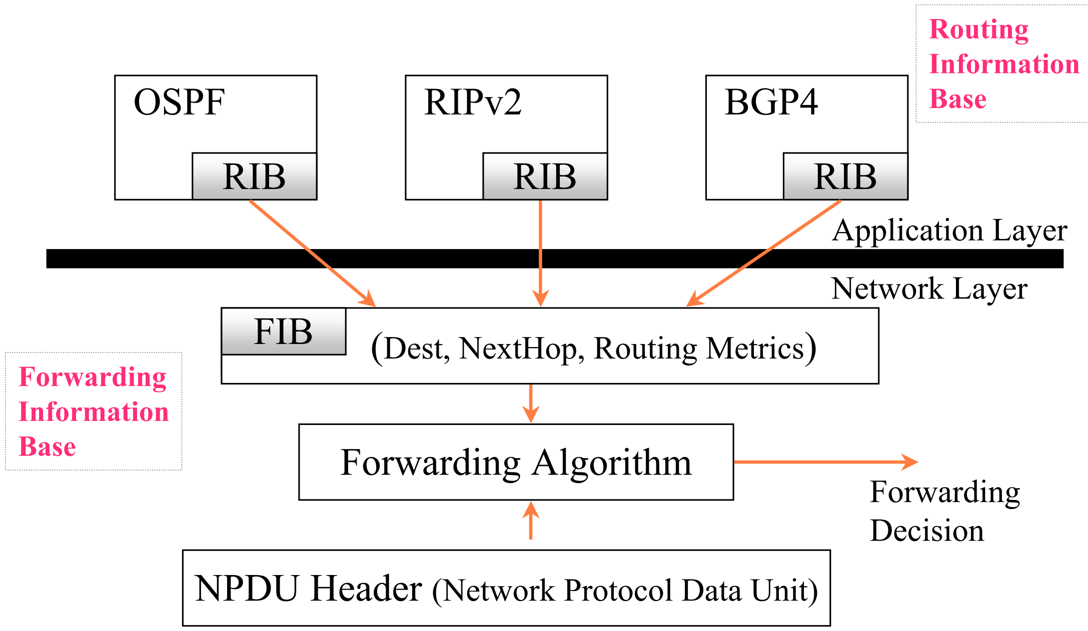
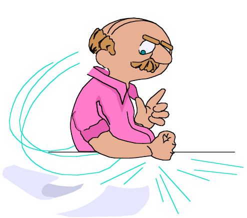
Routing Protocol Framework
Information Model

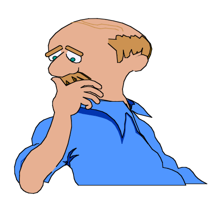


Operation Model
Routing Information Exchange

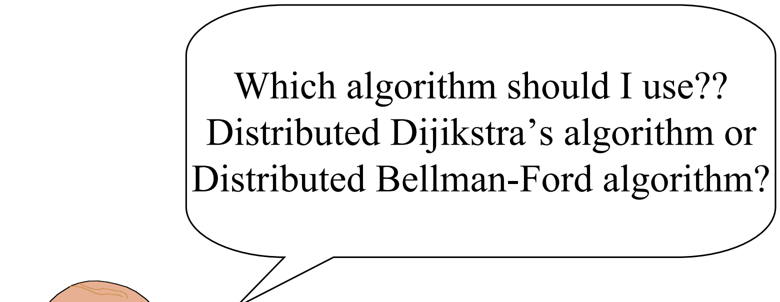


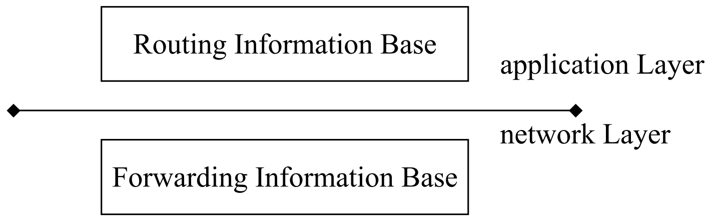
Hey, Here is the routing information I got so far

Hmm, some of them are obsolete, Here is my update

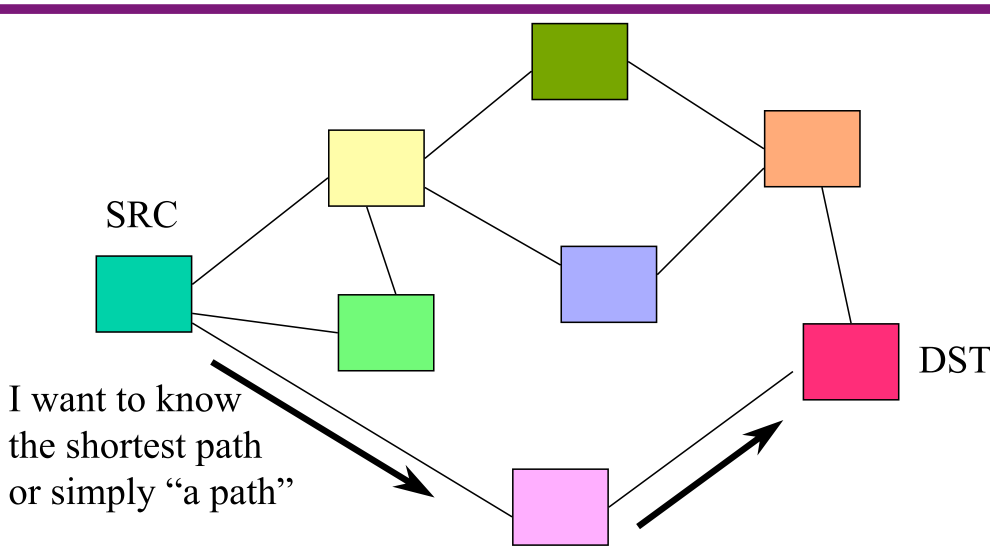


Operation Model
Route Generation and Selection



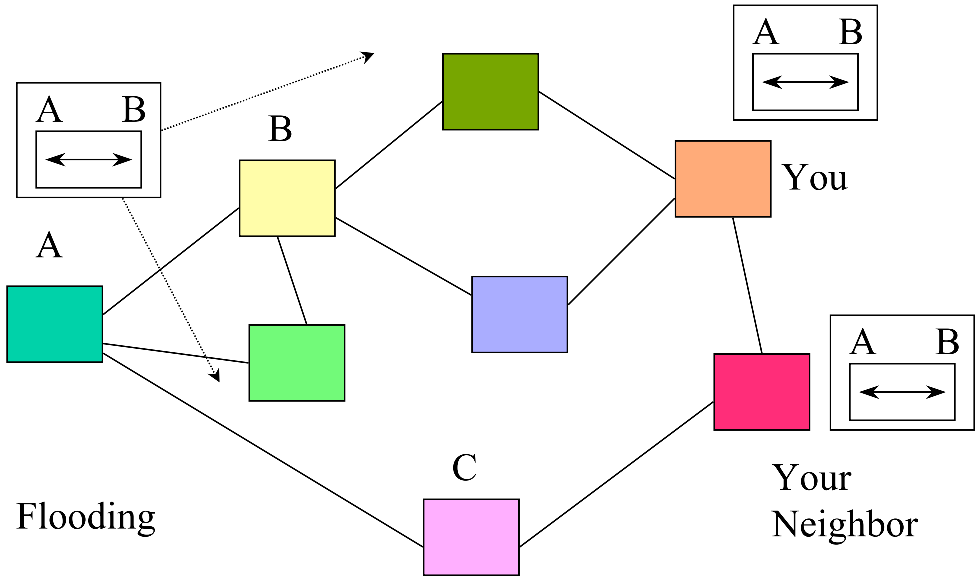


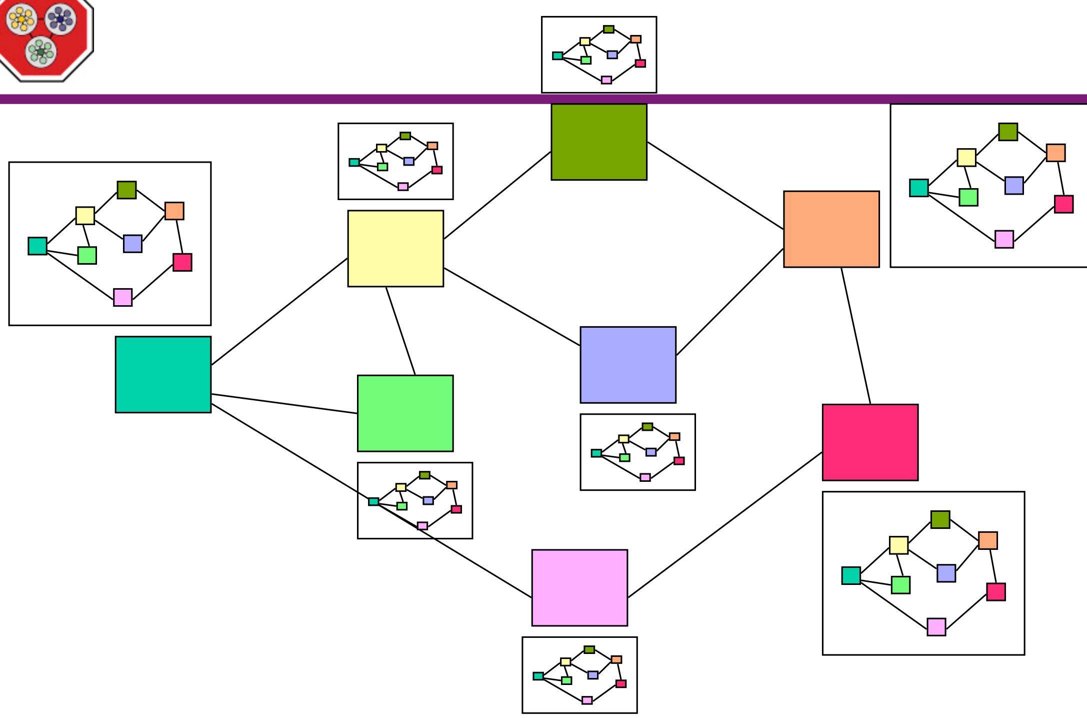
Routing



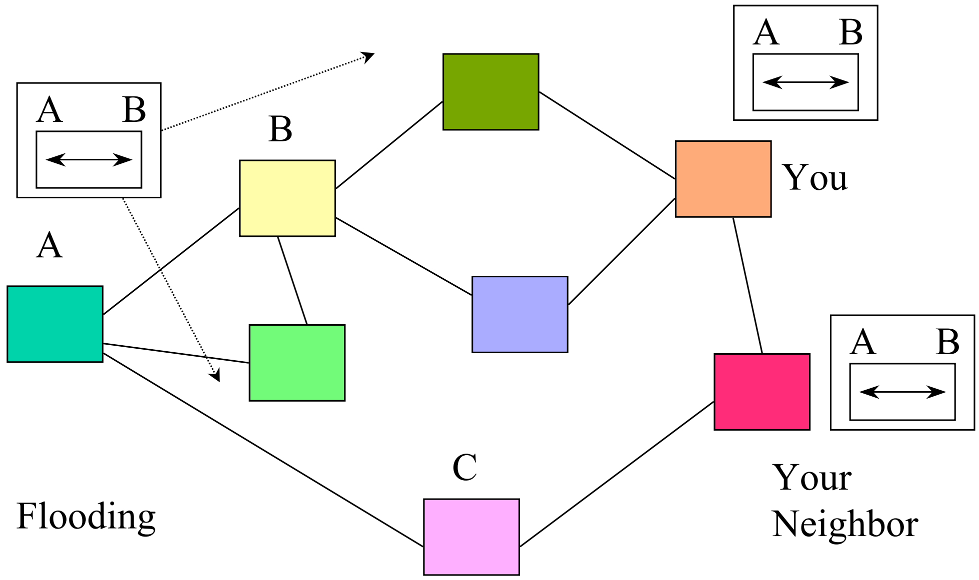
Routers exchange local information!

Link State





Link State

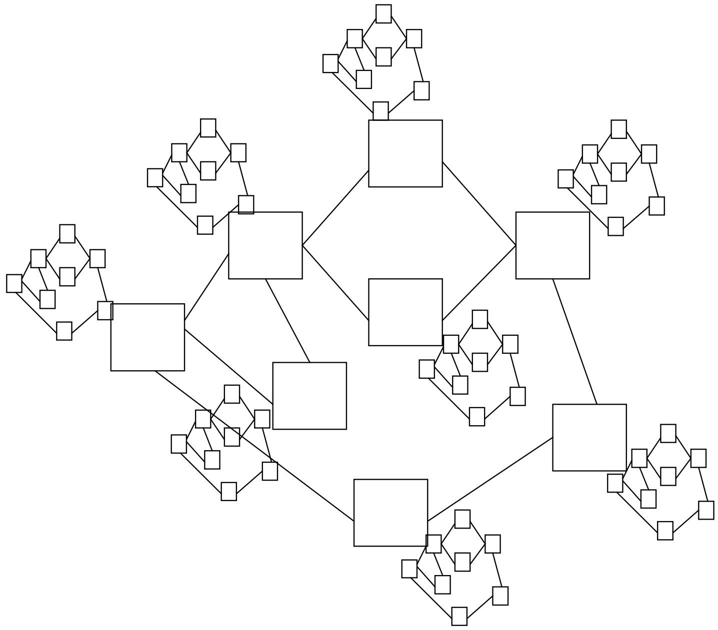


You tell the whole world about your relationship with your neighbor

Routing Information

- Link State:
  - I let the whole world knows about my relationship with my neighbors.
  - (Felix, Neighbor-X) is up!
- Distance Vector:
  - I let all my neighbors knows about my relationship with the rest of the world.
  - (Felix can get to Remote-Y) in 5 hops.

Link-State



LSA and an **LSA instance**

- An **LSA** is associated with a particular link of network, which is identified by its ***LS type, LS ID, Advertising Router ID***.
- An **LSA instance** gives the state of a particular LSA at a particular time, which can be differentiated by ***LS sequence number, LS age, LS checksum***.

0x80000000 0x80000001 → 0x7FFFFFFF

LSA Format

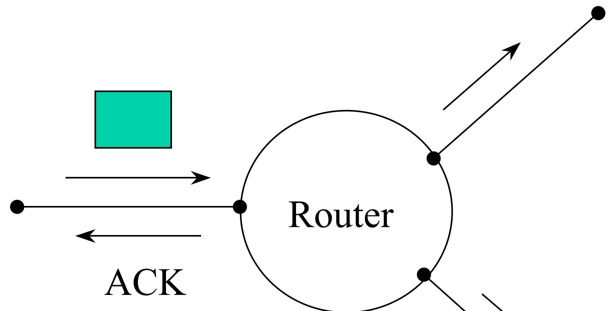
- Type (Hello, Link, Networ, Summary)
- Advertizing Router ID (Originator)
- Advertized Link or Network.
- Sequence Number
  - smallest: 0x 80000001
  - largest: 0x7FFFFFFF
- Age (0, 60 minutes)

RIB - OSPF

LSA-ID	ADV	Seq#	Checksum	Age
A=B	A	0x850012a7	0x452b	20:13
A=B	B	0x84230b41	0x3729	13:12
A=D	A	0x9012000e	0x2567	01:22

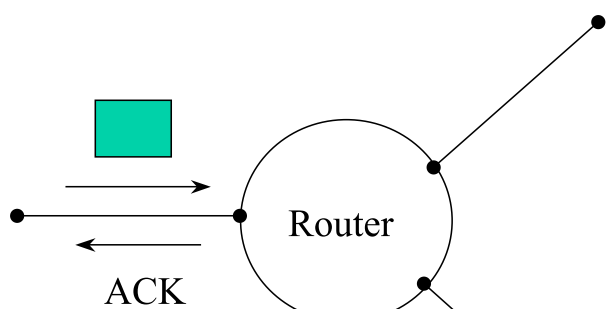
…

OSPF LSA Flooding, I



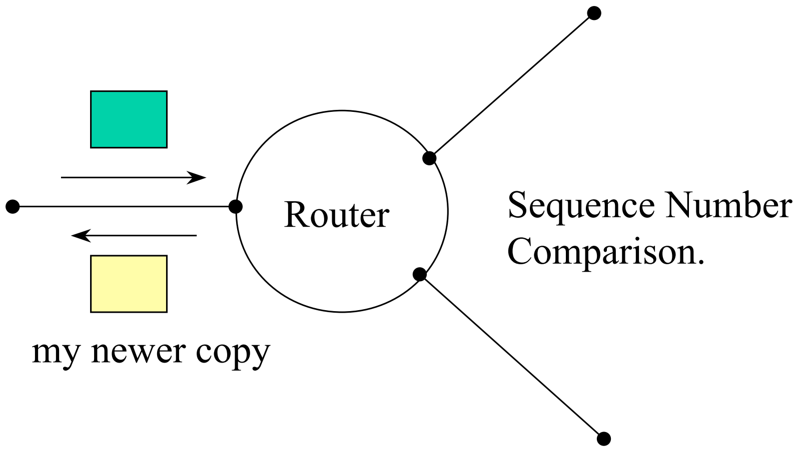
If I have not received this LSA (Link State Advertisement).

OSPF LSA Flooding, II



If I have received this LSA (Link State Advertisement).

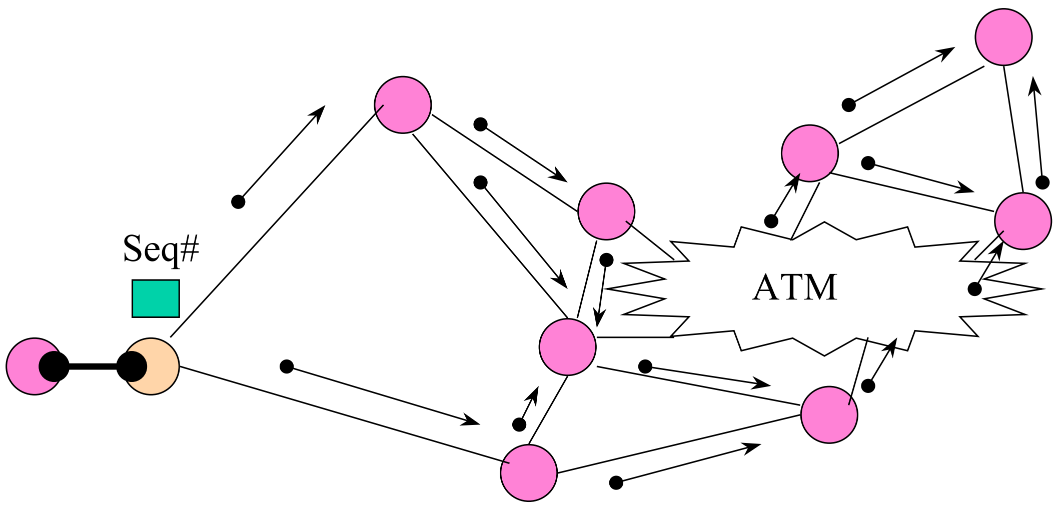
OSPF LSA Flooding, III



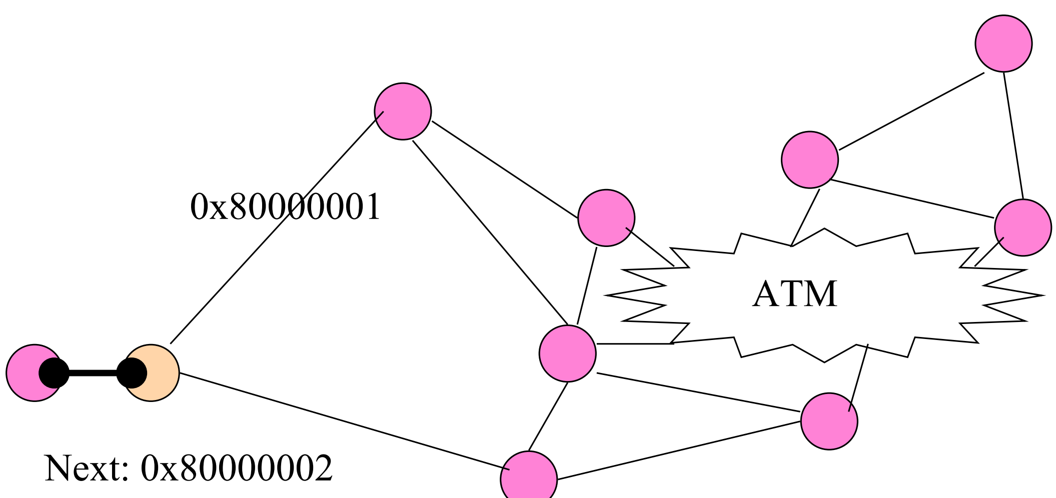
If I have something
better/fresher/newer..

How to decide "freshness"??

Sequence

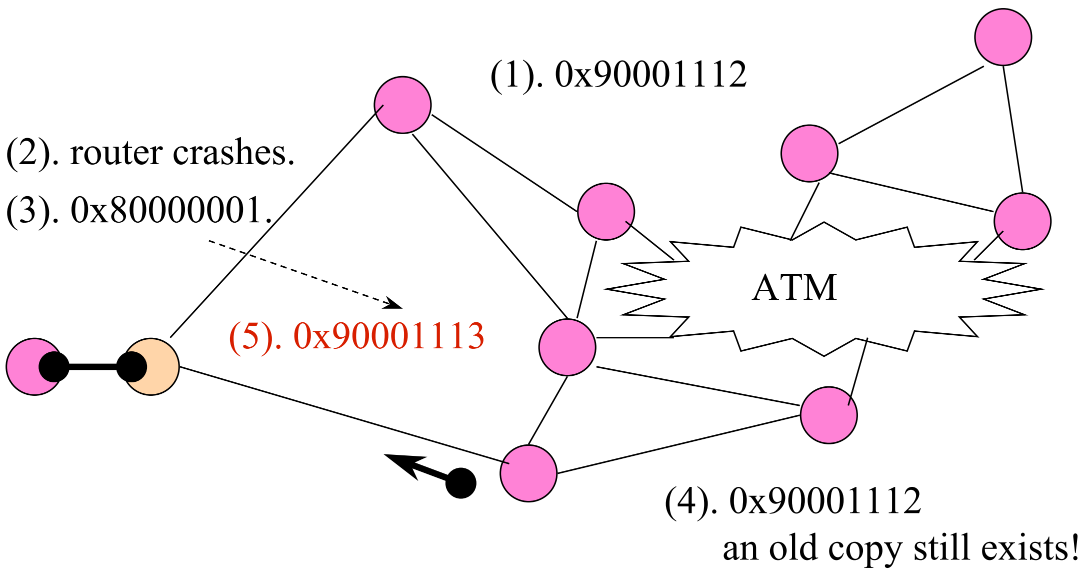


Sequence #: old vs. new LSAs

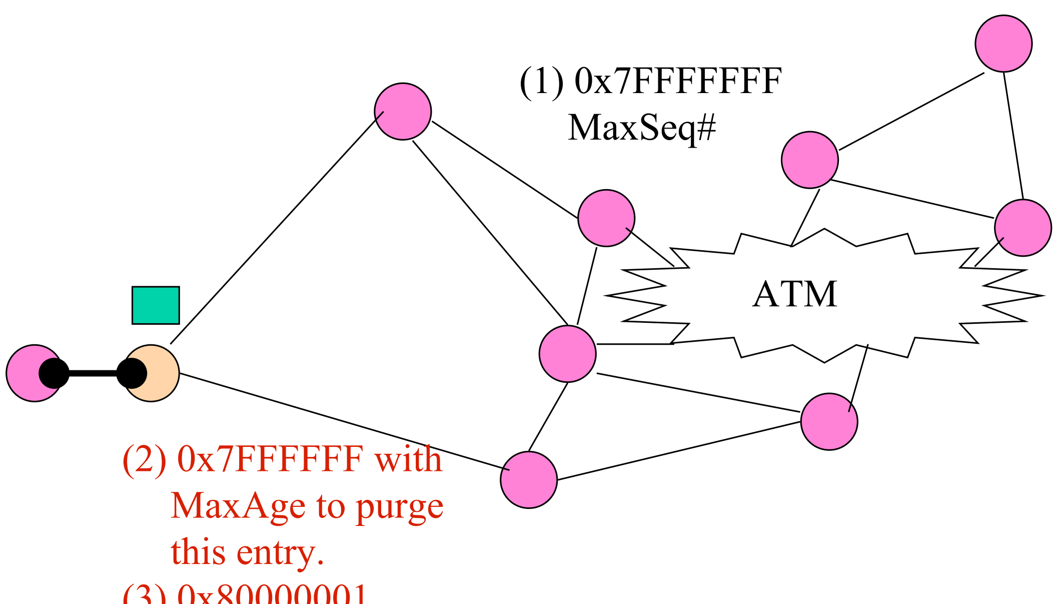


Only accept LSAs with newer/larger Seq#.

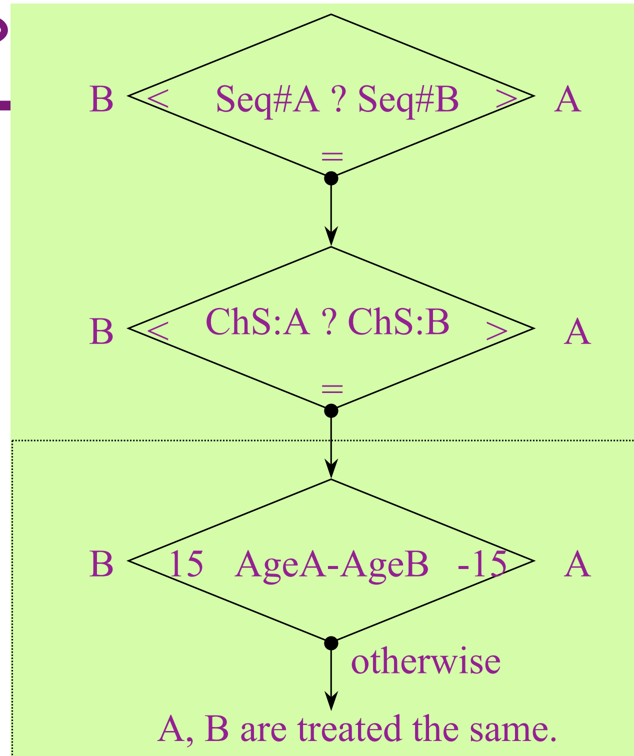
Sequence # Self-Stabilization



Sequence #: Counter Flushing



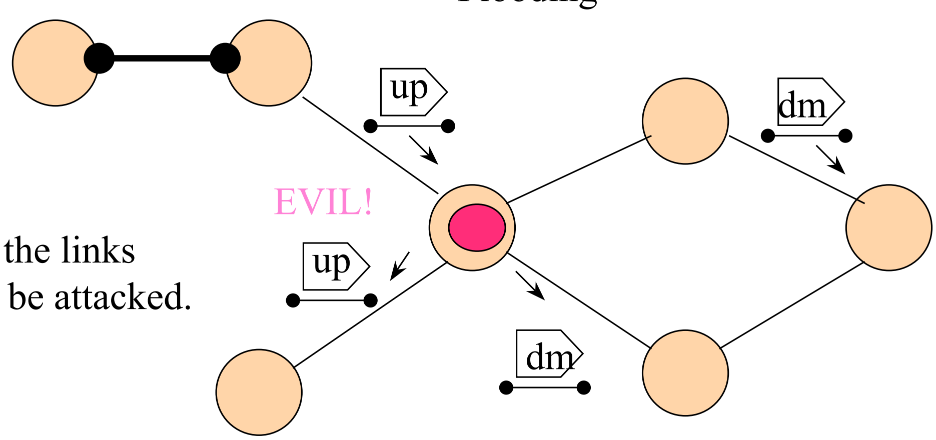
Fresher LSA?



Three parameters for LSA:
- Sequence Number
- Checksum
- Age

Malicious Intermediate Routers

Flooding



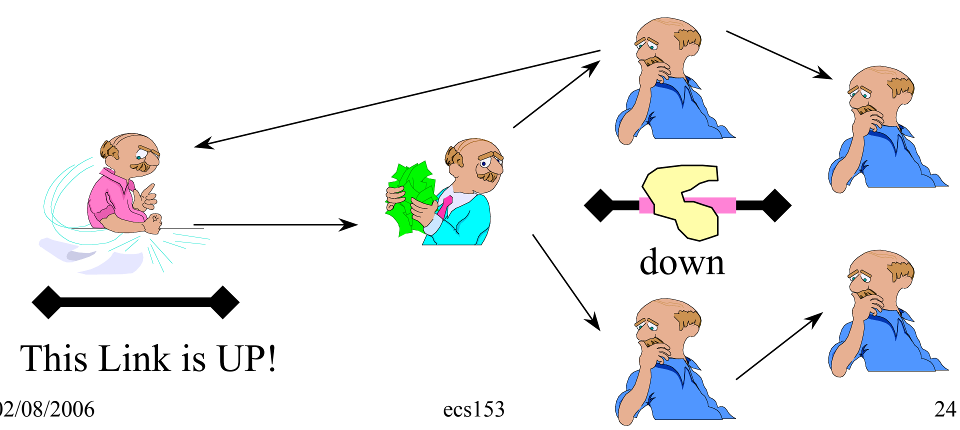
All the links can be attacked.

How to Attack OSPF?

- Think…
- Try it!!
  - What is the objective?
  - How to accomplish your goal?

Problem

- Prevent/Detect compromised intermediate router(s) from tampering "Link State Advertisements (LSAs)" originated from some other routers.

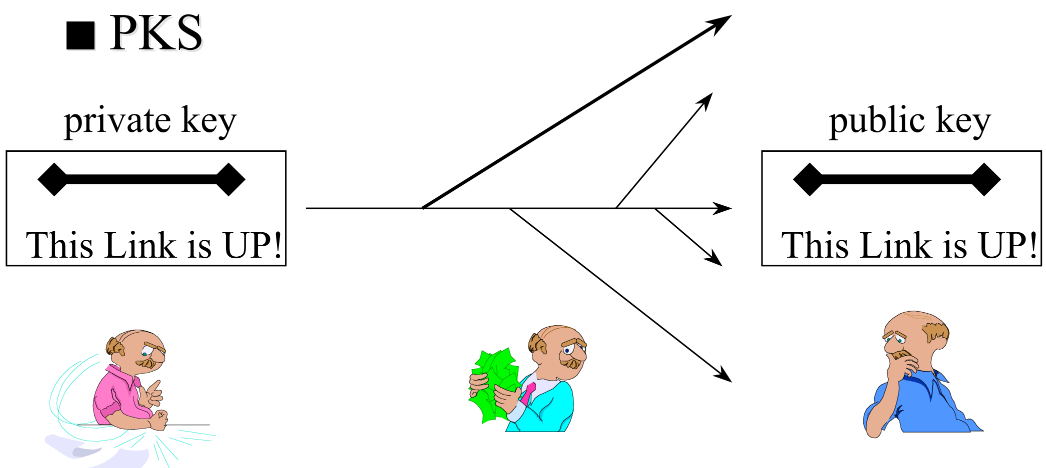


Defense??

- Crypto-based
- Non-crypto-based

LSA Digital Signature

- PKS



Advantages for PKS

- One compromised router will not be able to affect others about other links.
- With only key-MD5, one compromised router can disable all the crypto.

Public Key is Expensive

- At least, in software.
- Experiments on Pentium/133, Linux 2.027:
  - HMAC MD5: 78.37 usec
  - RSA/MD5 (verify): 88.00 msec
  - RSA/MD5 (sign): 166.00 msec
- RSA Hardware available, where MD5 is inherently hard to parallelize.

Prevention

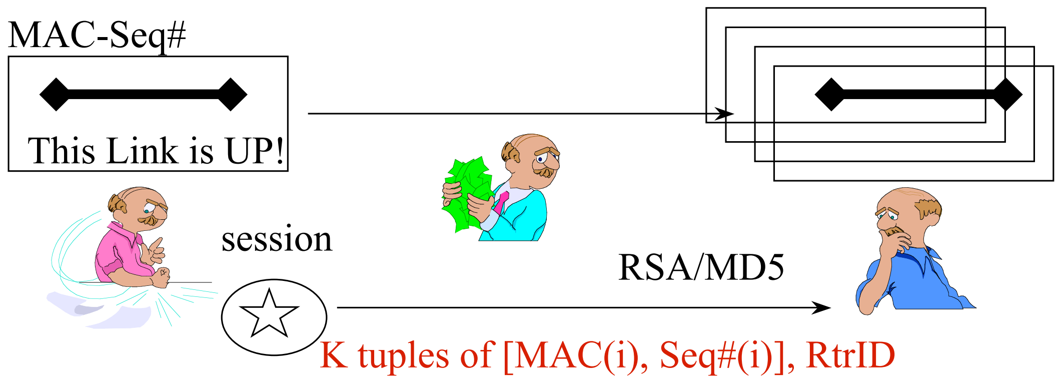
- LSA Originator Digital Signature (Perlman, Murphy/Badger, Smith/JJ)
- Debatable Concerns: (OSPF wk-group)
  - RSA is too expensive (about 1,000 times worse in signature verification with 512 bit keys)
  - PKI Certificate is expensive.
  - There are other routing infrastructure attacks that can not be prevented by LSA Digital Signatures. (Cost/Market concern)
  - Political and Technical.

Can we do it without PKI?

- Preventing compromised intermediate routers???

Can we detect the problem?

- Authenticated LSAs but the authentication information is kept until a session is over.



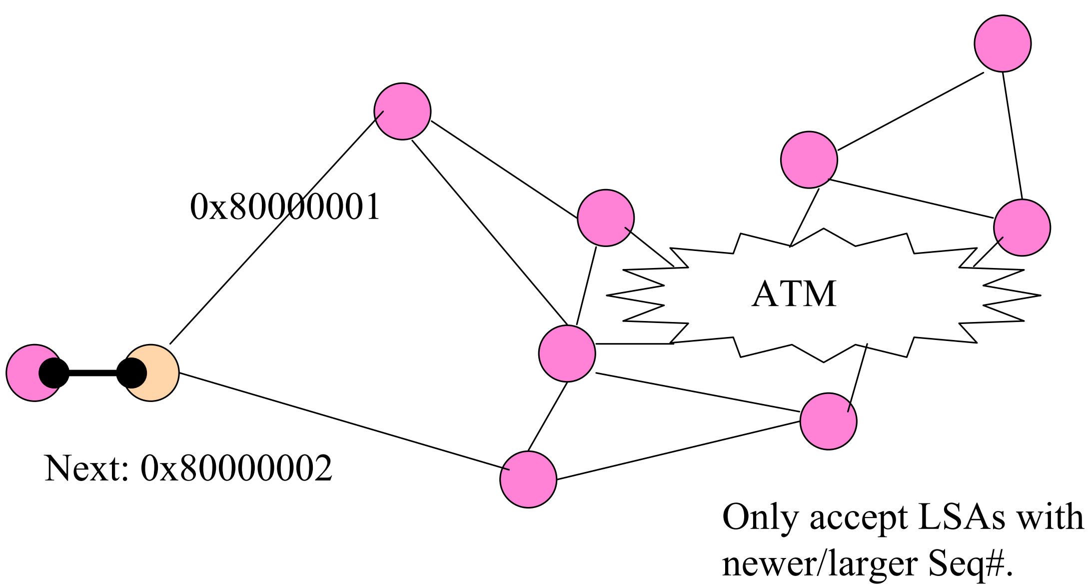
Prevention versus Detection

- Prevention: pay a fixed price anyway, even no bad guy exists.
- Detection/Isolation: "hopefully" pay less when no bad guy exists. pay more when trying to isolate the bad guys.
- Self-Stabilization Time:
  - (Detection + Isolation)

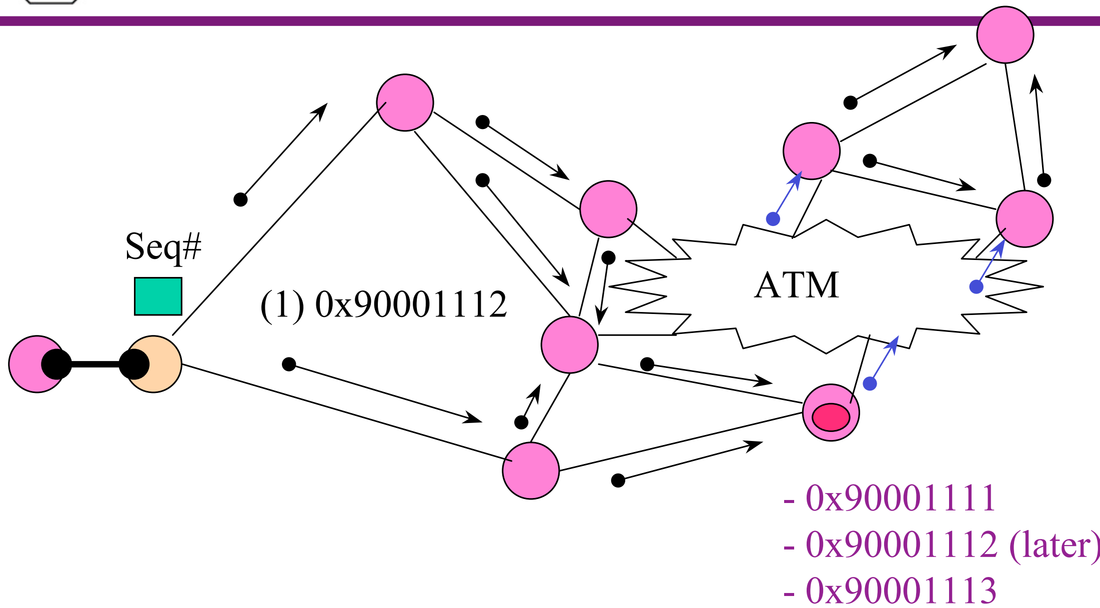
Let's look at the attack again!!

- Why do we need to ADD something to handle OSPF attacks?

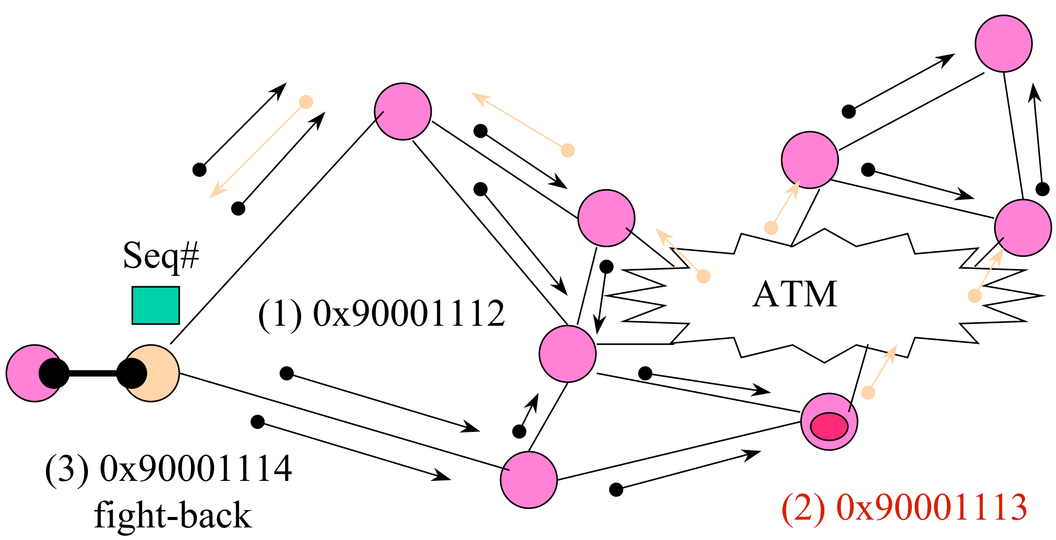
Sequence #: old vs. new LSAs



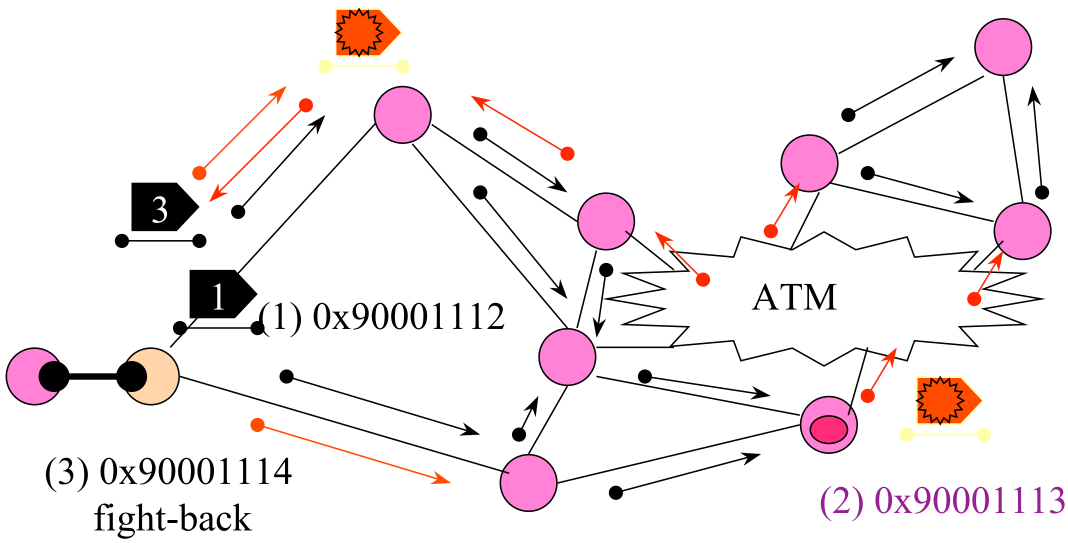
Attack



Attack and Fight-Back



Seq++ Attack and Fight-Back

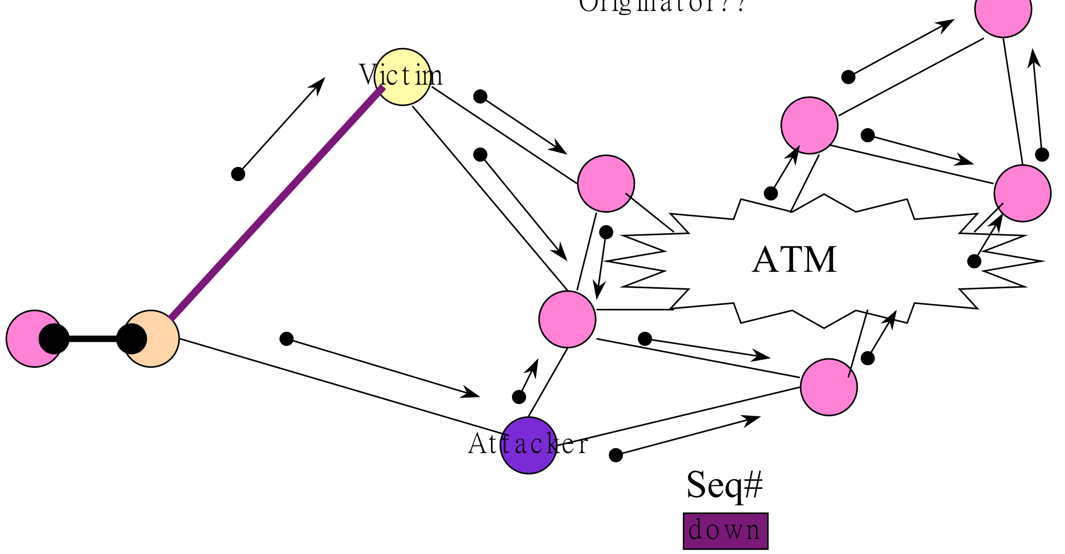


Current Seq# 9123abe0

Attacking Seq# 9123abe1

Responding Seq# 9123abe2

Originator??

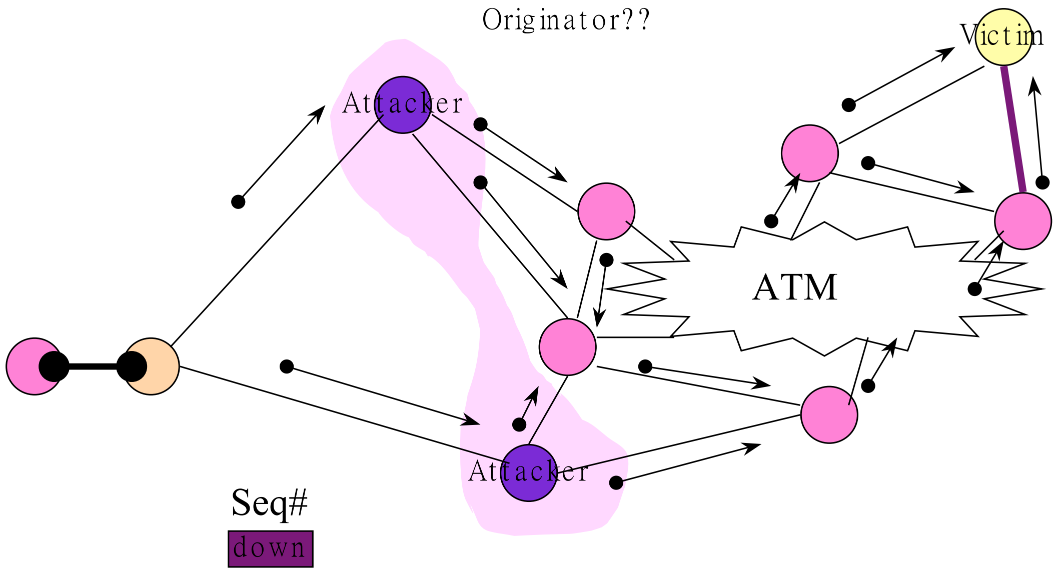


Current Seq# 9123abe0

Attacking Seq# 9123abe1

Responding Seq# 9123abe2

Partition



OSPF Security Strength

- In most cases, if something goes wrong, the advertizing router will detect it and try to correct it.
- The bad guy has to persistently inject bad LSAs.
- Self-Stabilization Protocols: can not handle continuous faults but force the attacker to perform only persistent attacks.

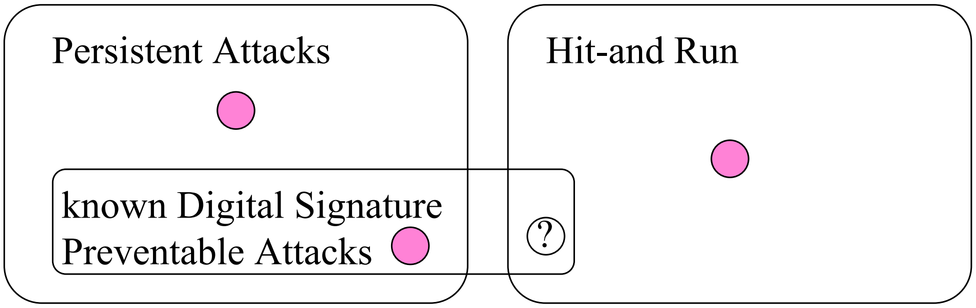
A Principle/Heuristic Rule of Intrusion Detection

- Hit-and-Run Attacks: Hard to Detect/Isolate
  - Inject one (or very few) bad packet causing permanent or long term damage.
- Persistent Attacks:
  - The bad guy has to continuously inject attack packets.

Network Protocol/System Design

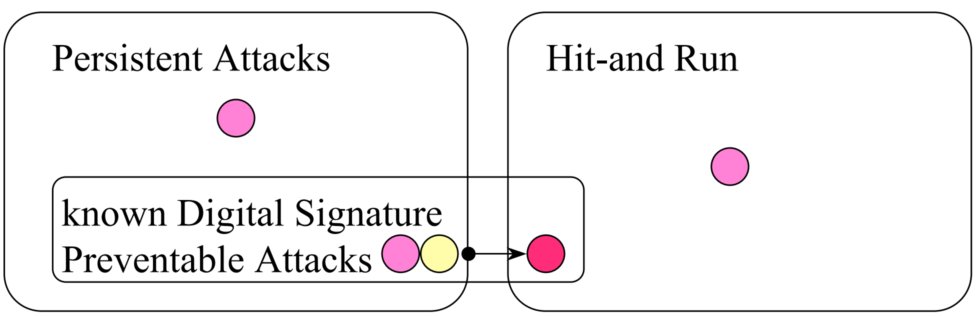
- If we can force the attackers to only launch "persistent attacks," we have a better chance to detect and isolate the attack sources.
- OSPF Flooding, for example, does a fairly good job. (still need some formal/theoretical research work here…)

Attacks on OSPF/RFC



One "sort-of" Hit-and-Run attack in OSPFv2 RFC is the "External-Forwarding-Link LSA Attack," and it can not be prevented by Digital Signature.

Attacks on OSPF/Implementation



MaxSeq# attack ( ) was a Persistent Attack in OSPF/RFC, but, with implementation bugs, it becomes a Hit-and-Run attack ( ).

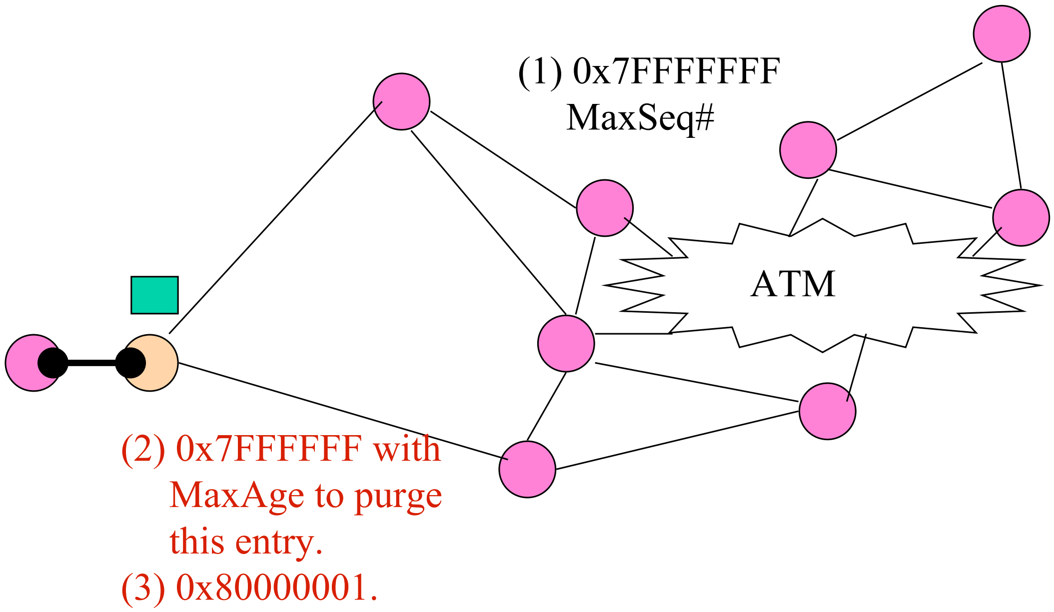
Results for OSPF:

- According to the RFC, all the known Digital-Signature-preventable attacks can be efficiently detectable. (There are no known Hit-and-Run OSPF attacks that can be prevented by PKS digital-signature.)
- According to the OSPF Implementations, one such Hit-and-Run attack does exist.

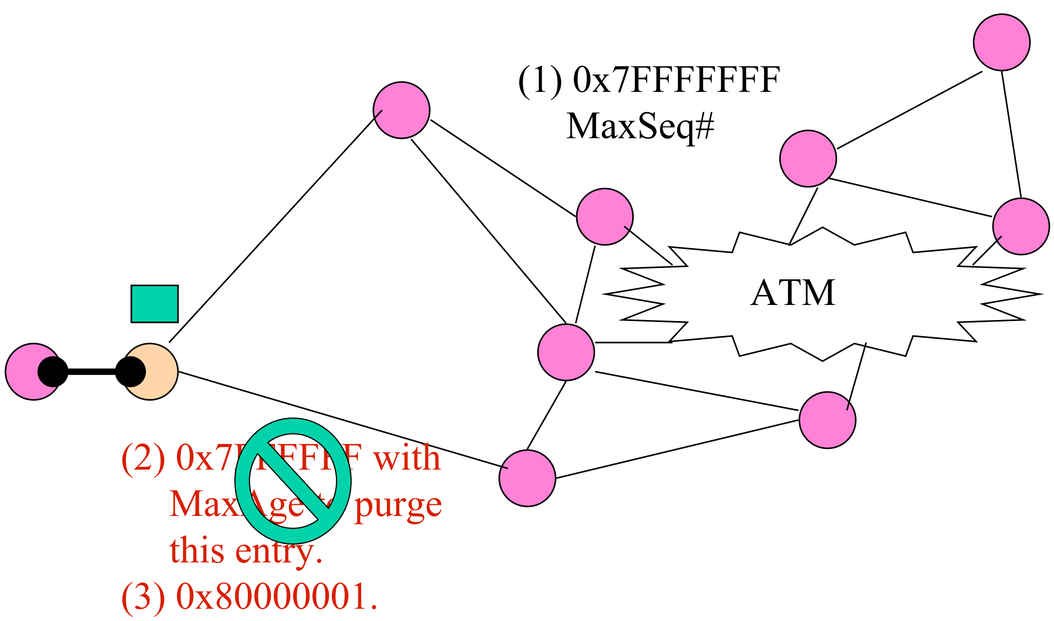
Max-Sequence Number Attack

- Block LSA updates for one hour by injecting one bad LSA. (You can hit it once and come back in an hour.)
- Implementation Bug! (Two Packages)
- MaxSeq# LSA Purging has not been implemented correctly!!

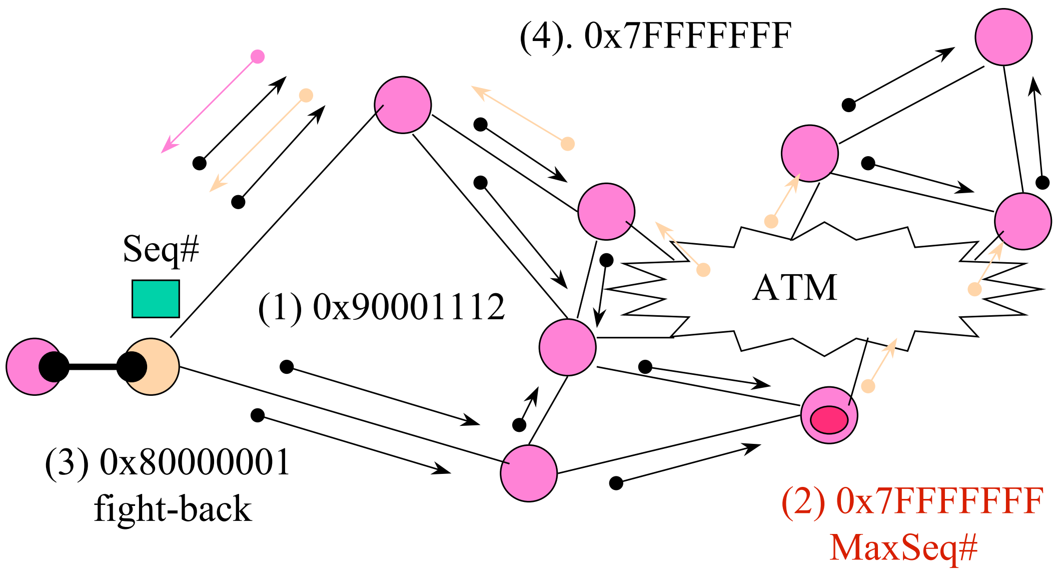
Sequence #: Counter Flushing



Sequence #: Counter Flushing



MaxSq# Attack



Properties of MaxSeq# Attacks

- Hit-and-Run for an Hour. The bad guy can "control" the topology database for an hour.
- The Victim continuously argues with its (very likely, honest) neighbors about which LSA is fresher. (0x7FFFFFFF versus 0x80000001).
- To eliminate the problem before one hour, "All" routers must be shut down "simultaneously."
- Or, have an active process to pump the purging packets into the network.

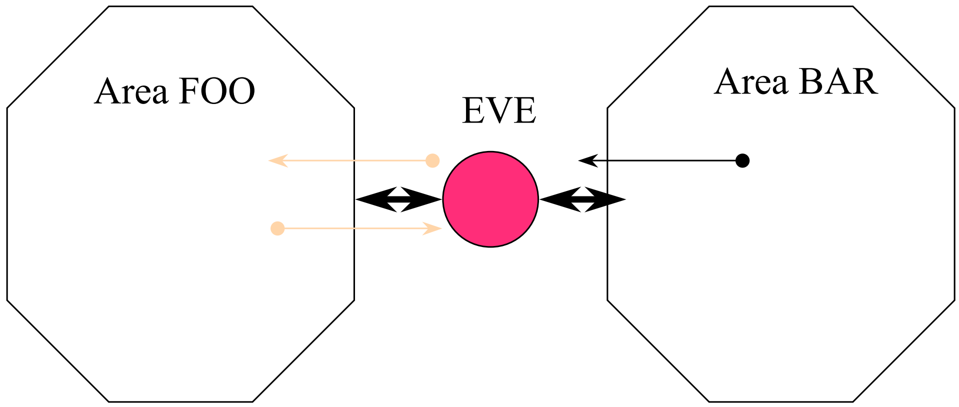
Max-Sequence Number Attack

- Block LSA updates for one hour by injecting one bad LSA. (You can hit it once and come back in an hour.)
- Implementation Bug! (Two independently developed OSPF packages.)
- MaxSeq# LSA Purging has not been implemented correctly!!
- Announced in May, 1997.

Detection → Isolation

- Detection
- Understand
- Isolation

Partitioned by Bad Router(s)



EVE can cheat FOO about BAR's topology without being detected by BAR.
(EVE can intercept the tampered BAR's LSAs from FOO to BAR.)

But….

- Any packets from FOO to BAR will pass EVE anyway. (I.e., EVE already has the access to all the packet streams between FOO and BAR.)
- It is not necessary for EVE to attack the routing information exchange protocols.

Is the network partitioned?

- YES.
  - The bad guy doesn't need to attack RIB!
- NO.
  - With OSPF, the bad LSAs should flowed back to the originator.
  - The originator will fight back to correct the problem. (Self Stabilization)
  - The bad guy has to persistently attack.