

**Corso di Laurea in Ingegneria Informatica**



**Corso di Reti di Calcolatori I**

**Roberto Canonico ([roberto.canonico@unina.it](mailto:roberto.canonico@unina.it))**

**Giorgio Ventre ([giorgio.ventre@unina.it](mailto:giorgio.ventre@unina.it))**

**Il livello rete**

**I lucidi presentati al corso sono uno strumento didattico  
che NON sostituisce i testi indicati nel programma del corso**

**Nota di copyright per le slide COMICS**



## Nota di Copyright

Questo insieme di trasparenze è stato ideato e realizzato dai ricercatori del Gruppo di Ricerca COMICS del Dipartimento di Informatica e Sistemistica dell'Università di Napoli Federico II. Esse possono essere impiegate liberamente per fini didattici esclusivamente senza fini di lucro, a meno di un esplicito consenso scritto degli Autori. Nell'uso dovranno essere esplicitamente riportati la fonte e gli Autori. Gli Autori non sono responsabili per eventuali imprecisioni contenute in tali trasparenze né per eventuali problemi, danni o malfunzionamenti derivanti dal loro uso o applicazione.

**Autori:**

Simon Pietro Romano, Antonio Pescapè, Stefano Avallone,  
Marcello Esposito, Roberto Canonico, Giorgio Ventre

## Il livello rete



- Stack OSI



Percepisce dal livello inferiore un servizio di consegna di pacchetti tra host visibili sulla rete

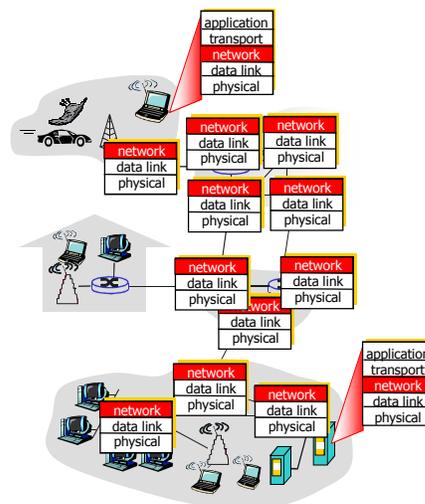
Offre al livello superiore un servizio di consegna di frame tra host fisicamente connessi

3

## Le funzioni del livello rete



- Trasportare i pacchetti dall'host mittente a quello ricevente
- Implementare protocolli di livello rete in *tutti* i router e in *tutti* gli host



4

## Le funzioni del livello rete



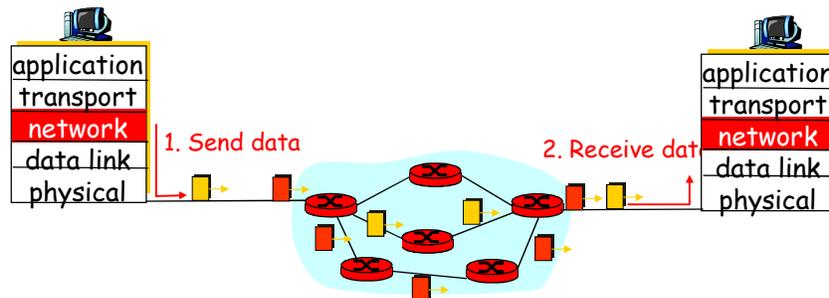
- Le reti possono essere classificate a seconda del metodo utilizzato per trasportare i pacchetti dalla sorgente alla destinazione:
  - **Reti a datagrammi**  
ogni pacchetto è instradato indipendentemente dagli altri pacchetti dello stesso flusso
  - **Reti a circuiti virtuali**  
viene precalcolato un percorso e tutti i pacchetti del flusso seguono questo percorso
  - **NB: parliamo comunque di reti a commutazione di pacchetto!**

5

## Packet switching: reti a datagrammi



- Ogni nodo che riceve un pacchetto decide in maniera **indipendente** a quale altro nodo inoltrarlo, sulla base dell'indirizzo destinazione contenuto nel pacchetto
  - **Indipendente** rispetto agli altri nodi
  - **Indipendente** rispetto agli altri pacchetti passanti per lo stesso nodo
- Pacchetti tra la stessa coppia sorgente-destinazione possono seguire percorsi differenti



6

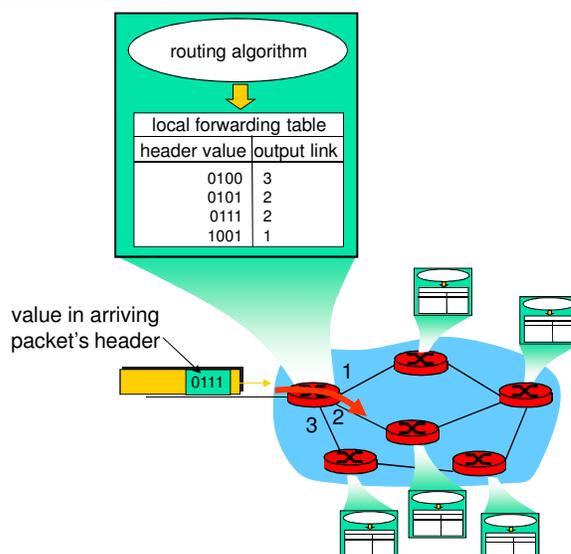
## Le funzioni di forwarding e routing



- **forwarding**: spostare i pacchetti dalla coda/interfaccia di ingresso a quella di uscita
  - **routing**: determinare la strada che un pacchetto deve seguire da una sorgente ad una destinazione:
    - *routing algorithms*
- analogia:**
- **routing**: processo di costruzione di un viaggio dalla partenza all'arrivo
    - A mano
    - Usando un software
    - ...
  - **forwarding**: processo di movimentazione ad ogni singola rotonda...

7

## La relazione tra forwarding e routing



8

## Forwarding table



4 miliardi ( $2^{32}$ ) di entry nella tabella di inoltro: occorrono soluzioni per compattarla !

	<u>Destination Address Range</u>	<u>Link Interface</u>
Da:	11001000 00010111 00010000 00000000	
A:	11001000 00010111 00010111 11111111	0
-----		
Da:	11001000 00010111 00011000 00000000	
A:	11001000 00010111 00011000 11111111	1
-----		
Da:	11001000 00010111 00011001 00000000	
A:	11001000 00010111 00011111 11111111	2
-----		
	Altrimenti ( <i>regola di default</i> )	3

## Longest prefix matching



<u>Prefix Match</u>	<u>Link Interface</u>
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
otherwise	3

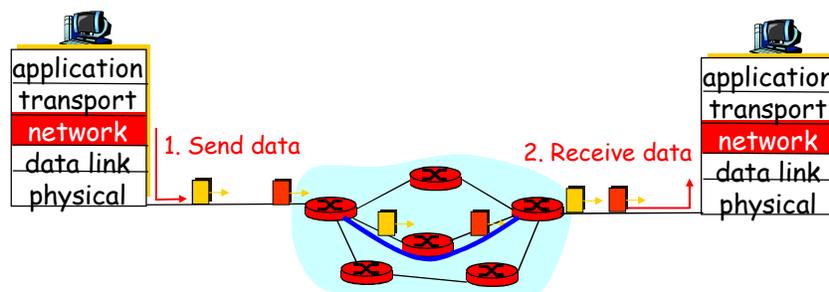
### Esempi

DA: 11001000 00010111 00010110	10100001	Quale interfaccia? 0
DA: 11001000 00010111 00011000	10101010	Quale interfaccia? 1
DA: 11001000 00010111 00011001	10101010	Quale interfaccia? 2

## Packet switching: reti a circuiti virtuali



- Ogni pacchetto contiene il numero del circuito virtuale
- Il circuito virtuale è stabilito prima della trasmissione dei dati
- I nodi devono conservare informazioni sui circuiti virtuali che li attraversano



11

## Circuito virtuale



Percorso associato ad una coppia di terminali che intendono comunicare

- Insieme di risorse assegnate in ciascun dispositivo intermedio (router) per il trattamento dei pacchetti associati al circuito virtuale
- Fase di **call setup** prima che i dati possano fluire attraverso la rete
- Ogni pacchetto associato ad un circuito virtuale porta nell'header un identificatore di circuito virtuale (VC)
- I router inoltrano i pacchetti sulla base del VC e non dell'indirizzo destinazione
- Ogni router attraversato dal circuito virtuale deve mantenere delle informazioni di stato per quel circuito
- E' possibile assegnare delle risorse (banda trasmissiva, buffer) a ciascun VC → possibilità di prestazioni predicibili

## Forwarding table (rete a circuiti virtuali)



- Un circuito virtuale è identificato da un numero VC su ogni tratto (link) del percorso
- Lo stesso circuito sarà identificato da numeri di VC diversi sui vari tratti che lo compongono

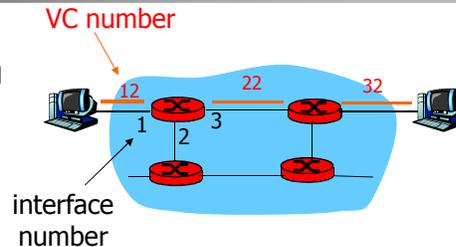


Tabella di forwarding nel router in alto a sx:

Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

## Datagrammi vs circuiti virtuali



Proprietà	datagrammi	circuiti virtuali
Creazione del circuito	Non richiesta	Richiesta
Indirizzamento	Ogni pacchetto contiene l'intero indirizzo della sorgente e della destinazione	Ogni pacchetto contiene un numero di VC
Informazioni sullo stato	I nodi di rete non mantengono informazioni sullo stato	Ogni VC richiede uno spazio di memoria sui nodi
Instradamento	Ogni pacchetto è instradato indipendentemente	Percorso pre-calcolato: ogni pacchetto segue questo percorso
Effetti di guasti ai nodi	Nessuno (solo i pacchetti persi durante il guasto)	Tutti i VC che attraversano quel nodo sono chiusi
Controllo di congestione	Complicato	Semplice se possiamo allocare spazio sufficiente per ogni VC

14