Sicurezza delle reti[1]

Mattia Monga

Dip. di Informatica e Comunicazione
Università degli Studi di Milano, Italia
mattia.monga@unimi.it

a.a. 2010/11

Lezione IV: Port scanning

Port scanning

La conoscenza di quali *porte* sono accessibili per connessioni TCP o UDP è molto importante perché identifica i possibili canali di comunicazione:

- per scopi di difesa (quali applicazioni monitorare)
- e attacco (quali canali utilizzare)
- "Cartografia di reti e servizi": quando l'amministratore della rete non ha altre informazioni dirette

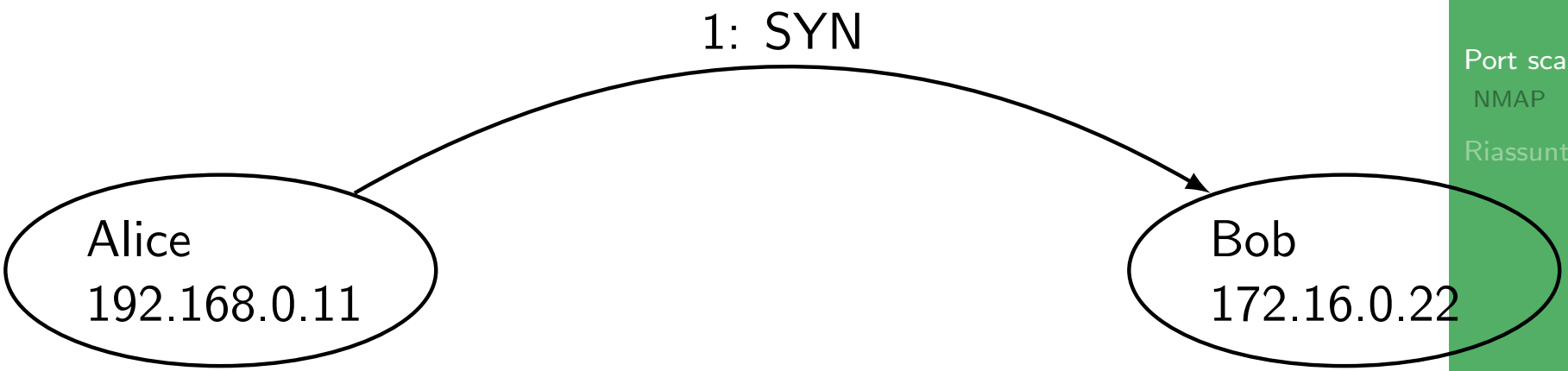
Stato di una porta

open Esiste la possibilità di connessione con un'applicazione (non necessariamente quella standard!)

closed La porta è accessibile, ma non c'è nessuna applicazione in ascolto

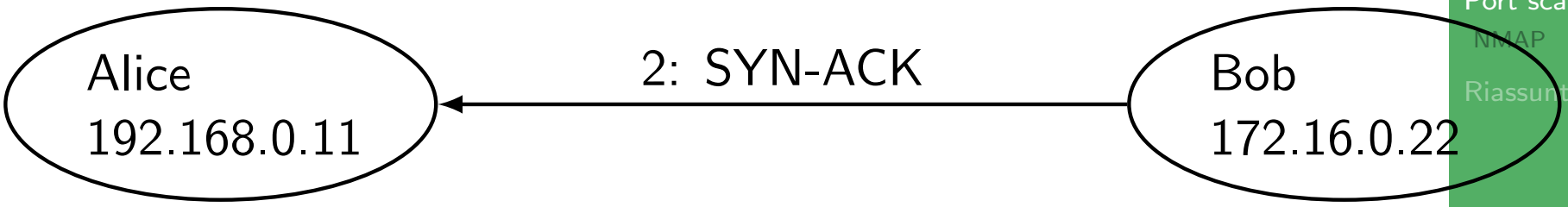
filtered La porta appare *closed* (o meglio, *not open*) per l'intervento di procedure di filtraggio (del router, firewall, ecc.)

Nel caso di TCP



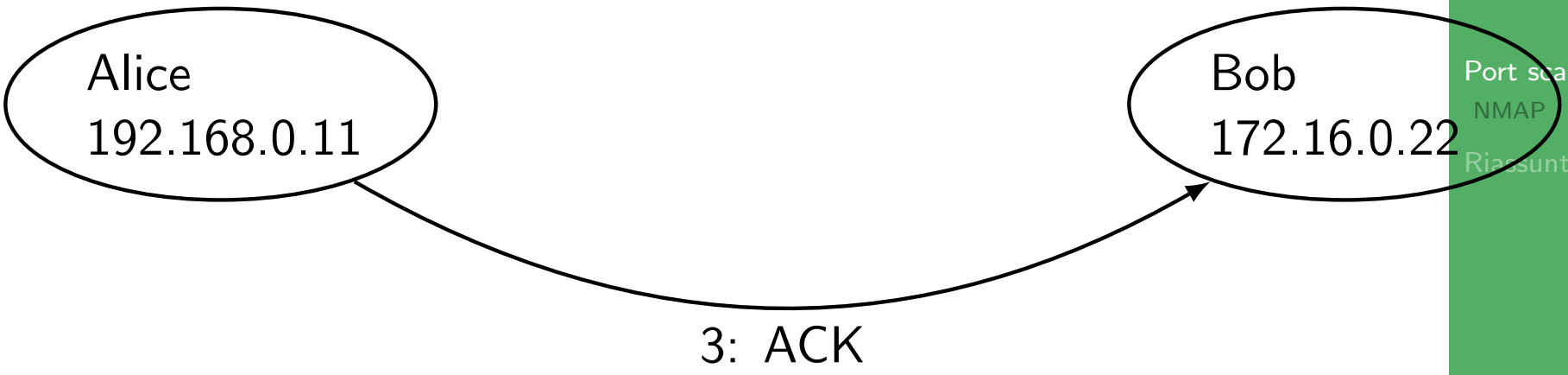
- Un SYN a porta chiusa dà un RST
- Un SYN-ACK non sollecitato dà un RST
- Un RST non sollecitato viene ignorato

Nel caso di TCP



- Un SYN a porta chiusa dà un RST
- Un SYN-ACK non sollecitato dà un RST
- Un RST non sollecitato viene ignorato

Nel caso di TCP



- Un SYN a porta chiusa dà un RST
- Un SYN-ACK non sollecitato dà un RST
- Un RST non sollecitato viene ignorato

UDP scan

Per UDP, privo di *handshake*, è un po' piú complicato

- lo stato della porta è segnalato tramite messaggi ICMP di risposta
- lento, e sostanzialmente basato su timeout
- non molto affidabile, anche perché spesso i messaggi ICMP sono filtrati (per esempio, potrebbe esserne permesso solo un numero limitato nell'unità di tempo)

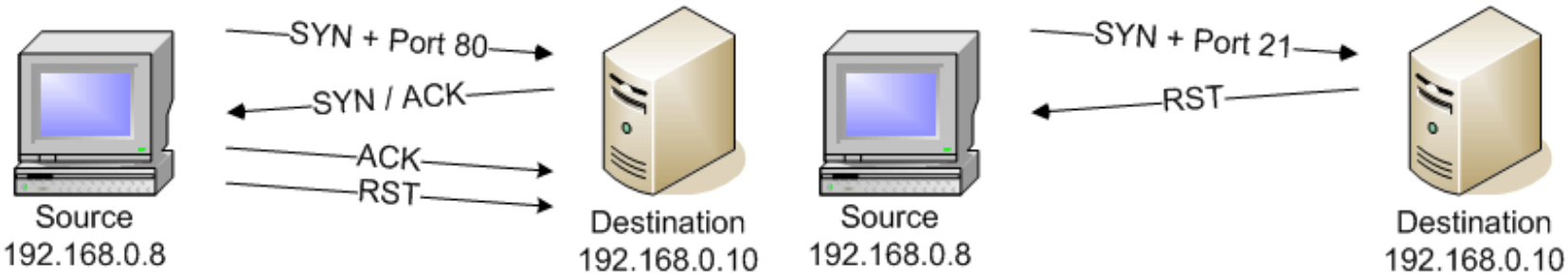
Le tecniche di base (nmap)

Nmap Scan	Command Syntax	Requires Privileged Access	Identifies TCP Ports	Identifies UDP Ports
TCP SYN Scan	`-sS`	YES	YES	NO
TCP connect() Scan	`-sT`	NO	YES	NO
FIN Scan	`-sF`	YES	YES	NO
Xmas Tree Scan	`-sX`	YES	YES	NO
Null Scan	`-sN`	YES	YES	NO
Ping Scan	`-sP`	NO	NO	NO
Version Detection	`-sV`	NO	NO	NO
UDP Scan	`-sU`	YES	NO	YES
IP Protocol Scan	`-sO`	YES	NO	NO
ACK Scan	`-sA`	YES	YES	NO
Window Scan	`-sW`	YES	YES	NO
RPC Scan	`-sR`	NO	NO	NO
List Scan	`-sL`	NO	NO	NO
Idlescan	`-sI`	YES	YES	NO
FTP Bounce Attack	`-b`	NO	YES	NO

Connect

La modalità piú semplice è tentare una connessione (`connect()`)

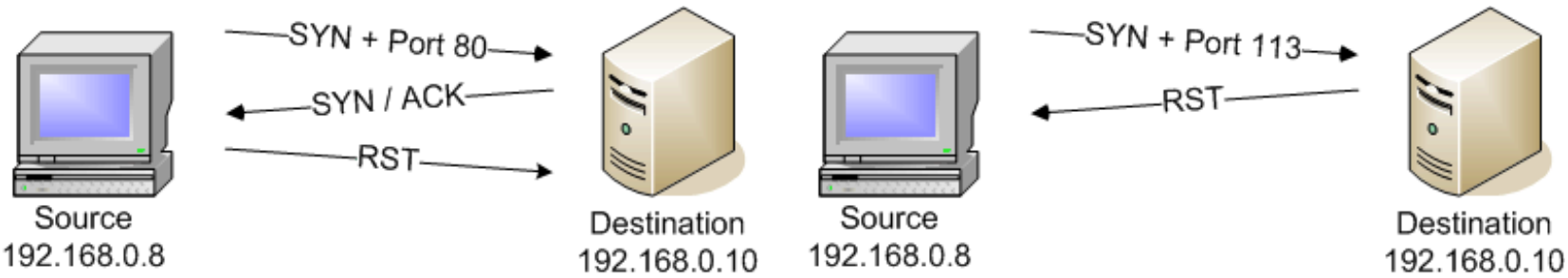
- non richiede privilegi particolari
- molto spesso l'evento viene registrato (e se la connessione avviene con lo stack standard il numero IP è quello reale)



SYN scan (half open)

Si risponde al SYN-ACK con un RST.

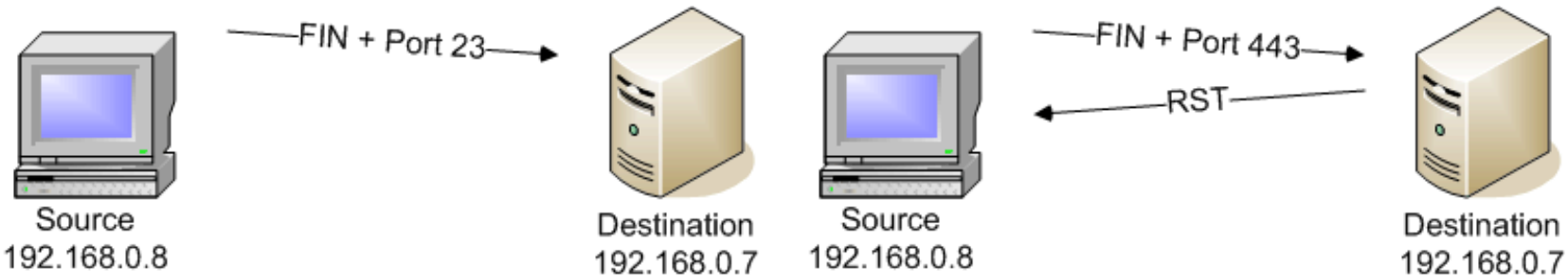
- È il metodo piú usato: veloce ed efficace
- Richiede i privilegi di root (non si può usare lo stack TCP standard)
- Piú difficile da "loggare"



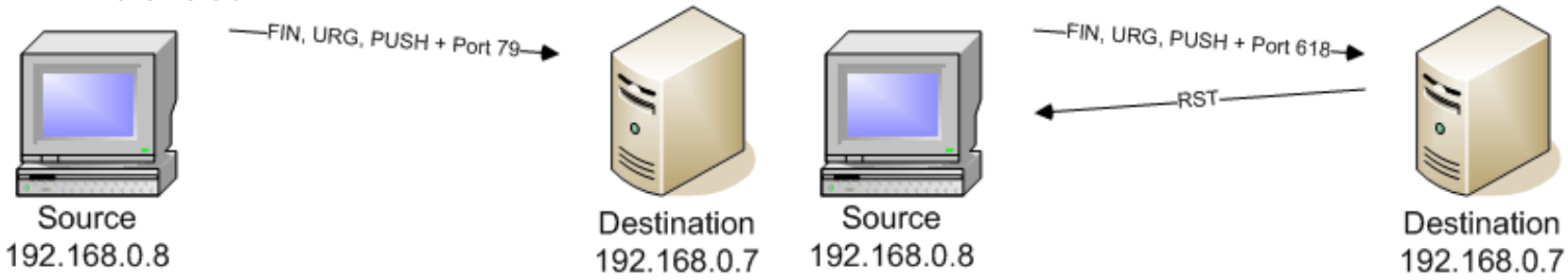
TCP NULL, FIN, Xmas scan

In questo caso si usano i flag in modo "creativo": invece di usare SYN si usano tutti gli altri in varie combinazioni; una porta chiusa risponde con un RST, una aperta invece li scarta (aspetta solo i SYN).

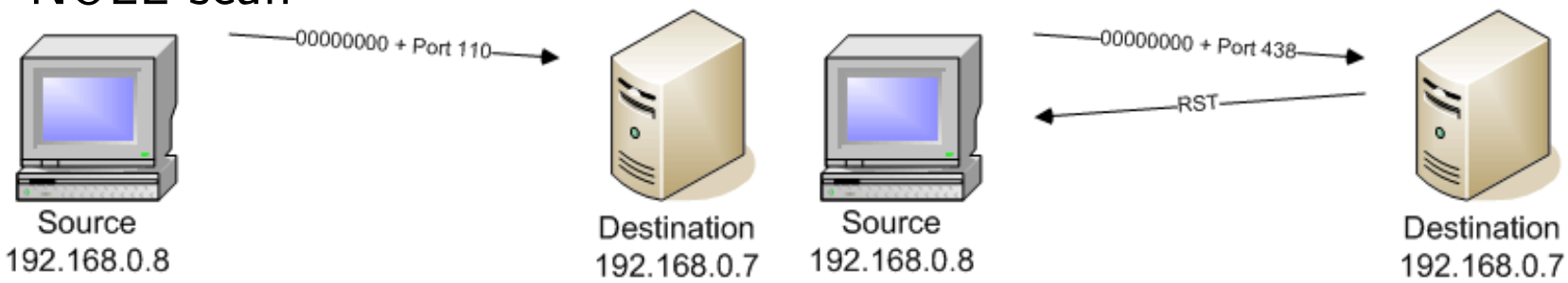
- Analoghi al SYN
- richiedono i privilegi di root
- ma ancora meno probabile una registrazione dell'evento
- lo stack destinazione, se non RFC 793 compliant, potrebbe agire in modo anomalo facendo apparire tutto chiuso



Xmas scan



NULL scan



Maimon scan: FIN-ACK; È anche possibile provare differenti combinazioni dei flag

ACK scan, Window scan

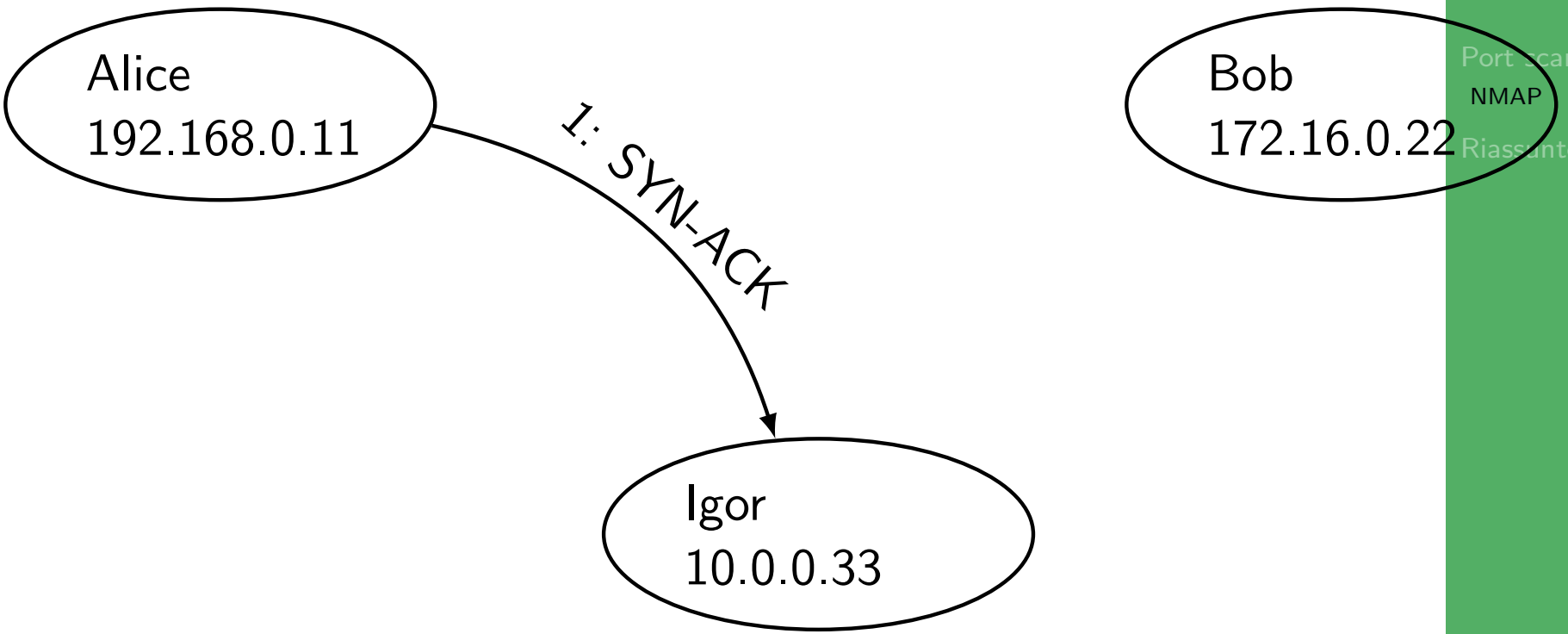
- Serve a determinare se c'è filtraggio.
- Si manda un ACK: se non c'è filtraggio open e closed danno un RST
- se non c'è risposta o si riceve un messaggio ICMP: filtered
- la versione Window sfrutta la window size del RST ricevuto per distinguere fra open e closed (diversa in alcune implementazioni)

Idle scan

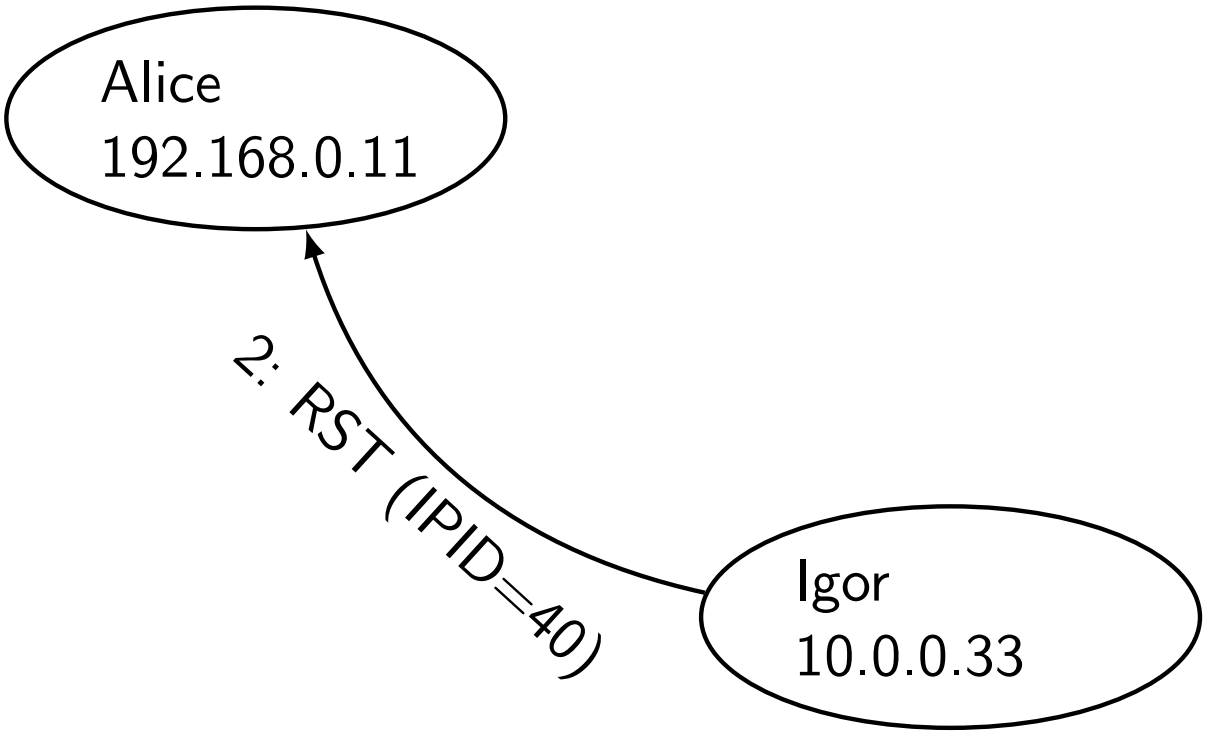
Lo scan viene compiuto da un nodo inconsapevole sfruttando il meccanismo di generazione degli ID dei pacchetti IP (IPID), che generalmente è sequenziale.

- Ogni possibile registrazione dell'evento conterrà l'IP della macchina "prestanome" (non è spoofing perché il nodo esiste e ha operato nel modo registrato)
- Il nodo deve essere "idle", cioè non produrre traffico di rete suo durante lo scan
- Il nodo deve implementare lo stack IP in maniera opportuna, incrementando gli IPID

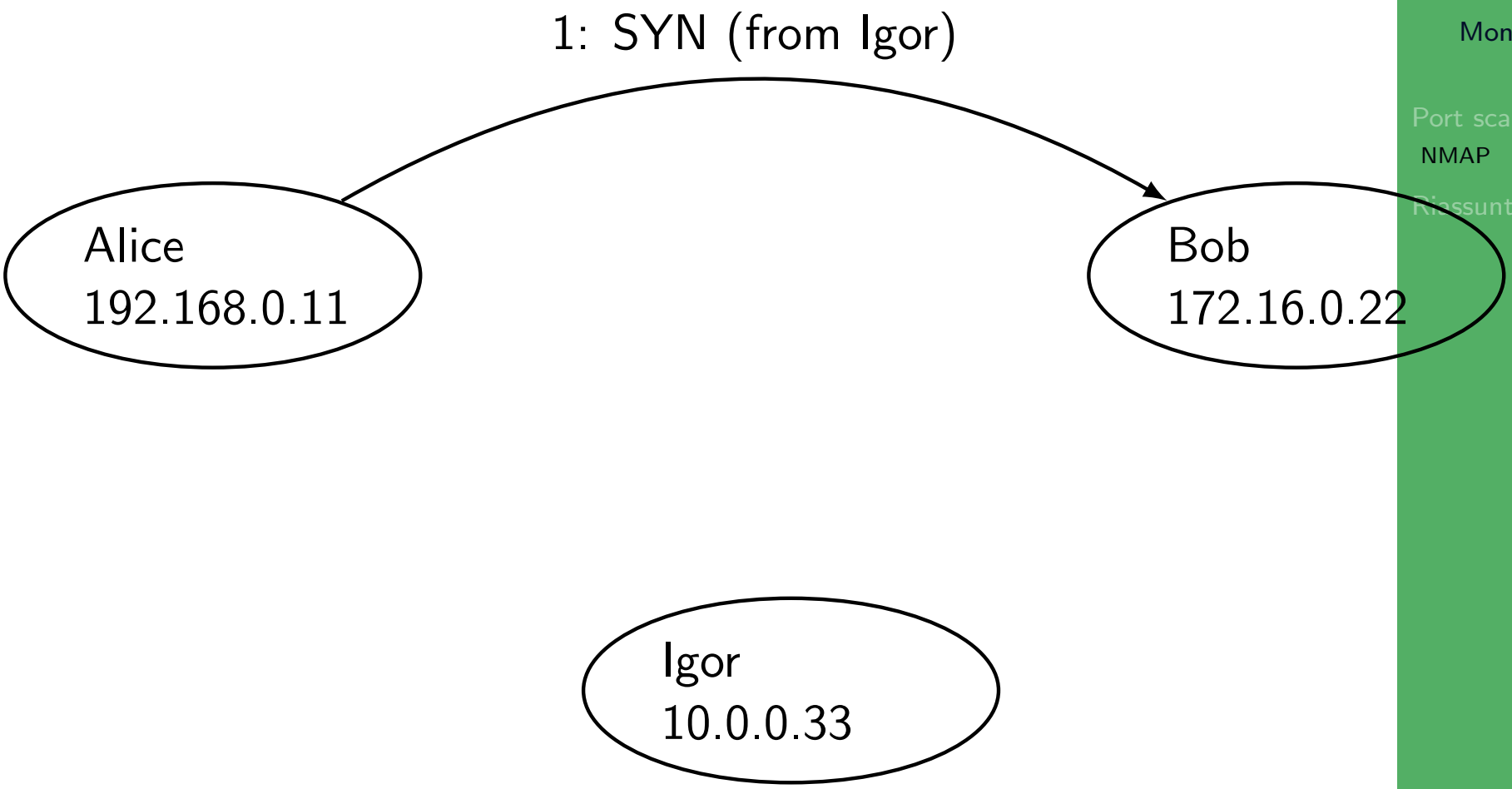
Idle scan con porta aperta



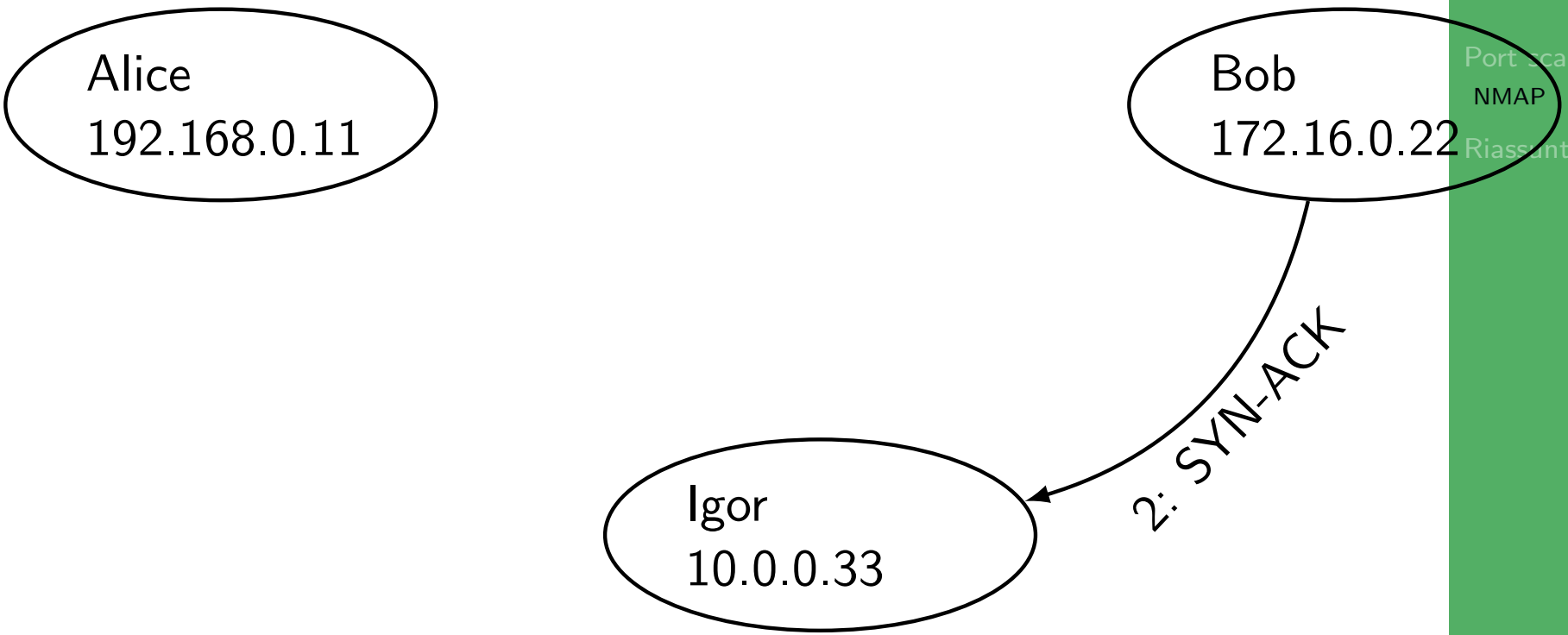
Idle scan con porta aperta



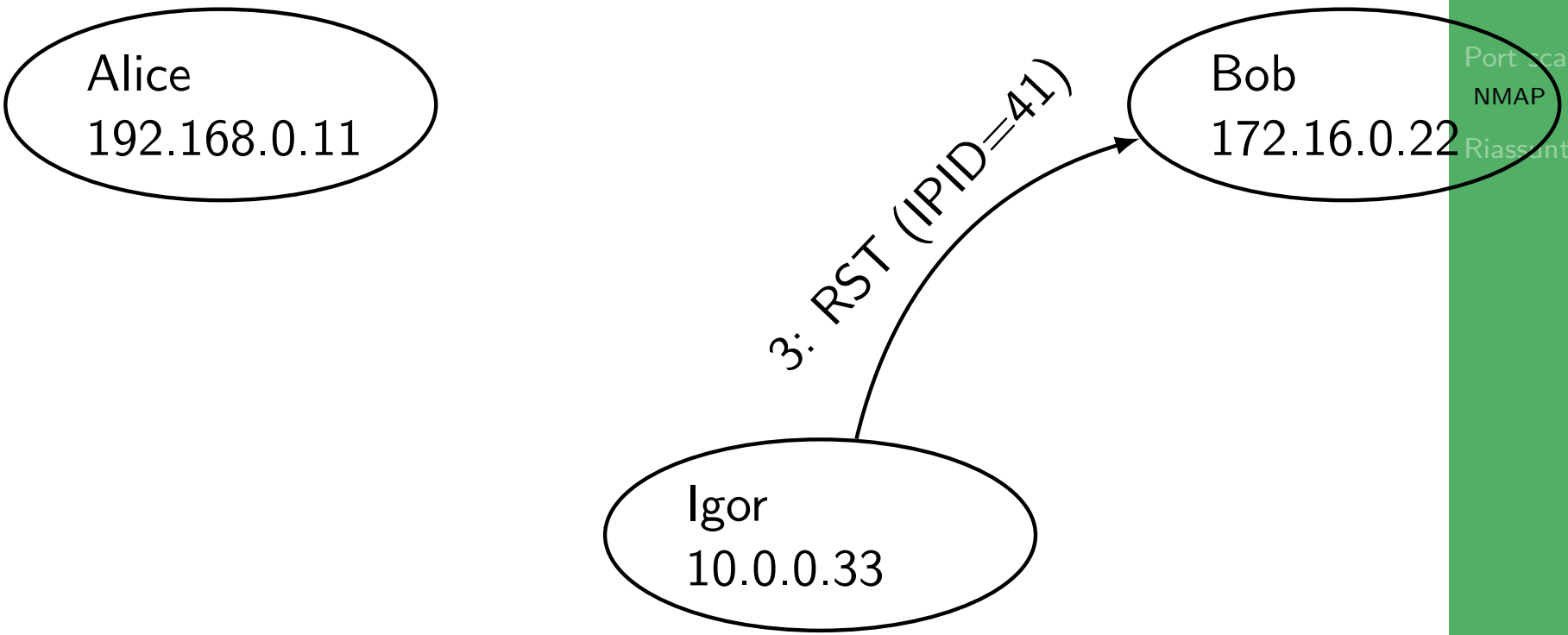
Idle scan con porta aperta



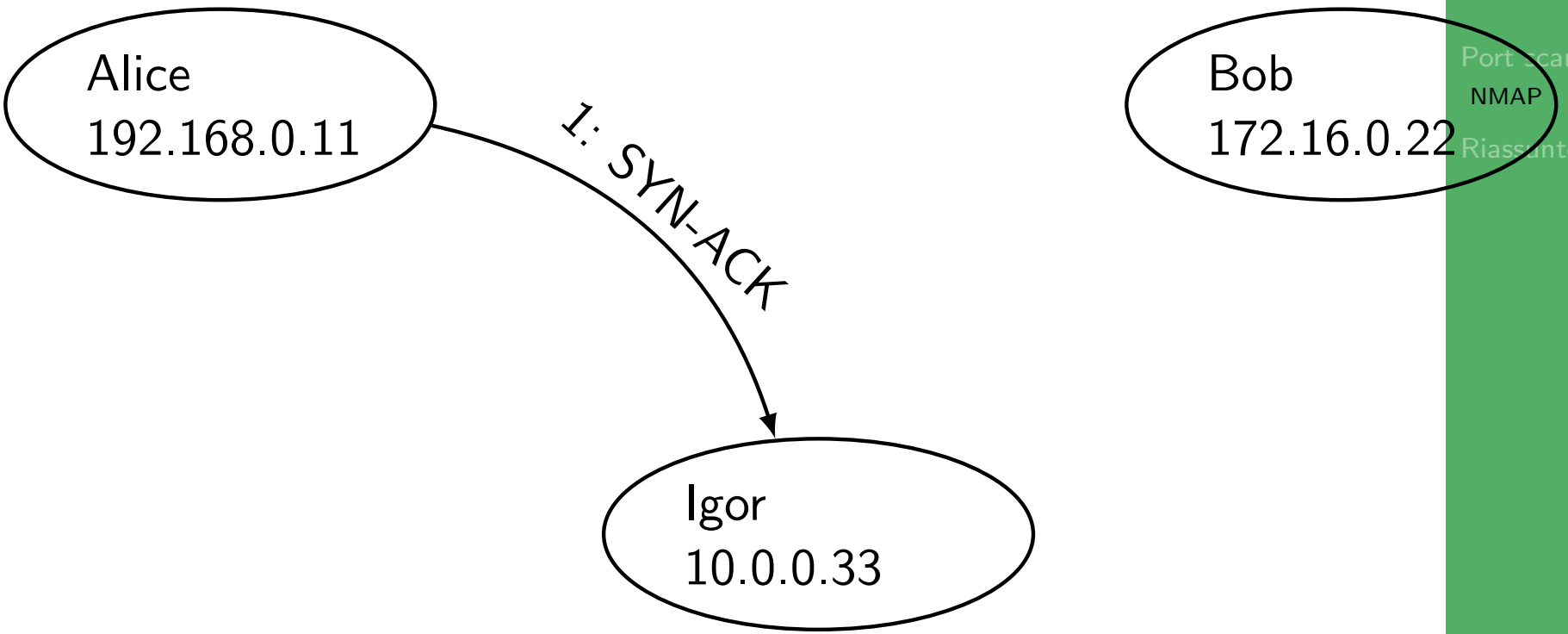
Idle scan con porta aperta



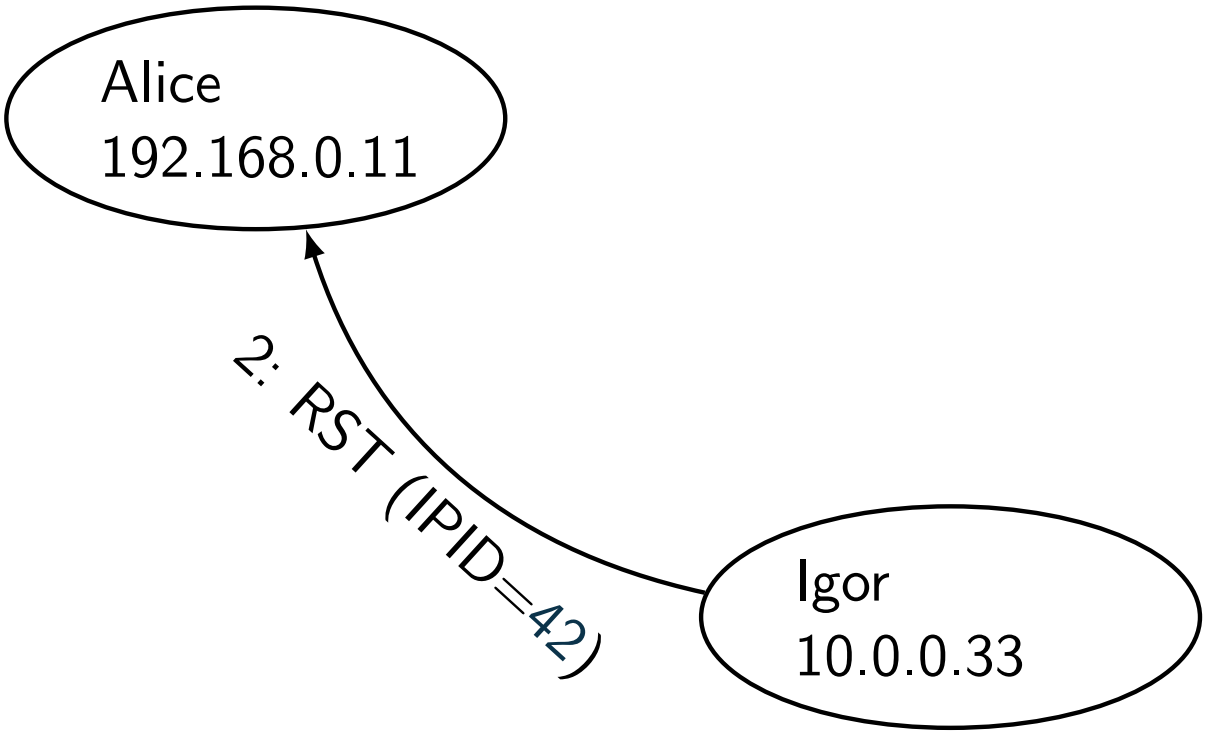
Idle scan con porta aperta



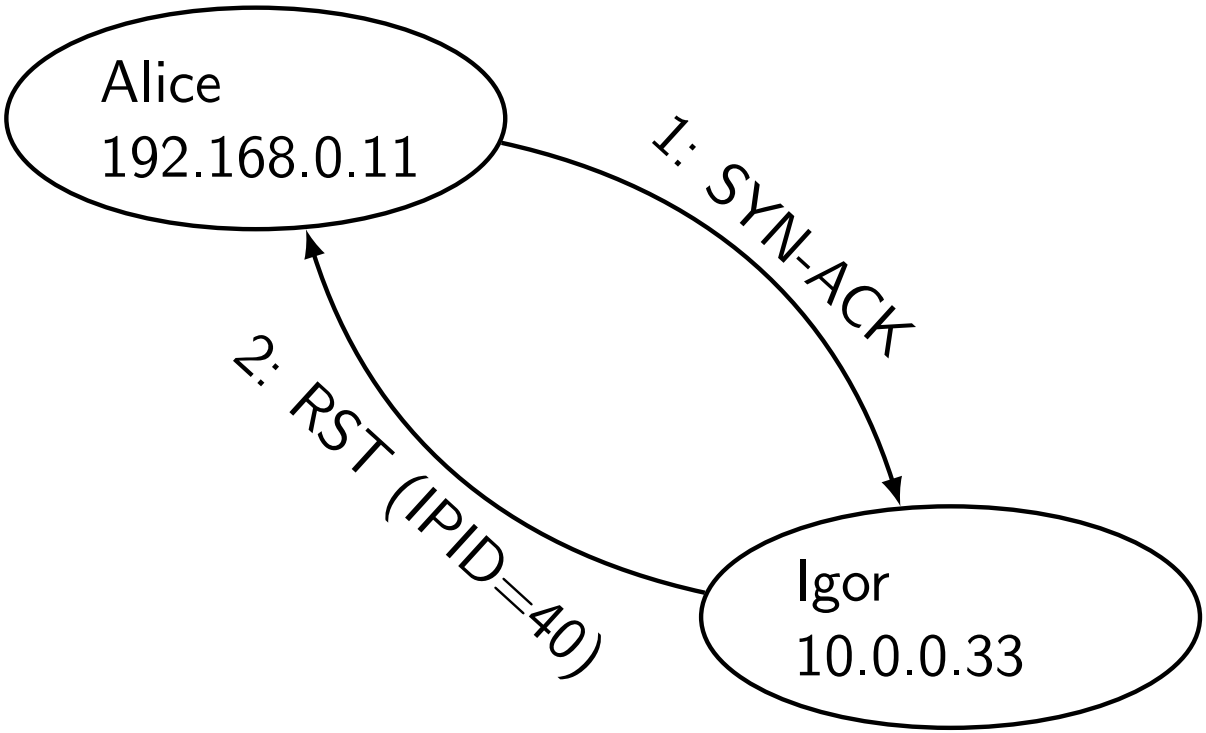
Idle scan con porta aperta



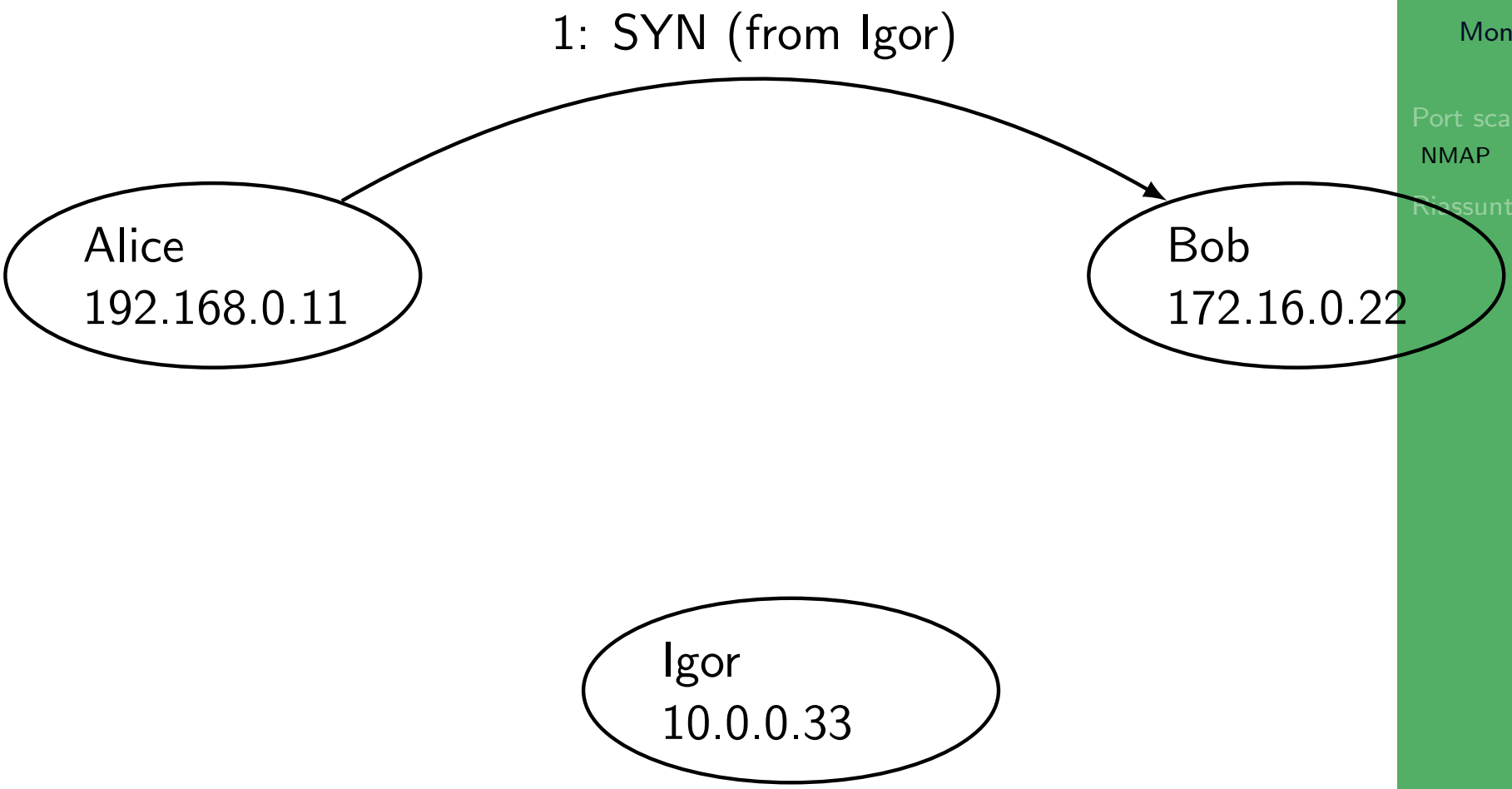
Idle scan con porta aperta



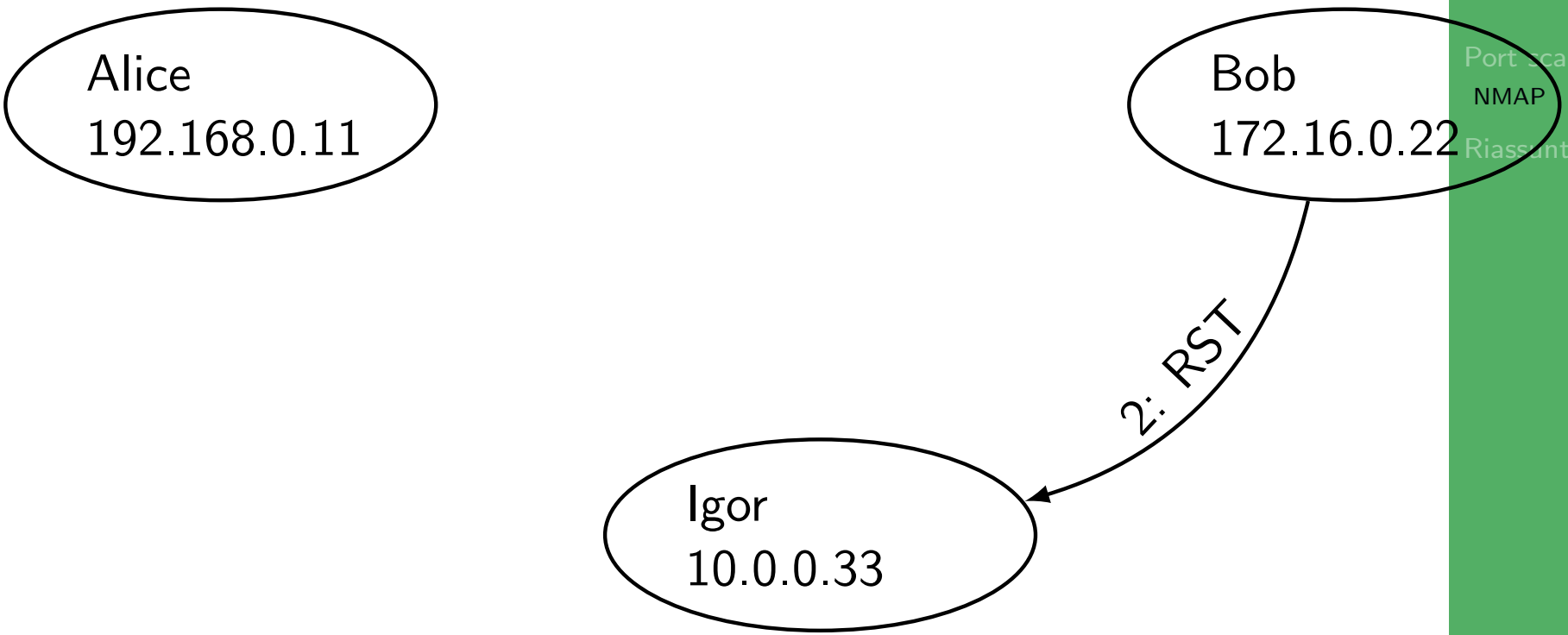
Idle scan con porta chiusa



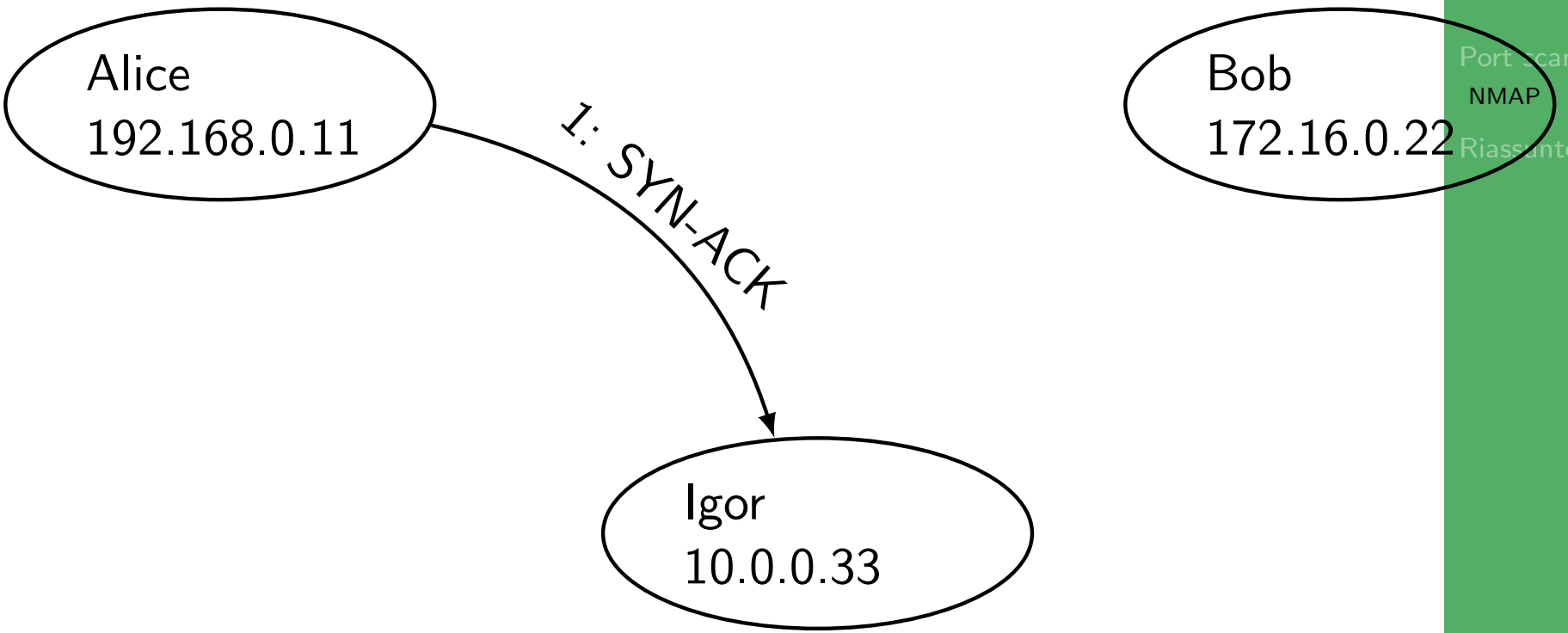
Idle scan con porta chiusa



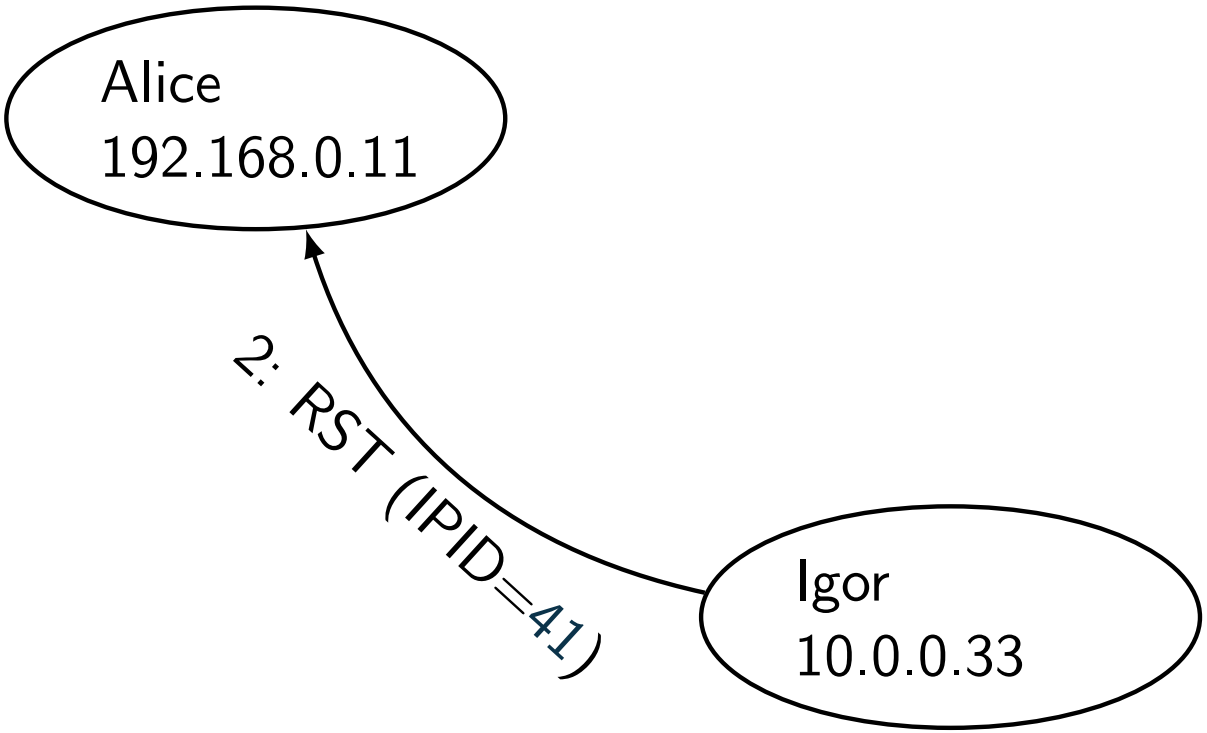
Idle scan con porta chiusa



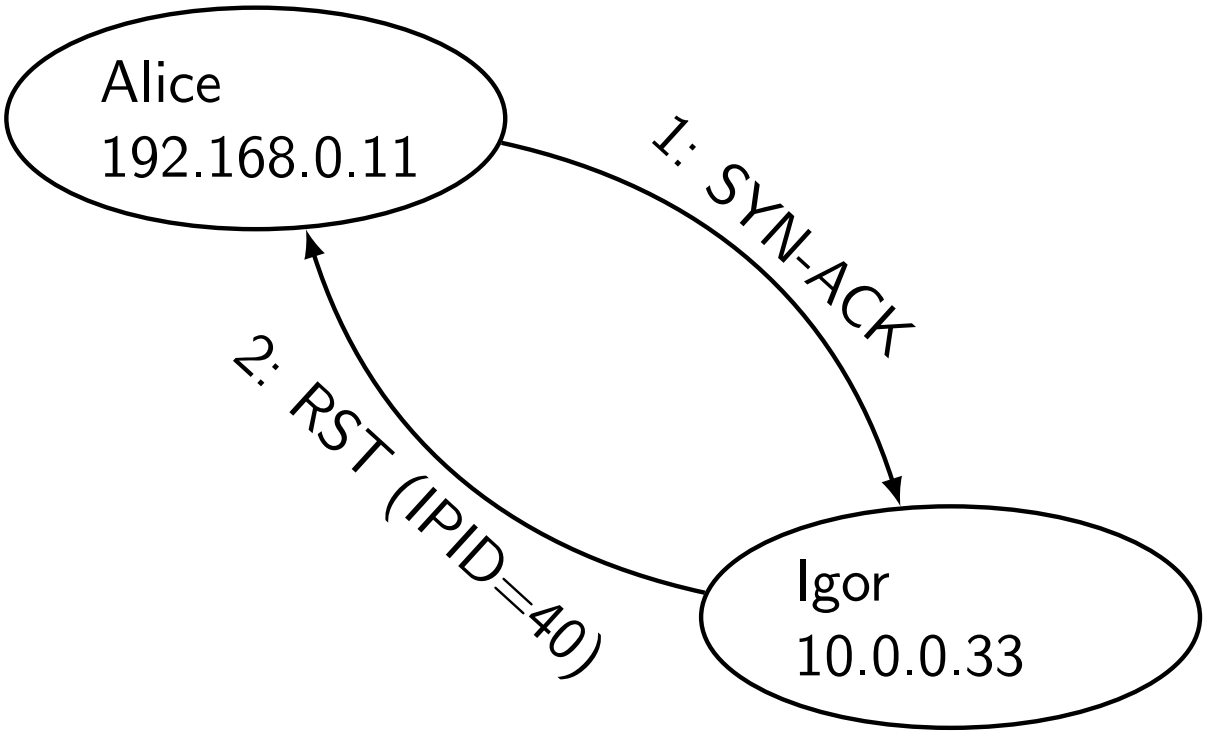
Idle scan con porta chiusa



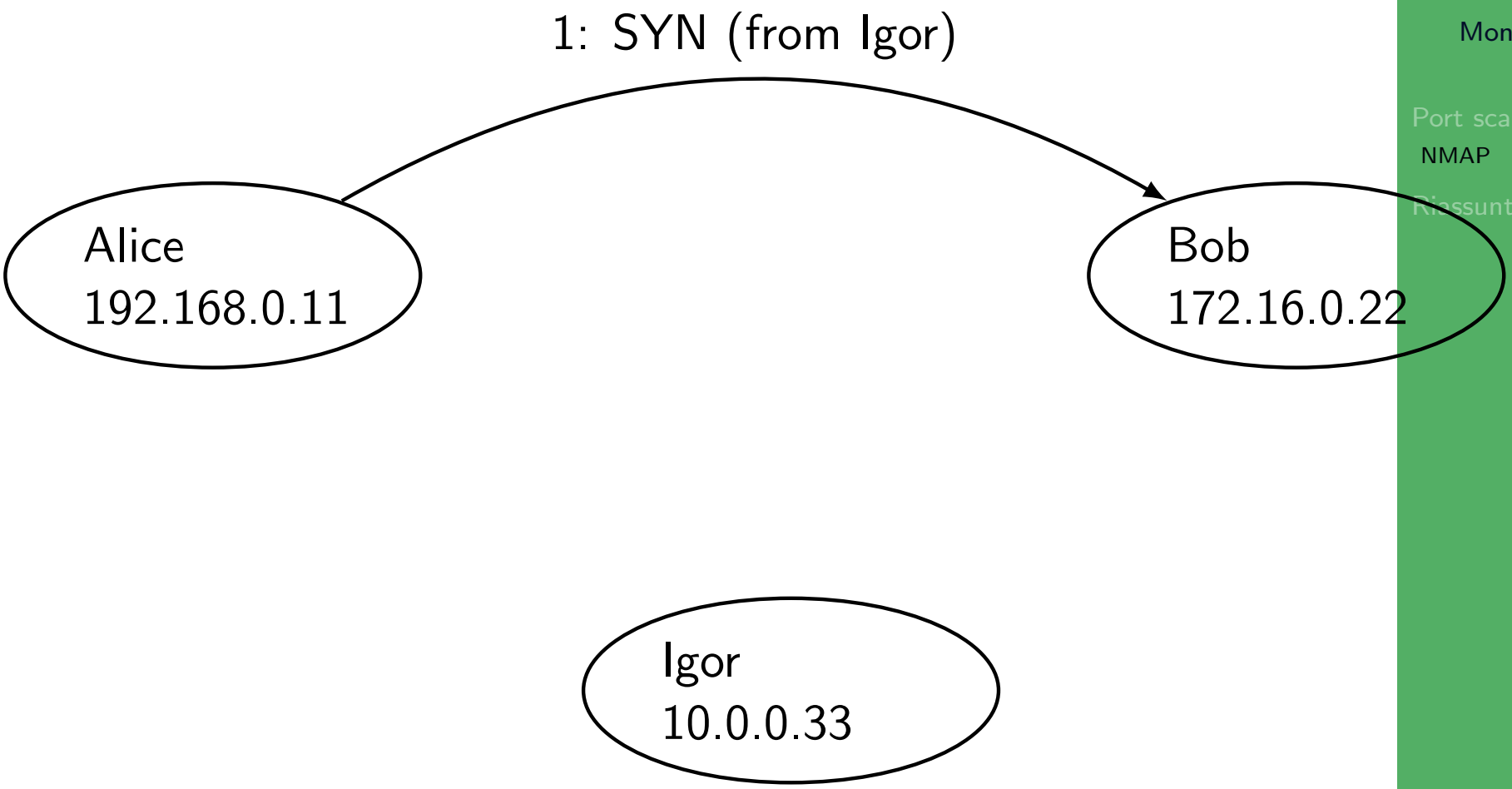
Idle scan con porta chiusa



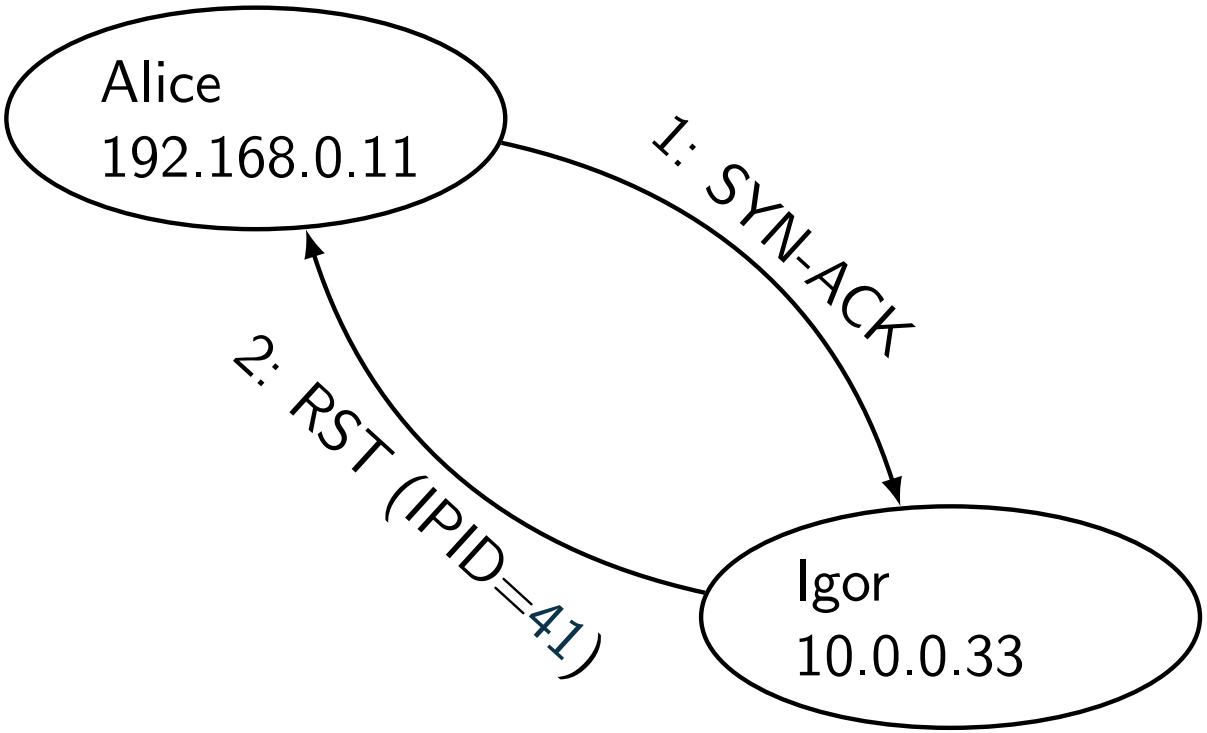
Idle scan con porta filtrata



Idle scan con porta filtrata



Idle scan con porta filtrata



Riassunto dei punti fondamentali

- La conoscenza dei canali di comunicazione disponibili è fondamentale per attaccanti e difensori
- Sono note diverse tecniche per rilevare se una porta è aperta
  - Semplice connessione
  - Pacchetti creati ad hoc
  - Idle scan