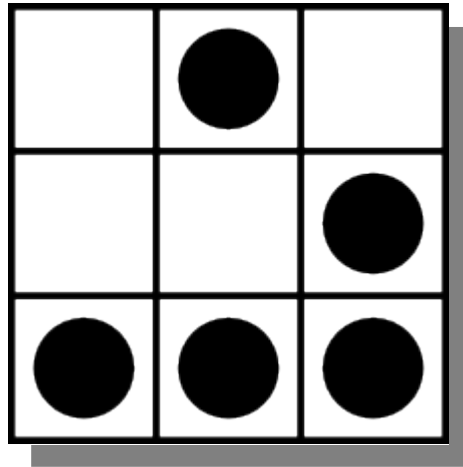


Denial of Service Attacks



Giacomo Rizzo [a.k.a. alt-os]
alt-os@poul.org



DoS: “Denial of Service”

Il nome in se, indica le conseguenze, non il metodo di attacco.

Il termine DoS è spesso utilizzato a sproposito, e nell'immaginario collettivo indica una singola metodologia di attacco.

Con questa presentazione, che non vuole assolutamente essere tecnicamente completa e/o esaustiva, andremo a fare una panoramica sugli attacchi DoS e DDoS, sulle metodologie più comuni, e sulle possibili precauzioni da prendere.

Significato

- DoS: “Denial of Service”

Letteralmente, si traduce con “Negazione di servizio” ma non rende l'idea. Come al solito, le traduzioni letterali dall'Inglese perdono in quanto a significato.

Ad ogni modo, gli attacchi di tipo DoS, sono volti a rendere materialmente impossibile fornire un servizio. Si dovrebbe infatti parlare di “DoS nei confronti di un servizio”.

- Server web
- Server di posta
- Carrier e providers
- Autostrade? L'esodo è un tipico esempio di DoS

Tipi di DoS

Tre grandi categorie:

- **Nuke**

Si sfrutta un bug del software, per causare un'interruzione nel servizio.

- **Flood**

Si impedisce l'usufrutto del servizio, “rubando” tutte le risorse disponibili.

- **Rose**

Attacchi piuttosto recenti (la teoria applicabile risale all'aprile 2004).

Nuke

Esempio di Nuke: TearDrop

- Autore: route|daemon9
- Piattaforme colpite:
 - Windows 3.1/95/NT
 - Linux < 2.0.32, < 2.1.63
- Bug: TCP/IP fragmentation error
- Altri esempi:
 - 56k ping of death (+++ATH)
 - Winnuke
 - ...

Frammentazione IP

Supponiamo di voler spostare tra due macchine un file molto grande (qualche Mb)

Sapendo che il nostro doppino di rame è in grado di inviare un solo pacchetto alla volta, se trasmettessimo tutto il file in un unico pacchetto, impediremmo a chiunque altro di usare il canale di comunicazione finchè non è terminata la nostra trasmissione

Inoltre, Internet è fatta da tipi di reti molto diversi (Ethernet, ATM, PPP...) che hanno capacità di trasferimento (MTU, velocità) molto diverse tra loro.

La soluzione è quella di dividere in “pezzetti” il file, e trasmetterli uno dopo l'altro.

la frammentazione

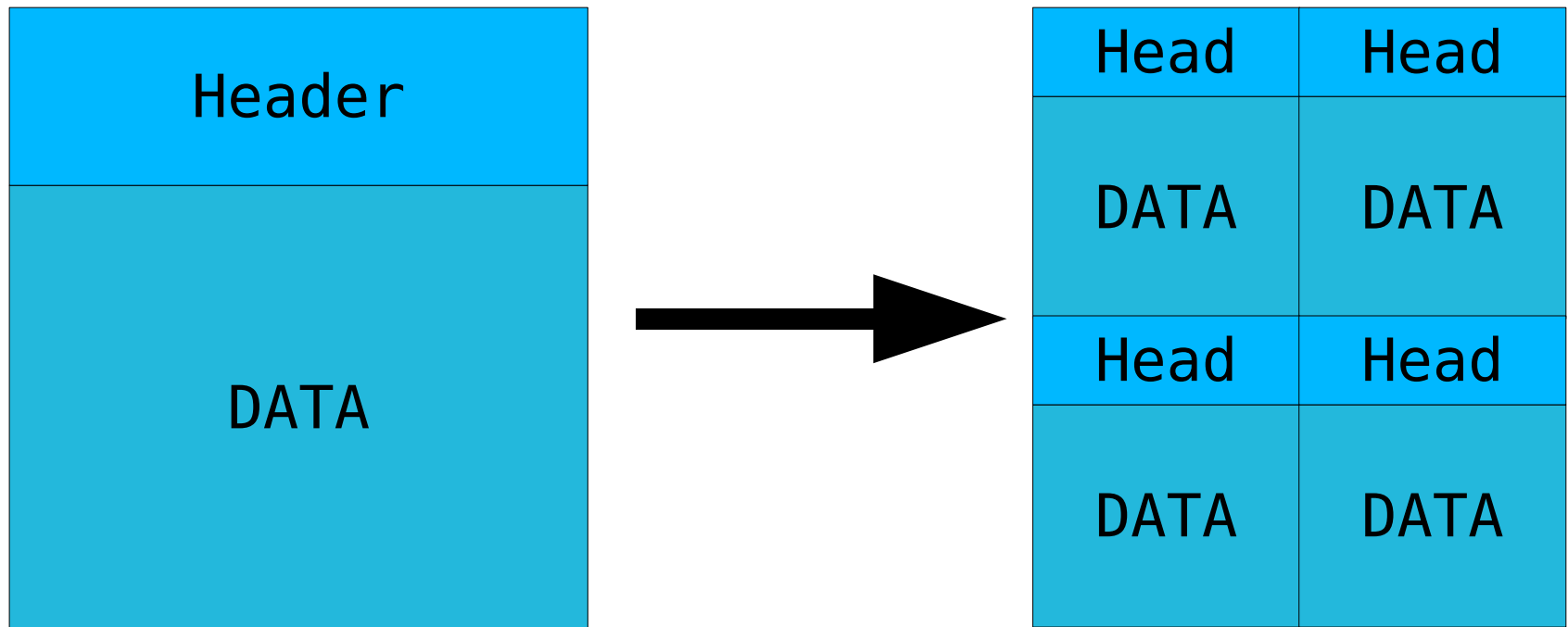
Frammentazione IP

Prendiamo il nostro “pacchettone”.



Frammentazione IP

Il nostro pacchetto viene diviso in “pezzi”. Si dice che viene “frammentato”.



Frammentazione IP

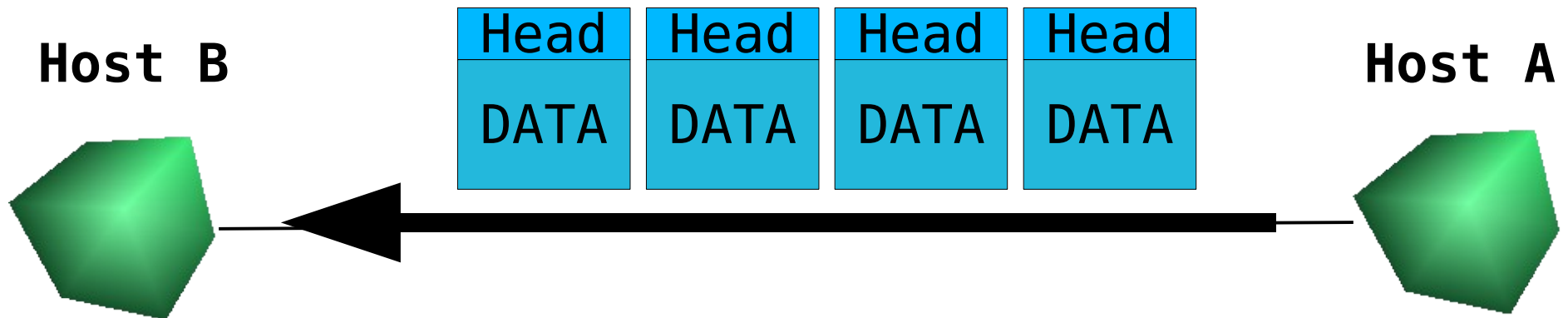
I frammenti vengono poi inviati, uno dopo l'altro, lungo il canale di trasmissione.

Head	Head
DATA	DATA
Head	Head
DATA	DATA



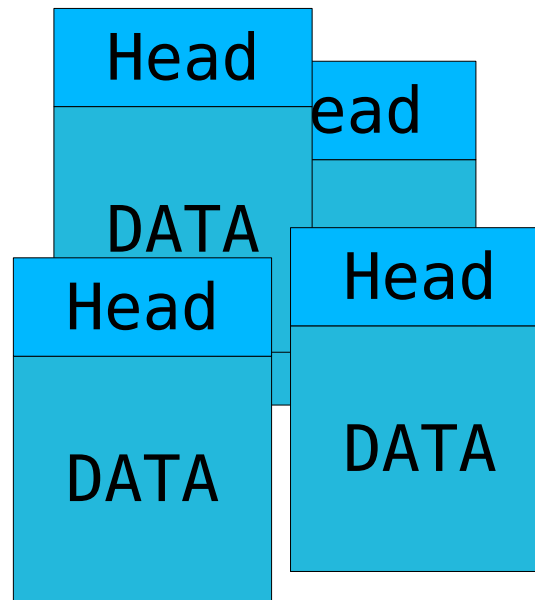
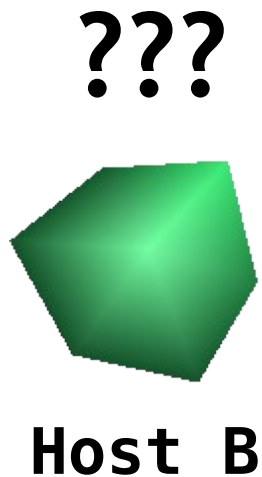
Frammentazione IP

I frammenti vengono poi inviati, uno dopo l'altro, lungo il canale di trasmissione.



Frammentazione IP

Come si possono però ricostruire i pacchetti originali?

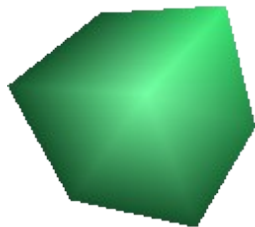


Frammentazione IP

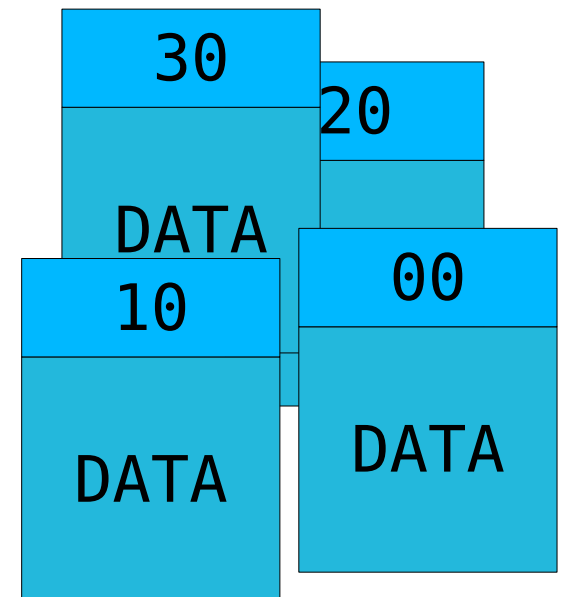
- Nell'intestazione (“header”) di ogni frammento, viene inserito un campo detto “offset” che indica il numero del primo byte dei dati trasmessi nel “payload” (“DATA”) del frammento.
- Ricevuti un frammento, il sistema operativo si preoccupa di “posizionare” il pacchetto, ordinando il payload in base all'offset indicato nell'header, ed inserendolo in un buffer di “ricomposizione”.
- Questo consente anche la ritrasmissione di un frammento in caso di errore nella trasmissione (non importa più l'ordine di invio dei frammenti, che non è comunque garantito ne da IP ne da TCP)

Frammentazione IP

Il sistema operativo ricompile i frammenti.

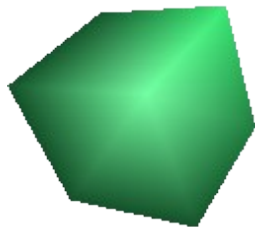
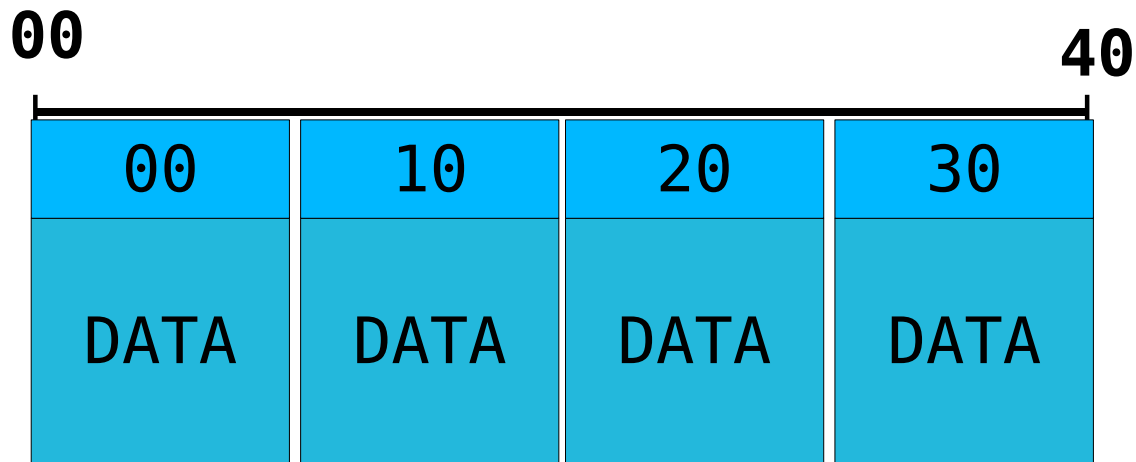


Host B



Frammentazione IP

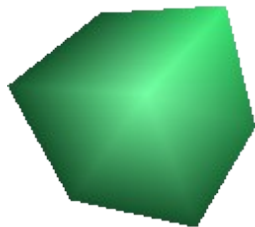
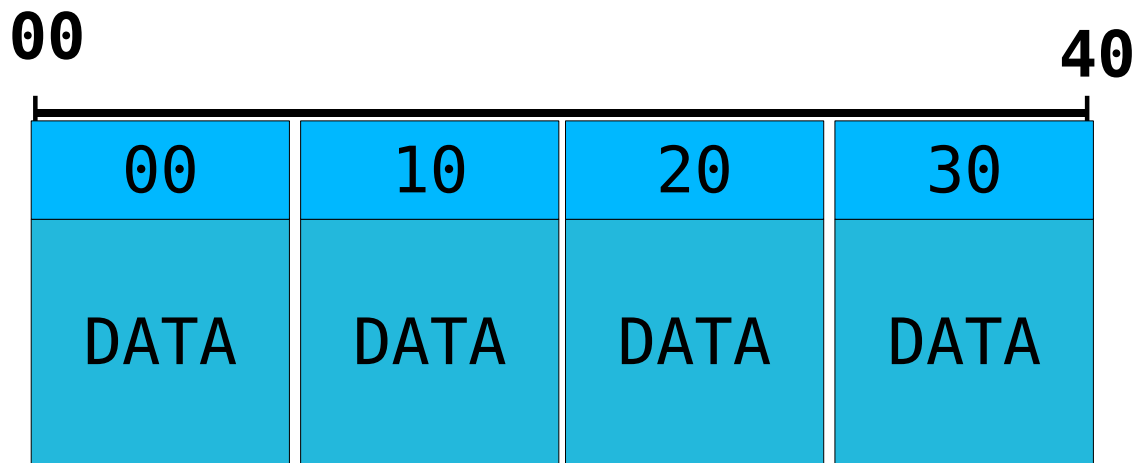
Il sistema operativo ricompone i frammenti.



Host B

Frammentazione IP

Il sistema operativo ricompone i frammenti.

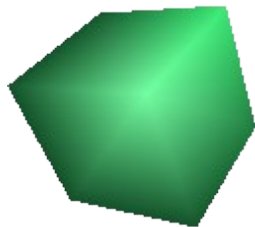
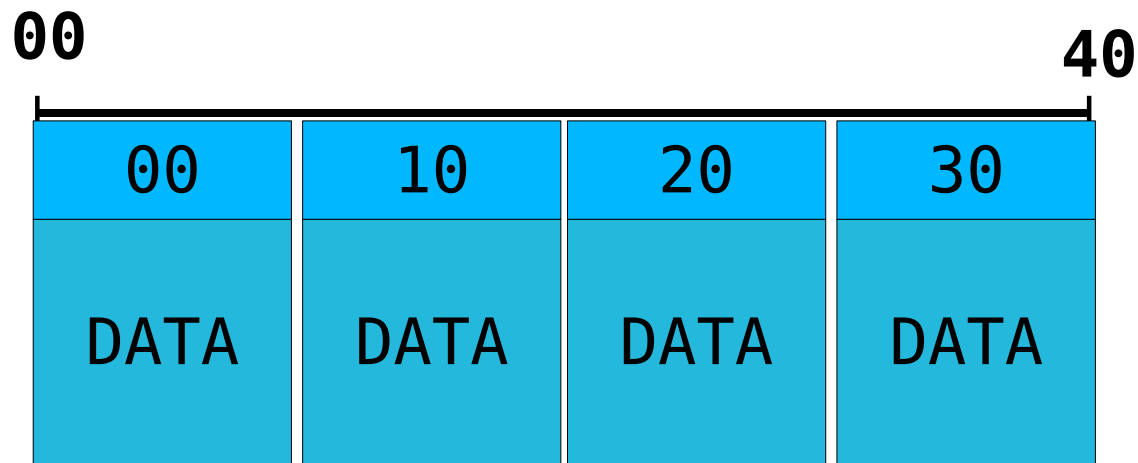


Host B

Quanto trasmesso nel campo “offset” dei pacchetti, viene “aggiunto” ad un indirizzo “base” (00) per sapere dove metterlo.

Frammentazione IP

Il sistema operativo ricompone i frammenti.

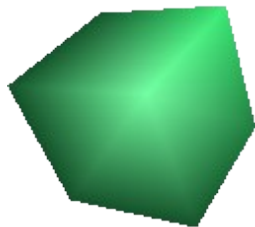
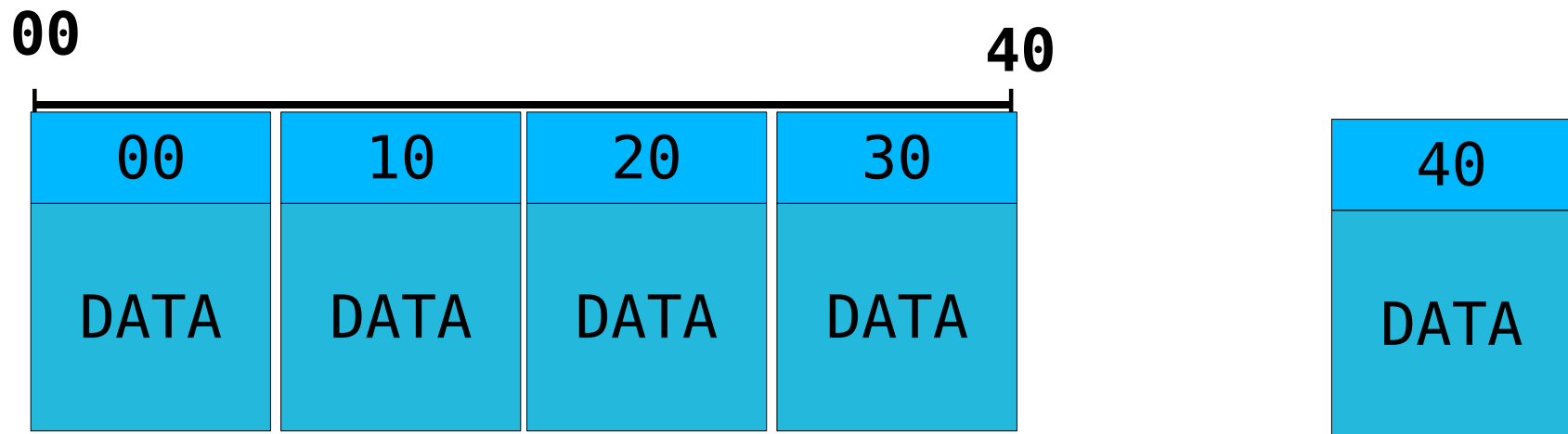


Host B

Quando arriva un nuovo pacchetto, il kernel alloca nuovo spazio in memoria a seconda del suo offset (fa spazio)

Frammentazione IP

Il sistema operativo ricompone i frammenti.

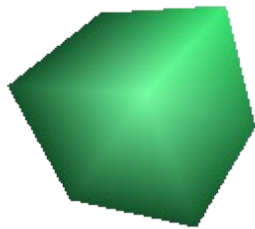
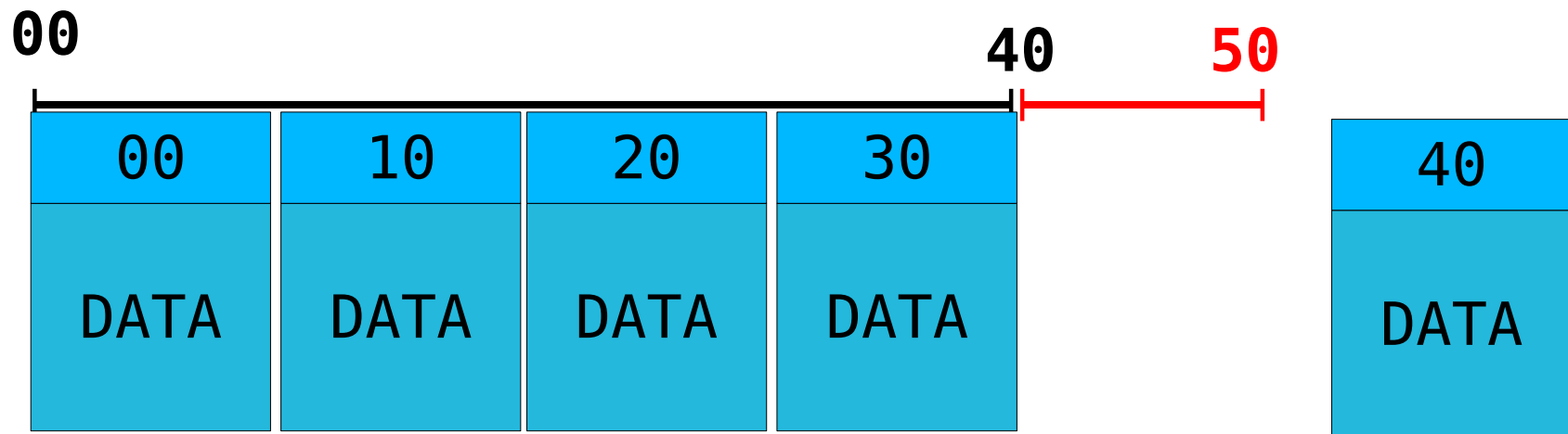


Host B

Quando arriva un nuovo pacchetto, il kernel alloca nuovo spazio in memoria a seconda del suo offset (fa spazio)

Frammentazione IP

Il sistema operativo ricompone i frammenti.

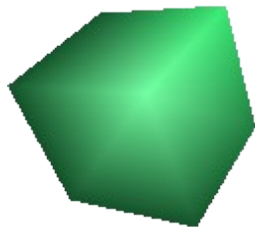
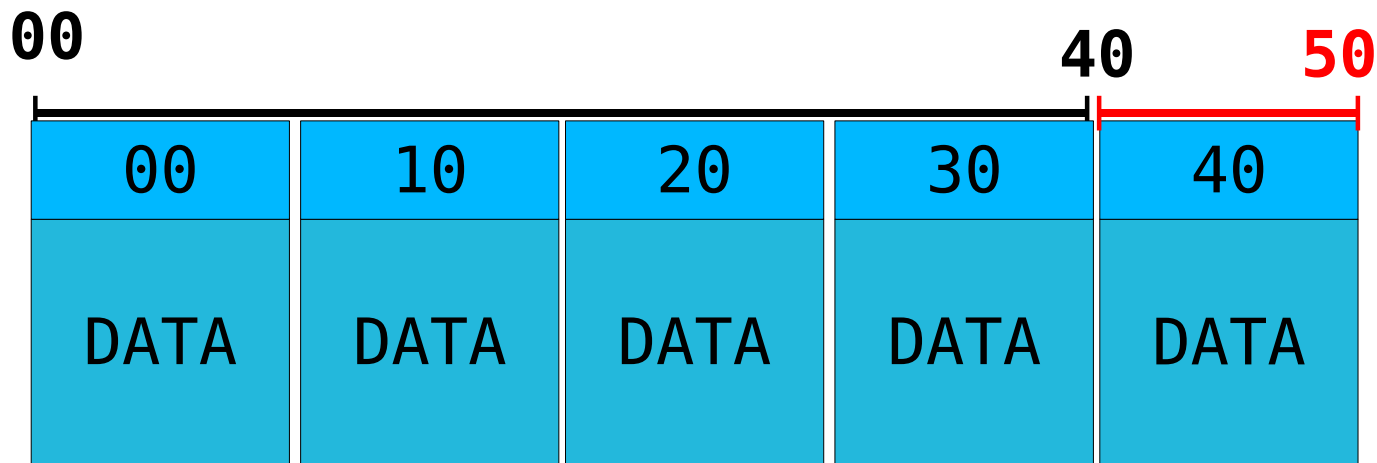


Host B

Quando arriva un nuovo pacchetto, il kernel alloca nuovo spazio in memoria a seconda del suo offset (fa spazio)

Frammentazione IP

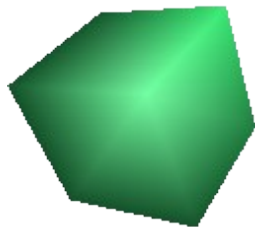
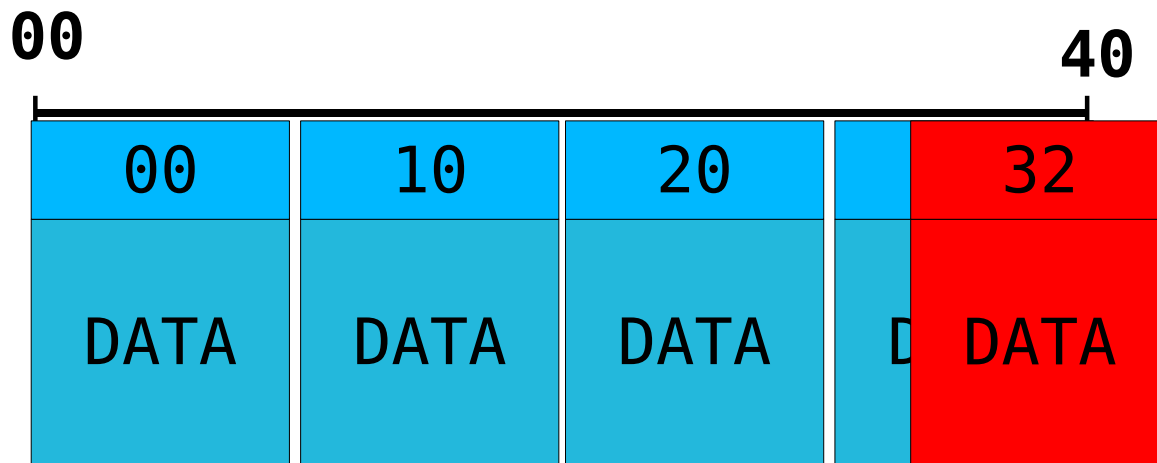
Il sistema operativo ricompone i frammenti.



Host B

Quando arriva un nuovo pacchetto, il kernel alloca nuovo spazio in memoria a seconda del suo offset (fa spazio)

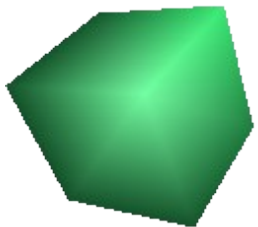
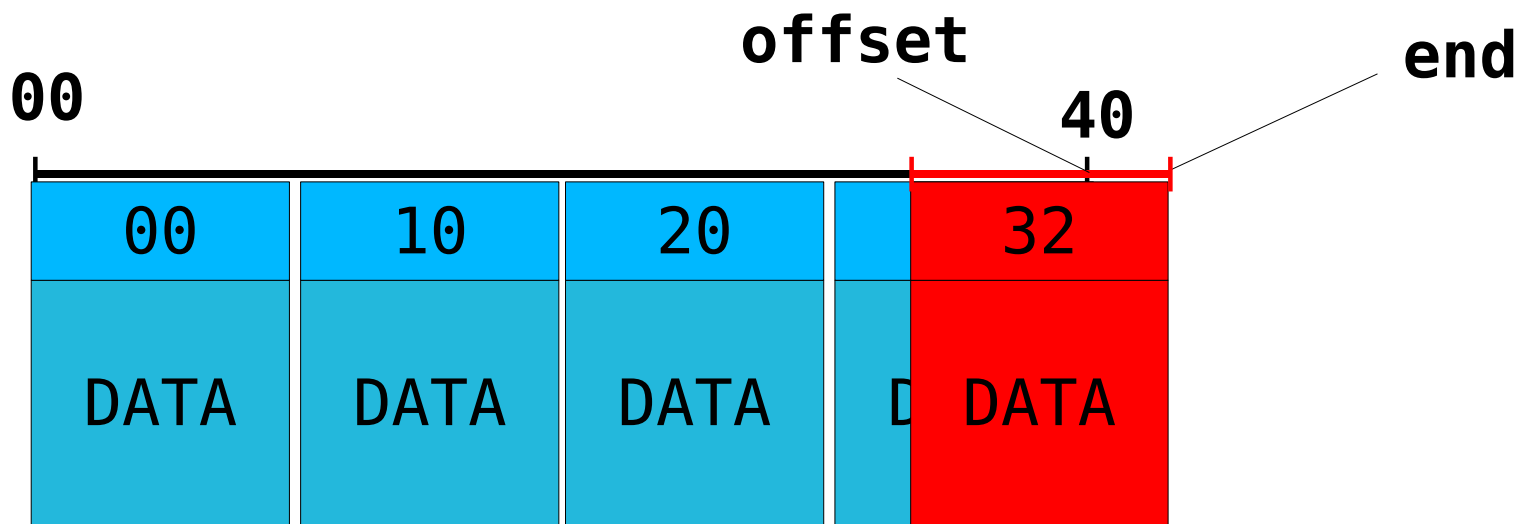
Frammenti sovrapposti



Host B

Quando arriva un frammento che si sovrappone a dati già ricevuti, dobbiamo cercare di allineare il puntatore in modo che non si incasini.

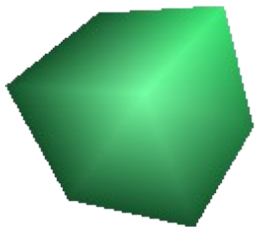
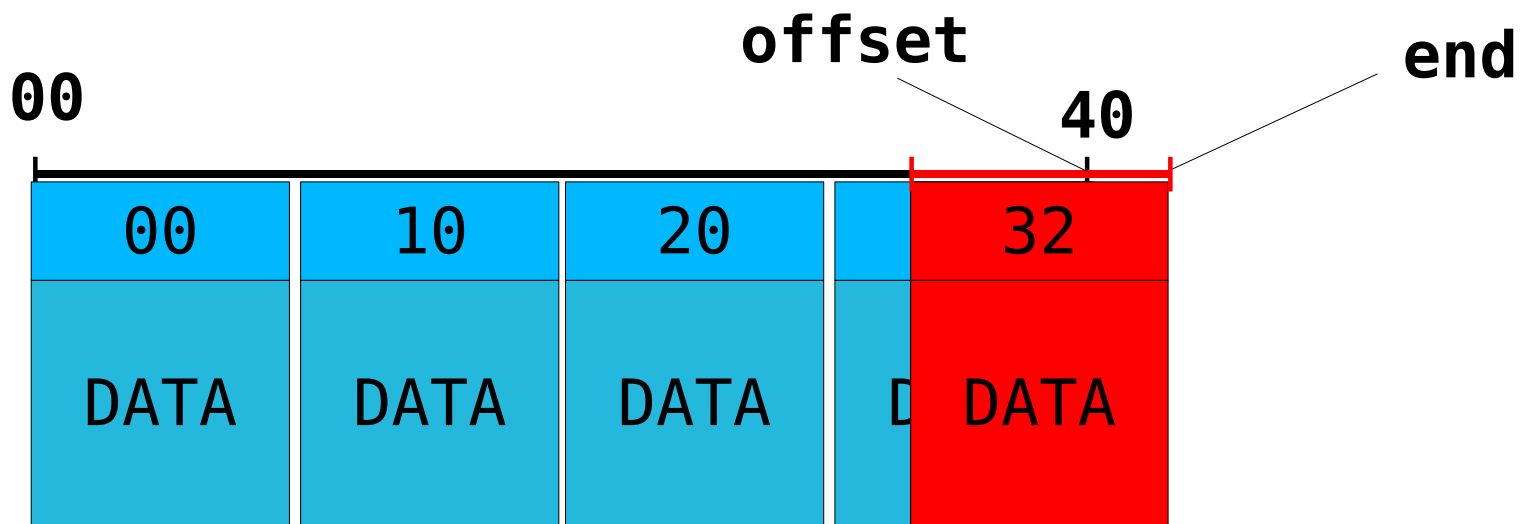
Frammenti sovrapposti



Host B

Quanto spazio deve allocare il kernel?

Frammenti sovrapposti

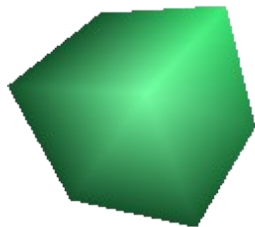
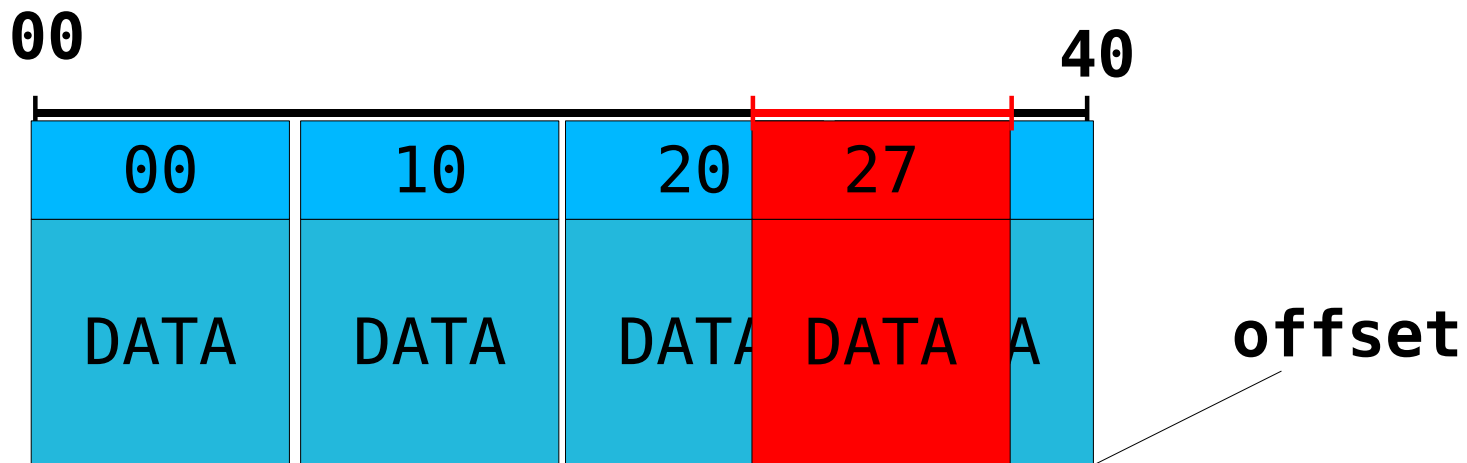


Host B

Quanto spazio deve allocare il kernel?

$end - offset$

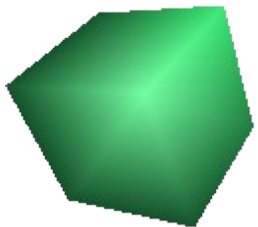
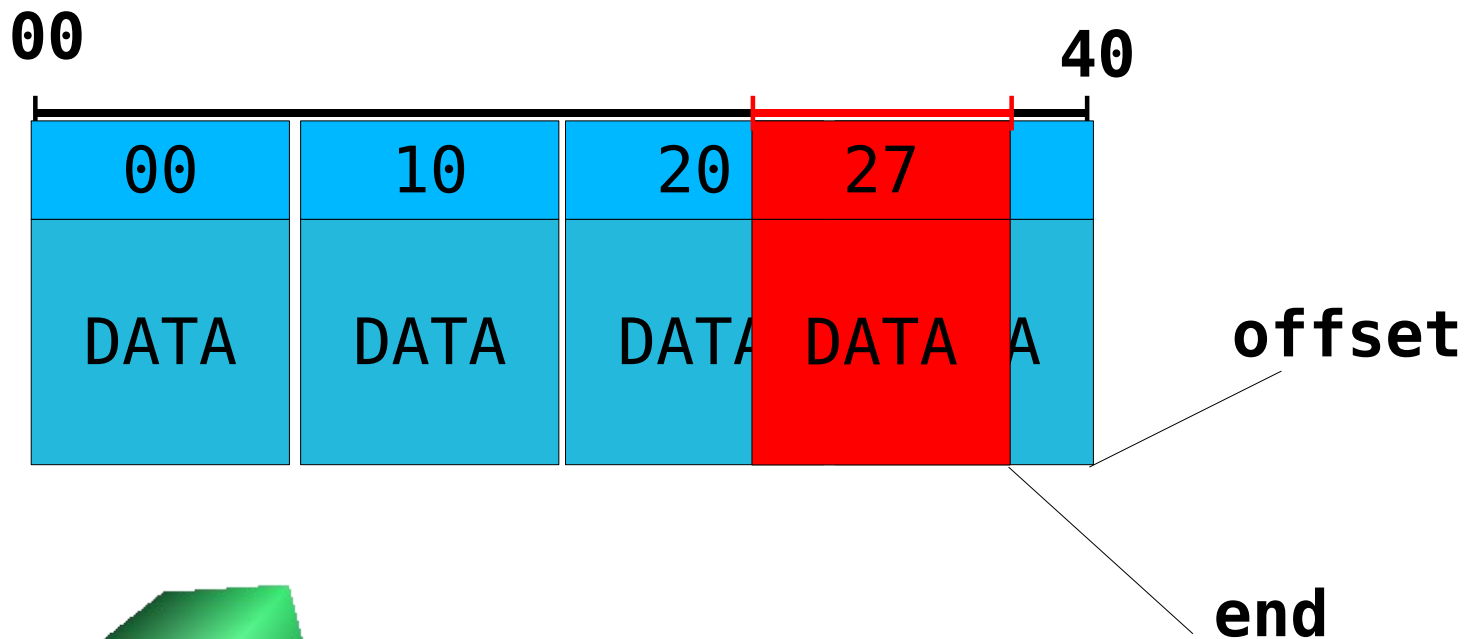
Frammenti sovrapposti



Host B

Ed in questa situazione?

Frammenti sovrapposti



Host B

Ed in questa situazione?

end - offset = valore negativo!!

Perché è un problema?

Quando

end - offset

da un risultato negativo, qual'è il problema?

Perché è un problema?

Quando

end - offset

da un risultato negativo, qual'è il problema?

CHE LA MEMORIA NON HA SEGNO!!

Conseguenze

- Crash e riavvio della macchina
- Qualsiasi servizio attivo sulla macchina, interrotto.
- Se non si usa un filesystem journalised, dati possibilmente corrotti, macchina che non torna su correttamente.

Conseguenze

- Crash e riavvio della macchina
- Qualsiasi servizio attivo sulla macchina, interrotto.
- Se non si usa un filesystem journalised, dati possibilmente corrotti, macchina che non torna su correttamente.

Denial of Service

TearDrop

- Veloce
(2 pacchetti e via)
- Pulito
(nessuna traccia)
- Preciso
(non colpisce altri sistemi)
- Colpisce solo la piattaforma buggata
(non tutti hanno lo stesso bug)
- Un problema temporaneo
(corretto il bug...)

TearDrop

- Veloce
(2 pacchetti e via)
- Pulito
(nessuna traccia)
- Preciso
(non colpisce altri sistemi)
- Colpisce solo la piattaforma buggata
(non tutti hanno lo stesso bug)
- Un problema temporaneo
(corretto il bug...)

Bug corretti dopo oltre un anno? (MS)

TearDrop

- Veloce
(2 pacchetti e via)
- Pulito
(nessuna traccia)
- Preciso
(non colpisce altri sistemi)
- Colpisce solo la piattaforma buggata
(non tutti hanno lo stesso bug)
- Un problema temporaneo
(corretto il bug...)

Service Pack che introducono bug? (MS)

Esempio di Nuke: LAND

- Tipo di attacco che si basa su una scorretta gestione dello stack TCP/IP
- Piattaforme colpite:
 - Windows (tutte fino a 2003 Server. Vista?)
 - Cisco IOS (molte versioni)

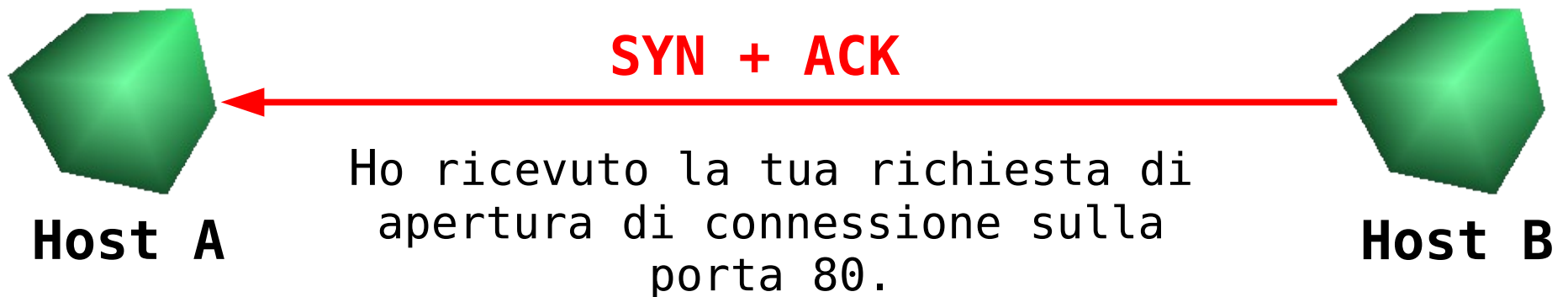
TCP: apertura connessione

- Secondo le specifiche del protocollo TCP, l'apertura delle connessioni si effettua con una procedura detta
- “3 way handshaking”
- Vengono inviati in serie 3 pacchetti con flag diversi:
 - SYN
 - ACK
 - SYN+ACK

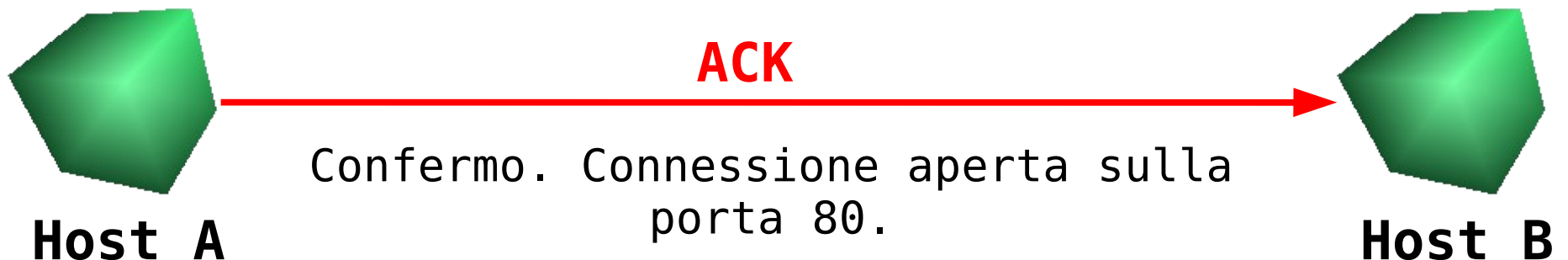
TCP: apertura connessione



TCP: apertura connessione



TCP: apertura connessione



TCP: apertura connessione

- La procedura di “3 way handshaking” consente l'apertura corretta di una connessione, controllando che non vi siano errori nella trasmissione dei pacchetti di inizializzazione.
- La scelta della porta su cui viene aperta la connessione, è fatta direttamente nel frame TCP, che presenta i campi SOURCE_PORT, SOURCE_IP, DEST_PORT, DEST_IP.

Attacchi LAND

- A causa di una errata gestione dello stack TCP/IP (nel software che gestisce questi protocolli), i sistemi operativi Windows si rivelano vulnerabili ad attacchi di tipo LAND.
- Questi attacchi consistono semplicemente nell'inviare un pacchetto SYN con:
 - SOURCE_PORT = DEST_PORT
 - SOURCE_IP = DEST_IP = VICTIM_IP

Conseguenze

- Su sistemi relativamente vecchi
(Windows 95, 98, NT, ME)
 - Blue screen of Death
- Su sistemi recenti
(Windows 2000, XP, 2003 Server)
 - Freeze (blocco) della macchina per 15-30 secondi
- Non risolve la situazione: basta un nuovo pacchetto ogni 30 secondi per causare un DoS

Prevenire

- Un attacco DoS basato su una metodologia di tipo Nuke, SI PUO' PREVENIRE (almeno in gran parte dei casi).
 - Tenere aggiornato il sistema
 - Chiudere servizi non necessari
 - Proteggere adeguatamente la rete

Flood

Flood

- Il concetto di Flood è piuttosto ampio, quasi quanto quello di DoS.
- Anche qui distinguiamo tre categorie:
 - I flood da parte di singoli utenti
 - Gli smurf (che devono il nome allo Smurf2k)
 - I DDoS (che a loro volta si dividono...)
- Ogni tipo di flood ha bisogno di condizioni di partenza diverse.

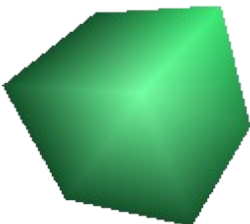
Andiamo quindi a vedere qualche esempio, che ci consenta di comprendere le differenze tra i vari tipi di flood.

SYN Flood

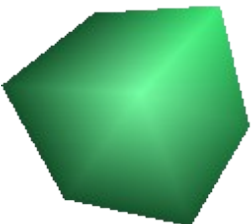
- Il modo più classico e banale
- I pacchetti SYN sono quelli che si usano per “iniziare” una connessione
- Ogni volta che un sistema riceve un pacchetto SYN, alloca una serie di risorse in modo da permettere al mittente di collegarsi

SYN Flood

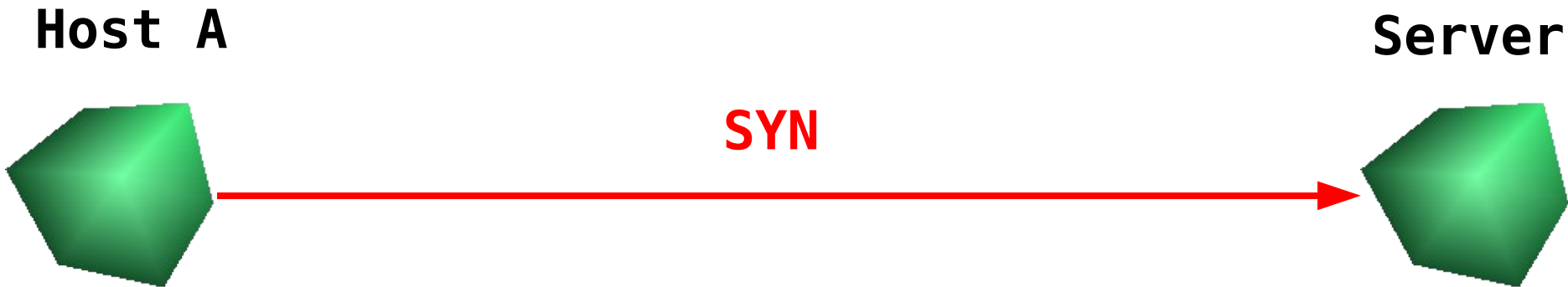
Host A



Server

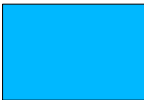
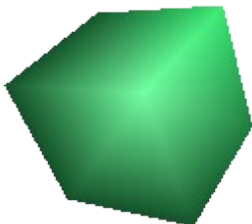


SYN Flood

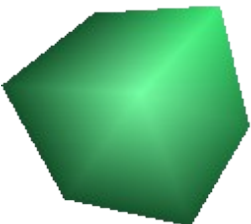


SYN Flood

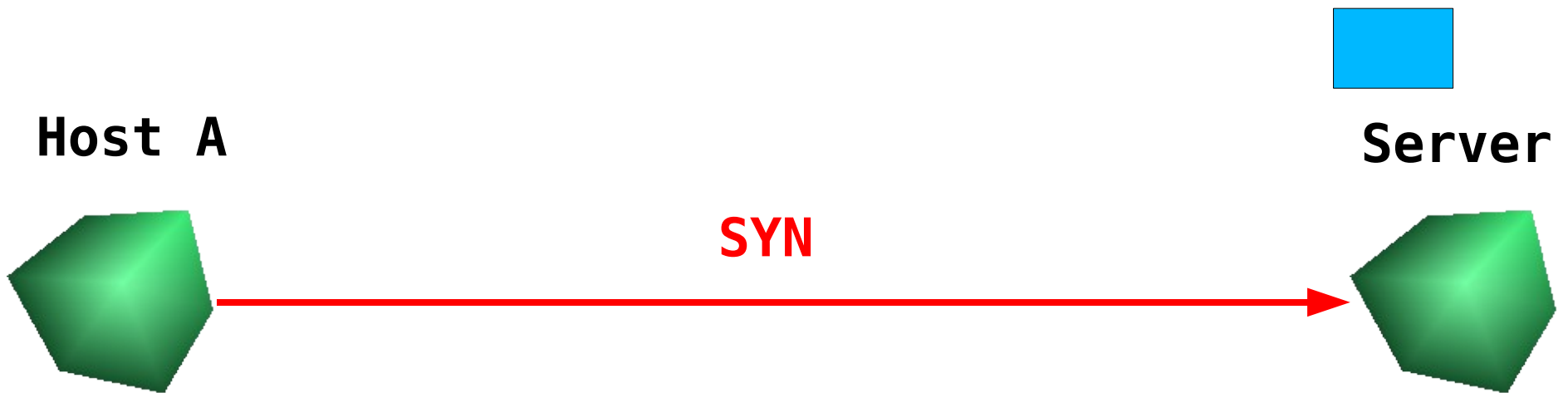
Host A



Server

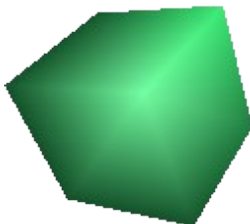


SYN Flood

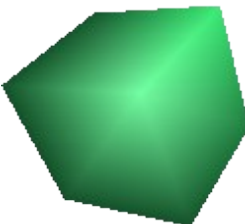
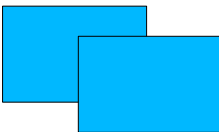


SYN Flood

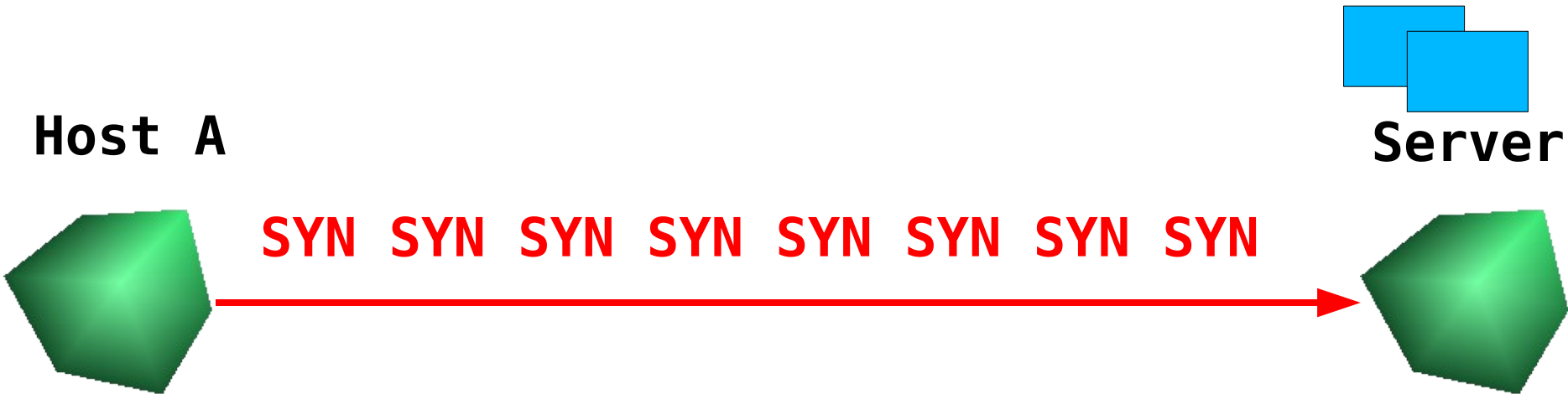
Host A



Server



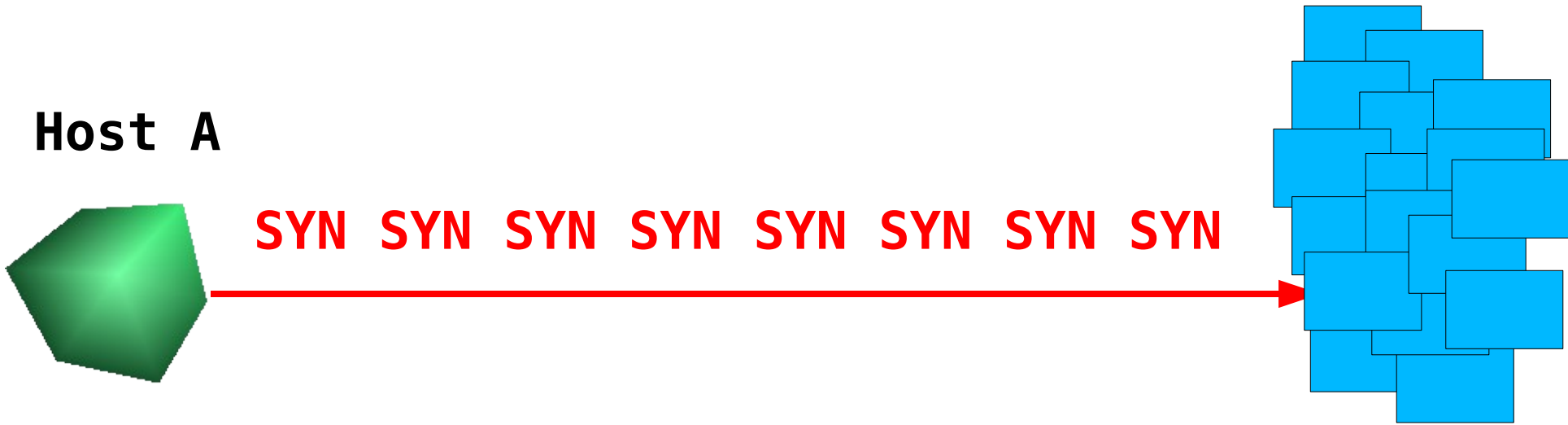
SYN Flood



SYN Flood

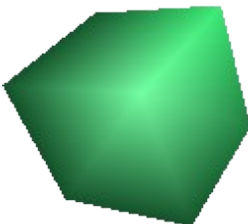


SYN Flood

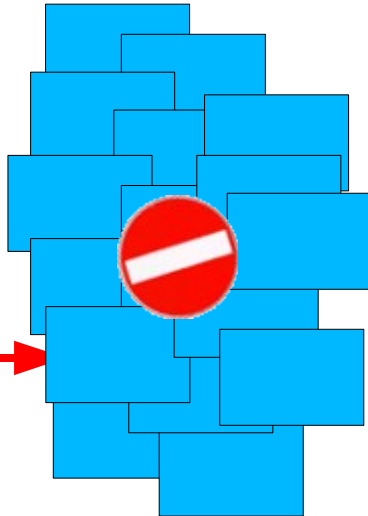


SYN Flood

Host A



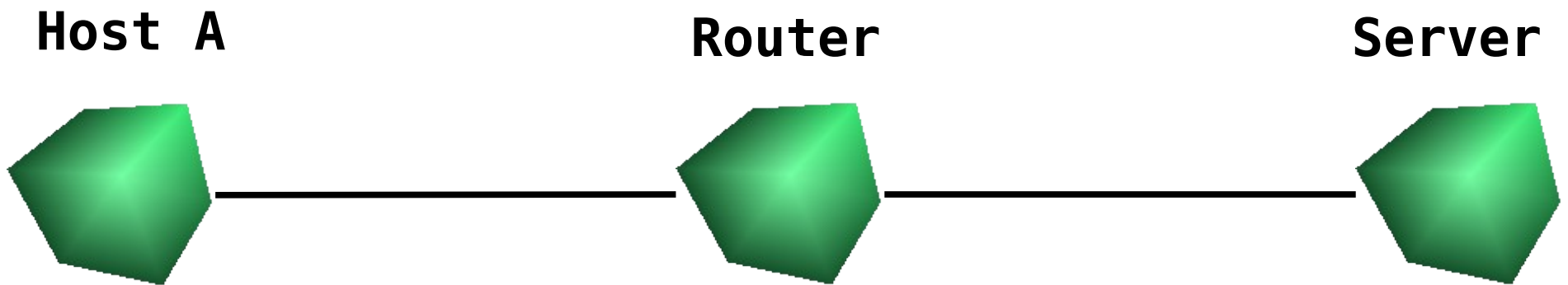
SYN SYN SYN SYN SYN SYN SYN SYN



SYN Flood

- Vantaggi
 - Estremamente semplice da implementare
 - Non necessita di situazioni particolari
 - Non si sfruttano bug -> non correggibile
- Svantaggi
 - Nel corso del tempo, la comunità del software libero ha ideato un meccanismo detto “SYN-Cookies” che consente di non allocare RAM a fronte di un SYN, fino alla ricezione dell'ACK
- Esiste un'altra risorsa molto limitata però... la banda passante! Cosa succede se finisce quella?

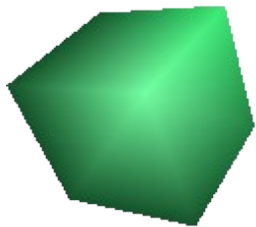
Flood



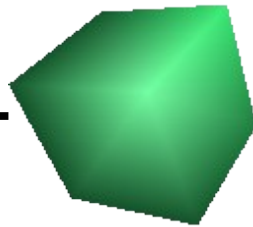
Flood



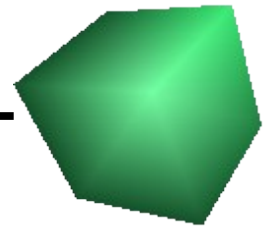
Host A



Router



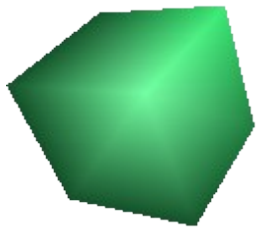
Server



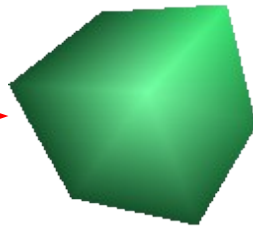
Flood



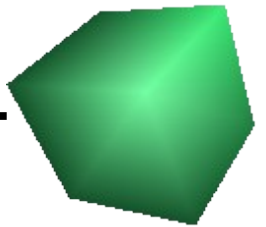
Host A



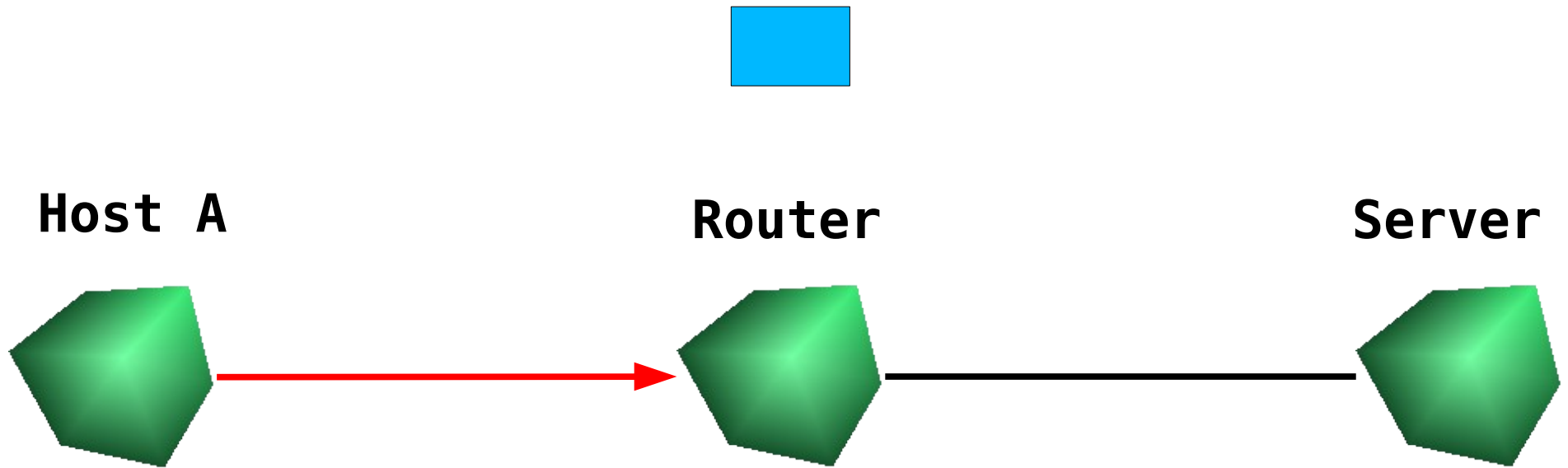
Router



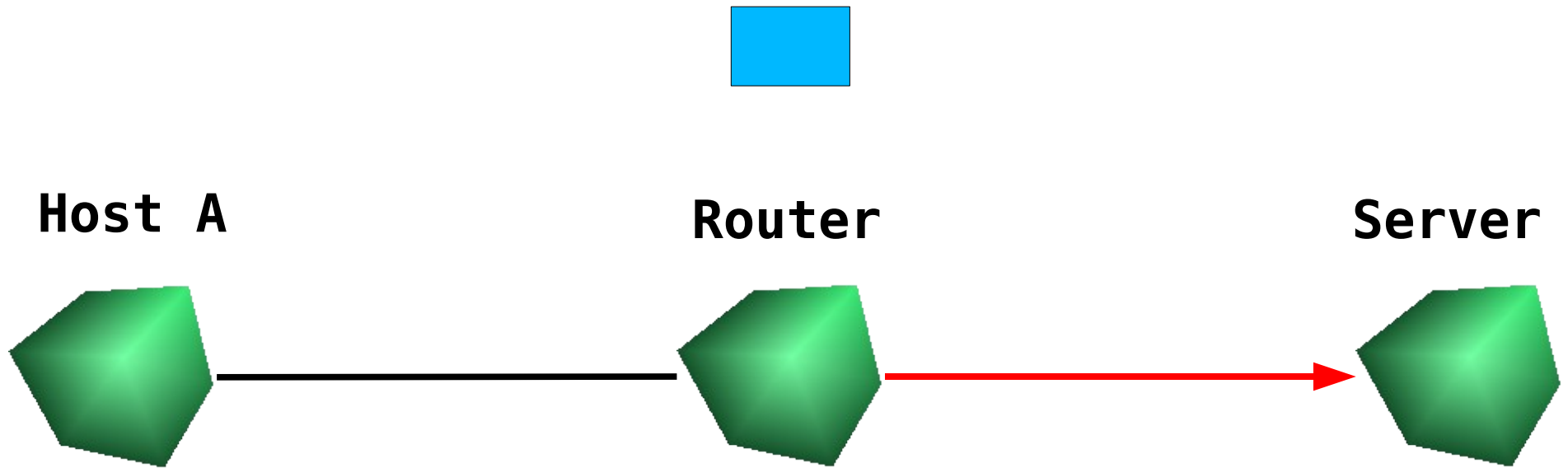
Server



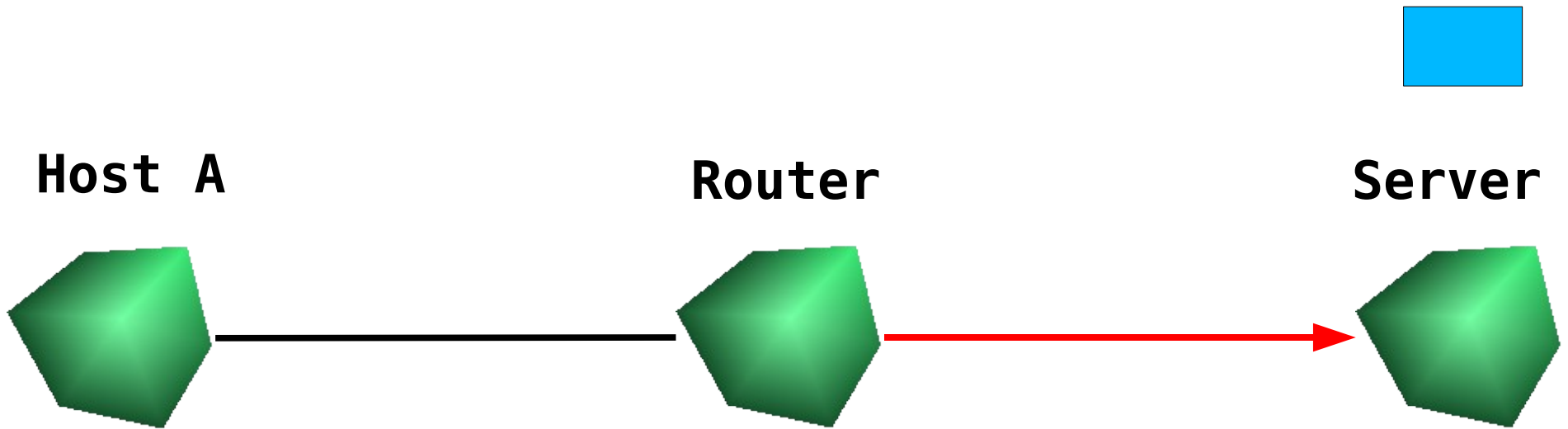
Flood



Flood

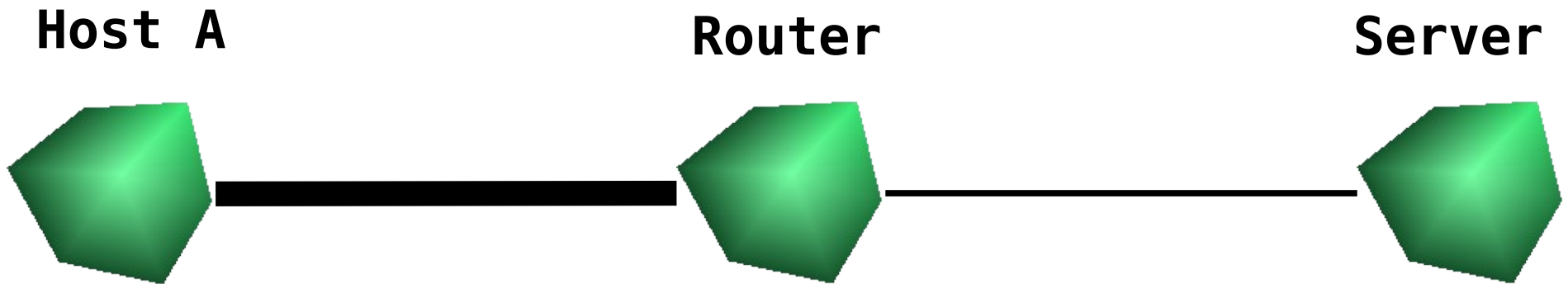


Flood

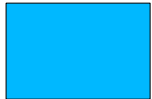


Flood

Cosa succede quando la connessione tra Host A e Router è molto più veloce della connessione tra Router e Server (ad esempio 2 volte tanto)?



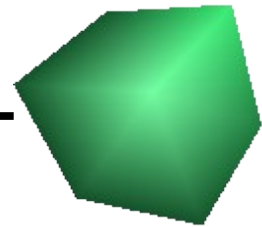
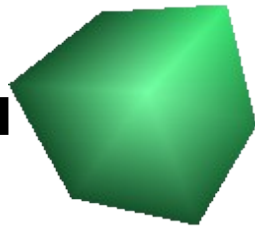
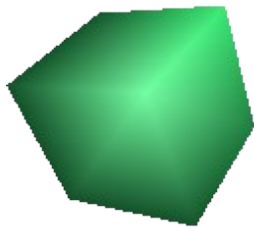
Flood



Host A

Router

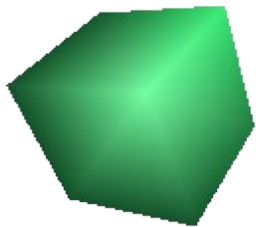
Server



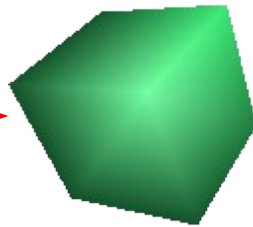
Flood



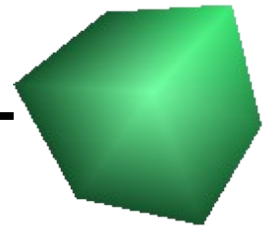
Host A



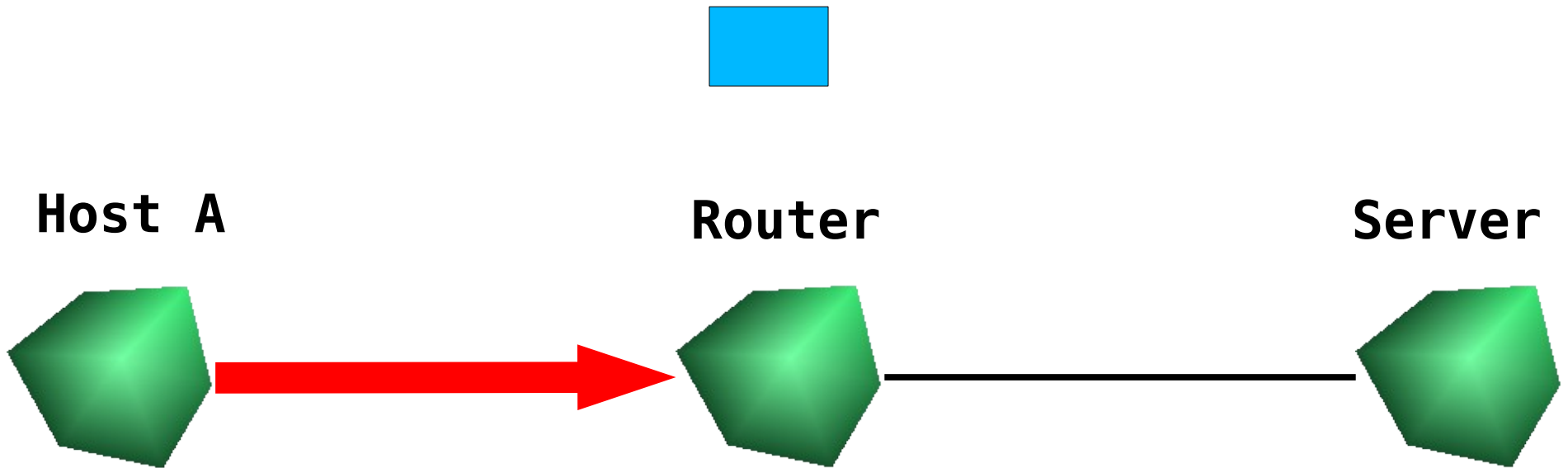
Router



Server



Flood



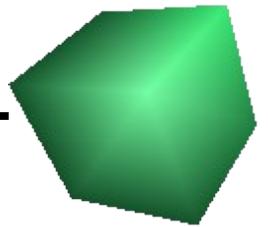
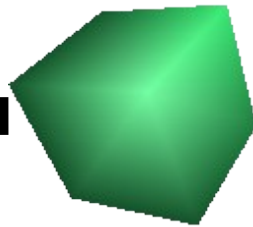
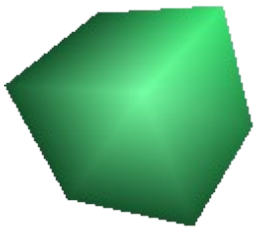
Flood



Host A

Router

Server



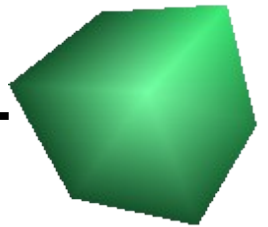
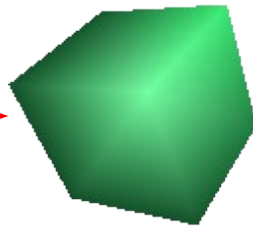
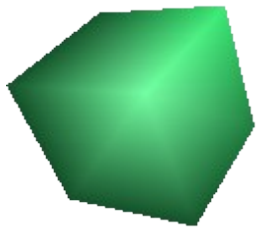
Flood



Host A

Router

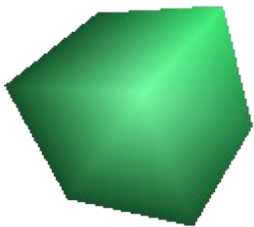
Server



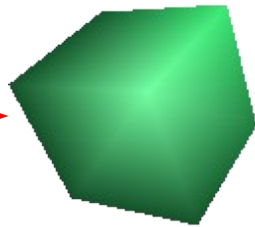
Flood



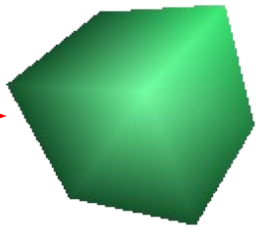
Host A



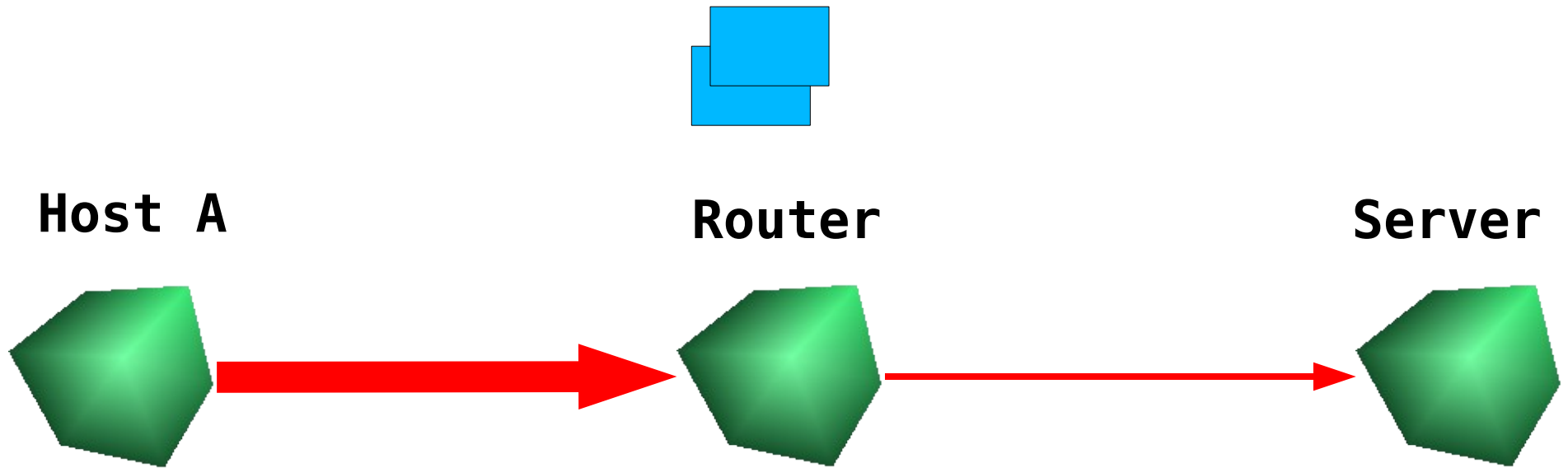
Router



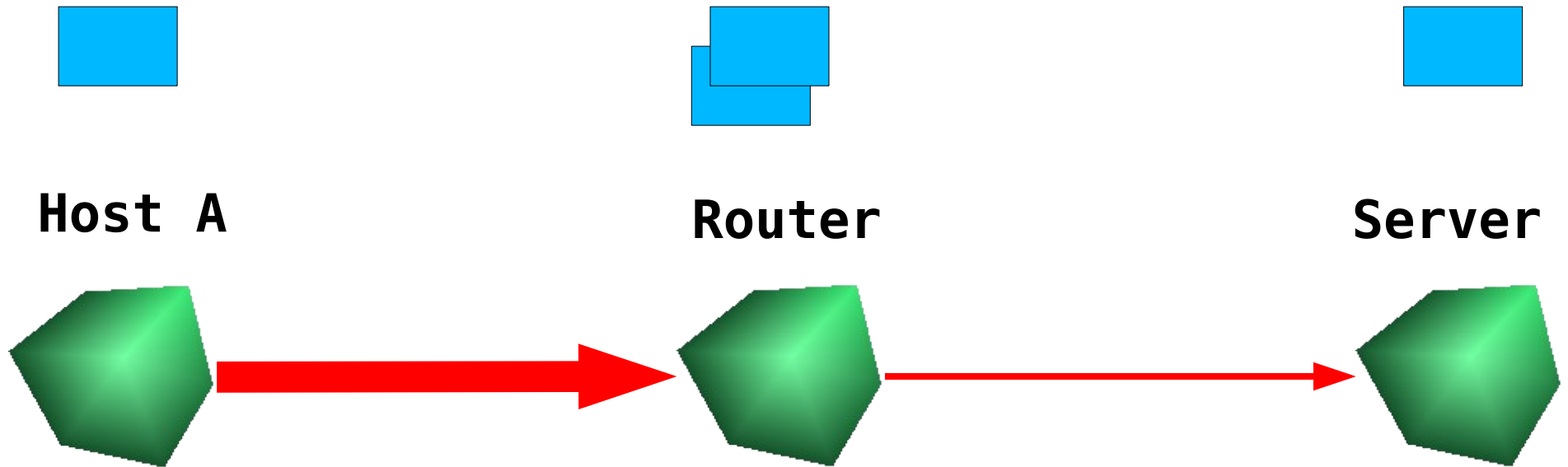
Server



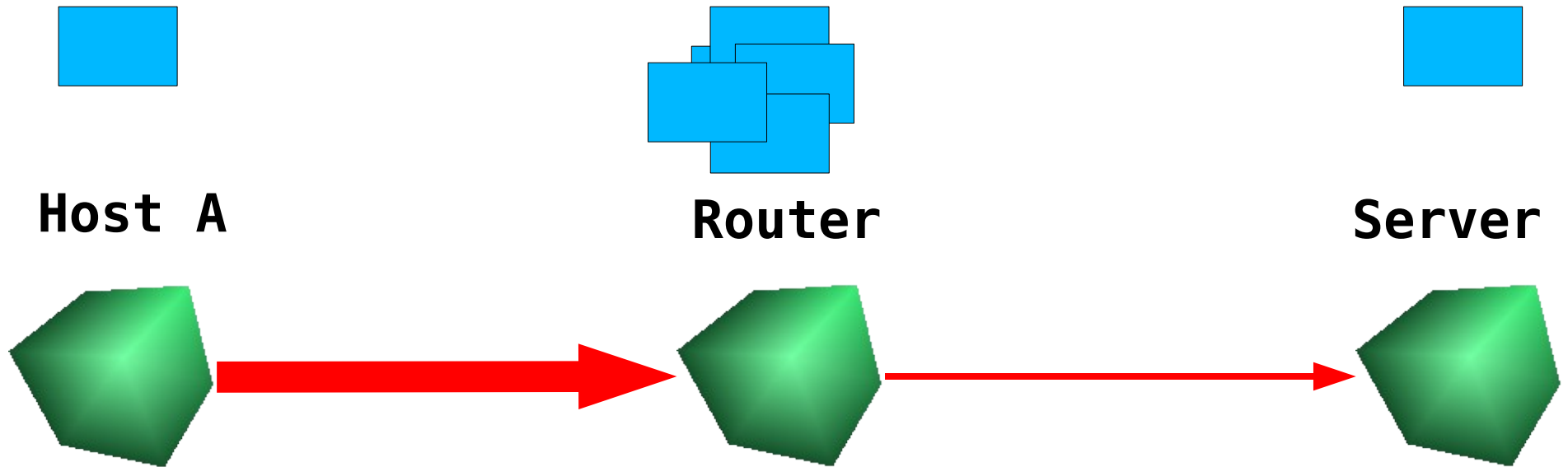
Flood



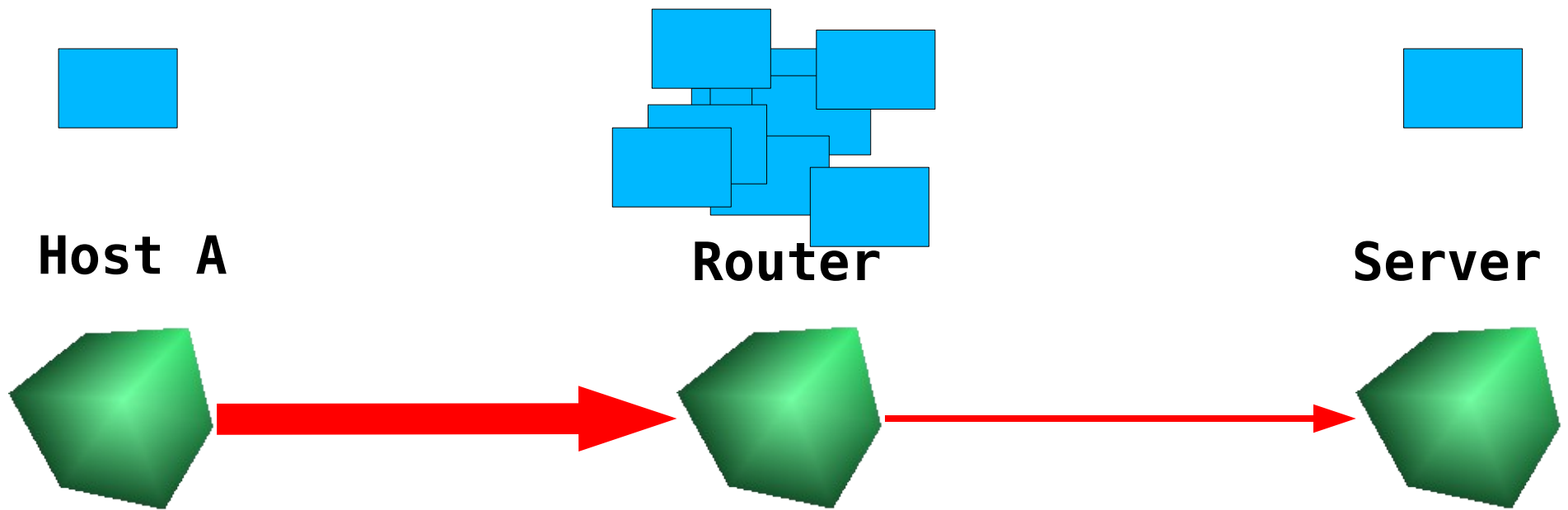
Flood



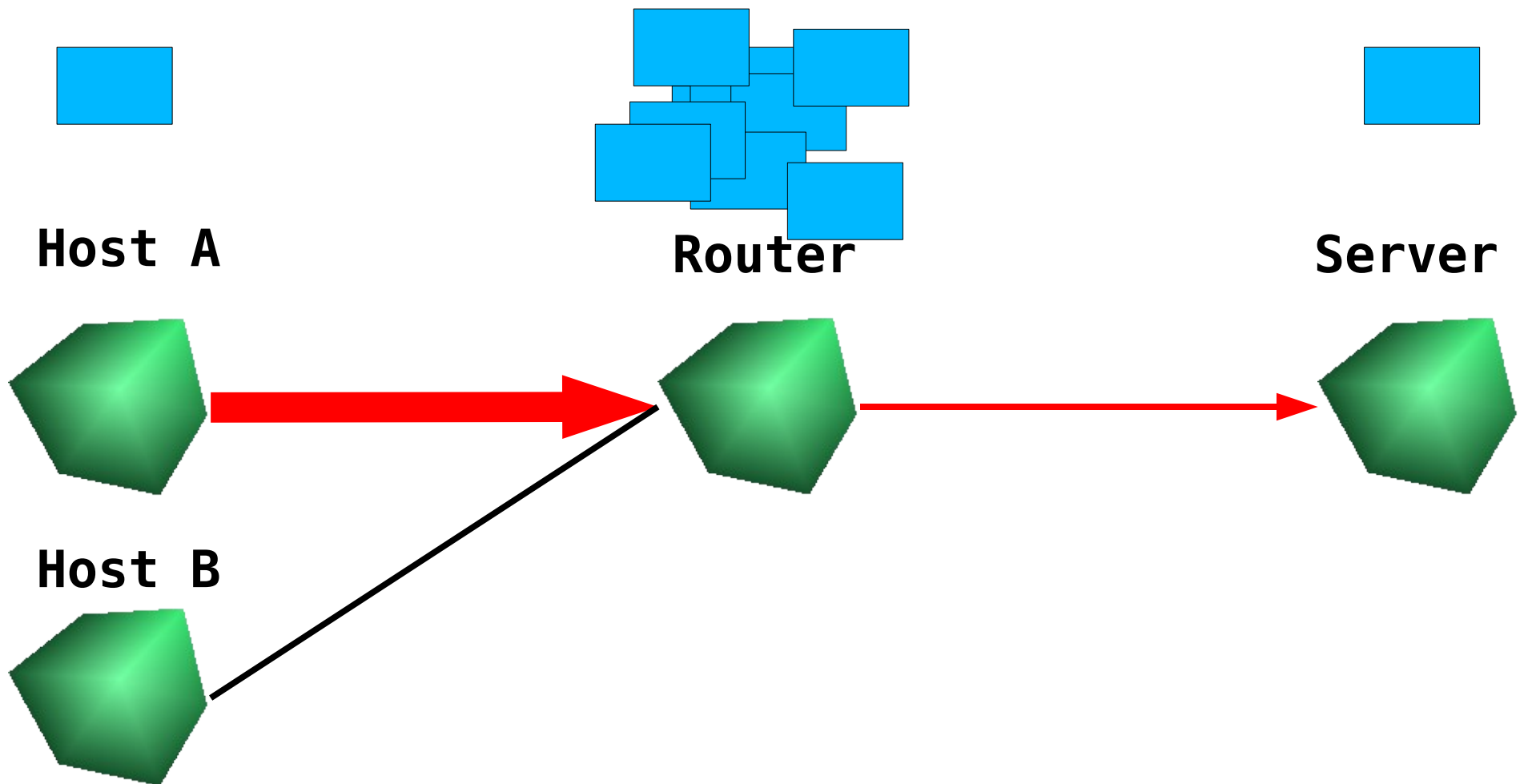
Flood



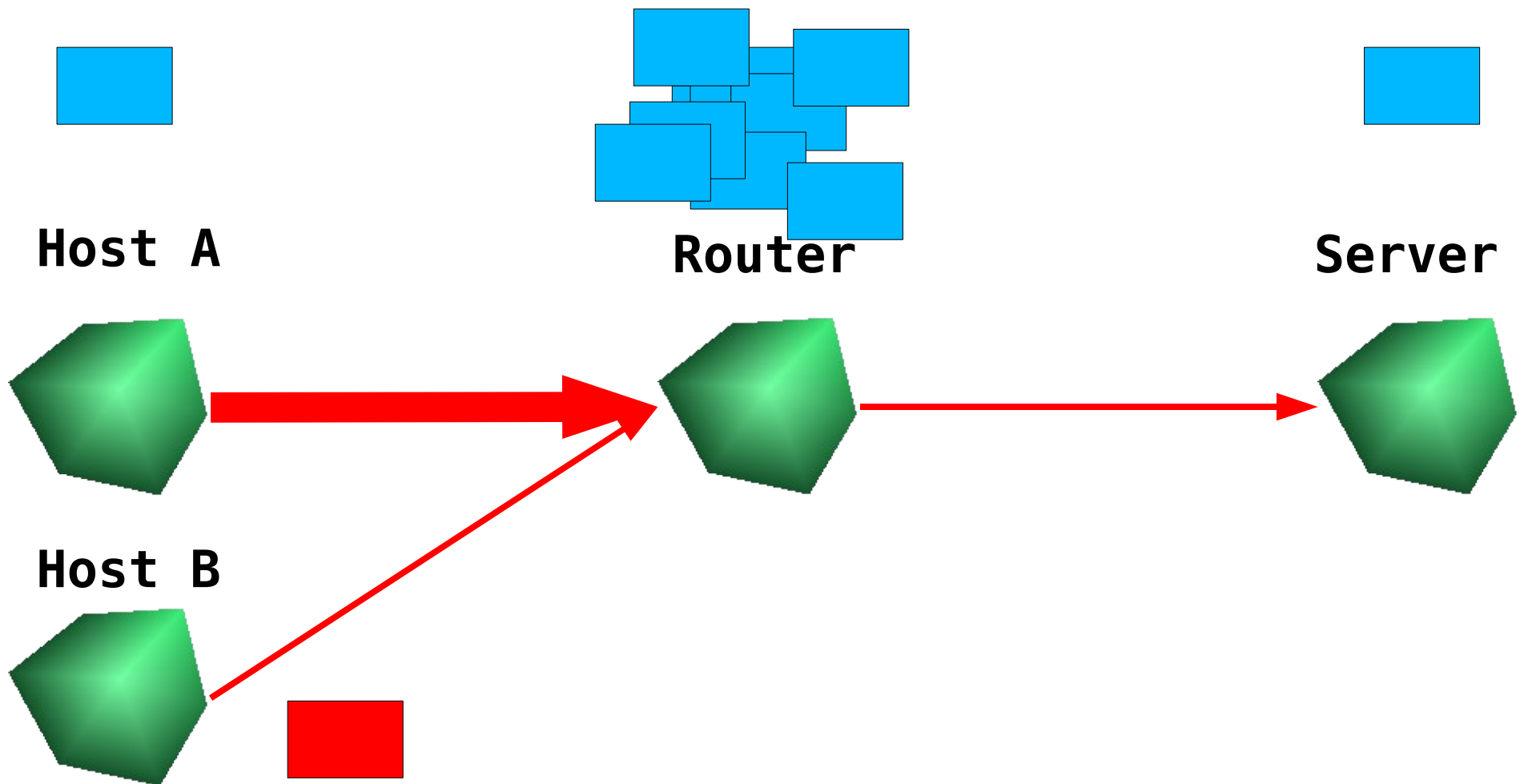
Flood



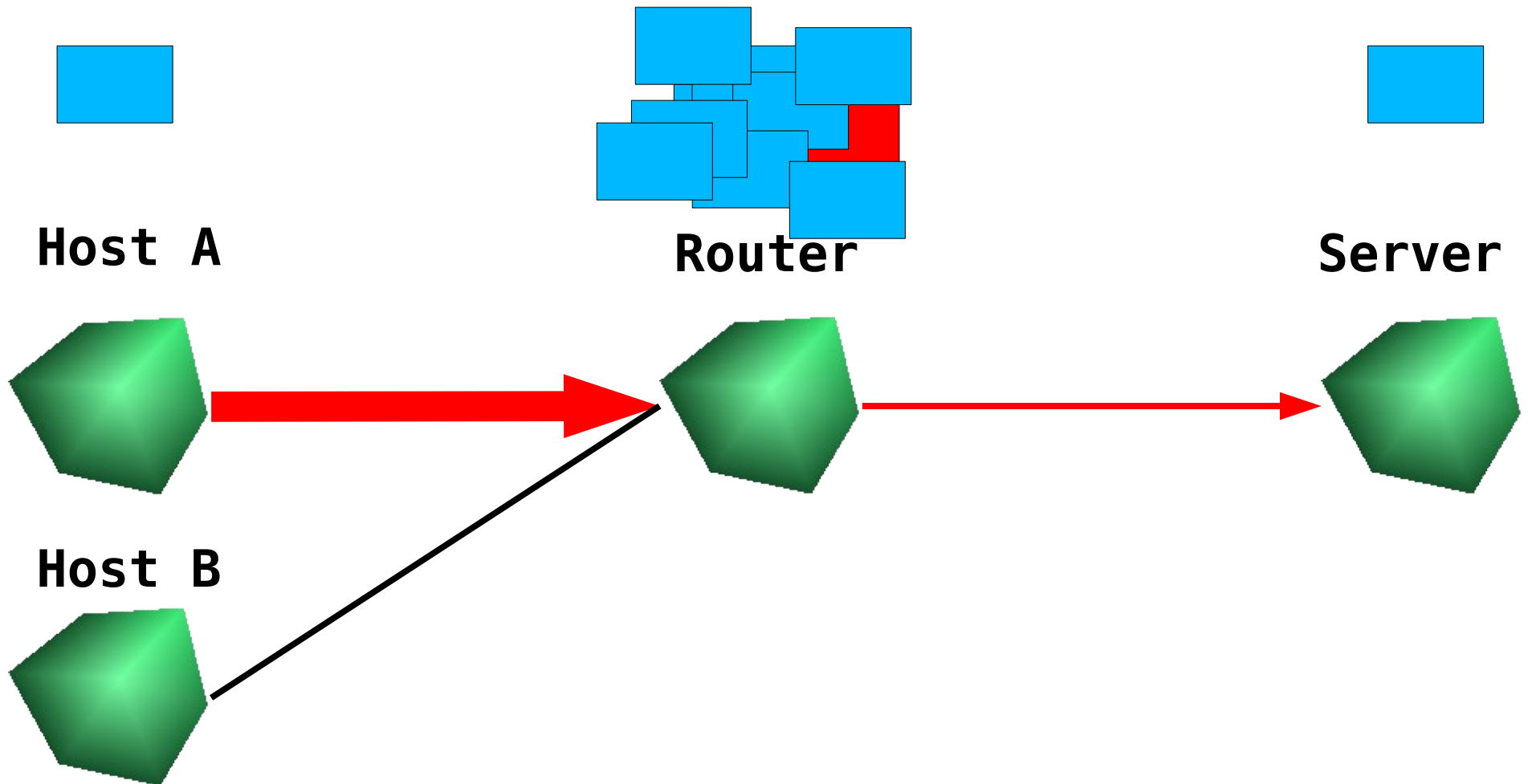
Flood



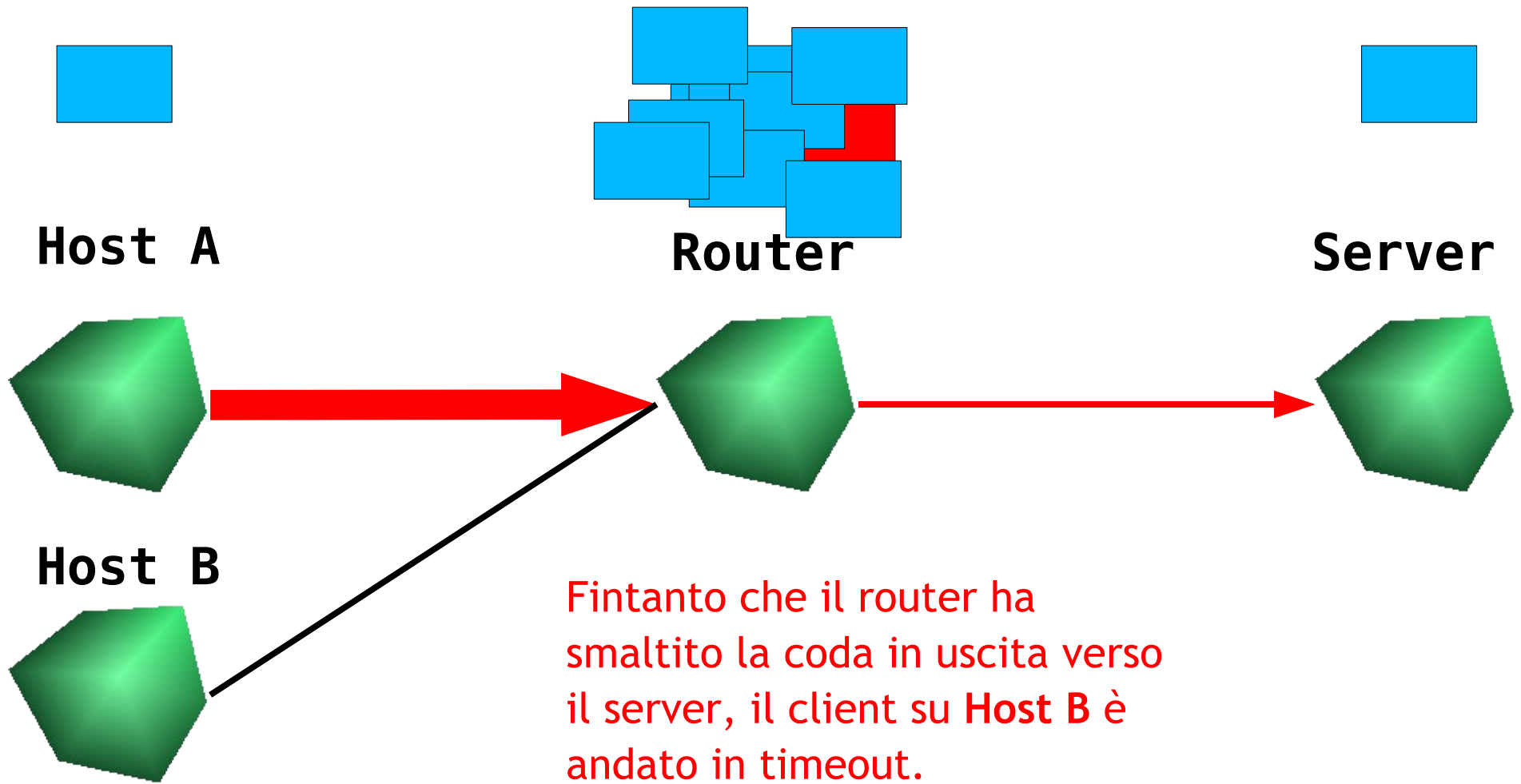
Flood



Flood



Flood

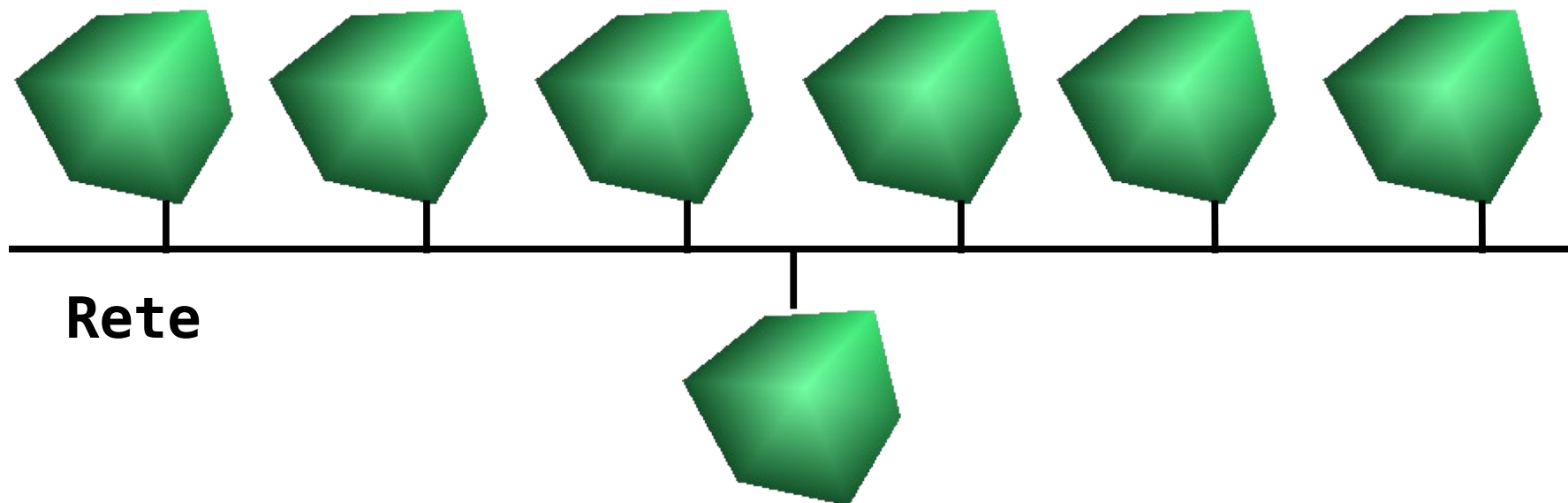


Flood

- Abbiamo ottenuto un DoS.
- I servizi sulla macchina attaccata (Server) non sono più raggiungibili (pur essendo paradossalmente attivi)
- Vantaggio: il sistema attaccato non può fare assolutamente nulla per difendersi.
- Problema: quando mai la rete di un utente è più veloce di quella di un provider?

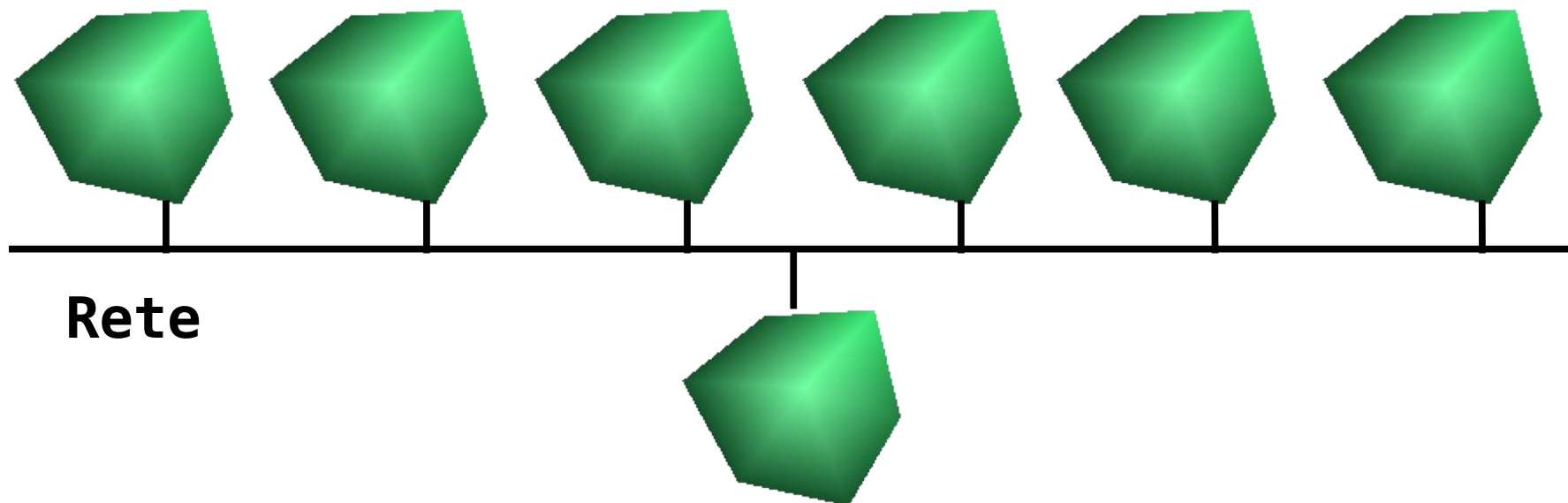
E' NECESSARIA PIU BANDA

Broadcast



Secondo il protocollo IP, ogni rete che viene definita, deve avere riservati due indirizzi “speciali”, non assegnabili ad un host della stessa rete.

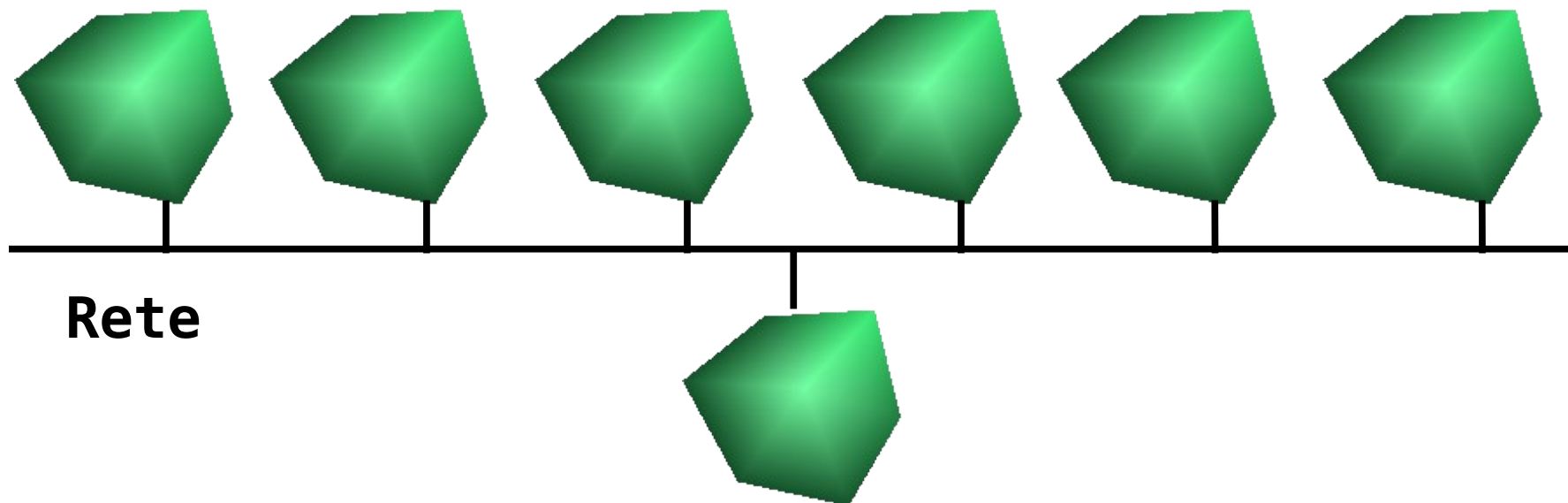
Broadcast



192.168.3.0

Il primo di questi due indirizzi, detto “indirizzo di rete” (quello con tutta la parte host dell'indirizzo fatta di 0), serve principalmente per il routing dei pacchetti. Non ci interessa.

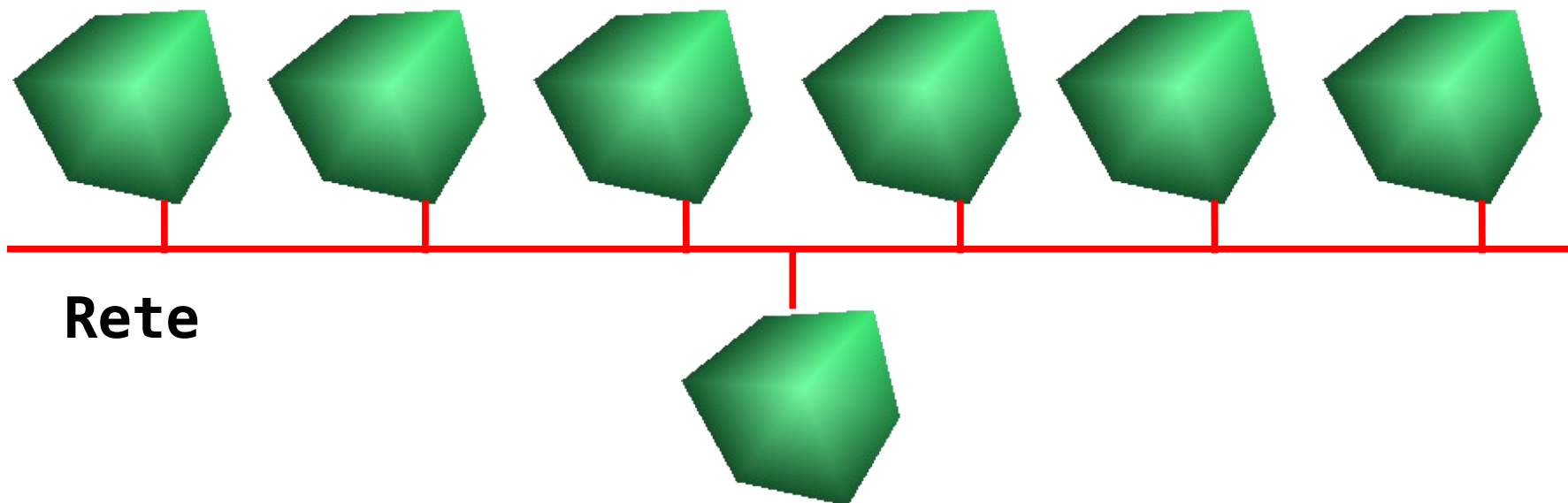
Broadcast



192.168.3.255

Il secondo di questi due indirizzi, detto “indirizzo di broadcast” (quello con tutta la parte host dell'indirizzo fatta di 1), serve per recapitare un pacchetto a tutti gli host della rete.

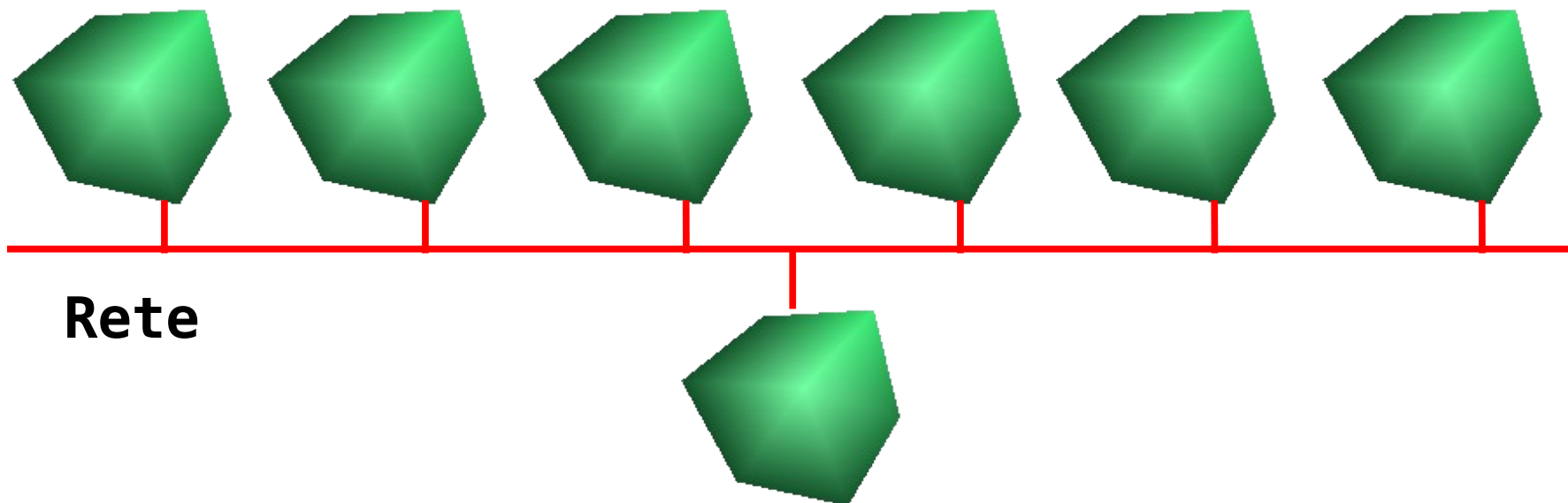
Broadcast



192.168.3.255

Se invio un pacchetto “alla broadcast” di una rete, questo verrà recapitato a tutti gli host della rete.

Broadcast



192.168.3.255

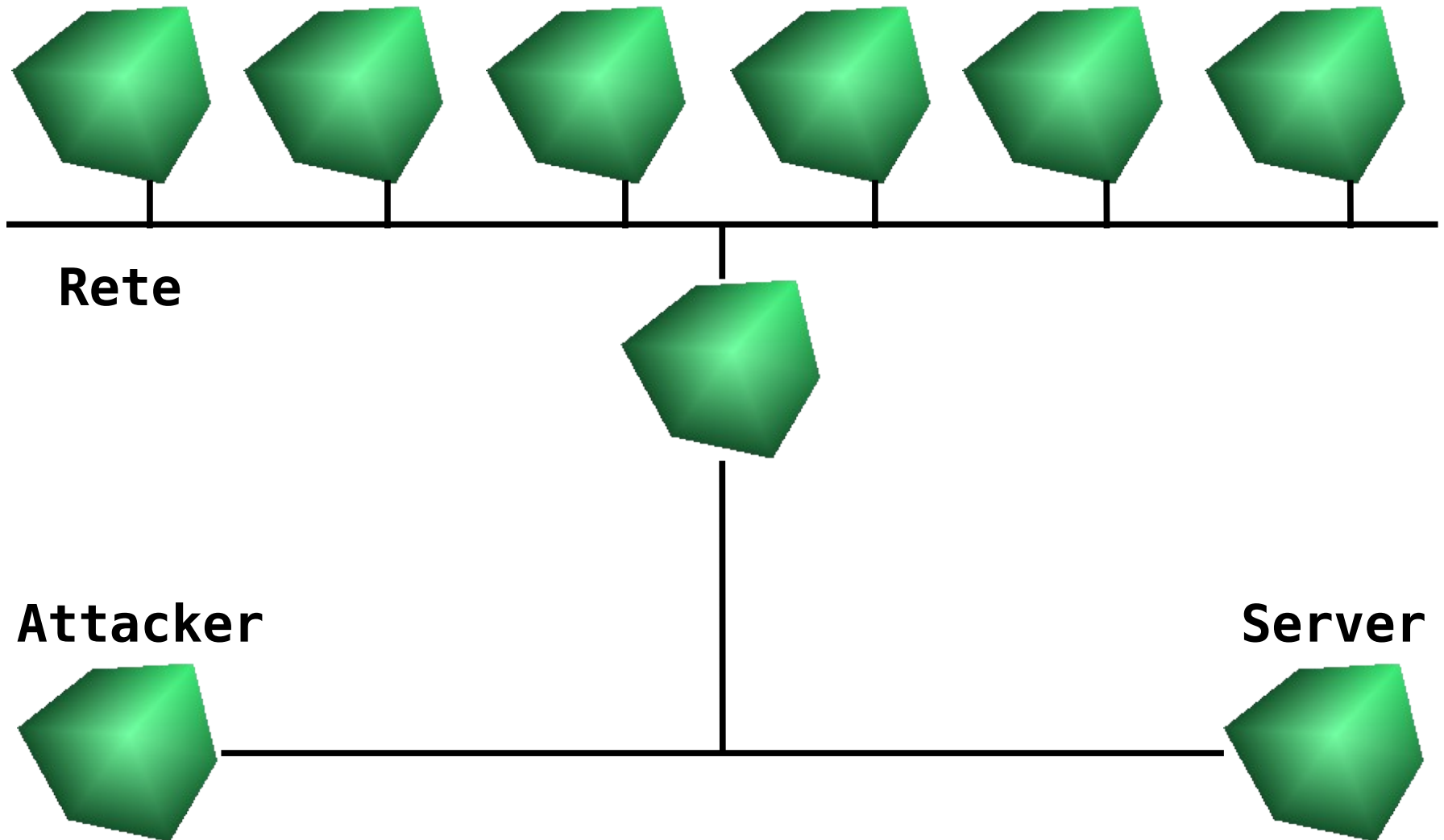
Supponiamo di inviare un pacchetto di tipo ICMP ECHO (il cosiddetto “ping”) alla broadcast di una rete.

Ogni host della rete, risponderà al mittente con un pacchetto ICMP ECHO REPLY.

Smurf

- E' un attacco DDoS (Distributed DoS) piuttosto atipico
- Si basa sullo sfruttamento delle broadcast che lasciano accesso dall'esterno della rete per amplificare gli effetti.
- Utilizza un indirizzo “spoofato”
- Il nome deriva da uno dei programmini che per primi sfruttavano questo genere di metodologia: lo Smurf2K

Smurf

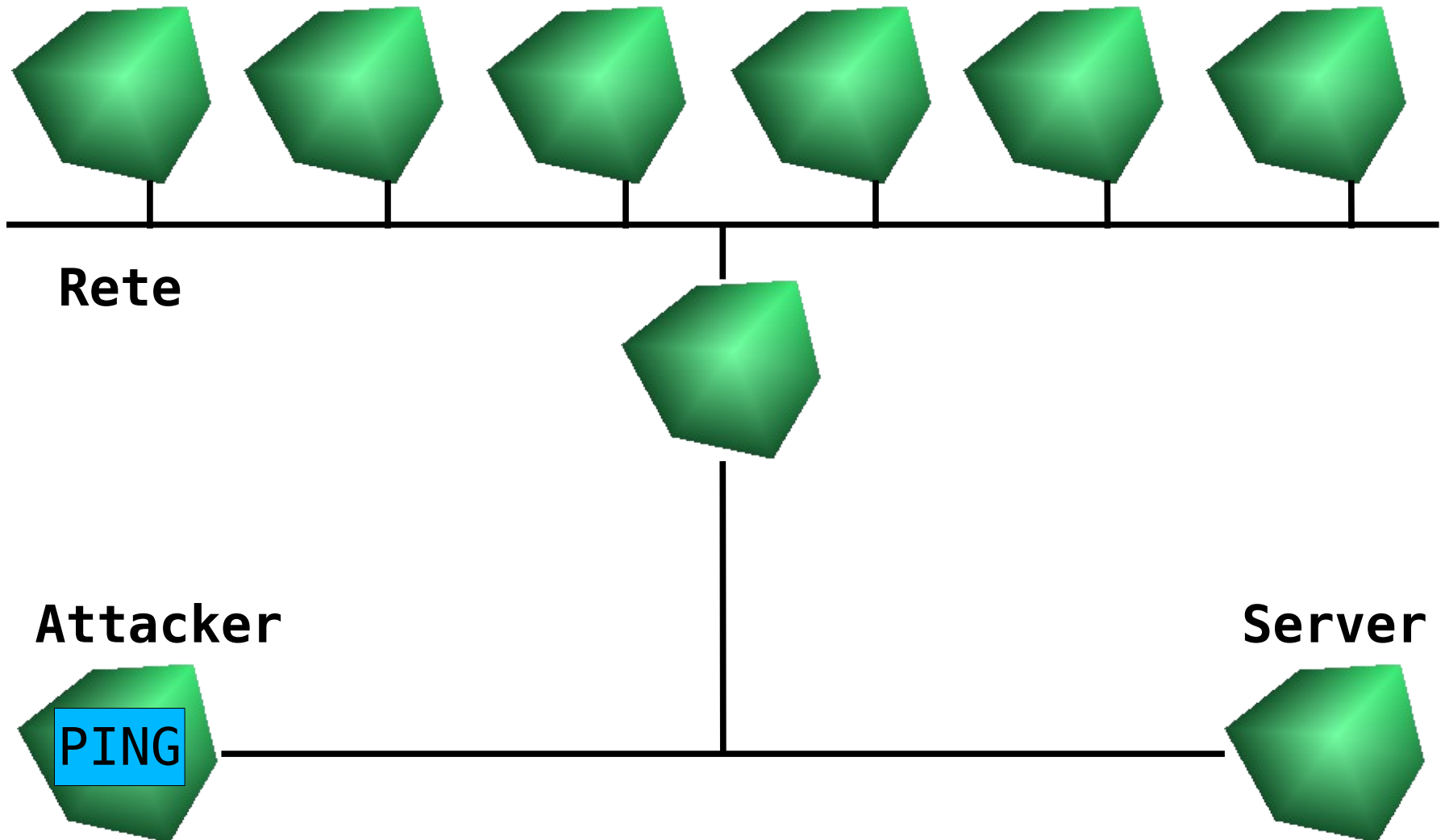


Smurf

- Un pacchetto ICMP ECHO viene imbustato, per la spedizione, in un frame IP.
- Questi presenta, oltre ai vari campi, i campi “Source Address” e “Destination Address”.
- Questi vengono utilizzati per indicare il mittente ed il destinatario del pacchetto che viene trasmesso.
- Se indico come destinatario del mio pacchetto l'indirizzo broadcast della rete, e come indirizzo del mittente, quello della vittima del mio attacco Smurf, otterrò che inviando un singolo pacchetto, tutti gli host della rete risponderanno all'attaccato, moltiplicando i pacchetti che io invio.

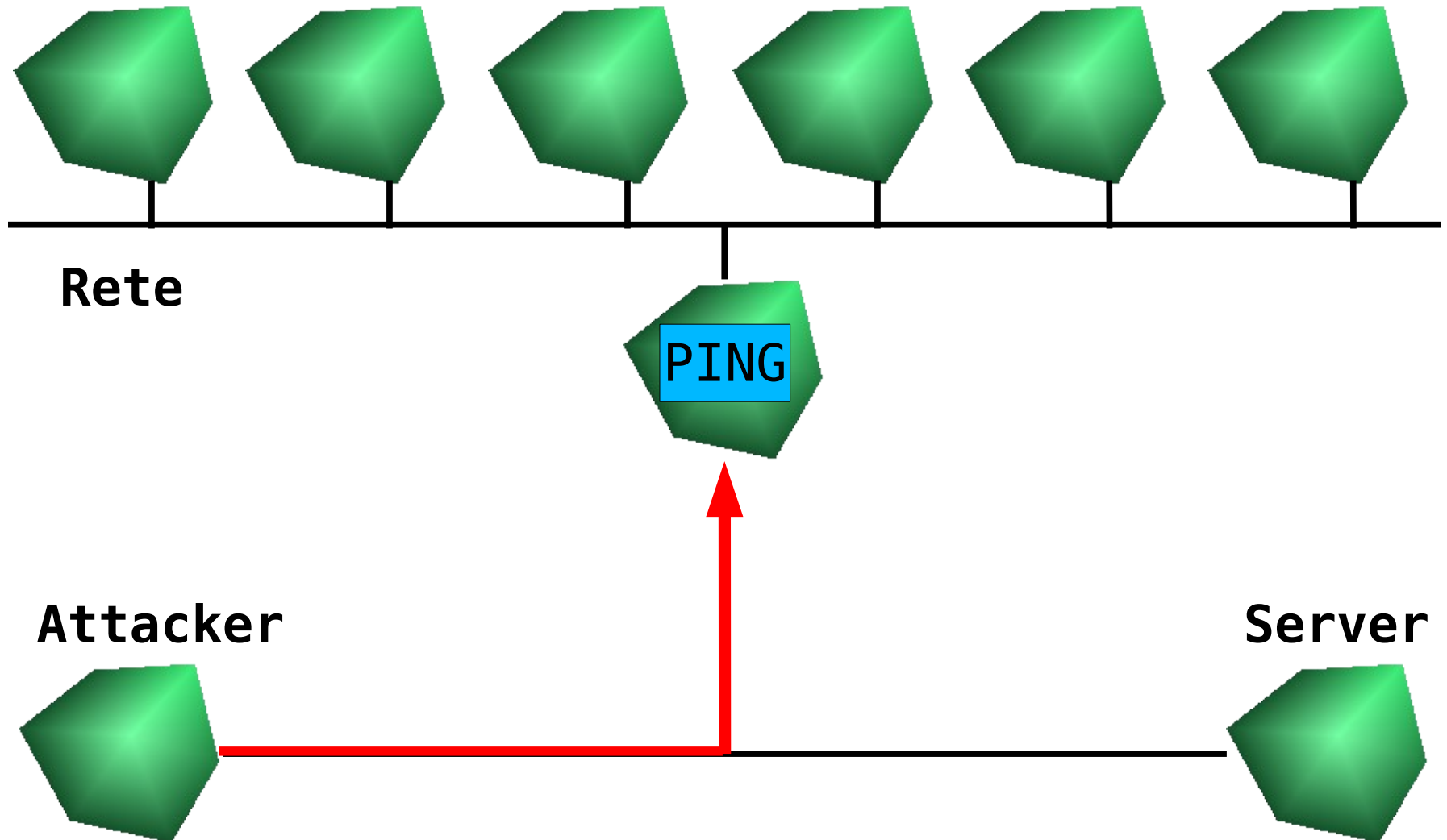
Smurf

Attacker genera un pacchetto "PING", indicando come "source" l'indirizzo di "Server"



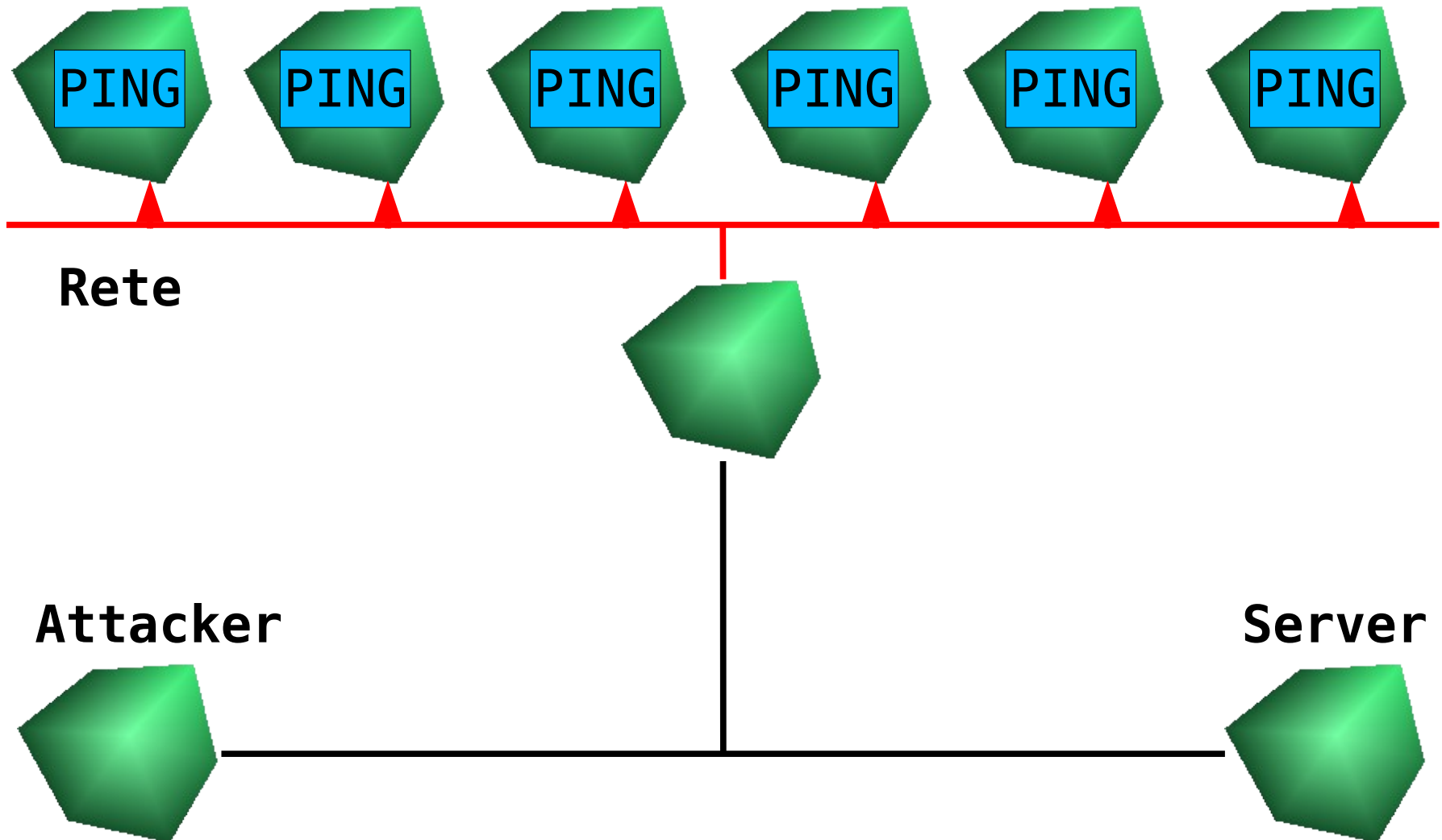
Smurf

Il destinatario è la broadcast di "Rete". Il pacchetto arriva al router.



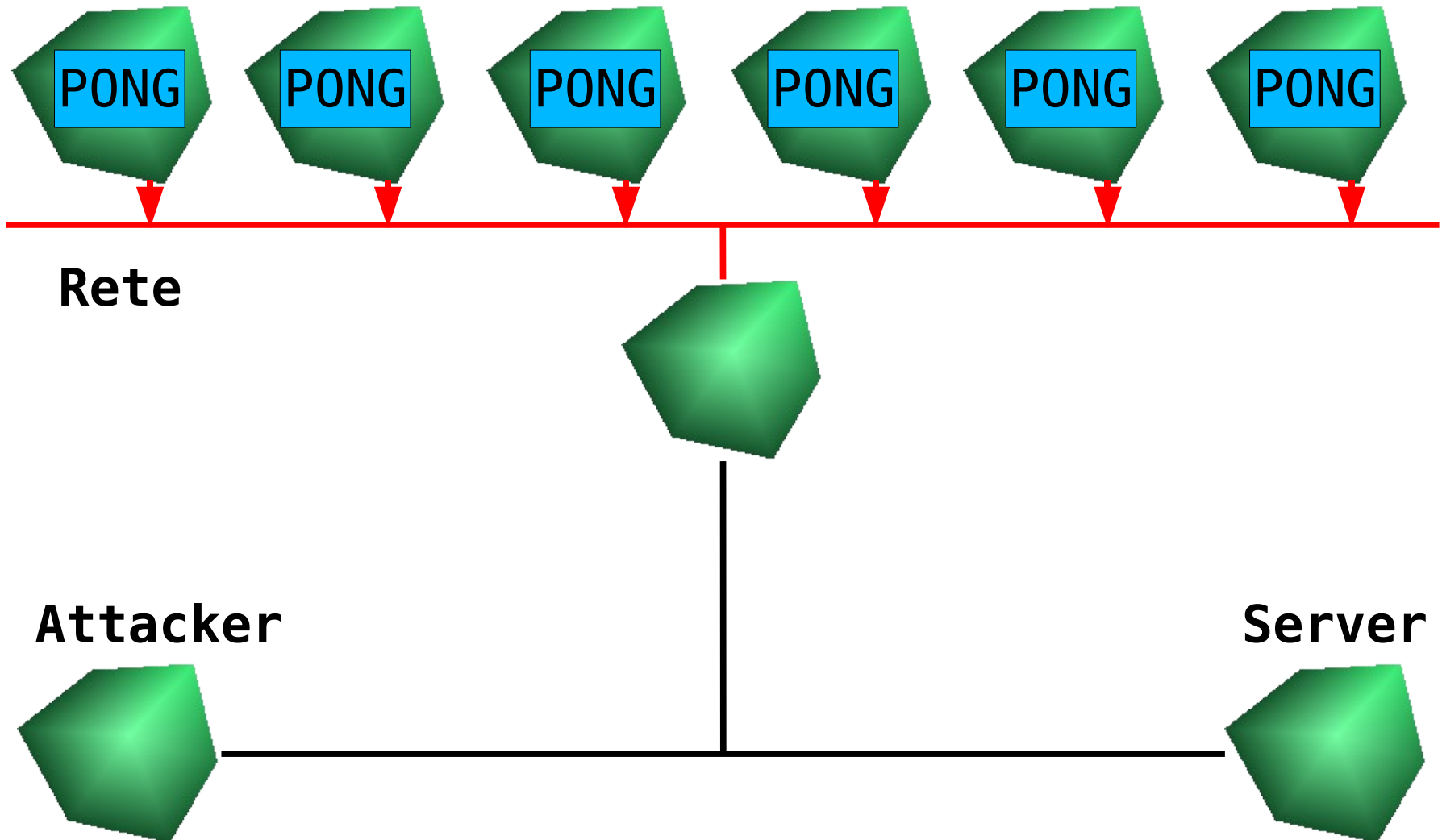
Smurf

Il router recapita il pacchetto a tutti gli host della rete.



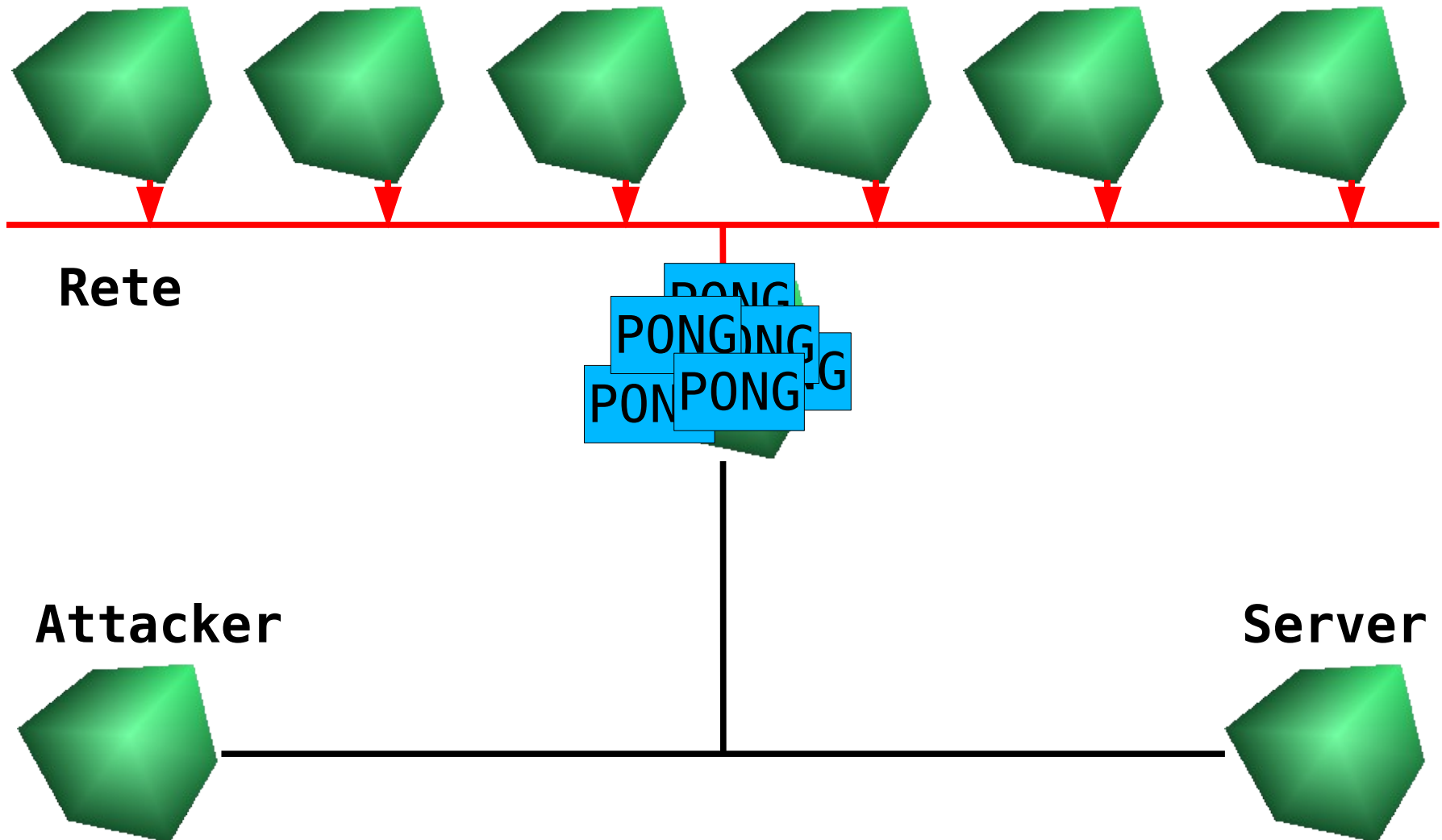
Smurf

Ogni host, risponde con un pacchetto “PONG” destinato al “source” del “PING” (“Server”!!)



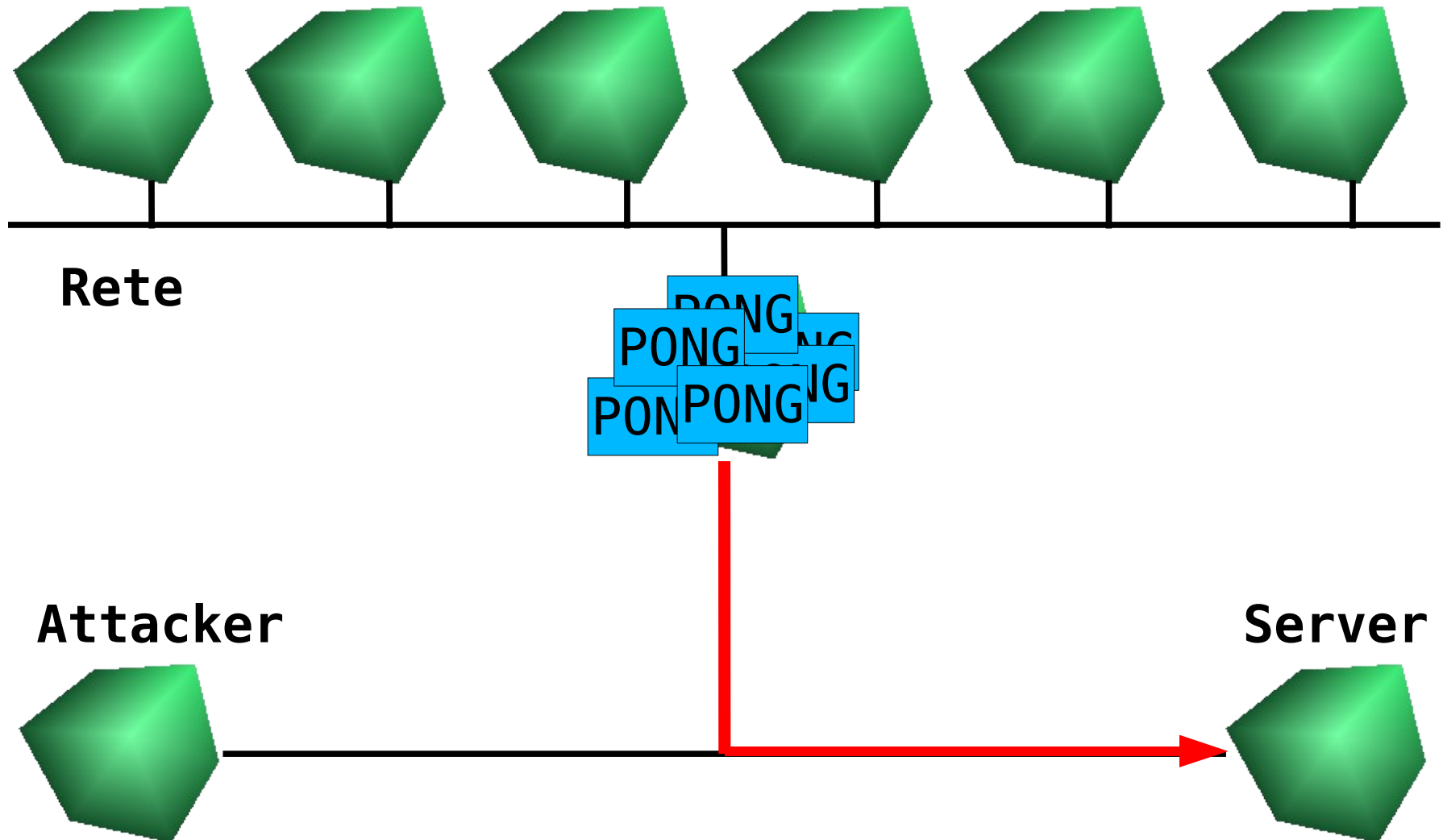
Smurf

I pacchetti arrivano al router, e questi lo spedisce a "Server".



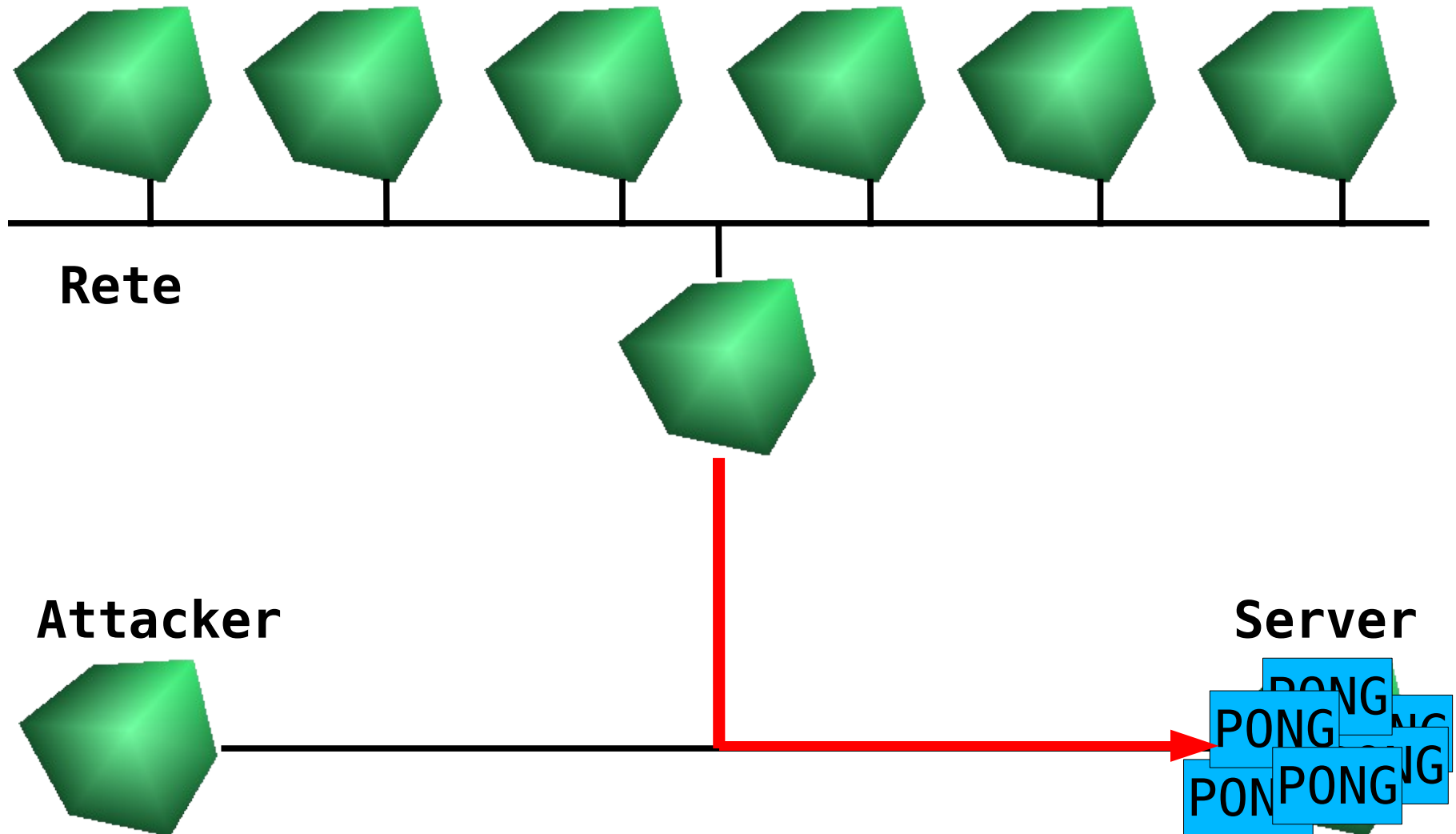
Smurf

I pacchetti arrivano al router, e questi lo spedisce a "Server".



Smurf

“Server” riceve una quantità di pacchetti proporzionale al numero di host di “Rete”.



Smurf

- **Vantaggi:**

- Effetto moltiplicazione
- Anonimo

- **Requisiti:**

- Una “broadcast” sufficientemente veloce da non “auto DoSsarsi” al momento di inviare i pacchetti al di là del router (intasandolo)
- Che la broadcast sia accessibile dall'esterno (che rappresenta un grave errore di configurazione da parte di chi amministra la rete)

DDoS

- **Due tipi di DdoS (Distributed DoS):**
 - “Sociale” (Hacktivismo, NetStrike)
 - “Tecnico” (Smurf)
- **Sociale:**
 - Tanti utenti si trovano (d'accordo o meno) a fare richieste ad uno stesso servizio (http, mail...)
- **Tecnico:**
 - Una “rete” di macchine compromesse che richiedono servizi a comando dell'attaccante (droni, zombies).

Prevenire

- Un attacco DoS basato su una metodologia di tipo Smurf, **NON SI PUO' PREVENIRE** (non lato vittima almeno).
 - La miglior prevenzione deve essere fatta dai provider e dai carrier, con tecniche di rilevamento dei DoS
 - Configurare i router che mettono in comunicazione reti diverse perchè non trasmettano pacchetti di broadcast a livello 3 (IP)
 - Configurare il proprio sistema operativo perchè non risponda agli ICMP diretti alla broadcast!
 - Tenere aggiornati i sistemi operativi connessi in rete. Questo riduce la quantità di droni, e quindi la potenza dei DdoS (mi rendo conto che su certi OS può essere davvero difficile)

Contromisure

- Le contromisure applicabili una volta che l'attacco è in corso, non possono che essere temporanee.
- Se l'attacco è stato fatto ed ha effetto (al punto da richiedere contromisure), può essere ripetuto quando si vuole.
- Bisogna limitare il traffico in ingresso
 - Se l'attacco proviene da un host/rete, è facile: si chiede al proprio provider/carrier di bloccare quel traffico da quella sorgente, magari direttamente alla fonte (sfruttando accordi tra carrier diversi)
 - Resta il problema dell'affidabilità del carrier (Telecom?)
 - Se si tratta di un attacco distribuito, c'è poco da fare. Come discernere infatti il traffico lecito da quello nocivo?

Evoluzione...

- Ricapitolando i problemi dello smurf:
 - Broadcast accessibile da fuori
 - Si tratta di un errore di configurazione sempre meno frequente
 - Unicità
 - Tutto si svolge nel mondo dell'ICMP Echo
 - Basta che il provider “droppi” quei pacchetti, e l'attacco è finito
- Anche il DdoS tramite zombies non è una cosa facile
 - Bisogna costruirsi una rete di zombies, compromettendo dei sistemi per assogettarli al proprio controllo
 - Bisogna poterli comandare
- In entrambi i casi, i sistemi che fanno flood sono facilmente localizzabili, ed è facile risalire al punto di controllo.

Evoluzione...

- Ci servirebbe allora:
 - Un generatore di pacchetti che non abbia bisogno di essere compromesso, liberamente accessibile
 - Che sia un distributore di pacchetti legittimo, in modo da non poter risolvere il problema, una volta segnalato, correggendo un eventuale errore di configurazione (come le broadcast)
 - Avere la possibilità di modificare la tipologia di pacchetti (porte, protocolli) che inviamo
- Qualcuno di voi ha una risposta?

Evoluzione...

- Ci servirebbe allora:
 - Un generatore di pacchetti che non abbia bisogno di essere compromesso, liberamente accessibile
 - Che sia un distributore di pacchetti legittimo, in modo da non poter risolvere il problema, una volta segnalato, correggendo un eventuale errore di configurazione (come le broadcast)
 - Avere la possibilità di modificare la tipologia di pacchetti (porte, protocolli) che inviamo
- Qualcuno di voi ha una risposta?

**TUTTO CIO' CHE OFFRE UN SERVIZIO SU INTERNET
SERVER, ROUTER...**

Evoluzione...

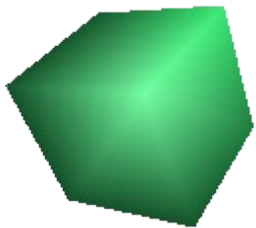
- Server di posta: 25 (smtp), 110 (pop3), 220 (imap3), ...
- Server web: 80 (http), 443 (https), 3306 (mysql), ...
- Server ftp: 20/21 (ftp), 989/990 (ftps), ...
- Server DNS: 53 (dns), ...
- Time servers: 119 (ntp)
- Char servers: 6667 (IRC), 5190 (AOL), ...
- Streaming sever: 554 (RTSP)
- Backup server: 873 (rsync)
- Usenet server: 119 (nntp), ...
- Proxy server: 3128, 8080, ...
- Routers: 179 (BGP), 161 (SNMP)

Evoluzione...

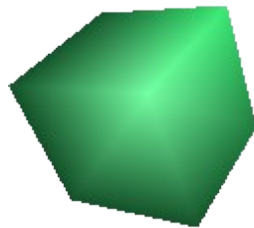
- Cos'hanno in comune?
- Se le scegliamo minimamente bene:
 - Sono tutte macchine molto potenti, che non risentirebbero minimamente di qualche nostro pacchetto
 - Sono tutte macchine con connessioni veloci
 - Sono tutti servizi offerti al pubblico, quindi le nostre richieste sono “perfettamente legittime” (apparentemente) e non c'è bisogno di compromettere nulla
 - Sono tutti servizi diversi: porte diverse, protocolli diversi
 - Sono molto diffusi: è facile trovarne in ogni angolo della rete ed ancora più facile raccogliarli in liste (spider)

SYN Reflection

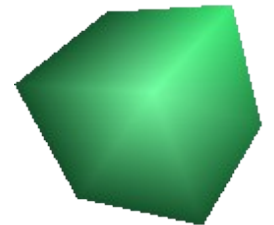
Attacker



SMTP



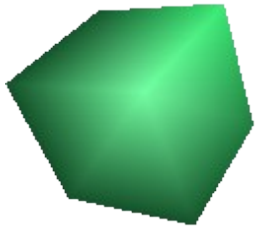
Victim



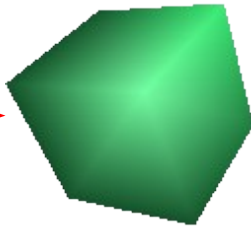
SYN Reflection

```
FROM: $IP_VICTIM:*  
TO: $SMTP_SERVER:25  
TYPE: SYN
```

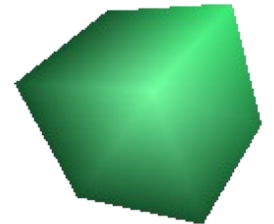
Attacker



SMTP



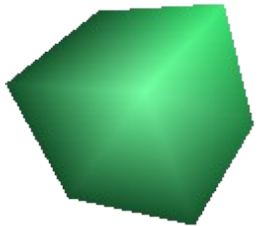
Victim



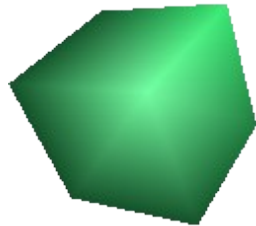
SYN Reflection

```
FROM: $SMTP-SRV:25  
TO: $VICTIM:*  
TYPE: SYN-ACK
```

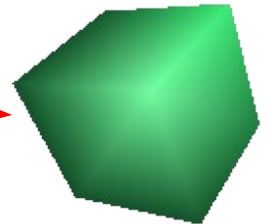
Attacker



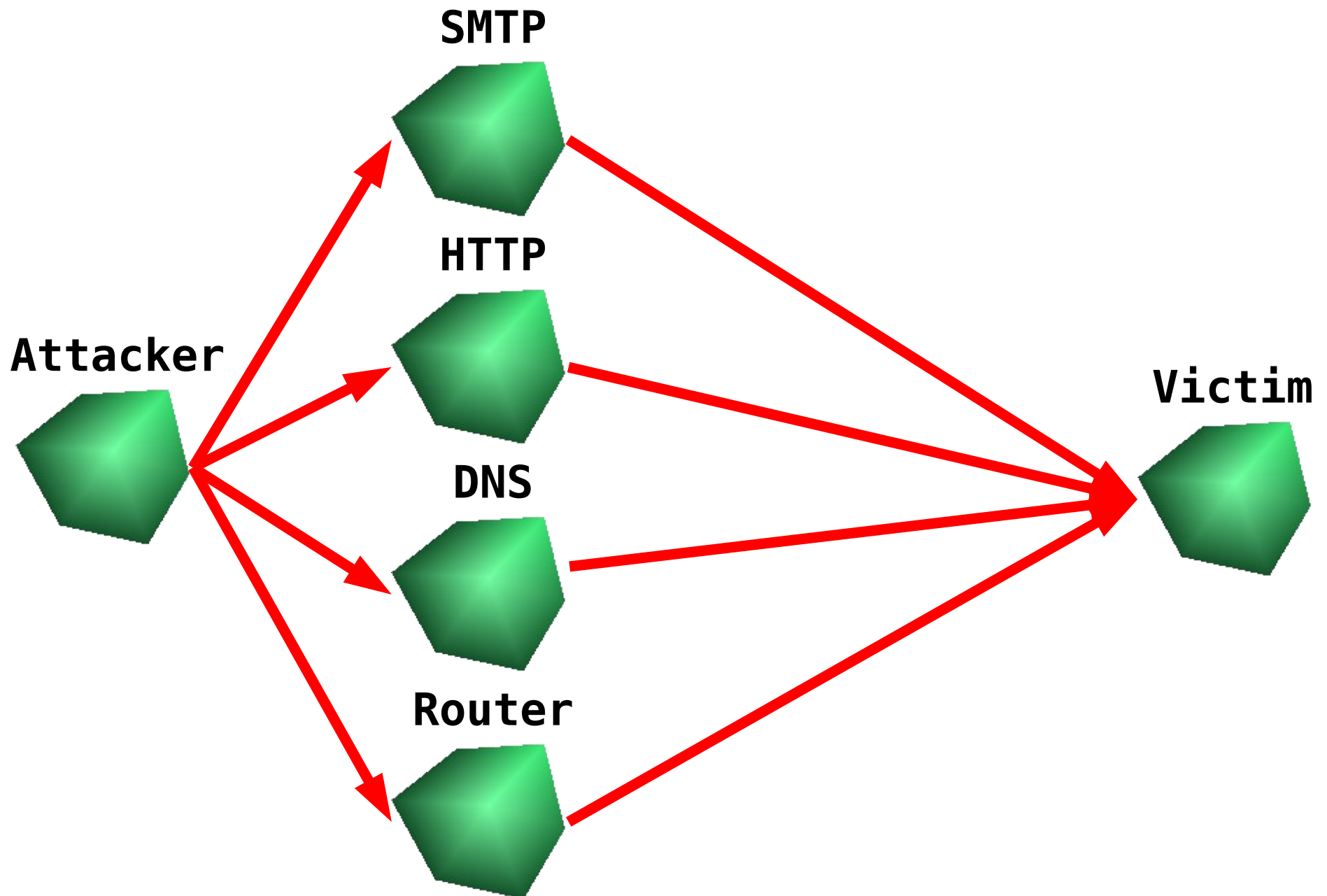
SMTP



Victim

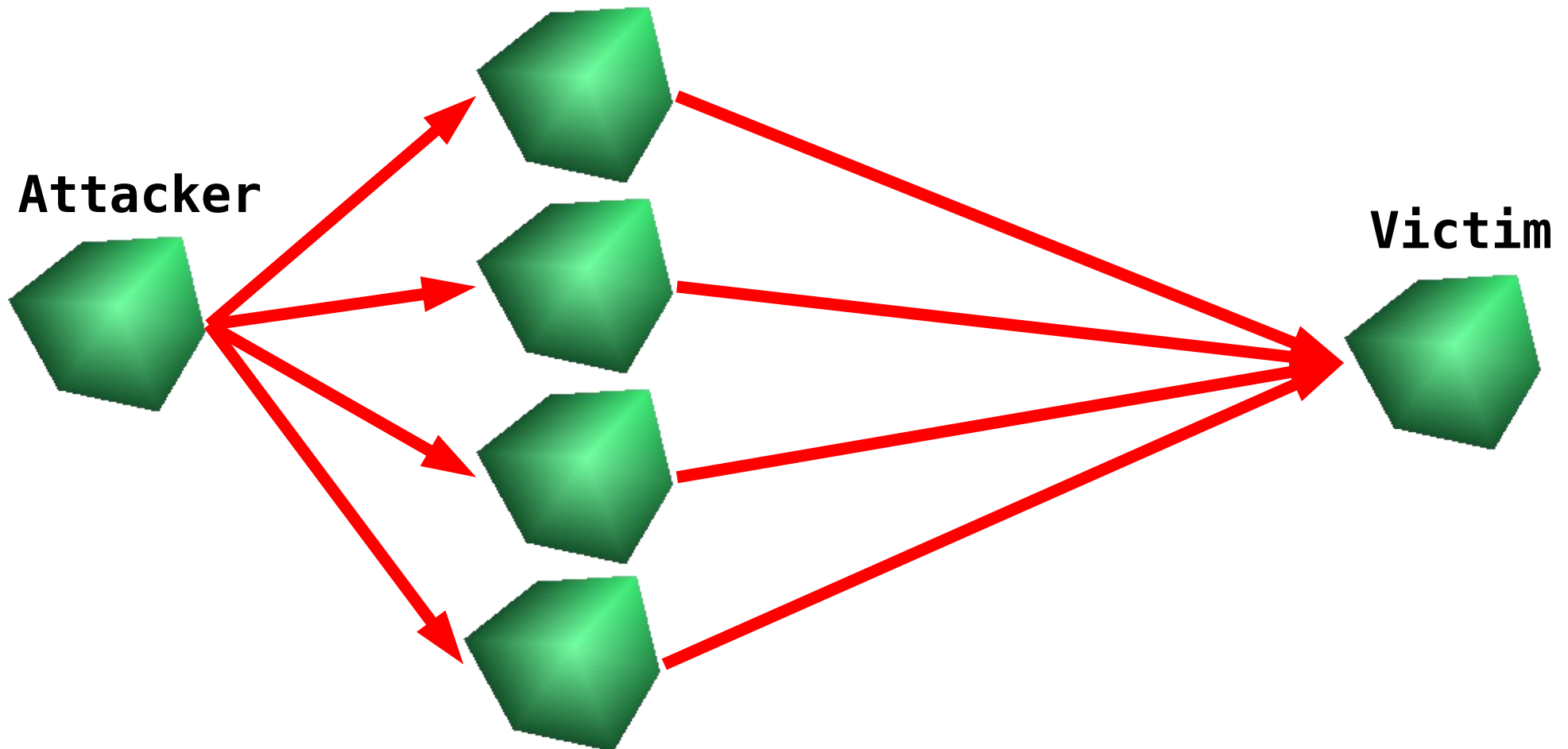


Multiple SYN Reflection



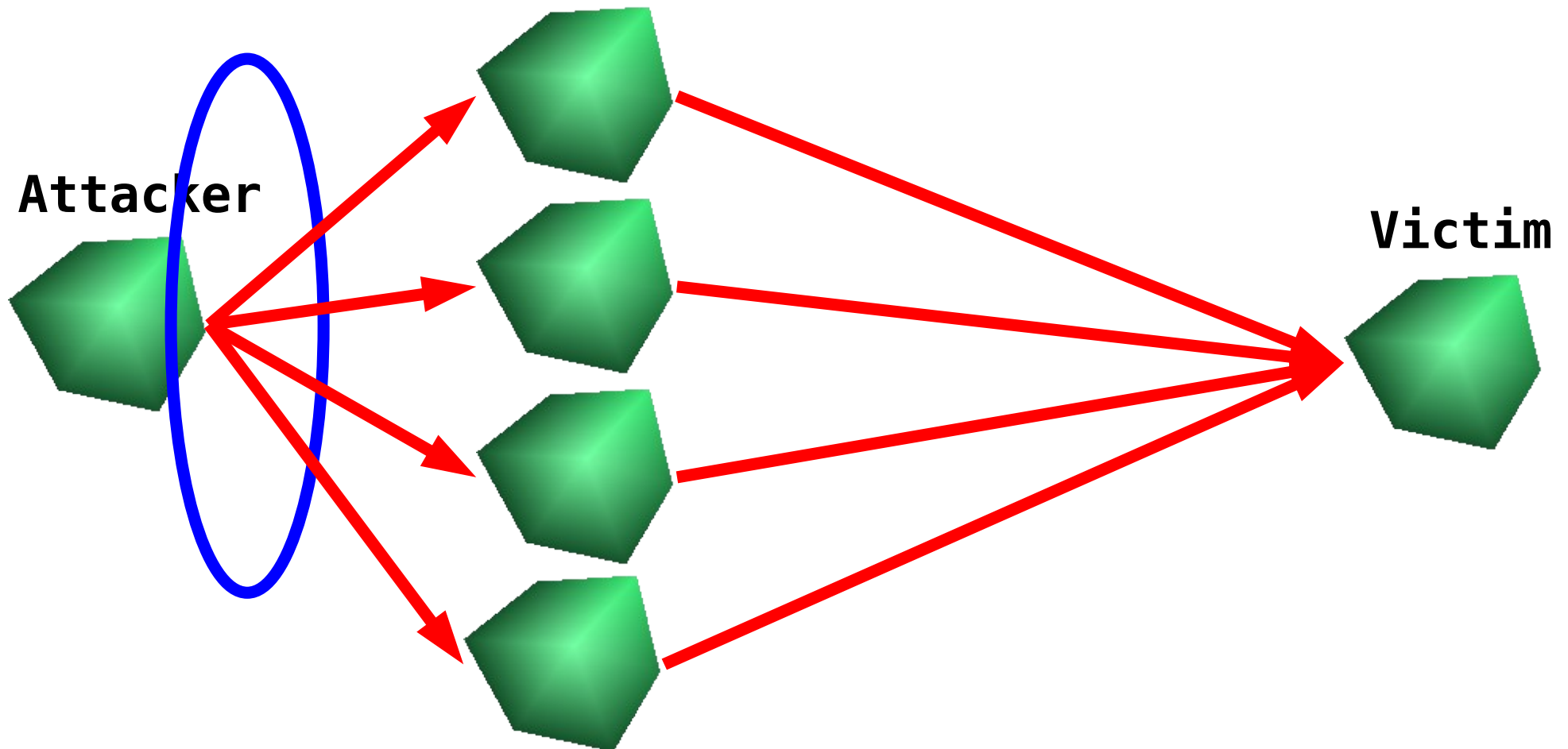
Vantaggi?

- Ora la vostra domanda sarà: che vantaggio c'è rispetto ad un SYN Flood?



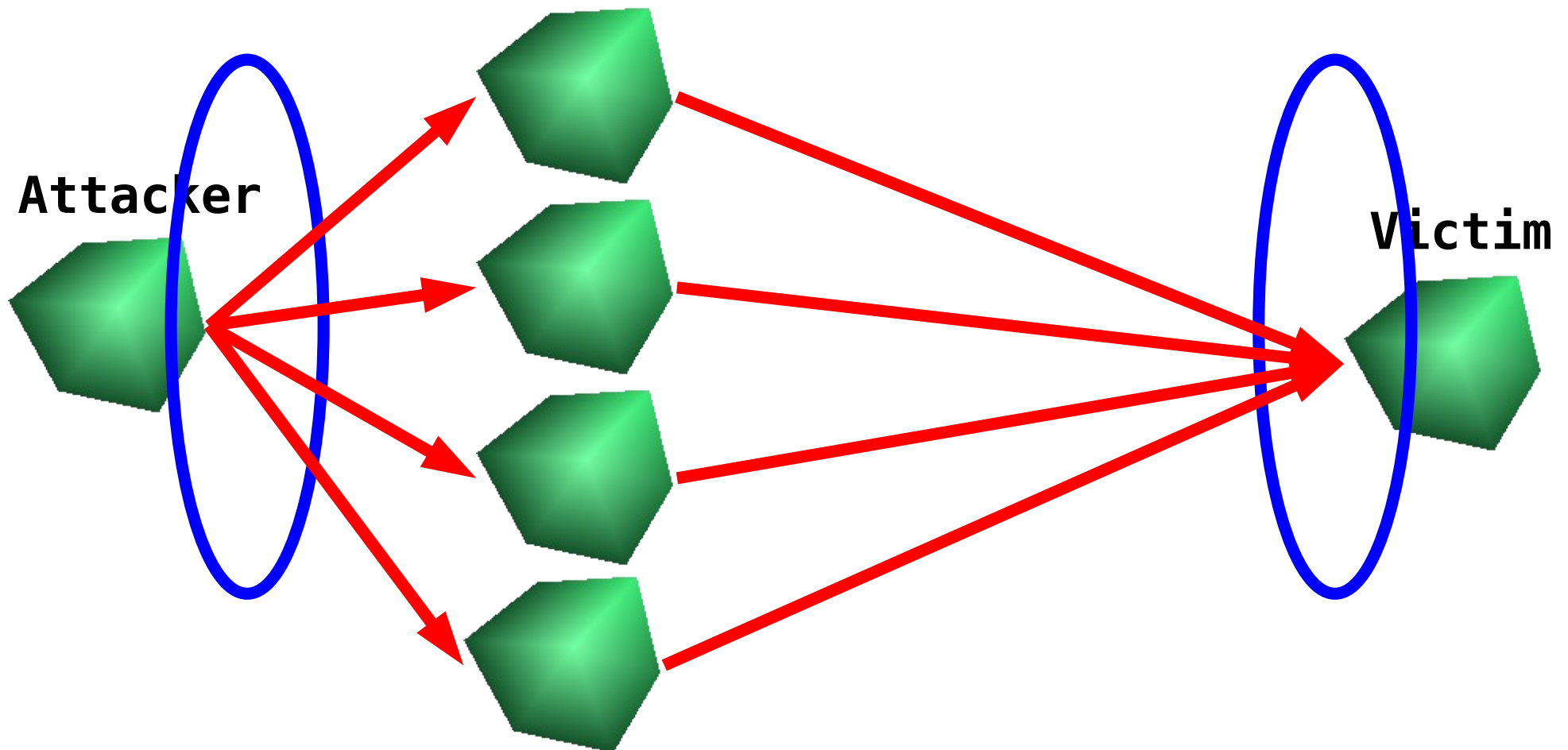
Vantaggi?

- Ora la vostra domanda sarà: che vantaggio c'è rispetto ad un SYN Flood?

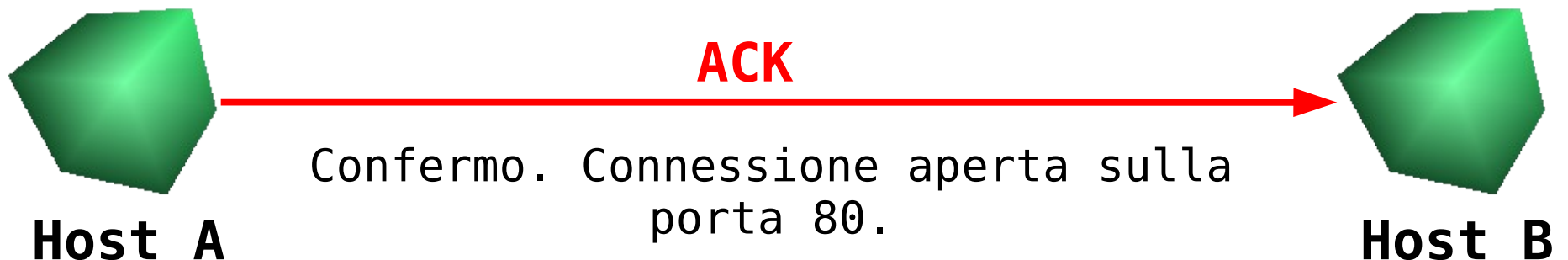


Vantaggi?

- Ora la vostra domanda sarà: che vantaggio c'è rispetto ad un SYN Flood?



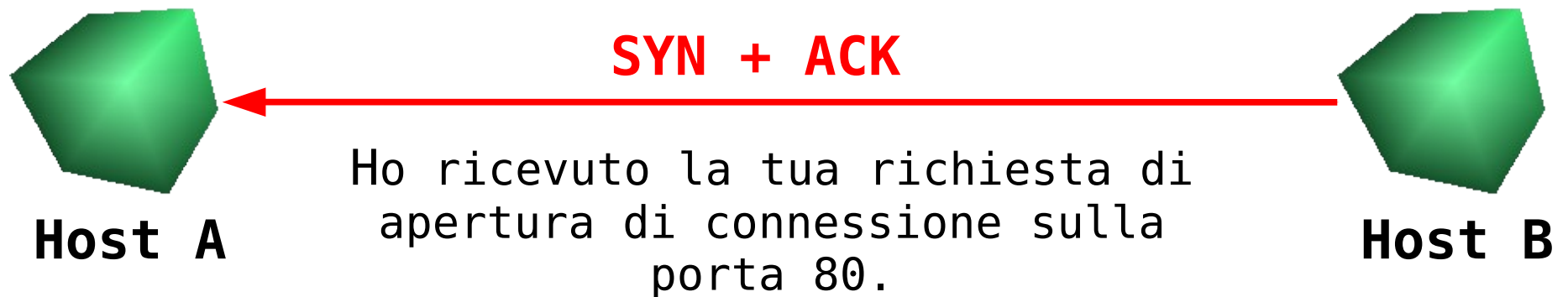
TCP: apertura connessione



TCP: errore?



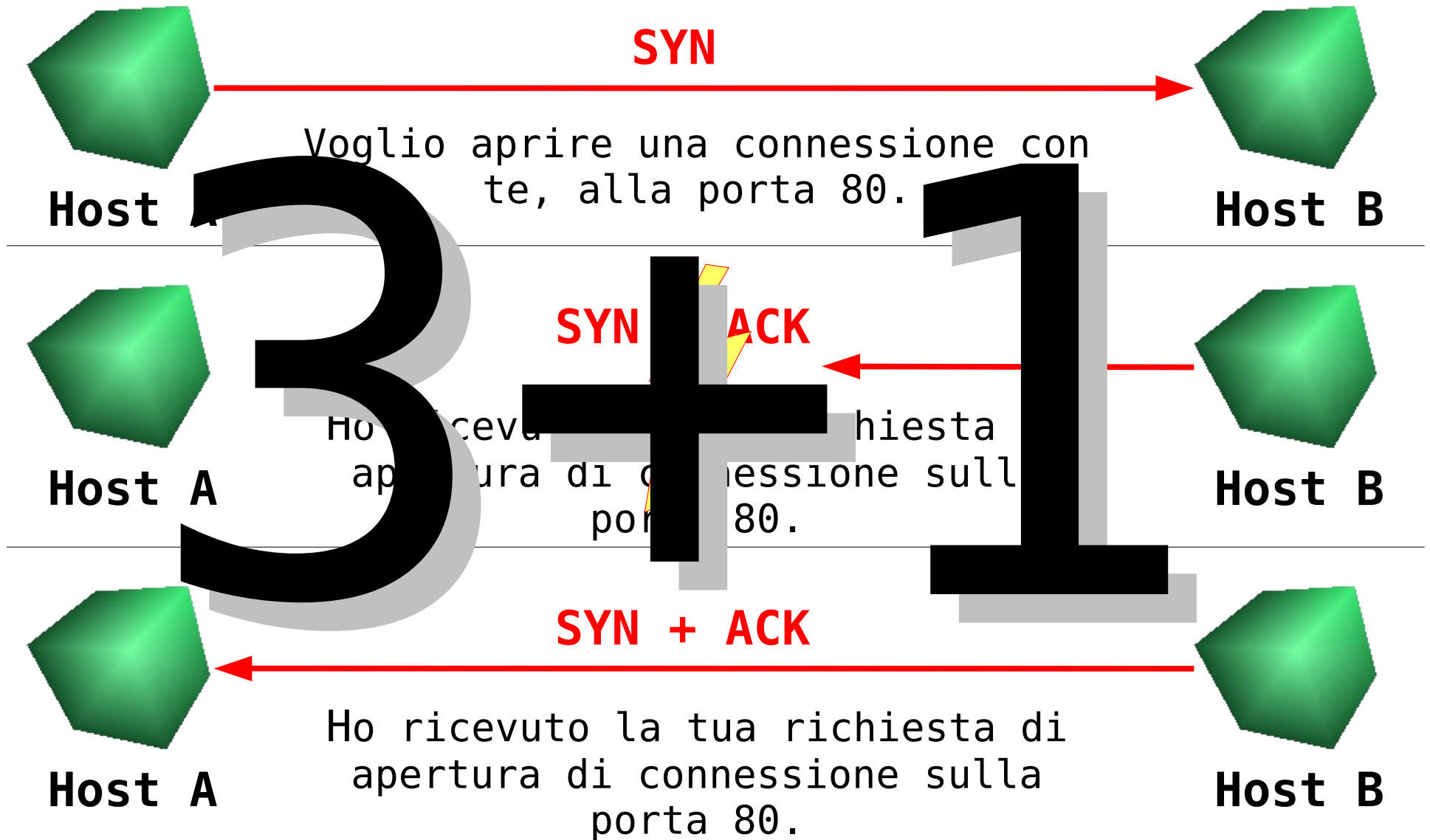
TCP: ritrasmissione!



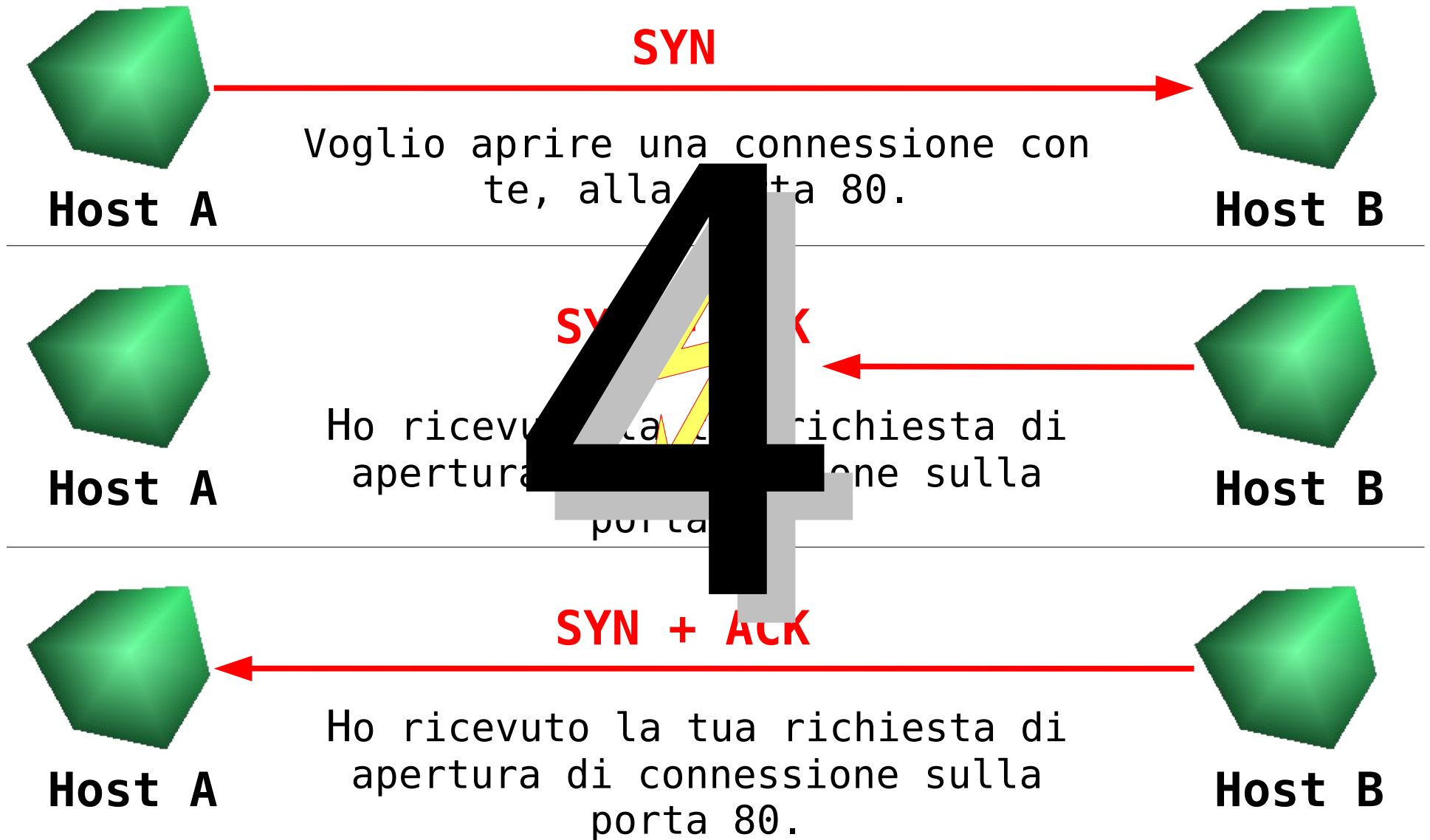
Quante ritrasmissioni?



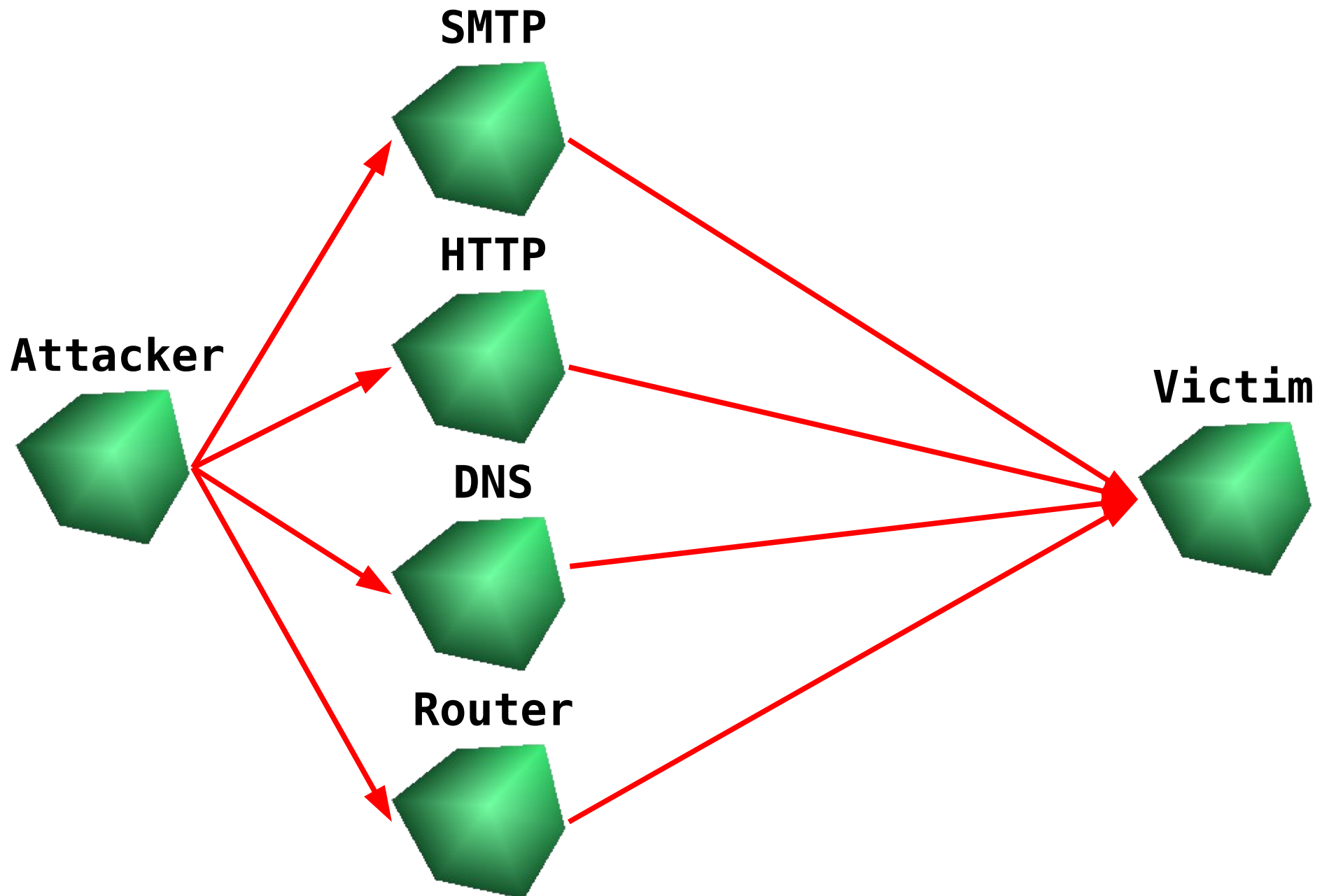
Quante ritrasmissioni?



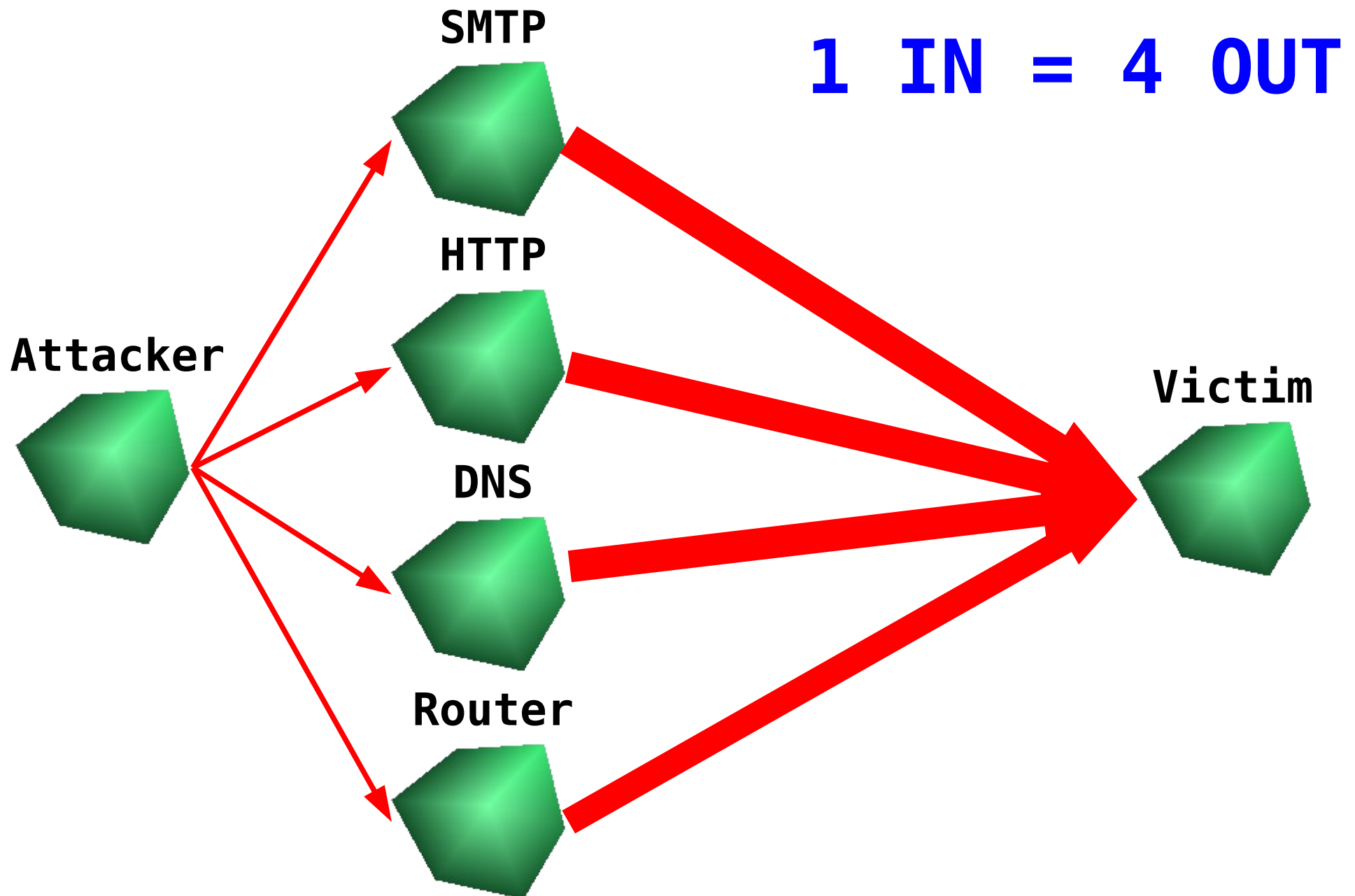
Quante trasmissioni?



Effetto moltiplicazione



Effetto moltiplicazione



DRDoS

- Abbiamo ottenuto un DoS
 - Otteniamo l'effetto moltiplicazione sfruttando le ritrasmissioni del TCP
 - Otteniamo banda e velocità utilizzandone poca per un alto numero di server
 - I servizi che utilizziamo non possono essere “disabilitati” (altrimenti sarebbe un DoS!)
 - Non siamo facilmente rintracciabili (è difficile seguire un piccolo flusso di pacchetti, soprattutto perchè la destinazione è la vittima!)
 - Possiamo rendere intermittente il flusso, chiamando liste di server diverse ad intervalli regolari per rendere ancora più difficile il lavoro di backtracking

Distributed Reflection Denial of Service

Rose

Attacchi Rose

- Attacchi molto recenti: la teoria è piuttosto vecchia, ma la sua fattibilità è stata dimostrata solo recentemente (Paul A. Watson, Aprile 2004).
- Sfrutta alcuni errori nella classica implementazione del TCP per “resettare” le connessioni.
- Attacco comunque piuttosto teorico, in quanto difficile da realizzare la dove ci siano dei router ben configurati a controllare il traffico.

TCP: vita di una connessione

- La vita di una connessione TCP comincia con (li abbiamo visti) il così detto “3 way handshaking”, durante il quale viene “deciso” un numero di sequenza (teoricamente casuale, deciso da chi apre la connessione) che identificherà univocamente ogni pacchetto.

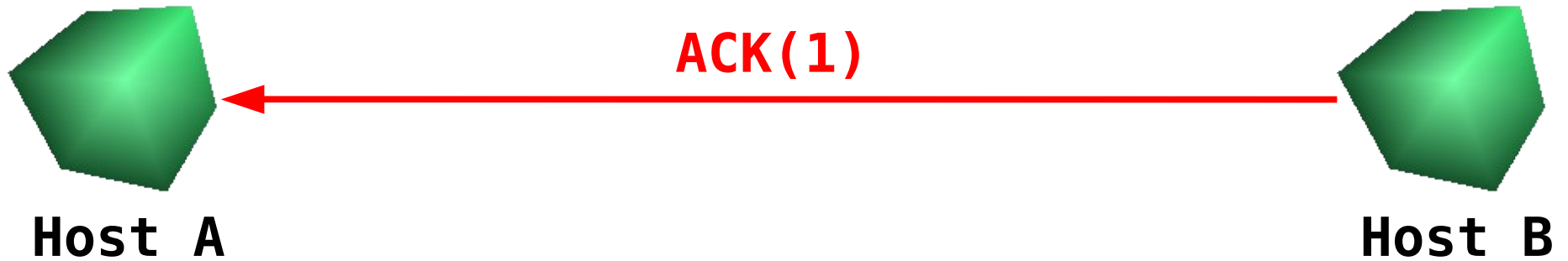
Viene inoltre definita una “finestra” (campo window dei pacchetti di apertura, deciso da chi apre la connessione, più o meno fisso), necessario per i controlli di congestione e flusso implementati in TCP

- Infine, ogni pacchetto viene caratterizzato da un “numero di sequenza” che lo identifica univocamente all'interno della connessione

TCP: gestione della finestra

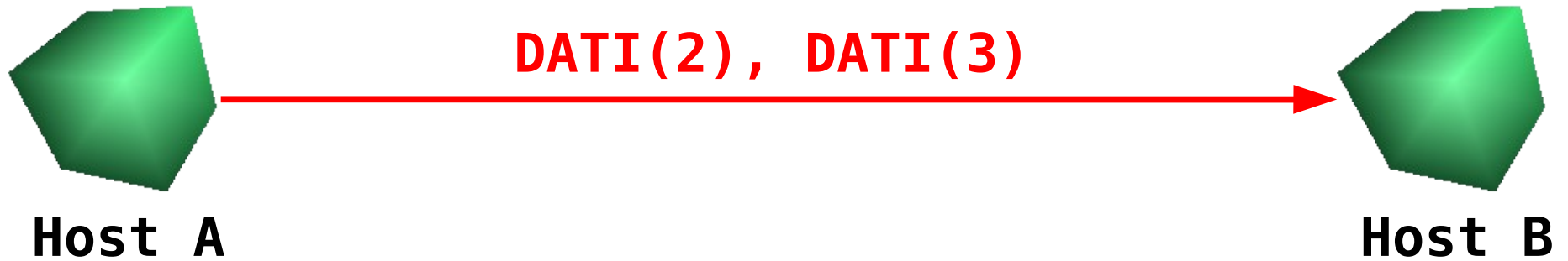


TCP: gestione della finestra



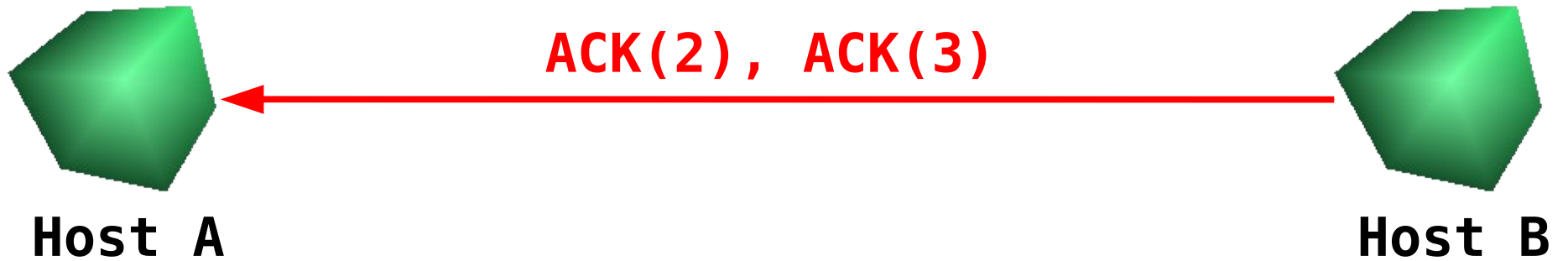
WINDOW: 1

TCP: gestione della finestra



WINDOW: 2

TCP: gestione della finestra

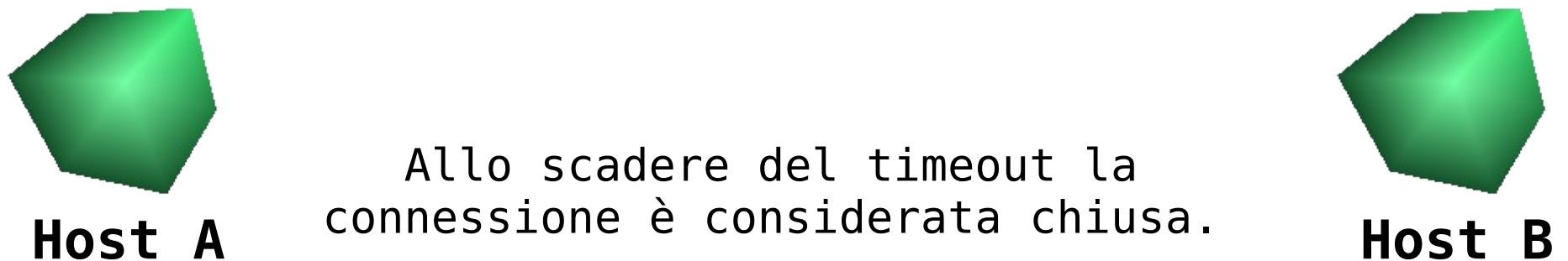
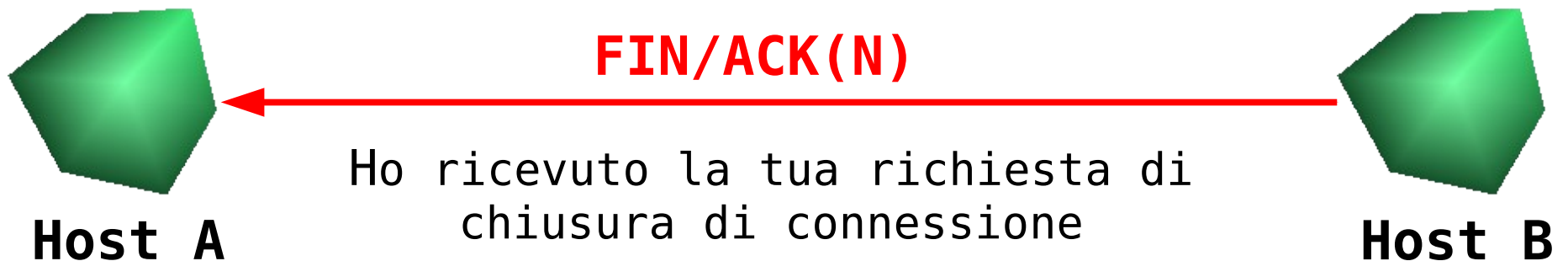
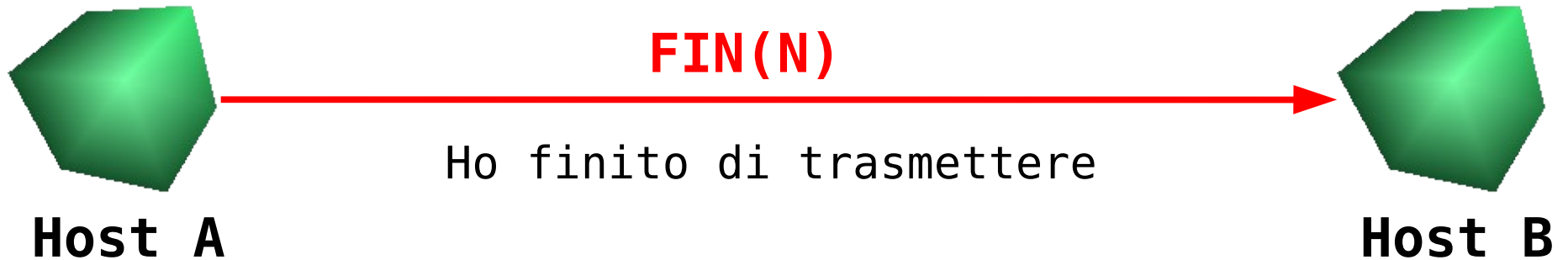


WINDOW: 2

TCP: vita di una connessione

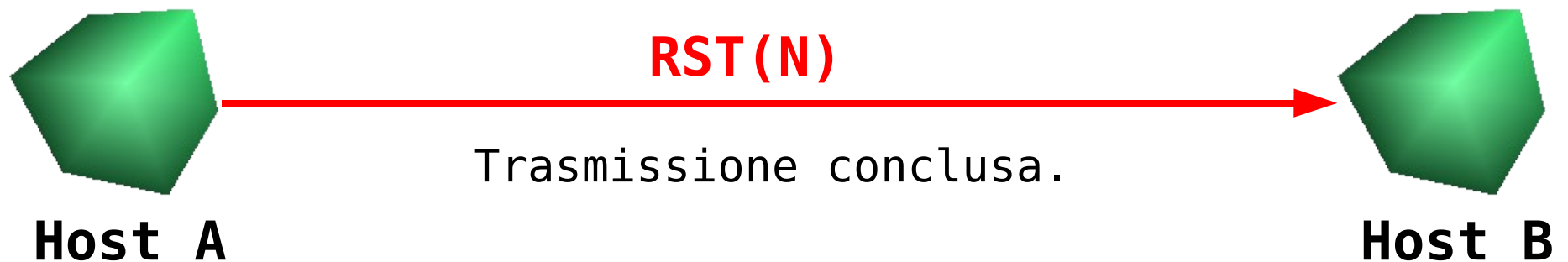
- Segue tutto il traffico della connessione, con i vari pacchetti inviati e gli ACK di conferma da parte del ricevente (si controlla che tutti i pacchetti all'interno della finestra di connessione siano ricevuti correttamente).
 - La variazione di dimensione della window serve a controllare la quantità di pacchetti che viaggiano lungo la connessione. Una window a 0, significa che non potranno viaggiare pacchetti, pur avendo la connessione aperta (magari a causa della congestione della coda di qualche router che perde pacchetti).
 - La trattazione del controllo di congestione e flusso esula da questa presentazione.

TCP: chiusuta connessione



TCP: vita di una connessione

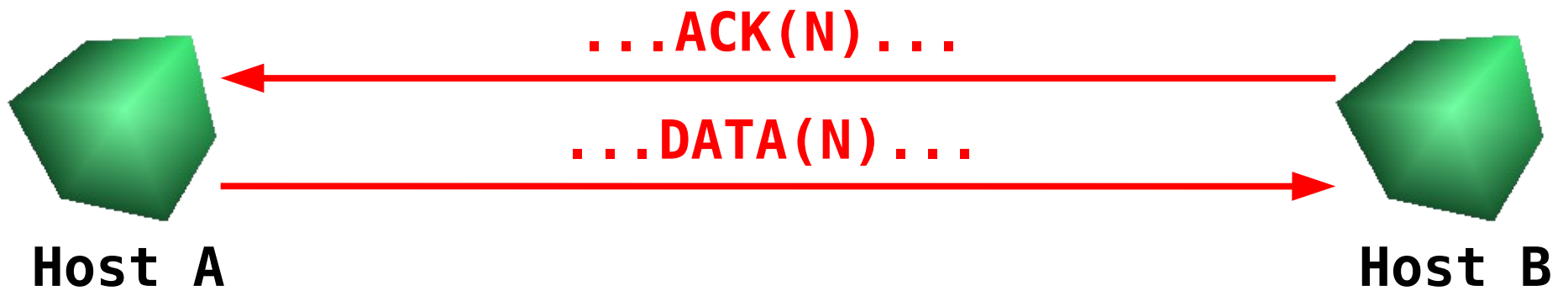
- Esiste un altro modo per chiudere una connessione TCP, un po' più brutale: si tratta dei pacchetti RST (reset). Quando un pacchetto RST viene inviato da una macchina, rende persino superfluo il pacchetto di ACK dall'altro lato. La connessione si considera chiusa immediatamente.



Rose Attacks

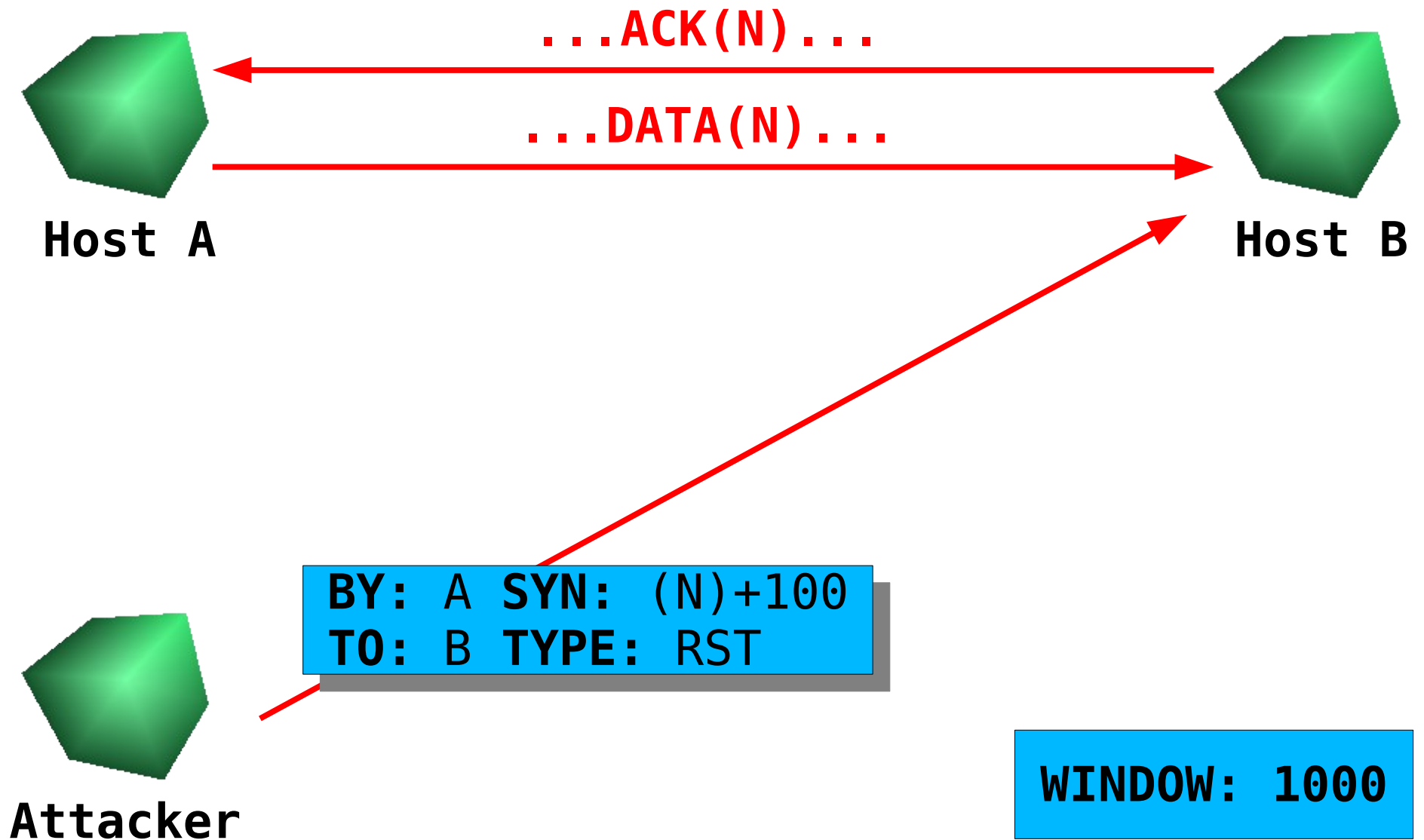
- Nonostante sia semplice intuire che quando un attaccante riesce a mandare un pacchetto RST con indirizzo sorgente soopfato (falsificato) la connessione possa cadere, questa è da sempre stata considerata una vulnerabilità completamente teorica, visto che è molto difficile indovinare l'esatto Sequence Number (pseudo-casuale e cambia molto velocemente), in modo che il pacchetto RST falsificato sia scambiato per uno vero.
- Uno studio di Paul A. Watson, pubblicato nell'Aprile 2004, ha però fatto notare che non serve che l'SN del pacchetto RST sia corretto, basta che ricada nella Window! Questo riapre di colpo la fattibilità di questi attacchi.

Rose Attacks



WINDOW: 1000

Rose Attacks



Rose Attacks

- La procedura non è utile là dove ci sono connessioni brevi. Indovinare il SN infatti, richiede connessioni di lunga durata (ad esempio le connessioni BGP delle backbone di Internet)
- Più la connessione è lunga infatti, più è facile che la window assuma una notevole dimensione (legata alla quantità di traffico) rendendo più probabile che il nostro RST finisca all'interno della window
- In più, i generatori di numeri non sono mai realmente casuali, il che rende leggermente più semplice l'individuazione dell'SN
- Vista la difficoltà dell'attacco, le conseguenze non vanno solitamente al di là della necessità di riaprire la connessione di tanto in tanto.

Precauzioni

- Visto che si tratta di un tipo molto recente di attacchi, non è facile dare risposte per quel che riguarda la prevenzione di questi attacchi. Sicuramente avere una rete in cui i router impediscono l'invio di pacchetti con mittente non corretto, risolve a priori gran parte di questi (e non solo) attacchi.
- Esiste inoltre la possibilità di utilizzare IPsec (o Ipv6), che consentendo la trasmissione di pacchetti crittografati e/o firmati, consente di controllare l'autenticità del pacchetto in modo più sicuro.
- Alcune di queste tecniche vengono già utilizzate da parecchi anni per la protezione delle comunicazioni tra backbones.

Credits

- A Matteo Falsetti (Hackers&C) per l'articolo apparso sul n° 8 che ha ispirato questa presentazione.
- A Gerardo di Giacomo (aka. astharot, Zone-H) per il suggerimento sull'inserimento degli attacchi LAND e Rose.

Per domande e/o suggerimenti
chiedere ora, oppure:

Giacomo Rizzo
alt-os@poul.org