Model and Infrastructure for Peer-to-Peer Programming

Alexandre di Costanzo

Supervisor: Pr. Denis Caromel

- Goals and Related Woks
1. **Peer-to-Peer Infrastructure**
2. **Programming Model and API**
- Experimentations
- Perspectives

Université Nice SOPHIA ANTIPOLIS

Goals

- Hosts are not available all time
- Dynamic Computational Peer-to-Peer (P2P):
  - Intranet & Internet
  - Proposing a High Level Model
    - Dynamic JVMs Network (computation nodes)
    - Not a Network's Protocol, Not DHT
    - P2P Programming Model for Branch and Bound (B&B) problems
- ProActive Context: no modification of Java language, of JVMs, …

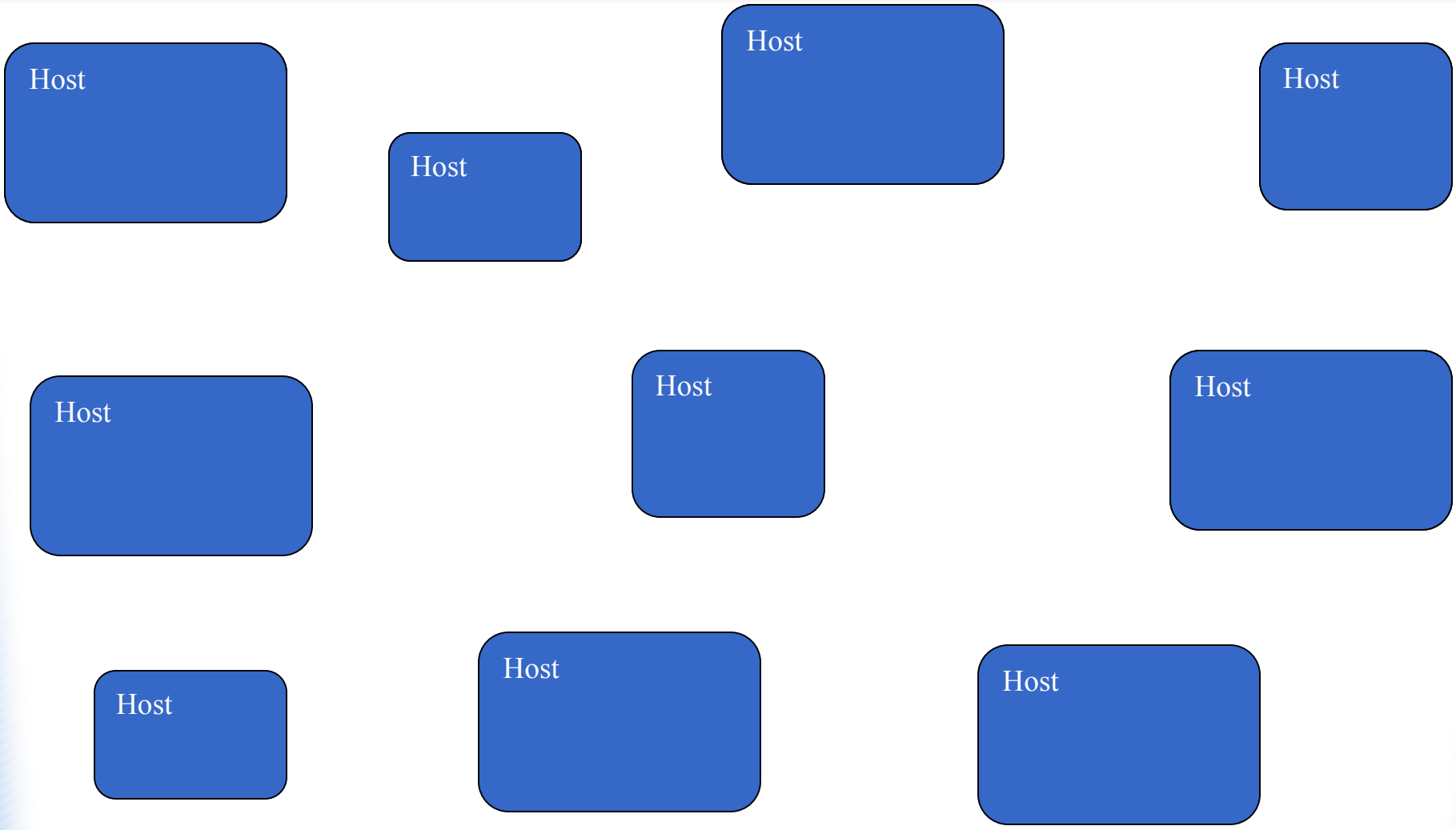
Related Works (Short)

- InteGrade :
  - Grid infrastructure for Intranet
  - Using desktop workstations
  - Master-slave, hierarchic
- XtremWeb :
  - Global Computing infrastructure
  - Large scale
  - No communication between tasks
- JXTA :
  - P2P library
  - Large scale, asynchrony and heterogeneity (languages & platforms)
  - Complete, low-level
- ProActive
  - Java library for Grid
  - Asynchrony, group's communications
  - Static deployment and heterogeneity (Java)

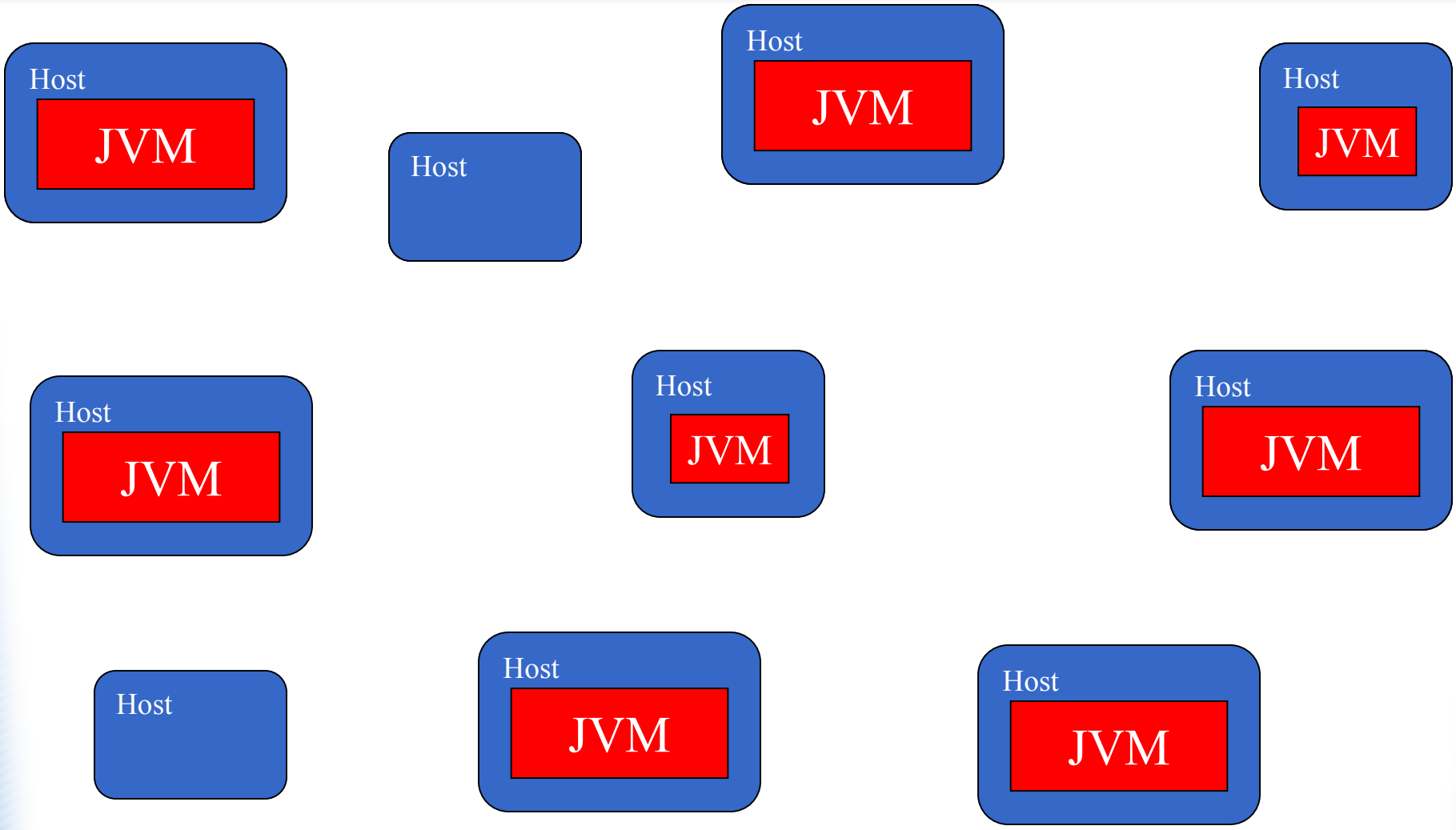
P2P Infrastructure

1st Part

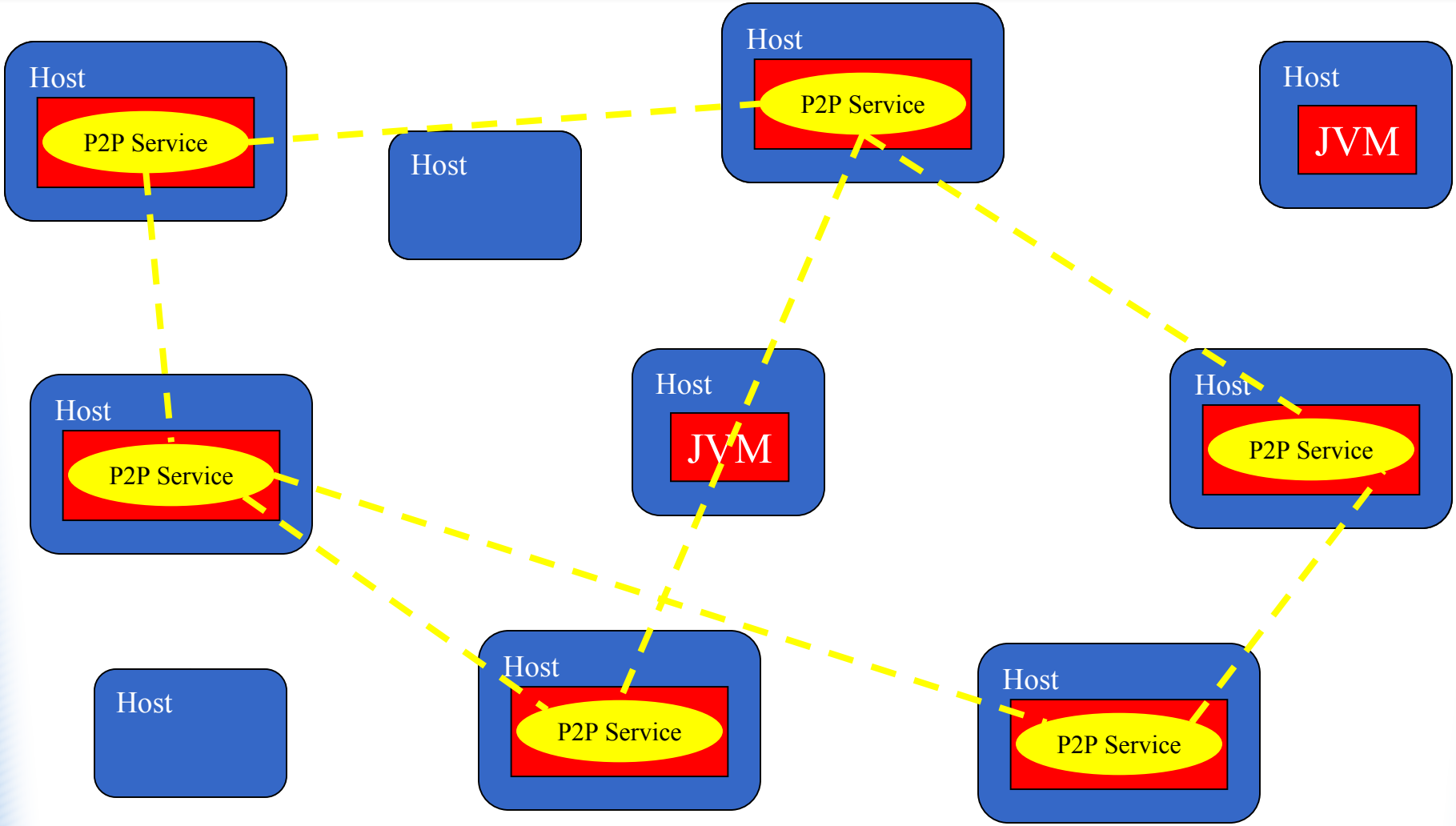
Proposed Architecture



Proposed Architecture



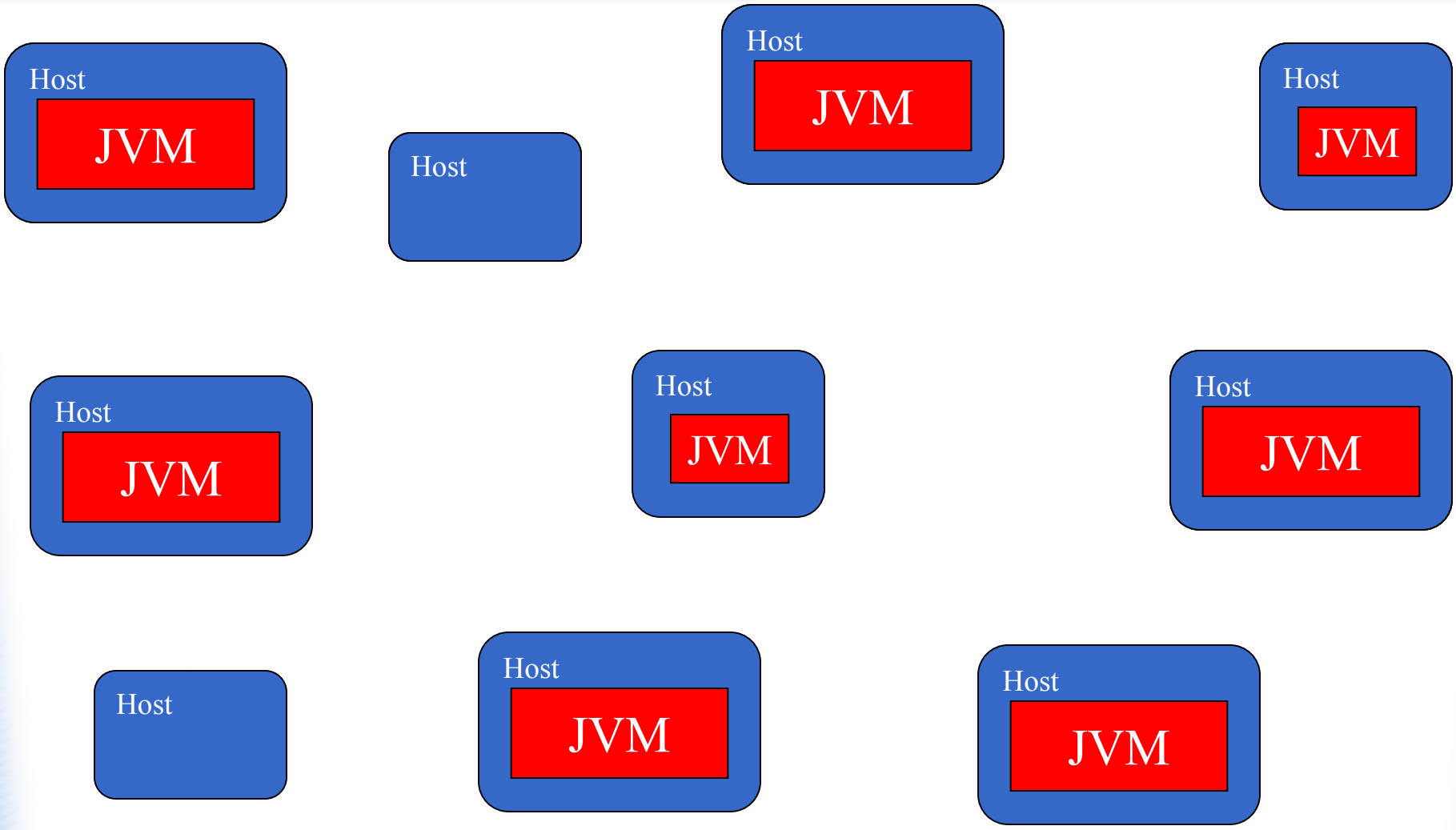
Proposed Architecture



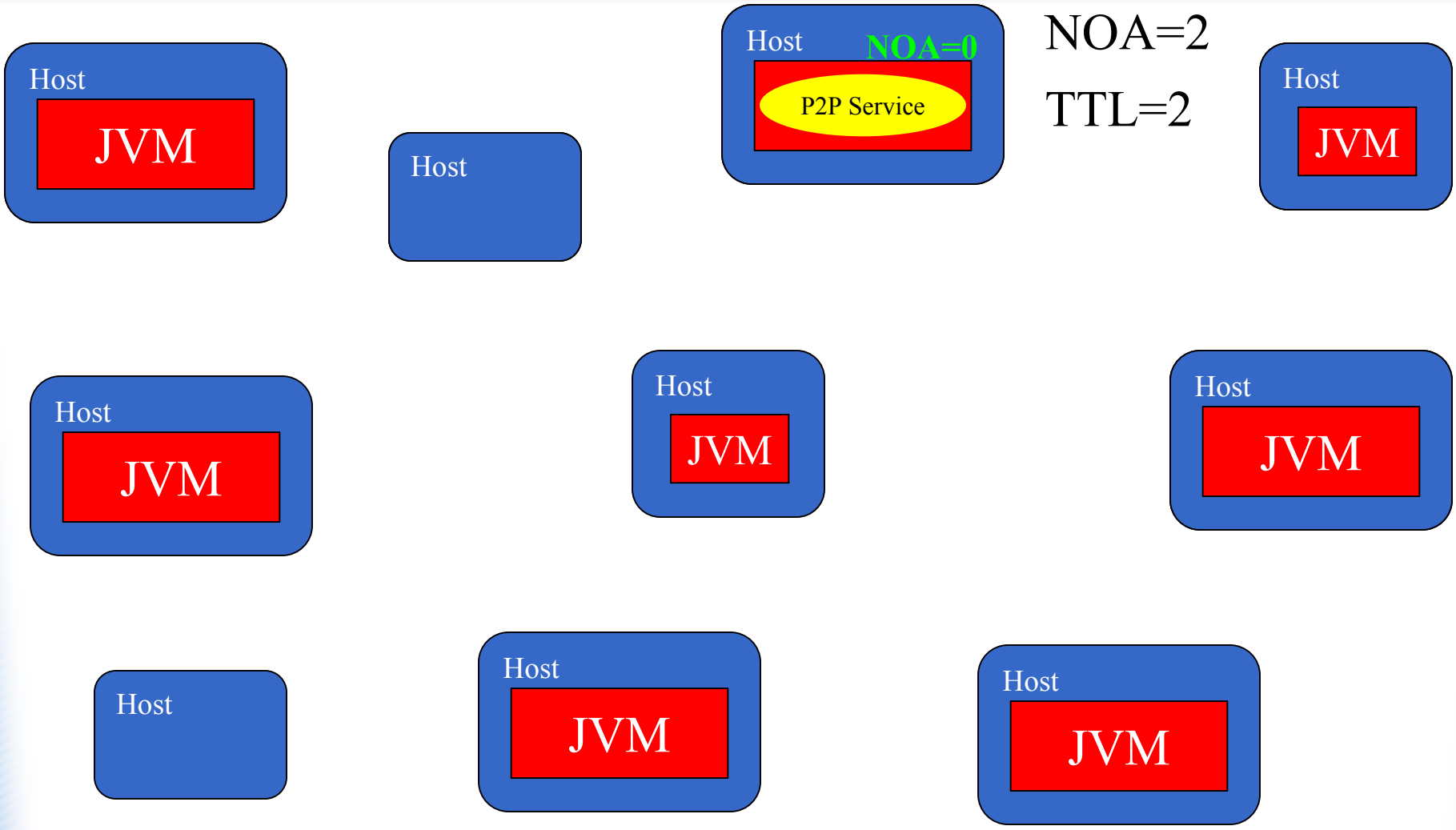
P2P Infrastructure

- Hosts Network ➔ JVMs Network
  - Overlay Network ≠ Indexed Search (DHT)
- Bootstrapping: many protocols (Jini, RmiRegistry, …)
- Dynamic environment :
  - Discovery : recording and un-recording
  - Resources (JVMs) acquisition
- Self-Organizing and Tunable Infrastructure :
  - Time To Update (TTU): peers (JVMs) availabilities
  - Number Of Acquaintances (NOA): keep up infrastructure
  - Time To Live (TTL): in hop for JVMS depth search, use for NOA
- Recursive Search of JVMs

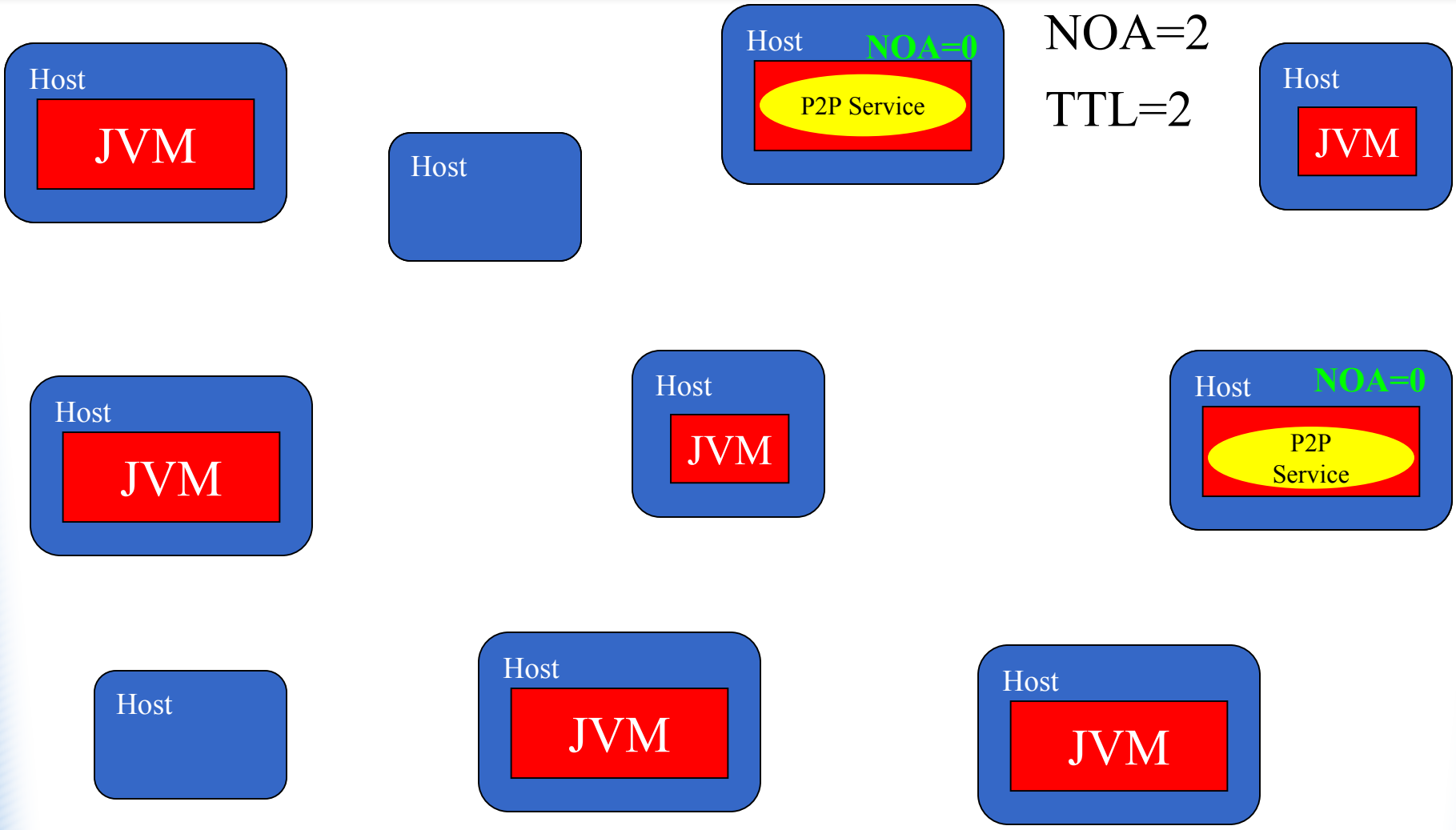
P2P Infrastructure



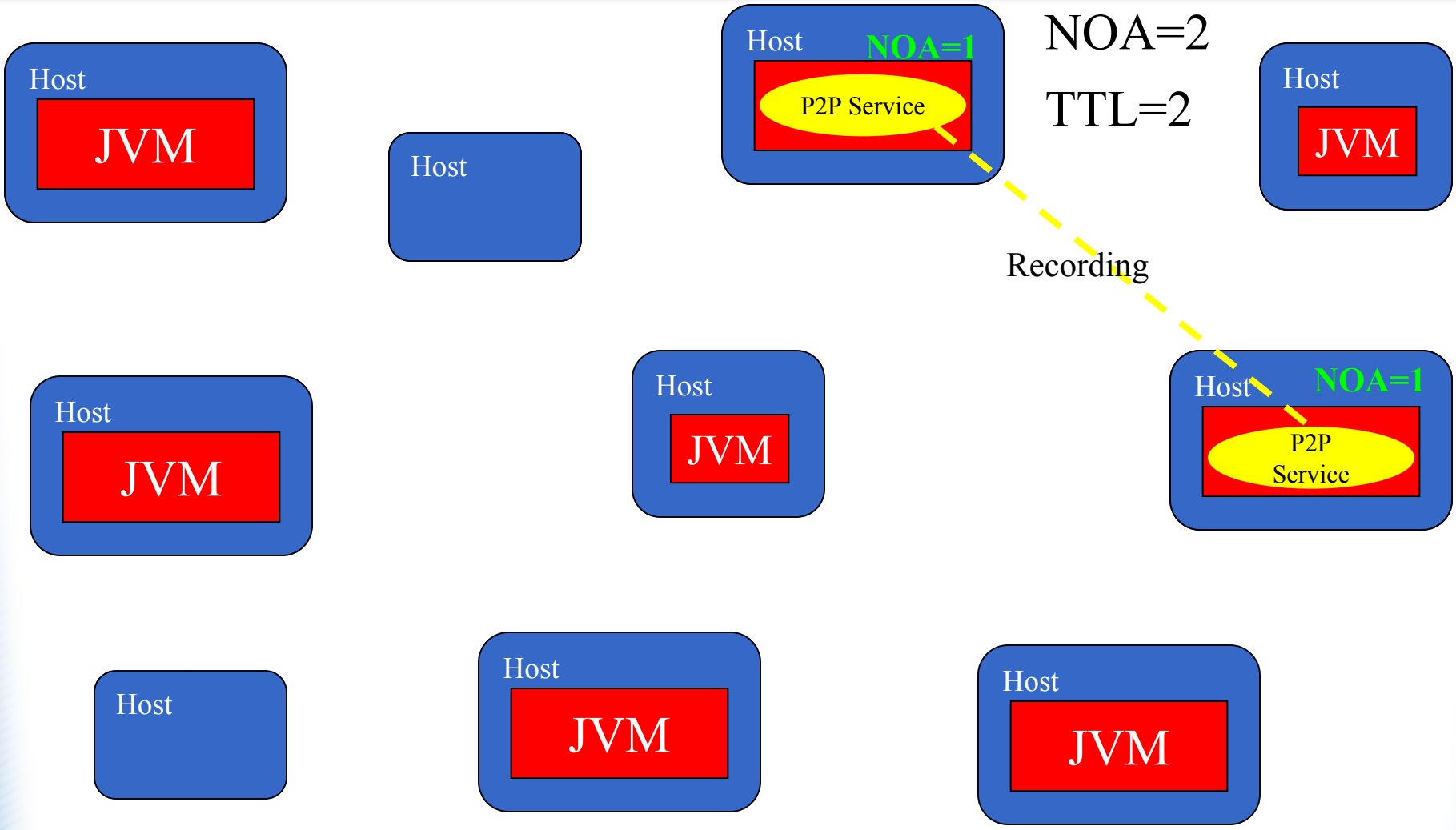
P2P Infrastructure



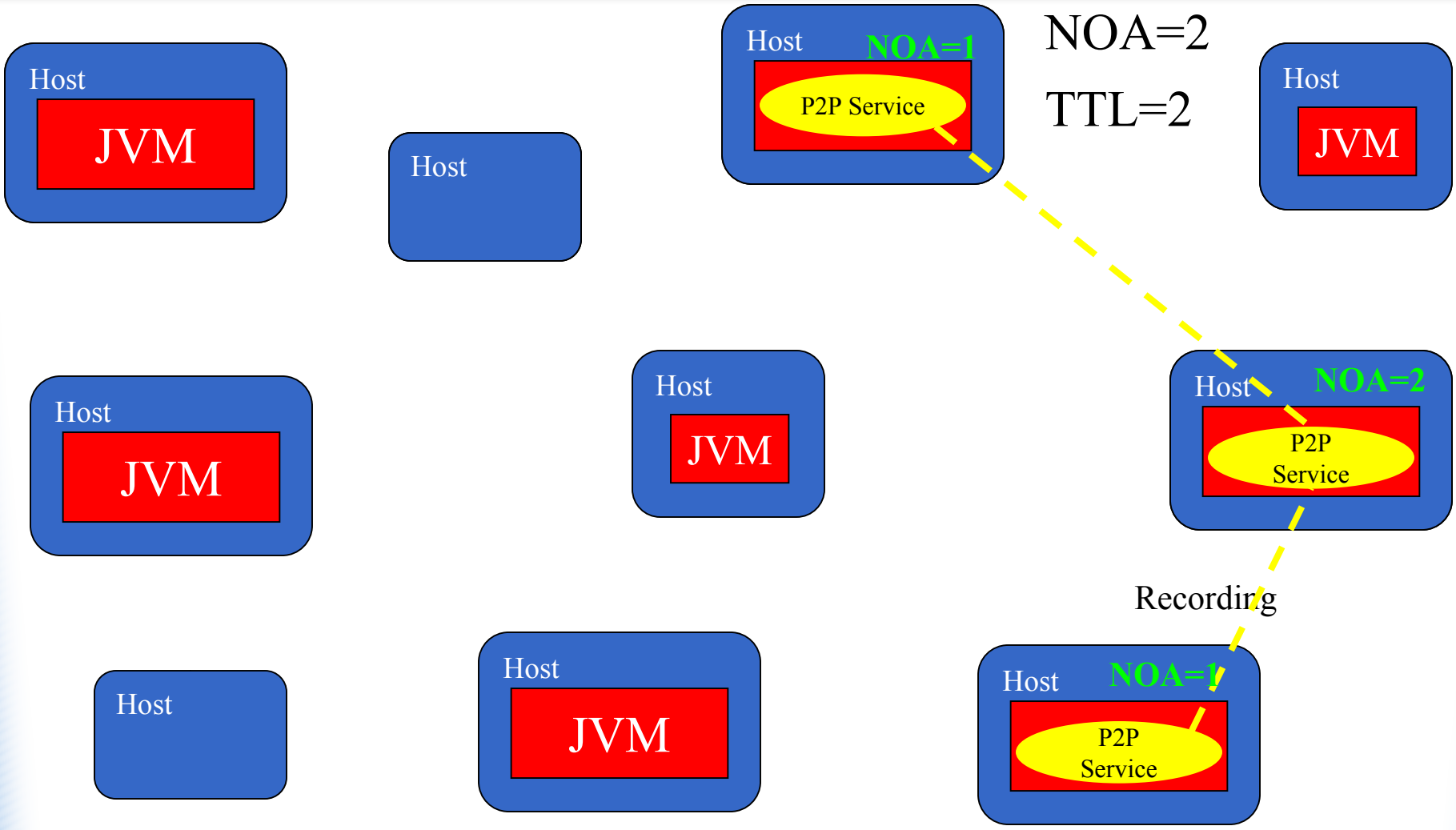
P2P Infrastructure



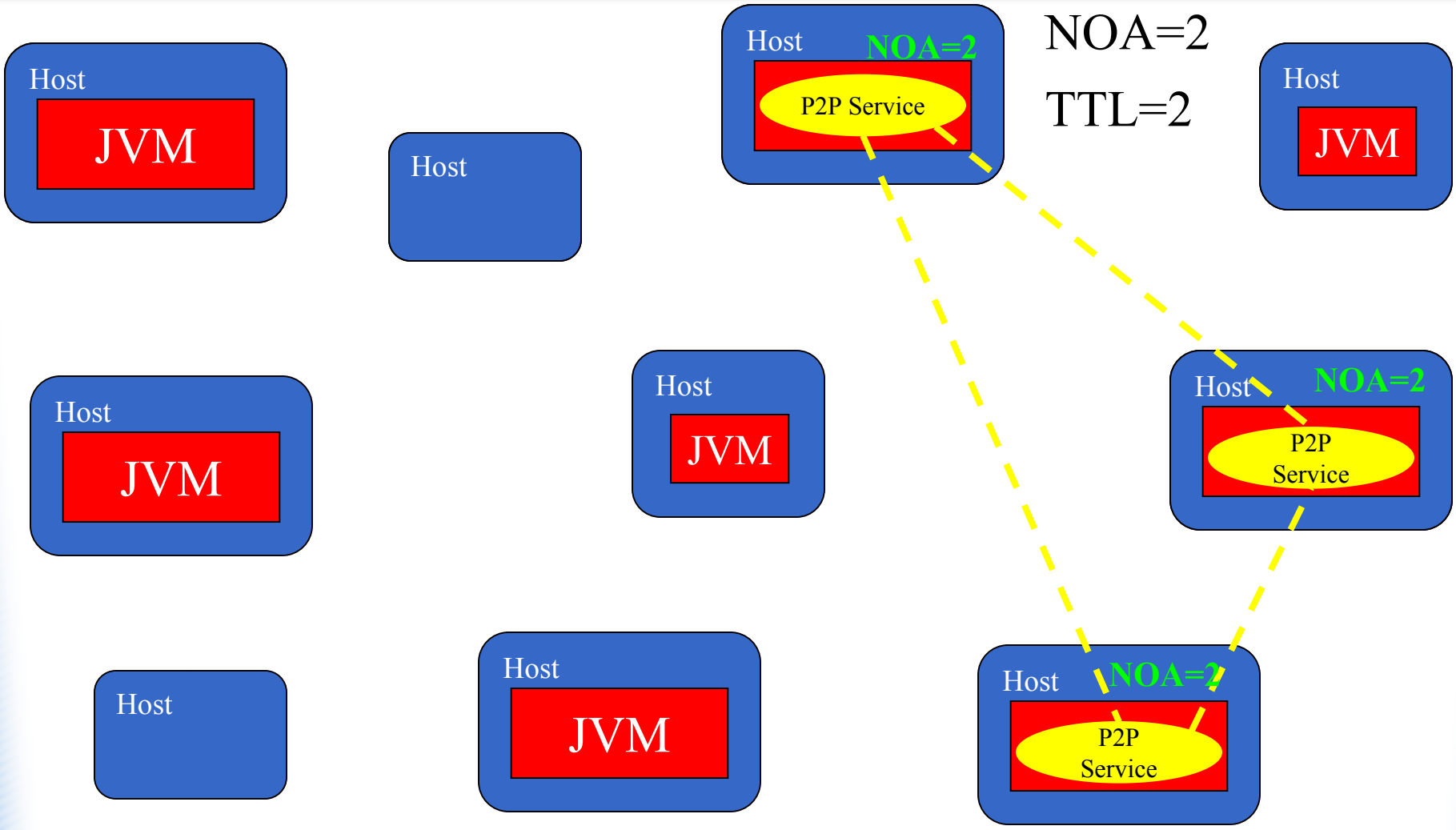
P2P Infrastructure



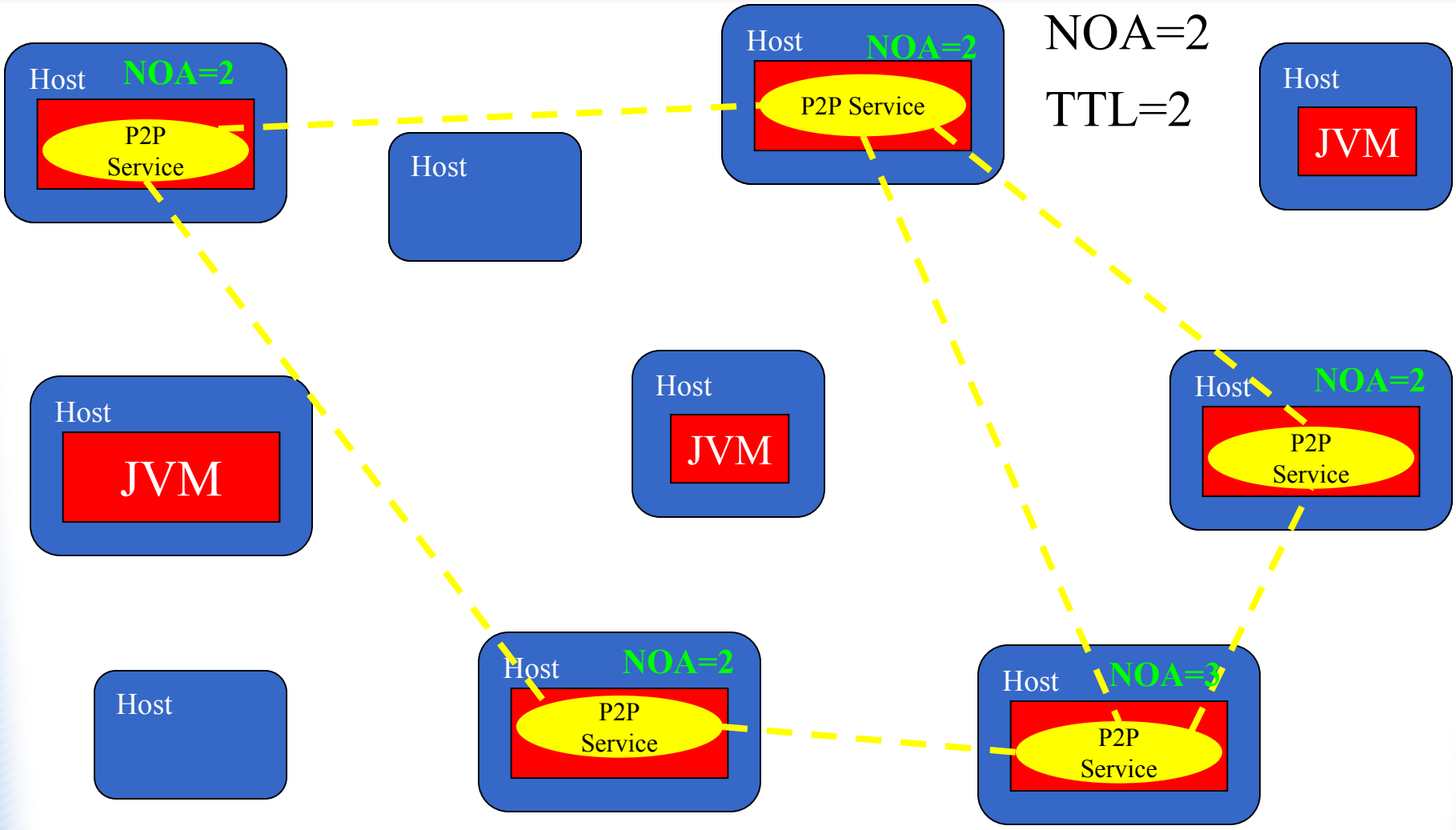
P2P Infrastructure



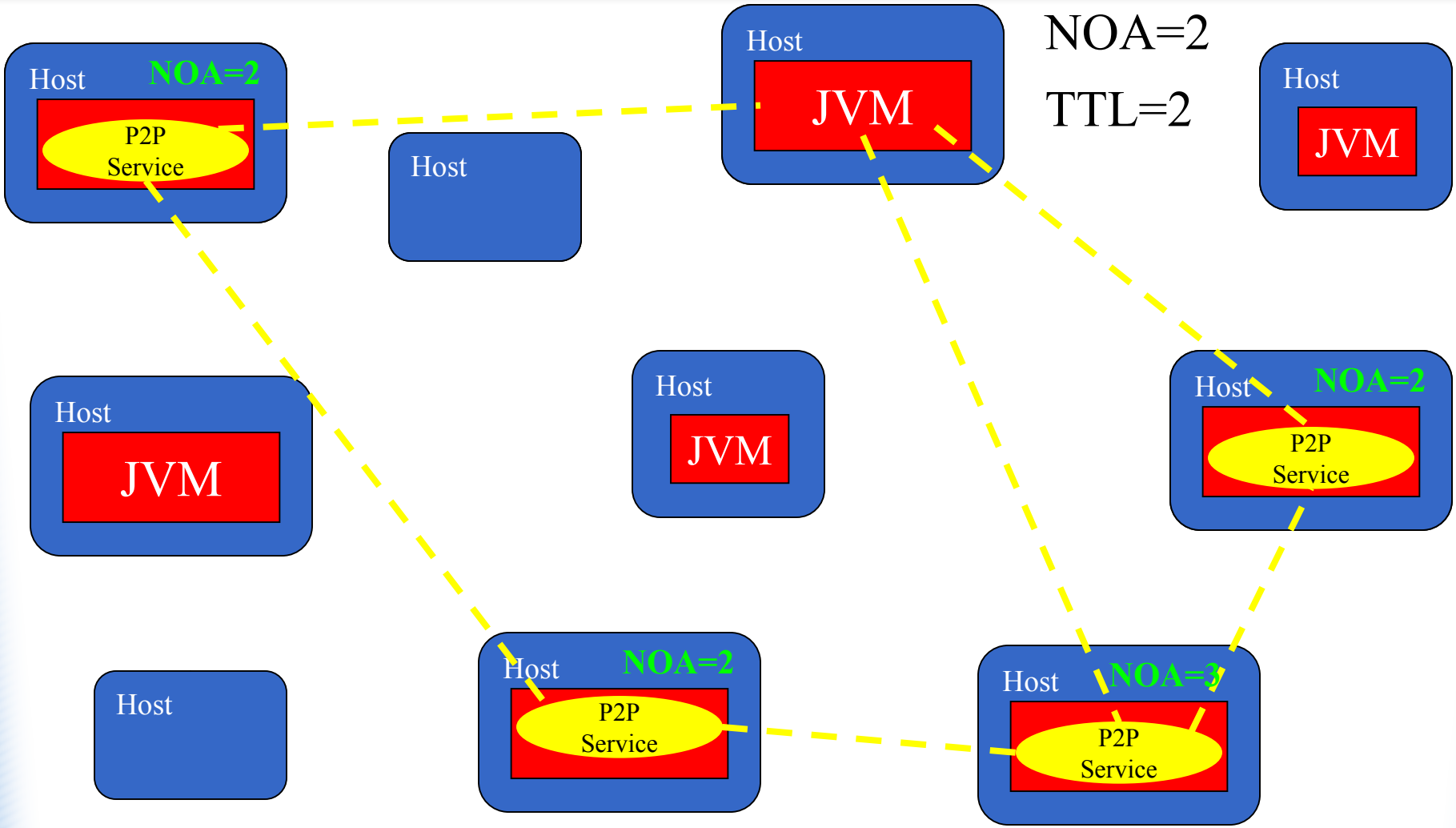
P2P Infrastructure



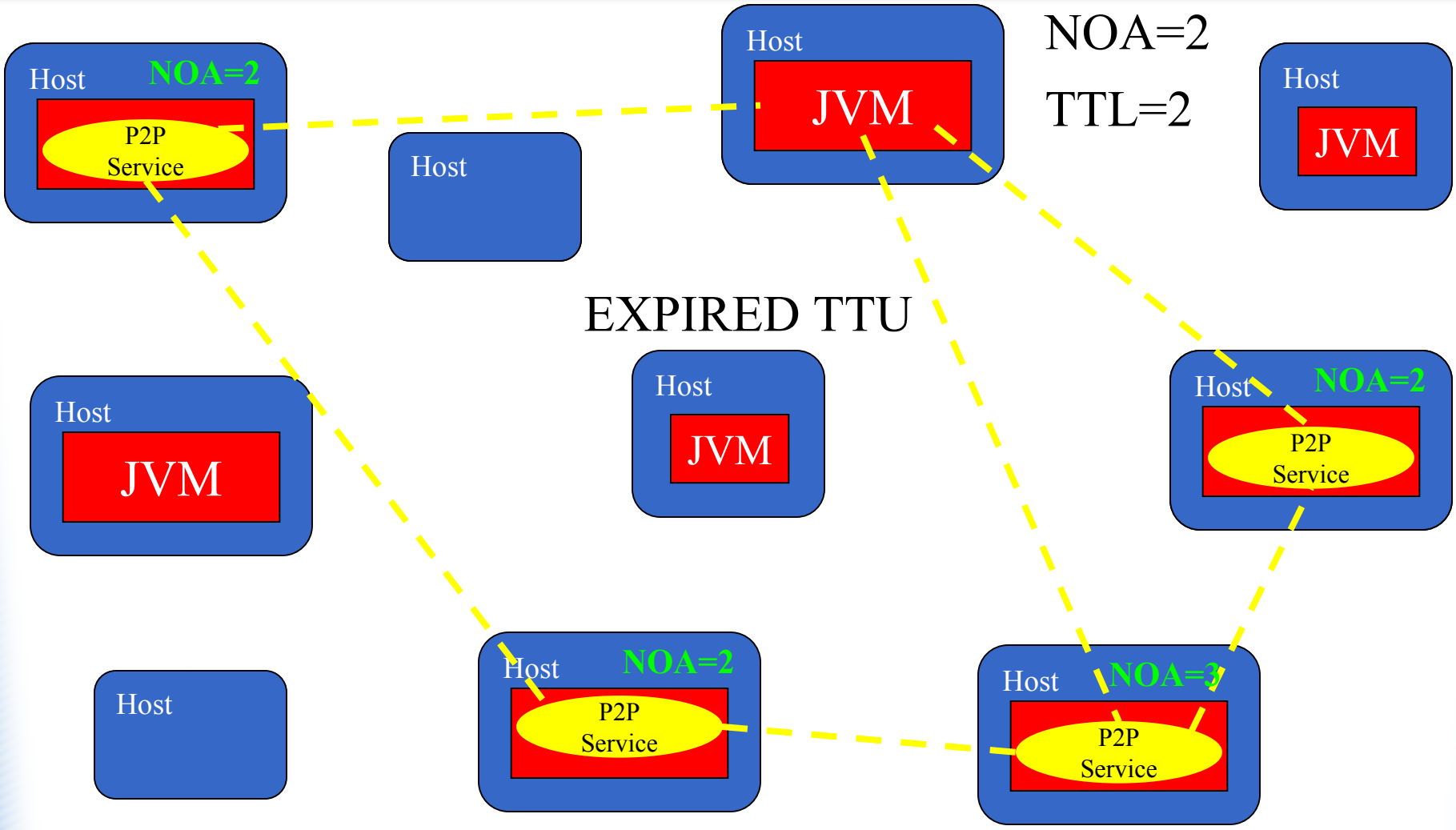
P2P Infrastructure



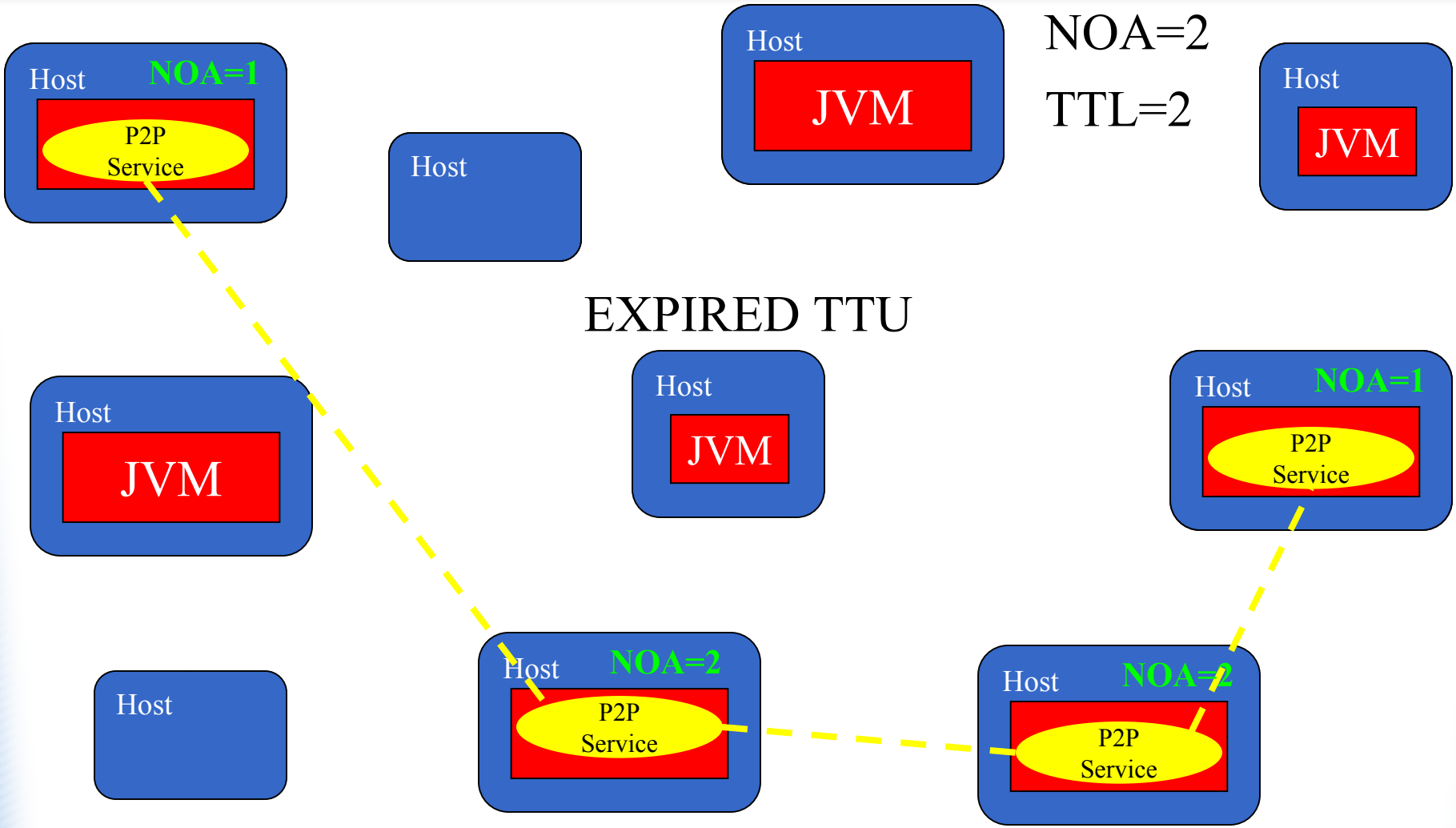
P2P Infrastructure



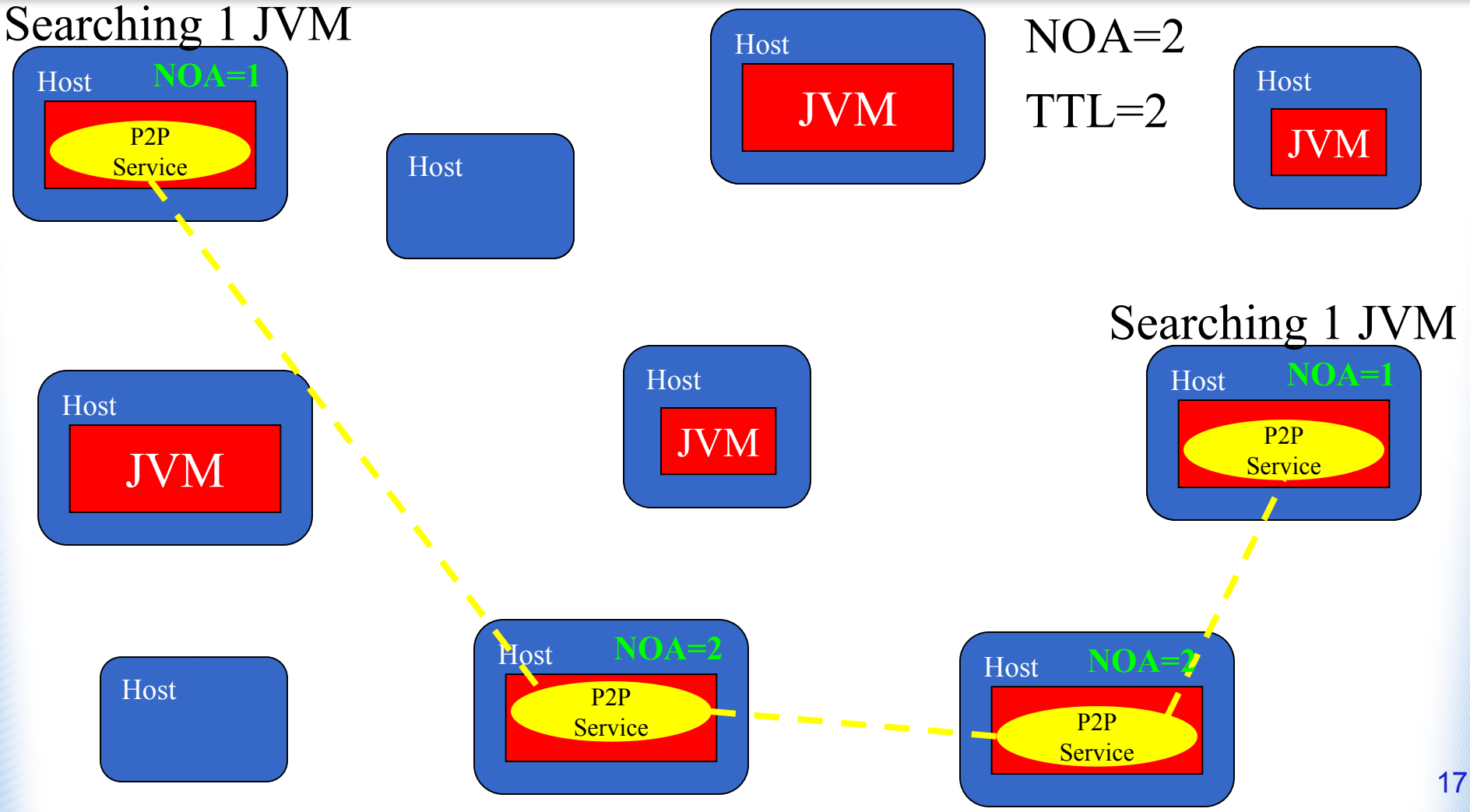
P2P Infrastructure



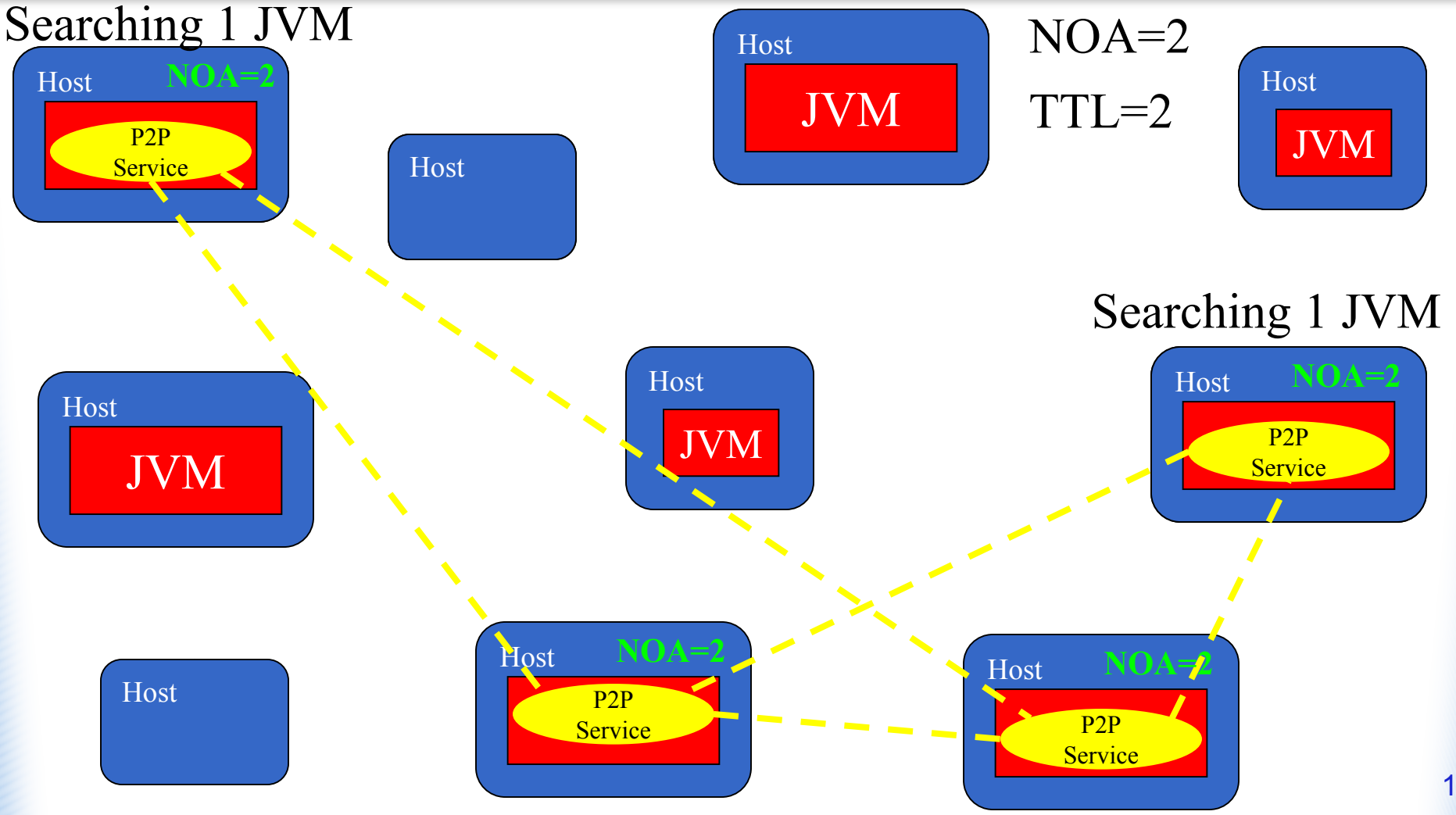
P2P Infrastructure



P2P Infrastructure



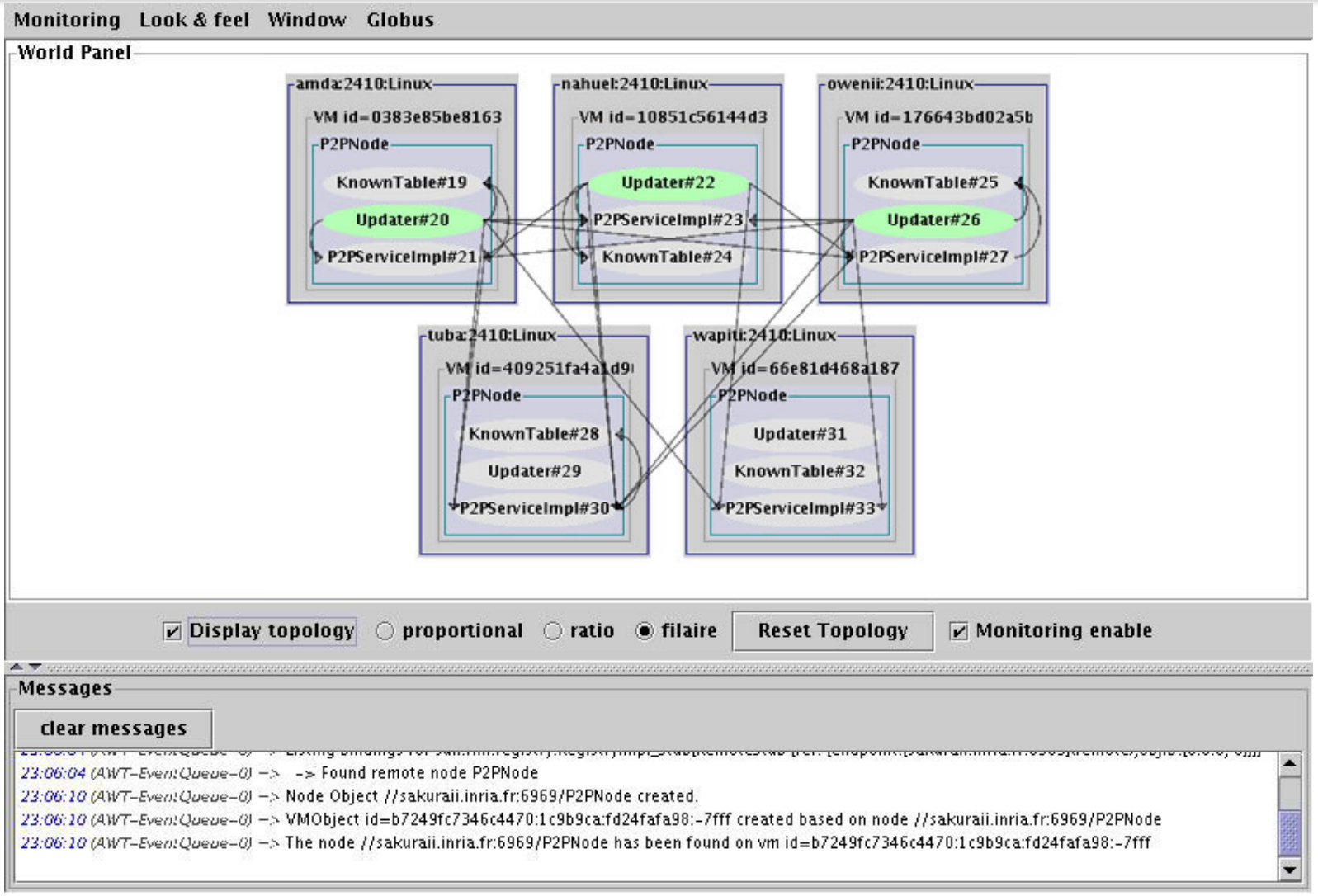
P2P Infrastructure



Implementation

- ProActive Runtimes: Standard JVMs
- Runtimes remote reaching and referencing to execute tasks:
  - Incompletes
  - Adding Active Objects:
    - **P2PService**: serves requests for recording and un-recording, accedes to Runtime's features and provides recursive search of JVMs
    - **Updater**: self-organizing and keeping up
    - **KnownTable**: list of acquaintances (peers)
  - Above communications protocols
    - Jini, RMI, Ibis and Web Services

Execution


Monitoring
Look & feel
Window
Globus
World Panel
amda:2410:Linux
VM id=0383e85be8163
P2PNode
KnownTable#19
Updater#20
P2PServiceImpl#21
nahuel:2410:Linux
VM id=10851c56144d3
P2PNode
Updater#22
P2PServiceImpl#23
KnownTable#24
owenii:2410:Linux
VM id=176643bd02a5b
P2PNode
KnownTable#25
Updater#26
P2PServiceImpl#27
tuba:2410:Linux
VM id=409251fa4a1d9
P2PNode
KnownTable#28
Updater#29
P2PServiceImpl#30
wapiti:2410:Linux
VM id=66e81d468a187
P2PNode
Updater#31
KnownTable#32
P2PServiceImpl#33
Display topology
proportional
ratio
filaire
Reset Topology
Monitoring enable
Messages
clear messages
23:06:04 (AWT-EventQueue-0) -> -> Found remote node P2PNode
23:06:10 (AWT-EventQueue-0) -> Node Object //sakuraii.inria.fr:6969/P2PNode created.
23:06:10 (AWT-EventQueue-0) -> VMObject id=b7249fc7346c4470:1c9b9ca:fd24fafa98:-7fff created based on node //sakuraii.inria.fr:6969/P2PNode
23:06:10 (AWT-EventQueue-0) -> The node //sakuraii.inria.fr:6969/P2PNode has been found on vm id=b7249fc7346c4470:1c9b9ca:fd24fafa98:-7fff

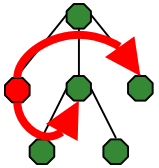
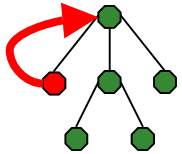
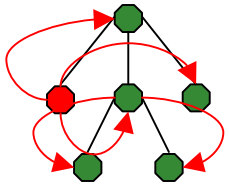
Programming Model

2nd Part

Programming Model

- Dynamic P2P Programming for B&B, etc.:
  - How to manage tasks, sub-tasks ?
  - Tasks communications: discovery tasks, volatility?
- Entities:
  - **Worker**: connects model with infrastructure
    - Task container
    - Getting new computational nodes
    - Tasks Communications
    - Managing dynamicity
  - **Solver**: Worker associate. Takes in input a problem creates sub-tasks (Problems) and found its best result
  - **Problem**: Worker associate
    - Dividing sub-problems
    - Merging results
    - Finding one solution
  - **Result**: solution abstraction

Programming Model

- Model is dynamic
- Communications between Workers :
  -  Brothers
  -  Father
  - All 
- Implementation built above ProActive Groups
  - Hierarchic organization
  - Communications between tasks
  - Tasks are parallels
- Troubles: volatility of Workers, comm.

Experimentations

P2P Infrastructure

-

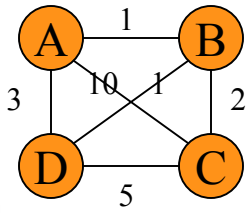
P2P Programming Model

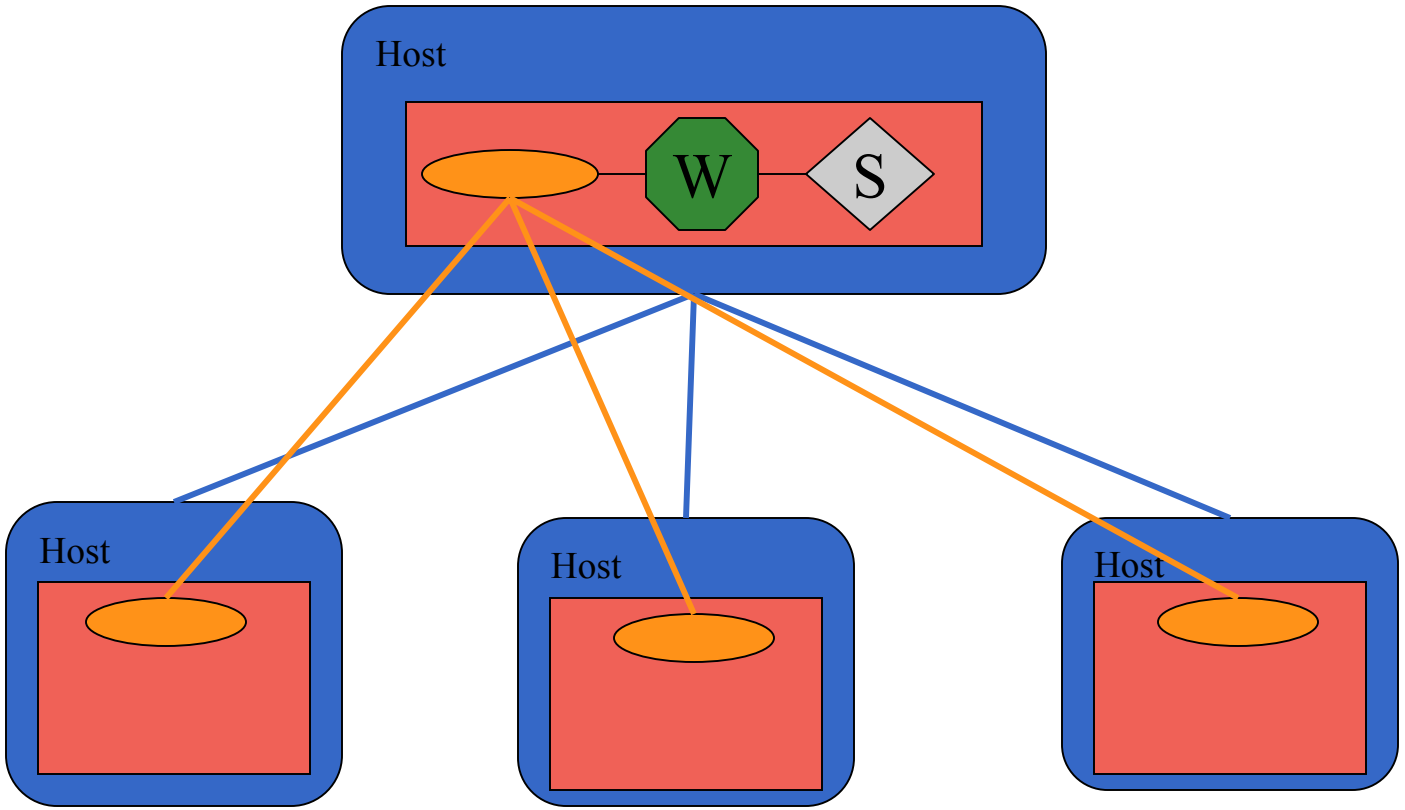
Infrastructure's Experimentation

- N-Queens:
  - Master-slave application with static deployment
  - Dynamic Workers Acquisition
  - Faults tolerant
  - TTU = 10 minutes ; TTL = 10 hops; NOA = 10 peers

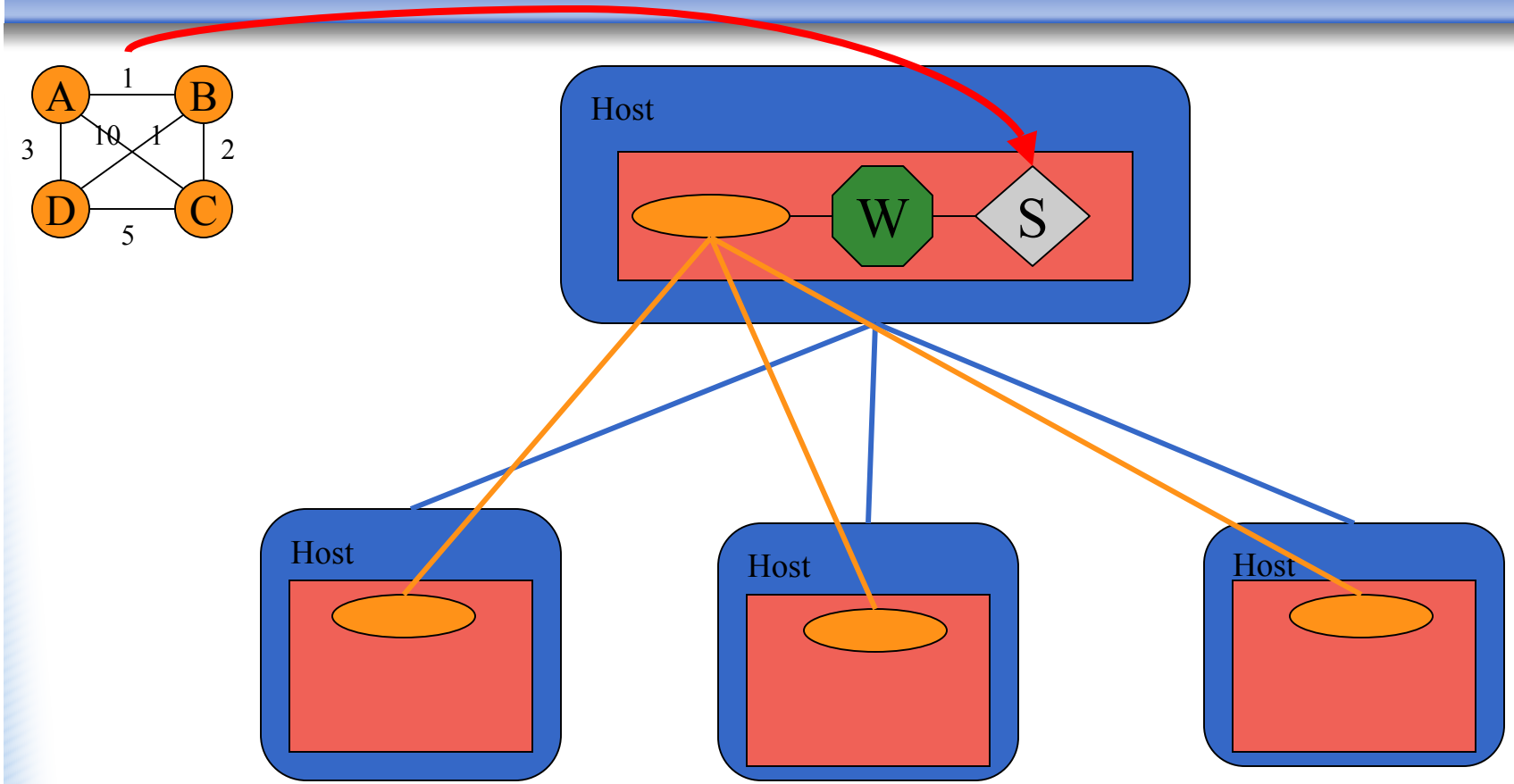
20-Q	23 Peers	Tasks : 2482	Cumulated CPU time: 73h7'22 ''	Real time : 3h15'20''	Solutions : 39029188884
21-Q	23 Peers	3081	569h23'16''	24h59'44''	314666222712
23-Q	23 day 130 night	116/4203	-	Since June 20th	-

Model's Experimentation
Traveling Salesman Problem

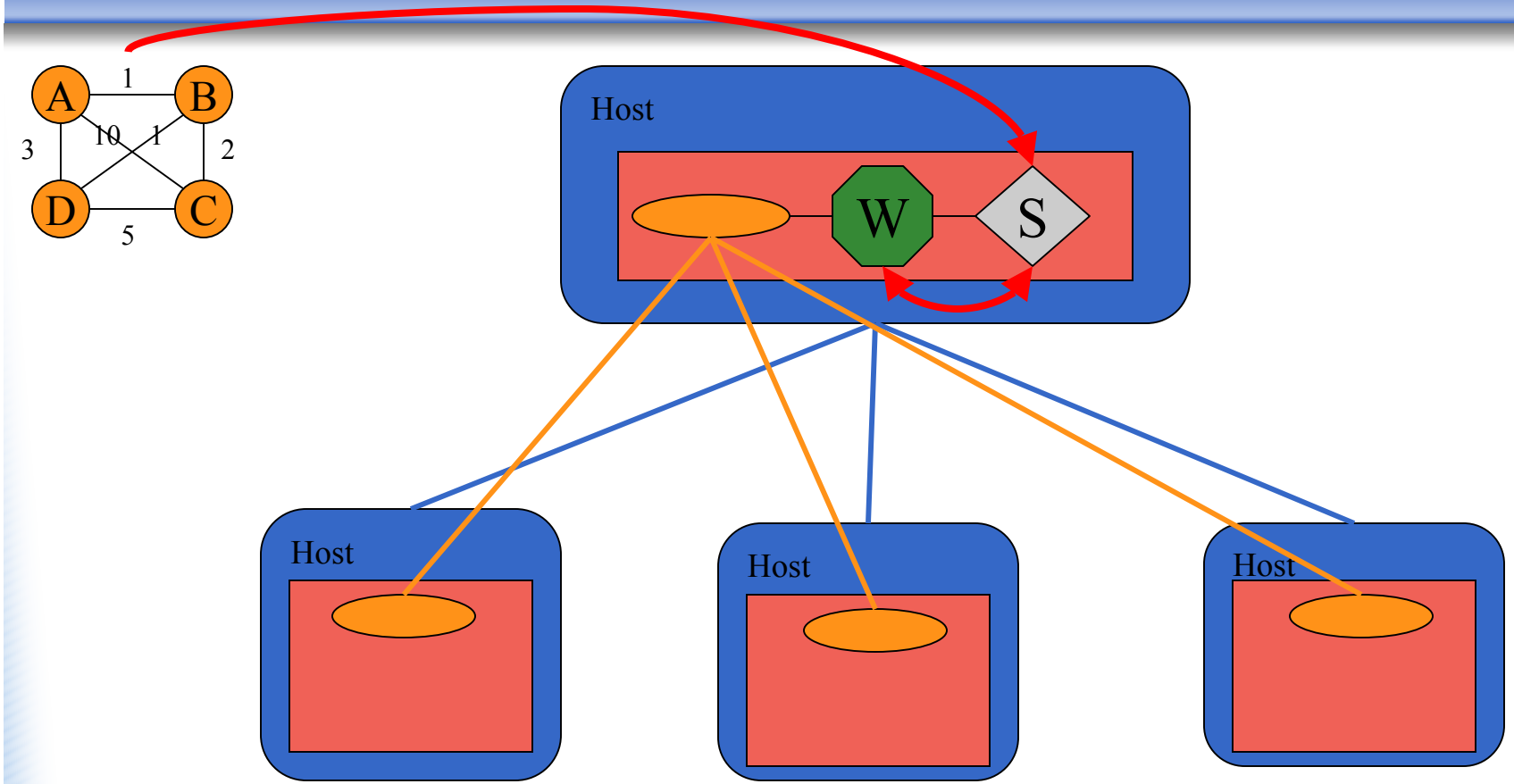




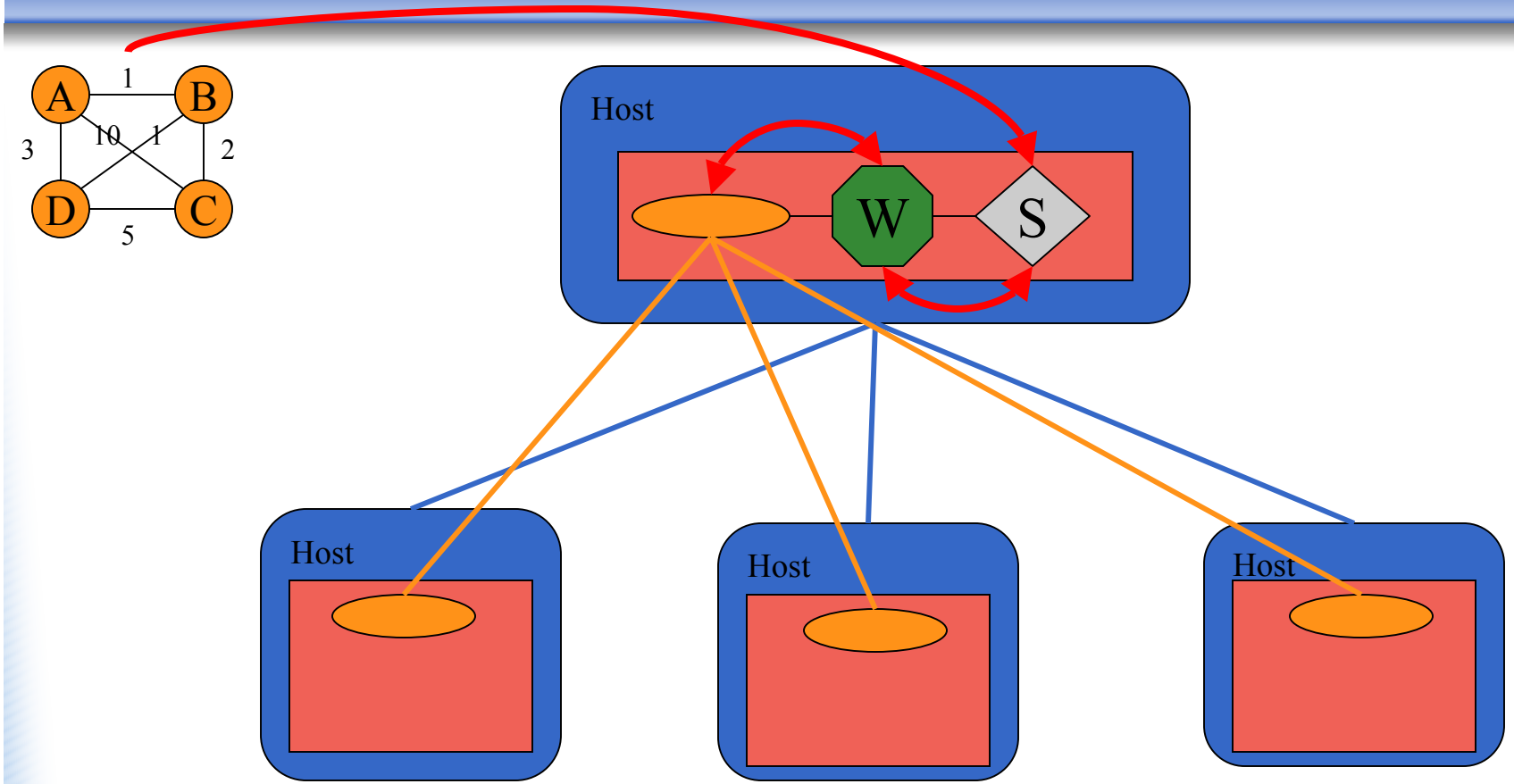
Model's Experimentation
Traveling Salesman Problem



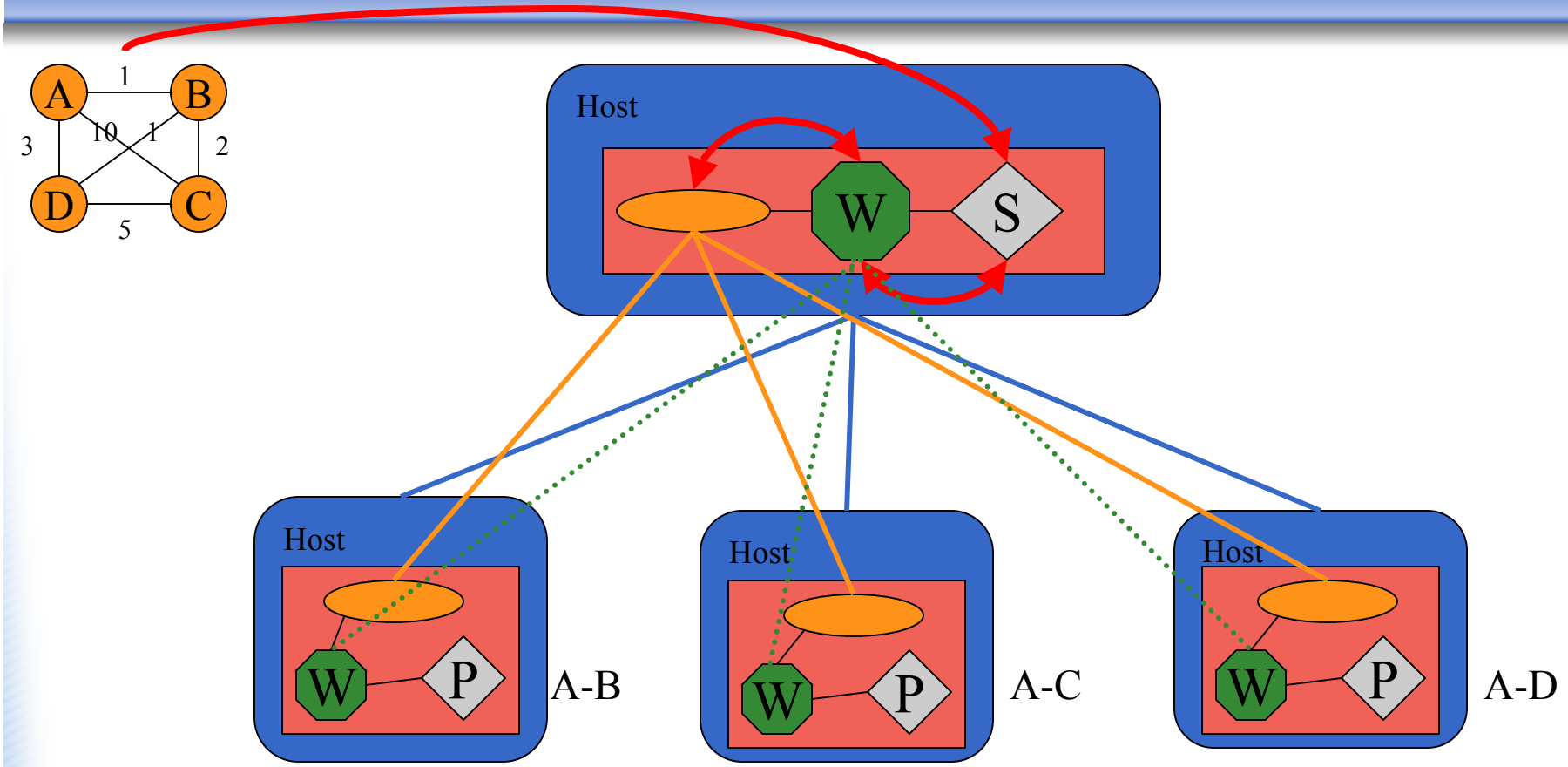
Model's Experimentation
Traveling Salesman Problem



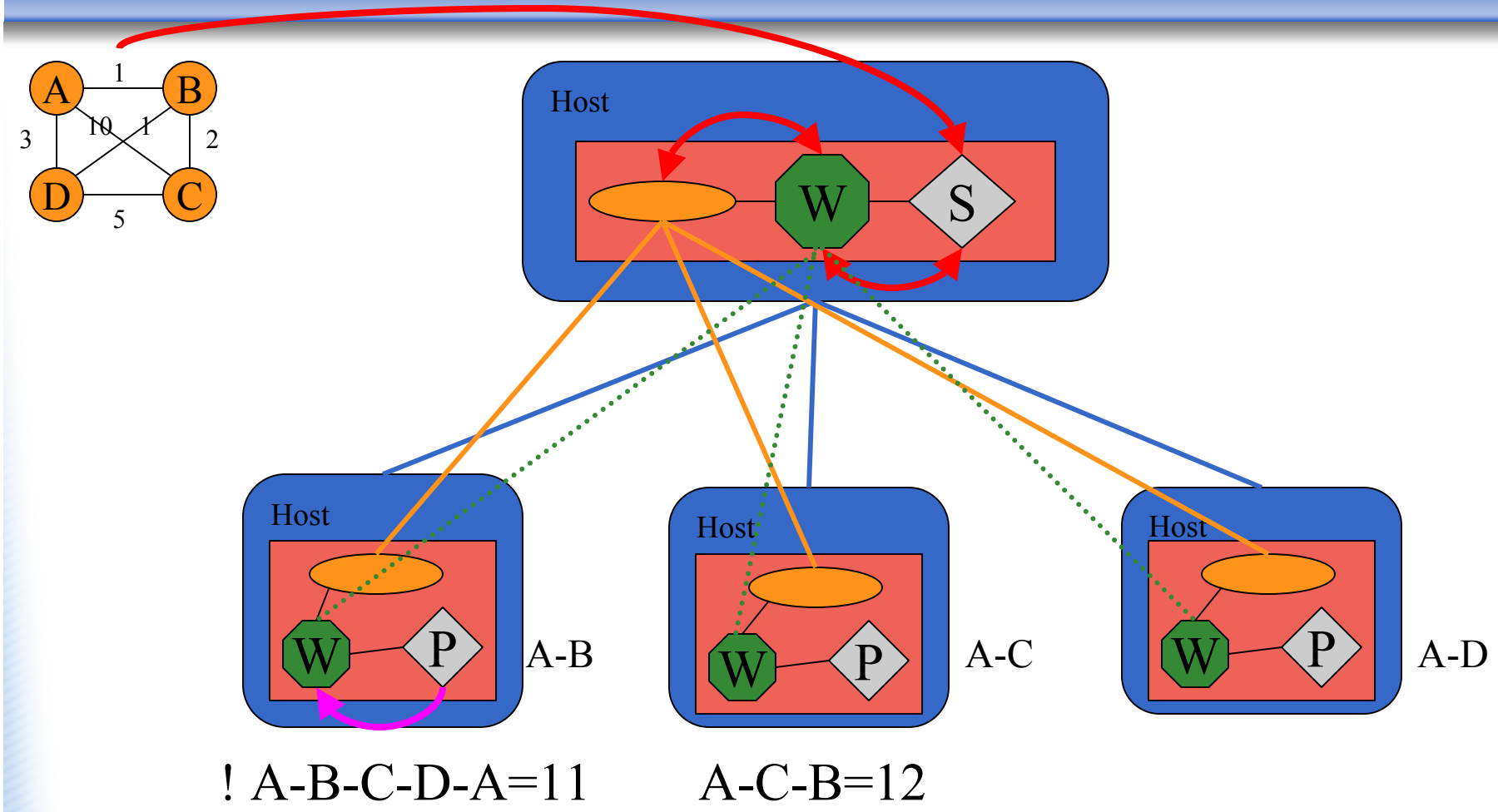
Model's Experimentation
Traveling Salesman Problem



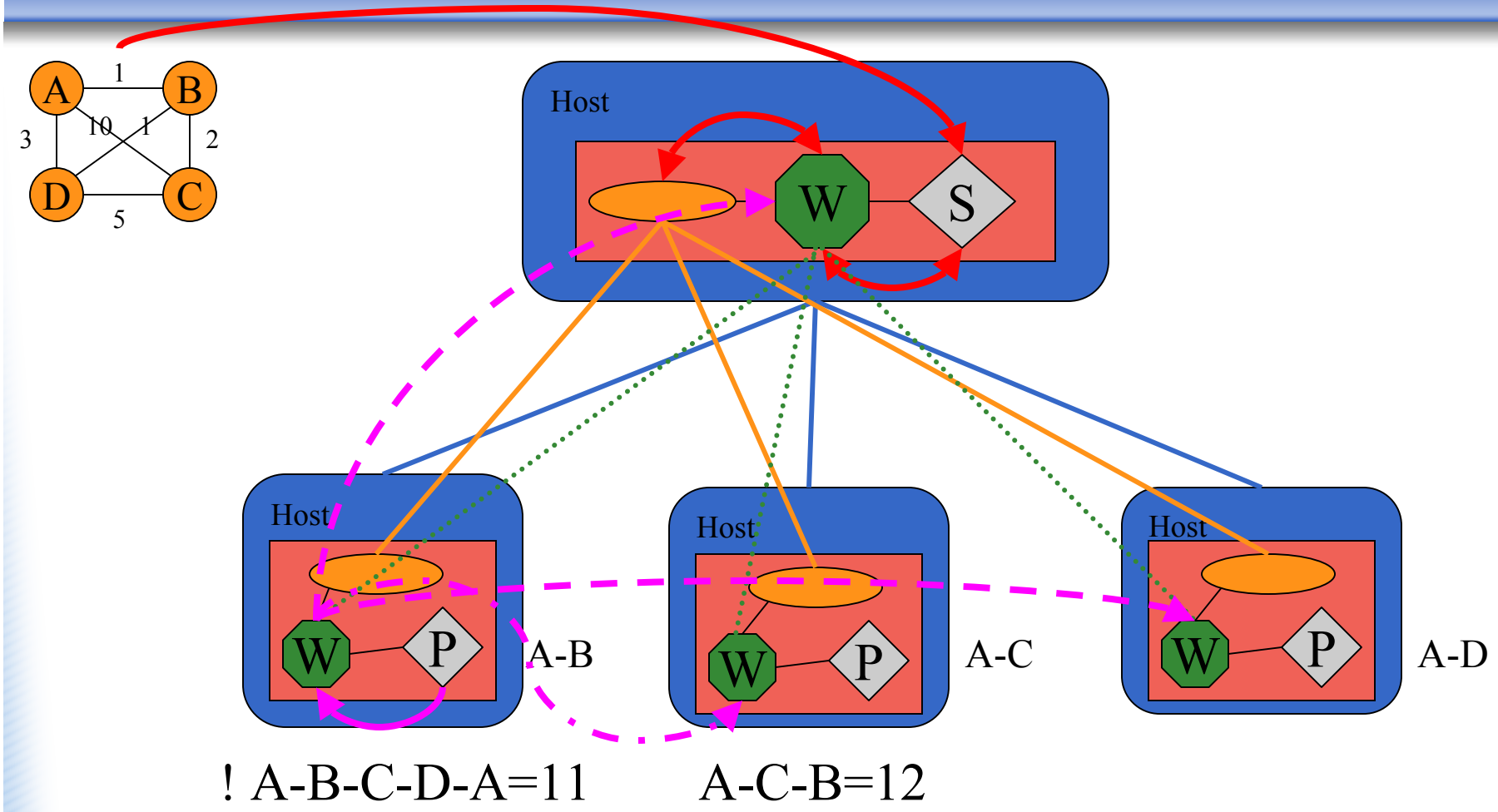
Model's Experimentation Traveling Salesman Problem



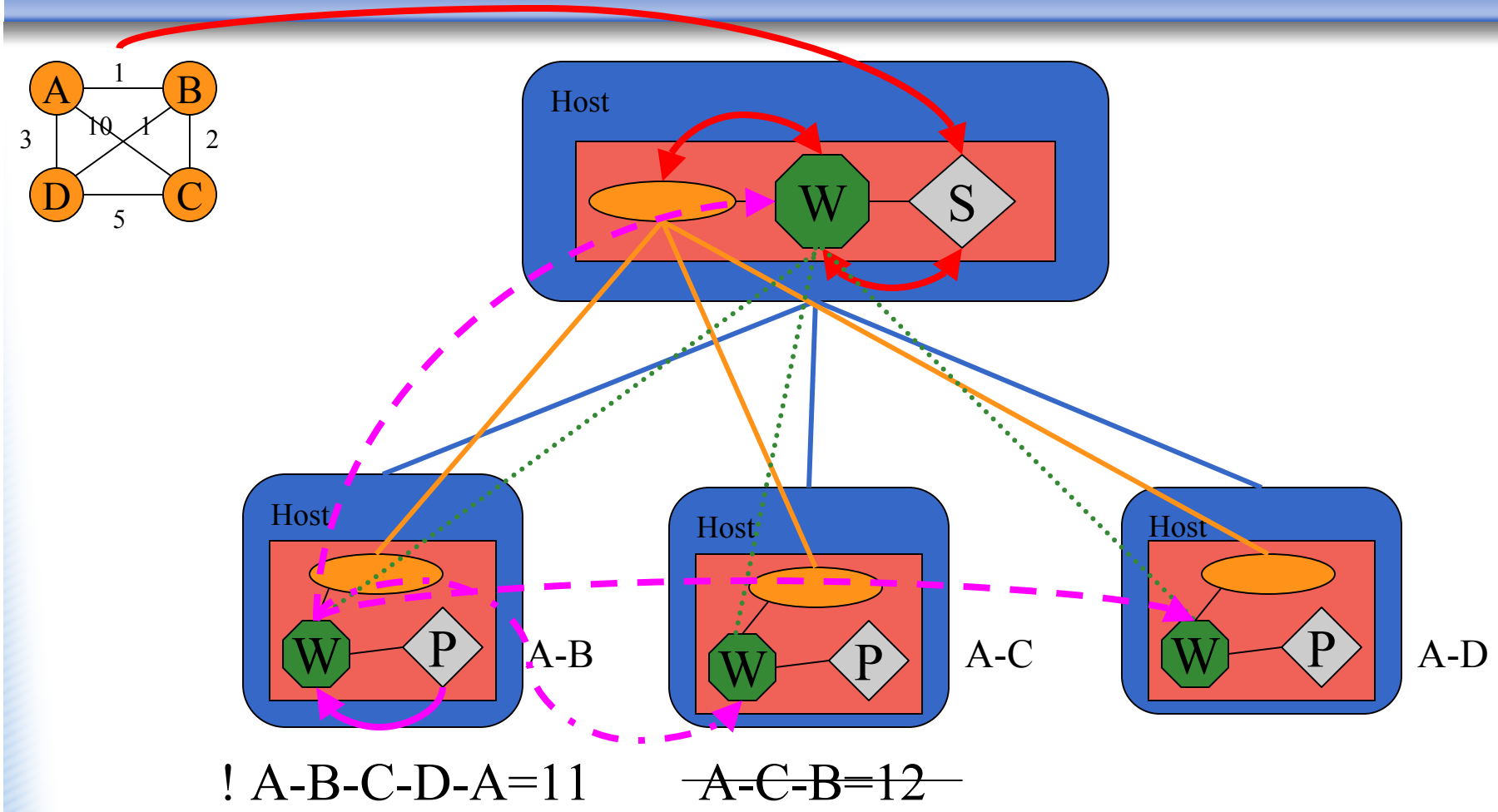
Model's Experimentation Traveling Salesman Problem



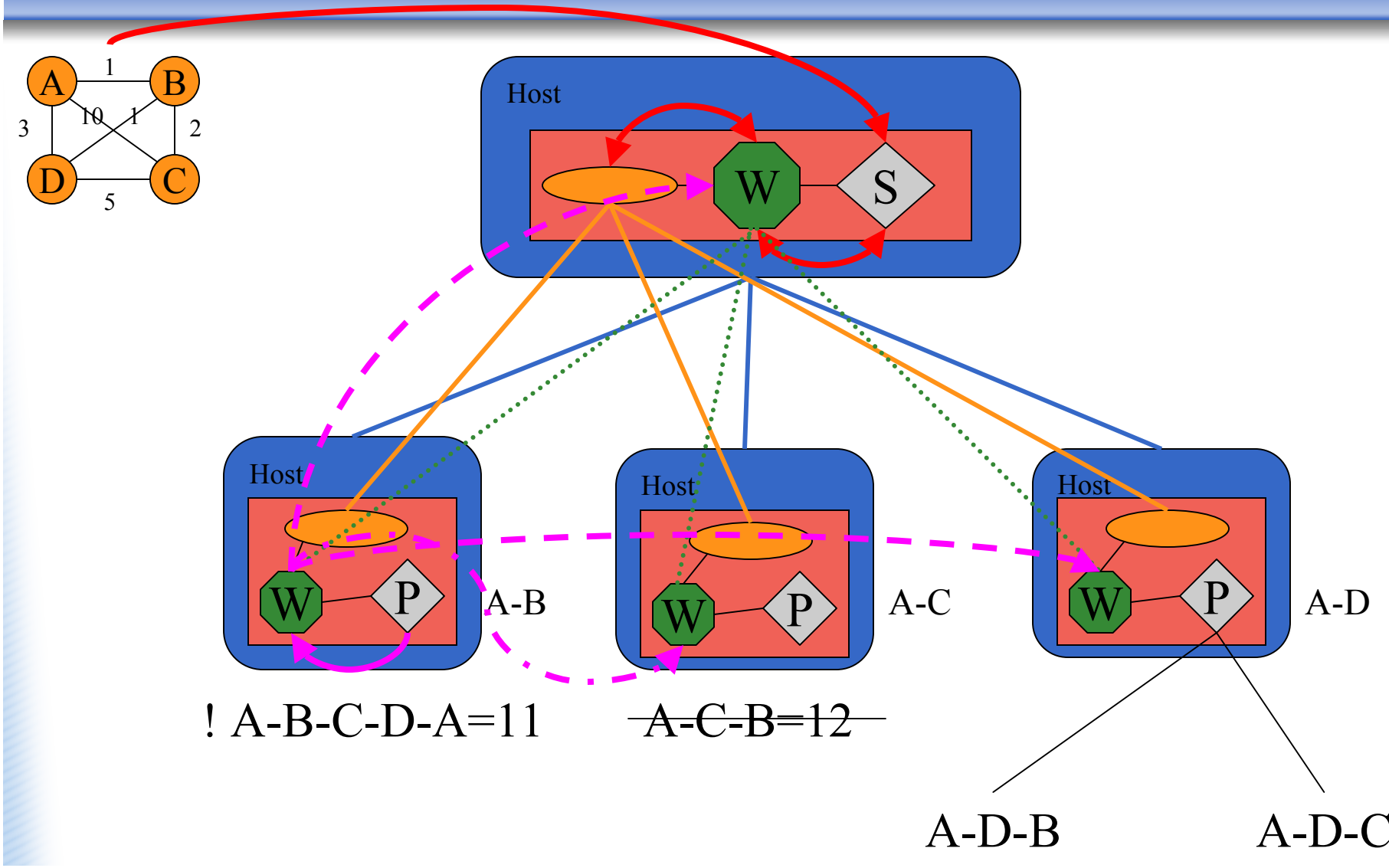
Model's Experimentation Traveling Salesman Problem



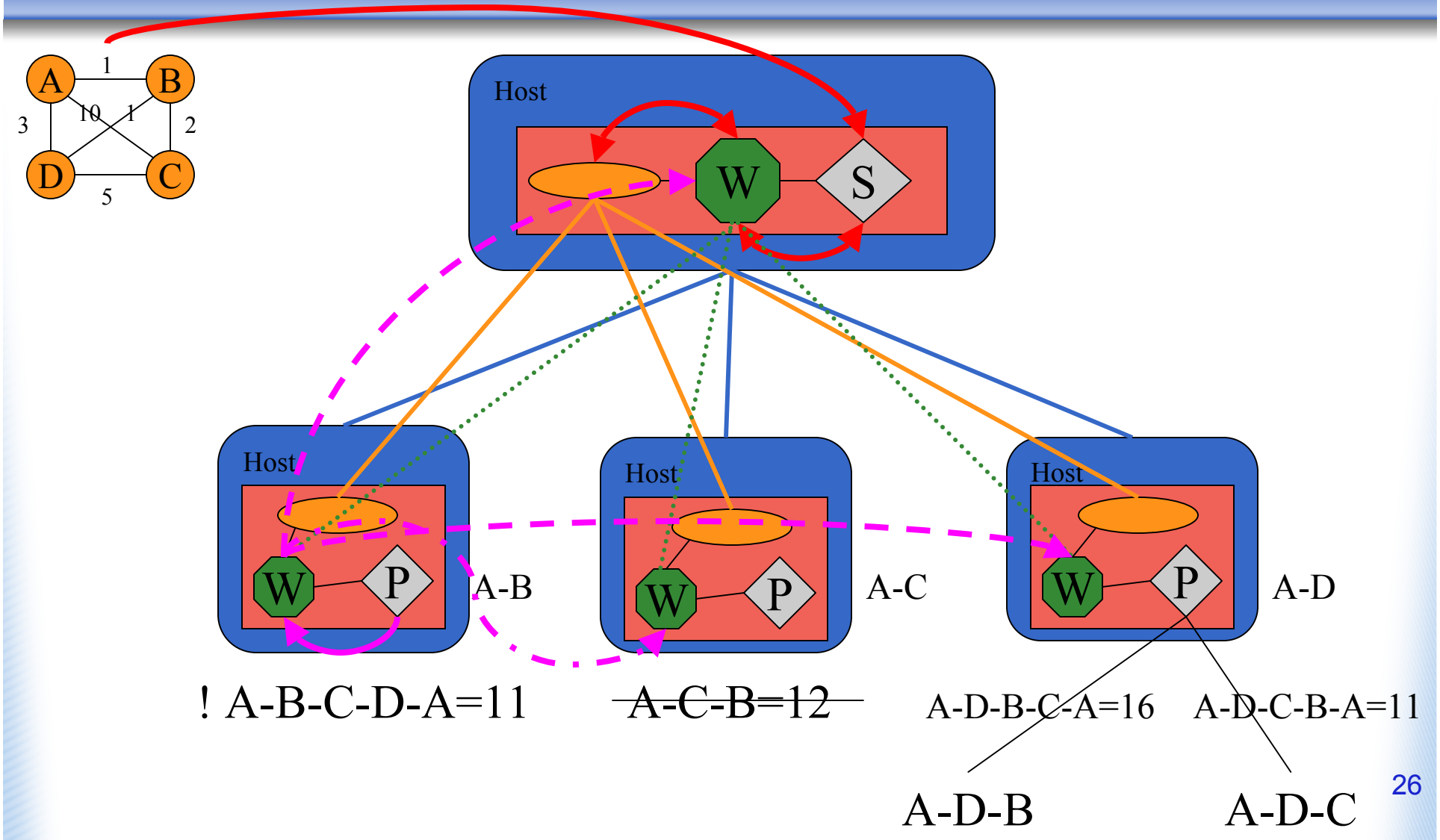
Model's Experimentation Traveling Salesman Problem



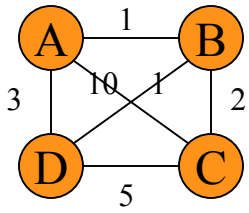
Model's Experimentation Traveling Salesman Problem



Model's Experimentation Traveling Salesman Problem



Model's Experimentation Traveling Salesman Problem



Host

W

S

Host

W

P

A-B

Host

W

P

A-C

Host

W

P

A-D

A-D-C-B-A=11

! A-B-C-D-A=11

~~A-C-B=12~~

A-D-B-C-A=16

A-D-C-B-A=11

A-D-B

A-D-C

Evaluation

- In relation to related works :
  - InteGrade
    - No static organization :

      Self-organizing
  - XtremWeb
    - Tasks communications
    - Synchronizations
  - JXTA
    - Infrastructure ready to use and
    - Model with programming API
  - ProActive
    - P2P dynamic deployment,
    - Computational and parallel P2P programming API

Conclusion

- Contributes :
  1. P2P Infrastructure of JVMs
     - Tunable (TTU, TTL et NOA)
     - Self organizing
     - 1st real P2P infrastructure for computational
  2. Model for P2P programming
     - Dynamic programming P2P (B&B)
     - 1st model with communications between tasks
  3. Implementations and Experimentations

- Perspectives :
  - Large scale tests (500, …, 5000)
  - Simulations or emulations
  - Modeling: Jackson's networks, Kelly's networks
  - Integrate some search methods : BFS, DFS