



# Modelli di comunicazione

Luca Veltri

(mail.to: luca.veltri@unipr.it)

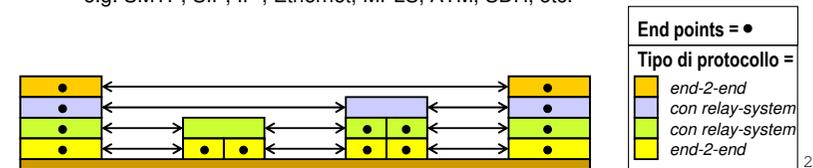
Corso di Reti di Telecomunicazione, a.a. 2013/2014

<http://www.tlc.unipr.it/veltri>

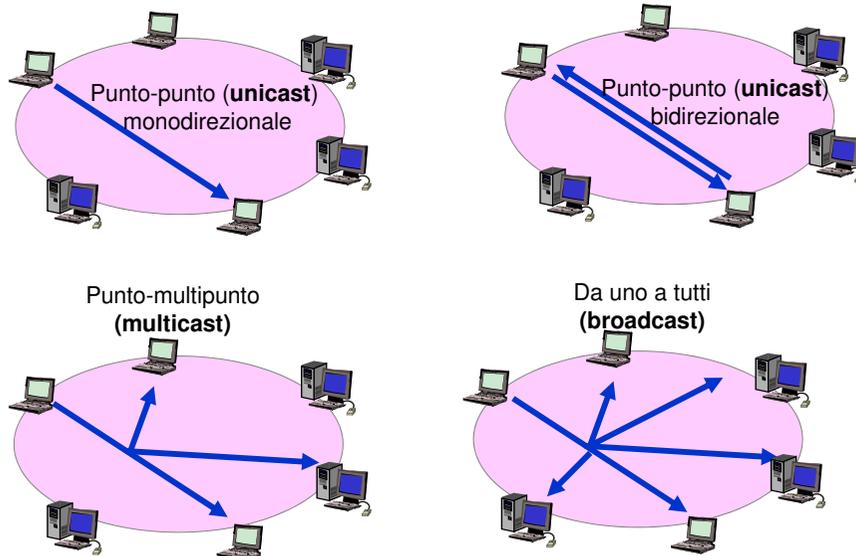


## Comunicazione End-to-end o Relayed

- Comunicazione da estremo a estremo (end-to-end)
  - **quando la comunicazione avviene direttamente tra entità sorgente e destinazione**
  - **non richiede l'indirizzamento di nodo ma solo dell'utente del servizio**
- Comunicazione rilanciata/commutata (relayed)
  - **quando la comunicazione avviene attraverso il rilancio di uno o più nodi intermedi di rilancio/commutazione (relay system)**
- Per estensione, un protocollo di comunicazione potrà essere di tipo:
  - **end-to-end**
    - e.g. POP3, Telnet, TCP, UDP, SLIP, PPP, LLC, AAL, etc
  - **relayed**
    - e.g. SMTP, SIP, IP, Ethernet, MPLS, ATM, SDH, etc.



## Comunicazione Unicast, Multicast, Broadcast



## Comunicazione Unicast, Multicast, Broadcast

- Punto-punto (Unicast)
  - **una sorgente e una destinazione**
  - **può essere uni o bi-direzionale**
  - **esempio: telefonia classica (fissa e mobile), circuiti dedicati, etc**
- Punto-multipunto (Multicast)
  - **trasmissione contemporanea da una sorgente a più destinazioni**
  - **esempio: servizi telefonici avanzati (telefonata a tre), IP multicast**
- Diffusiva (Broadcast)
  - **trasmissione contemporanea da una sorgente a tutte le possibili destinazioni/terminali**
  - **esempio: TV, LAN..**

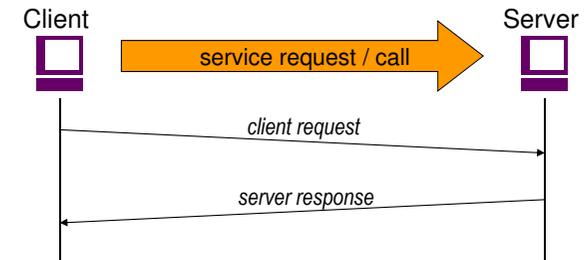
## Protocolli unicast, multicast, broadcast

- Alcuni protocolli supportano solo
  - la modalità unicast,
  - altri sia unicast che broadcast (e.g. Ethernet..)
  - altri ancora tutte e tre le modalità di comunicazione (unicast, multicast e broadcast) (e.g. IP)
- Le funzionalità di comunicazione broadcast e multicast di un protocollo possono essere realizzate
  - sfruttando eventuali funzionalità broadcast/multicast messe a disposizione dal protocollo sottostante (e.g. IP broadcast → Ethernet broadcast), oppure
  - utilizzando molteplici comunicazioni unicast a livello sottostante, eventualmente rilanciate da nodi intermedi

5

## Comunicazione Client-Server

- Comunicazione tra due entità (e.g. due terminali, applicazioni, etc.) in cui una delle due parti, il client, inizia una sessione di comunicazione verso l'altra parte, il server
  - il client inoltra le richieste (di servizio) ad un server (service request)
  - il server elabora le richieste e, se la richiesta è accettata, risponde al client fornendo il servizio richiesto (service response)
  - in genere la comunicazione risulta di tipo asimmetrica



6

## Comunicazione Client-Server (cont.)

- Definizione: Applicativo o processo terminale CLIENT
  - programma o sua parte che esegue la componente client di una comunicazione
  - è la parte "chiamante" (caller or calling party) o richiedente
  - può richiedere al server l'esecuzione di un servizio
- Definizione: Applicativo o processo terminale SERVER
  - programma o sua parte che esegue la componente server di una comunicazione
  - è la parte "chiamata" (callee or called party) o rispondente
  - può eseguire dei servizi e restituire al client il risultato
- Esempi di applicazioni client e server
  - browser (client) vs. server web (server)
  - client di posta elettronica (client) vs. POP3/SMTP server (server)

7

## Comunicazione Client-Server (cont.)

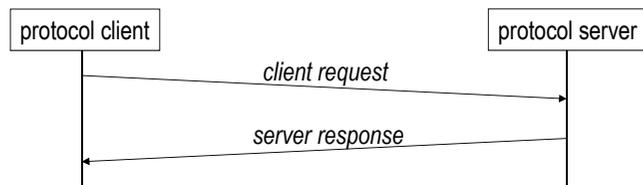
- Per estensione spesso si definisce
  - **Client**
    - dispositivo terminale (host, PC, workstation, etc.) che ospita principalmente applicazioni di tipo client
    - se le applicazioni ospitate hanno anche un lato server, spesso questo lato viene inibito (per configurazione, utilizzo o protezione)
  - **Server**
    - dispositivo terminale che ospita principalmente applicazioni di tipo server
- Alcune applicazioni implementano sia il lato client che il lato server di una comunicazione
  - e.g. SMTP server, HTTP proxy, etc.

8

## Protocolli Client-Server

- Analogamente alcuni protocolli si definiscono di tipo client-server quando la comunicazione tra le parti coinvolte evolve in modalità client-server

- in tal caso si può distinguere la parte client e la parte server del protocollo

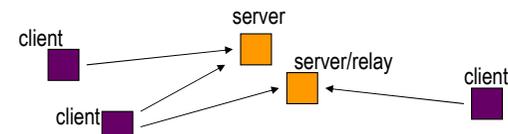


- alcuni protocolli prevedono che le due parti della comunicazione si possano scambiare la funzione client o server nell'arco della comunicazione stessa
- in alcuni casi la modalità client piuttosto che server caratterizza solo la prima parte della comunicazione delle due parti; il resto della comunicazione evolve in modo simmetrico (e.g. TCP)

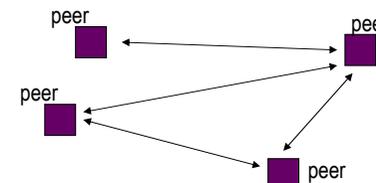
9

## Architettura Client-Server vs. Peer-to-Peer

- Una architettura di comunicazione si definisce di tipo client-server se distingue funzionalmente i suoi terminali tra quelli che operano in modalità client e quelli server



- Nel caso contrario si parla di architettura peer-to-peer (P2P), e i terminali vengono detti genericamente peer



10

## Architettura Peer-to-Peer

- I nodi di rete agiscono sia come client che come server
- Spesso non c'è neanche distinzione tra nodi intermedi (relay), se presenti, e nodi terminali
  - nodi omogenei, genericamente riferiti come peer
- Nel caso l'architettura preveda il rilancio della comunicazione attraverso peer (intermedi), viene creata una struttura di rete P2P basata sui peer stessi e sulla loro interconnessione
  - i peer collaborano insieme al mantenimento di questa struttura di rete
  - i nodi (peer) possono avere differenti caratteristiche di banda a disposizione, capacità di elaborazione e di memoria
  - in alcuni casi si fa distinzione tra differenti tipologie di nodi a seconda del loro ruolo all'interno dell'architettura di rete
    - e.g. nodi (o peer), super-nodi (o super-peer)

11

## Architettura Peer-to-Peer (cont.)

- Principali caratteristiche di una rete P2P
  - **Architettura di rete dinamica**
    - i nodi entrano a far parte (join) ed escono (leave) dalla rete
    - l'architettura di rete deve poter continuare a funzionare anche in presenza di tali variazioni dovute a ingressi e uscite (churn)
  - **Architettura di rete affidabile**
    - ridondanza delle risorse di comunicazione e dei servizi forniti
  - **Architettura di rete scalabile**
    - l'architettura di rete riesce a scalare all'aumentare del numero di peer
- Principali vantaggi
  - **affidabilità e resistenza ai guasti (fault-tolerance)**
  - **distribuzione automatica delle funzioni e servizi (self-organization)**
  - **scalabilità**

12

## Comunicazione CO e CL

A seconda delle modalità di gestione della comunicazione, è possibile definire due differenti modi di trasferimento dell'informazione:

- Senza connessione (CL, Connection Less)
  - un'unica fase temporale
  - assenza di negoziazione, indipendenza e autoconsistenza dei segmenti informativi
  - accordo solo tra N-utente e (N-1)-fornitore
- Con connessione (CO, Connection Oriented)
  - strutturazione in tre fasi temporali
  - accordo tra gli estremi della connessione
  - negoziazione dei parametri di trasferimento
  - una volta stabilita la connessione, il trasferimento delle UI avviene come attraverso un "tubo":  
le UI vengono inviate dal sorgente ed estratte in modo ordinato dal destinatario
    - uso di identificatori di connessione
    - è presente una relazione tra i segmenti informativi scambiati

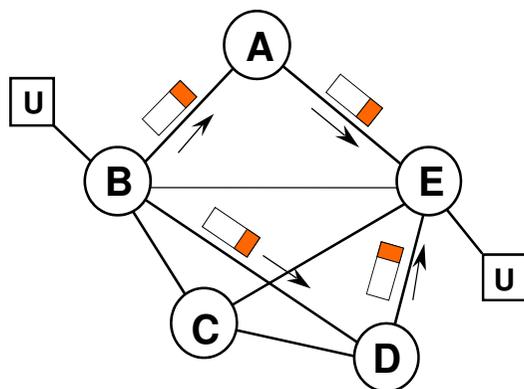
13

## Comunicazione CL (a datagramma)

- Nella modalità di trasferimento a DATAGRAMMA, il trasferimento delle UI (pacchetti) avviene senza accertare preventivamente la disponibilità dell'utente destinatario e/o delle risorse di rete all'effettuazione del trasferimento
- Non esistono le fasi di instaurazione e di abbattimento di una chiamata ed ogni pacchetto o datagramma (UI) è gestito dalla rete indipendentemente dagli altri, anche se fanno parte della stessa comunicazione
- Se sono presenti nodi di commutazione, questi operano la funzione di instradamento solo sulla base dei singoli pacchetti (UI)
- Ogni pacchetto è completamente indipendente
- E' possibile che i pacchetti vengano consegnati fuori sequenza

14

## Esempio di comunicazione CL (a datagramma)



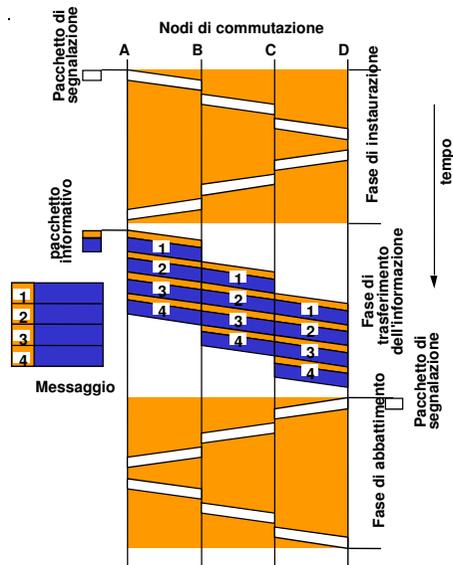
15

## Comunicazione CO

- Nella modalità di trasferimento con connessione, il servizio è articolato in tre fasi:
  - fase di instaurazione della connessione
  - fase di trasferimento informativo
  - fase di abbattimento della connessione
- Durante la fase di instaurazione normalmente viene:
  - a) accertata la possibilità di instaurare la connessione (controllo di accettazione di chiamata) ed eventualmente effettuata un'assegnazione logica delle risorse necessarie
  - b) assegnati opportuni identificatori della chiamata che saranno trasportati da tutte le UI appartenenti alla chiamata stessa (connessione)
  - c) nel caso di attraversamento di più nodi, viene determinato il cammino che i pacchetti seguiranno in rete (funzione di instradamento)
- Durante la fase di abbattimento vengono rilasciate le risorse precedentemente allocate

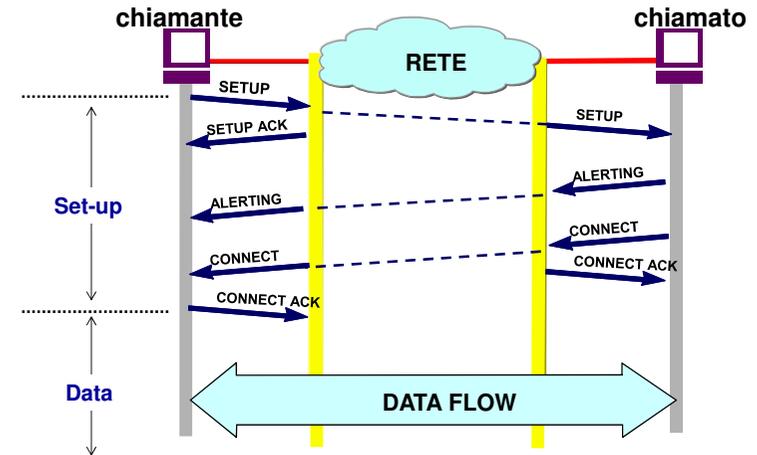
16

## Esempio di comunicazione CO



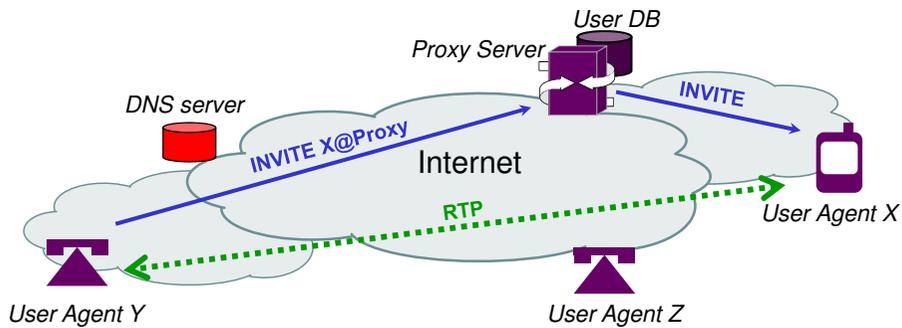
17

## Esempio CO: Chiamata Q.931



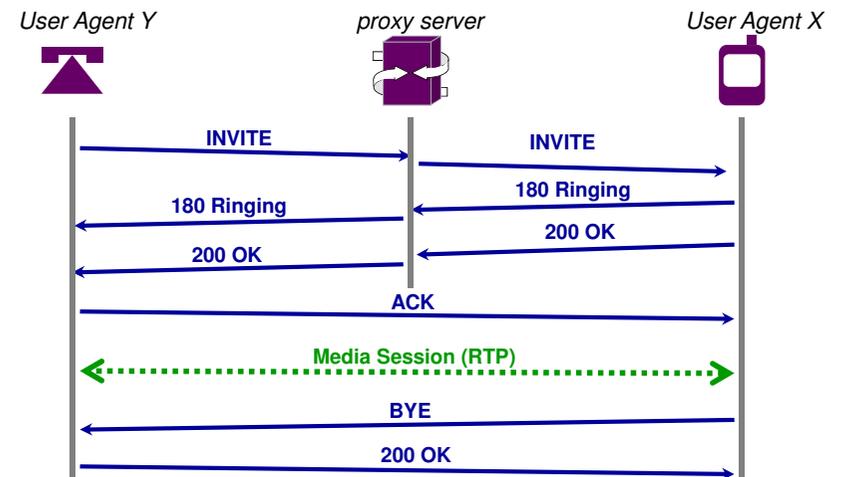
18

## Esempio CO: VoIP call setup (SIP)



19

## Esempio CO: VoIP call setup (SIP) (cont.)



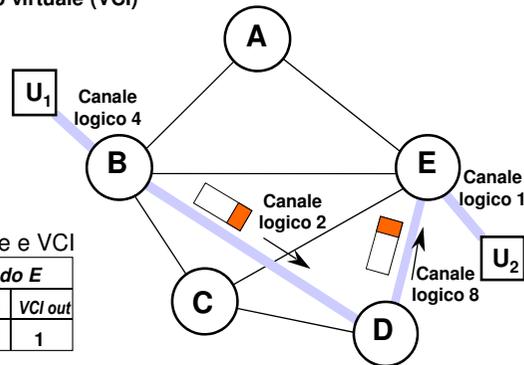
20

## Comunicazione CO

- Durante la fase di trasferimento le UI vengono trattate dai terminali e, eventualmente, dai nodi di rete sulla base della appartenenza delle stesse UI ad una specifica connessione (precedentemente instaurata)
  - a tal fine vengono considerati degli identificatori logici associati alle singole UI
  - tali identificatori vengono a volte chiamati numero di canale logico, o identificatore di circuito virtuale (VCI)

Esempio di corrispondenza tra circuito virtuale e VCI

Nodo B		Nodo D		Nodo E	
VCI in	VCI out	VCI in	VCI out	VCI in	VCI out
4	2	2	8	8	1



21

## Comunicazione CO

- In genere, durante la fase di trasferimento:
  - le UI preservano il loro ordinamento (sequenza)
  - rispetto a comunicazioni in modalità CL, la probabilità di perdita può essere minore (aumentando l'integrità informativa)
    - grazie ad eventuali meccanismi di recupero di errore
    - grazie ad eventuali meccanismi di riservazione/gestione delle risorse di rete
  - rispetto a comunicazioni in modalità CL, il ritardo end-to-end può essere minore (aumentando la trasparenza temporale)
    - nel caso lo strato CO esegua la funzione di commutazione (ovvero prevede nodi intermedi che operano in modalità CO)

22

## Comunicazione Affidabile e Non affidabile

- Una comunicazione viene detta "affidabile" se garantisce l'integrità informativa dei dati trasferiti
  - in particolare deve preservare sia l'integrità delle singole UI e sia il loro ordinamento
- Nel caso contrario, la comunicazione è detta "non affidabile"
- Se un protocollo offre un servizio di comunicazione di tipo affidabile, tale protocollo è detto *affidabile*
  - e.g. TCP, HTTP, SMTP, etc.
- Nel caso contrario il protocollo è detto *inaffidabile*
  - e.g. Ethernet, IP, UDP
- Il servizio di comunicazione affidabile può essere fornito grazie a ad apposite funzioni implementate dal protocollo stesso o grazie al servizio offerto dal protocollo sottostante
  - un esempio del 1° caso è il TCP
  - un esempio del 2° caso è HTTP (tramite servizio offerto da TCP)

23

## Comunicazione Real-time e Non real-time

- Una comunicazione è detta di tipo real-time (in tempo reale) se nel trasferimento preserva (o deve preservare) la trasparenza temporale
- La trasparenza temporale può essere garantita in termini di:
  - mantenimento/ricostruzione in ricezione della stessa cadenza temporale delle UI inviate
  - basso ritardo end-to-end
- Il primo punto può essere garantito attraverso opportune funzioni di riequalizzazione dei ritardi
  - soprattutto in caso di bassa variabilità (jitter) dei ritardi end-to-end
- Il secondo punto richiede specifiche caratteristiche di trasferimento nella rete attraversata
  - basso ritardo massimo end-to-end
- Ci sono protocolli più o meno adatti a comunicazioni real-time

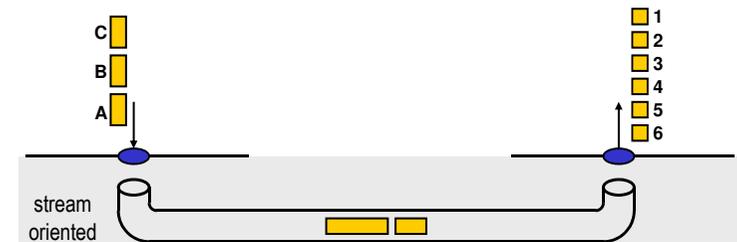
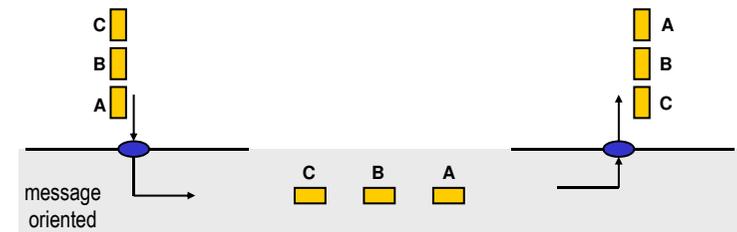
24

## Comunicazione Message o Stream oriented

- A seconda della modalità con cui i dati vengono inviati e ricevuti, si possono distinguere due modalità di comunicazione:
  - **Message oriented (orientata al messaggio)**
    - vengono inviati e ricevuti dallo strato superiore specifici blocchi di dati
    - in ricezione viene garantita la appartenenza dei dati ad un specifico blocco (messaggio)
    - e.g. HTTP, SMTP, UDP, IP, PPP, Ethernet, etc.
  - **Stream oriented (orientata al flusso)**
    - i dati dello strato superiore vengono gestiti come un flusso continuo (non necessariamente temporalmente continuo) di bit o byte
    - in ricezione viene garantita la medesima sequenza dei dati e riconsegnati allo strato superiore in ordine
    - i dati non vengono strutturati e delimitati in blocchi, ma al contrario vengono consegnati allo strato superiore come un unico flusso
    - e.g. SSH, TCP, SDH, PCM, e in generale la maggior parte degli strati PH
- La modalità di comunicazione a Stream richiede l'instaurazione di una "connessione virtuale" (comunicazione di tipo CO), mentre non è vero il viceversa

25

## Comunicazione Message o Stream oriented



26

## Information-Centric Networking (ICN)

- Nuovo paradigma di comunicazione che mette al centro l'informazione (i dati) che si vogliono scambiare
  - **al contrario delle architetture di rete tradizionali che offrono un servizio incentrato sulla comunicazione tra nodi terminali (indirizzamento, routing, sicurezza, etc.)**
- Questo nuovo approccio comporta un capovolgimento delle funzioni offerte da una rete e un ridisegno dei protocolli utilizzati
  - **vengono indirizzati i contenuti (i dati) e non gli host terminali**
  - **la comunicazione avviene secondo un paradigma di tipo richiesta(interest)/risposta(data), o anche publish/subscribe**
  - **i nodi della rete possono effettuare caching e diventare sorgenti per i dati (vantaggi: prestazioni, multicast, mobilità)**
  - **la sicurezza viene legata direttamente ai dati e non ai nodi che li gestiscono**
- Riferito anche come Data-centric o Data-oriented networking, oppure Content-Centric Networking (CCN)

27