Telecommunications Infrastructure Security

# Toward the HLR, attacking the SS7 & SIGTRAN applications.

## one step further and mapping the phone system.

Philippe Langlois, P1 Security Inc.
phil@p1sec.com

# SS7 Basics

Introduction to SS7 in the Phone System

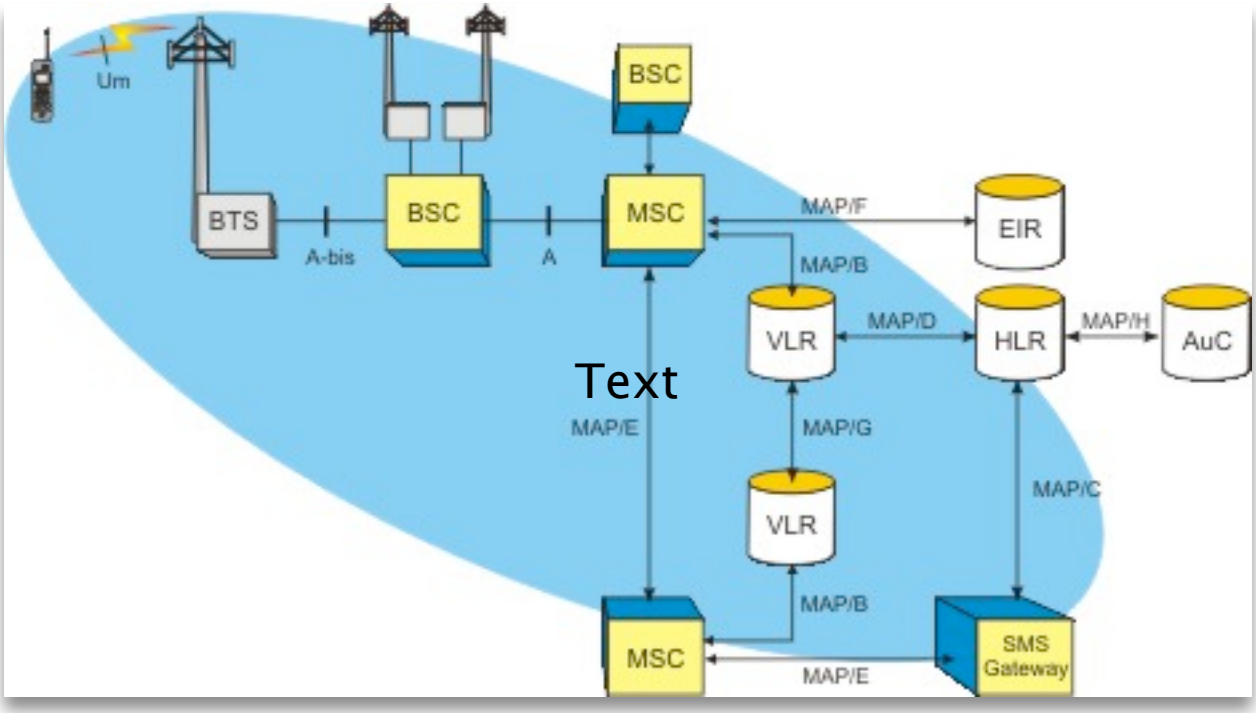
P1 Security Inc, http://www.p1security.com

# Why do we have SS7?

Steve Jobs and Steve Wozniak in 1975 with a bluebox

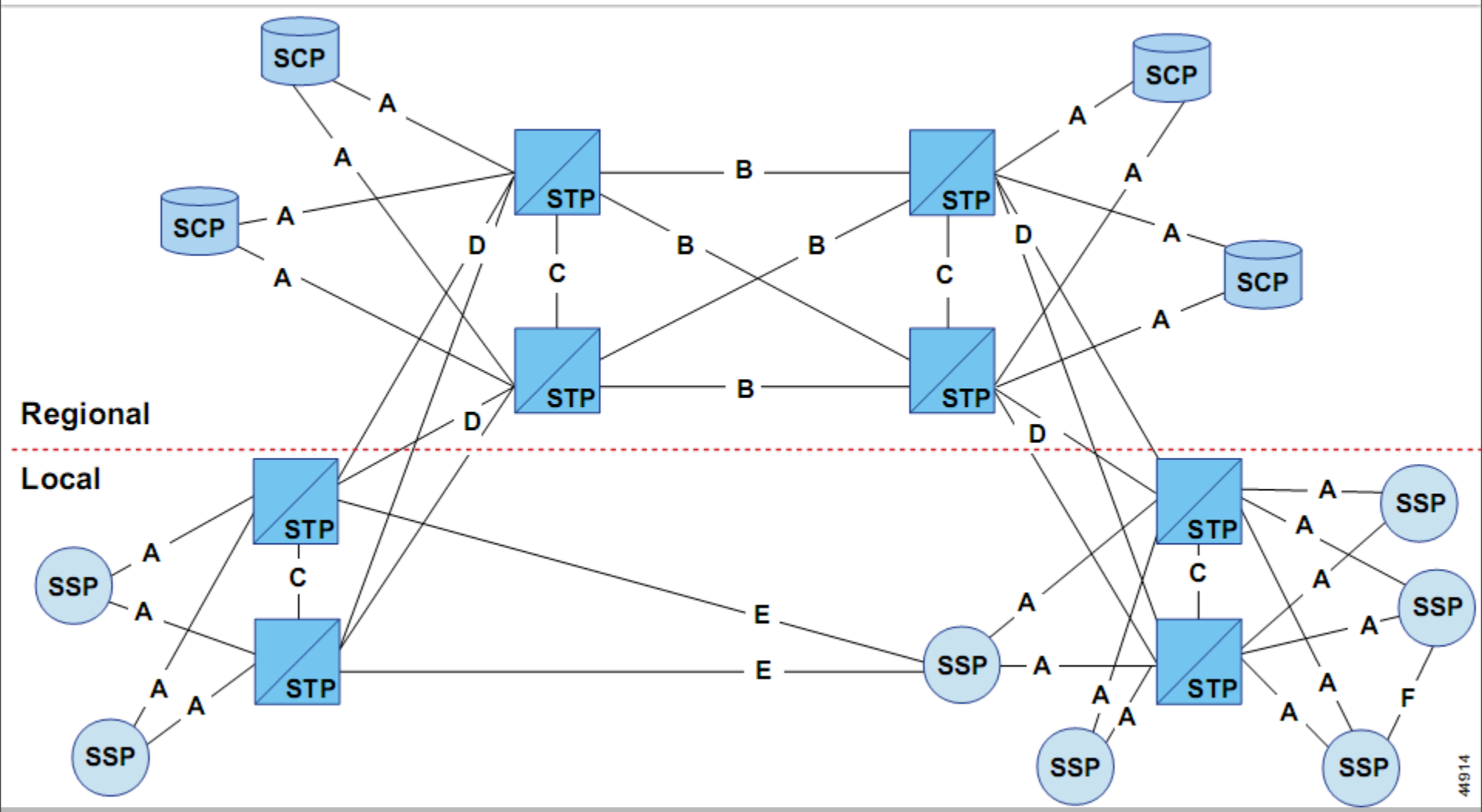
- Thanks to hackers!
- CCITT#5 in-band signalling sends control messages over the speech channel, allowing trunks to be controlled
- Seize trunk (2600) / KP1 or KP2 / destination / ST
- Started in mid-60's, became popular after Esquire 1971
- Sounds produced by whistles, electronics dialers, computer programs, recorded tones

# SS7 basic architecture




- **HLR/VLR** Home Location Register, Visitor Location Register
- **AuC** : Authentication Center (within HLR)
- **EIR** : Equipment Identity Register
- **MSC** : Mobile Switching Center
- **STP** : Signaling Transfer Point (i.e. Router)

# SS7 network

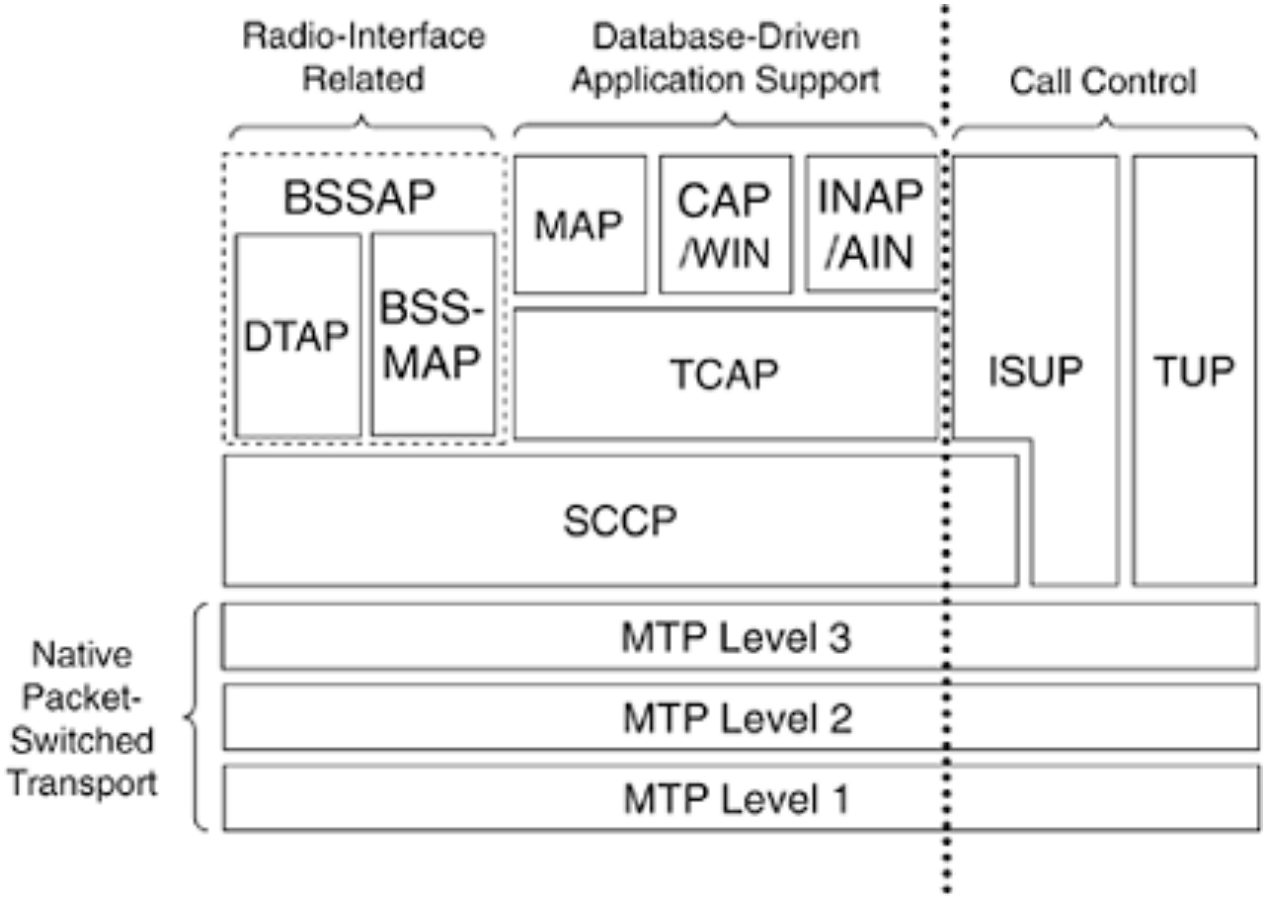

SCP
SCP
SCP
SCP
STP
STP
STP
STP
STP
STP
STP
STP
SSP
SSP
SSP
SSP
SSP
SSP
A
B
C
D
E
F
Regional
Local
44914

# Main focus: reliability

To meet the stringent reliability requirements of public telecommunications networks, a number of safeguards are built into the SS7 protocol:

- STPs and SCPs are normally provisioned in **mated pairs**. On the failure of individual components, this duplication allows signaling traffic to be automatically diverted to an alternate resource, minimizing the impact on service.
- Signaling links are provisioned with some level of **redundancy**. Signaling traffic is automatically diverted to alternate links in the case of link failures.
- The SS7 protocol has built-in **error recovery** mechanisms to ensure reliable transfer of signaling messages in the event of a network failure.
- Management messages (Link Status Signal Units) are constantly sent over the links to **monitor** its status.

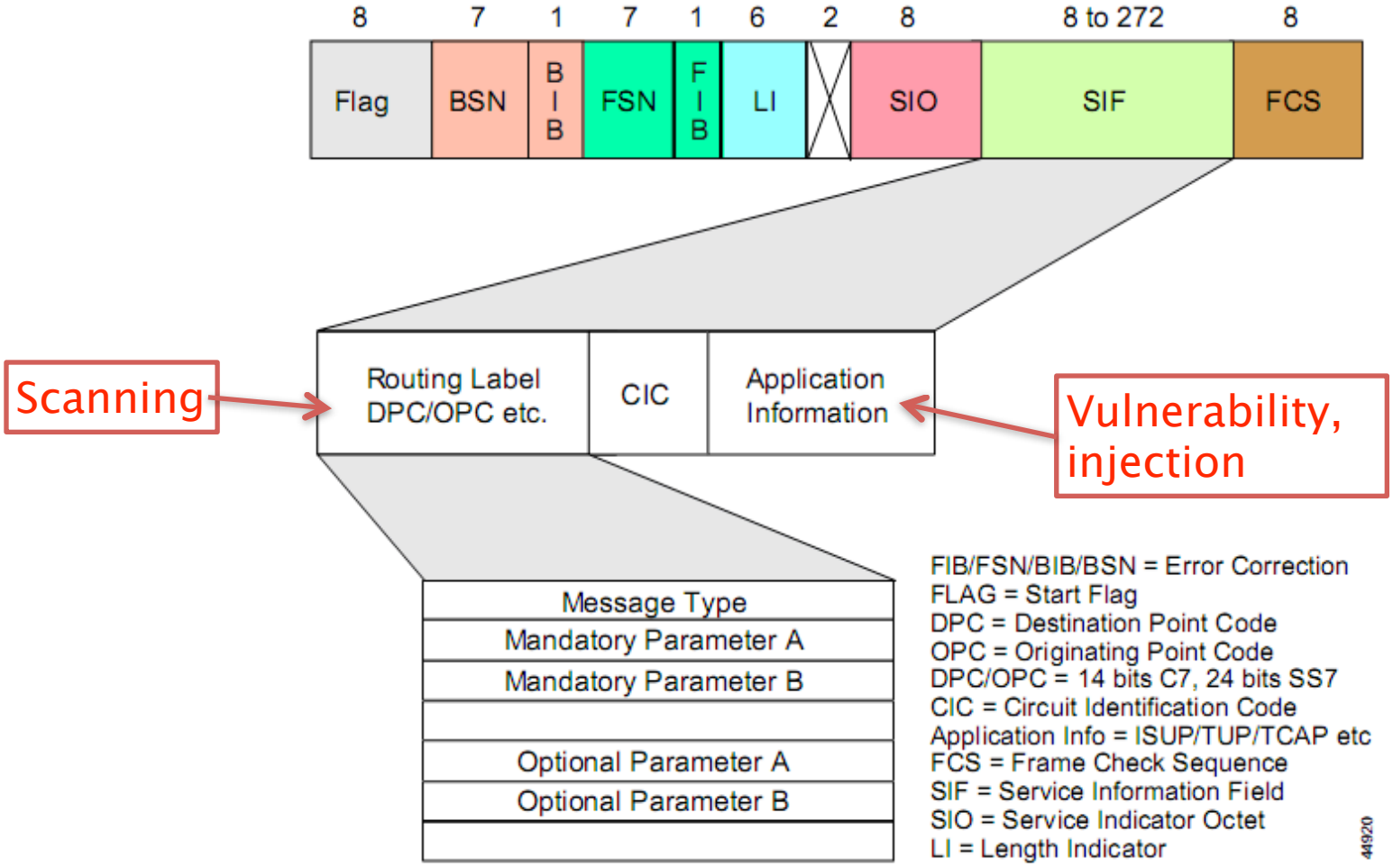
# Under the hood: SS7 stack



# Important SS7 protocols

- **MTP** (Message Transfer Part) Layers 1-3: lower level functionality at the Physical, Data Link and Network Level. They serve as a signaling transfer point, and support multiple congestion priority, message discrimination, distribution and routing.
- **ISUP** (Integrated Services Digital Network User Part): network side protocol for the signaling functions required to support voice, data, text and video services in ISDN. ISUP supports the call control function for the control of analog or digital circuit switched network connections carrying voice or data traffic.
- **SCCP** (Signaling Control Connection Part): supports higher protocol layers such as TCAP with an array of data transfer services including connection-less and connection oriented services. SCCP supports global title translation (routing based on directory number or application title rather than point codes), and ensures reliable data transfer independent of the underlying hardware.
- **TCAP** (Transaction Capabilities Application Part): provides the signaling function for communication with network databases. TCAP provides non-circuit transaction based information exchange between network entities.
- **MAP** (Mobile Application Part): provides inter-system connectivity between wireless systems, and was specifically developed as part of the GSM standard.
- **INAP** (Intelligent Network Application Part): runs on top of TCAP and provides high-level services interacting with SSP, SCP and SDP in an SS7 network.

# MSU: Message Signal Unit



# Entry points in an SS7

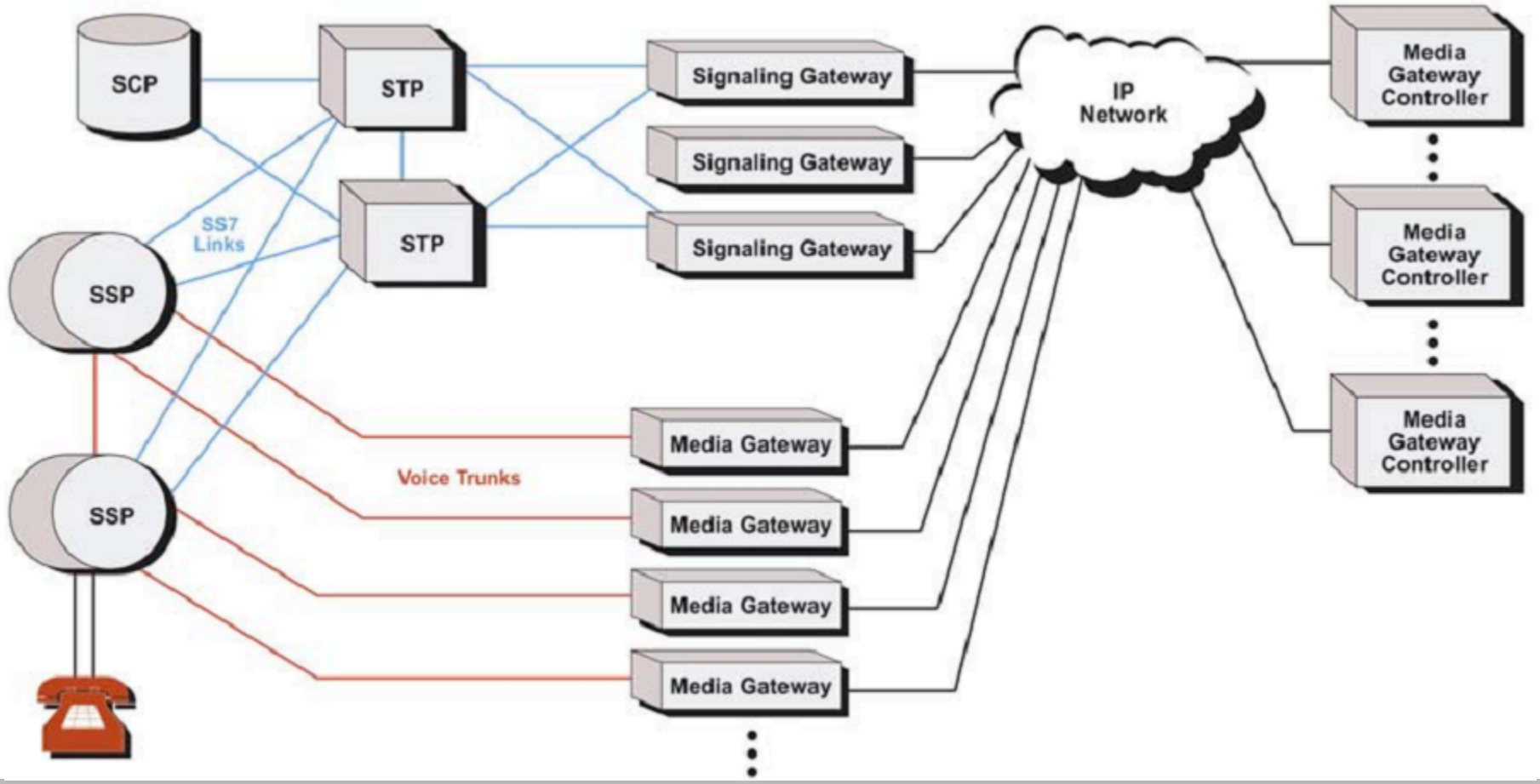
- Peer relationships between operators
- STP connectivity
- SIGTRAN protocols
- VAS systems e.g. SMSC, IN
- Signalling Gateways, MGW
- SS7 Service providers (GRX, IPX)
- GTT translation
- ISDN terminals
- GSM phones
- LIG (pentest & message relaying madness)
- 3G Femtocell
- SIP encapsulation

# SS7 and IP: the SIGTRAN evolution and problems

Basics of IP telephony
SIGTRAN protocols & SCTP scanning

P1 Security Inc, http://www.p1security.com

# SIGTRAN network




P1 Security Inc, http://www.p1security.com

# SIGTRAN evolution

- The **SIGTRAN protocols** specify the means by which SS7 messages can be reliably transported over IP networks (with **SCTP**).
- The architecture identifies two components: a **common transport protocol** for the SS7 protocol layer being carried and an **adaptation module** to emulate lower layers of the protocol. For example:
  - If the native protocol is MTP (Message Transport Layer) Level 3, the SIGTRAN protocols provide the equivalent functionality of MTP Level 2.
  - If the native protocol is ISUP or SCCP, the SIGTRAN protocols provide the same functionality as MTP Levels 2 and 3.
  - If the native protocol is TCAP, the SIGTRAN protocols provide the functionality of SCCP (connectionless classes) and MTP Levels 2 and 3.

# SCTP Specs & Advantages

- RFC4960
  - SCTP: Stream Control Transmission Protocol
- Advantages
  - Multi-homing
  - DoS resilient (4-way handshake, cookie)
  - Multi-stream
  - Reliable datagram mode
  - Some of TCP & UDP, improved

# SCTP association

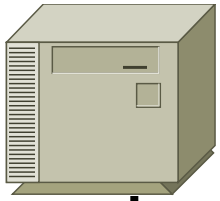
Client

`socket(), connect()`

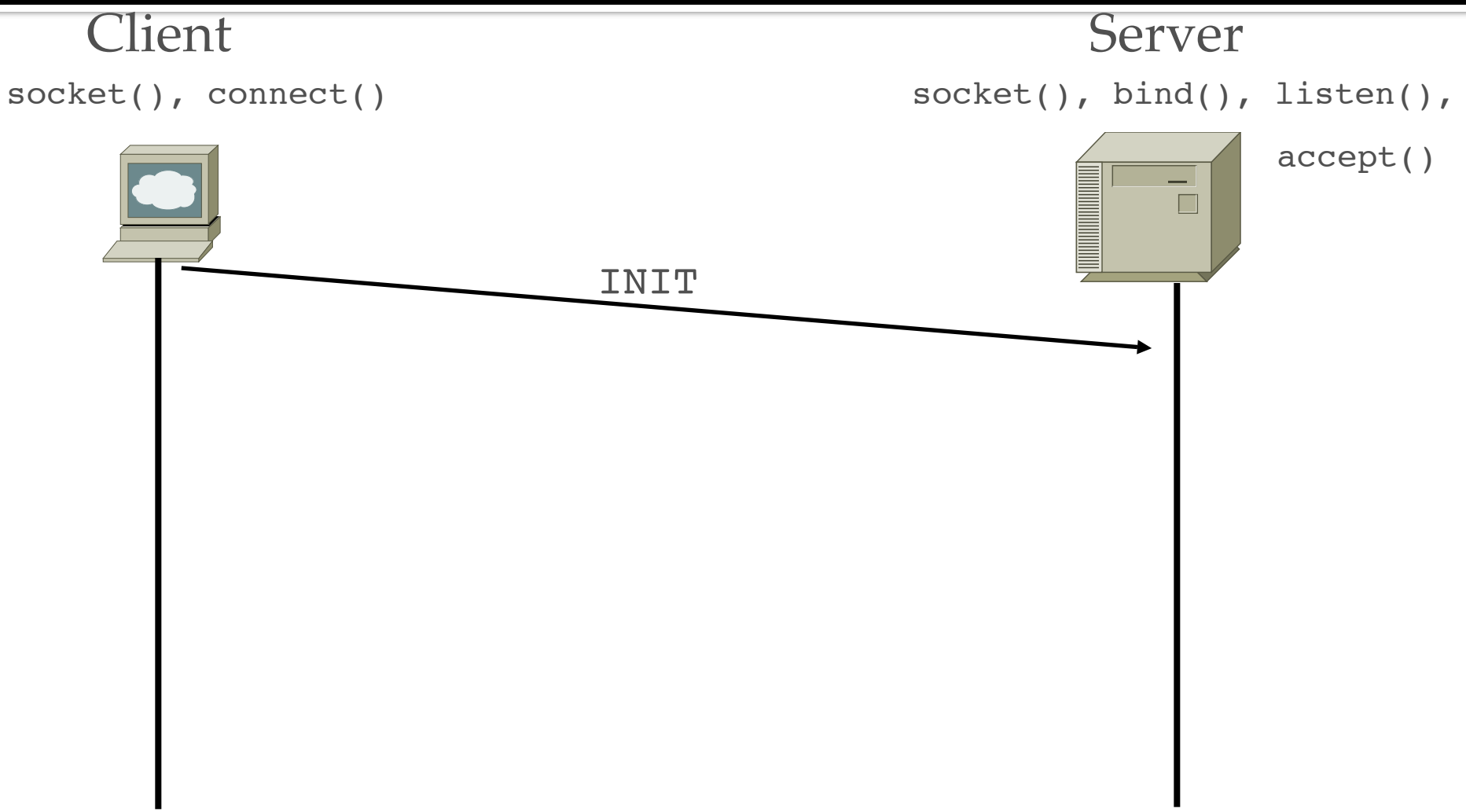
Server

`socket(), bind(), listen(),`

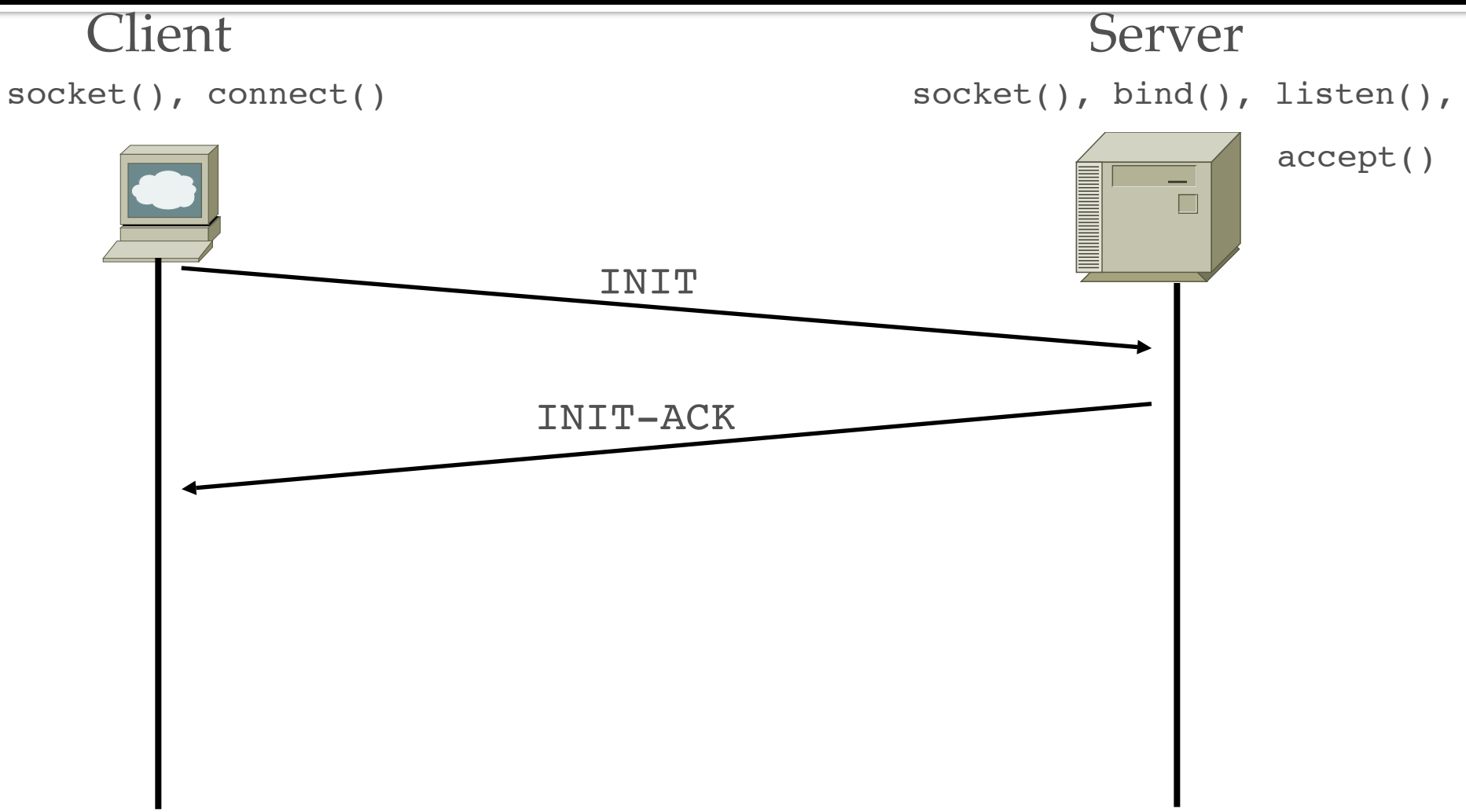
`accept()`



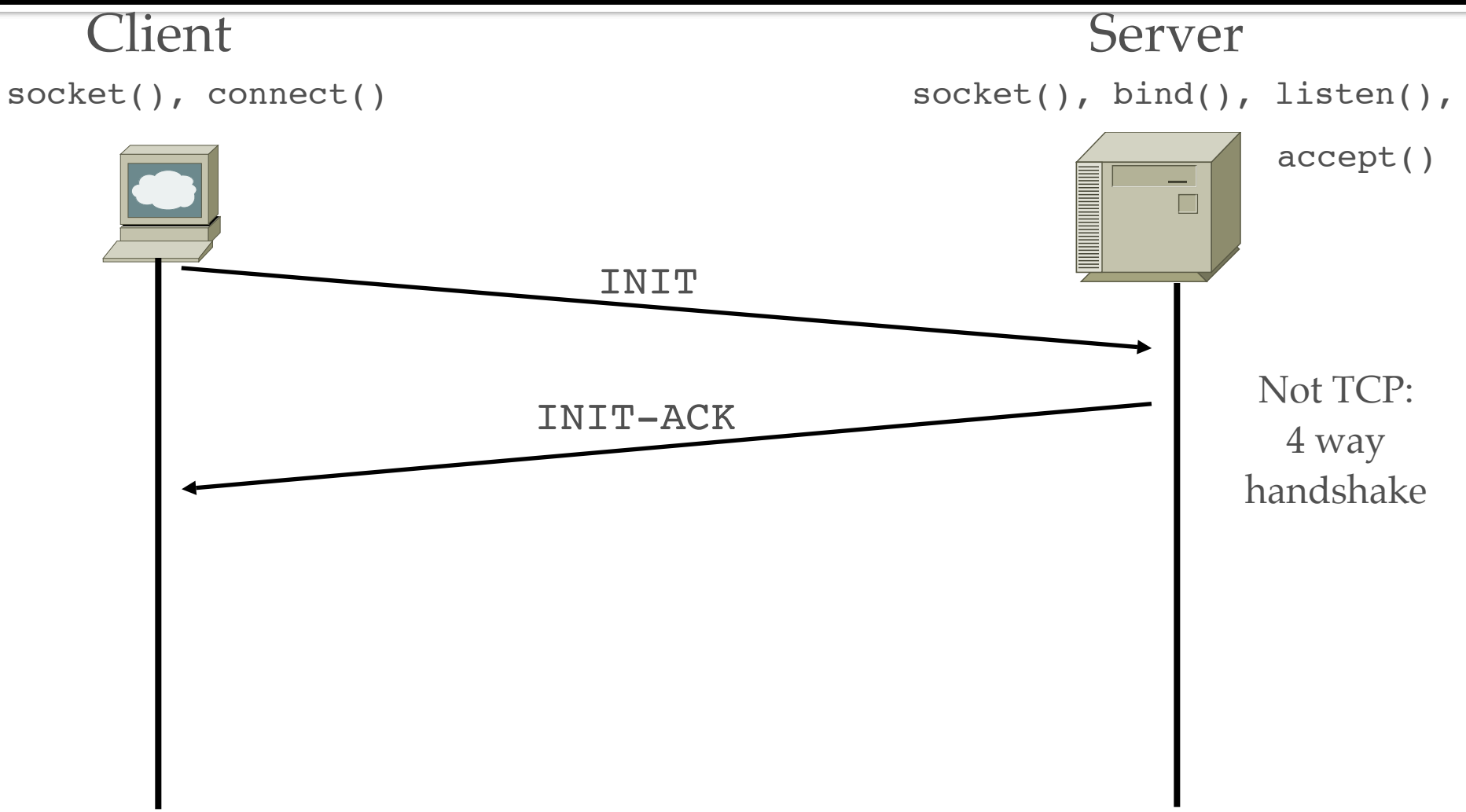
# SCTP association



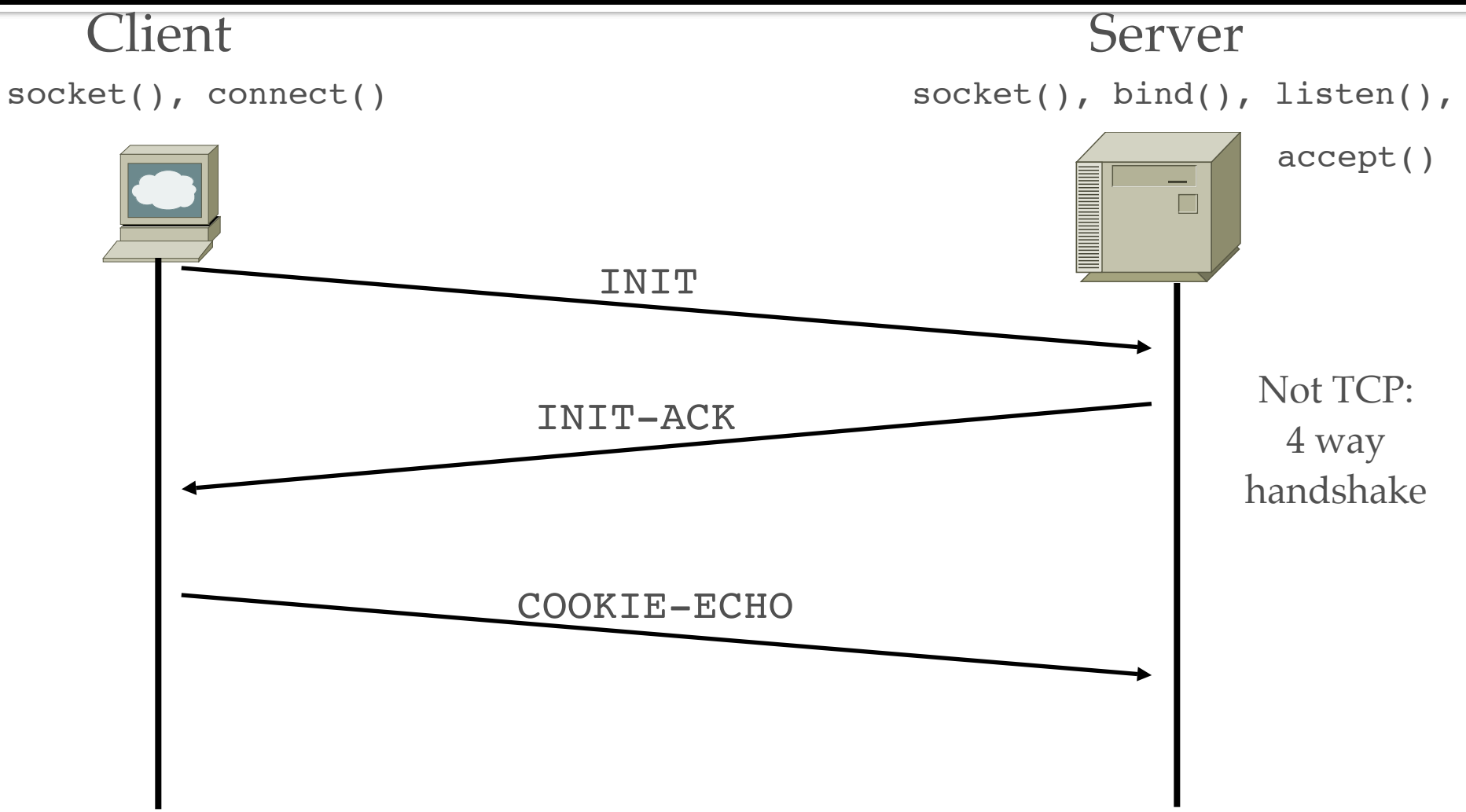
# SCTP association



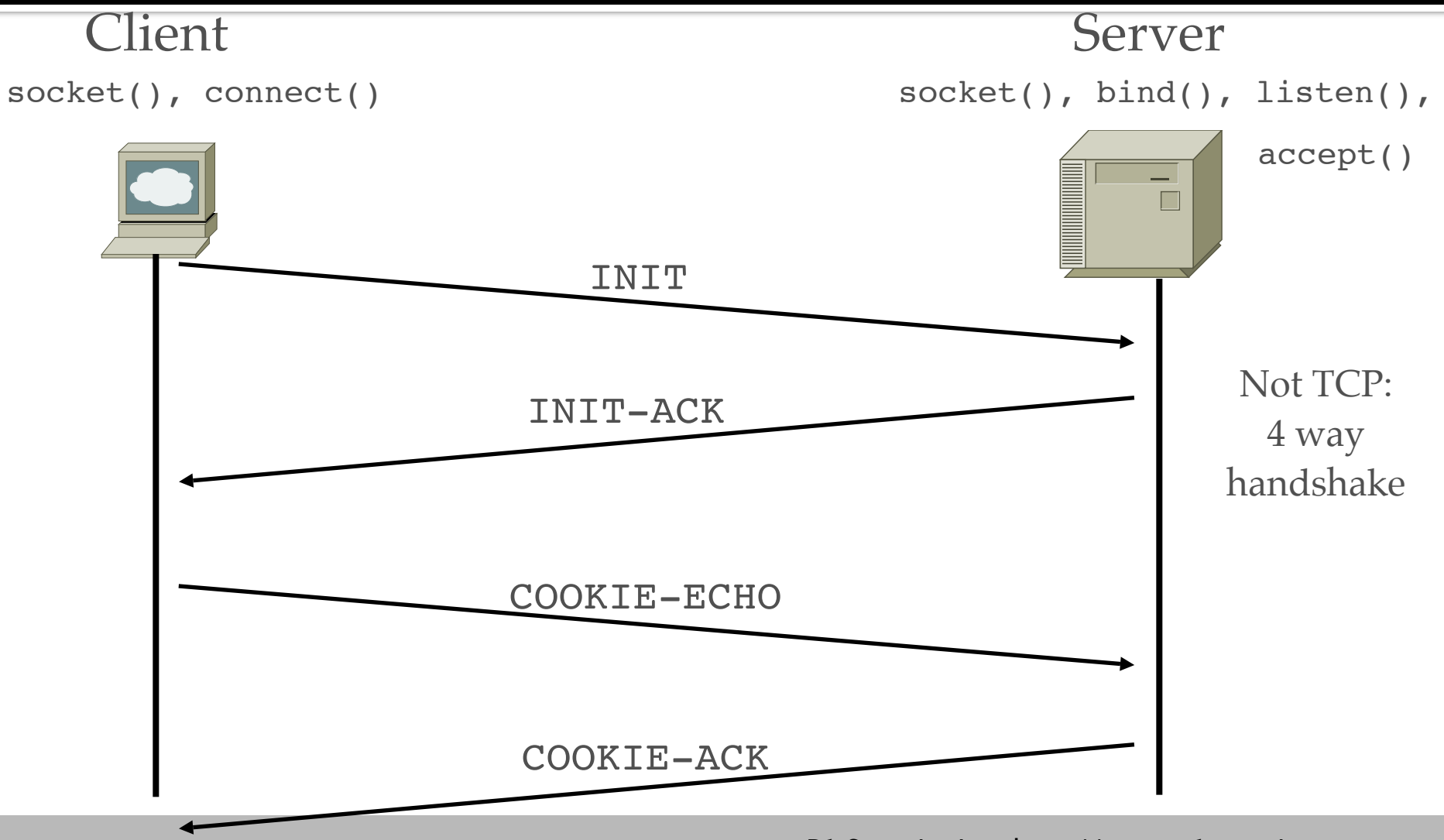
# SCTP association



# SCTP association

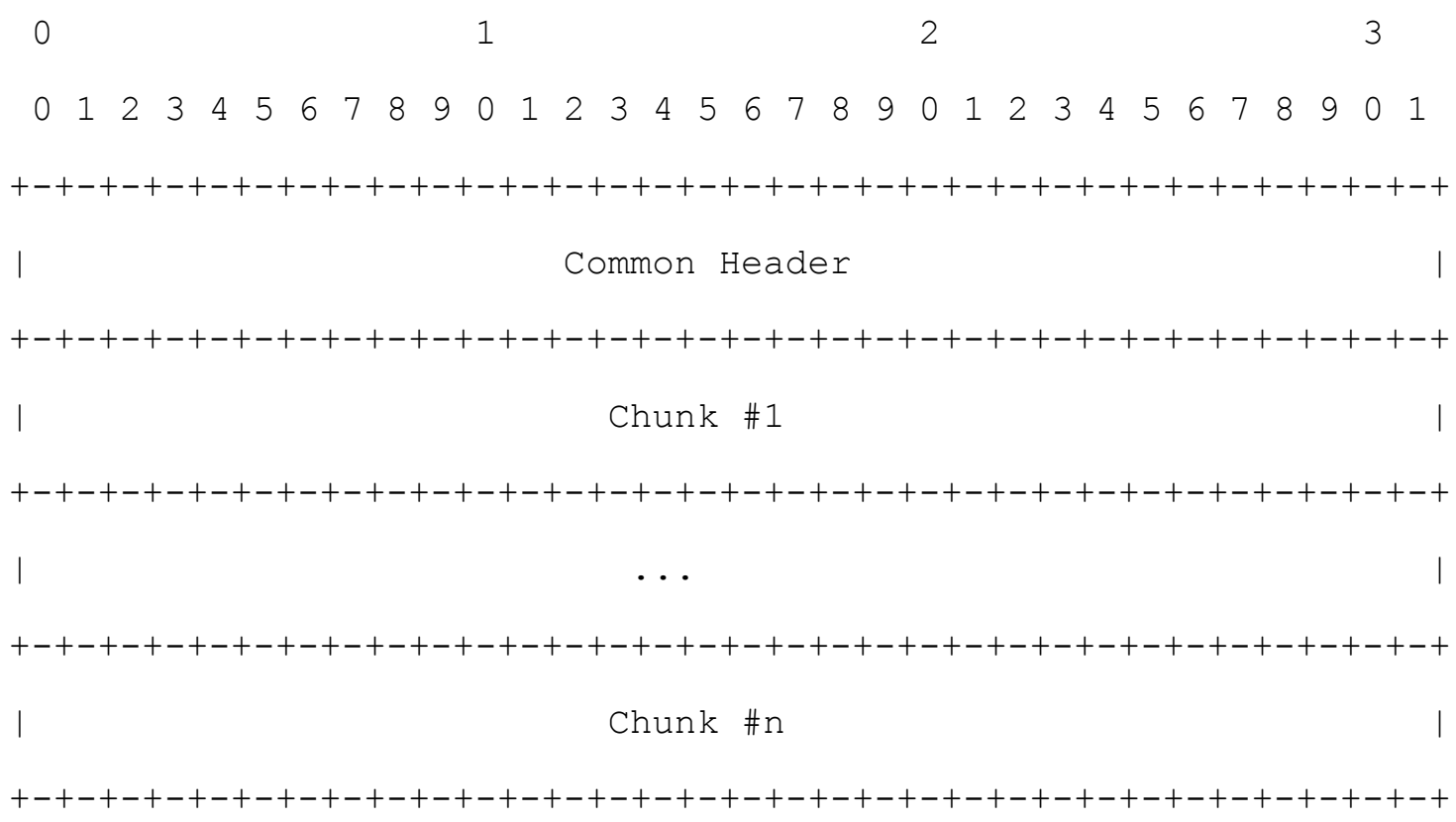


# SCTP association



# SCTP Packets

**SCTP packet Format (ascii art straight from RFC4960)**

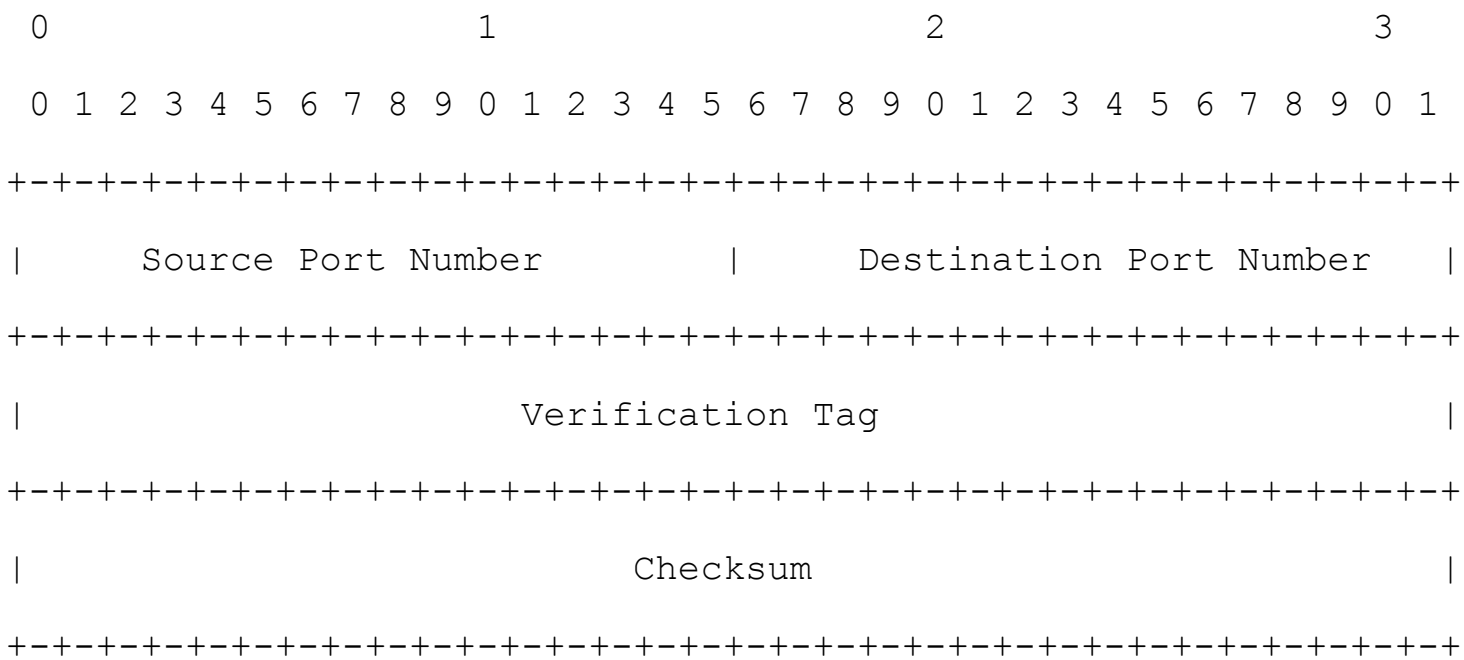


# SCTP Chunk types

| ID Value | Chunk Type |
|---|---|
| 0 | - Payload Data (DATA) |
| **1** | **- Initiation (INIT)** |
| **2** | **- Initiation Acknowledgement (INIT ACK)** |
| 3 | - Selective Acknowledgement (SACK) |
| 4 | - Heartbeat Request (HEARTBEAT) |
| 5 | - Heartbeat Acknowledgement (HEARTBEAT ACK) |
| **6** | **- Abort (ABORT)** |
| 7 | - Shutdown (SHUTDOWN) |
| **8** | **- Shutdown Acknowledgement (SHUTDOWN ACK)** |
| 9 | - Operation Error (ERROR) |
| 10 | - State Cookie (COOKIE ECHO) |
| 11 | - Cookie Acknowledgement (COOKIE ACK) |
| 12 | - Reserved for Explicit Congestion Notification Echo (ECNE) |
| 13 | - Reserved for Congestion Window Reduced (CWR) |
| 14 | - Shutdown Complete (SHUTDOWN COMPLETE) |

# SCTP Header

- **SCTP Common Header Format**



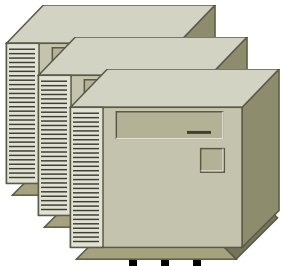
# SCTPscan: Mapping SIGTRAN

- SCTPscan
  - Linux, BSD, MacOS X, Solaris, ...
  - IP scan, portscan, fuzzing, dummy server, bridge
  - Included in BackTrack
- SCTP Tricks: port mirroring, instreams connections
  - NMAP new SCTP support (-Y), lacks tricks
- SIGTRAN usually requires peer config
  - This is not the average TCP/IP app

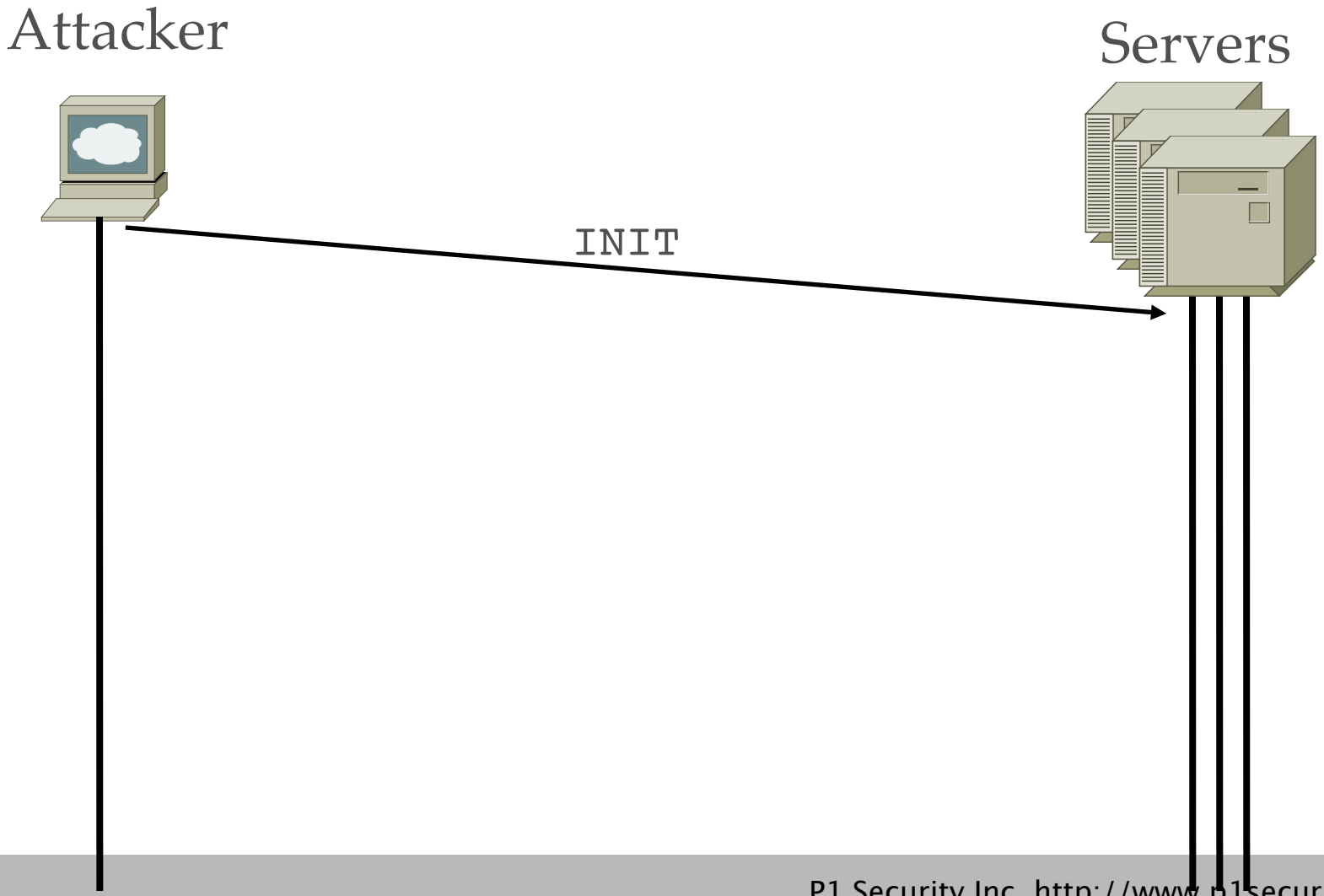
# From RFC...

Attacker

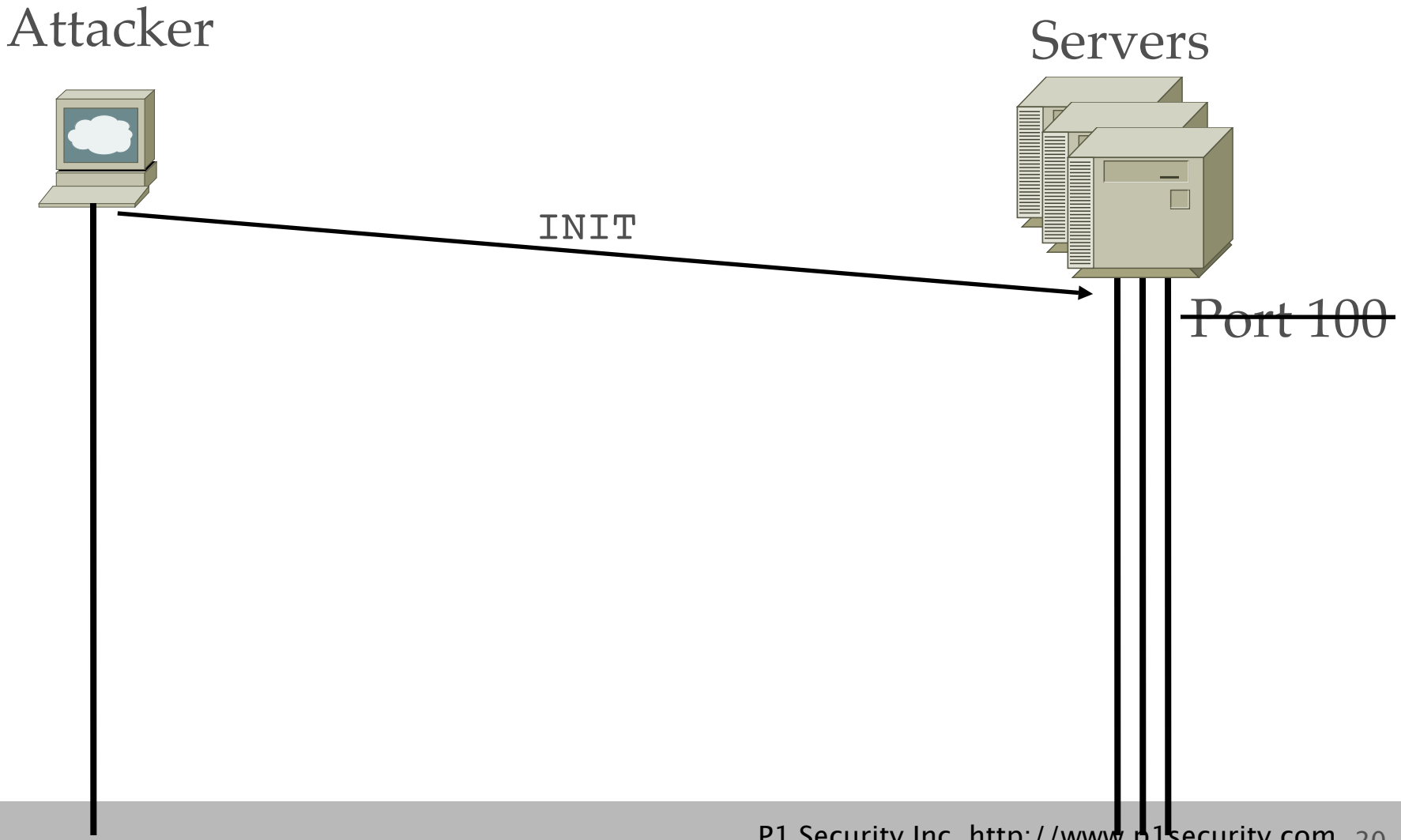
Servers



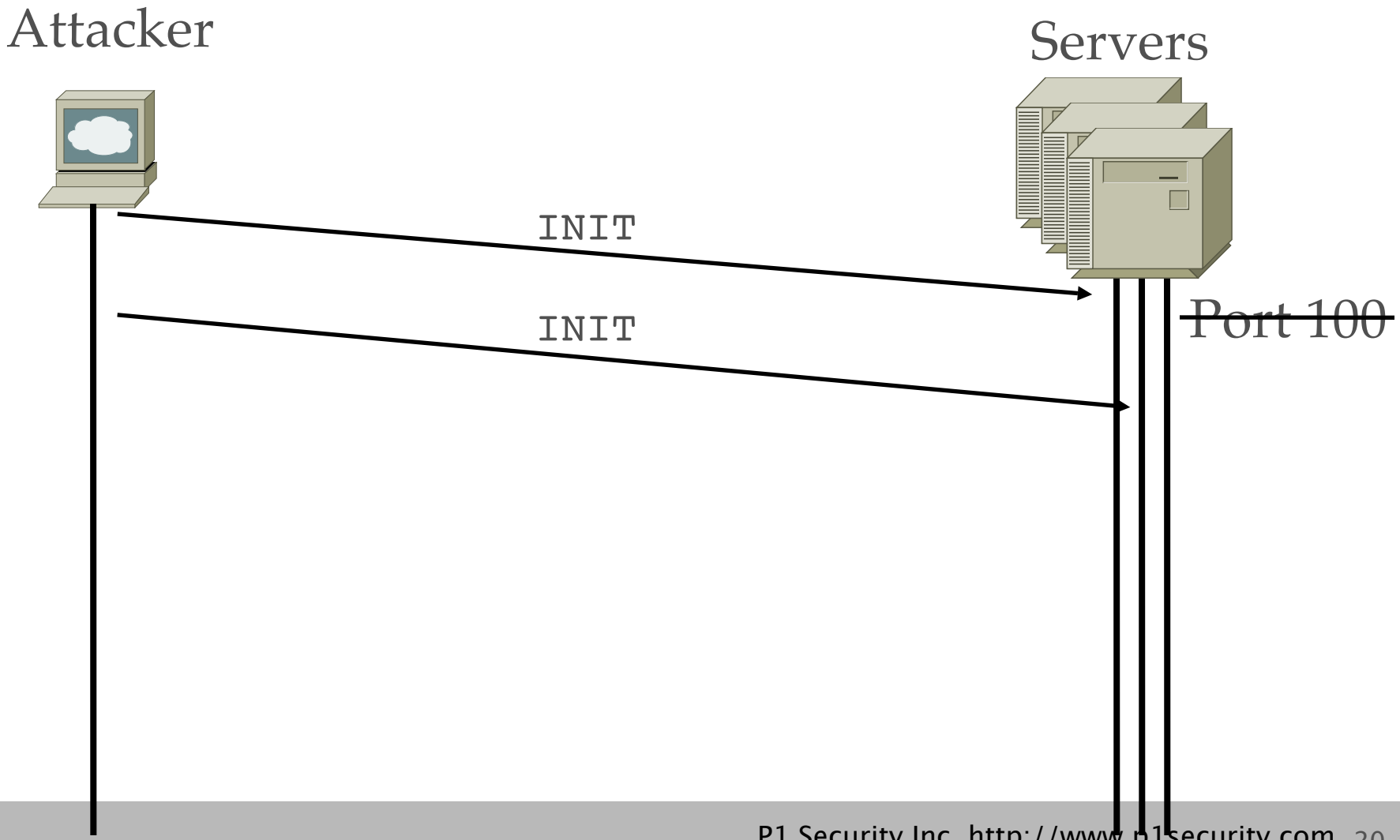
# From RFC...



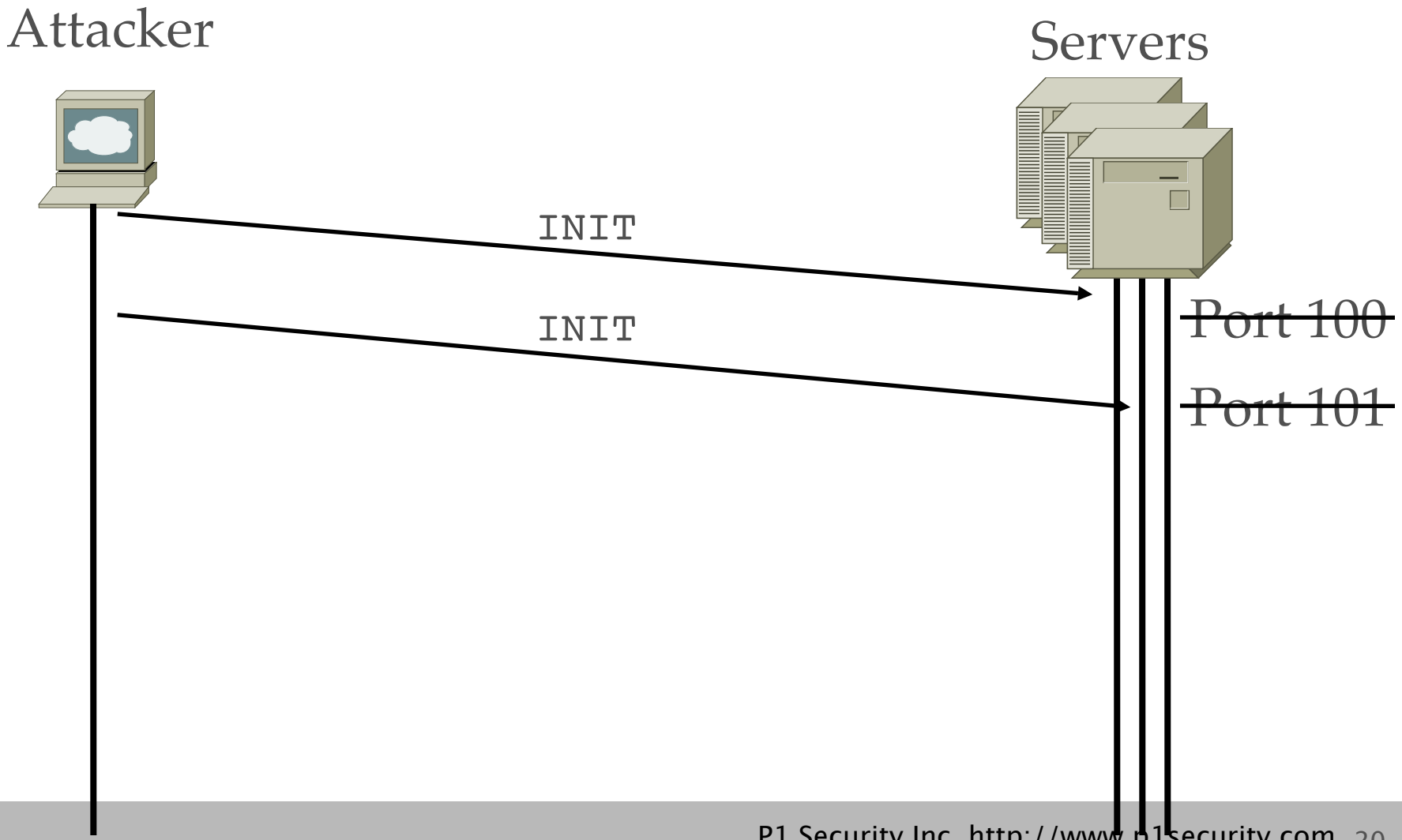
# From RFC...



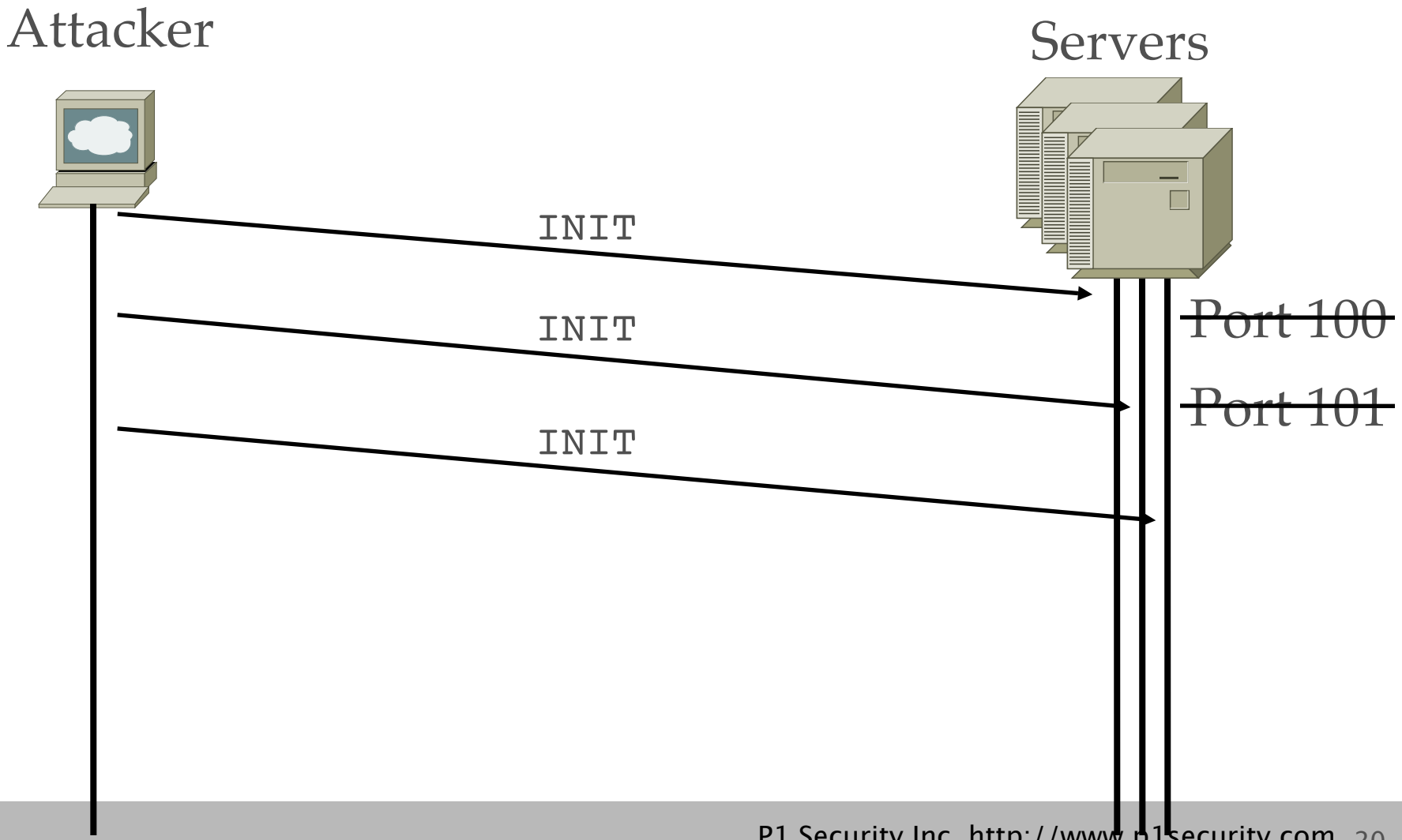
# From RFC...



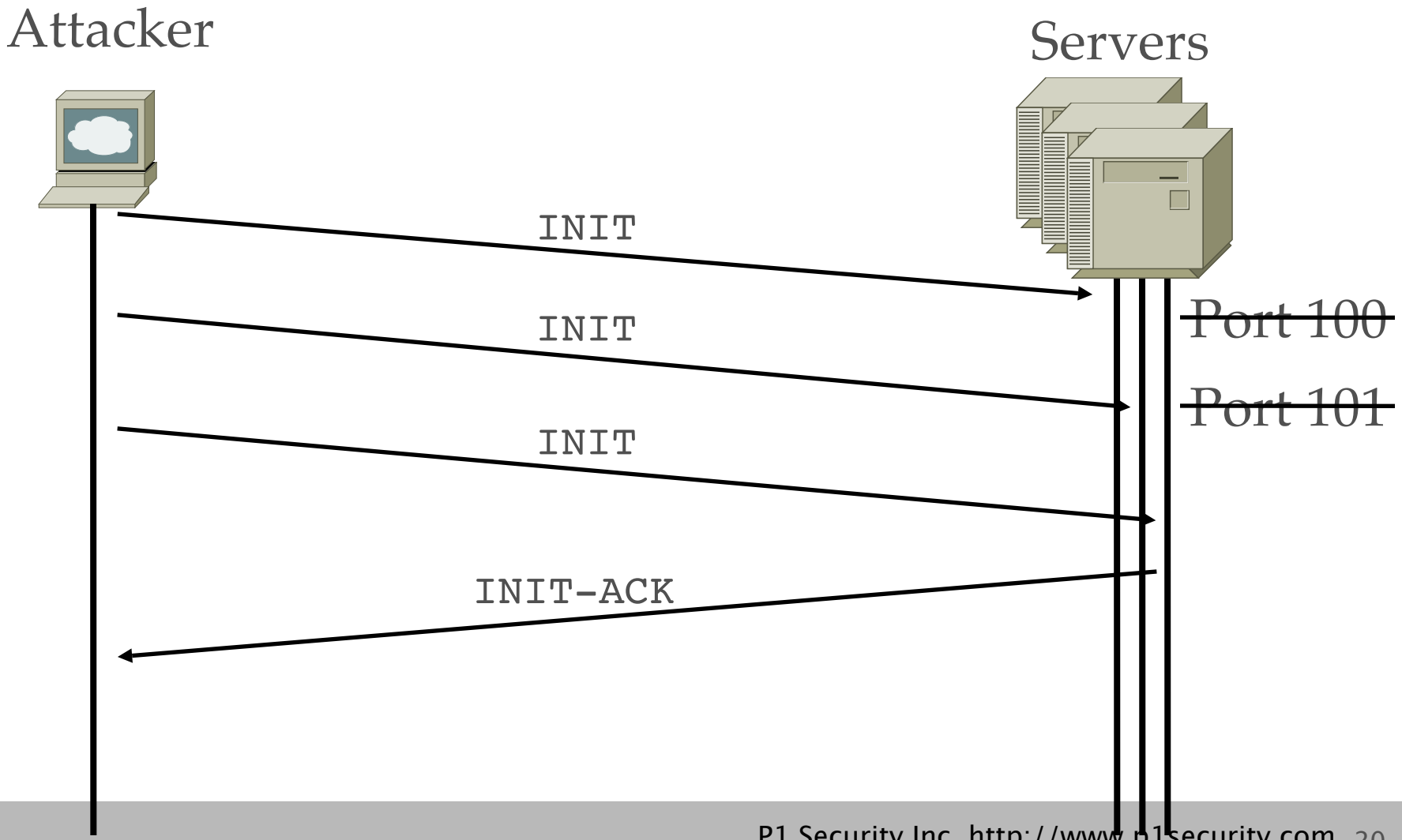
# From RFC...



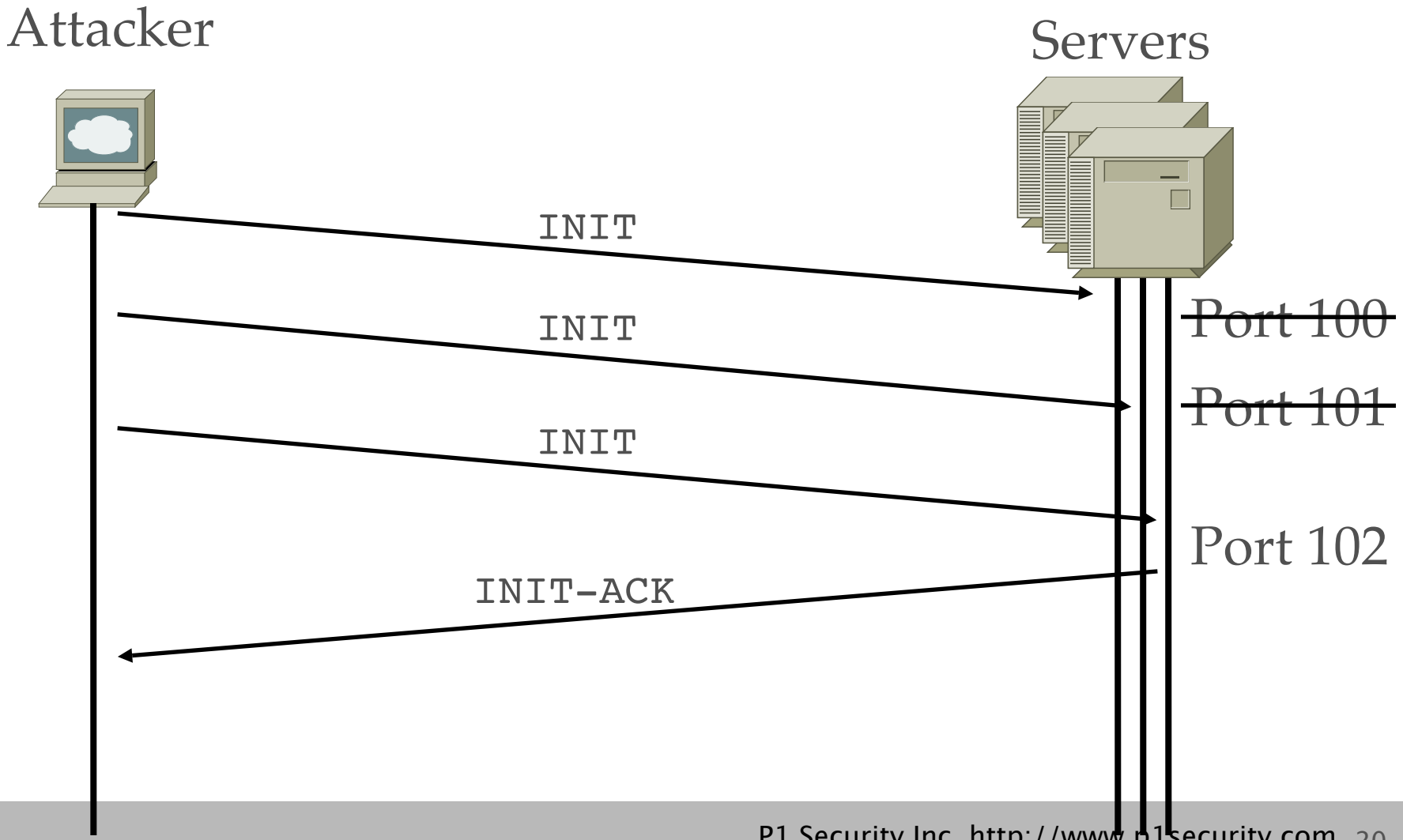
# From RFC...



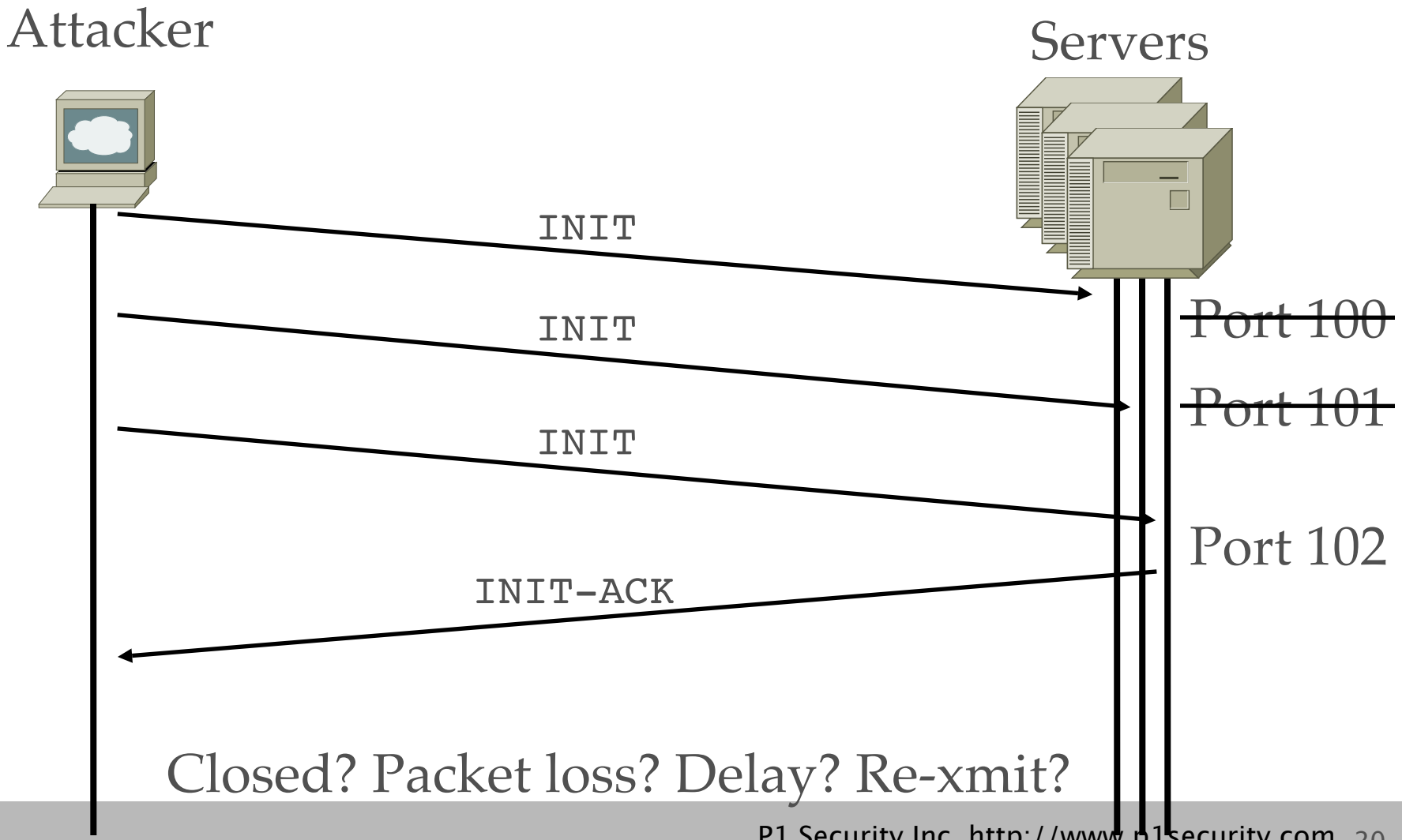
# From RFC...



# From RFC...

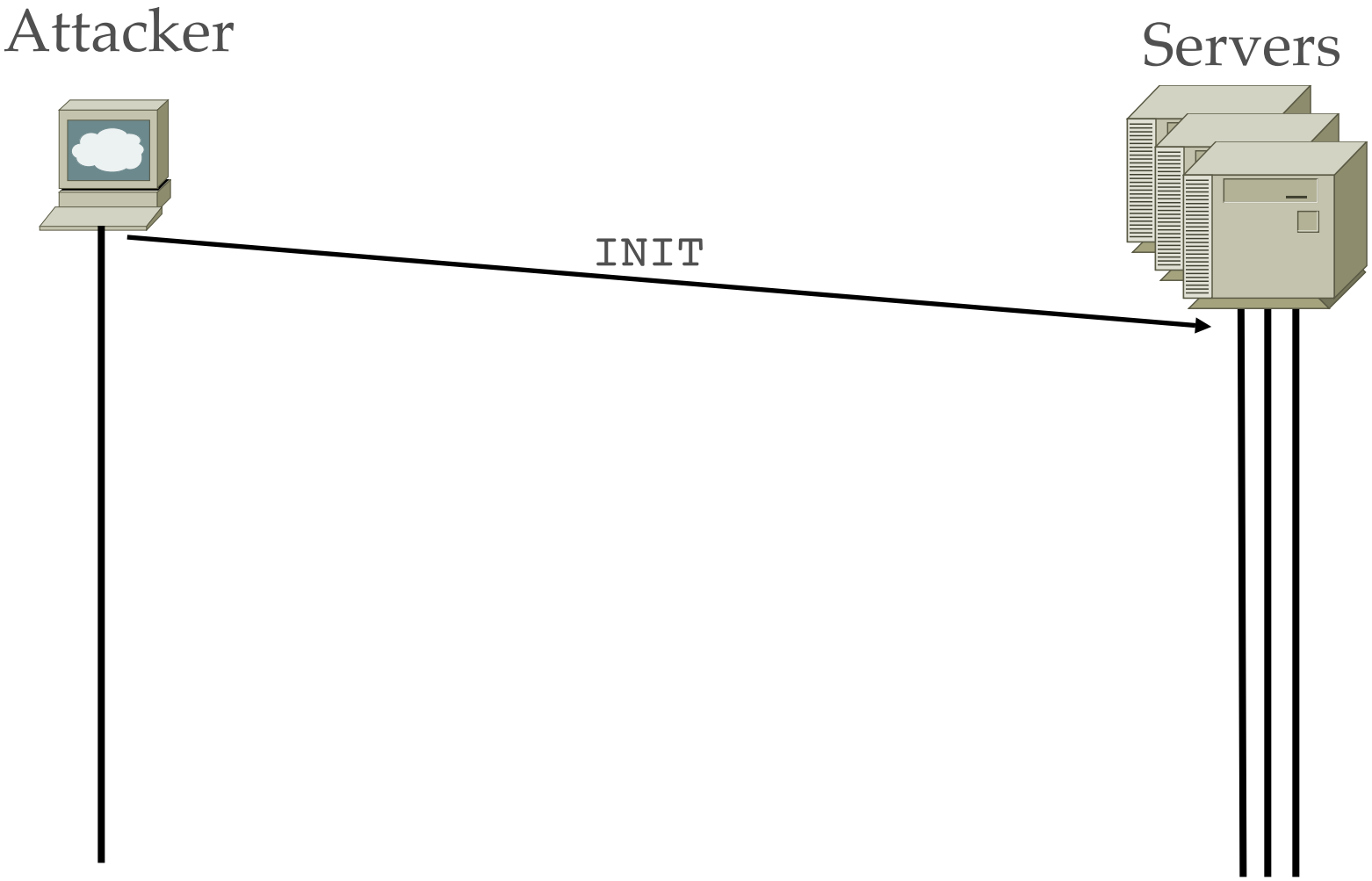

Attacker
Servers
INIT
INIT
INIT
INIT-ACK
Port 100
Port 101
Port 102

# From RFC...

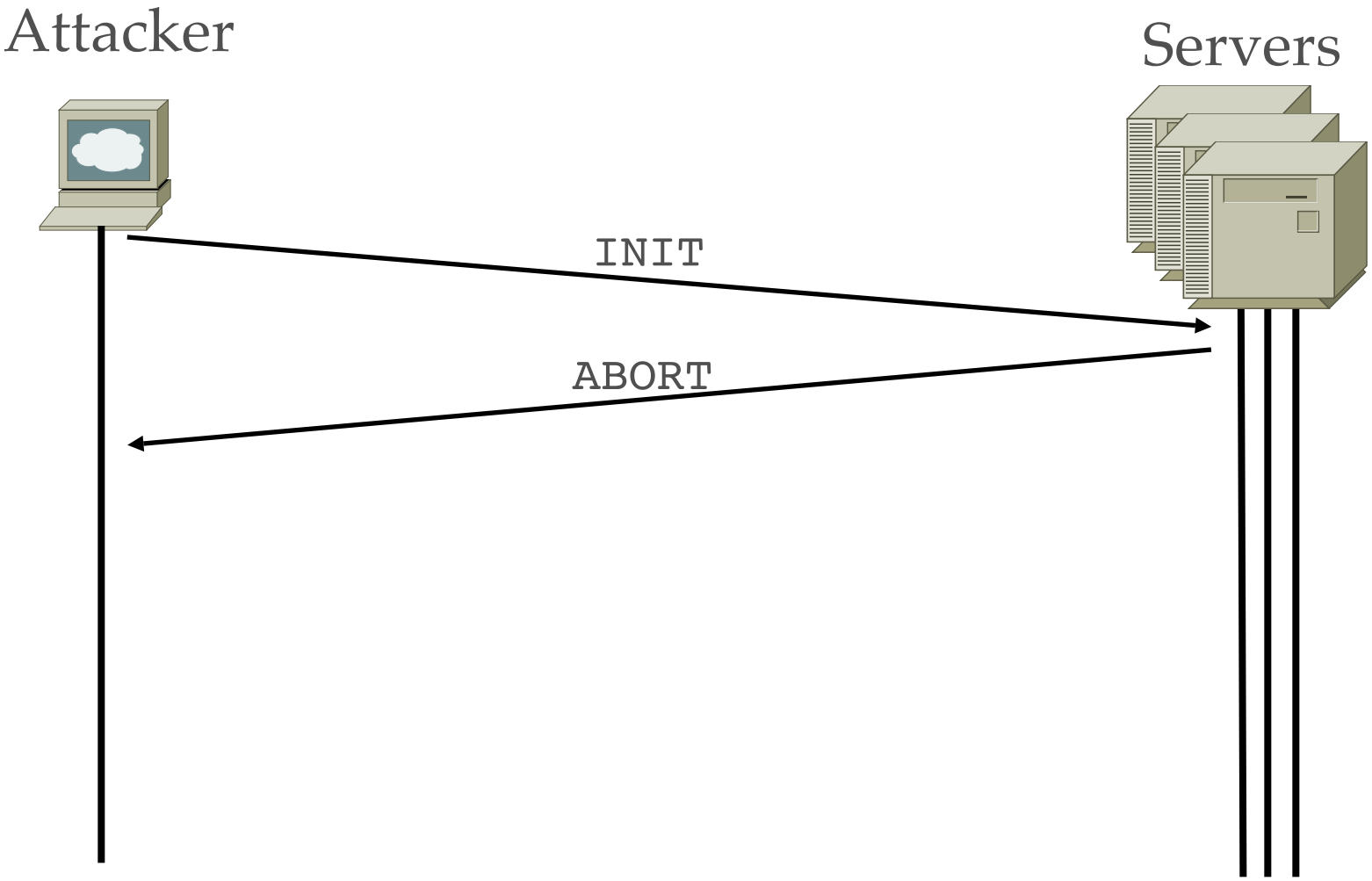

Attacker
Servers
INIT
INIT
INIT
INIT-ACK
Port 100
Port 101
Port 102
Closed? Packet loss? Delay? Re-xmit?

# Improved SCTPscan: stealth scan

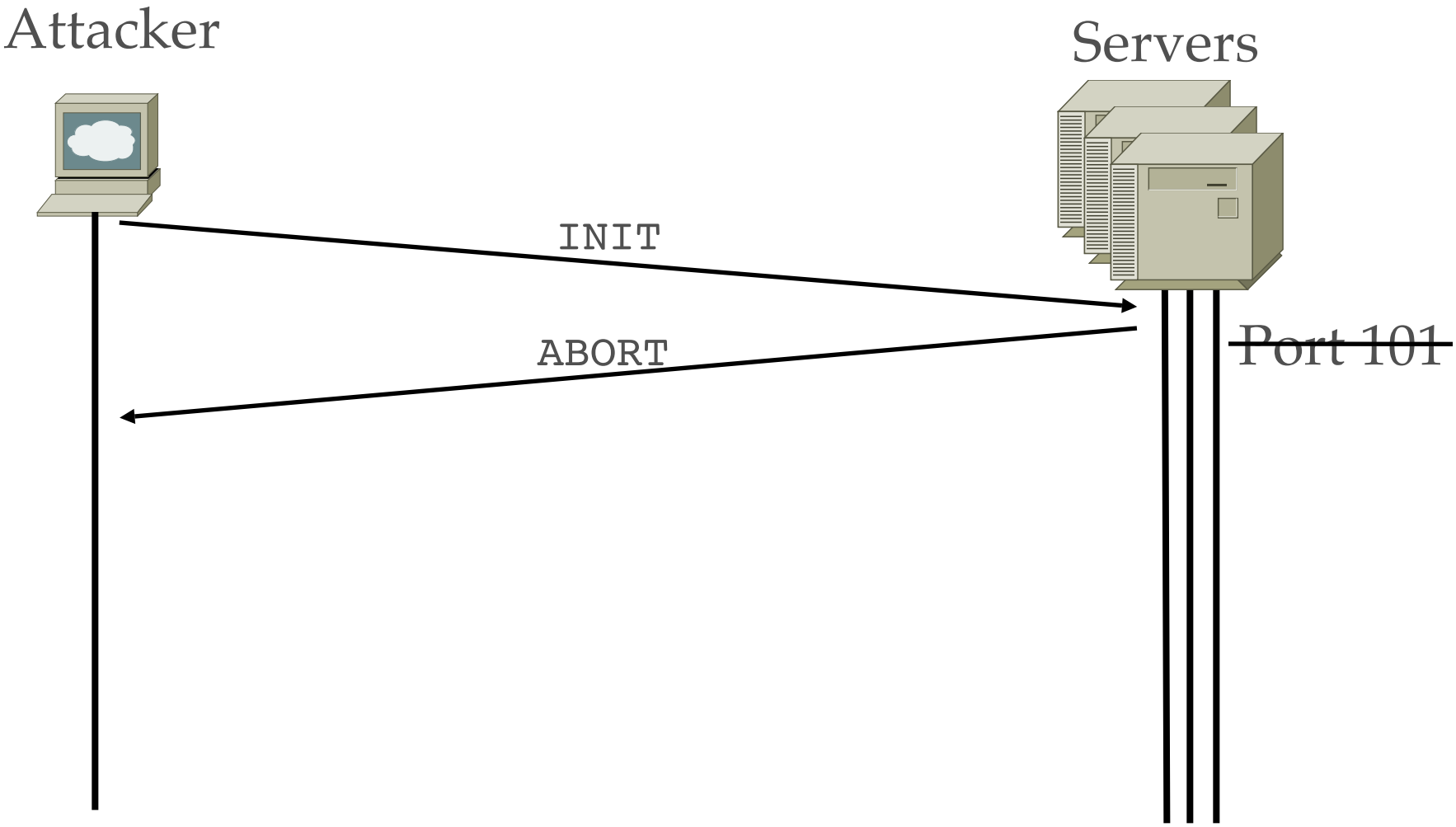
# Improved SCTPscan: stealth scan



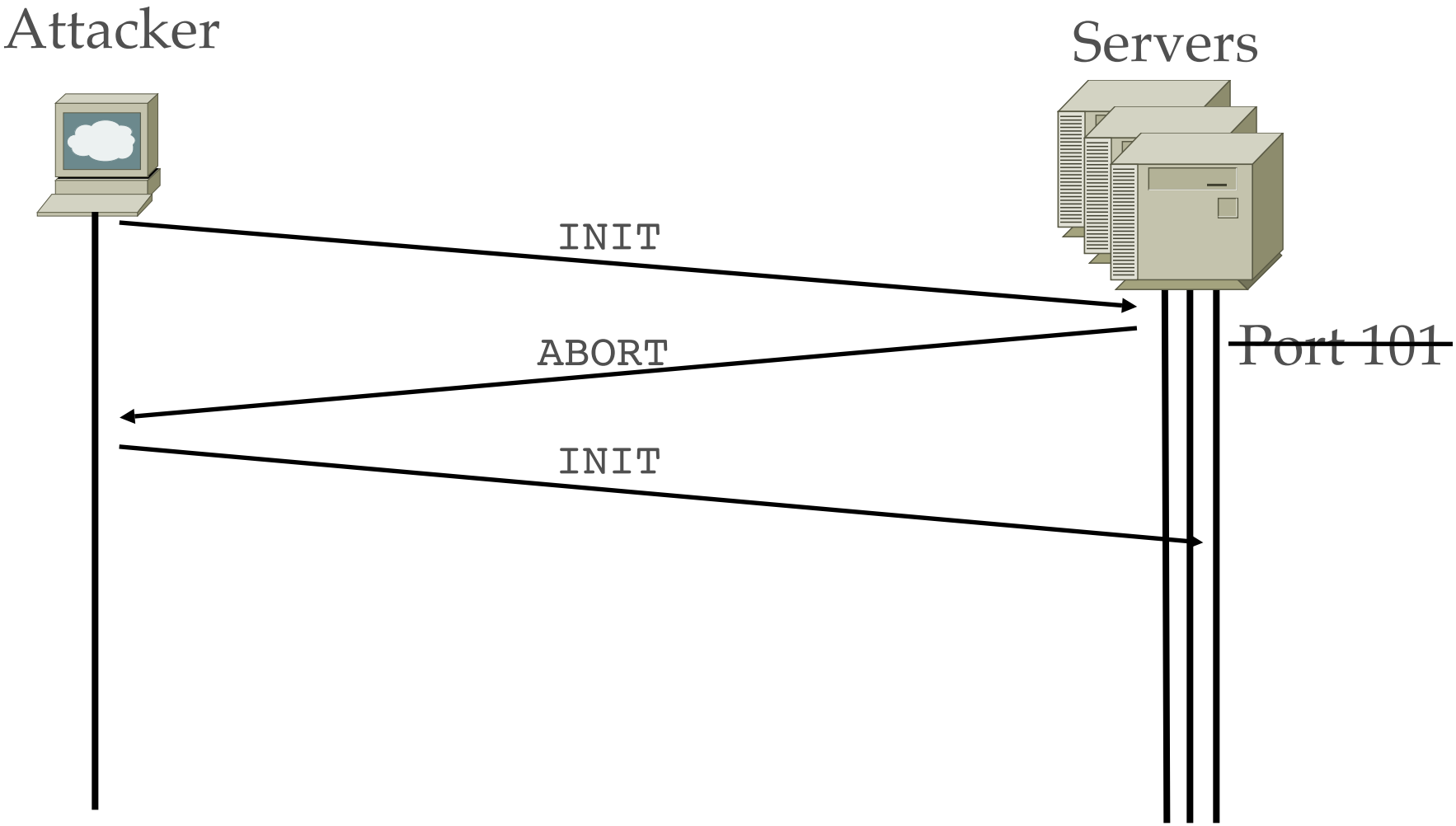
# Improved SCTPscan: stealth scan



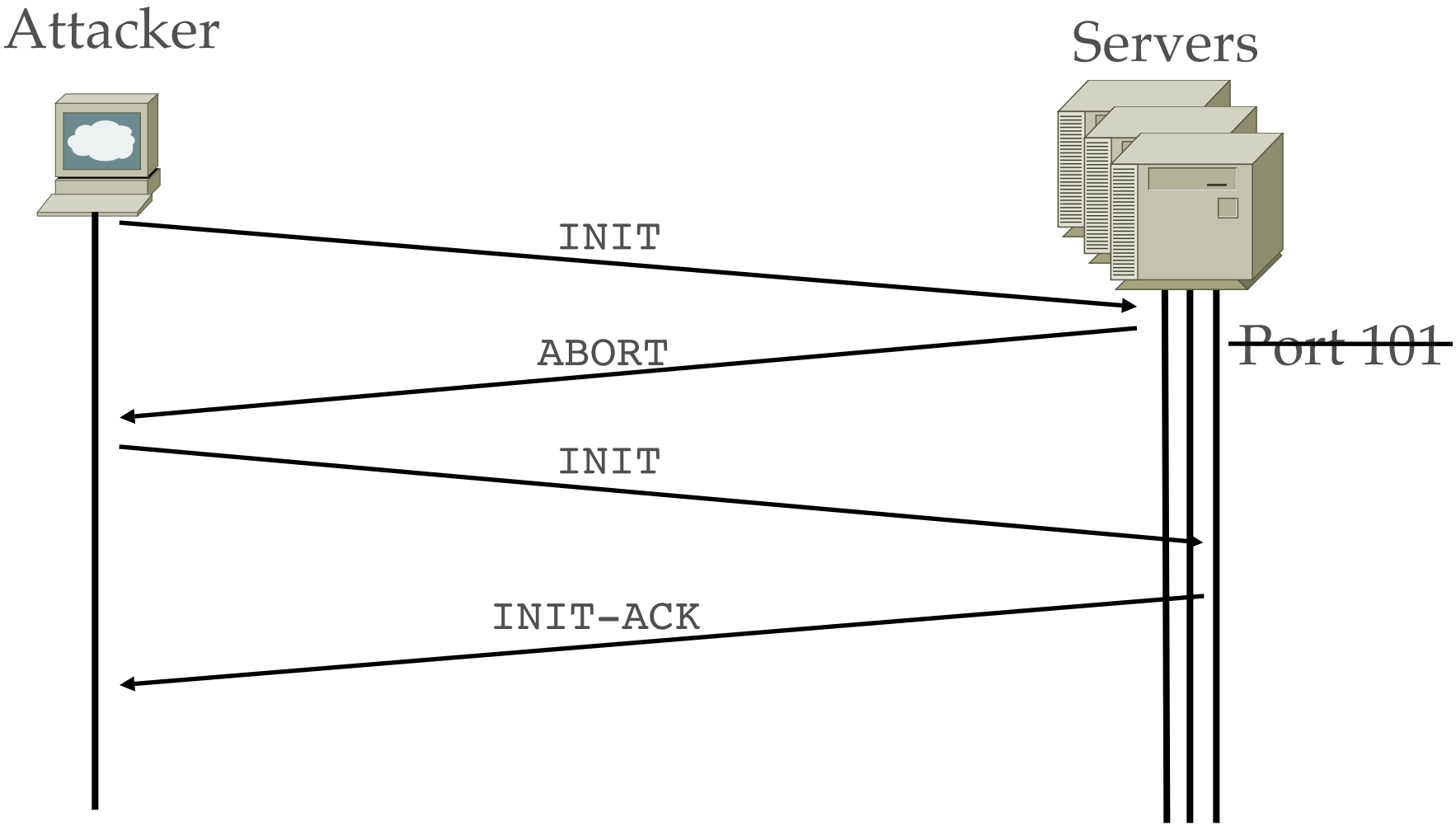
# Improved SCTPscan: stealth scan



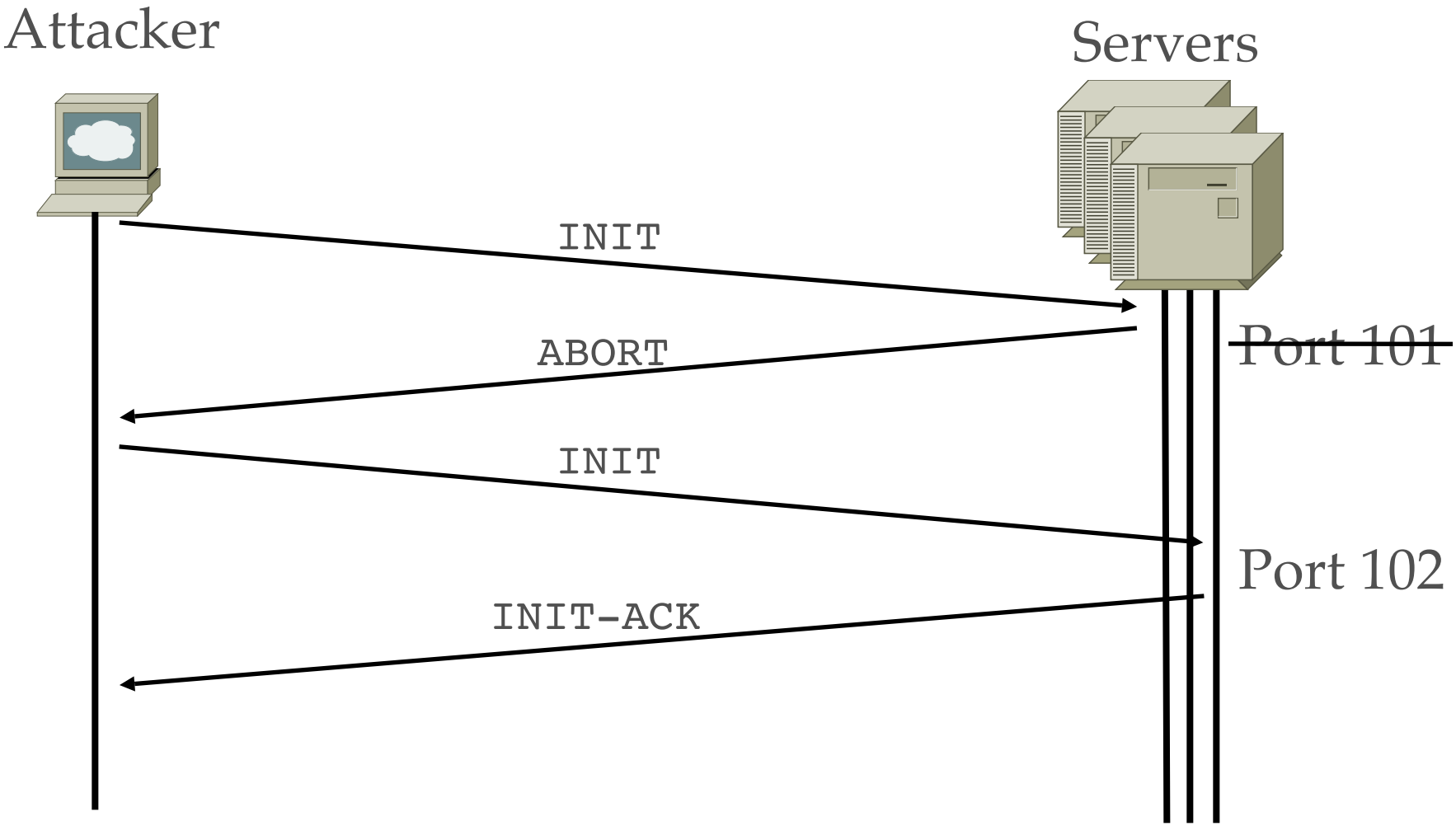
# Improved SCTPscan: stealth scan



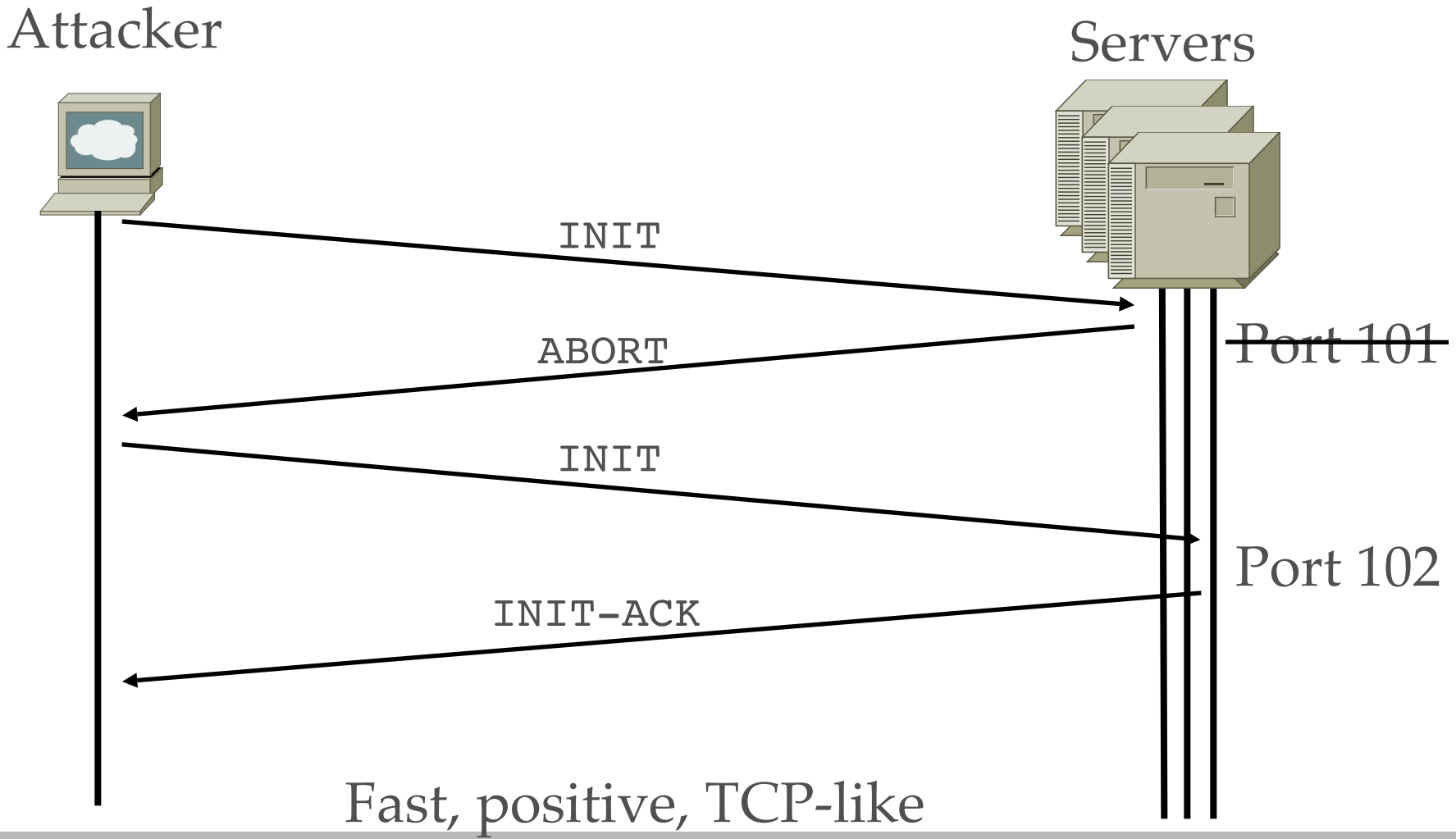
# Improved SCTPscan: stealth scan



# Improved SCTPscan: stealth scan



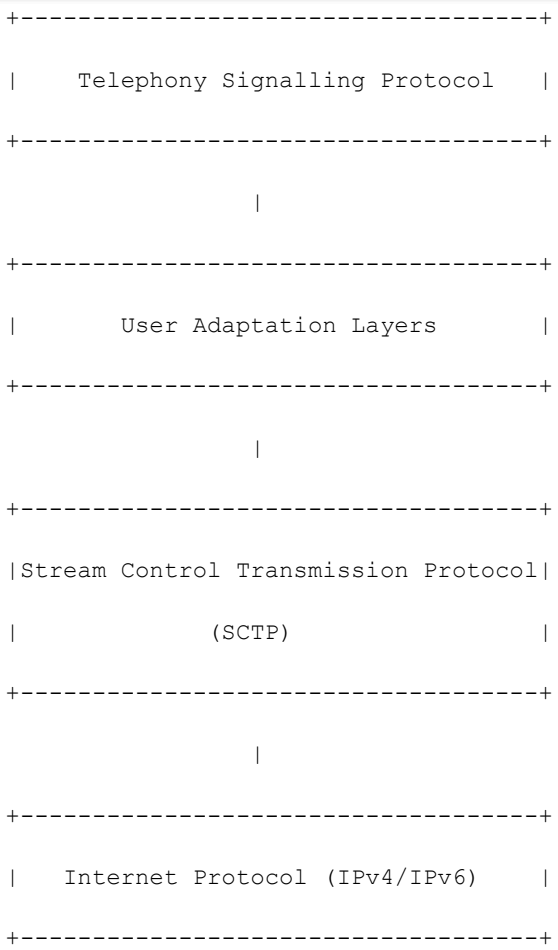
# Improved SCTPscan: stealth scan



# SCTPscan Usage

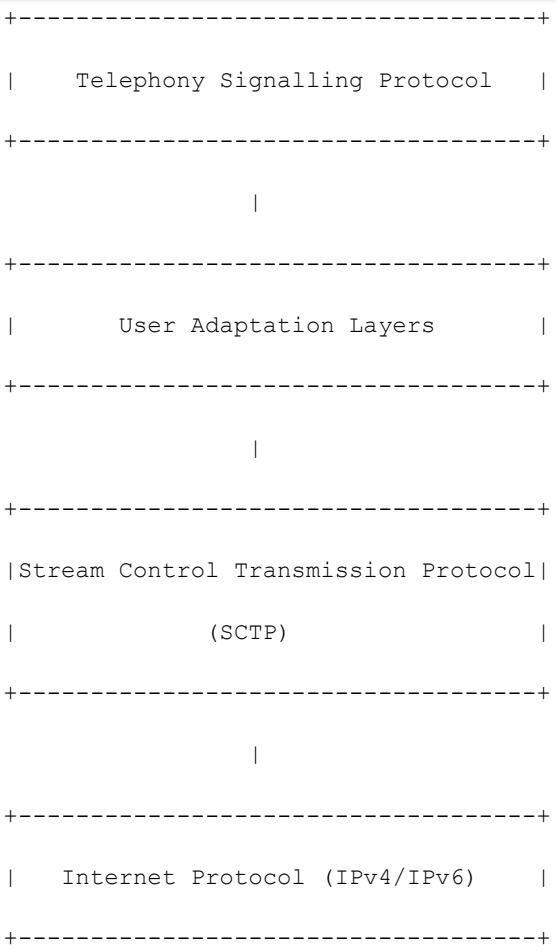
```
root@gate:~/sctp# ./sctpscan --scan --autoportscan
-r 203.151.1
Netscanning with Crc32 checksumed packet
203.151.1.4 SCTP present on port 2905
203.151.1.4 SCTP present on port 7551
203.151.1.4 SCTP present on port 7701
203.151.1.4 SCTP present on port 8001
203.151.1.4 SCTP present on port 2905
root@gate:~/sctp#
```

# What goes over SCTP?




From RFC 4166

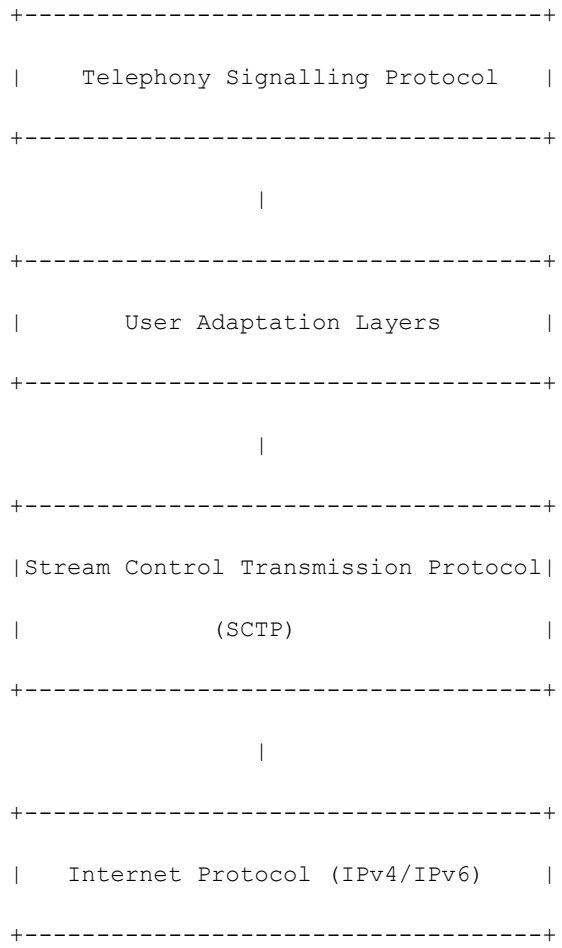
# What goes over SCTP?



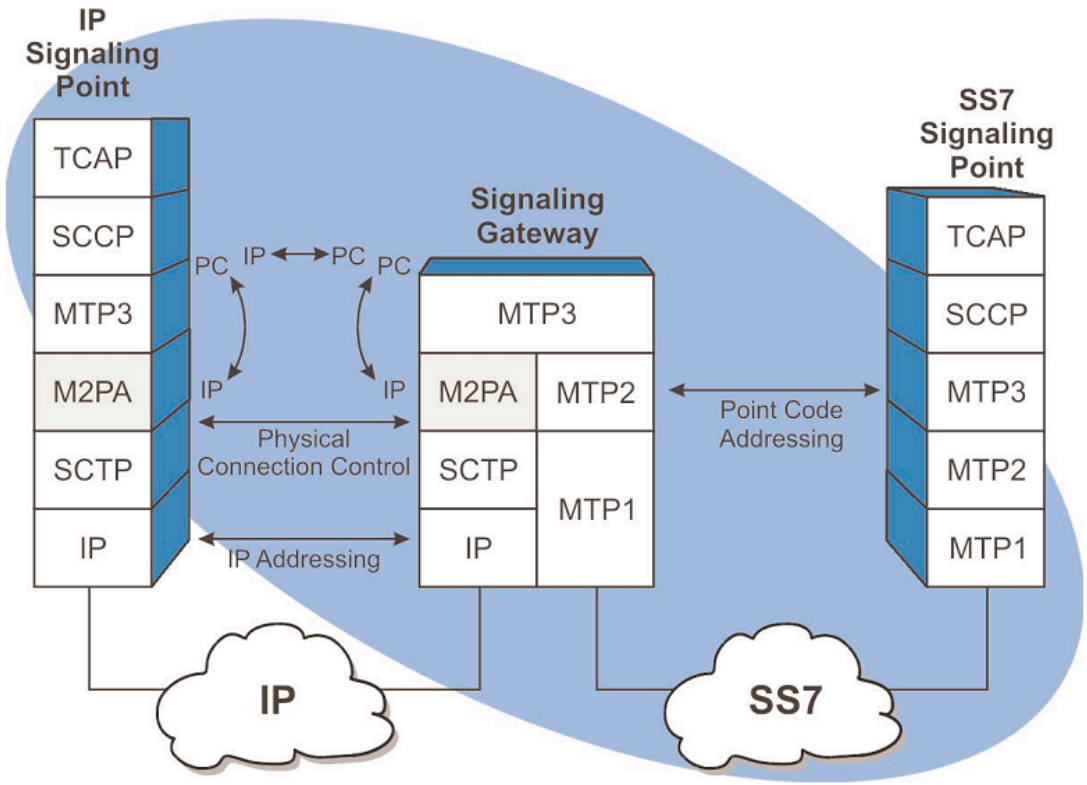

User Adapation Layer: M2PA

From RFC 4166

# What goes over SCTP?

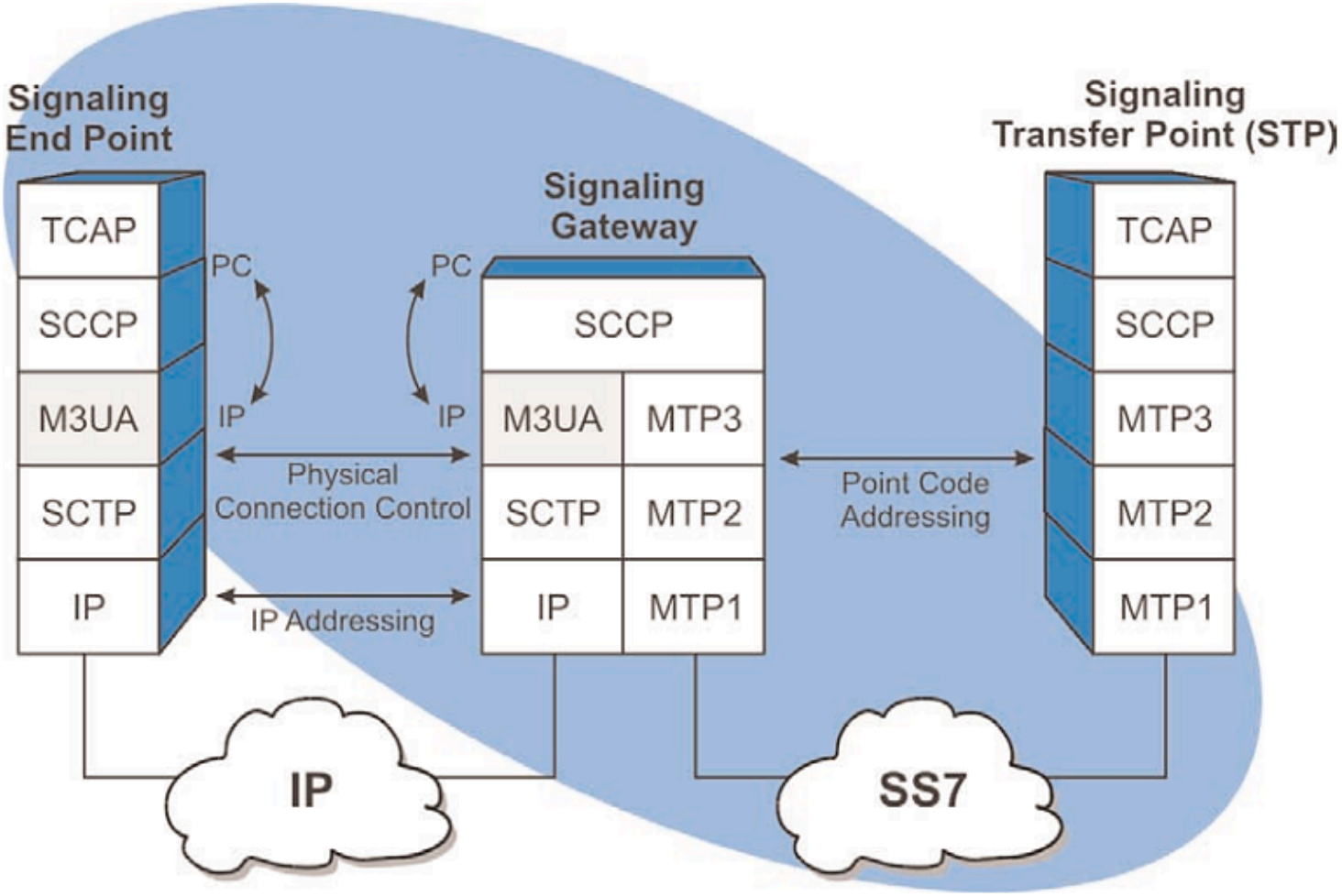



## User Adaptation Layer: M2PA




From RFC 4166

# M3UA Protocol Adaptation Layer

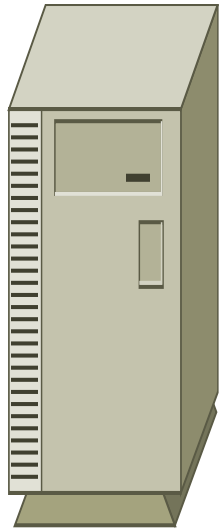


# SS7 Peering: attacker enemy

Legitimate Peer

Server or STP

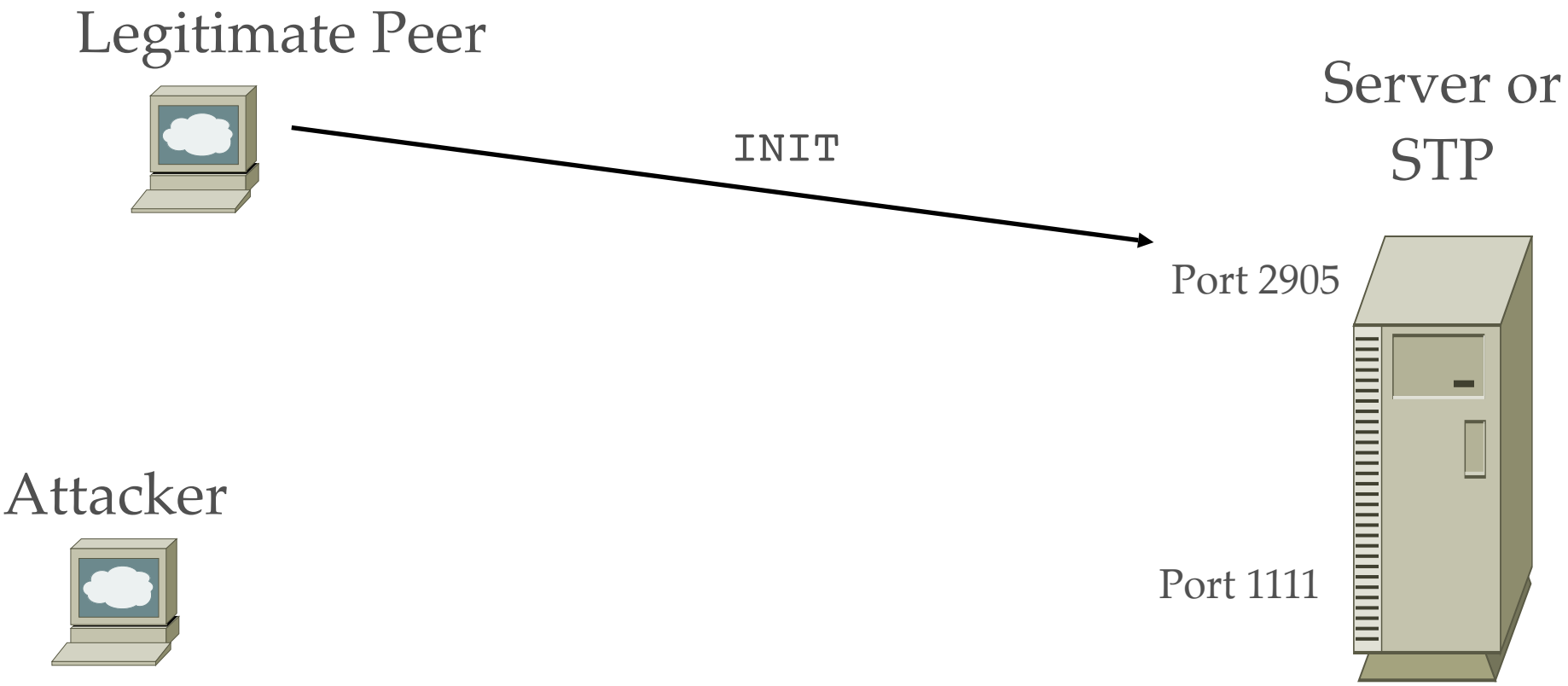
Port 2905



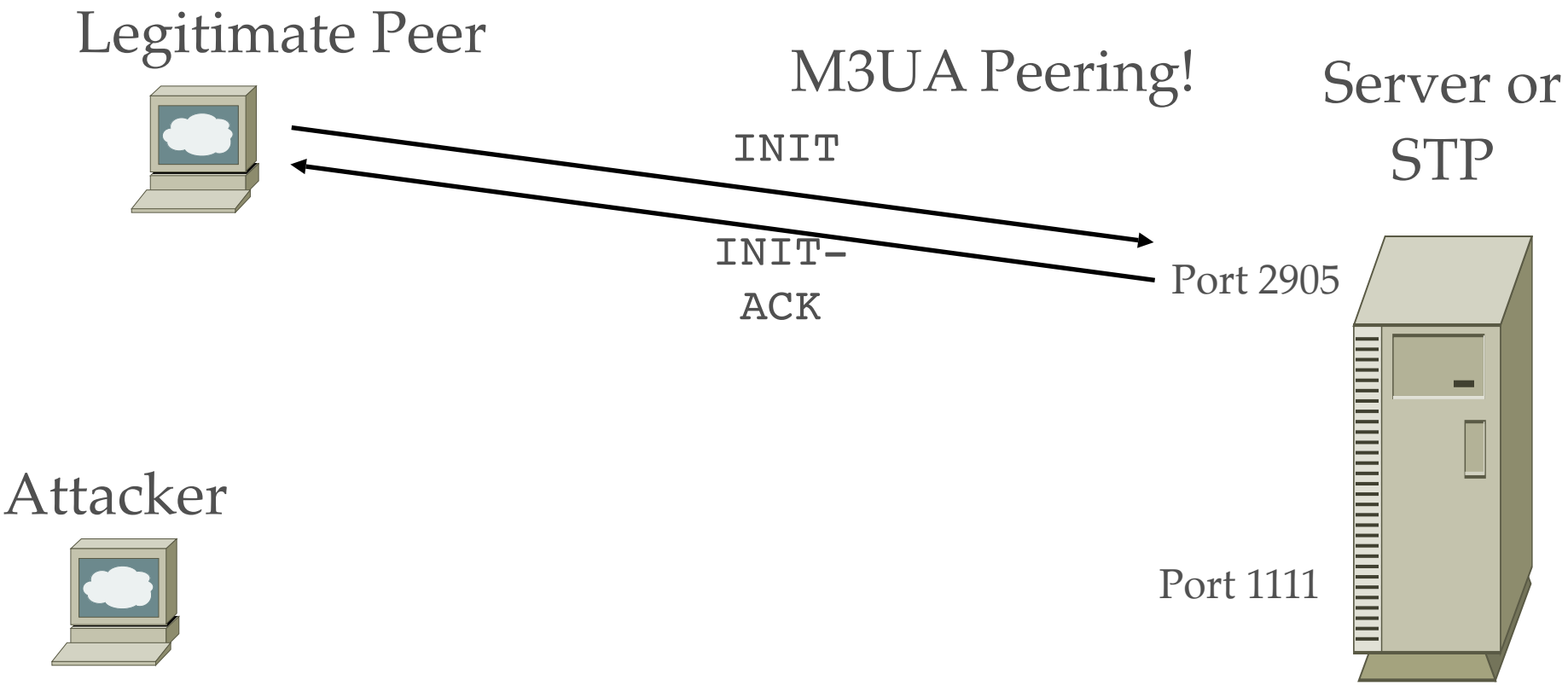
Attacker

Port 1111

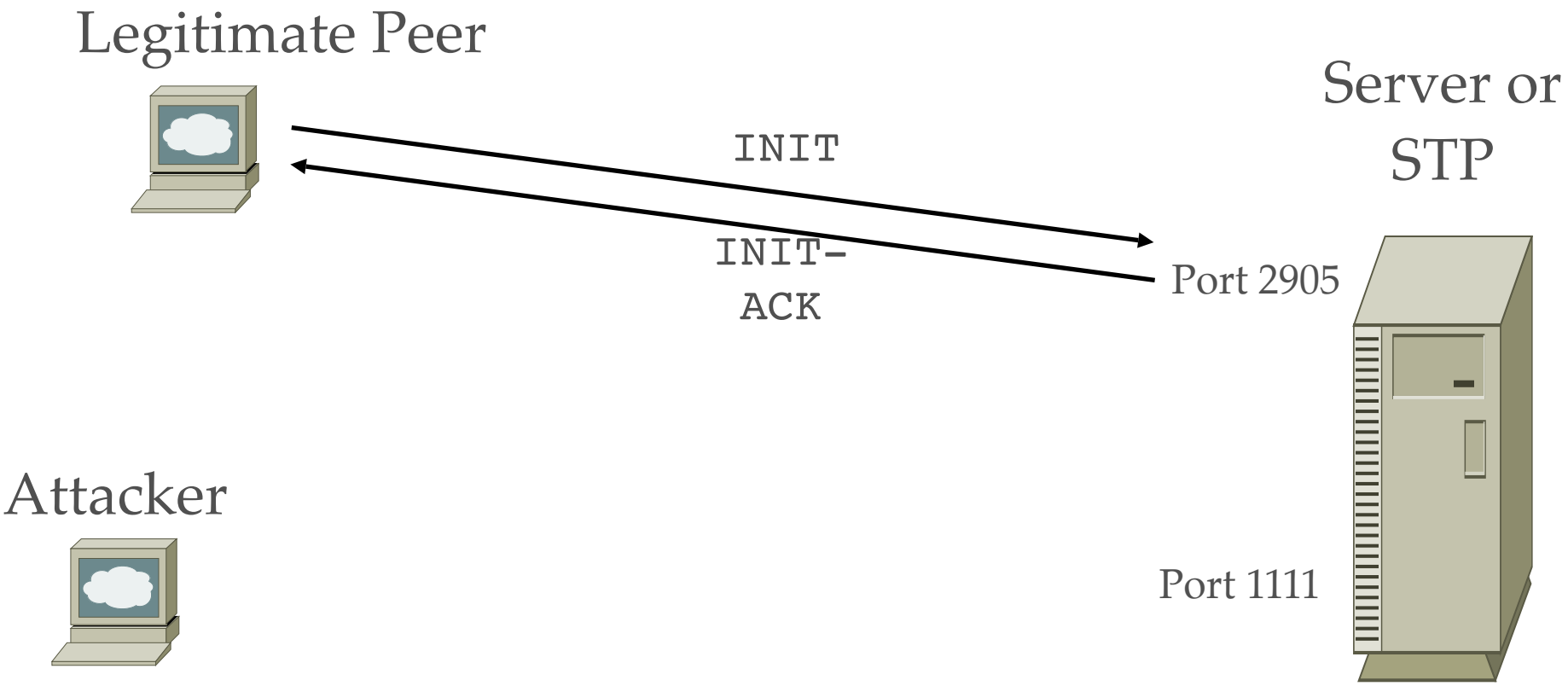
# SS7 Peering: attacker enemy



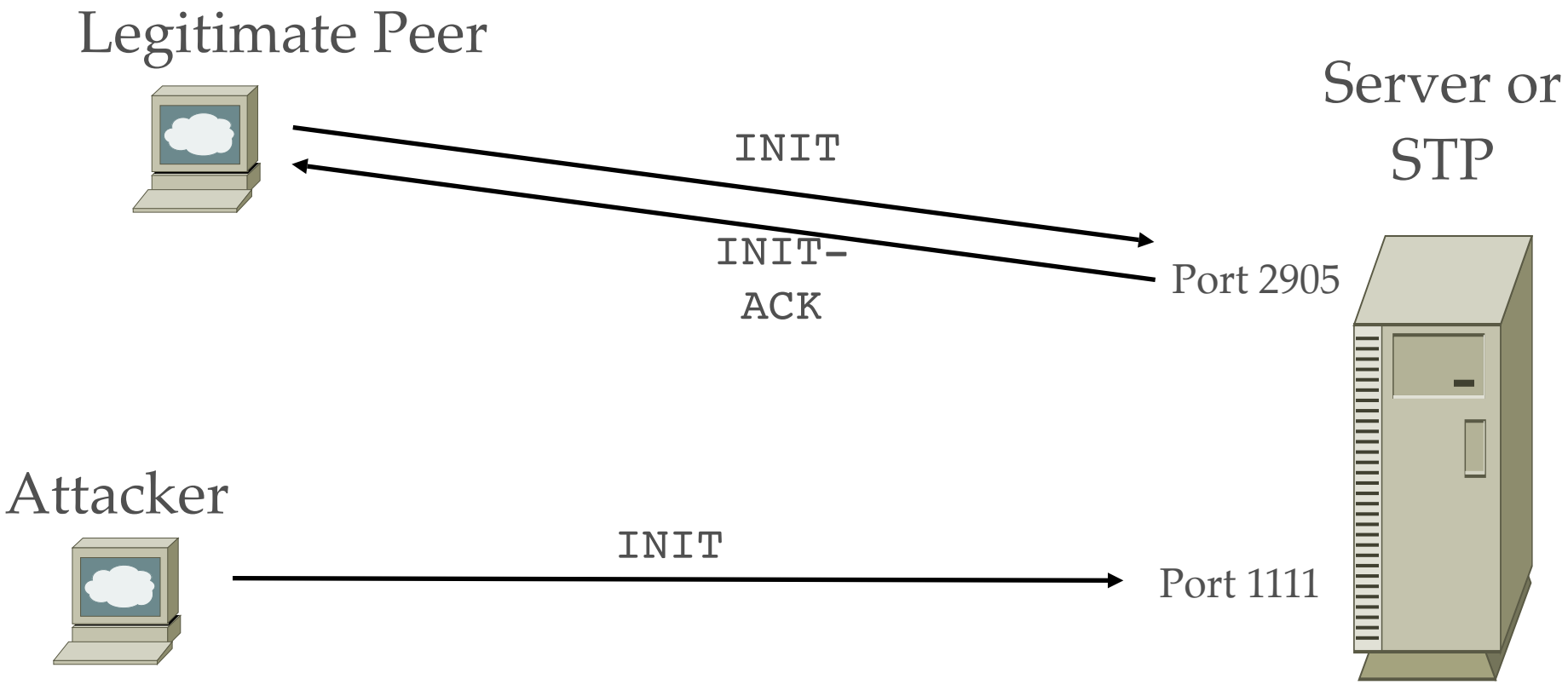
# SS7 Peering: attacker enemy



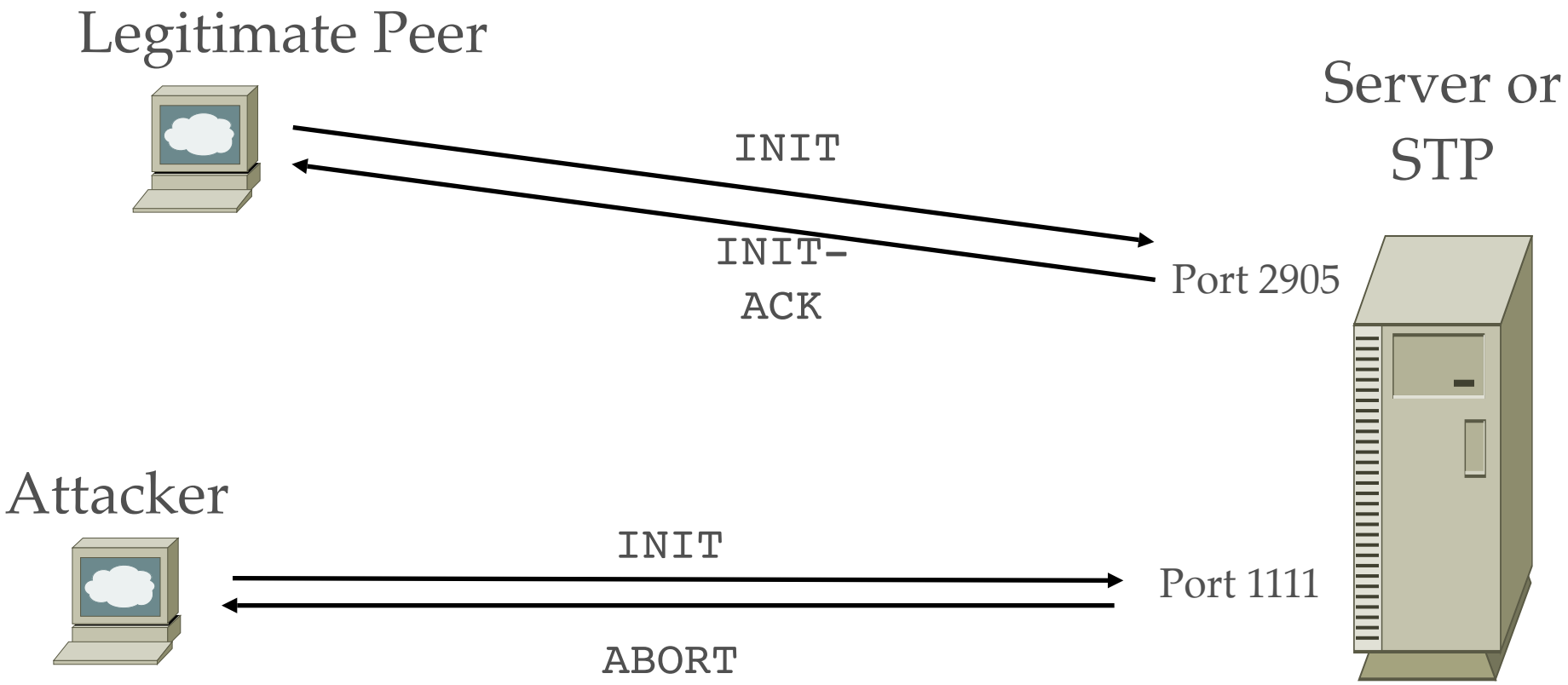
# SS7 Peering: attacker enemy



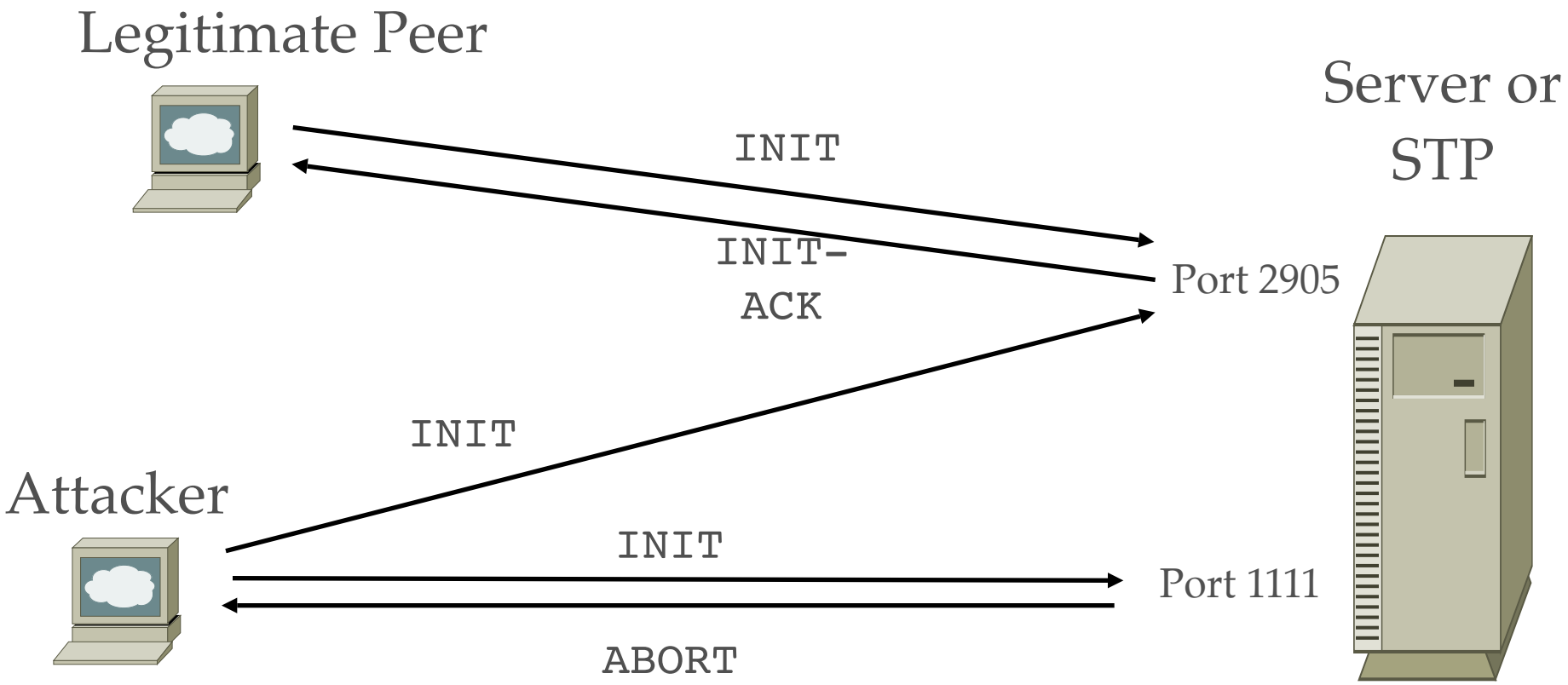
# SS7 Peering: attacker enemy



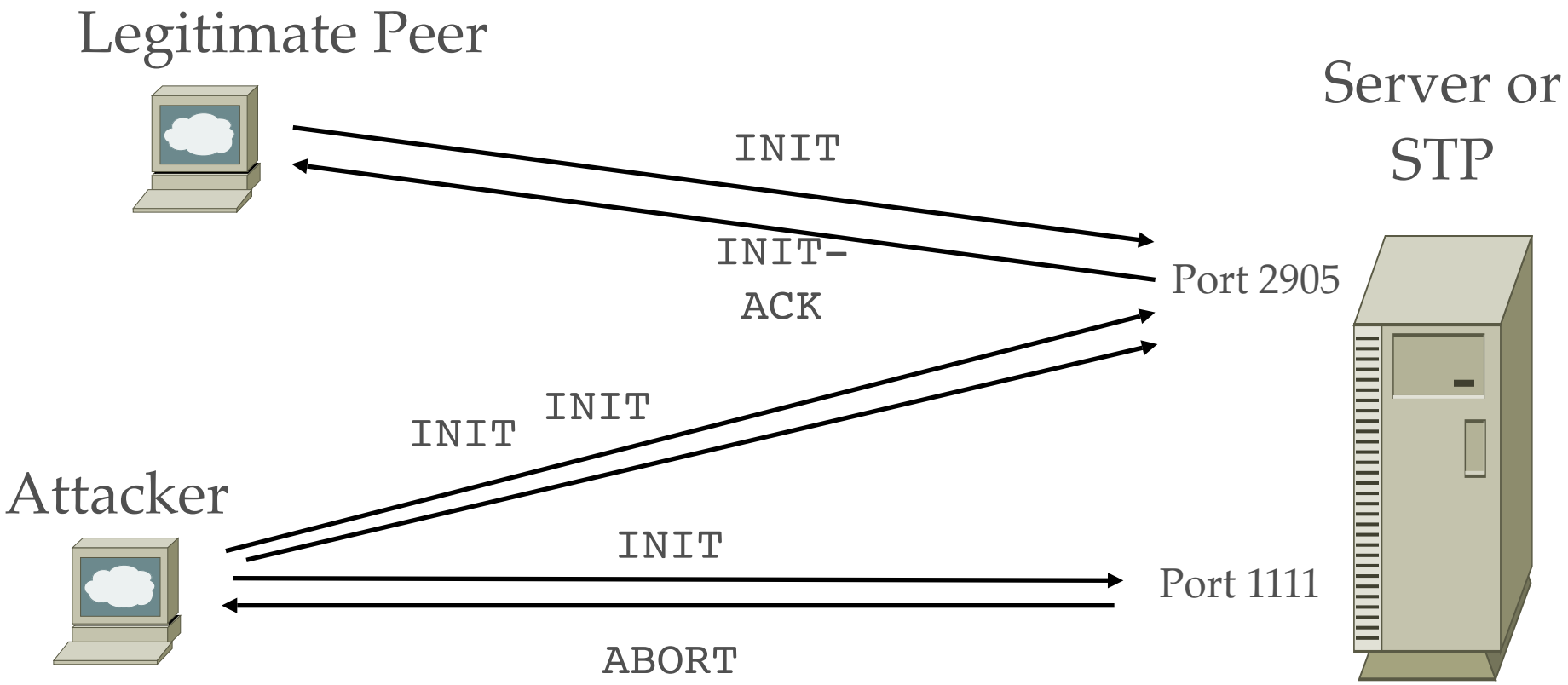
# SS7 Peering: attacker enemy



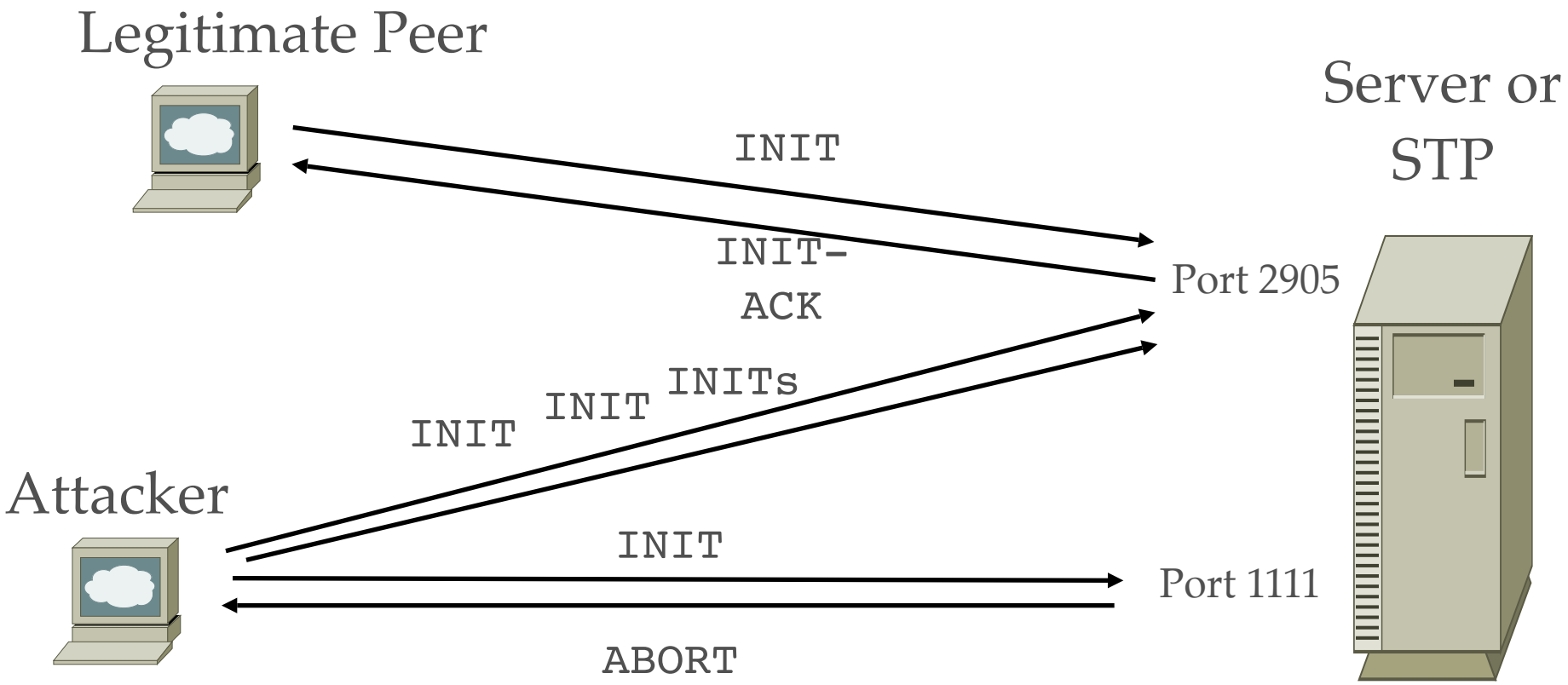
# SS7 Peering: attacker enemy



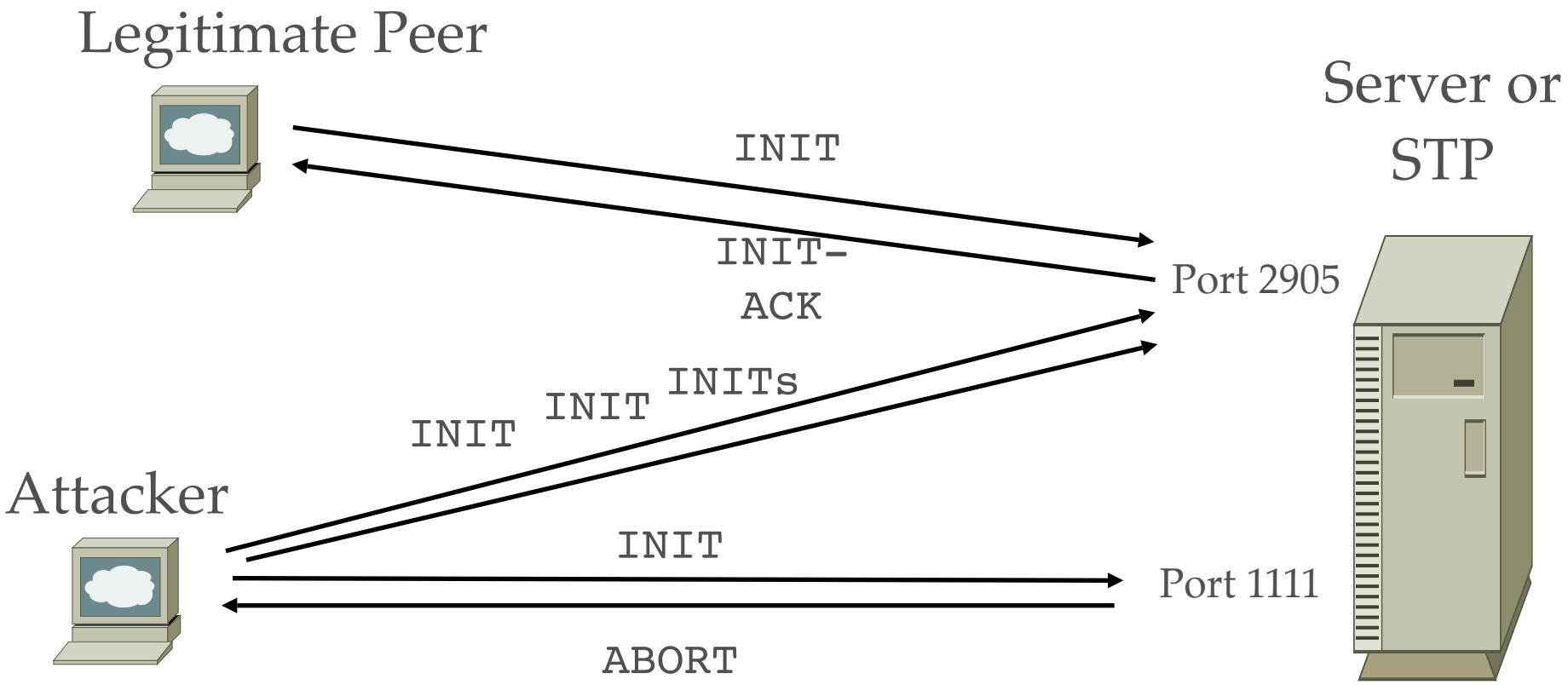
# SS7 Peering: attacker enemy



# SS7 Peering: attacker enemy

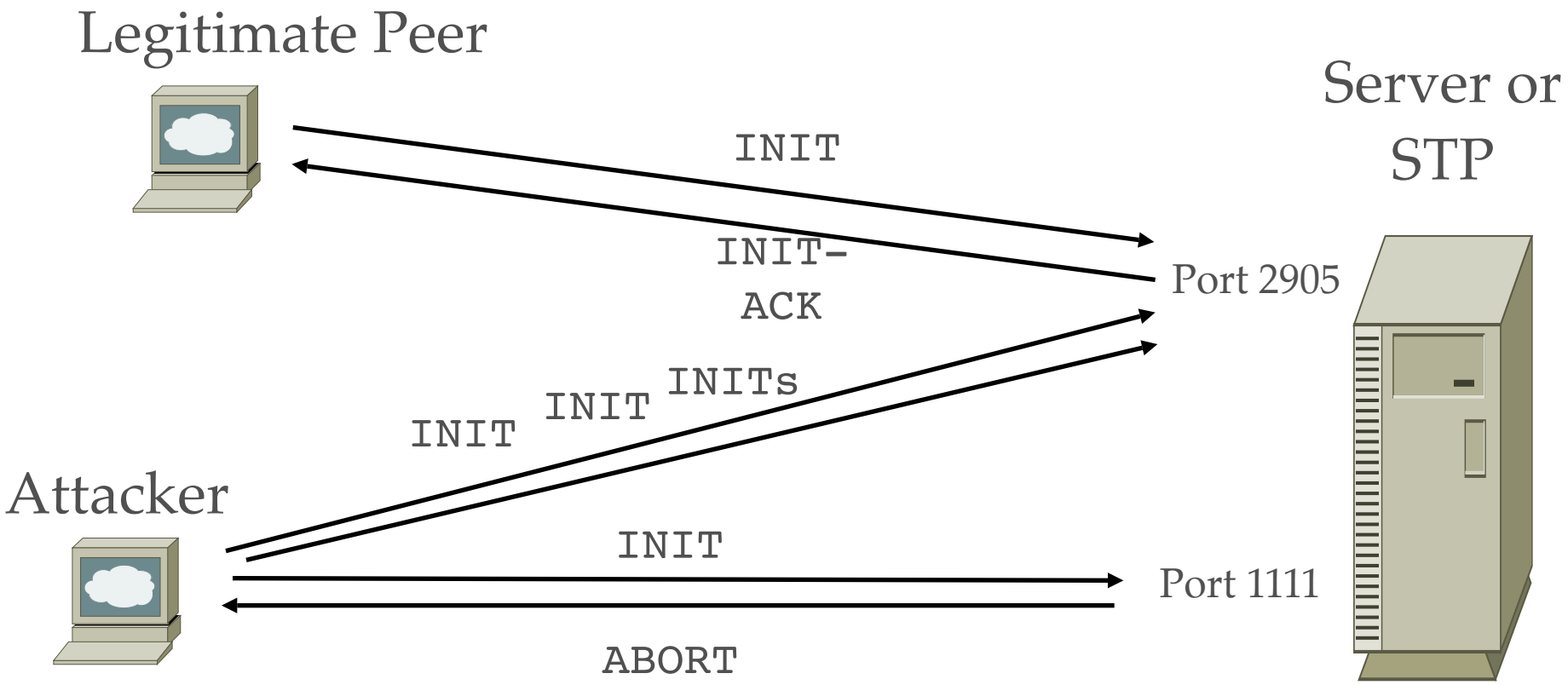


# SS7 Peering: attacker enemy




No answer on actual peering port: How rude!

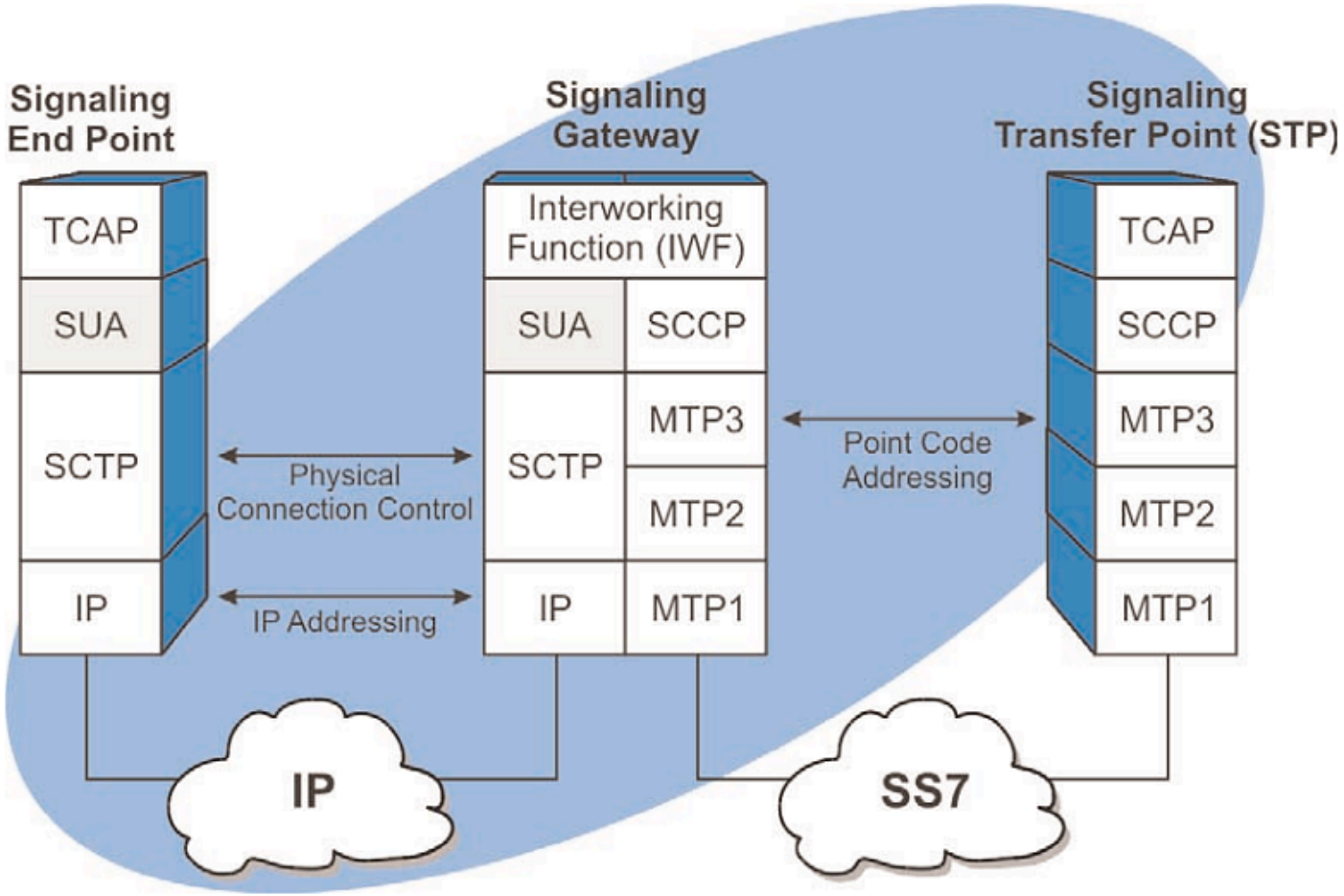
# SS7 Peering: attacker enemy




No answer on actual peering port: How rude!

On SS7 application attacks: hackers loose

# SCCP User Adaptation (SUA) Layer

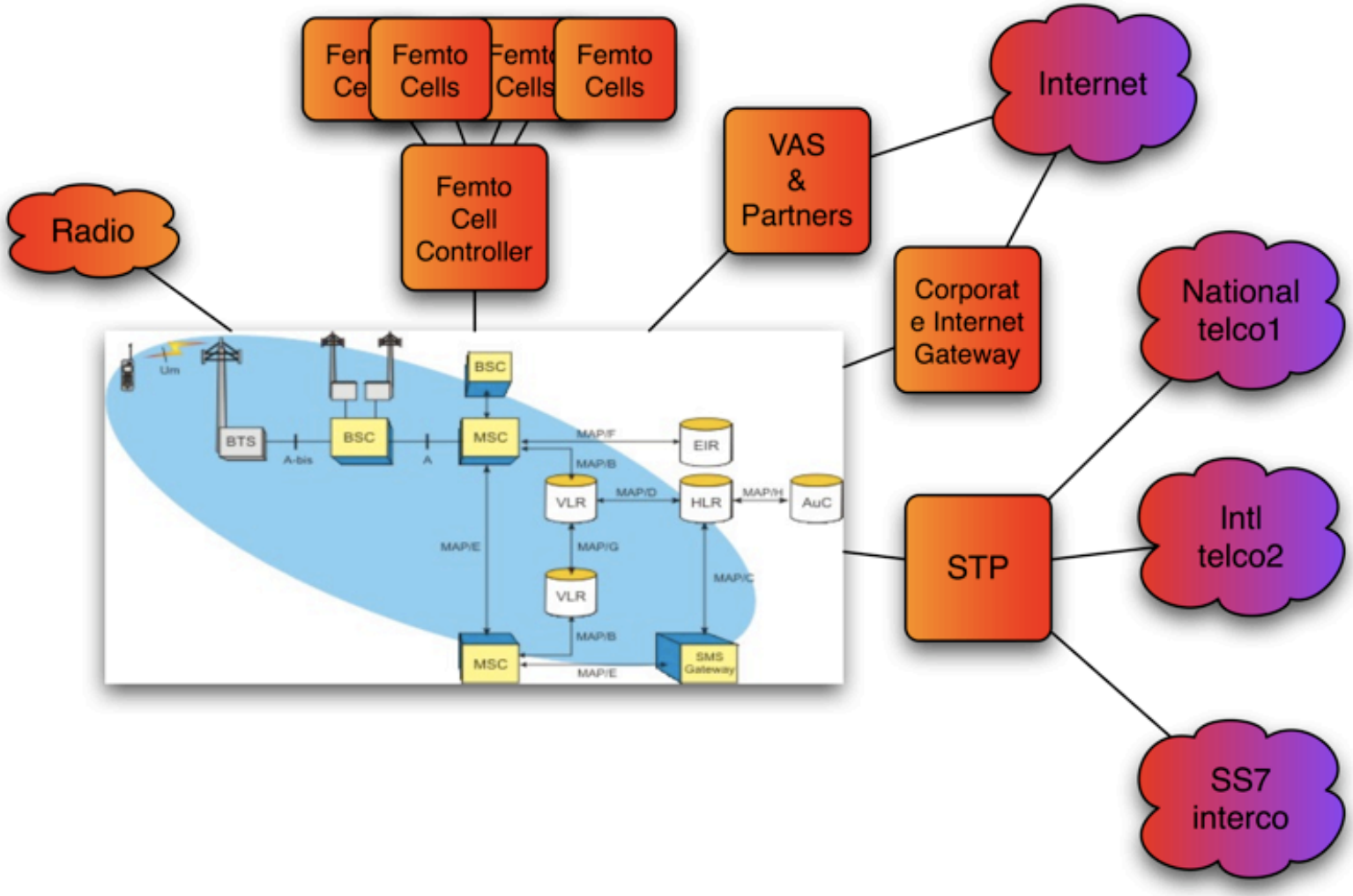


# Scanning the SS7 perimeter

SS7 protection methods and vulnerabilities
SS7 scanning and audit strategies

P1 Security Inc, http://www.p1security.com

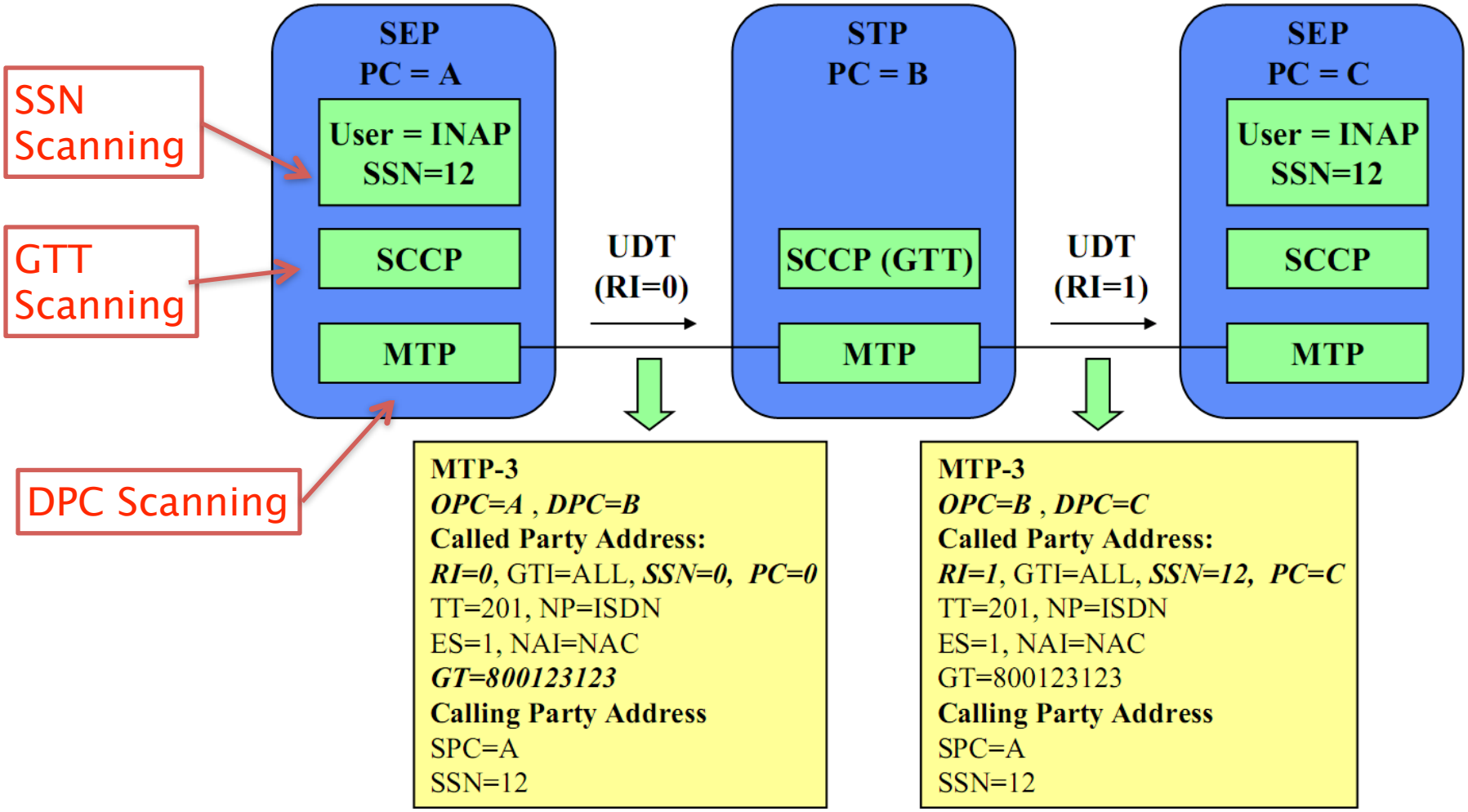
# SS7 Perimeter Boundaries



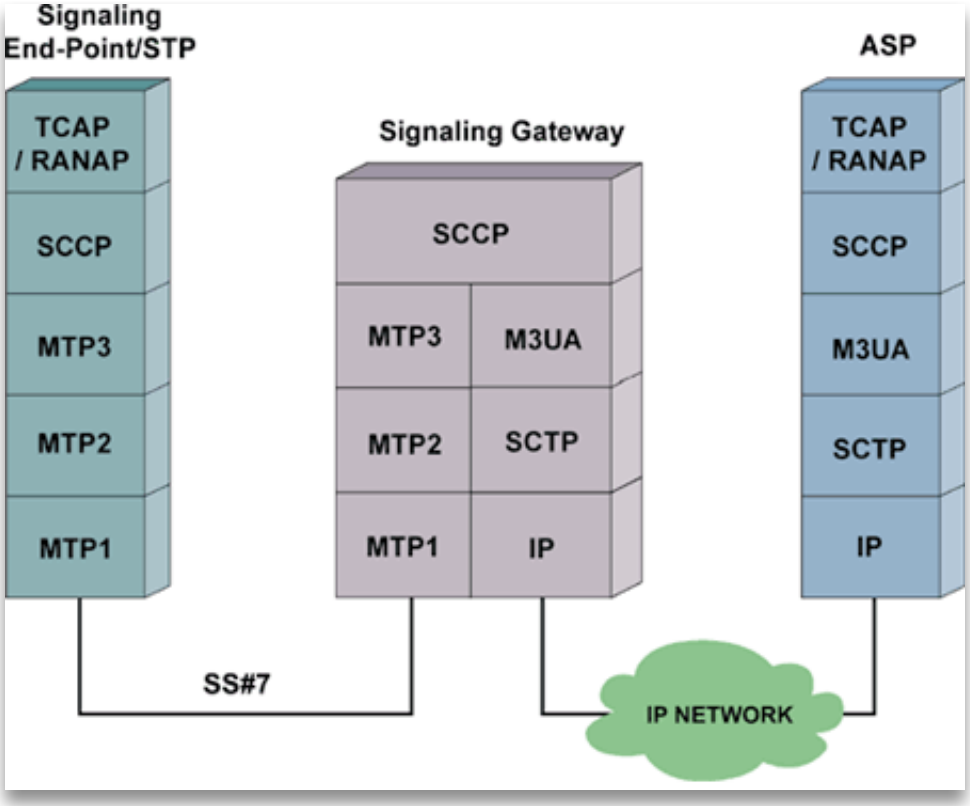
# STP as SCCP Firewall

- A “kind of” NAT
  - SubSystems allowed by STP, protection=route
  - SubSystem scanning & Message injection.
- NI (Network Indicator) Isolation
  - NI=0 : International 0, outside world
  - NI=2 : National 0, telco Internal
  - NI=3 : National 1, country-specific
- List of Signaling Point Code for each perimeter, automation needed.

# STP boundary: attacking SS7

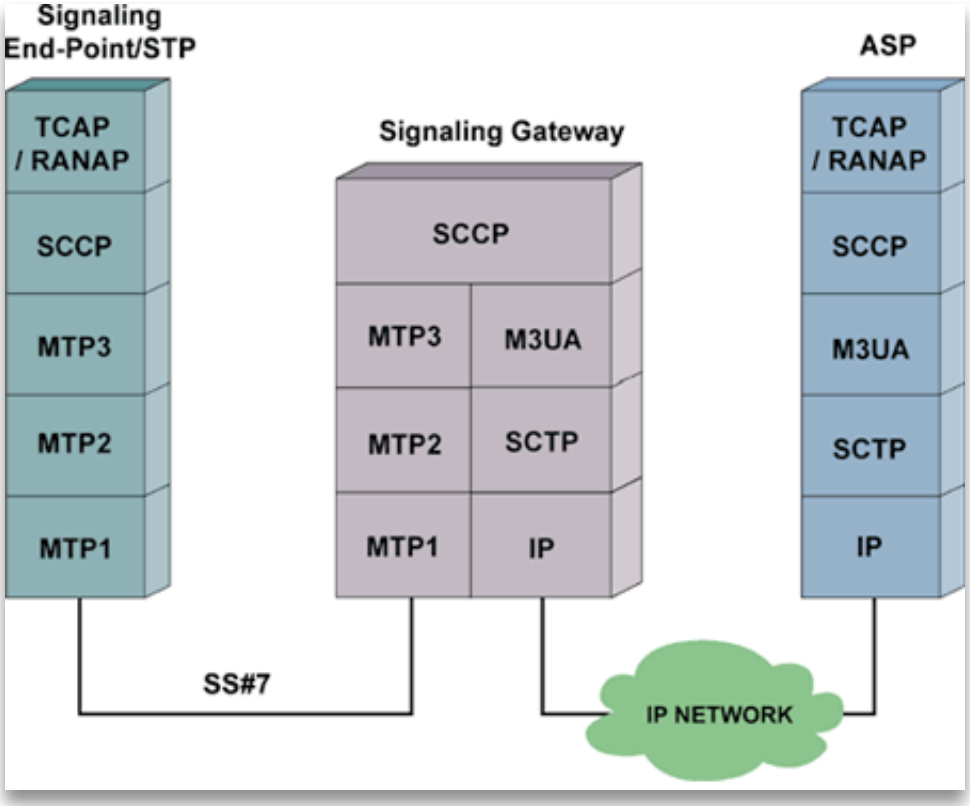


# Stack de-synchronization: more exposure & attacks




- Different stacks standardized by different people with different goals

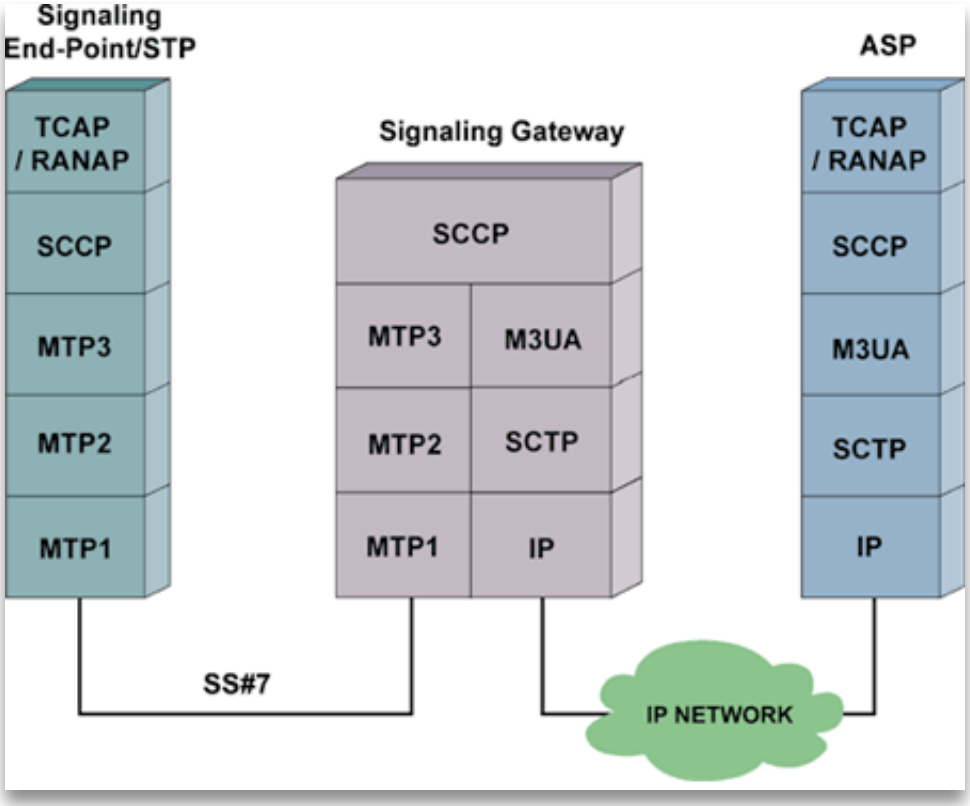
# Stack de-synchronization: more exposure & attacks




- Different stacks standardized by different people with different goals

→ SubSystem scanning

# Stack de-synchronization: more exposure & attacks

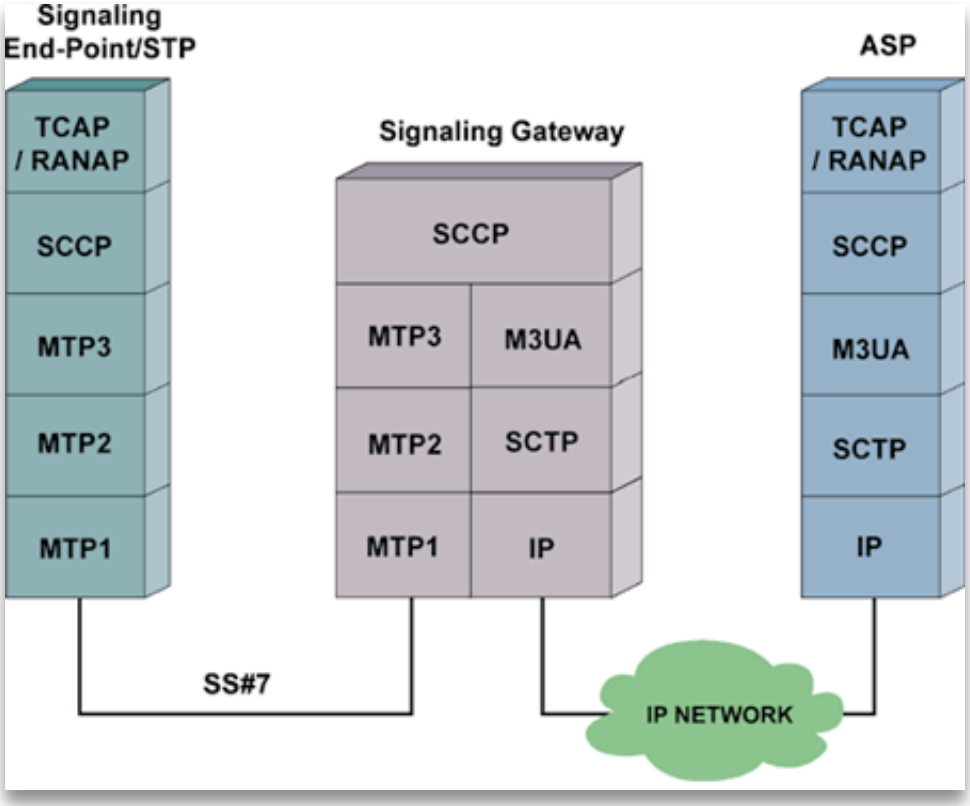



- Different stacks standardized by different people with different goals

→ SubSystem scanning

→ Topology discovery (needed for IP-based topologies)

# Stack de-synchronization: more exposure & attacks




- Different stacks standardized by different people with different goals

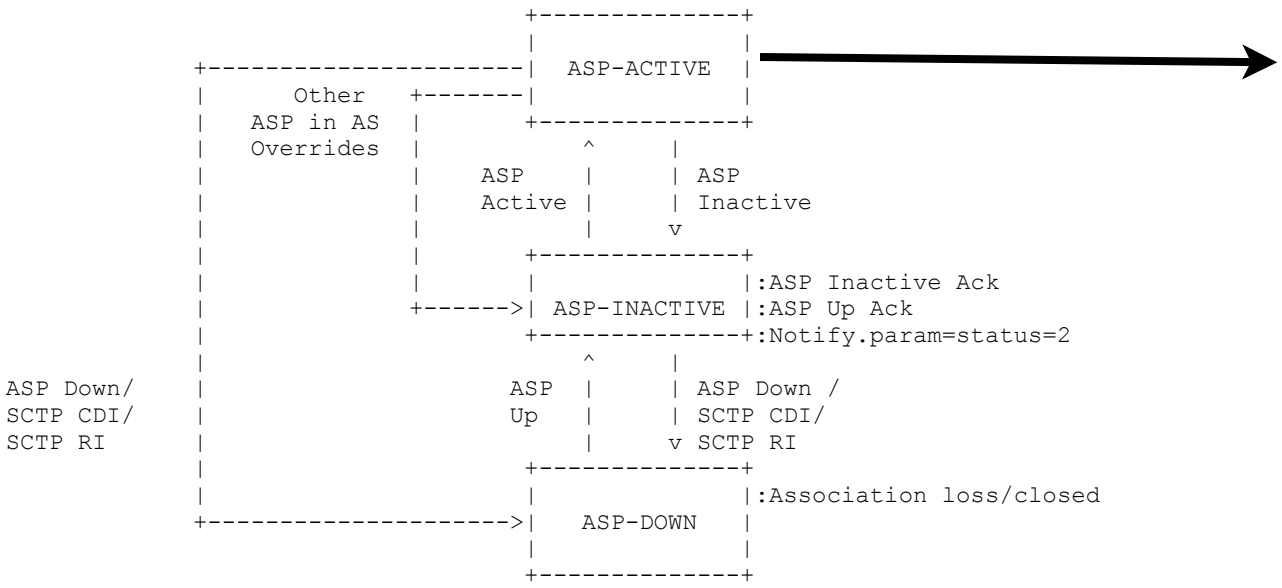
→ SubSystem scanning

→ Topology discovery (needed for IP-based topologies)

- Action available depends on State Machine's state
- Needs a special engine to inject attack at proper time/state

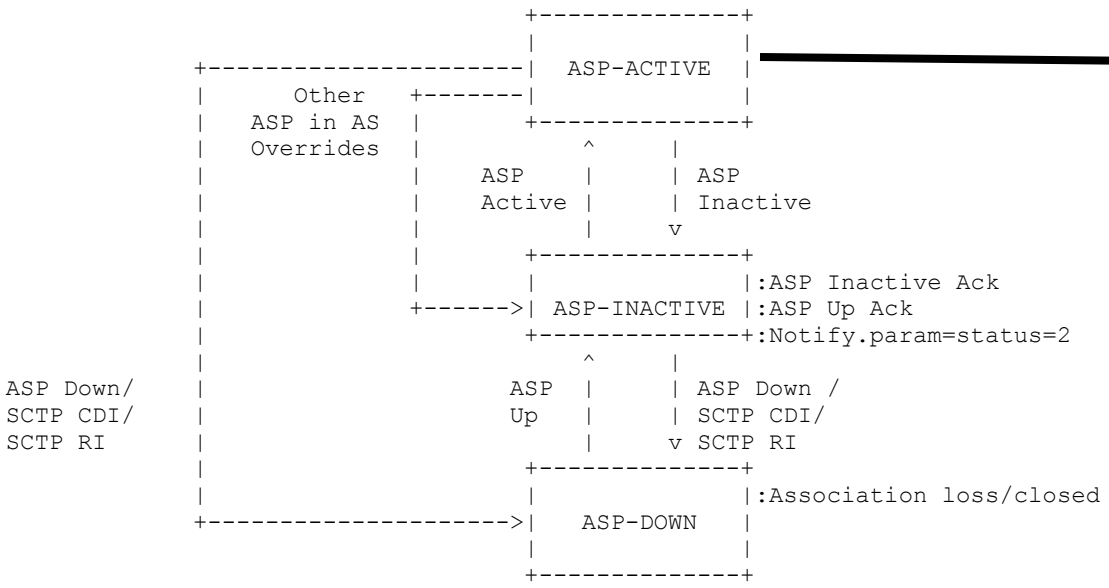
# M3UA Finite State Machine

Figure 3: ASP State Transition Diagram, per AS



# M3UA Finite State Machine

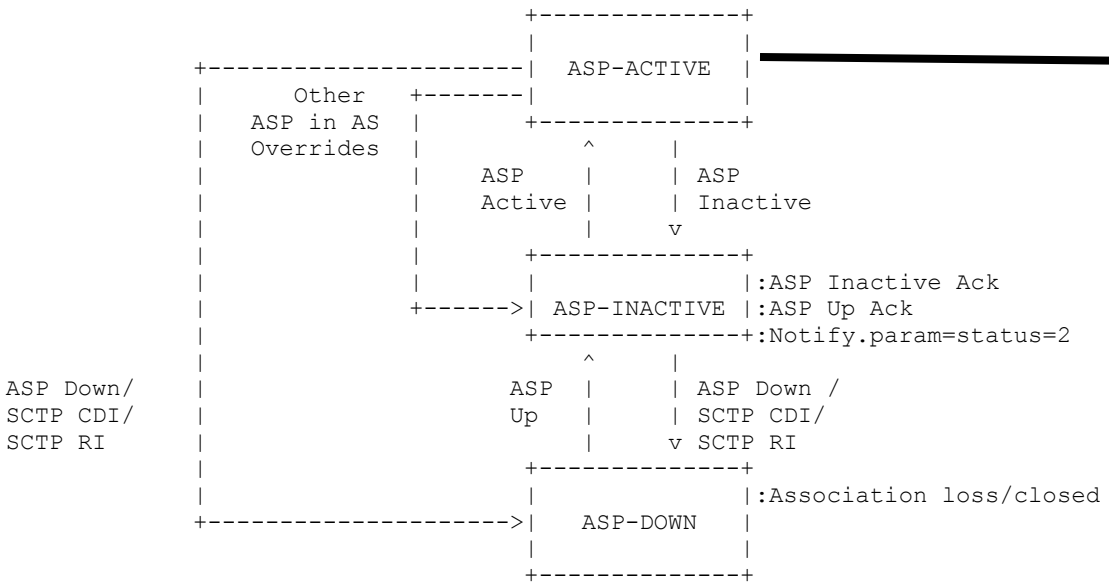
Figure 3: ASP State Transition Diagram, per AS




- M3UA test

# M3UA Finite State Machine

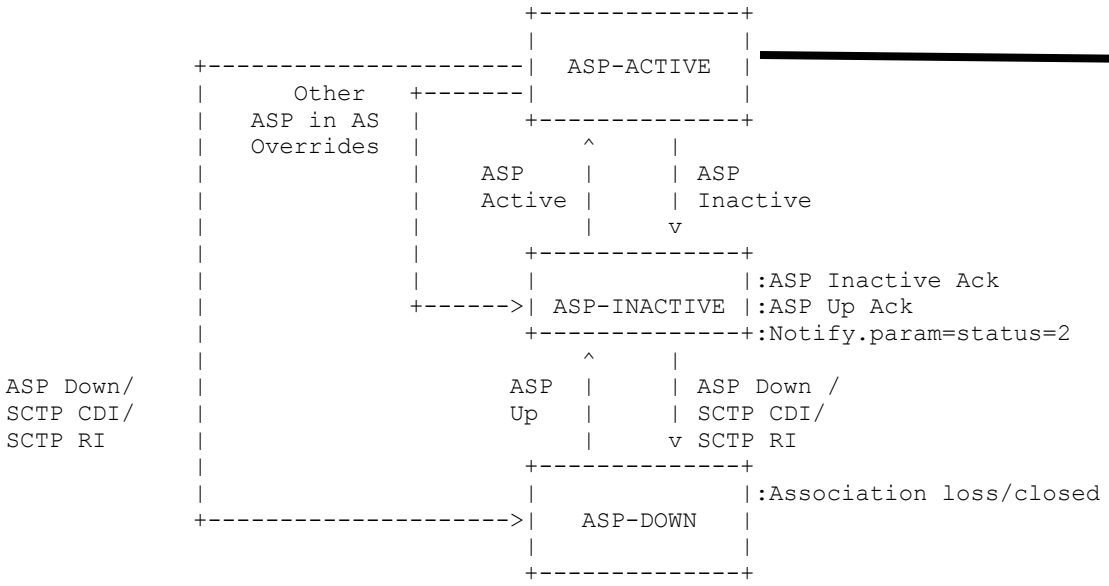
Figure 3: ASP State Transition Diagram, per AS




- M3UA test
- SCCP tests

# M3UA Finite State Machine

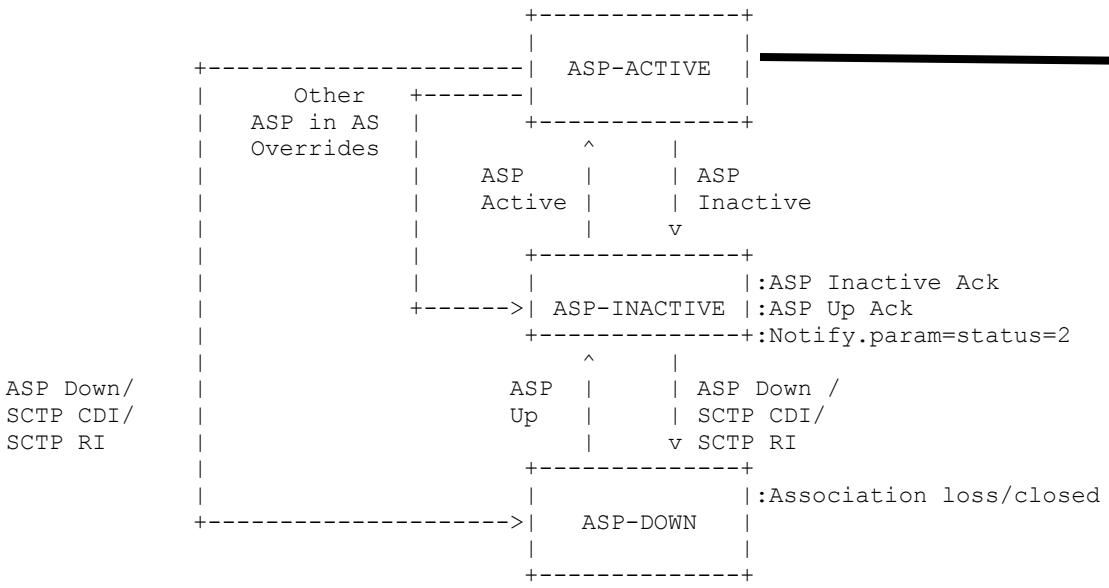
Figure 3: ASP State Transition Diagram, per AS




- M3UA test
- SCCP tests
- MAP tests

# M3UA Finite State Machine

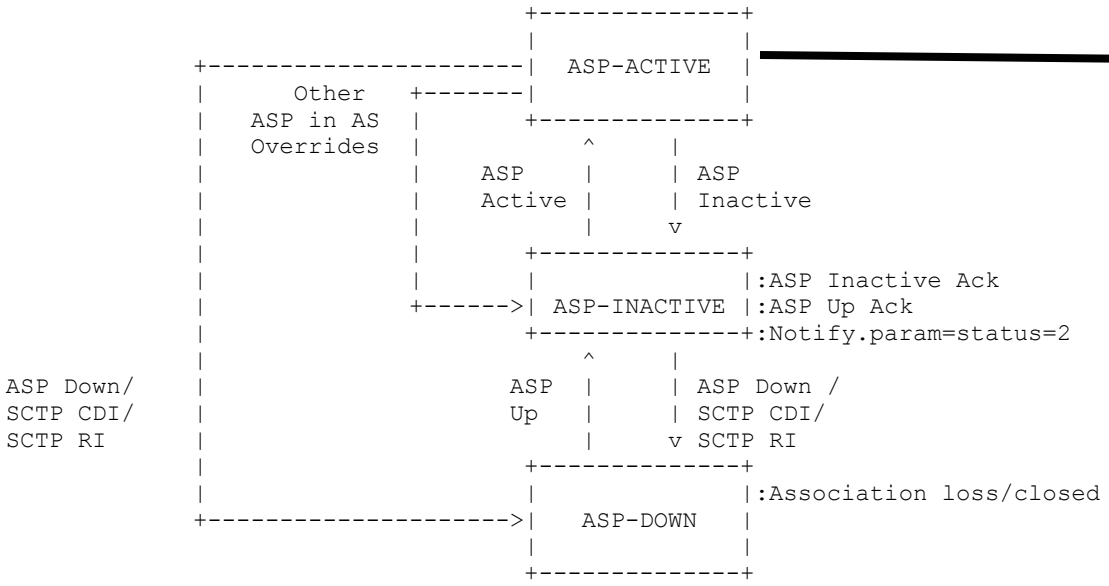
Figure 3: ASP State Transition Diagram, per AS




- M3UA test
- SCCP tests
- MAP tests
- INAP tests

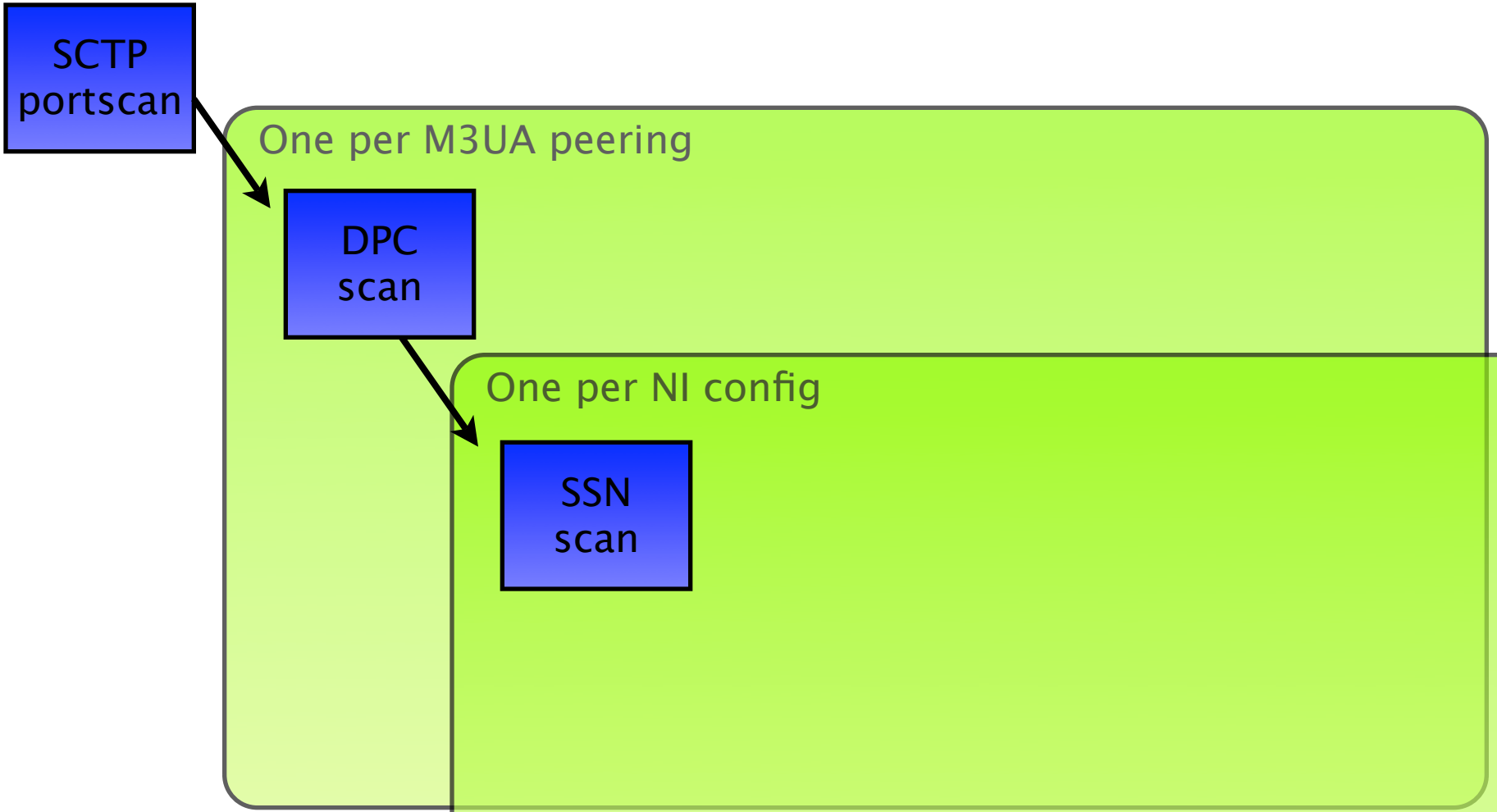
# M3UA Finite State Machine

Figure 3: ASP State Transition Diagram, per AS




- M3UA test
- SCCP tests
- MAP tests
- INAP tests
- Each depends on configuration

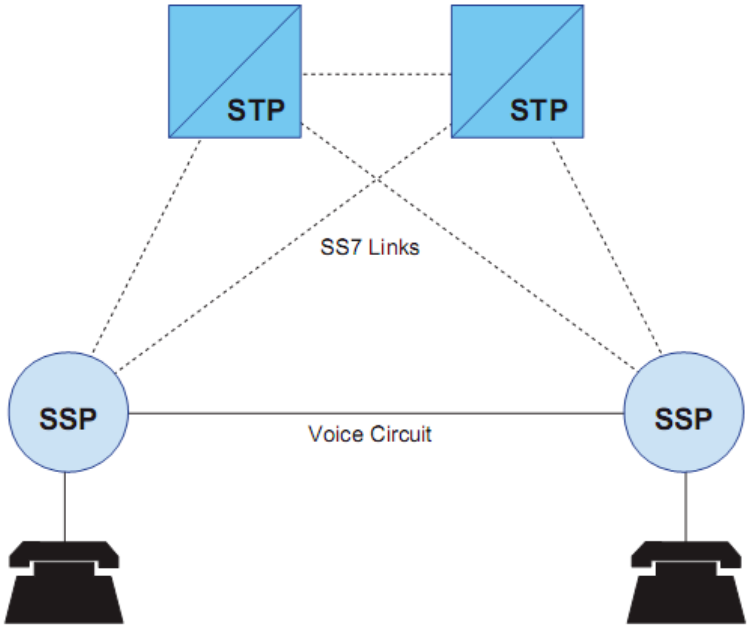
# SS7 Audit Strategies



# Example of SS7 protocol: ISUP & related attacks

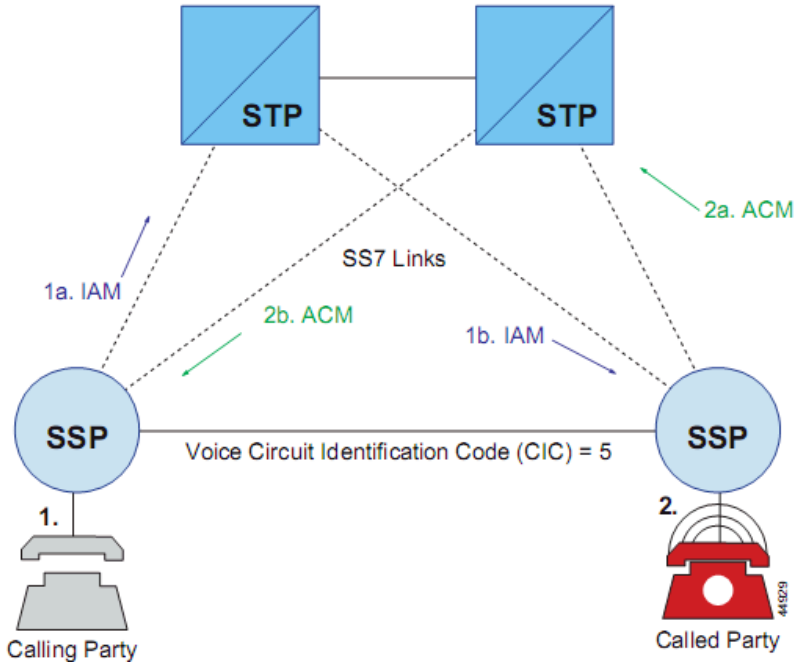
ISUP message types
ISUP call flows

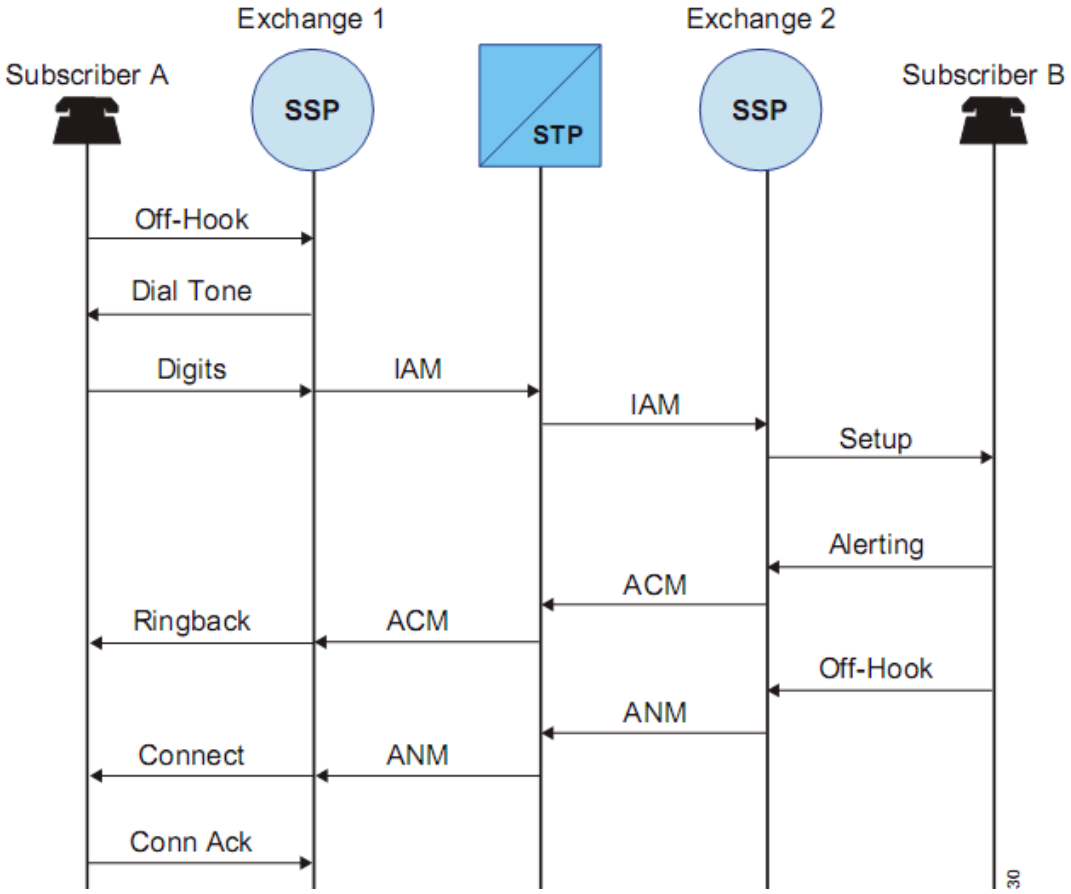
# ISUP message (ITU-T)




| Subservice Field | Service Indicator |
|---|---|
| DPC Low-Order Octet | |
| OPC Low-Order 2 bits | DPC High-Order 6-bits |
| OPC Middle-Order Octet | |
| 4-bit SLS/SLC | OPC High-Order 4-bits |
| CIC Low-Order Octet | |
| 4-bit SLS/SLC | CIC High-Order 4-bits |
| Message Type | |
| Interpretation varies according to Message Type variable | |

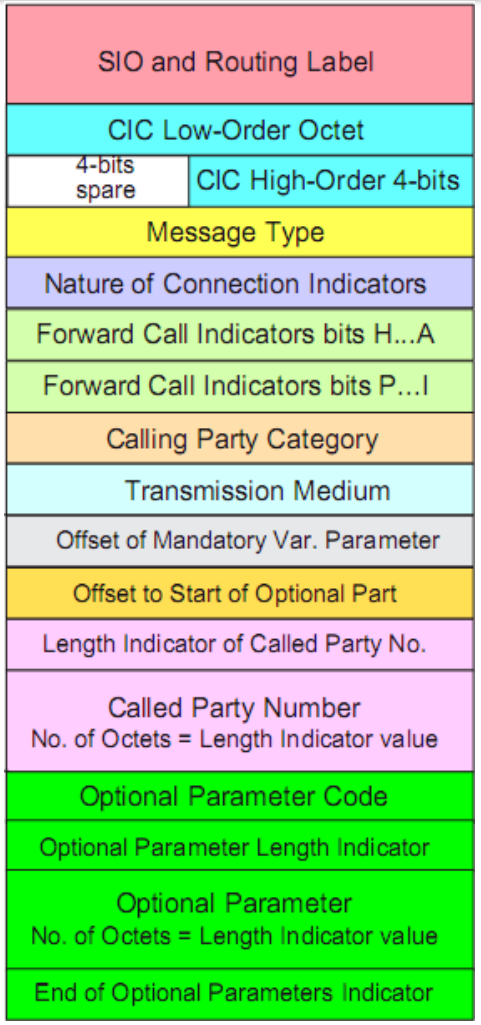
# ISUP Call Initiation Flow



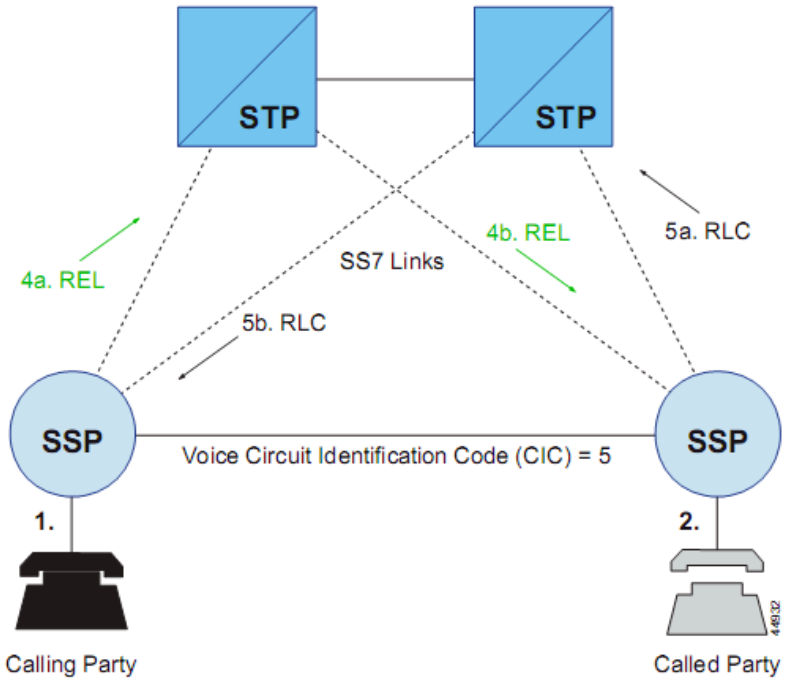



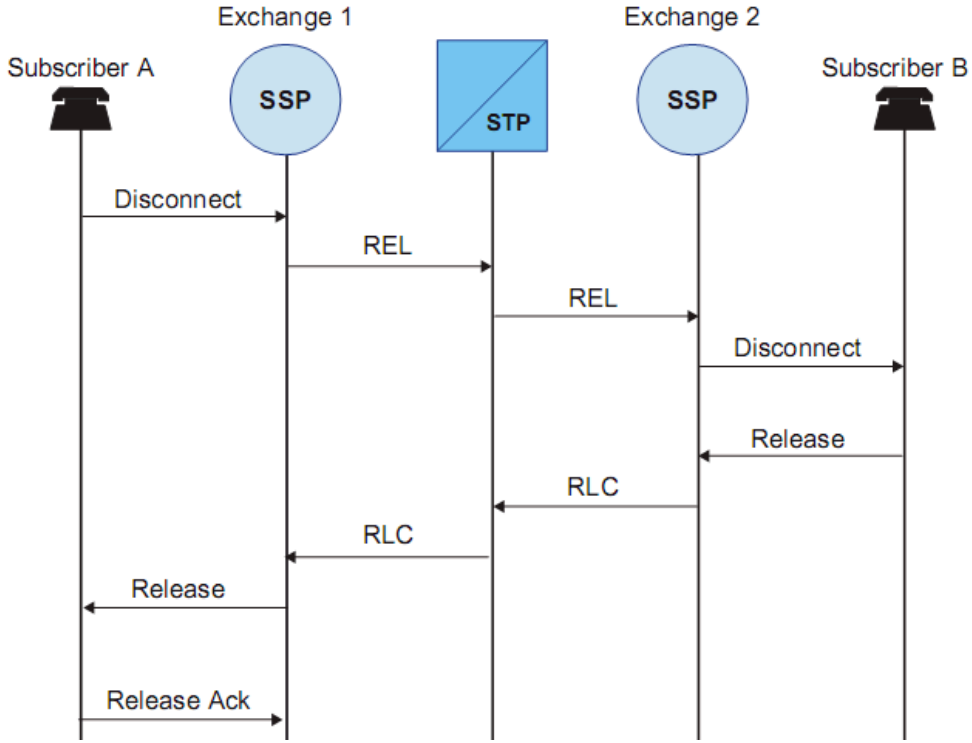
# ISUP AIM

- An **initial address message** (IAM) is sent in the "forward" direction by each switch in the circuit between the calling party and the destination switch of the called party.
- An IAM contains the **called party number** in the mandatory variable part and may contain the **calling party name** and number in the optional part.
- **Attack: Capacity DoS**



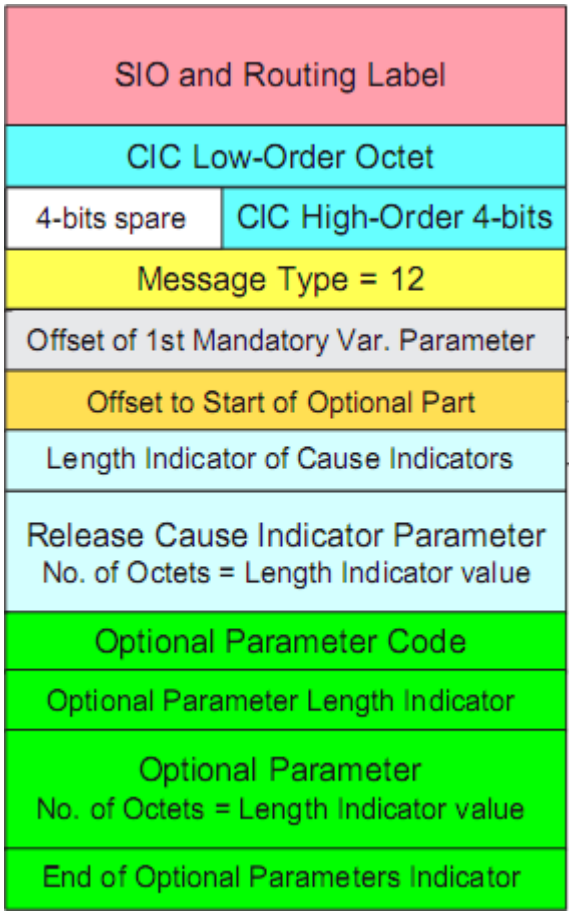
# ISUP Call Release Flow



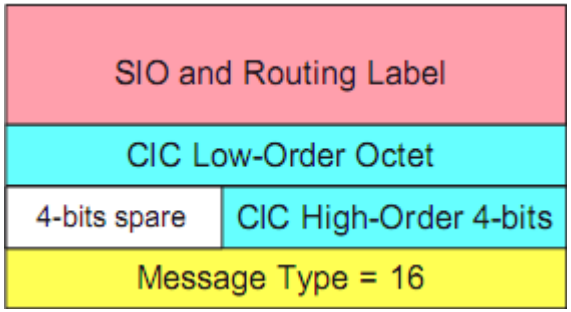
# ISUP REL

- A **release message** (REL) is sent in either direction indicating that the circuit is being released due to a specified cause indicator.
- An REL is sent when either calling or called party **hangs up** the call (cause = 16).
- An REL is also sent back to the calling party if the called party is **busy** (cause = 17).
- **Attack: Selective DoS**



# ISUP RLC

- A **release complete message** (RLC) is sent in the opposite direction of an REL to acknowledge the release of the remote end of a trunk circuit and to end the billing cycle, if appropriate.



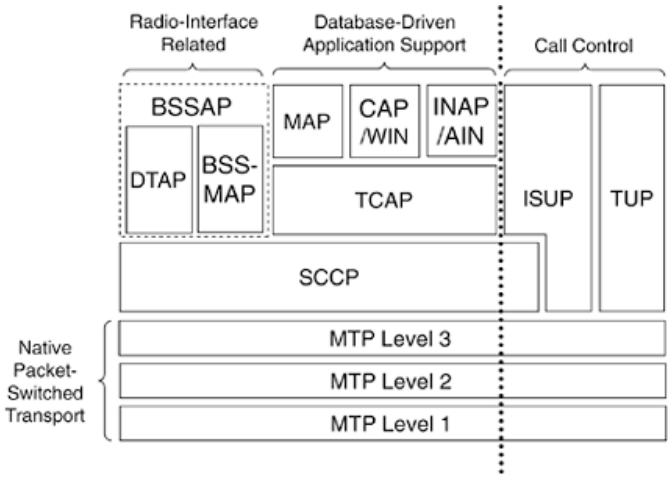
# A Practical SS7 Information Gathering

Send Routing Info or monitoring anyone with a phone, anywhere...

P1 Security Inc, http://www.p1security.com

# Geolocation & Information Gathering

- SS7 MAP message: SendRoutingInfo (SRI)
- Sends back the **MSC in charge**. Correlates to country.
- Nobody knows i'm not an HLR.
- **Real world usage: Identification for SPAM, 150 EUR for 10k, HTTP APIs & GW**
- **Attack: Global tracking and geolocation of any phone**



# A practical SS7 attack

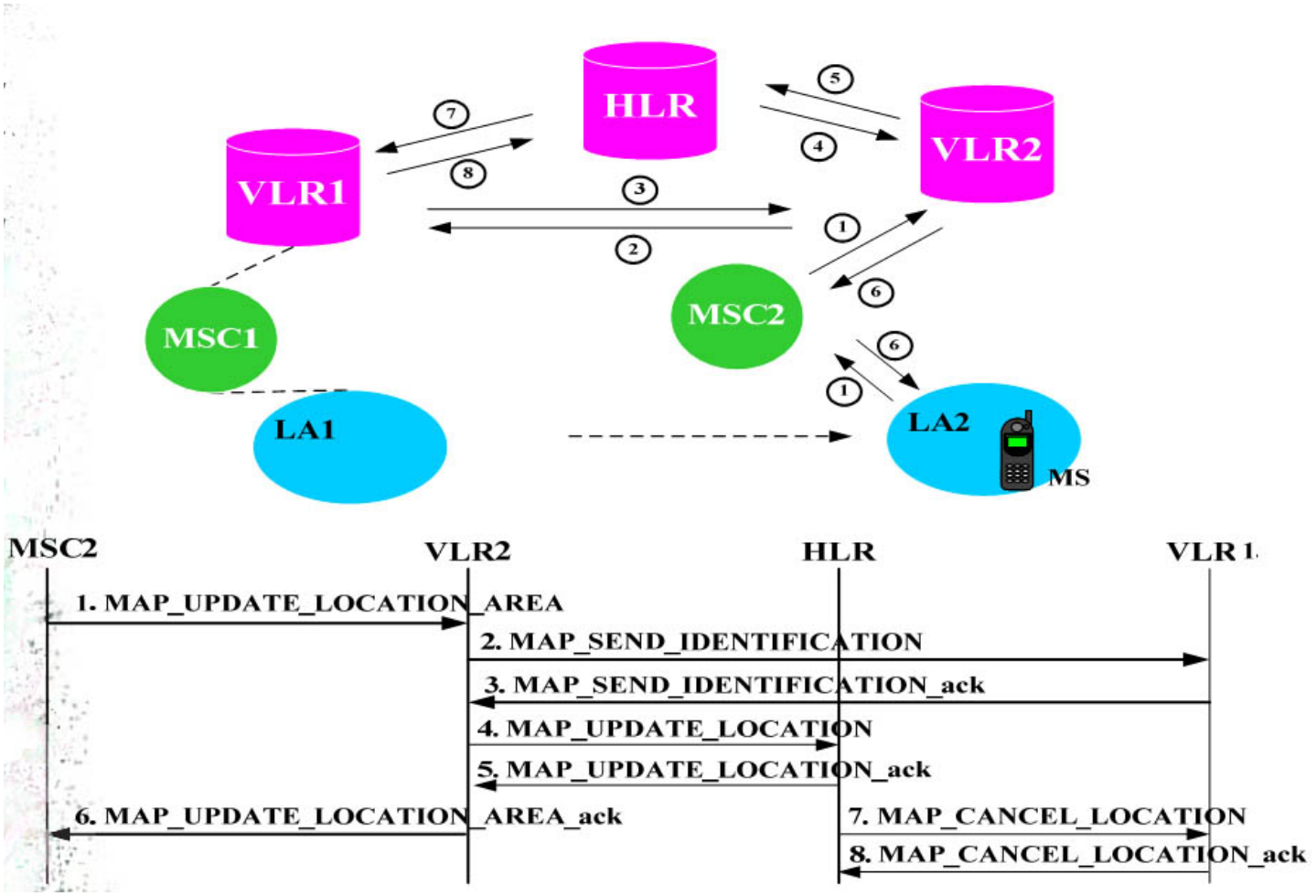
Disabling incoming calls to any subscriber

P1 Security Inc, http://www.p1security.com

# Location Update process

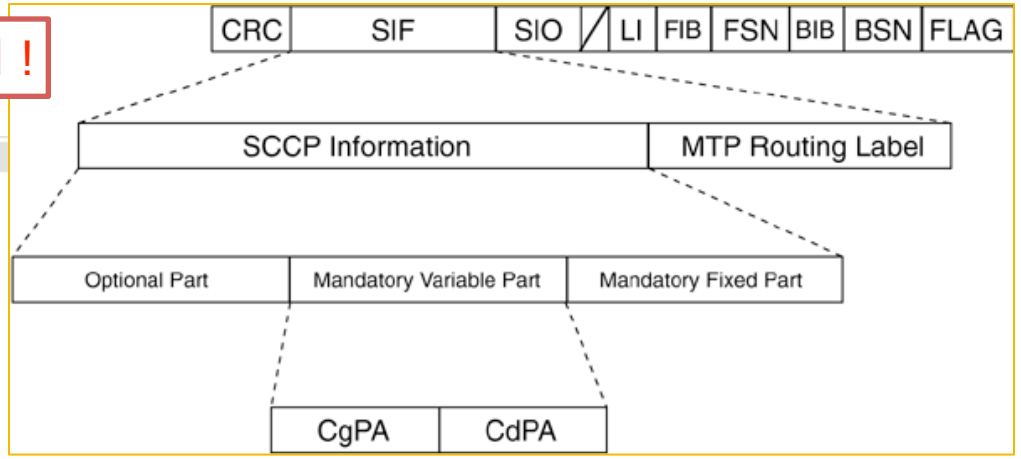
- The MAP **updateLocation (UL)** message contains subscriber's IMSI and MSC/VLR addresses.
- Once UL reaches the HLR, it changes the serving MSC/VLR address in subscriber's profile using MAP **insertSubscriberData** messages.
- From then on the HLR will use MSC/VLR addresses from it as addresses of real MSC/VLR.
- It's not even necessary to complete whole UL-ISD-ISDack-ULack transaction!
- The HLR will complete the operation by sending a MAP **cancelLocation** message to the serving VLR to delete subscriber's information from it.

# Location Update Call Flow



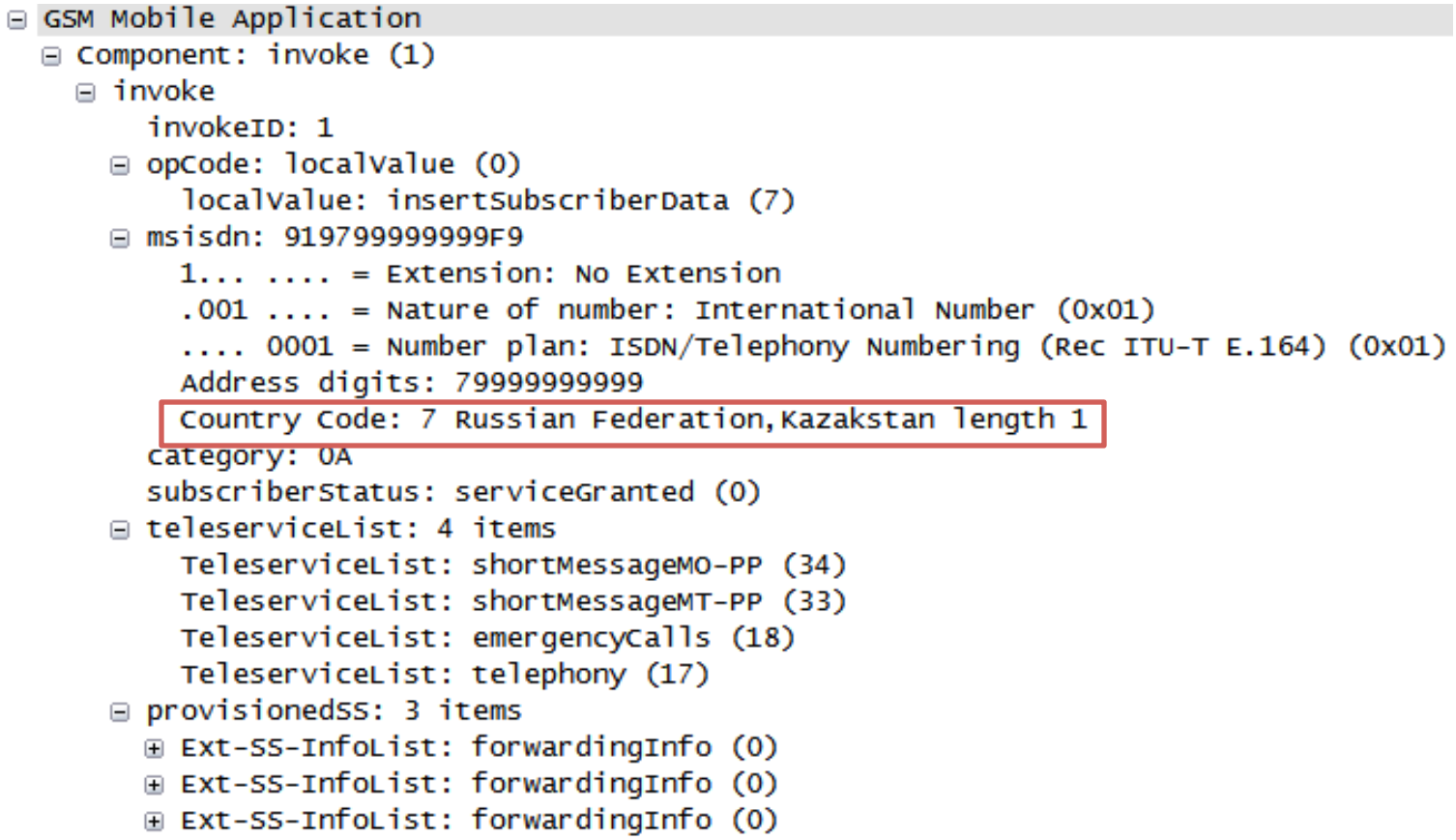
# Attack implementation

IMSI scanning / querying needed !




```
GSM Mobile Application
  Component: invoke (1)
    invoke
      invokeID: 1
      opCode: localValue (0)
        localValue: updateLocation (2)
      imsi: 52009299999999F9
      TBCD digits: 250029999999999
      msc-Number: 91839099999999
        1... .... = Extension: No Extension
        .001 .... = Nature of number: International Number (0x01)
        .... 0001 = Number plan: ISDN/Telephony Numbering (Rec ITU-T E.164) (0x01)
        Address digits: 380999999999
        Country Code: 380 Ukraine length 3
      vlr-Number: 91839099999999
        1... .... = Extension: No Extension
        .001 .... = Nature of number: International Number (0x01)
        .... 0001 = Number plan: ISDN/Telephony Numbering (Rec ITU-T E.164) (0x01)
        Address digits: 380999999999
        Country Code: 380 Ukraine length 3
      vlr-Capability
        Padding: 4
        supportedCamelPhases: C0 (phase1, phase2)
        Padding: 4
        supportedLCS-CapabilitySets: F0 (lcsCapabilitySet1, lcsCapabilitySet2, lcsC
```

# Attack success



# 3G: New threat perimeters

The walled garden is opening up...

# Femto Cell & user control

- Node B in user home, IPsec tunnel, SIGTRAN
- Real world example: ARM hw with RANAP
- Insecure
  - Untested hw
  - Unprotected IPsec
  - No regular pentest
  - No tools! Need for Binary vulnerability audit

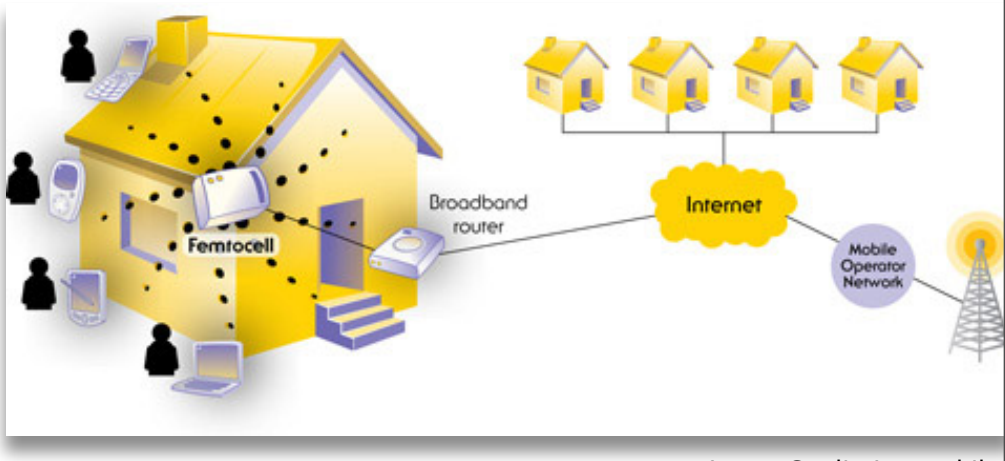



Image Credit: Intomobile

# Femto-cell attack vectors

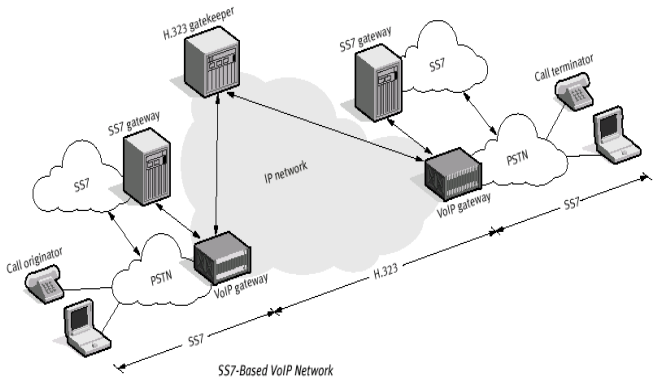
- Unaudited Proprietary software from Alcatel
  - Attack: **Binary vulnerability audit gives 0day**
  - Attack: **Vulnerable Linux 2.6 kernel**
- Global settings for IPsec tunnels
  - Attack: **Border access**
- Lack of SS7 and SIGTRAN filtering
  - Attack: **Injection of RANAP and SS7 in the Core Network**

# Injecting SS7 through SIP

New perimeters, new entry points, new threats

# SIP to SS7 ?

- SIP is used to connect two SS7 cloud
- Support to bridge SS7 context through SIP
- SIP injection of SS7 adds a header to standard SIP headers
  - New SS7 perimeter, even for non-telco



# Getting secure...

How to secure an insecure network being more and more exposed?

# Tools and methods

# Tools and methods

- Manual SS7 audit & pentest (hard!)

# Tools and methods

- Manual SS7 audit & pentest (hard!)

# Tools and methods

- Manual SS7 audit & pentest (hard!)
- P1security SIGTRANalyzer to audit perimeters
  - SS7 interconnect, Value Added Services
  - Core Network
  - Femto Cell access network
  - SIP & Convergent services

# Tools and methods

- Manual SS7 audit & pentest (hard!)
- P1security SIGTRANalyzer to audit perimeters
  - SS7 interconnect, Value Added Services
  - Core Network
  - Femto Cell access network
  - SIP & Convergent services
- Customer Acceptance Testing : equipment reverse engineering and binary auditing.

# Current developments

- SCTPscan
  - Bridging support, instream scanning
  - Open source
- ss7calc
  - Like ipcalc (FLOSS), to understand network topology
  - Complexity: ITU: 3-8-3, 5-4-5, ANSI: 8-8-8
- SIGTRANalyzer
  - SS7 and message injection audit, information gathering, leak analysis,
  - Commercial product

# Conclusions

- SS7 is not closed anymore
- Industrializing the solution
  - From pentest to continuous testing (hardware and operations)
  - Security services and products
- Mindset are changing: more open to manage the SS7 security problem.

# Credits

- Key2, Emmanuel Gadaix, Telecom Security Task Force, Fyodor Yarochkin
- Bogdan Iusukhno
- Skyper and the THC SS7 project
- All the 7bone security researchers


- CISCO SS7 fundamentals, CISCO press
- Introduction to SS7 and IP, by Lawrence Harte & David Bowler
- Signaling System No. 7 (SS7/C7) – Protocol, Architecture and Services, by Lee Dryburgh, Jeff Hewett

# THANKS!

- Questions welcome
- Philippe Langlois, phil@p1sec.com
- More slides on http://www.p1security.com