POLITECNICO
MILANO 1863

IP LOOKUP ALGORITHMS

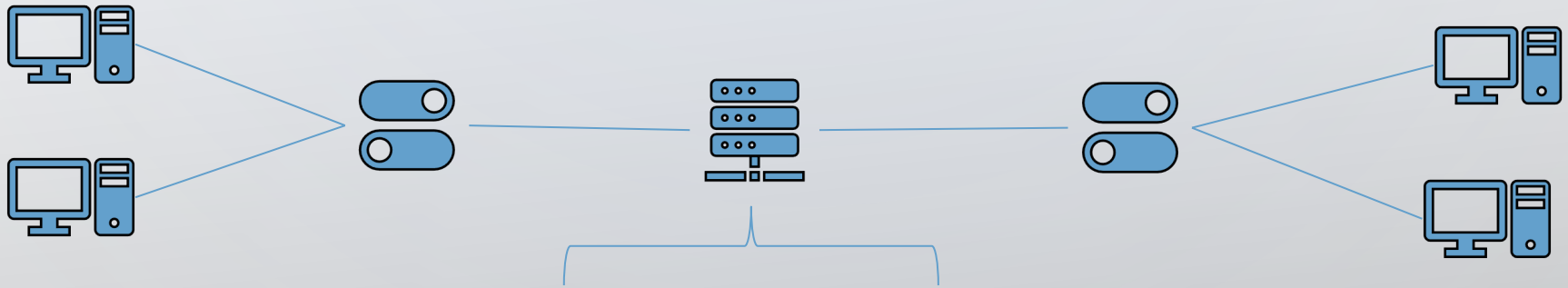
Algorithm implementation & Complexity analysis

Switching and Routing-2019/20 Professor: **Guido Alberto Maier** Tutor: **Sebastian Troia** **Group 2**

IP LOOKUP

POLITECNICO
MILANO 1863

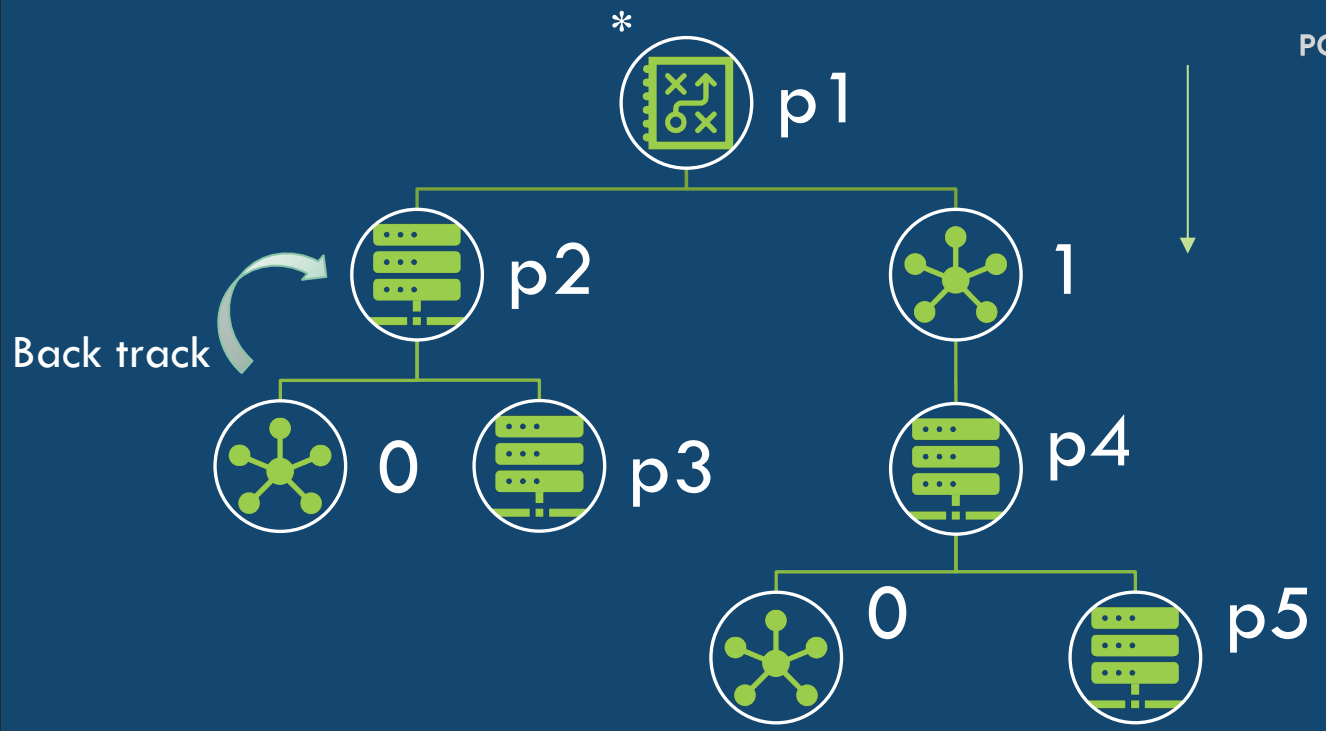
A router holds a routing table which contains prefix entries, and destination port.
When a router receives a packet, it searches for the longest prefix of the destination IP address in its routing table.



Network ID (Destination)	Netmask	Next hope	Interface	Metric

routing table

BINARY TREE



*** A multi-way tree with each node containing next-hop info and 0/1/2 pointers to child node(s)**

IMPLEMENTATION FUNCTIONS

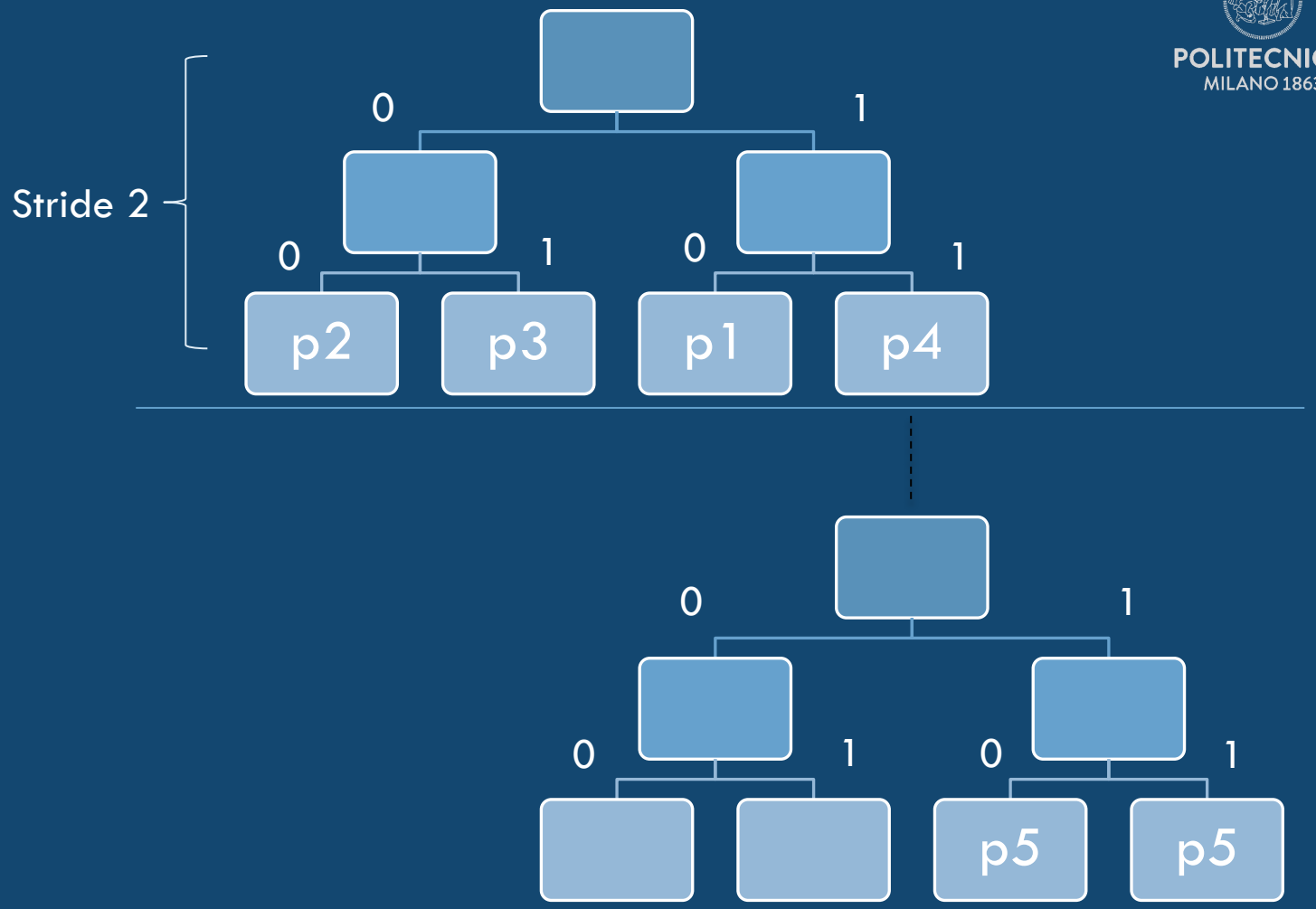
POLITECNICO MILANO 1863

Binary tree

AddChild	add child to each nodes reaching prefixes
Lookup (binary)	looking for the next hub for backtrack

Create	reading IP address from database Creating trie by calling AddChild
convert_in_bin	convert received IP to binary

MULTIBIT TREE



* Each node is a mini complete tree, with number of levels equal to the stride

IMPLEMENTATION FUNCTIONS

POLITECNICO MILANO 1863

Multibit tree

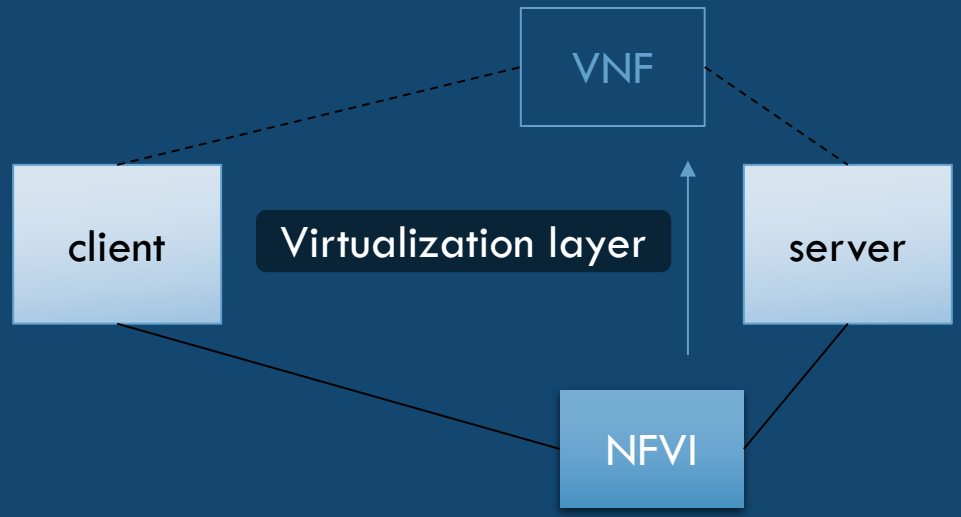
AddChild	add child to each nodes reaching prefixes
Lookup (multibit) stride	looking for the next hub for backtrack
GetCombinations	Combine the result of each stride

Create	reading IP address from database Creating trie by calling AddChild
convert_in_bin	convert received IP to binary

* Despite its advantage that lookup improved by N times when it is N bit and it has a disadvantage of using large memory space.

NFV SAR-APPLICATION

- A pre-developed sar-application has been modified to send IP addresses from a client virtual machine to a server virtual machine
- client side sar-application frequently sends IP addresses to be looked up on server side
- Server side sar-application calls unibit or multibit lookup algorithms to find the received IP address among the tree



SAR-APPLICATION

POLITECNICO
MILANO 1863

By running server-side sar-application we have two choices:

- Binary lookup
- Multibit lookup

```
Ready to monitoring
>>>>>>>>>>>>>>>>>>>
<<<<<<<<<<<<<<<<<<<
Select
 1- Binary lookup
 2- Multibit lookup :
2
Waiting for incoming request ...
```

After receiving every IP packets, Multibit lookup algorithm will run and starts looking up for the received IP address from client among its tree.

After each lookup, the consumed time will be printed on the screen also saved on a text document.

```
========================================
--- 4.41074371338e-05 Single Lookup ---
--- 0.292561531067 Sum of all Lookups ---
========================================
```

BINARY TREE COMPLEXITY

Performance

Lookup complexity: $O(W)$

Update complexity: $O(W)$

where

N : number of prefixes

W: maximum length (in bit) of the prefix

* A lookup by linear search would have complexity: O(N)

REAL RUN NFV IMPLEMENT **BINARY**

Numerical results under the following conditions

- Number of ips: 100 IPs
- From To search entries : 5000 IPs
- Sum of all lookup times: 0.031 s

```
===========
Lookup done.
--- Sum of binary_trie Lookup times = 0.03100419044494629 s ---
```

MULTIBIT TREE COMPLEXITY

Performance

- Lookup complexity: $O(\frac{W}{k})$

Lookup speed improves k times by a k-bit tree

- Update complexity: $O(W/k+2^k)$

Worst case: search through W/k nodes and access to each child of the last node (2k)

Where

W: maximum length of the prefix

k: length of lookup stride

REAL RUN NFV IMPLEMENT **MULTIBIT**

POLITECNICO MILANO 1863

Numerical results under the following conditions

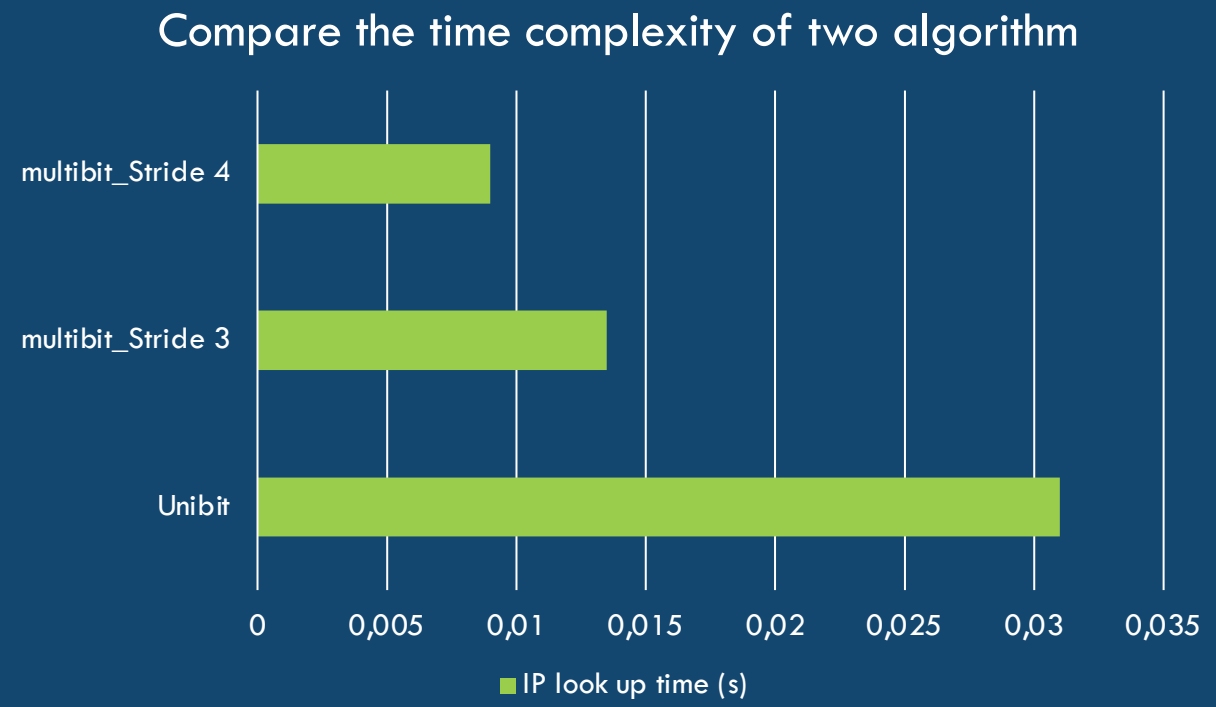
- Number of ips: 100 IPs
- From To search entries : 5000 IPs
- Stride : 3
- Sum of all lookup times: 0.0135 s

```
===========
Lookup done.
--- Sum of multibit-trie Lookup times = 0.013501167297363281 s ---
```

- Stride : 4
- Sum of all lookup times: 0.0090 s

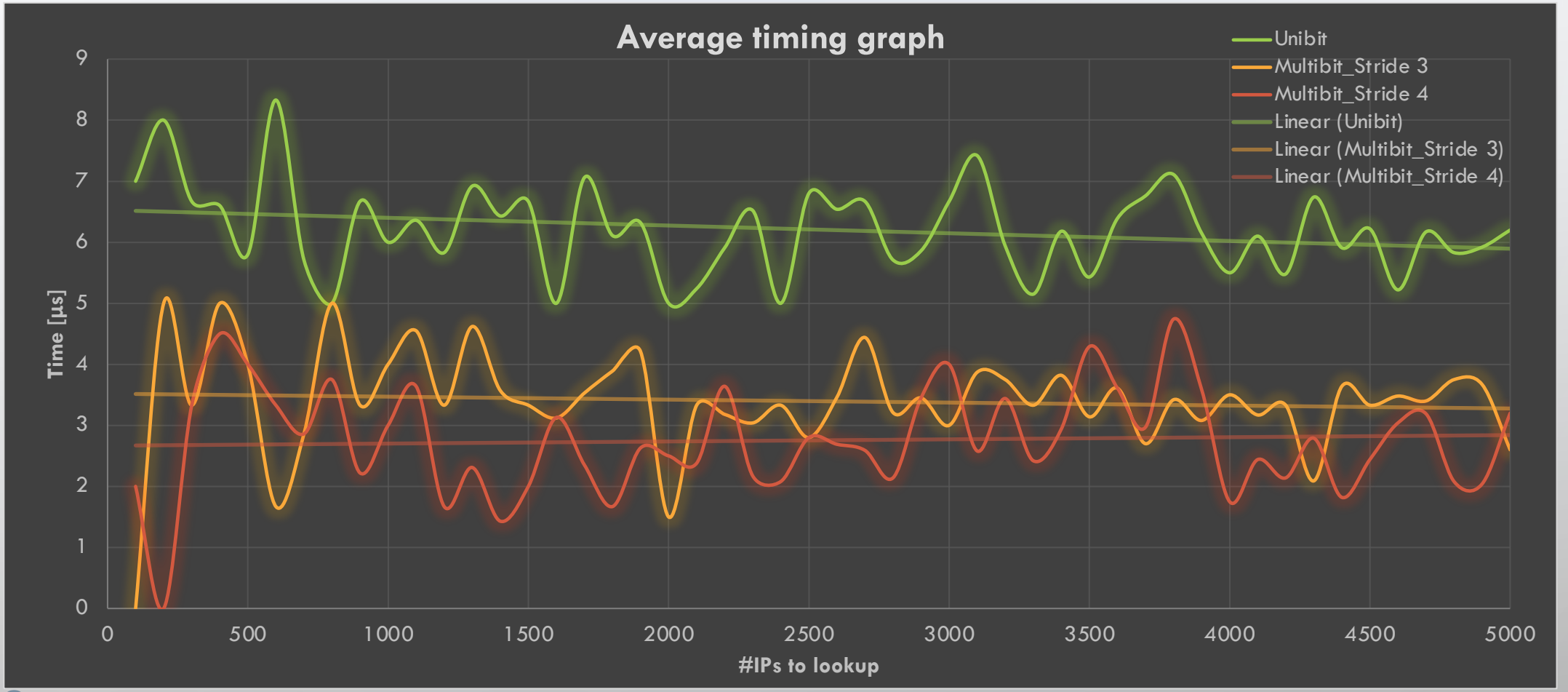
```
===========
Lookup done.
--- Sum of multibit-trie Lookup times = 0.009001255035400 39 s ---
```

COMPARE RESULTS



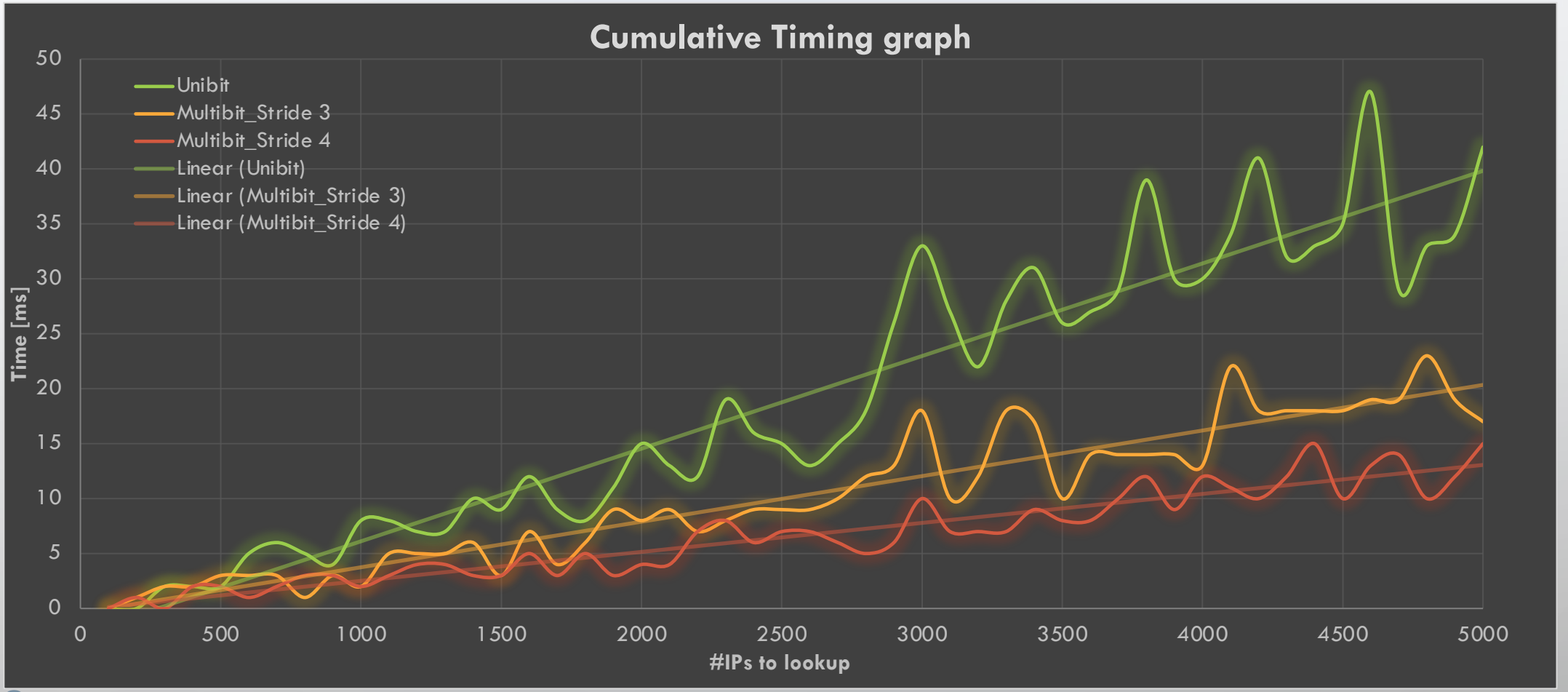
All the results achieved by CPU: intel core i5 – 3320M – 2.6 GHz

AVERAGE COMPARING



CUMULATIVE COMPARING

POLITECNICO
MILANO 1863



CONCLUSION

POLITECNICO
MILANO 1863

* AS IT CAN BE SEEN, THE COMPLEXITY OF THE MULTIBIT TREE IS “K” TIMES LOWER THAN THE COMPLEXITY OF THE BINARY TREE

Thanks