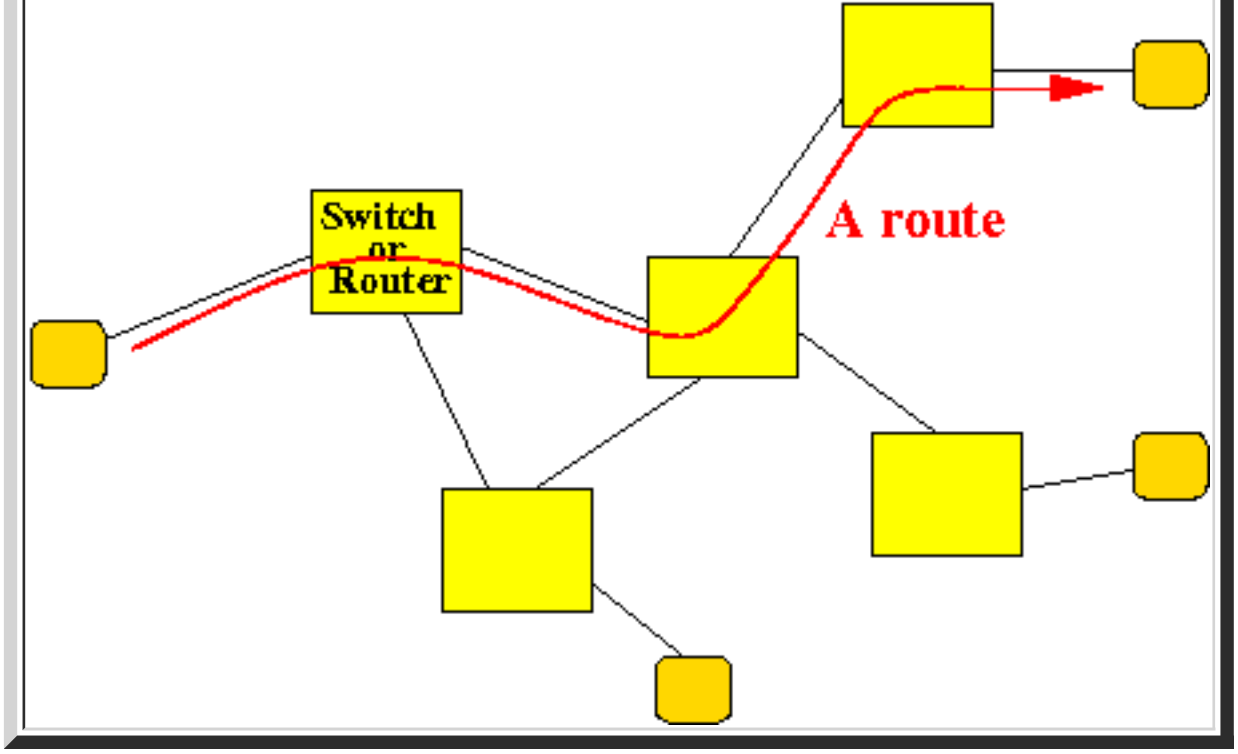
(The **Route discovery** problem is discussed a liitle more in the next section)

- **The "Routing Problem" in switched networks**

  - **Problem** in a ***switched* network**:

    - In **switched networks**, we need to **find/discover** a **correct route** from **source** to **destination**

Example:



- **Finding** a **route** through a **network** is a **2 step process**:

  1. **Discover** the **network *topology***:

     - ***How*** are the **nodes connected** to each other ????

     To **solve** this **problem**, the **nodes** must exchange ***connectivity* information** with each other

  2. Using **Dijkstra's Algorithm** to compute the ***shortest* paths** to the **destination**

     - This step can ***only*** be **used *after*** the **nodes** has discovered the **connectivity** of the **network** !!!!

We will discuss the **routing problem** in more **details *later***

Hybrid computer networks

- **Combining the *strengths* of broadcast and switched networks**
    - **Strengths:**
        - **Broadcast network** is **cheap**

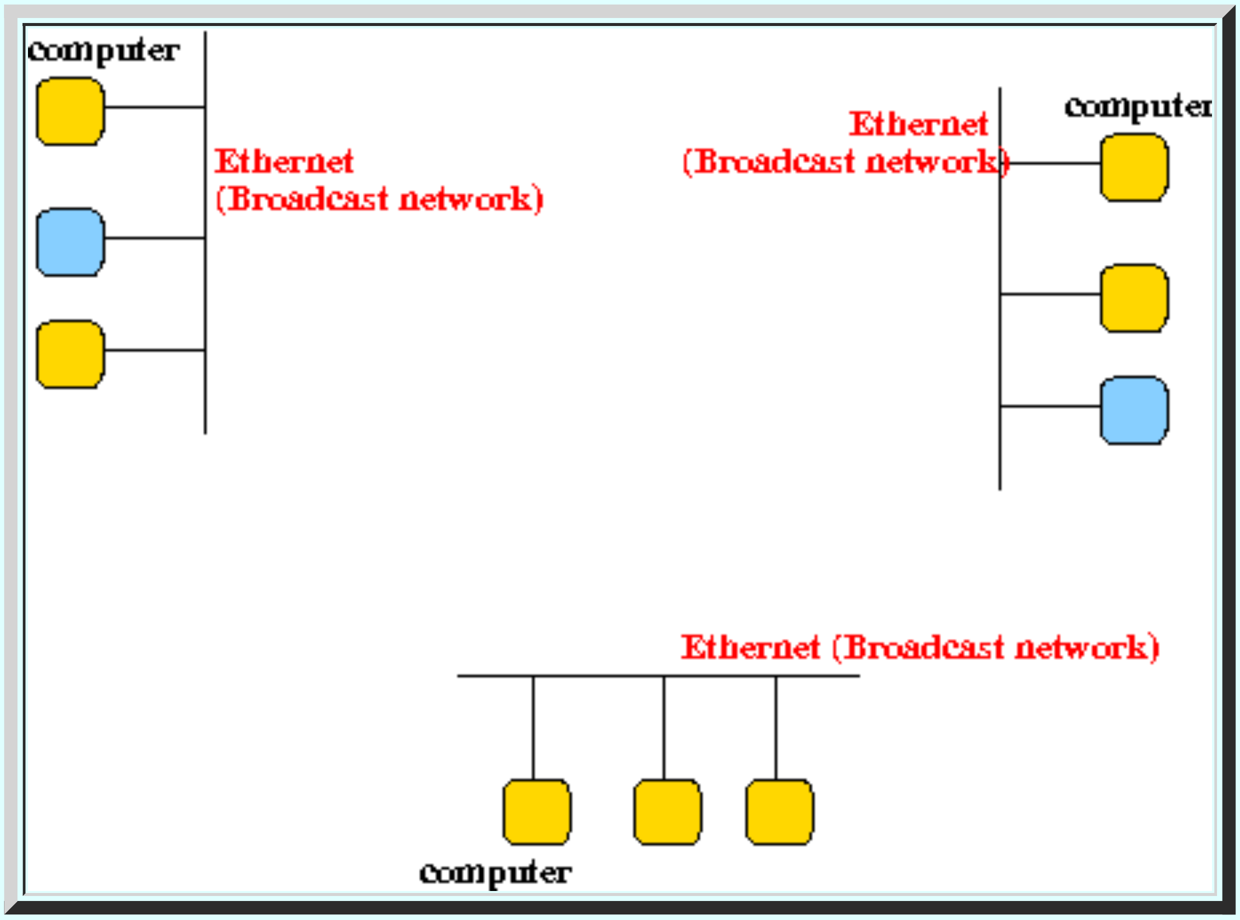
          (but can ***only*** support a ***small*** **number** of **computers**)

        - **Switched networks** can deliver **messages** to ***only*** the ***intended*** **receiver**

          (but the hardware **cost** is ***very*** **high**)

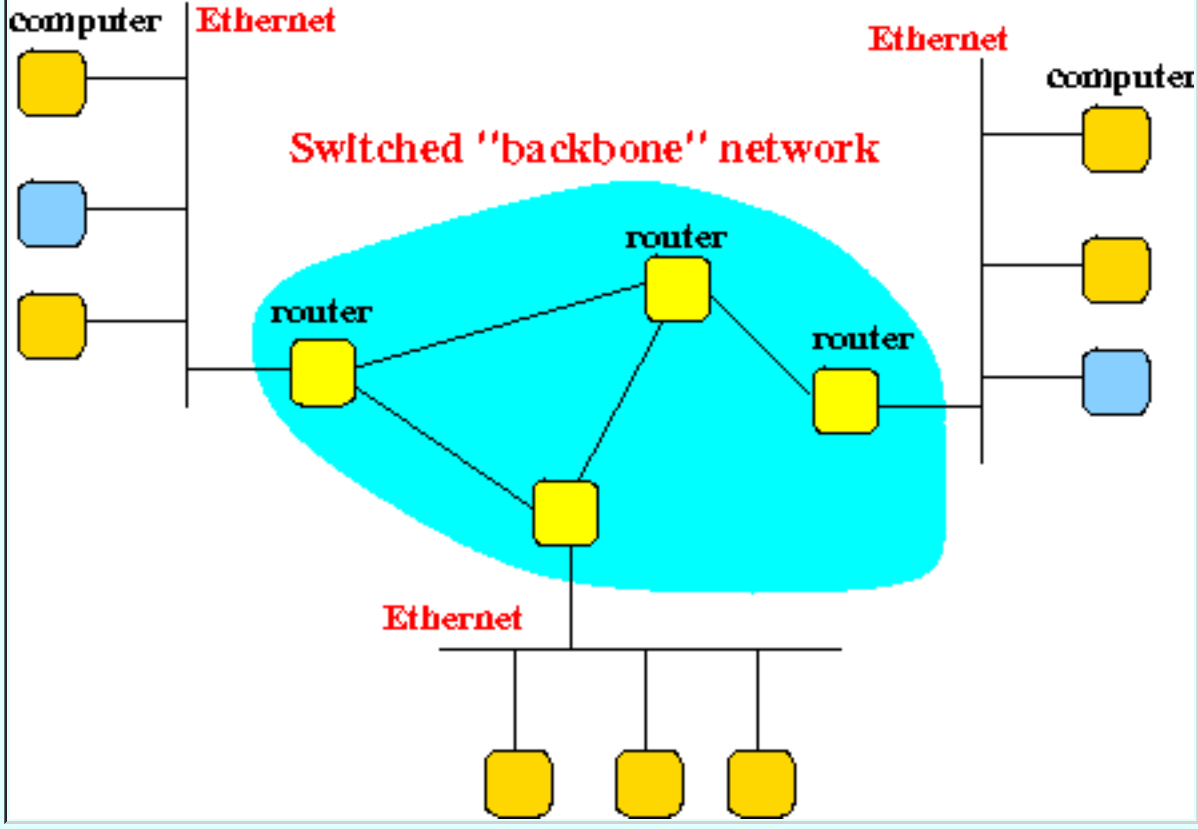
    - **Combining** the **strengths** of ***both*** **types** of **networks**:

        1. Build ***small*** **networks** with **(cheap)** ***broadcast*** **hardware** (for **each department**):

           ---

        2. ***Interconnect*** the ***broadcast*** **networks** together into a ***larger*** **network** using ***switching*** **technology**:



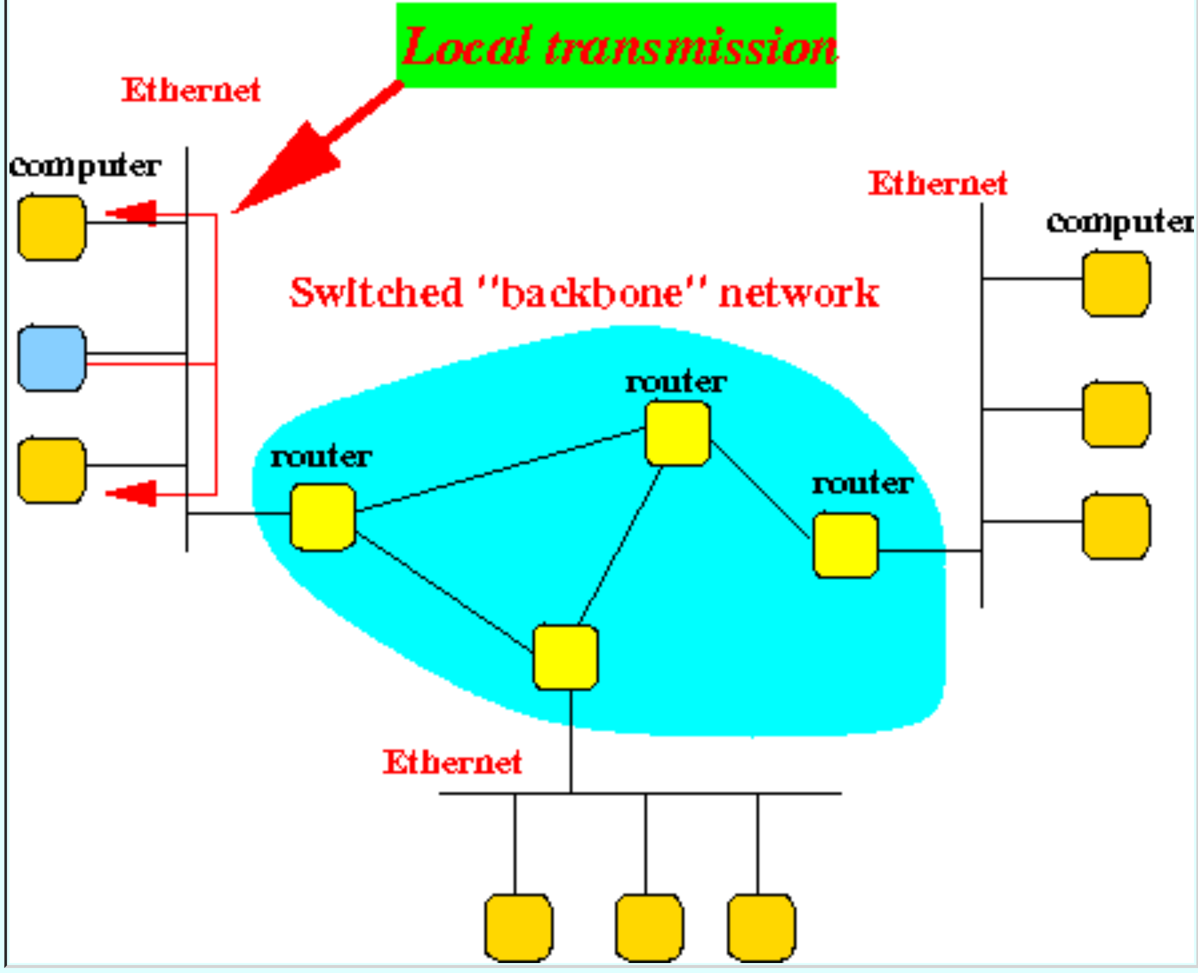
- **How the *hybrid* network operates**
  - **Fact:**
    - **Most** messages **transmitted** between **computers** are **between** computers on the ***same*** **(broadcast) network**

    Analogy:

    - People ***often*** talk to their ***neighbors***.

      People ***sometimes*** will talk to their ***distant* relatives**

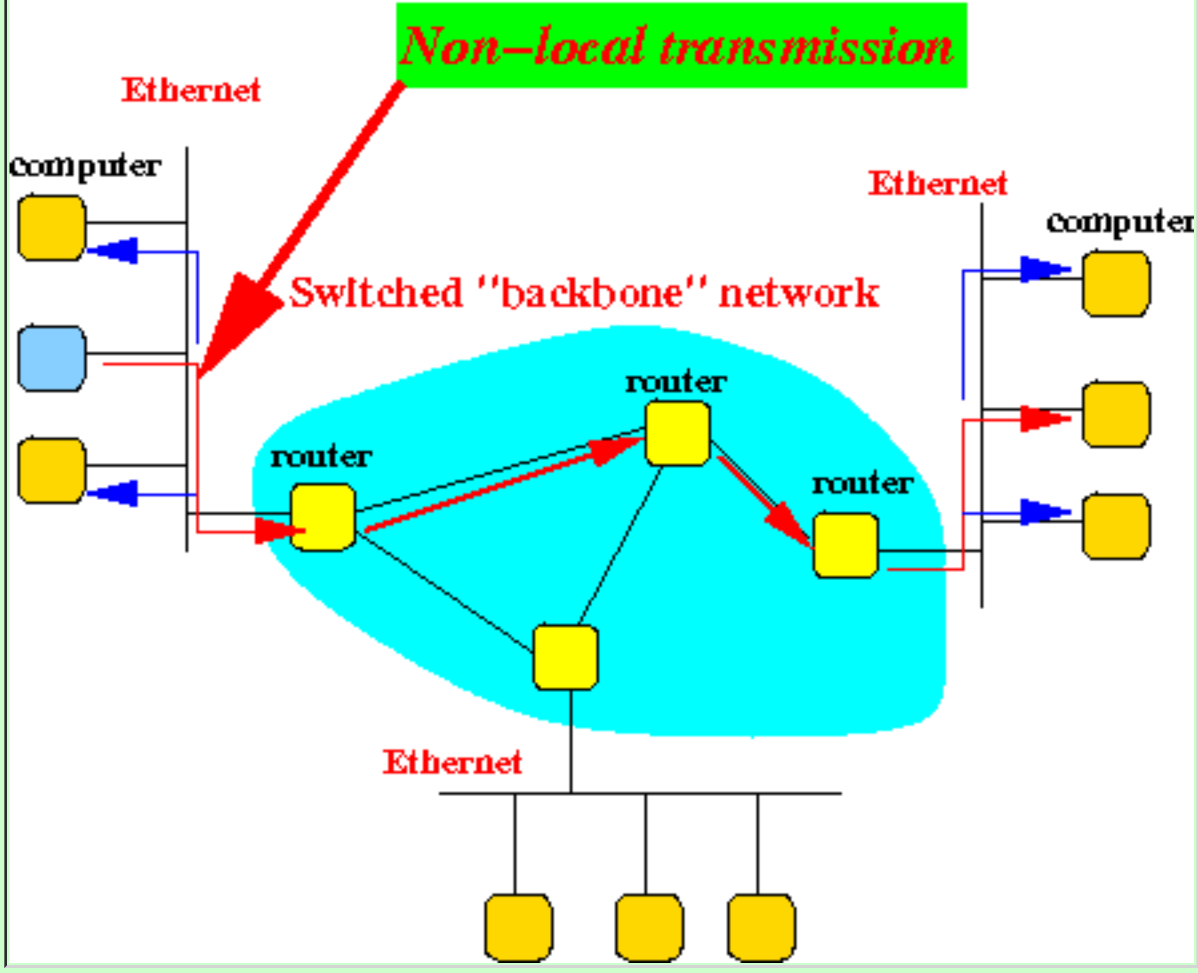
  - **Therefore**:
    - ***Most*** messages will be **transmitted** ***only*** on the **broadcast network**:



The ***local*** **transmissions** will **not** be **transmitted** to ***other*** **(broadcast) networks**

- **Note:**
  - ***All*** **nodes** on the ***same*** **broadcast network** as the **transmission** will **receive** the **transmission**....

- ***Non-local*** **transmissions** are **processed** as **follows**:



- the **sender** sends the **message** on its ***broadcast* network**

- The ***router* (switch)** connected on that ***broadcast* network** will **forward** the **message** to the ***destination* (broadcast) network**

Note:

- ***All* computers**:

  - on the ***source* network** **and**
  - on the ***destination* network**

  will ***receive*** the **transmission** !!!!

Another way to classify computer networks

- **Another way to classify networks: by "range"**
  - **Networks** can also be **classified** by the **"range"** that they span:
    - A **local area network** (LAN) spans a **small geographic area** (e.g., within a **building**).

      Example: Ethernet !

      LANs connects a limited number of computers (you don't usually put hundreds of thousands of computers in one building....)

      Not surprising, the technology used to build LANs is **broadcast** (cheap).

    - A **metropolitan area network** (MAN) spans a **medium geographic scale** area (e.g., within a **city**).

      Example: DQDB (Distributed Queue Dual Bus)

      NOTE: nowadays, because **switches** are so **cheap**, **MAN architectures** are not discussed/researched at all, so I will leave it at that...

    - A **wide area network** (WAN) spans a **large geographic scale** area (e.g., between state, a country, between countries...)

      WANs are built using **switching technology** (i.e., WANs are always switched networks).

Transmission Methods in Broadcast Networks

- **Transmission Methods in Broadcast Networks**
    - We will **discuss transmission methods** in ***broadcast* networks *later*** in the course....
        - See the **chapter** in the **CS455 syllabus**:
            - **Data Link Layer for Broadcast Networks**
    - **Main theme:**
        - The **topic** is all about **medium access protocols**...

Intro to Transmission Methods in Switched Networks

- **Some background information: congestion**
  - **Network congestion:**
    - Each **data processing system** has a certain ***maximum* processing capacity**

    - So it is **obvious** that:
      - Every **computer network** has a certain ***maximum* transmission capacity**

    - When a **system** reaches/exceeds its ***maximum* processing capacity**, there will be **backlogs** in **processing**

      We call these **"backlogs": congestion**

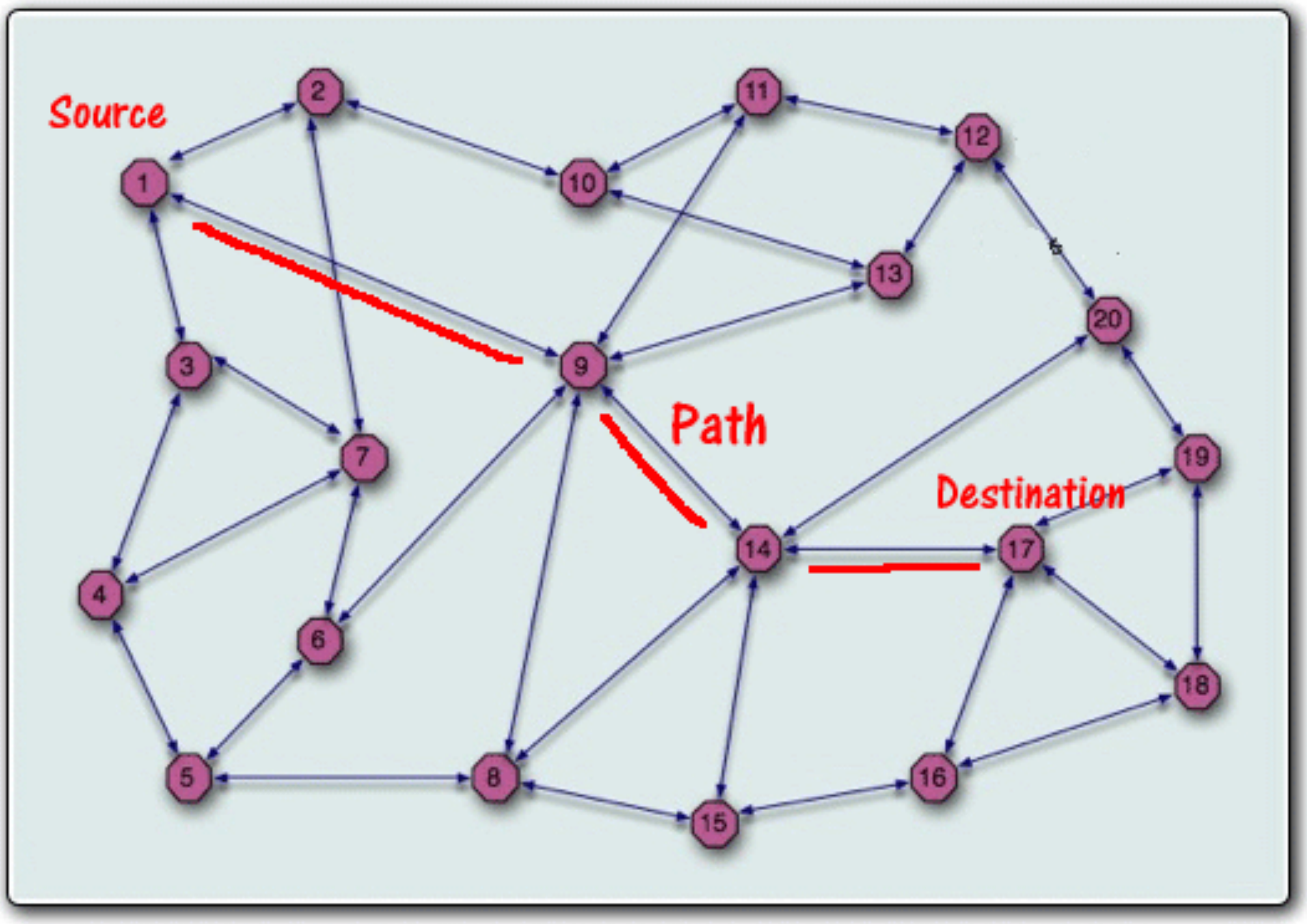
      

- **Best route to the destination**
  - **Facts:**
    - When **transmitting** a **message** to the **destination**, we **always** try to use:
      - The ***fastest* (= best) route** !!!!

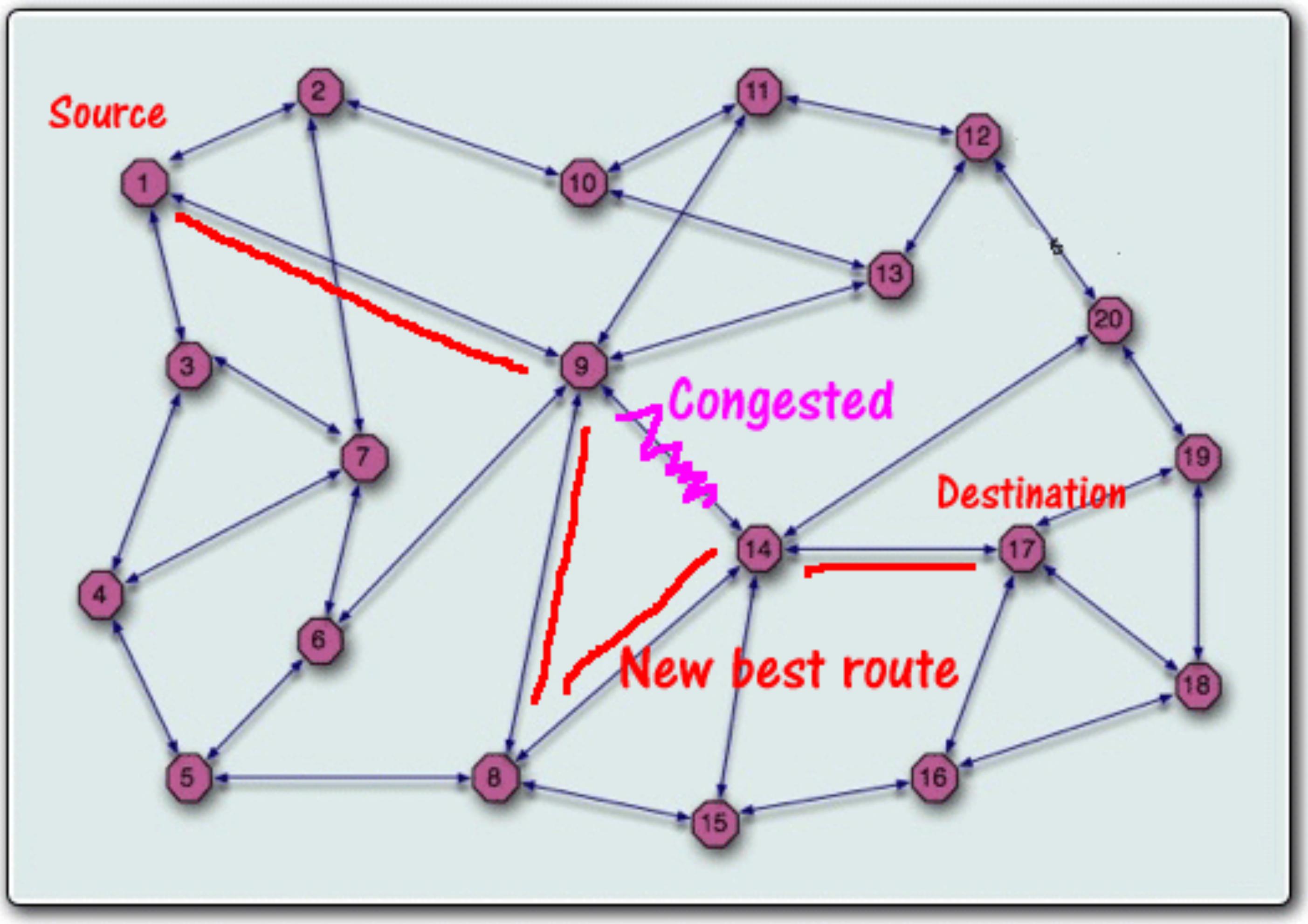
    - The **best route** is **dependent** on the ***traffic situation*** in the **network** !!!!

- **Example:**

  - **Best route** at **time 1**:



  - **Best route** at **time 2**:



(Yes, **computer networks** can become **congested** too :))

- **Communication session**
  - **Fact:**
    - **Computers** ***rarely*** send a ***single*** message to other **computers**

      Multiple **messages** are **usually** **exchanged** in a **single communication**

  - **Session**:
    - **Session** = the **serie** of **messages** transmitted between 2 computers to complete a **task**

- **Route and routing decision**
  - **Route:**
    - **Route** = the **path** taken by a **message** that **starts** at the **sender (computer)** and **ends** at the **receiver (computer)**

  - **Routing decision:**
    - **Routing decision** = **selecting** the **route**

- ***Timing*** **to make routing** ***decisions***
  - **Routing decision** can be made in **one** of **2 ways**:

    1. **Circuit switching**:
       - The **sender** make the **routing decision** ***once*** at the **start** of of a **communication session**

       Result:

       - ***All*** **messages** in the **same session** will use the ***same*** **route** to reach the **destination** !!!

       ---

2. **Message/packet switching:**

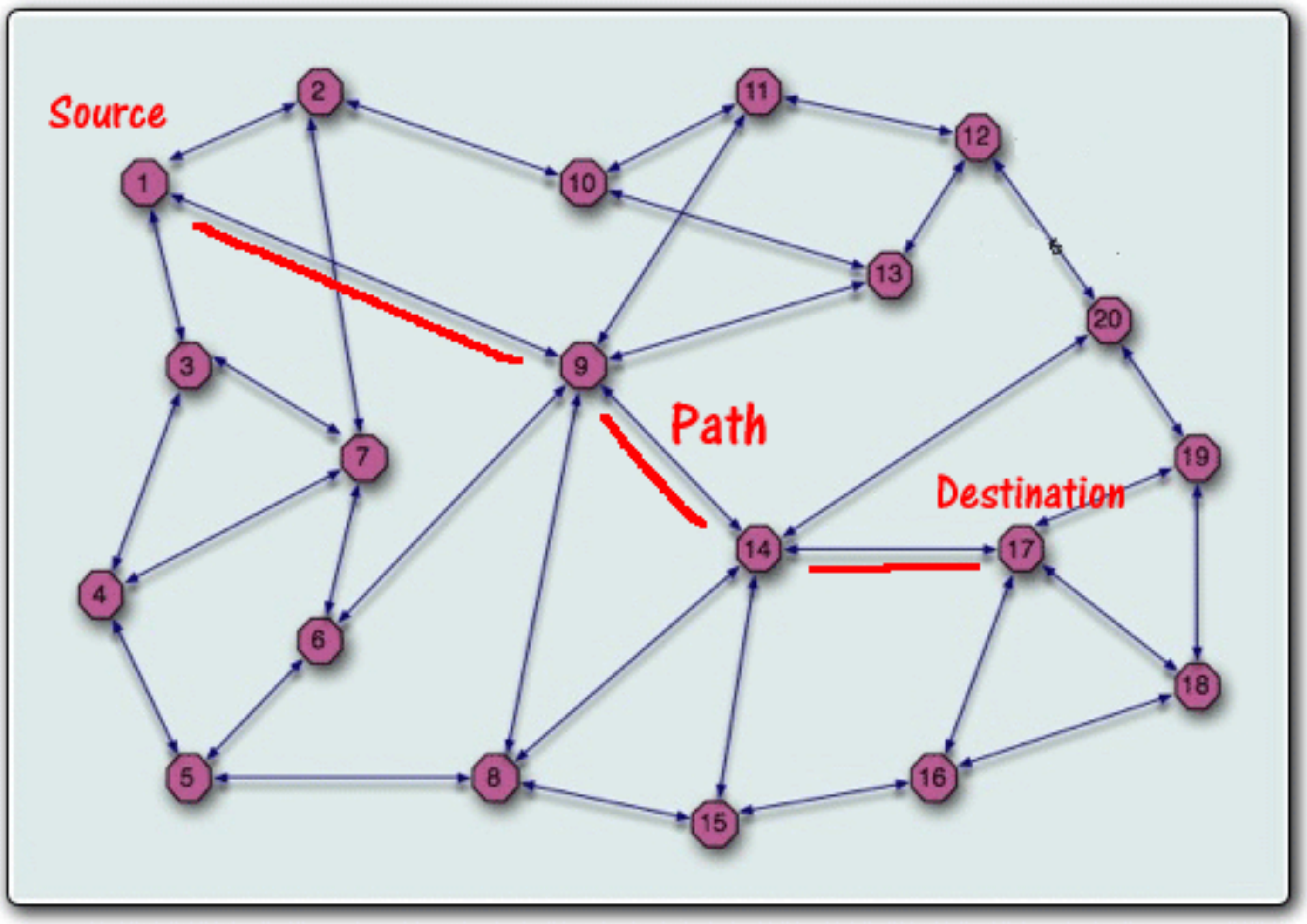
    - The **sender** make a **routing decision** ***every time*** it **sends** a **message/packet**

    Result:

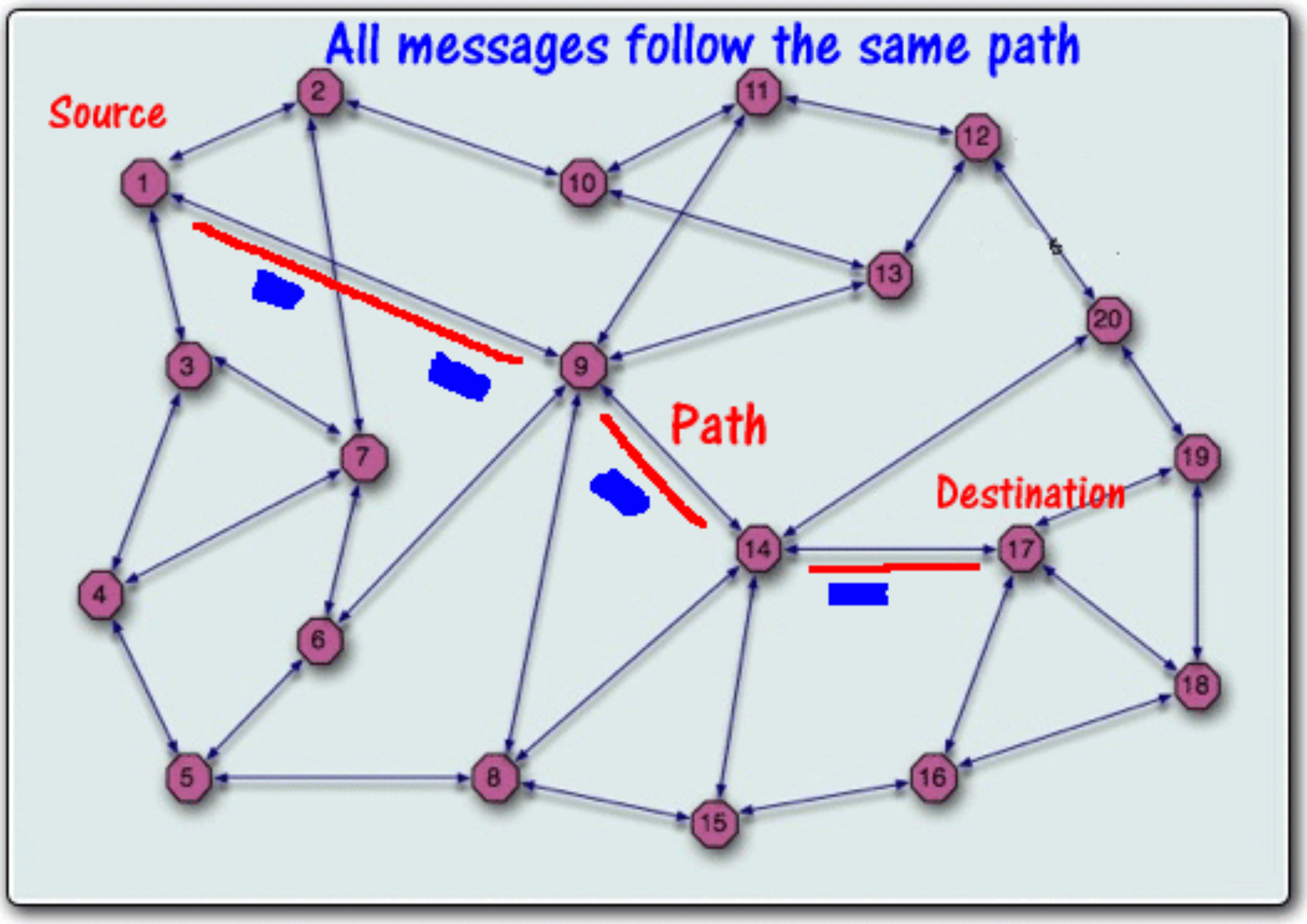
    - ***Each*** **message** will **use** its ***own*** **route** to reach the **destination** !!!

Properties of Circuit switching

- **Route taken by messages in a communication session in *circuit* switching**
  - **Transmission** using **circuit switching**
    - A **path (a.k.a.: circuit)** is **established** at the **at the *start*** of a **communication session**:



    - **Then**: ***all* messages** will travel the ***same* route**



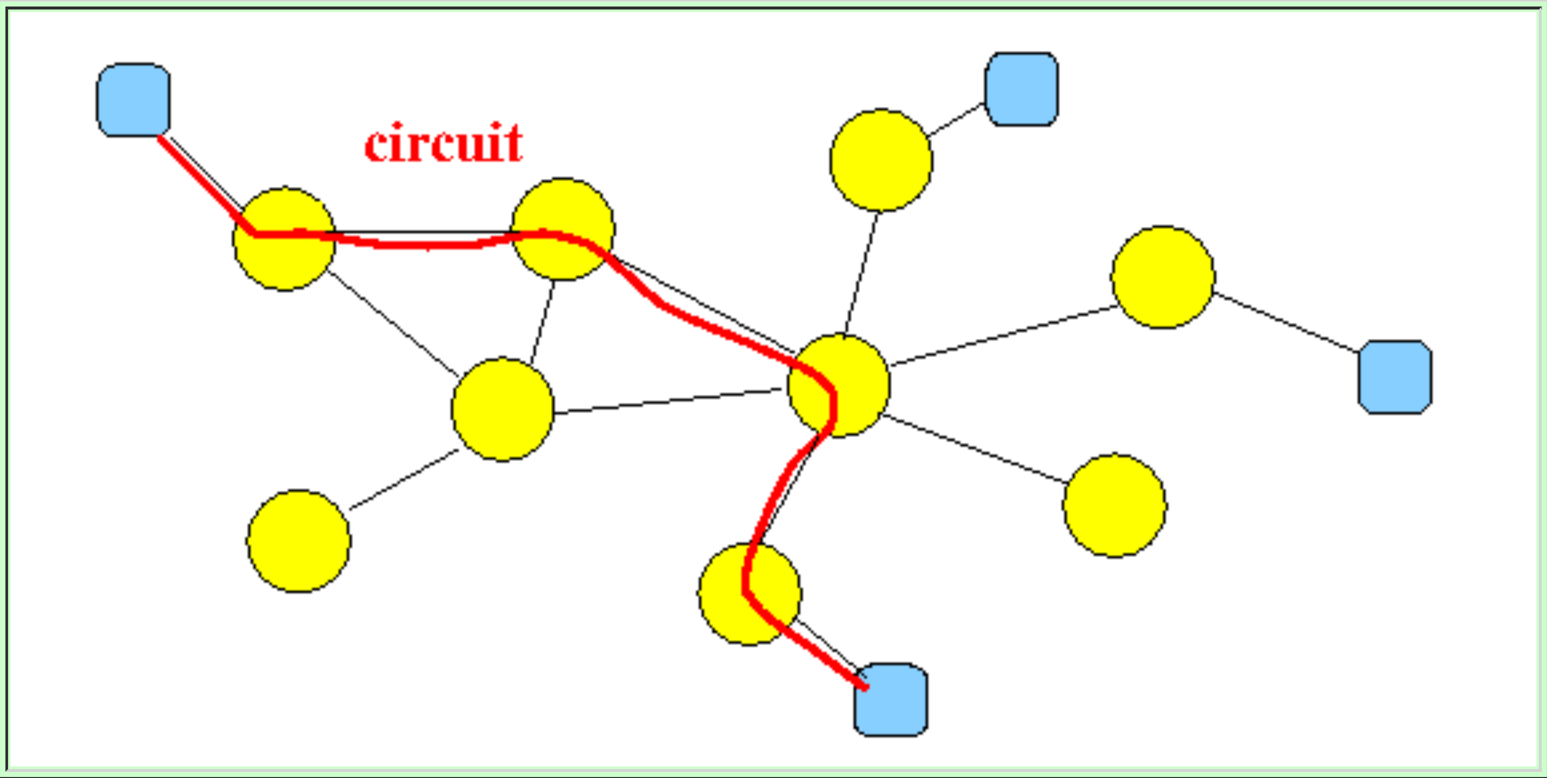
- **Properties of *circuit* switching**
  - **Properties** of **circuit switching**:
    - **Messages** will **arrive** at the **destination** in the ***same* ordering** as **transmitted**
      - Because **all messages** take the ***same* route** the **messages** cannot **overtake** one another
    - **Circuit switching** **cannot** guarantee **messages** will travel the ***best* route** ***all* the time**.
      - Because the **best route** can ***change*** over time....

        Circuit switching does **not** adapt to **future changes** in the **network conditions**

Performance guarantee in circuit switched networks

- **Circuit Switching**
  - **Recall that:**

    - In **circuit switching**, **all packets** from the same session **follows the *same* path** from **source** to **destination**

    Important consequence:

    - This **property** allows us to **make resourse *reservations*** at the **nodes along the path**

    - **Reserving network resources** can provide **performance guarantee** to the connection (= **user**) !!!

  - **Analogy:**

    - ***If*** I **know** that **route** I will take to get to a **destination (e.g.: Las Vegas)** on **Interstate Highways**:
      - I can **reserve** a **hotel room** in advanced on the **route** to **guarantee** that I have a **place to stay** for the **night** !!!

- **Phases in *Circuit Switching***
  - **Phases in the communication using circuit switching:**

    - **Setup phase**:
      - Before transmitting any data, a **transmission path (circuit)** with a given **transmission capacity** (e.g., 64 Kbps capacity) is **set up**

      - **Each router** on the **circuit** will **reserve** the given **transmission capacity**

      This **action** is called a **connection setup** and the **established *path* (= route)** is called a **circuit**

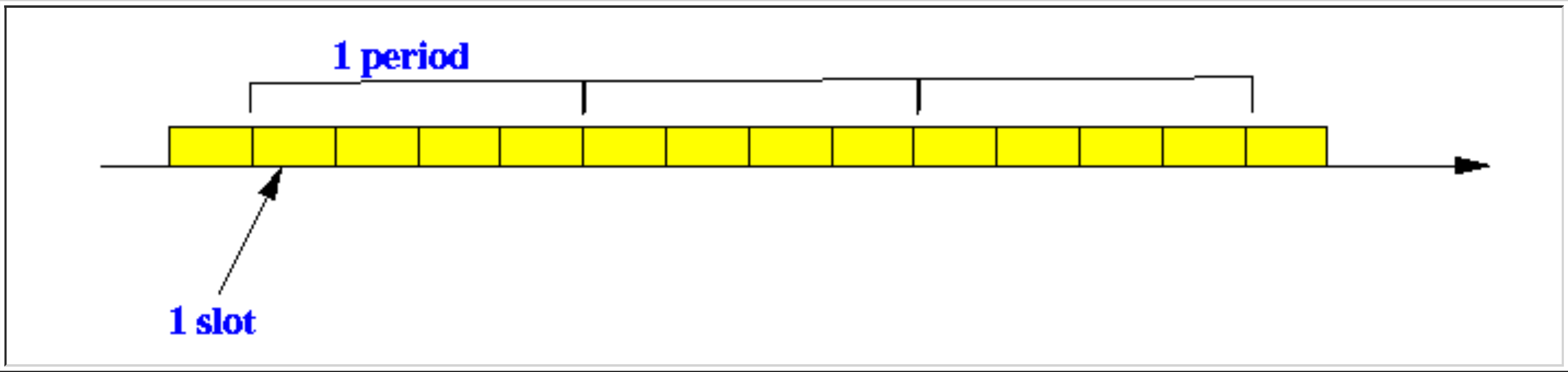
      The source can send data at the **requested transmission rate**, and it **cannot *exceed*** the requested rate

- **Transmission phase**:
  - The **actual message (data) exchange** occurs in this phase

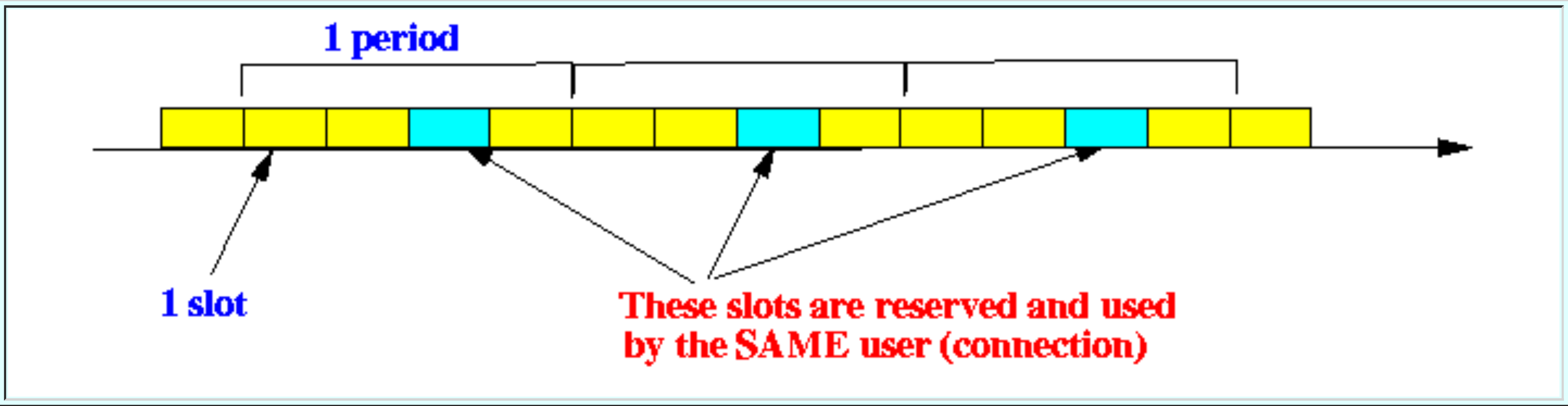
---

- **How to reserve transmission capacity on a communication *path***
  - **Reserving transmission capacity:**
    1. **Transmission time** is **divided** into **periods**

       Each **period** is divided in a number of **slots**:

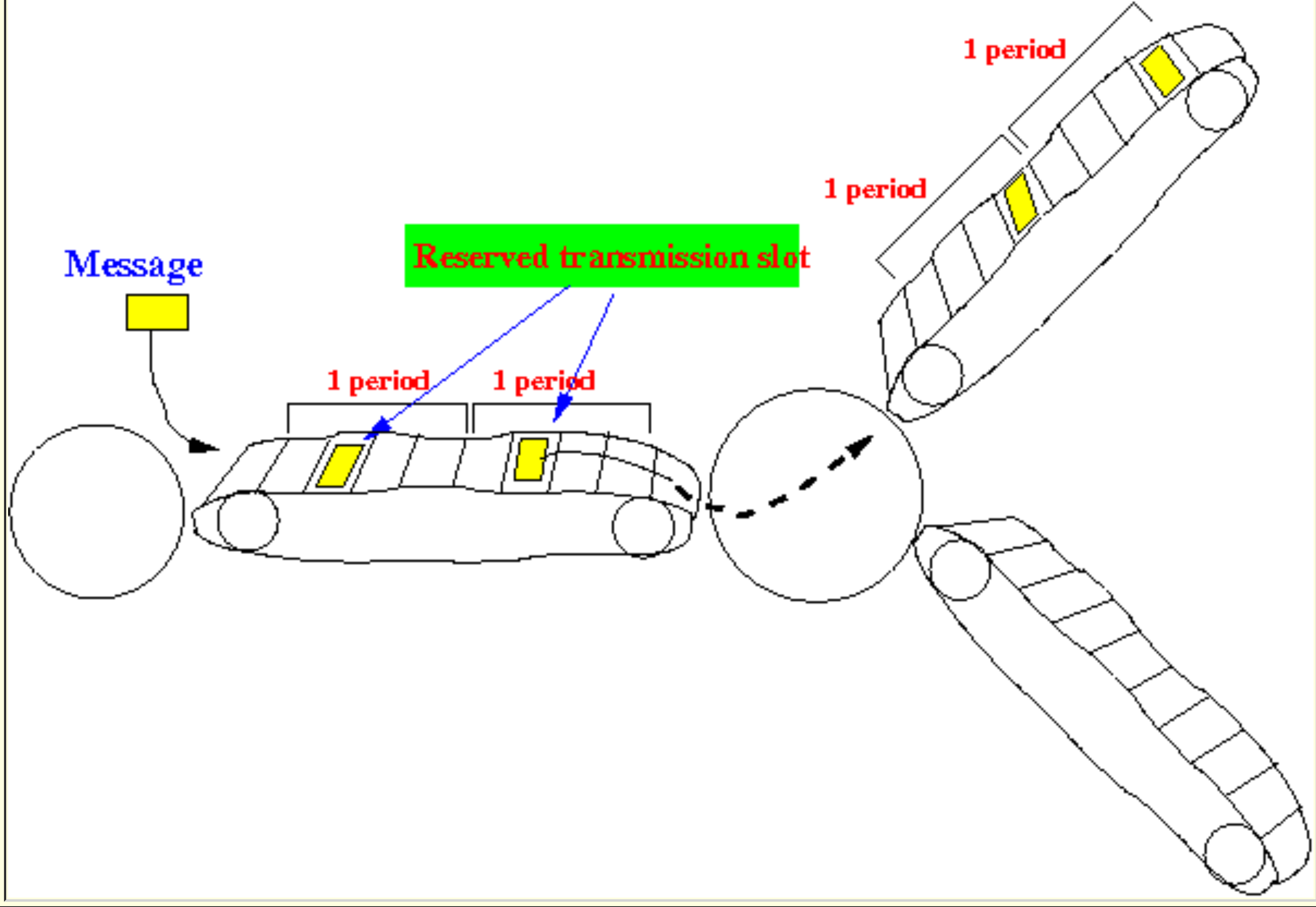
    2. When a **source** makes a **connection request**, **one (or more) *time slots*** will be **reserved** for packets from that source:

       This is done in **every node** along the **path** from **source** to the **destination**

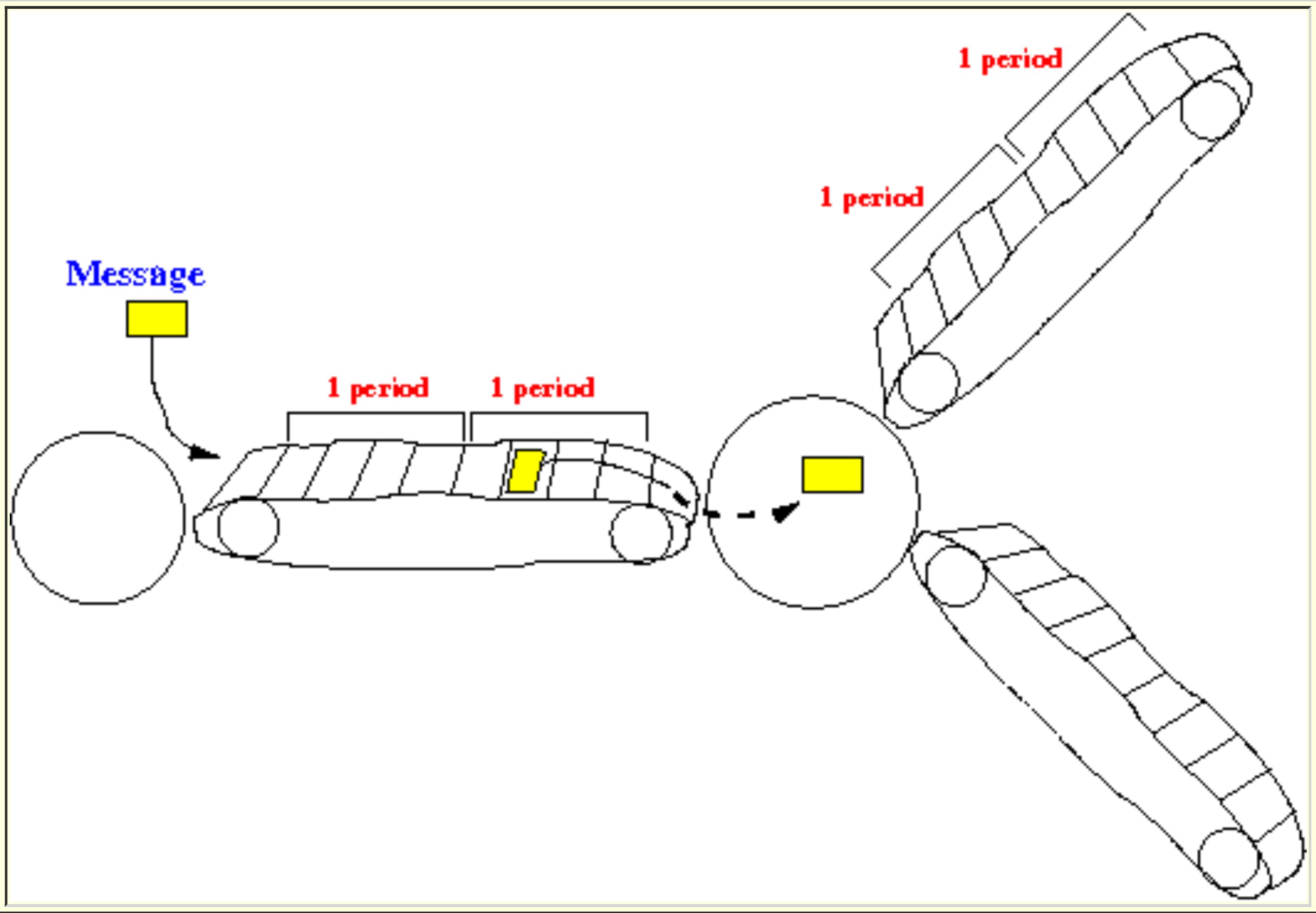
    Analogy:

    - Using a ***conveyer belt* analogy**, we can visualize the **result** of these **reservations**:

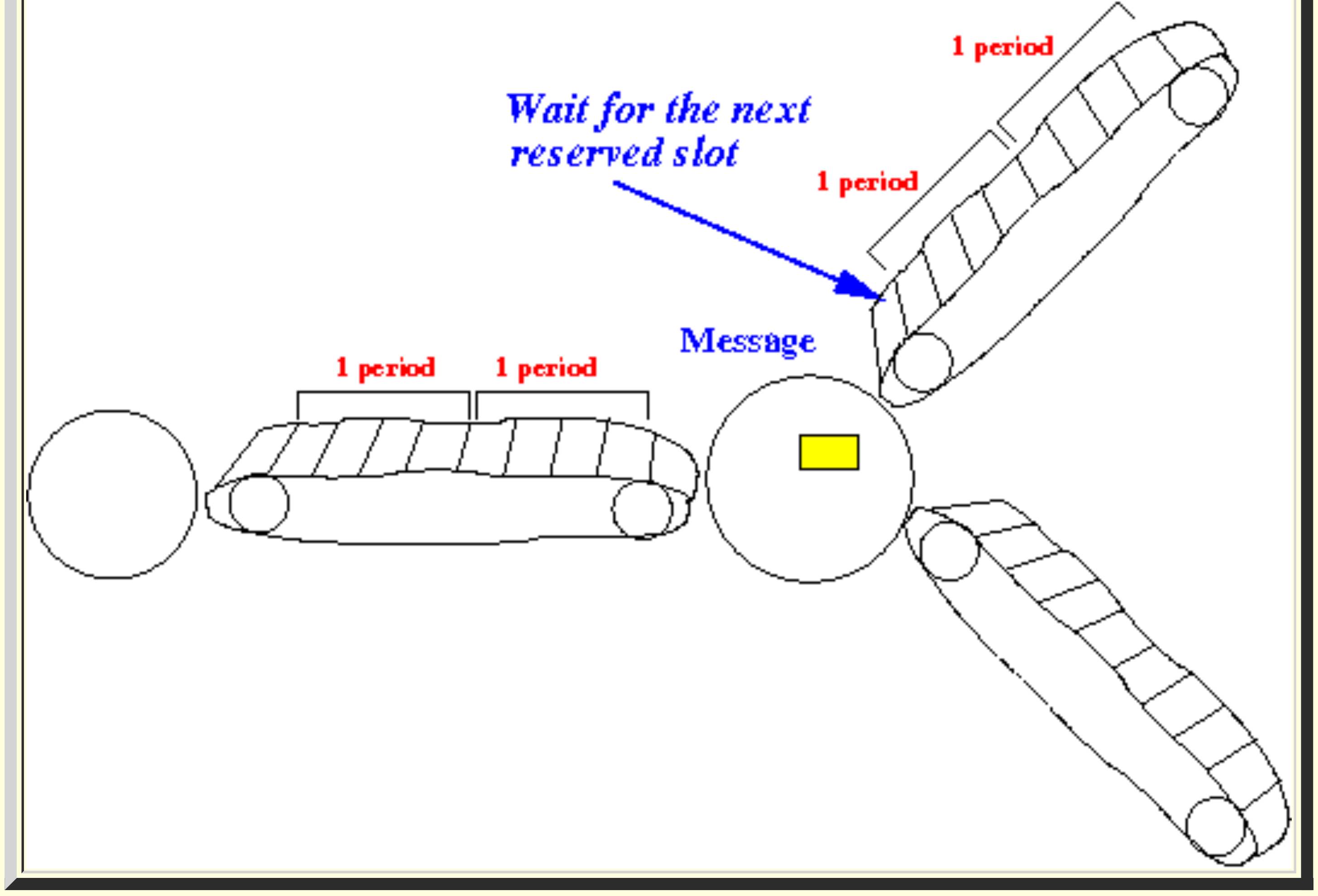


Operation:

- Packets from the **source** will **transverse** towards the **destination** **virtually *without* any delay**
- When a packet **arrives** at an **intermediate node**, it ***only*** need to wait **at most ONE period** of time before it will be **transmitted in the *reserved* slot** to the **next node**
- **Example:**
  - **Suppose** a **message** is **received** by a **node**:

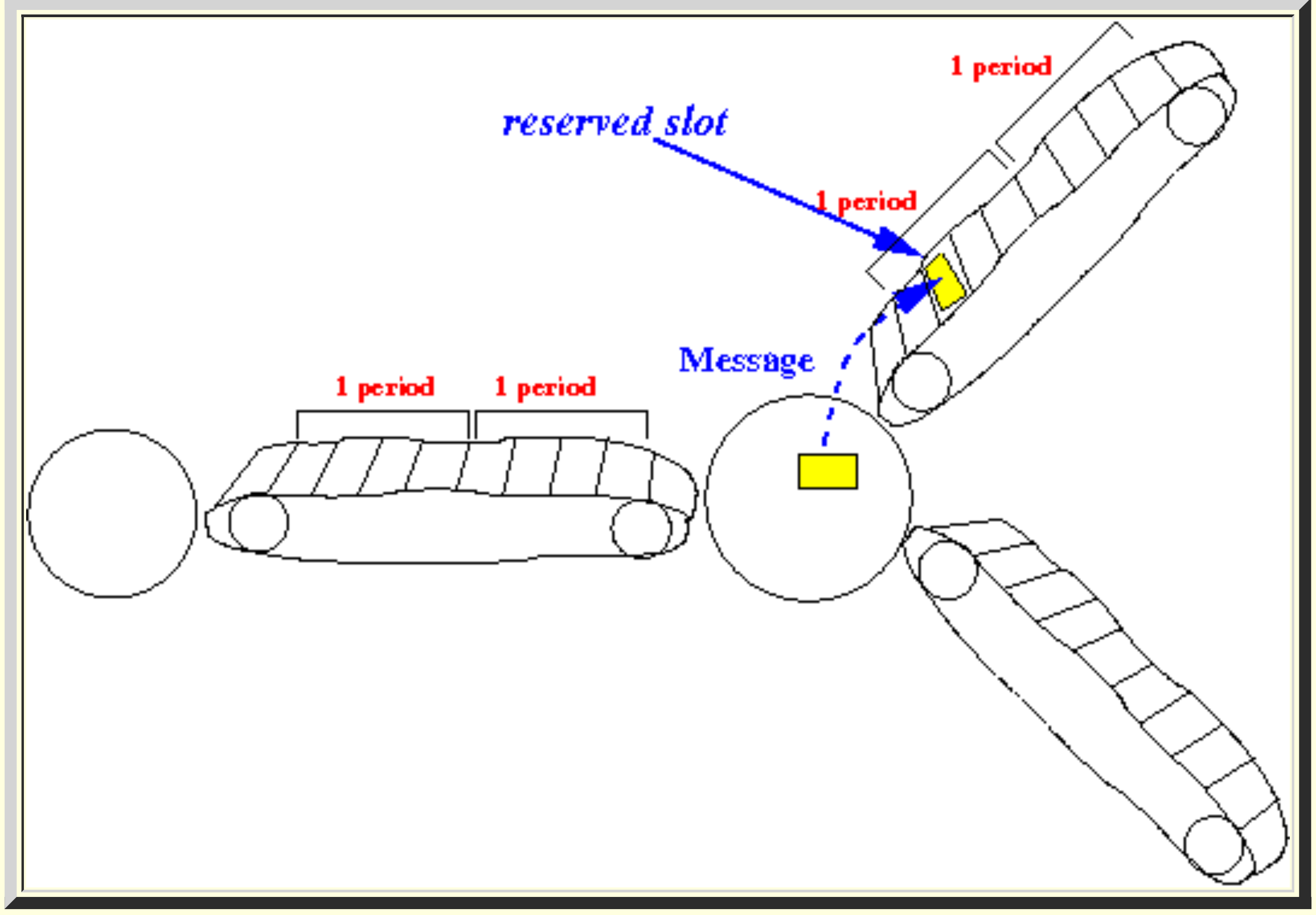
  - In order for the **node** to **transmit** the **message** to the ***next* node**, the **node** must ***wait*** for the **slot reserved** for this **message**:



Note:

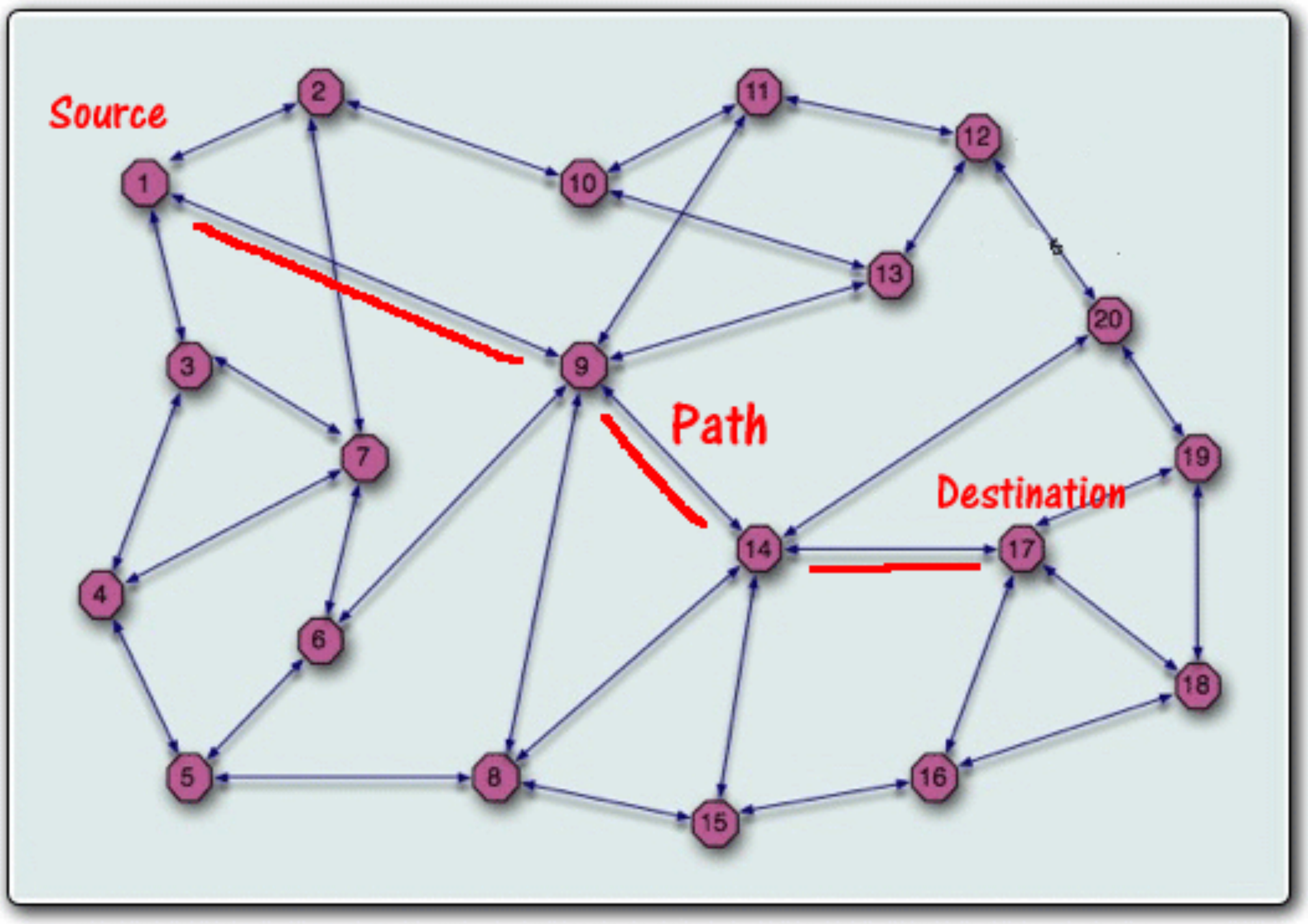
- The **waiting time** is ***very* short** because the **duration** of **1 period** is in the order of **micro seconds** !!!

- **As soon as** the ***reserved* slot** arrives, the **message** can be **transmitted** to the ***next* node**:

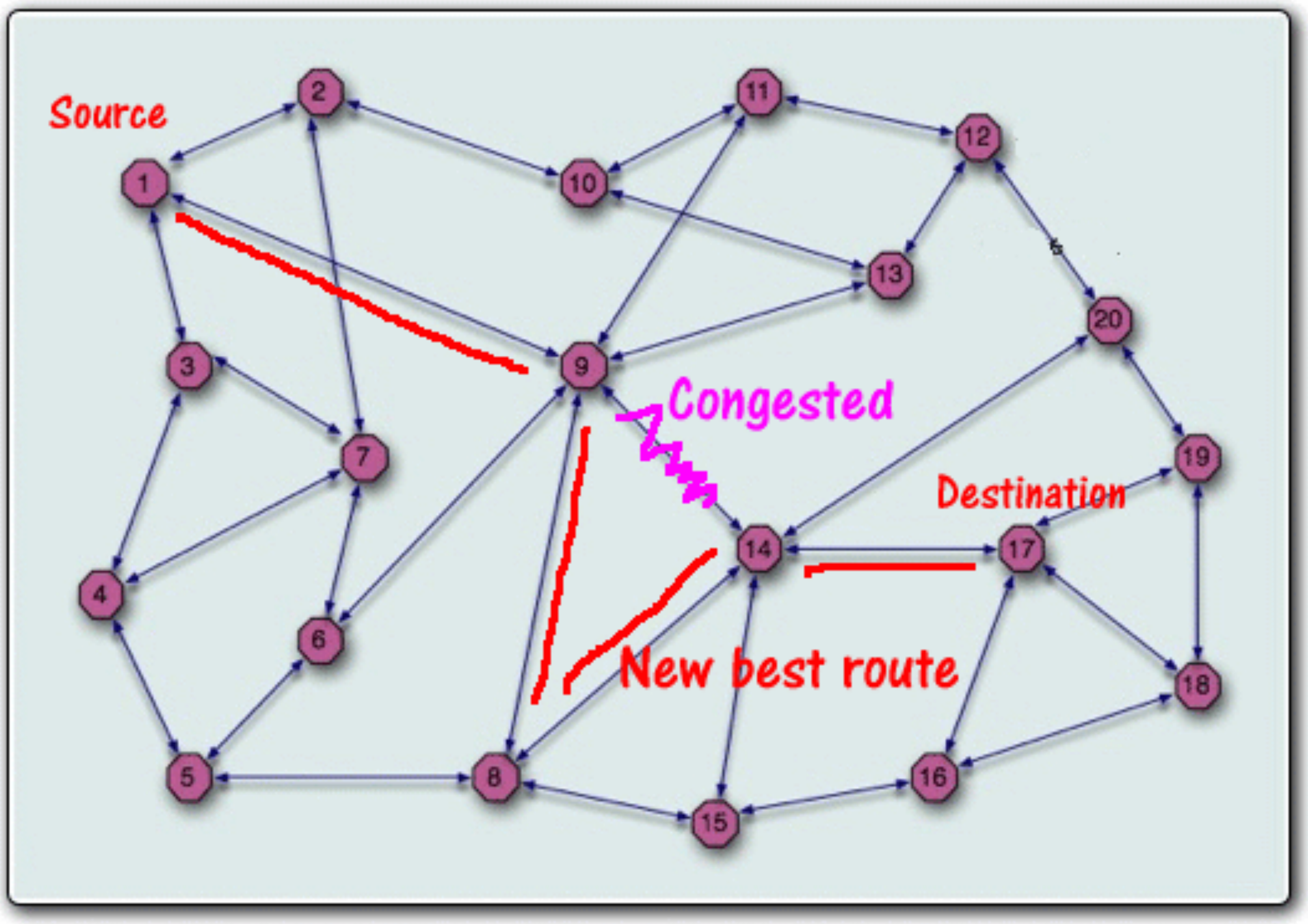


Properties of Message/packet (datagram) switching

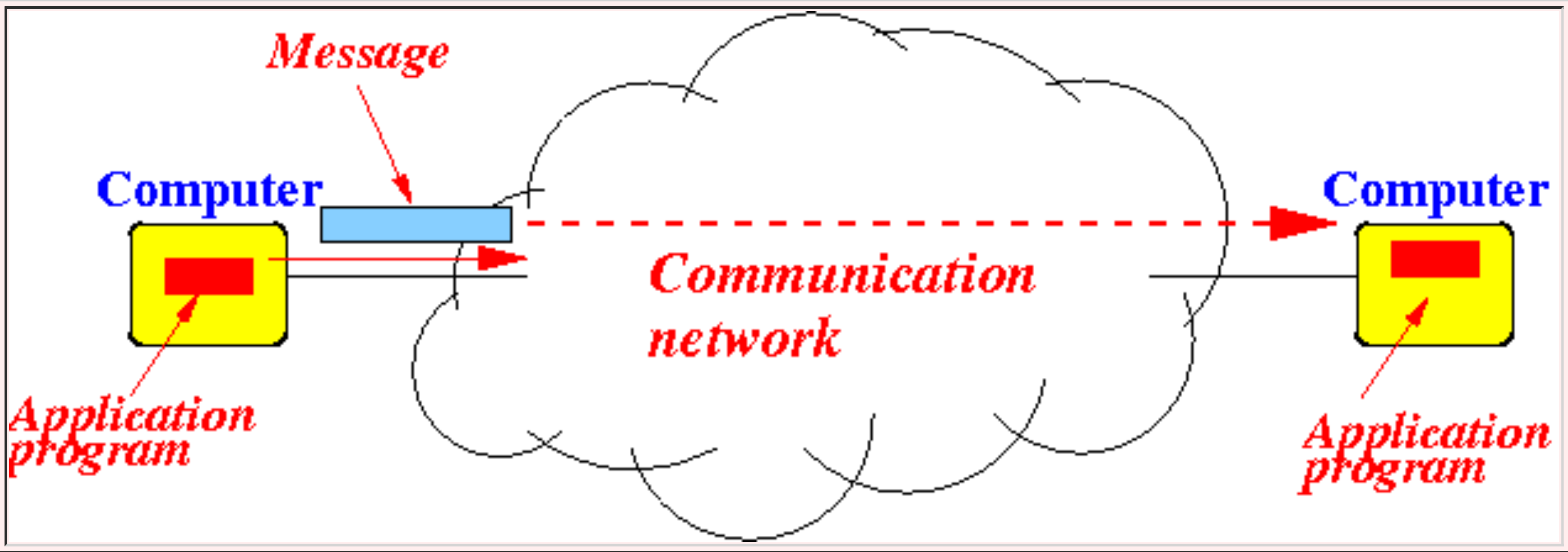
- **Routes used by messages in message switching**
  - **Route** used in **Message switching**
    - The **transmitted message** will use the ***best route*** at the **moment** that the **messages** is **sent**
  - **Example:**
    - The **(best) route** taken by **message** at some moment in time:



    - The **(best) route** taken by **message** at ***another*** **moment** in time:



- **Messages and packets**
  - **Message:**
    - **Message** = a **unit** of **data** that is **sent** by a **computer application (program)** (to ***another*** **computer application**)

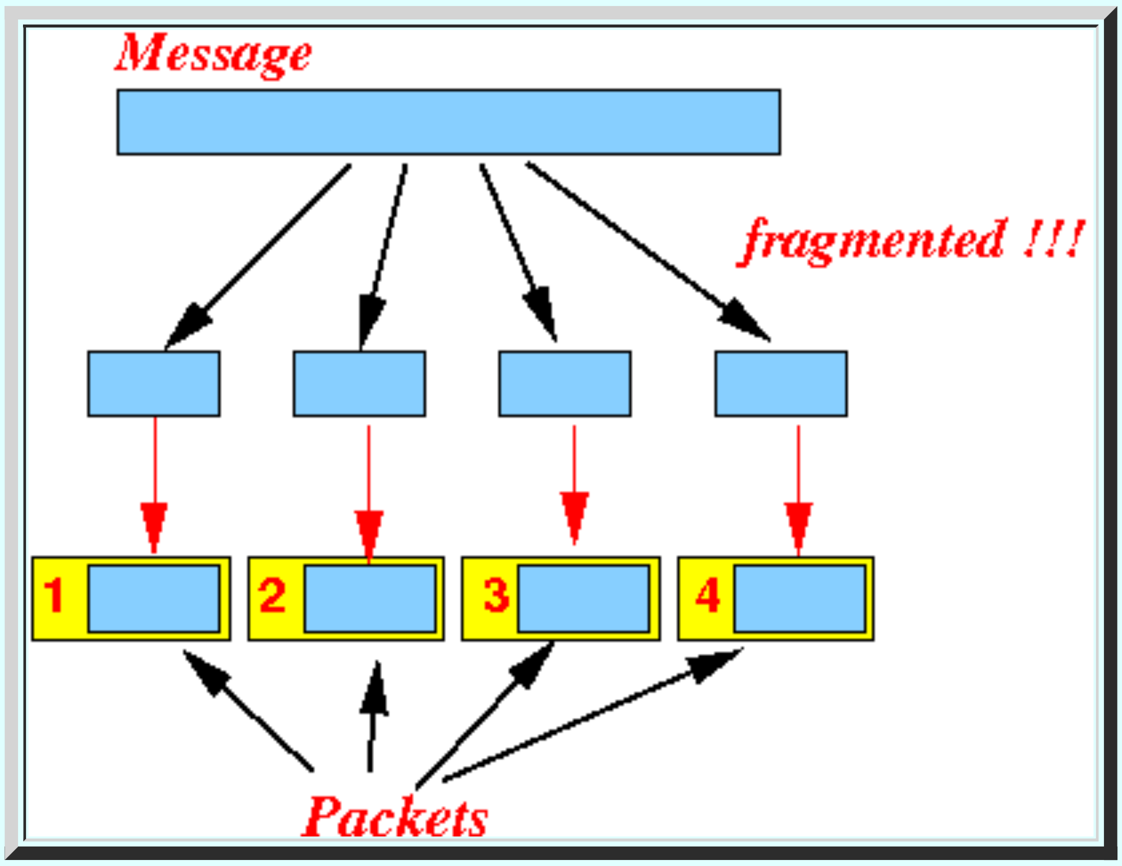


  - **Network limitations:**
    - **Most communication network** impose a ***maximum*** **size** on **transmissions** over the **network** !!!
      - This **limitation** is **imposed** due to **maximum buffer size** used

    ---

    - **Transmission** of of a **message** over **most computer networks**:
      - A **message** is **often transmitted** in **multiple *packets***

Illustrated:



Note:

- A ***sequence* number** is **included** to allow the **receiver** to ***re-assembler*** the **messages**

  (More on this topic **later**)

- **Packet:**

  - **Packet** = a **unit** of **data** that is ***transmitted*** on a **communication network**

- **Routing used in Packet switching**

  - **Routes** used in **Packet switching**

    - The **message** is **fragmented** into **packets**

    - **Each transmitted packet** will use the ***best route*** at the **moment** that the **packet** is **sent**

- **Notable Property of Message/Packet Switching**

  - **Properties** of **message/packet switching**:

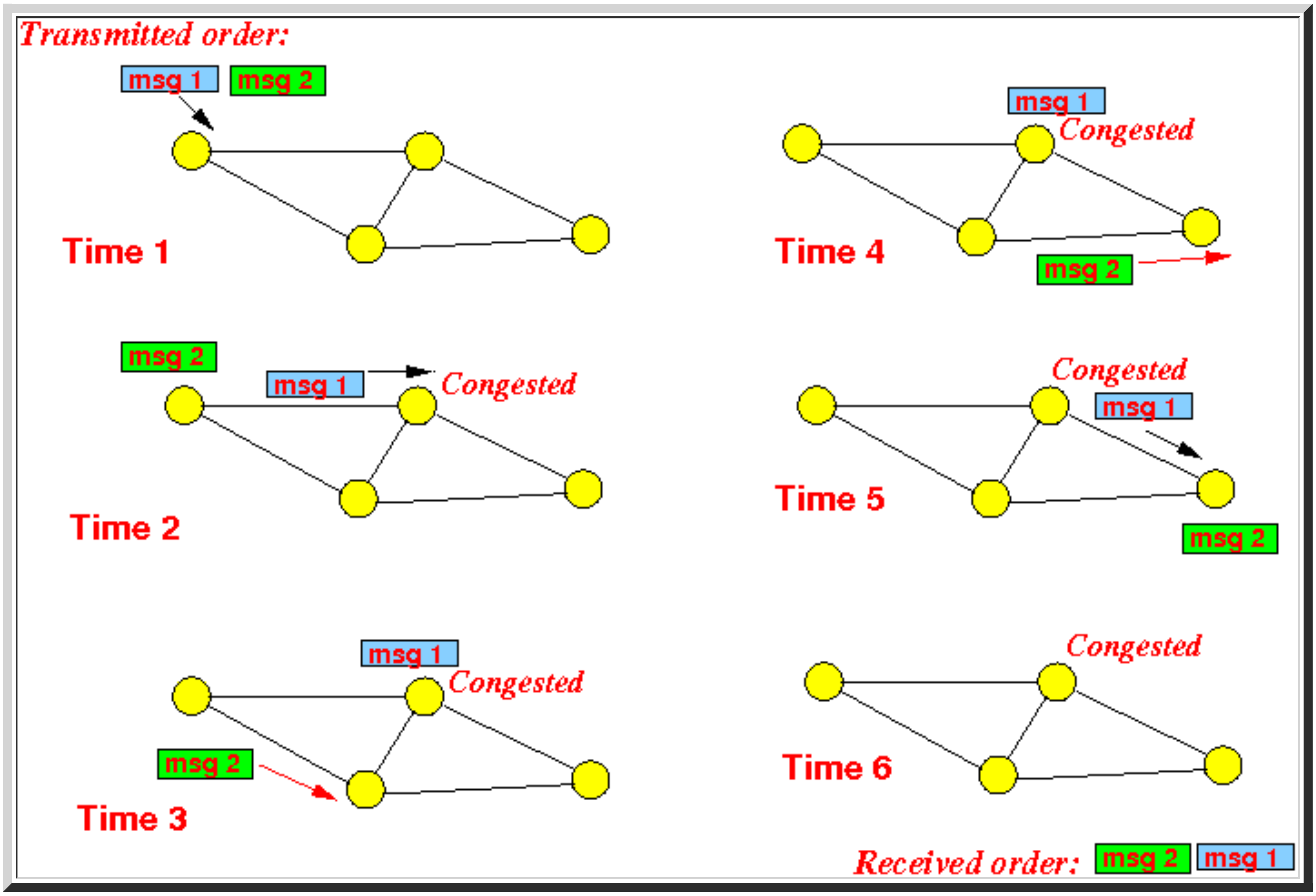
    - ***Different* messages/packets** can take ***different* routes** towards the **destination**

    - ***Notable* result:**:

      - **Messages/packets transmitted *later*** ***can*** arrive ***earlier*** than some **message/packet** that was transmitted **earlier** !!!!

because some **earlier messages/packets** were ***delayed*** due to **congestion**

Example: how a ***later* message** can **overtake** an ***earlier* message**



- **Terminology: datagram switching**
  - **Datagram:**
    - **Datagram** = a **unit** of **data transmission** on a **network**
  - **Datagram switching**:
    - **Datagram switching** = the **routing decision** is made on a ***per datagram* basis**
  - So a **datagram** can be:
    - A **message** **or**
    - A **packet**
- **Postscript....**
  - **Note:**

- **Message switching** is **no longer used** in today's network

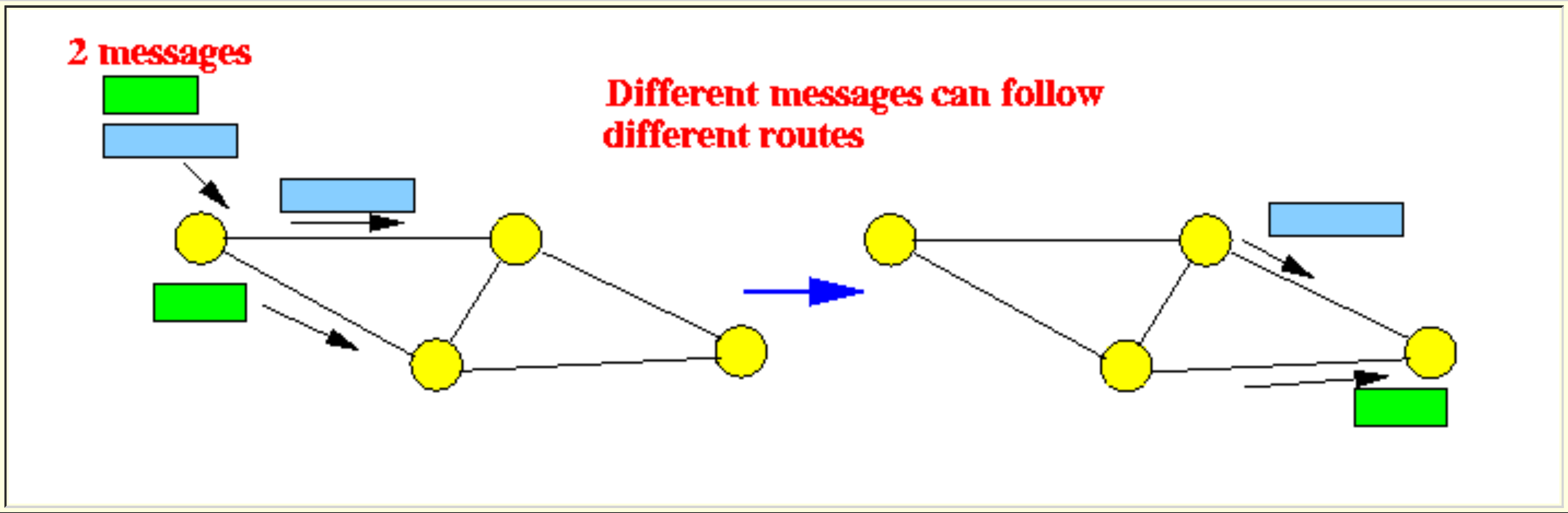
---

- **All communication networks** today **impose** some ***maximum* packet size**

- So **all** networks are are thus ***packet* switched** networks

Message Transmission and Performance of datagram service

- **Message/Packet Switching**
  - **Recall that:**
    - In **message/packet switching**, **different message/packet may traverse** a **different path (route)** through the network...



Undesirable consequence:

- Because the **nodes traversed** by the **packets** are not **determined *a-priori***:
  - We ***cannot*** make **resourse reservations** at the **nodes along the path**

    (because the set of **nodes** can **change**)

(**Resource reservations** can **provide *guarantees*** on **performance** !!!)

- **Analogy:**
  - ***If*** I let my **GPS** guide me to my **destination**, **then:**
    - The **GPS** will be able to **re-route** me to the ***fastest* route** to get to my **destination**

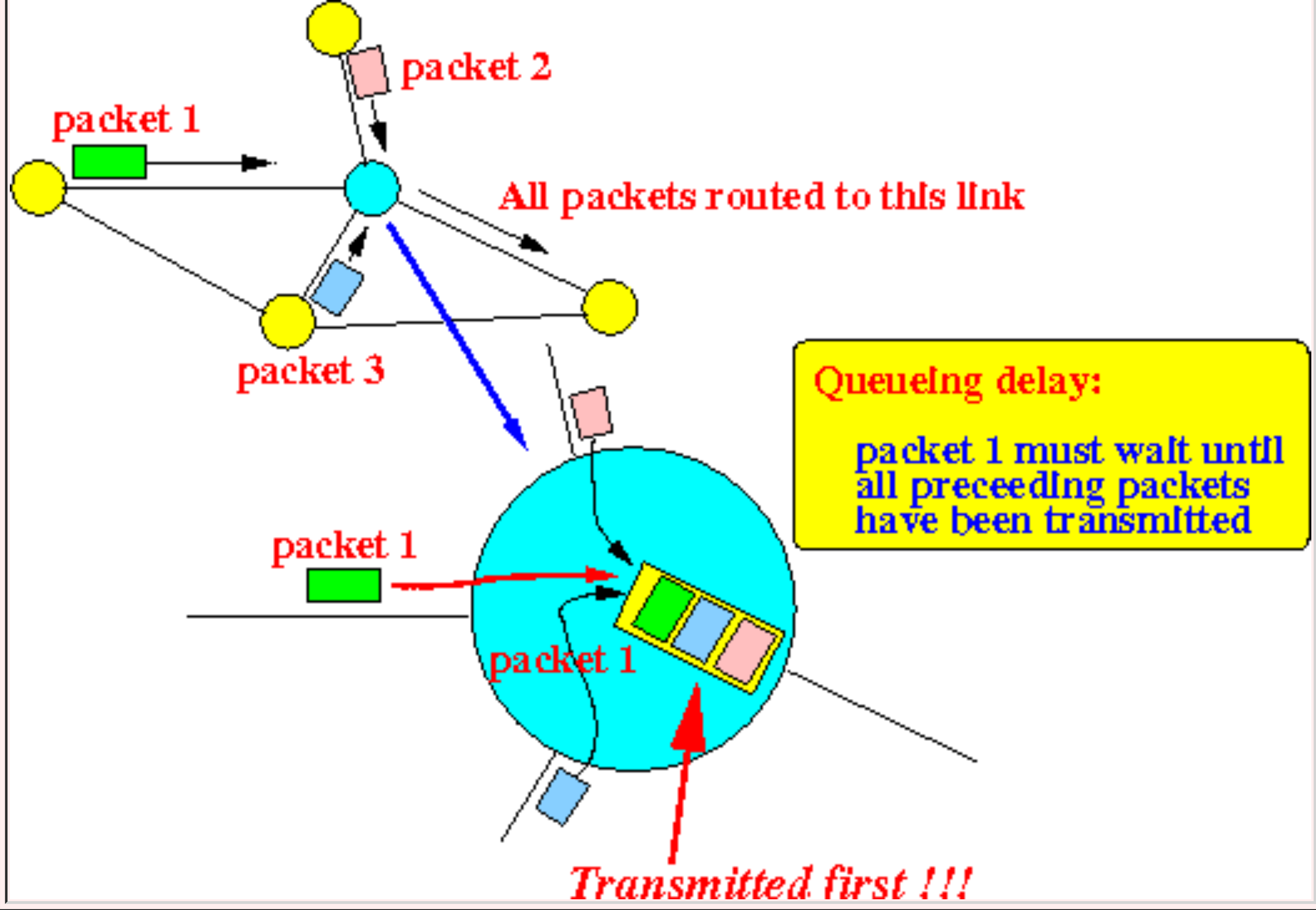
    - **However:**
      - I **cannot** make any **hotel reservation beforehand** !!!

        (Because I may **not** pass by the **hotel** where I made the **reservation** !!!)

- **The biggest headache in network performance: queueing delay**

Queueing delay:

- A **packet** must **wait for its turn** to be **transmitted**:



- **Queueing delay:**
  - **Queueing delay** of a **packet** = the **waiting time** from:
    - The **time** that the **packet arrives** at a switch, until
    - The **time** that the switch **starts transmitting** the **packet**

- **Important fact:**
  - The **queueing delay** can be **several order of magnitude (100x !!!) larger** that the **time needed** to **transmit *one* packet**: