



Università
degli Studi
della Campania
Luigi Vanvitelli

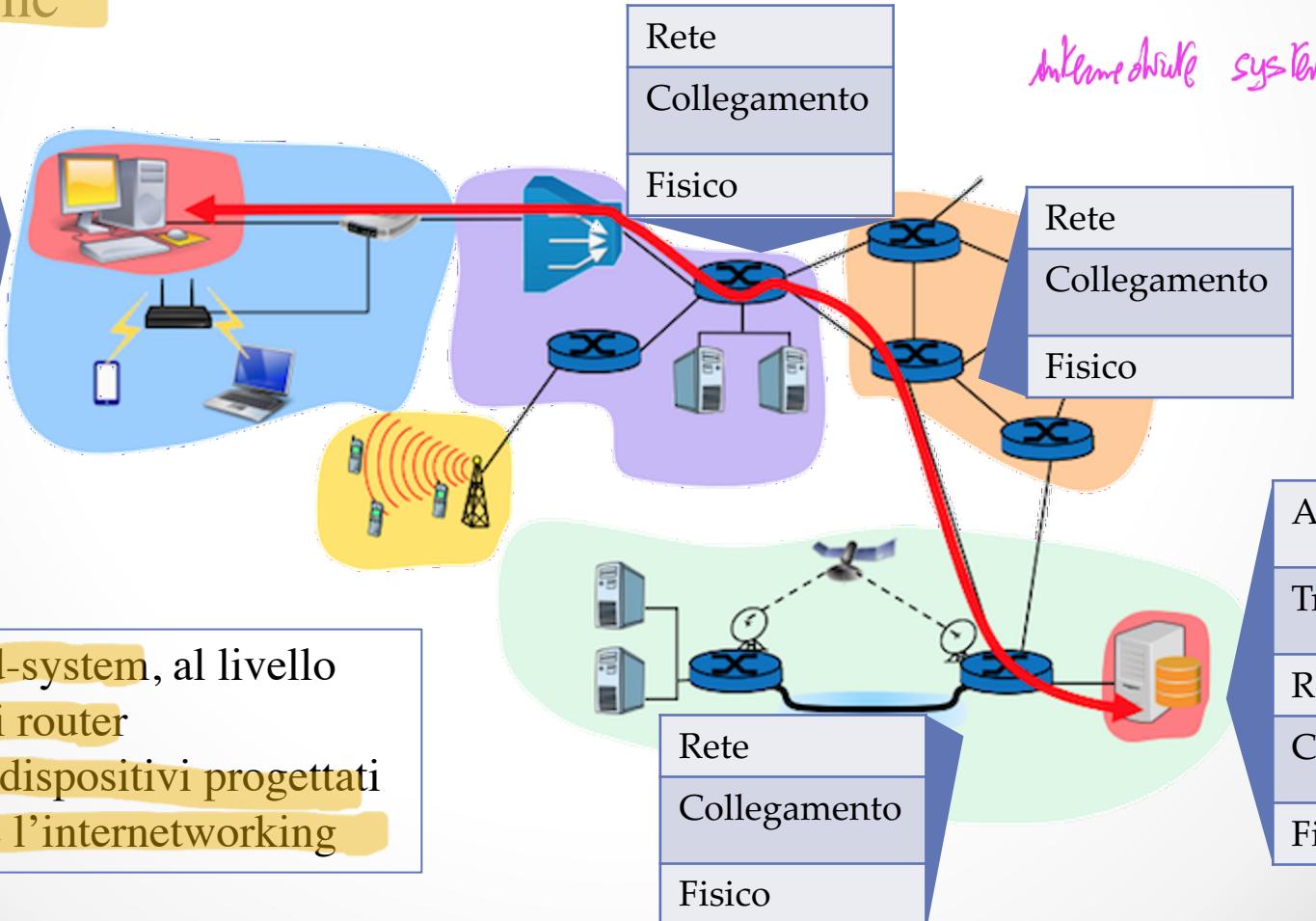
Reti di Calcolatori e Cybersecurity

Livello rete - IPv4

Ing. Vincenzo Abate

Livello rete

In una rete di computer ottenuta attraverso la interconnessione di reti distinte (internetwork), il compito del livello rete è quello di definire i percorsi dei pacchetti nel loro transito da host mittente a host destinazione



Livello rete – inoltro e instradamento

Due funzioni principali:

Inoltro (forwarding) – Quando un router riceve un pacchetto lo deve trasferire sull'appropriato link in uscita. Un pacchetto può anche essere bloccato (se ad esempio inviato da un mittente malevolo o destinato a un host vietato) o duplicato e inviato su più link in uscita.



Piano Dati

Instradamento (routing) - Il livello rete deve determinare il percorso che i pacchetti devono seguire tramite algoritmi di instradamento (alg. di routing).



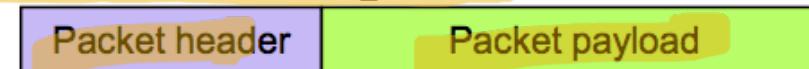
Piano controllo

Packet switching

Le reti di calcolatori operano secondo il modello detto **packet switching** o commutazione di pacchetto

In una rete a commutazione di pacchetto l'informazione è trasmessa in pacchetti formati da una intestazione (header) ed un payload

L'header contiene informazioni di controllo, tra le quali un indirizzo destinazione che serve ad identificare il terminale a cui il pacchetto deve essere consegnato



In uno stack protocolare, ciascun layer aggiunge un suo header (L2 anche un trailer); la struttura risultante è la seguente:

Dati link



I dispositivi intermedi che operano al livello rete funzionano in una modalità detta **store-and-forward**

- ogni pacchetto è ricevuto interamente, se ne controlla l'assenza di errori e se ne opera la ritrasmissione su un link di uscita
- all'interno dei dispositivi intermedi, i pacchetti sono mantenuti in buffer di memoria gestiti come delle code

Packet switching

- In una rete a commutazione di pacchetto **basata sul modello a datagram**, ciascun **pacchetto è inoltrato verso la sua destinazione indipendentemente dagli altri**
- **Ogni volta che un pacchetto arriva ad un dispositivo intermedio che opera al livello rete** (cioè un **router**), il dispositivo **inoltra il pacchetto verso un successivo dispositivo intermedio** (o verso il **destinatario finale del pacchetto**, qualora esso sia direttamente raggiungibile)
- **Pacchetti inviati da un terminale A verso un terminale B** in momenti **successivi possono seguire percorsi differenti** nella rete e, quindi, **arrivare a destinazione in ordine diverso da quello con il quale sono stato trasmessi**
E' possibile che dei pacchetti non arrivino a destinazione

QoS

Il servizio offerto da una rete a commutazione di pacchetto consiste nel recapitare pacchetti da un qualunque terminale mittente ad un qualunque terminale destinatario

La Qualità del Servizio (QoS) di una rete a commutazione di pacchetto è misurata da una molteplicità di "indici di prestazione"

Relativamente alla comunicazione tra due terminali collegati ad una rete, i parametri di QoS più comunemente utilizzati sono:

End-to-end delay: ritardo nella consegna dei pacchetti [s]

Packet delay variation (PDV): variazione temporale del ritardo one-way (anche indicata con il termine **packet jitter**)

Throughput: quantità di bit al secondo che la rete è in grado di trasferire tra i due terminali [b/s]

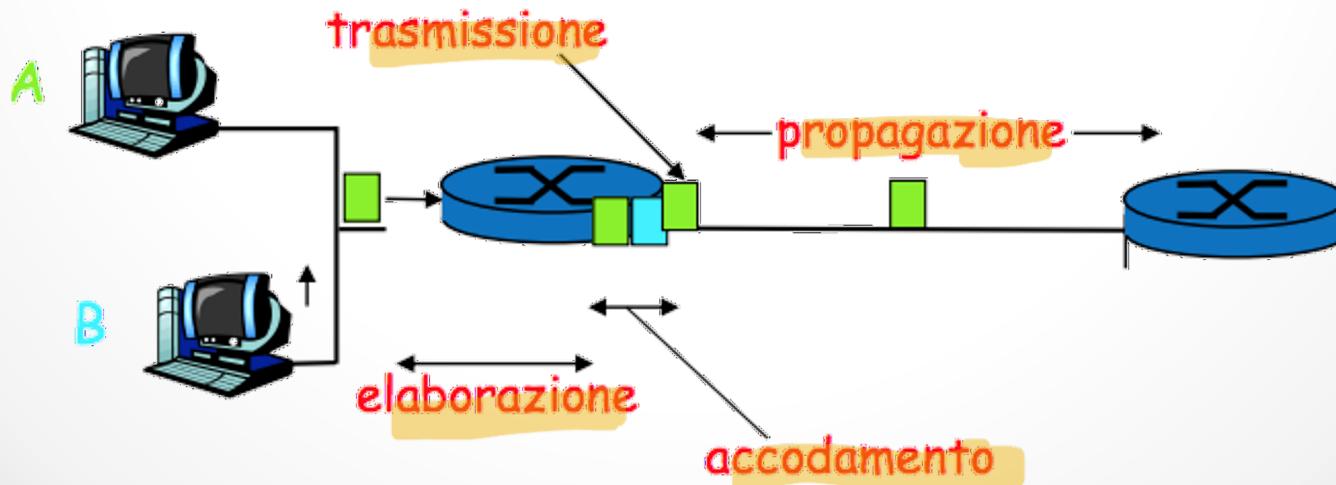
Loss-Rate: probabilità che un pacchetto non venga consegnato a destinazione



Ritardo in reti packet switching

Il ritardo nella consegna di un pacchetto alla destinazione è determinato da:

- **Tempo di elaborazione nel nodo:**
 - controllo di errori, determinazione link di uscita, ...
- **Tempo di trasmissione su ciascun link** = Lunghezza in bit/velocità in bps
- **Tempo di attesa nelle code dei router** (variabile)
- **Tempo di propagazione sulle linee** = lunghezza della linea/velocità del segnale



IP – Internet Protocol

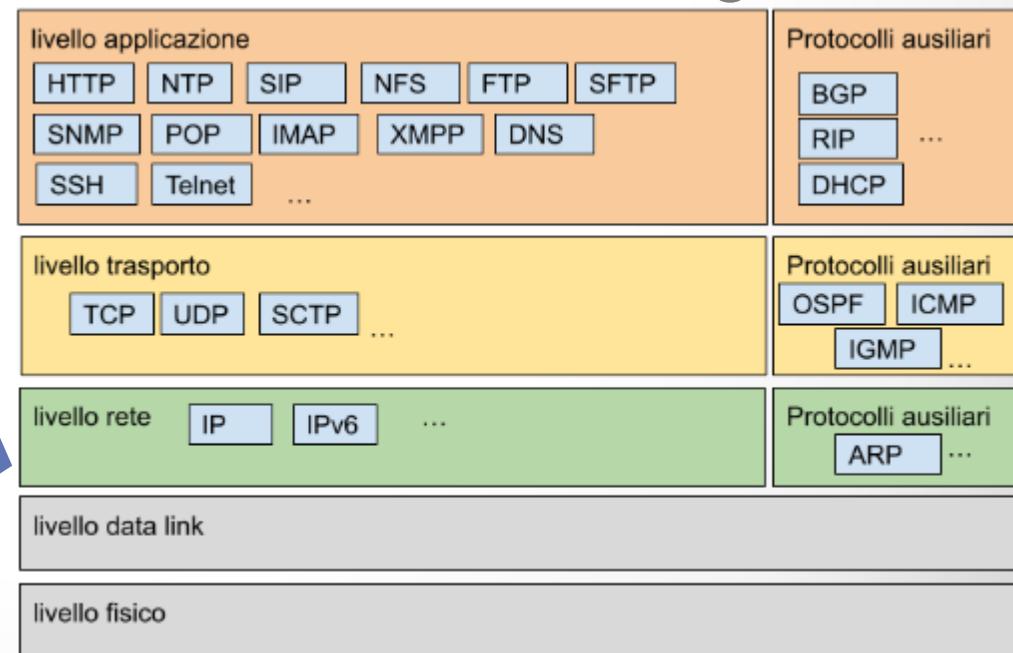
Nella rete Internet, la funzione principale del livello rete è svolta dal protocollo IP

La versione ancora oggi prevalentemente utilizzata è la versione 4 del protocollo IP

IP versione 6 è progressivamente introdotto ed utilizzato

La caratteristica principale del protocollo IP è quella di offrire un servizio di consegna elementare e senza garanzie (best effort) di pacchetti

La semplicità rende IP adattabile ad un'ampia varietà di tecnologie di livello inferiore



L'IETF

- La rete Internet è una “rete di reti” basata su standard aperti
- I protocolli di comunicazione utilizzati nei livelli Rete, Trasporto ed Applicazione in Internet sono definiti da una comunità aperta di esperti detta **Internet Engineering Task Force (IETF)**
- L'IETF è organizzata in gruppi di lavoro (working groups) che operano soprattutto tramite mailing list, aperte alla partecipazione di chiunque sia interessato
- Tre volte l'anno l'IETF organizza dei meeting plenari
- I gruppi di lavoro si occupano ciascuno di uno specifico argomento e sono organizzati in aree (protocolli applicativi, sicurezza, ecc...)
- Ogni gruppo produce dei documenti detti RFC (Request For Comments) che vengono sottoposti alla **IESG (Internet Engineering Steering Group)** per il loro avanzamento a standard ufficiale
- Prima di arrivare allo stato di RFC i documenti condivisi nei working group sono denominati **Internet Draft (I-D)**

IPv4

- IPv4 definito in RFC 791 (settembre 1981)
- IP realizza un servizio di consegna best-effort di pacchetti singoli (datagram)
- Al di sopra di IP, nello stack TCP/IP (Internet Protocol Suite), operano i protocolli di livello trasporto (UDP e TCP)
- Il protocollo IP gestisce indirizzamento, frammentazione, ri-assemblaggio e multiplexing dei protocolli
- È implementato sia negli end-system (terminali) che nei router
- È responsabile dell'instradamento dei pacchetti, cioè della scelta dell'interfaccia sulla quale un pacchetto deve essere trasmesso per arrivare a destinazione

IPv4

IP **NON** garantisce di prevenire:

- pacchetti duplicati
- consegna ritardata o fuori ordine
- corruzione di dati
- perdita di pacchetti

La consegna affidabile dei messaggi alle applicazioni può avvenire grazie a meccanismi di controllo realizzati nei protocolli di livello superiore (negli end-system)

Ogni router che riceve un pacchetto IP decide a quale altro nodo inoltrarlo, sulla base dell'indirizzo destinazione contenuto nel pacchetto, in maniera indipendente ...

- rispetto agli altri router
- rispetto agli altri pacchetti passati in precedenza per lo stesso router

Il protocollo IP è stato progettato per realizzare un servizio best-effort

Servizio best-effort significa che la rete

- non fornisce alcuna garanzia sulla consegna di un pacchetto
- ma non discrimina un pacchetto rispetto ad altri → network neutrality

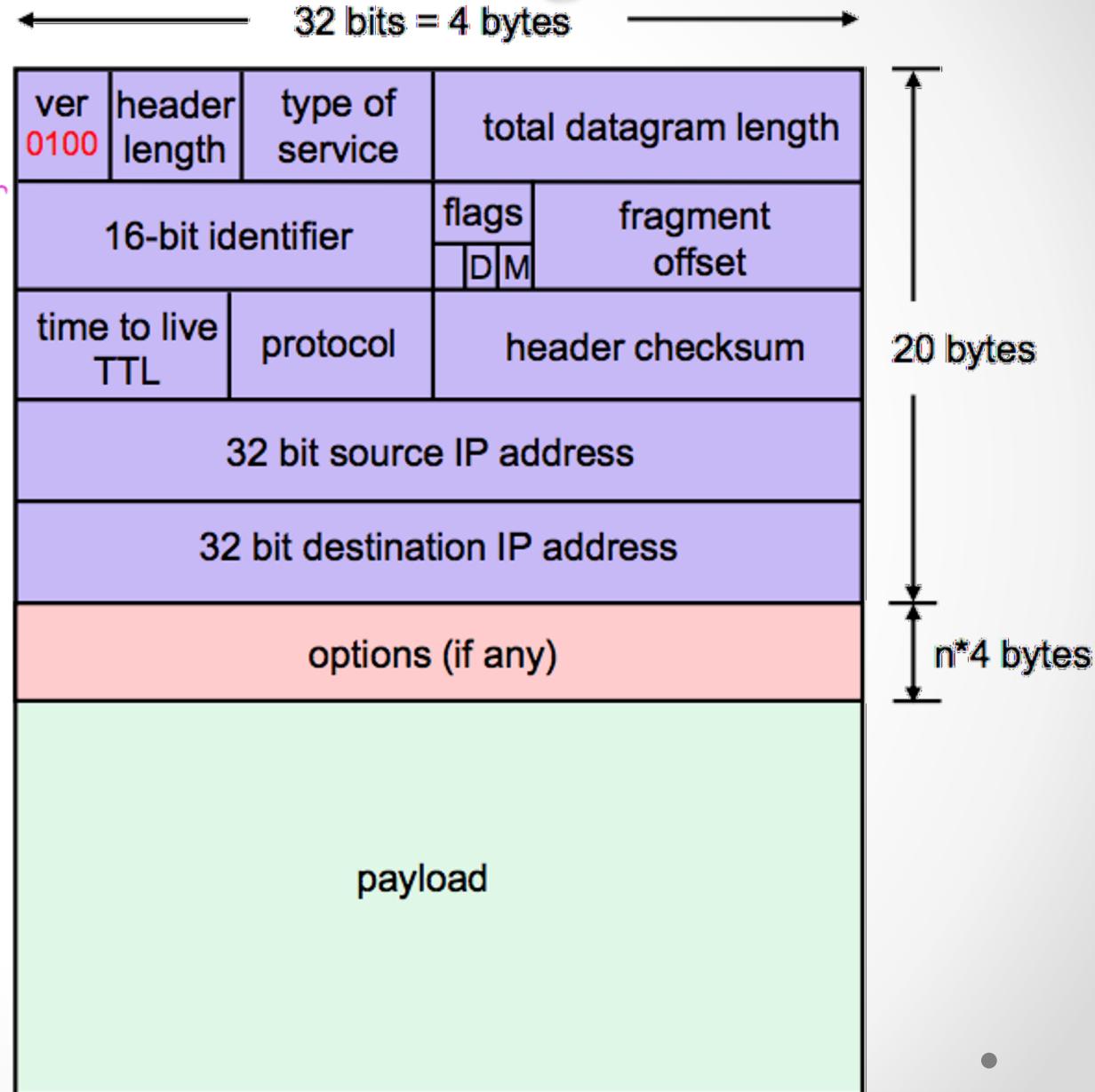
IPv4

- Un datagramma IPv4 può avere una dimensione massima di 65535 byte ($2^{16} - 1$) ed è costituito da un header ed un payload
- In IPv4 l'header è costituito da una parte a struttura fissa (20 byte) ed una opzionale
- Il payload è creato di norma da un protocollo di trasporto (TCP o UDP)
- In circostanze particolari, il payload di un pacchetto IP può contenere un altro pacchetto IP: **incapsulamento IP in IP**
- Alcuni protocolli ausiliari (cioè non intesi a supportare la comunicazione di applicazioni eseguite nei terminali) inviano i loro messaggi inserendoli direttamente in un payload IP: **ICMP, IGMP, OSPF**

IPv4 – struttura datagramma

L'header è costituito da una parte a struttura fissa (20 byte) ed una opzionale di lunghezza multipla di 4 byte

Riempiono un datagramma maggiore



IPv4

IP header length (4 bit): lunghezza dell'header, in multipli di 32 bit (max 60 byte)

Type-of-Service (8 bit): specifica il tipo di servizio che si richiede alla rete usato, in pratica, per scopi differenti

Total length (16 bit): indica la lunghezza in byte dell'intero pacchetto (header+dati)

Time-to-live TTL (8 bit): numero residuo di router attraversabili viene decrementato di 1 da ogni router, a 0 il pacchetto viene scartato serve, in caso di percorsi circolari (**loop**), ad evitare che un pacchetto resti perennemente in circolo

Protocol (8 bit): indica il protocollo di livello superiore associato al payload il valore 6 indica TCP, 17 indica UDP serve al de-multiplexing dei pacchetti a destinazione

Header checksum (16 bit): serve a verificare l'integrità dell'header IP

Source IP Address (32 bit): indirizzo IP del nodo mittente del pacchetto

Destination IP Address (32 bit): indirizzo IP del nodo destinatario del pacchetto

Identification (16 bit), Flags (3 bit), Fragment Offset (13 bit):

sono usati in caso di frammentazione del pacchetto da parte di un router e consentono al nodo destinatario di ricostruire il pacchetto originario

IPv4 - frammentazione

- Un pacchetto IPv4 può essere "spezzato" da un router in una sequenza di pacchetti che singolarmente viaggiano verso il destinatario
- Il livello IP del destinatario finale si occupa del "riassembaggio" del pacchetto originario prima di consegnarlo allo strato superiore
- Un pacchetto può essere frammentato anche più volte lungo il percorso
- La necessità di frammentare un pacchetto si presenta quando la dimensione del pacchetto supera la Maximum Transmissible Unit (MTU) sul link di uscita
- Il valore di MTU dipende dalla tecnologia usata al livello 2
Es. in Ethernet la MTU è 1500 byte

Identification

Questo campo (16 bit) è un identificativo del datagramma
Serve ad associare diversi frammenti ad un unico pacchetto originario

Flags

↳ tutti i frammenti hanno lo stesso ID

Il bit D (don't fragment) indica se il pacchetto può essere frammentato

Il bit M (more fragments) indica se il pacchetto è l'ultimo frammento

Fragment offset

13 bit, identifica la posizione del frammento all'interno del pacchetto

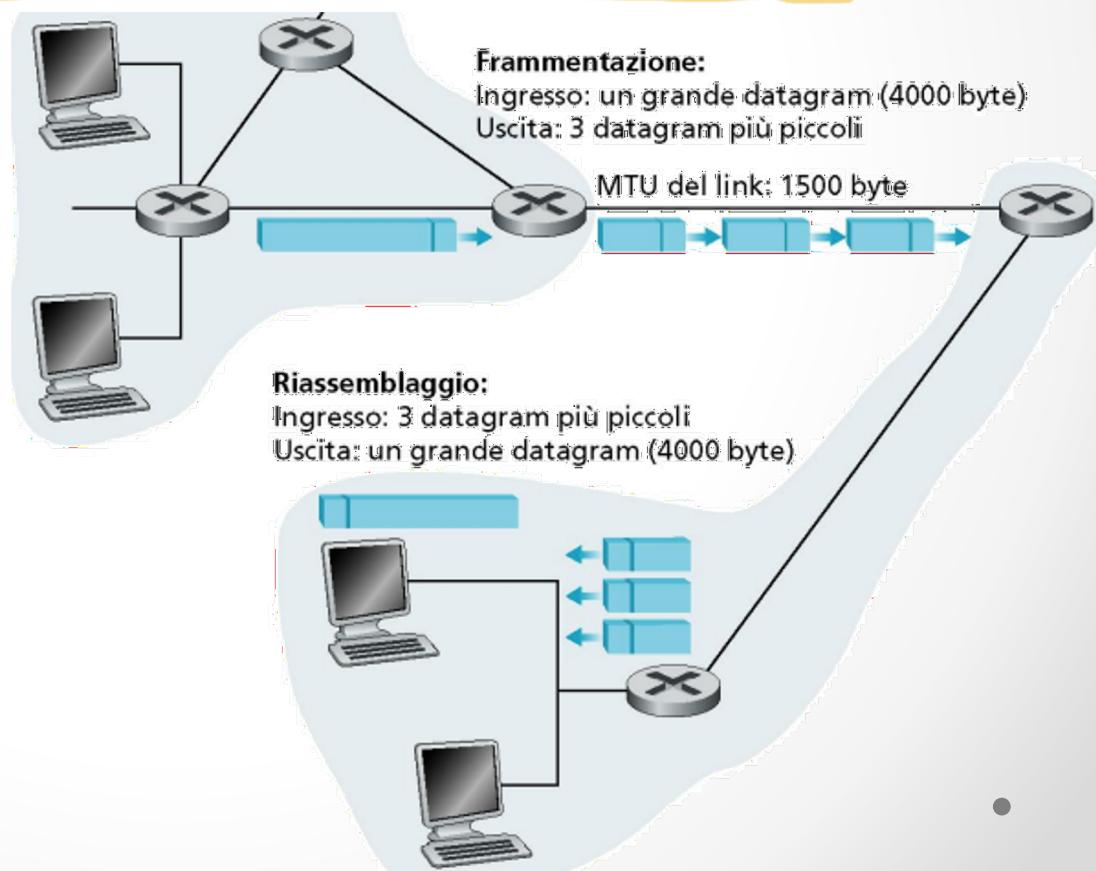
Frammentazione e riassemblaggio

Se un pacchetto di dimensione N arriva ad un router e deve essere trasmesso su un link di uscita con MTU $M < N$, il pacchetto è frammentato

Ogni frammento è trasmesso come singolo pacchetto IP

La dimensione del payload di ogni frammento è un multiplo di 8 byte

Tutti i frammenti hanno lo stesso ID number



Frammentazione e riassemblaggio

- Tutti i frammenti (tranne l'ultimo) hanno un payload di dimensione multipla di 8 byte
- Essendo la dimensione massima di un datagramma 65535 byte, ci possono essere al massimo $65535/8$ cioè 8192 frammenti per ogni datagramma
- La posizione del payload di un frammento rispetto al payload del pacchetto originario è espressa mediante un offset (spiazzamento) di 13 bit

Frammentazione esempio

N = 4000, MTU = 1500

Tre frammenti, ciascuno con header 20 byte

Frammento 1:

- payload 1480
- offset 0

Oggetto dei Frammenti
ha il suo header

Frammento 2:

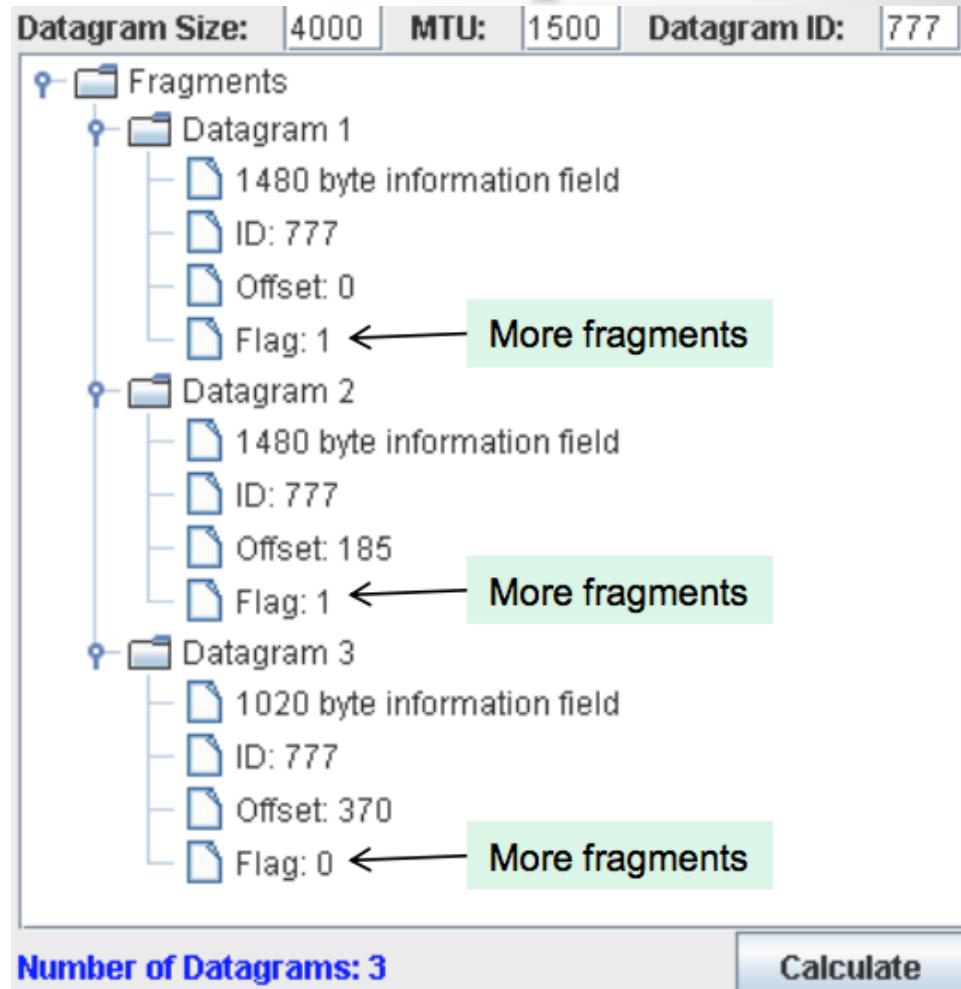
- payload 1480
- offset $(1480/8)=185$

Frammento 3:

- payload 1020
- offset $(1480+1480)/8=370$

NOTA:

$$20 + 1480 + 1480 + 1020 = 4000$$



Frammentazione esempio

Il pacchetto IP raffigurato di seguito deve attraversare un link avente Maximum Transfer Unit (MTU) pari a 1500 bytes. Come verrà trattato?

Datagramma

Sequence	Identifier	Total Length	DF May / Don't	MF Last / More	Fragment Offset
0	345	5140	0	0	0

Fragment

Sequence	Identifier	Total Length	DF May / Don't	MF Last / More	Fragment Offset
0-0	345	1500	0	1	0
0-1	345	1500	0	1	185
0-2	345	1500	0	1	370
0-3	345	700	0	0	555

<https://fixmycode.github.io/IPFCalc/>

↑ 1480 + 20 header → ogni frammento (datagramma) ha
1460 byte payload.

Frammentazione - Problemi

Il compito di riassemblaggio è oneroso

Il destinatario deve collezionare tutti i frammenti del pacchetto originario prima di consegnare il payload al livello superiore

Se non termina entro un determinato tempo, tutti i frammenti arrivati sono scartati

Può essere una tecnica per attaccare un host bersaglio

Per evitare la frammentazione dei pacchetti lungo il percorso, talvolta si effettua un **path MTU discovery**, cioè si determina la più piccola MTU lungo il percorso da un host A ad un host B

Conoscendo il path MTU, A evita del tutto la frammentazione se invia pacchetti di dimensione minore a tale valore

Un esempio di path MTU discovery

- A invia un pacchetto ICMP echo request a B di massima dimensione con flag D=1
- Se il pacchetto incontra sul percorso un router che non riesce a trasmettere il pacchetto, A riceve un messaggio ICMP "Destination unreachable: Fragmentation needed"
- A dimezza la dimensione e ritrasmette, se riceve da B l'echo reply incrementa la dimensione di un quarto, altrimenti dimezza

E così via fin quando non trova la dimensione adatta...

→ la faccio perdere 1000 anni a rassettare

IPv4 – Opzioni Header

L'header IP può essere esteso con dei campi "Opzione" mediante i quali si intende chiedere una elaborazione "speciale" del pacchetto da parte dei router

- Security
- Source routing
- Route recording
- Stream identification
- Timestamping

Per la presenza delle opzioni, l'header IP può essere di lunghezza variabile

Questo è il motivo della presenza del campo Header Length
Se l'opzione non occupa 4 byte (o un suo multiplo), vengono inseriti dei bit di riempimento (tutti zero) ↗ Parte di molte su richiesta da user
Nei router in cui il dataplane è implementato in hardware, l'elaborazione di questi campi non è effettuata in hardware (**fast path**) ma in software (**slow path**), oppure questi campi sono ignorati

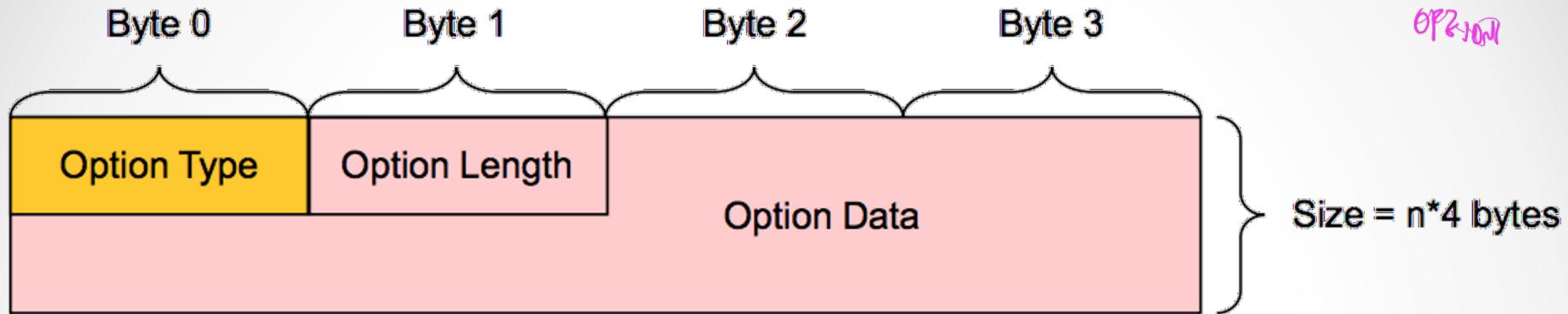
↳ gestione opzioni è fatta via software

Gli attacchi DoS di tipo "Christmas Tree" consistono nel trasmettere pacchetti IP con diverse opzioni (inutili) nell'header al fine di sovraccaricare i router

IPv4 – Opzioni Header

Nuv view
SRP/BRS

OPZIONI



Option Type byte

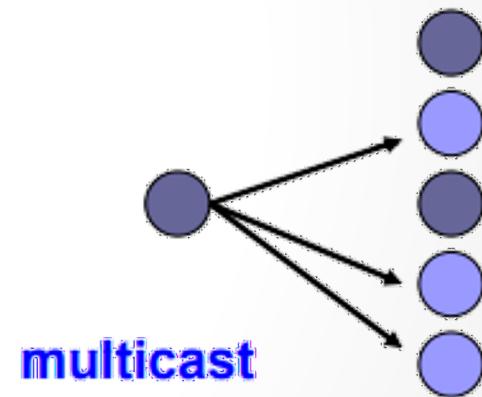
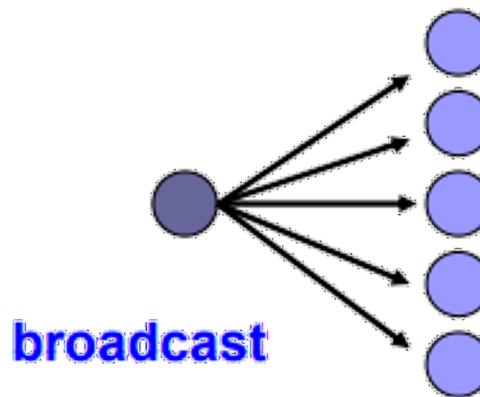
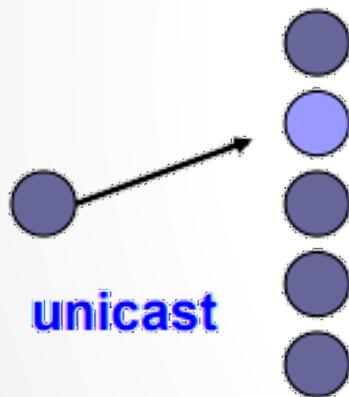
Ci sono tipo di option quale è lungo e dà

Subfield Name	Size (bits)	Description	
Copied	1	If 1: Option to be copied in all fragments If 0: Option only kept in first fragment	
Option Class	2	0: Control Options 2: Debugging/Measurement	1: Unused 3: Unused
Option Number	5	Up to 32 different Options for each class	

IPv4 – servizi

■ IP supports the following services:

- one-to-one (unicast)
- one-to-all *→ a tutti gli elementi della rete* (broadcast)
- one-to-several *es. sottoservizio di broadcast* (multicast)



- IP multicast also supports a many-to-many service.
- IP multicast requires support of other protocols (IGMP, multicast routing) *Routi si avranno algoritmi di routing multicast.*