4VP+: A Novel Meta OS Approach for Streaming Programs in Ubiquitous Computing

Yaoxue Zhang, Yuezhi Zhou

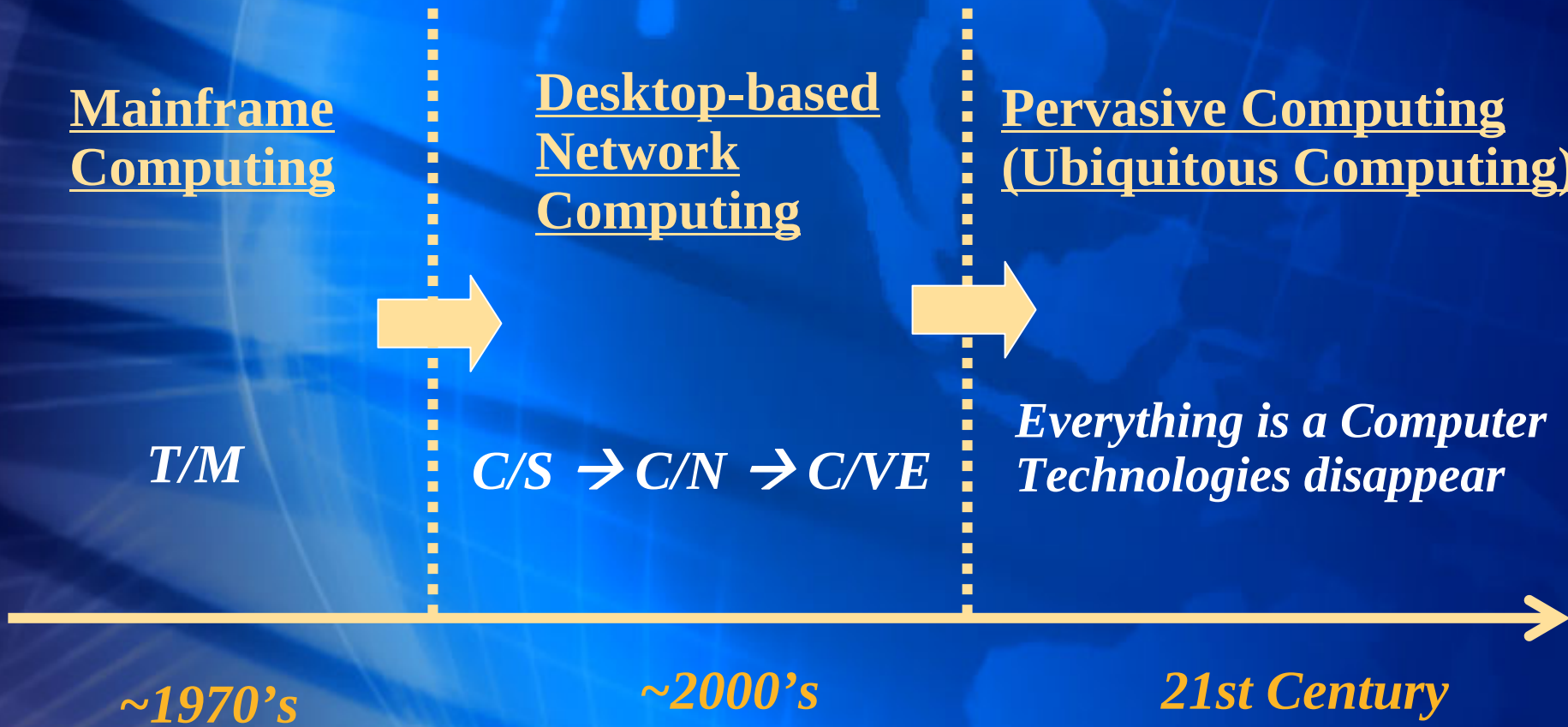
Tsinghua University

Contents

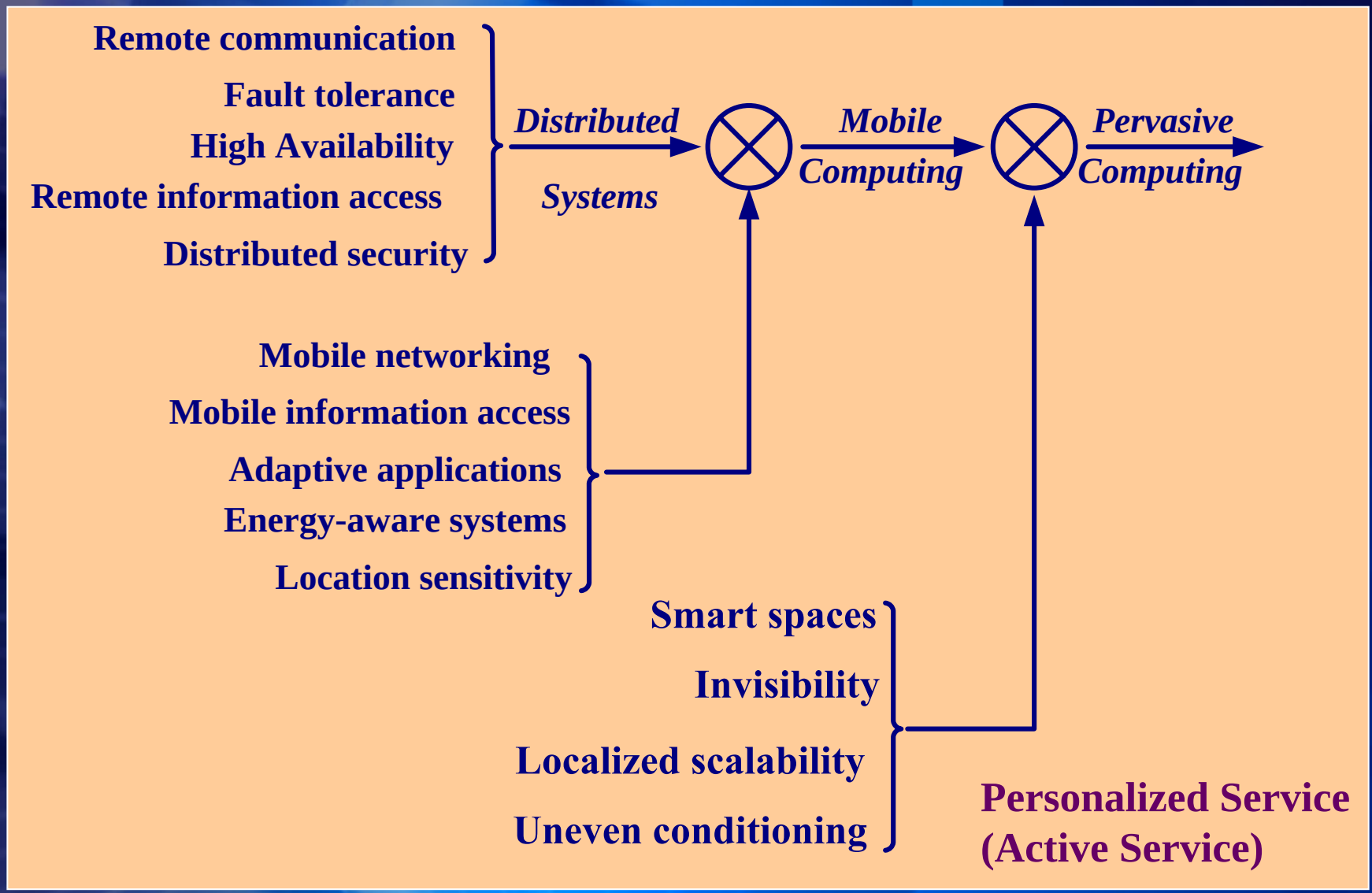
1. Introduction
2. Overview of Transparent Computing
3. Concept and architecture of Meta OS
4. Program streaming through 4VP+
5. System implementation & evaluation
6. Conclusions

1. Introduction

1.1 Change of Computing Paradigms



1.2 The Characteristics of Ubiquitous Computing

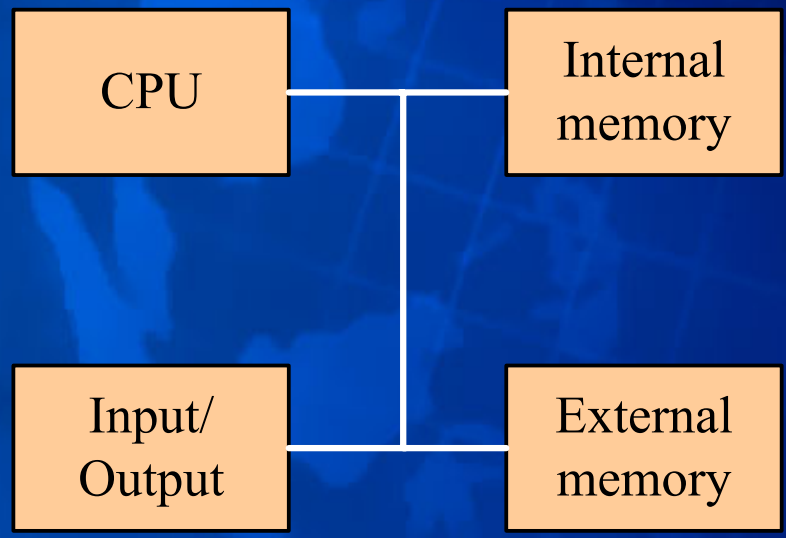


(From M. Satyanarayanan, 2001)

1.3 The concept of traditional stored program

"store its instructions in its internal memory and process them in its arithmetic unit, so that in the course of a computation they may be not just executed but also modified at electronic speeds."

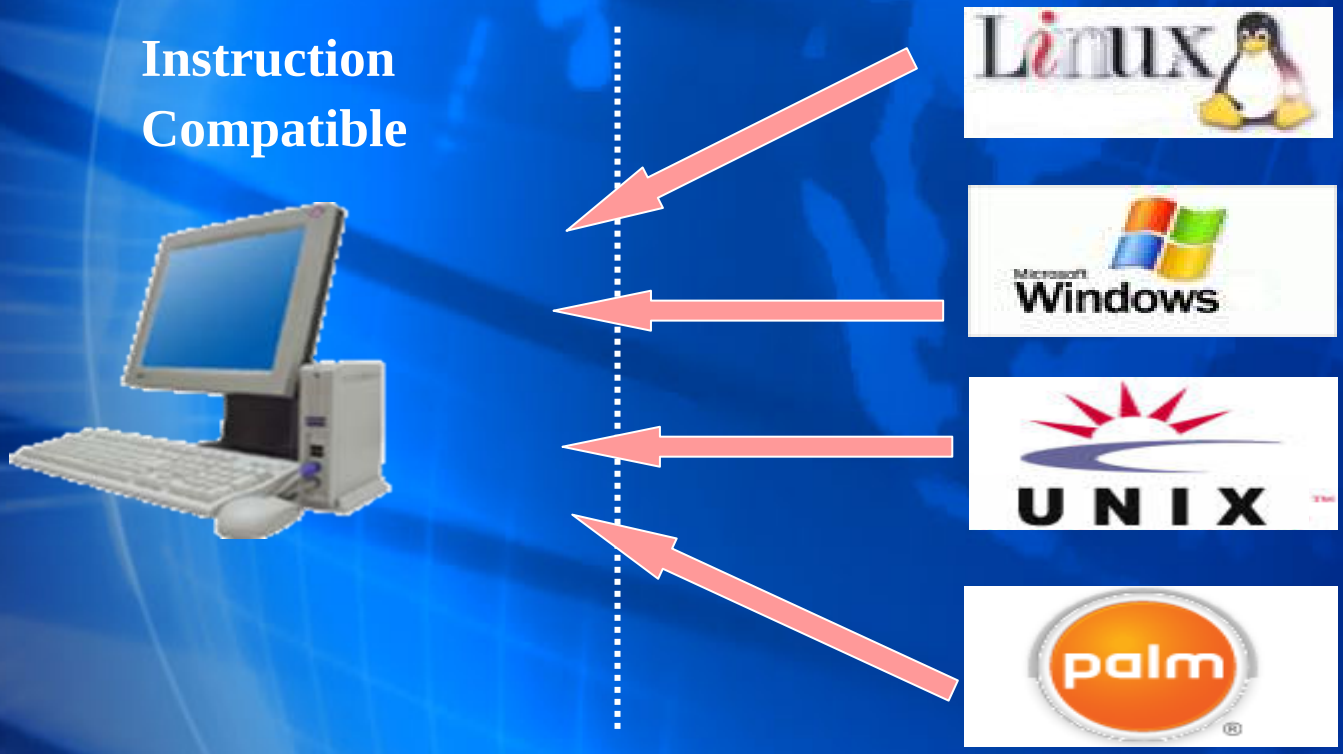
By Von Neumann 1945
(From "The stored program concept", Aspray. W. IEEE 1990)



1.4 Some problems to realize Ubiquitous Computing in traditional computer systems

- Users can not choose different OSes and applications in the same hardware platform

- **Regular OS can not be installed on light-weight devices, Embedded OS is not so sensitive to needs of users**



2. Overview of Transparent Computing

2.1 Transparent Computing Paradigm

- Users do not care the technical details of the computing; they only focus on the service provided!
- Users can select services based on heterogeneous OSes from the same hardware platform

- Execution and Storage of Programs are separated in different computers

Execution: light-weight devices or clients (as assembling factory)

Storage: Servers (as warehouse)

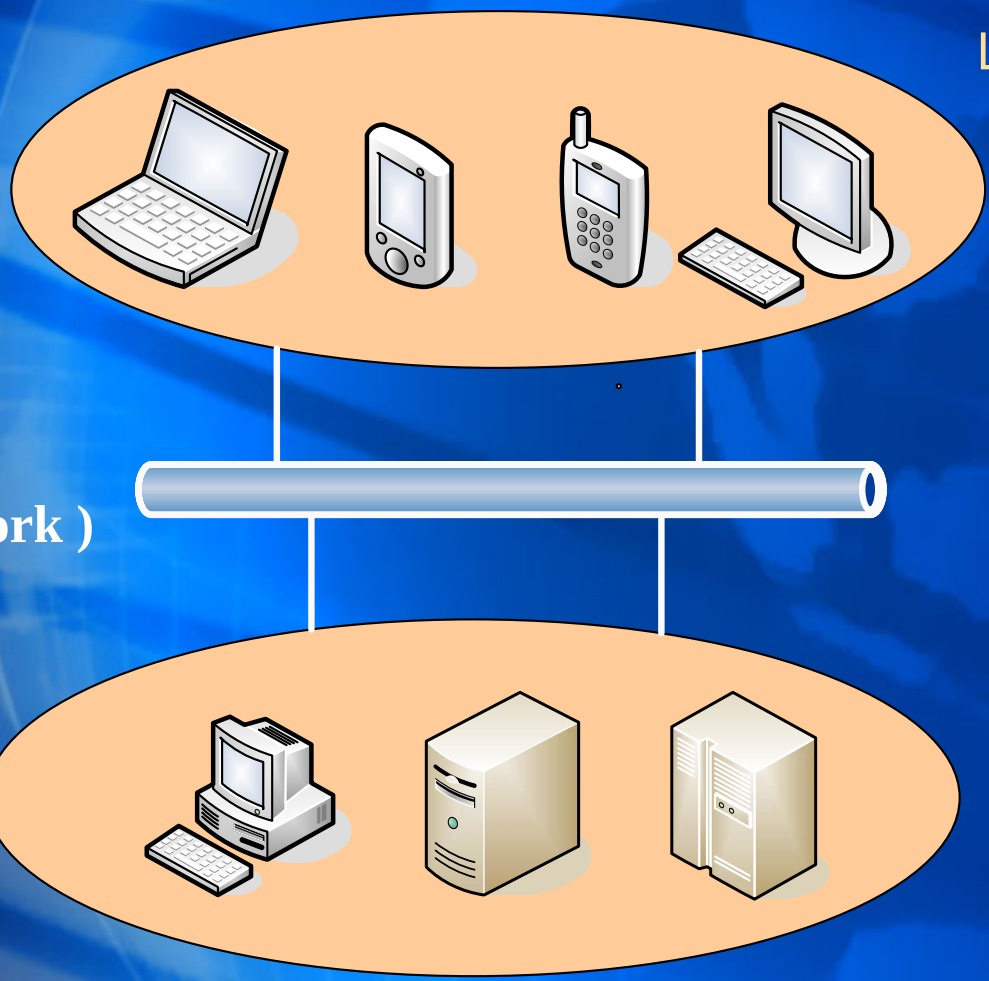
→JITC (Just In Time Computing)

2.2 The D&T extended Von Neumann architecture

- **The basic considerations:**
  - *Users only need services*
  - *Why users can not get smart services?*
    - → *Storage and execution have not been separated in traditional computers, so that the computers become more complex with applications increasing*
- **Separated Storage and execution of programs into different computers:**
  - *Server computer: Storage and management*
  - *Light-weight client: I/O and execution*
  - *Network: connecting servers and clients (Similar to bus in traditional computers)*
  - *Program streaming: block-streaming between server and clients according to interruption or I/O requirement*

2.3 Topology of Transparent Computing Paradigm

TC
(Transparent Clients)



Light-weight devices:

PC, PDA, Intelligent Mobile Phone, Digital Appliance, Dedicated Clients, et al.

TDN
(Transparent Delivery Network)

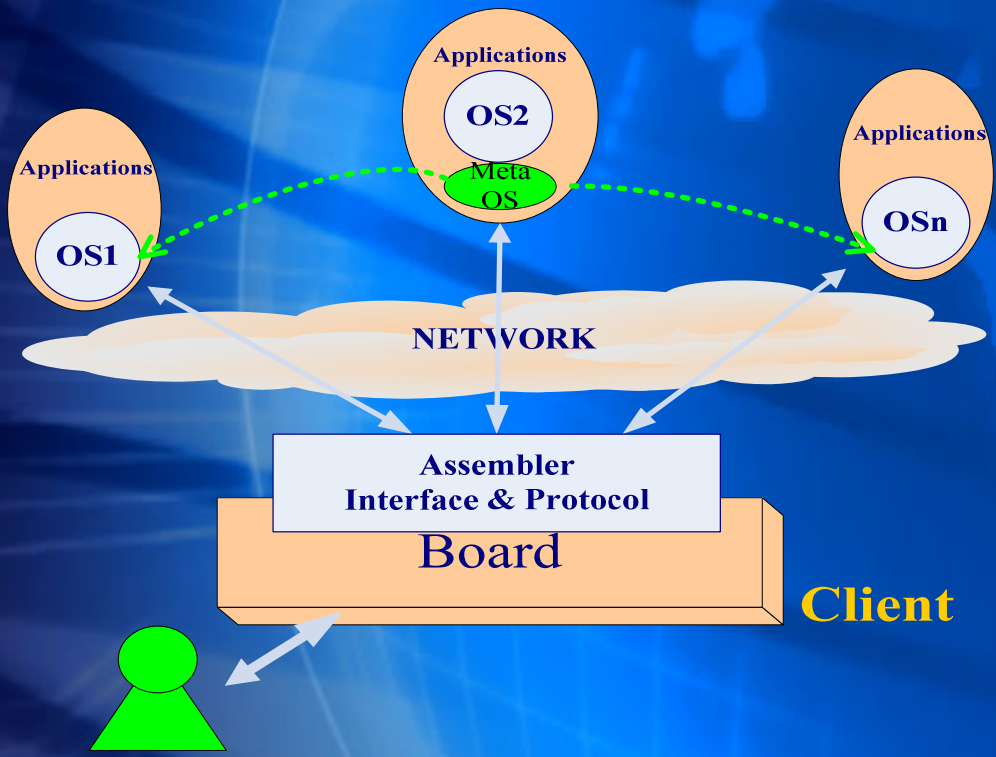
Ethernet, CATV, 802.11, IEEE 1394, et al.
(Ubiquitous Communication)

TS
(Transparent Servers)

Regular PC, PC Server, et al.

3. Concept and architecture of Transparent Computing

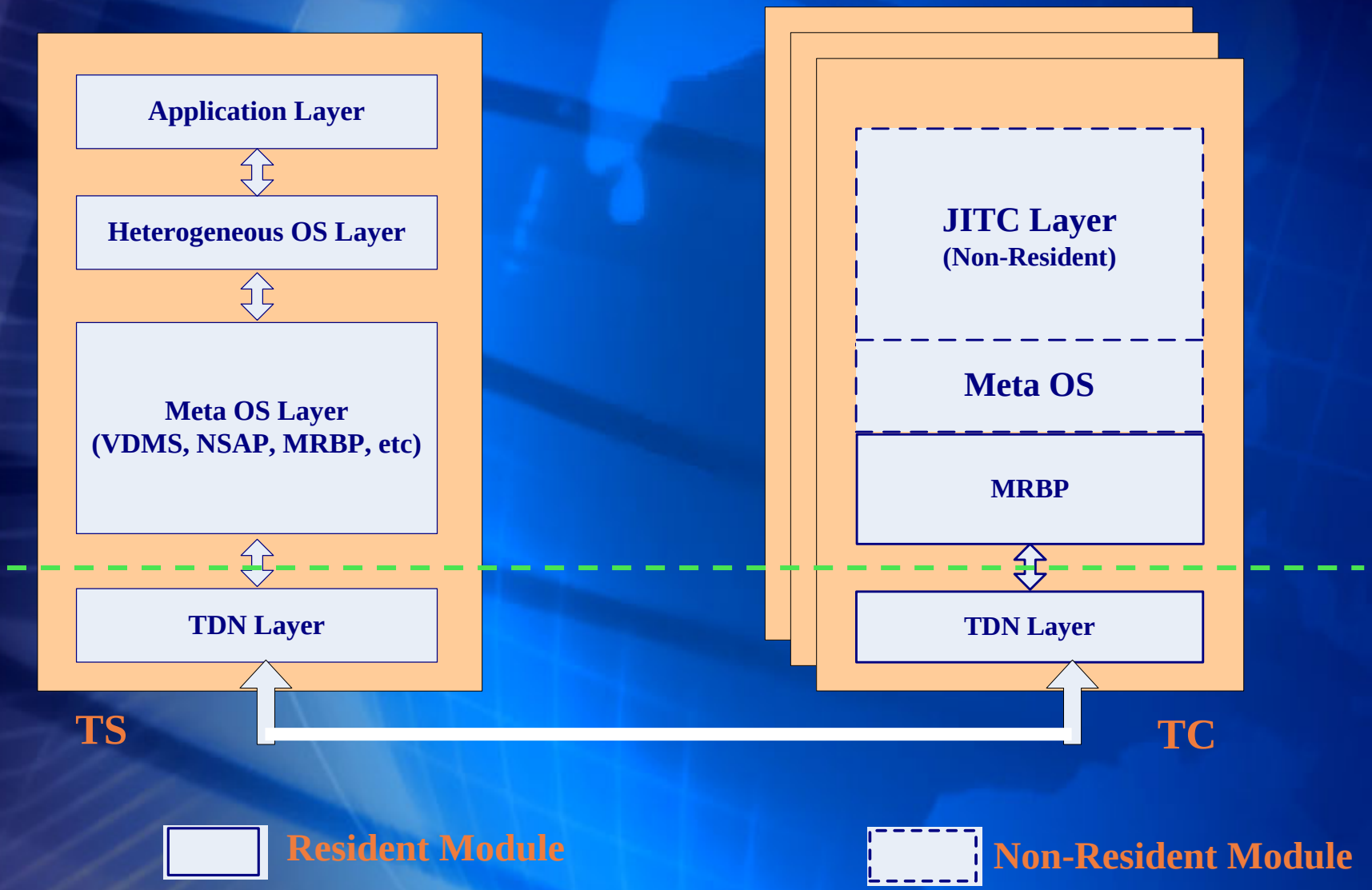
3.1 Execution environment of Meta OS



- **Instance OS:** Traditional commercial OS running on traditional computers, such as Linux, Windows, etc.
- **Meta OS:** A program to control Instance Oses.

- Users choose instance OSes & Applications Using light-weight devices.
- Meta OS schedules services to Clients to execute.

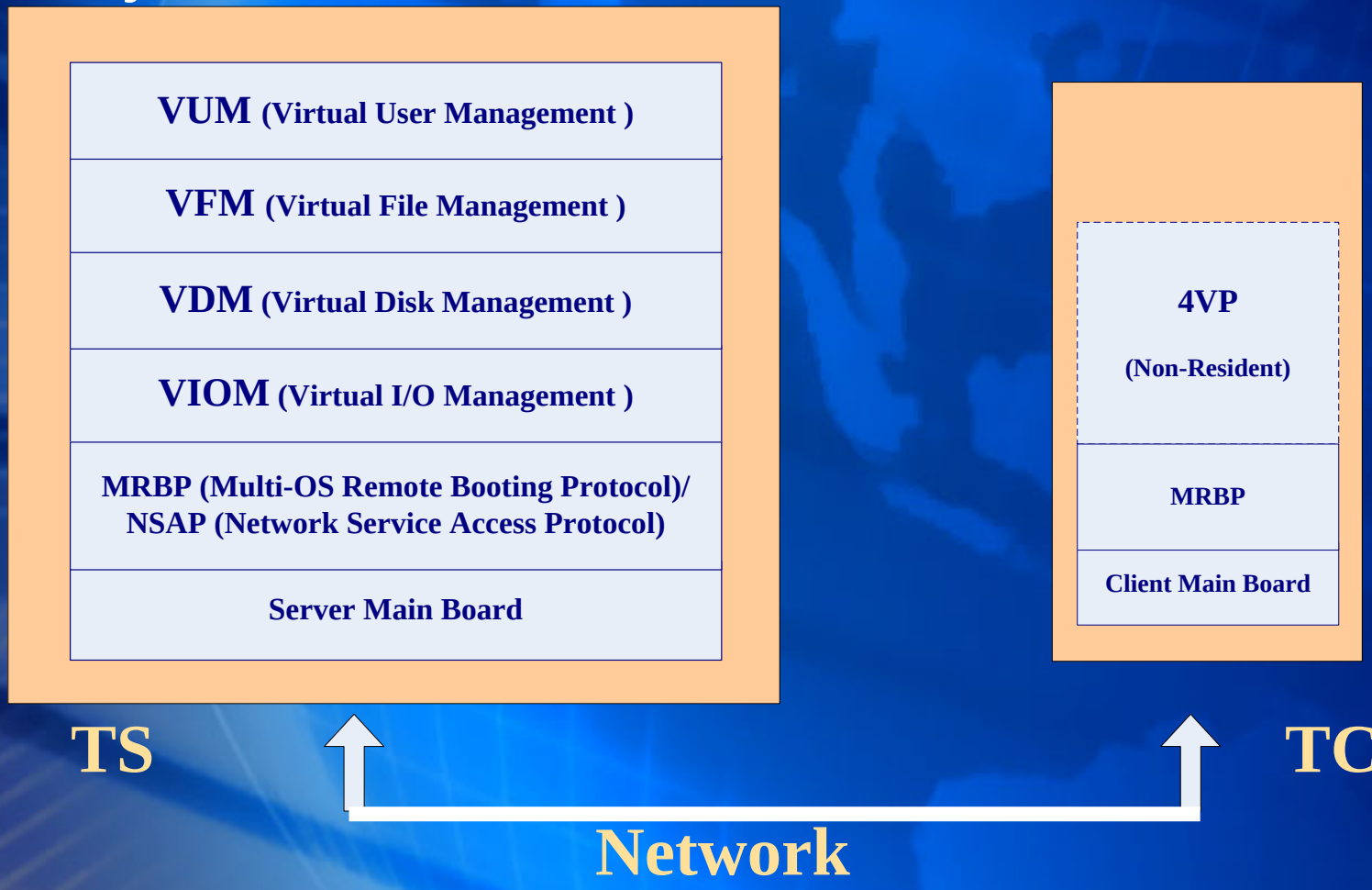
3.2 Layered Architecture of Meta OS environment



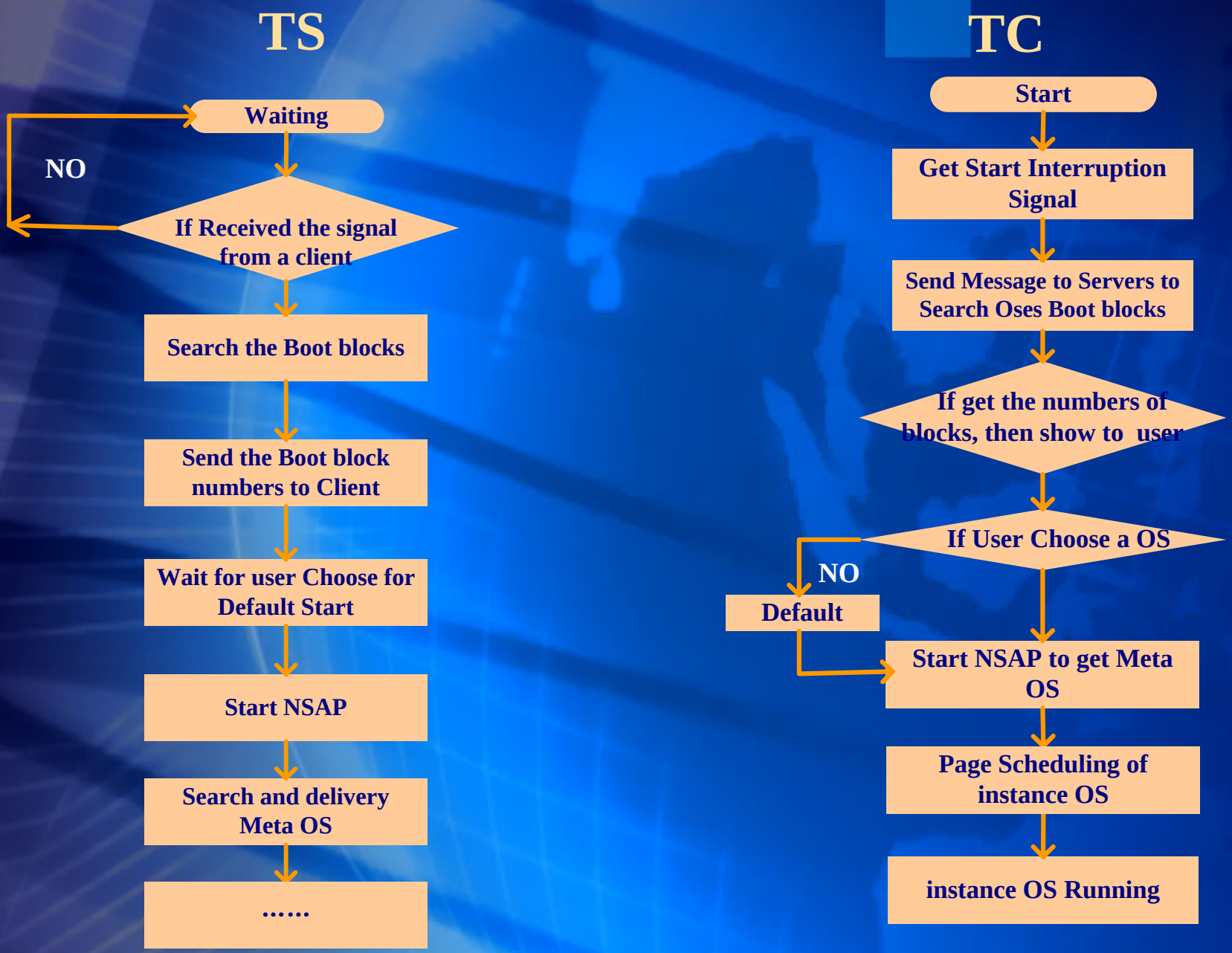
4. Programs Streaming through 4VP+

4.1 The Modules of Meta OS

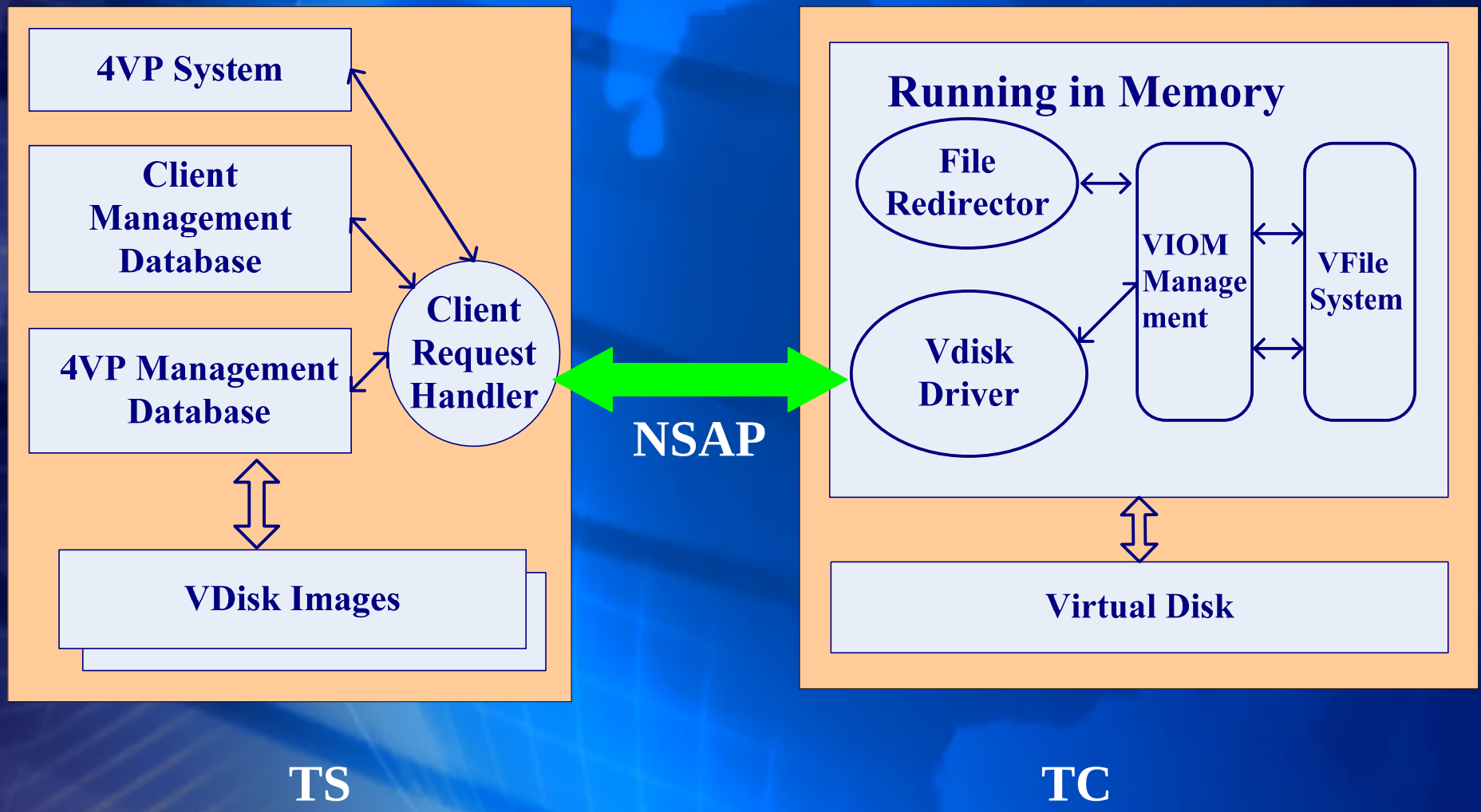
4VP+ System



4.2 MRBP: Start Delivery and Booting of OSes

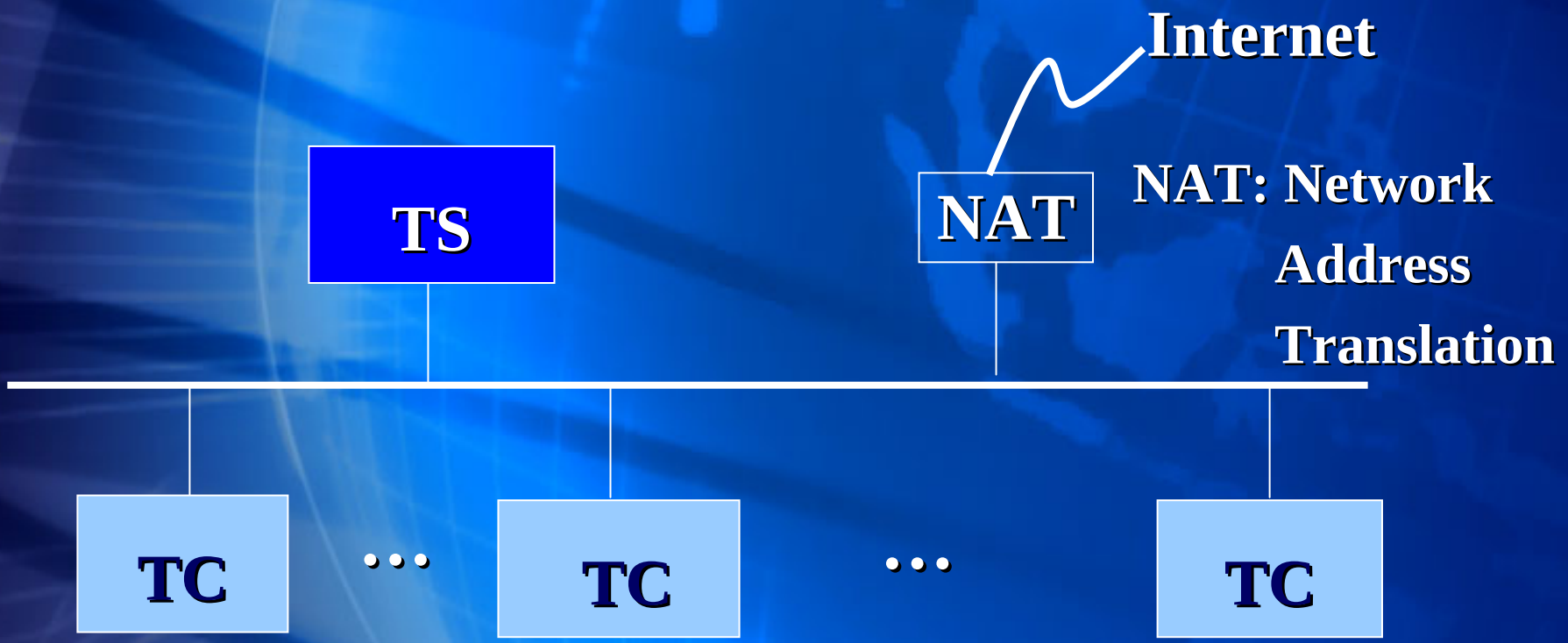


4.3 Network Service Access Protocol & Virtualization



5. System implementation & evaluation

5.1 Implementation Using C/S Model



TC:

- **Low Power:** <=15w
- **X86 Architecture:**
  Support Multimedia Instructions, and Windows, Linux
- **One-board Synthetic Design:**
  MPEG1, 3D/2D Graphical accelerator, IEEE1394, Ethernet, USB, TV-out, Fax/Modem, I/O, etc.
- **Low Cost:** about $100

"XiaoBao"

"WangRui"

"LongXing"

5.2 Performance Analysis

Testbed

- **Hardware Configuration**

  TC (30 sets)
  - Intel Celeron 1 GHz with 128 MB RAM
  - Onboard network card: 100 Mbps

  TS (3 sets)
  - AMD Athlon64 3000+ with 2 GB RAM
  - Hard Disk: Seagate SATA 7200 RPM, 2*80 G (RAID 0)
  - Onboard network card: 1 Gbps

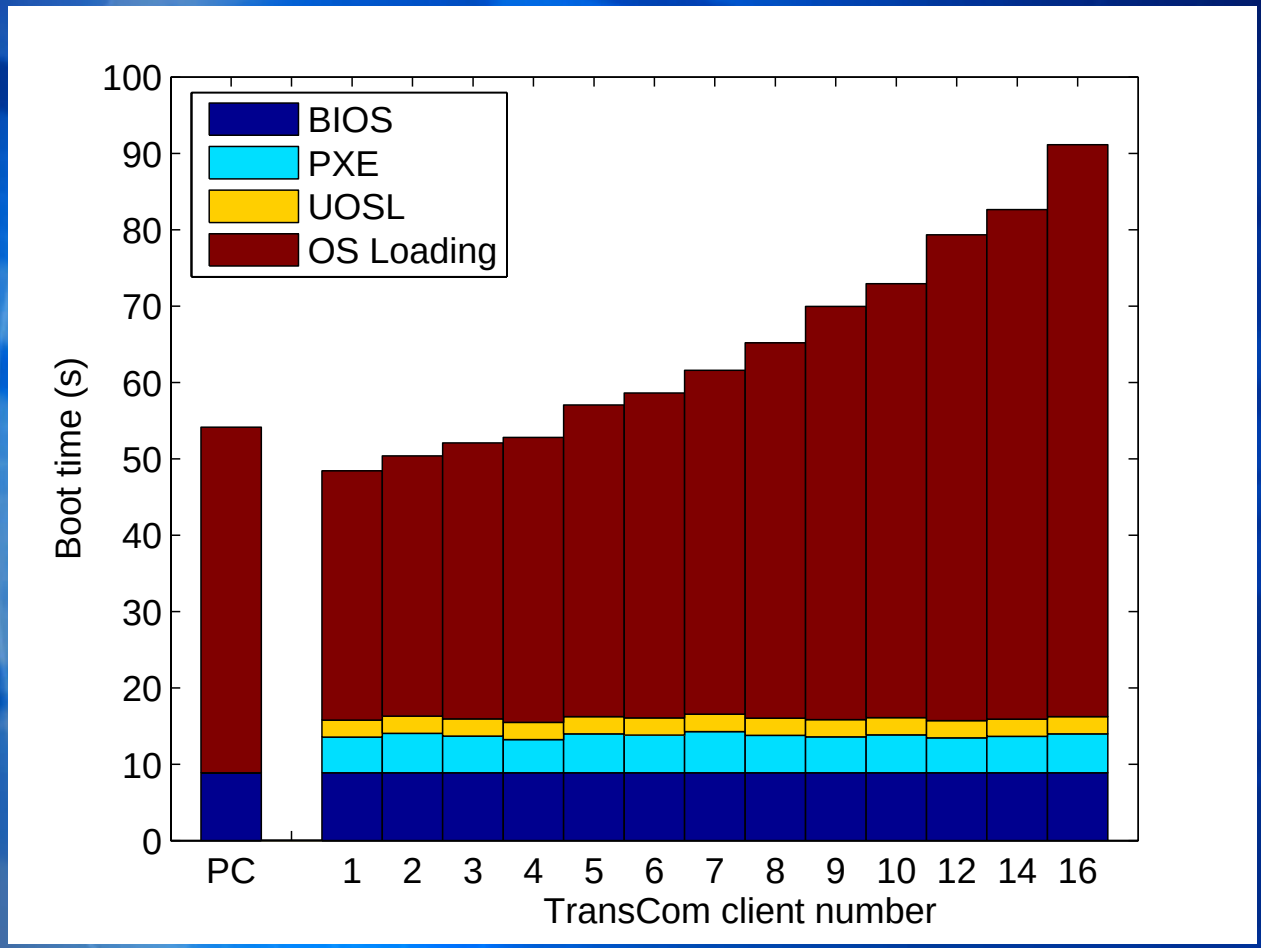
  TDN
  - Huawei-3Com Ethernet switch with 48 100 Mbps interfaces (for client) and 2 1 Gbps interfaces (for server)

- **Software Configuration**

  TS
  - Windows 98, Windows 2000, Linux and other application software

  TC
  - No *OS* and applications

Booting Performance



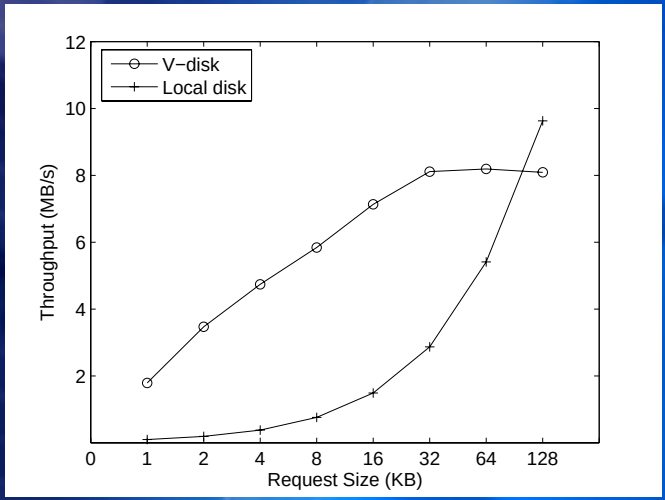
X-axis: Number of *TC*

Y-axis: Boot Time

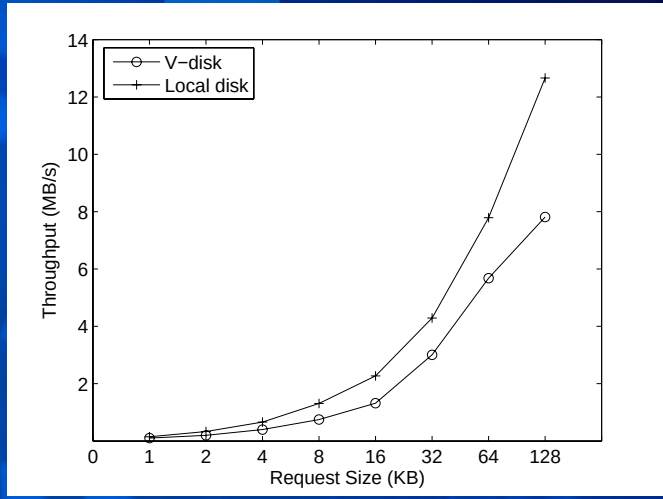
NSAP Throughput

X-axis: Request Size

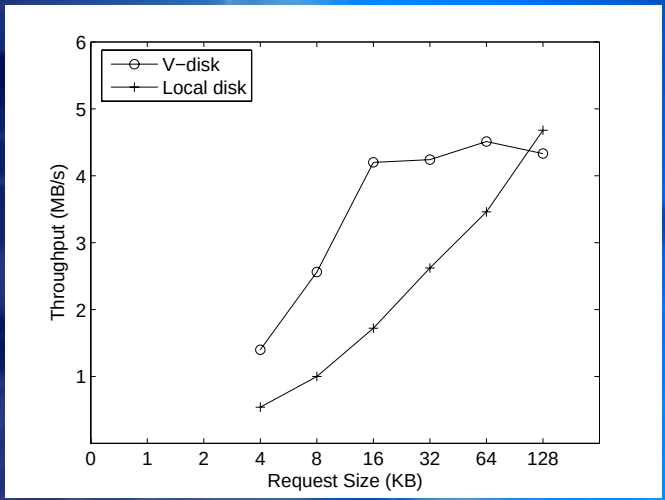
Y-axis: Throughput



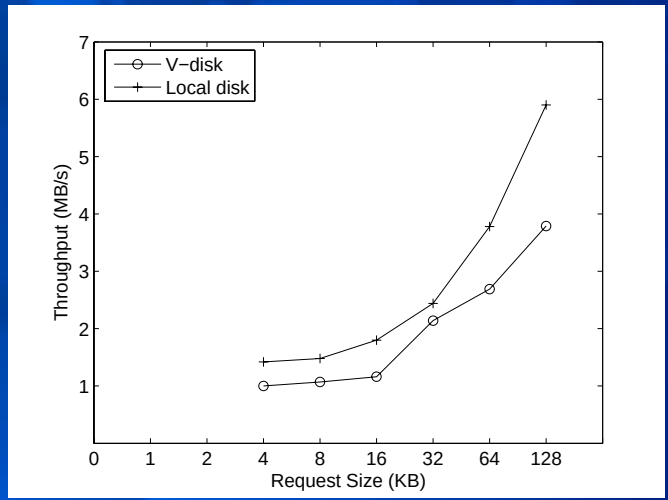
Read, unbuffered in Windows



Write, unbuffered in Windows



Read, unbuffered in Linux



Write, unbuffered in Linux

Function Evaluation (Start times in Windows 2000)

Applications/OS	1 PC	1 TC	10 TCs	28 TCs
Booting OS				
Windows2000 Server	53"13	48"73	70"62	142"57
Office Applications				
Word 2003	2"23	1"26	2"28	11"50
Image processing applications				
PhotoShop V7.0	13"29	11"08	16"48	1'0"51
Flash V6.0	18"62	7"16	31"41	1'16"56
3D MAX V8.0	29"71	25"68	34"24	1'16"56
Copying files				
	28"24	24"33	49"48	4'6"99
Playing multimedia				
Windows Media Player	smoothly	smoothly	smoothly	smoothly

Function Evaluation (Start times in Linux Redflag 4.1)

Applications/OS	1 PC	1 TC	10 TCs	28 TCs
Booting OS				
Red Linux 4.1	85"67	58"20	97"72	153"72
Office Applications				
EIOffice	2"03	1"07	2"35	10"43
Internet Explore				
Firefox	3"67	2"55	3"89	10"60
Copying files				
20MB	12"25	7"80	18"65	53"15
50MB	33"83	25"44	53"76	257"79

Applications

6. Conclusions

- We proposed a new OS: Meta OS and Transparent Computing for Ubiquitous Computing
- Many issues need to be researched further more, e.g.:
  - How to support more *OSes*?
  - How to support more devices?
  - How to extend to more applications?
  - How to support different communication networks?

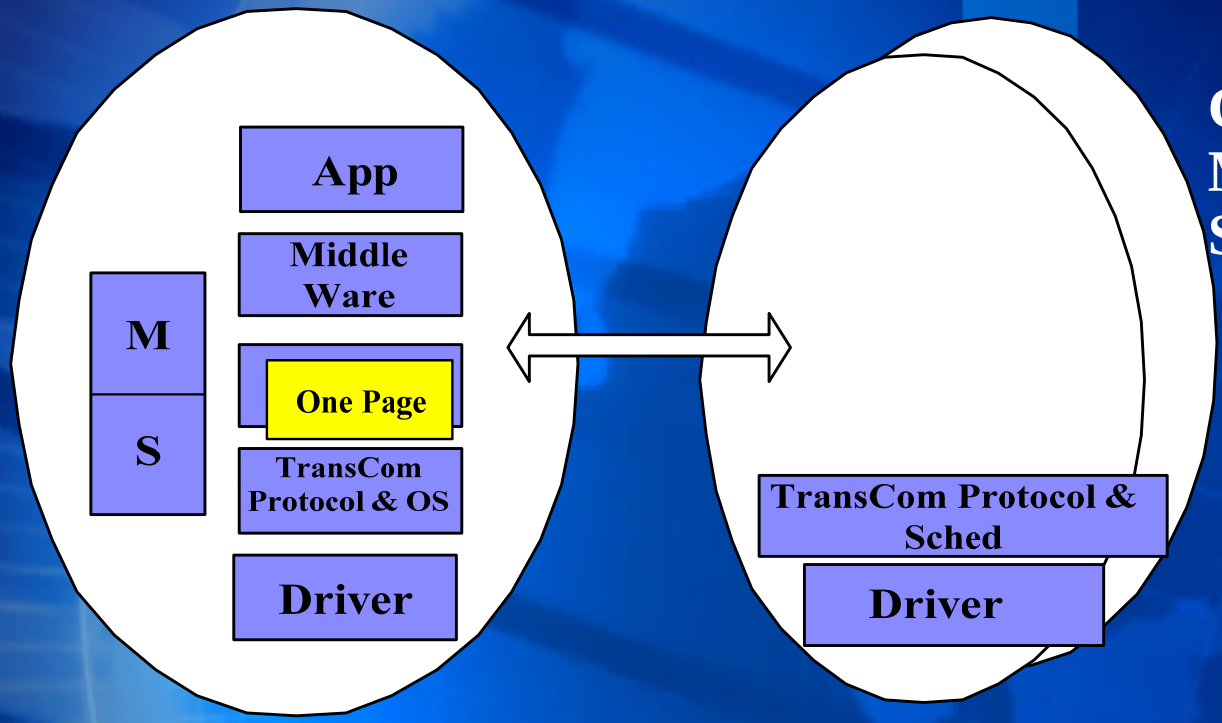
Thanks!

Q&A

Stored Program

Consider data and instruction as the same

- ***Stored Program*** **has realized in the single computer system by now, stored data has realized in network.**
- **have not realized storage of instructions in network environment.**



C: computing
M: management
S: storage

Server

Central Storage & Management

Client

Distributed Execution