Fuzzy-Token: An Adaptive MAC Protocol for Wireless-Enabled Manycores

Antonio Franques¹ (franque2@illinois.edu),
Sergi Abadal², Haitham Hassanieh¹, Josep Torrellas¹

¹University of Illinois at Urbana-Champaign
²Universitat Politècnica de Catalunya

CCF-1629431  863337 (WiPLASH)
Current trends are leading to larger manycores.

Wireless on-chip communication holds promise for the implementation of fast networks for these multiprocessors.

In complement of a wired NoC, wireless provides:

- Low latency
- Natural broadcast capabilities
- Flexibility
Wired+Wireless Network-on-Chip

Hybrid Network Interface (HNIF)

Controller

eNIF

wNIF

Router

Transceiver

MAC

PHY

Antenna

CORES + MEMORY
Wired+Wireless Network-on-Chip
As the core density increases, more wireless interfaces can be expected on chip.

Need for arbitration strategies (MAC protocols)
- That provide low access latency
- That scale with number of wireless nodes
- That adapt to different traffic patterns
- That are simple to implement
Medium Access Control (MAC)

- The MAC layer defines mechanisms to ensure that all nodes can access the shared wireless medium in an organized manner.
- Two common access methods: token passing, random access.

<table>
<thead>
<tr>
<th>Token passing</th>
<th>Random access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pass a token around a virtual ring. Only the token holder can transmit</td>
<td>Simultaneous accesses to the same channel collide and need to retry</td>
</tr>
</tbody>
</table>

- **Token passing**:
  - Core 1: ✓
  - Core 2: ✓
  - Core 3: ✓

- **Random access**:
  - Core 1: × ✓
  - Core 2: ×
  - Core 3: ✓
The Medium Access Control (MAC) layer defines mechanisms to ensure that all nodes can access the shared wireless medium in a reliable manner.

Two common access methods:

- **Token passing**
  - Pass a token around a virtual ring. Only the token holder can transmit.
  - ✓ No wasted cycles at high loads
  - × Unnecessary delays if contention is low

- **Random access**
  - Simultaneous accesses to the same channel collide and need to retry.
  - ✓ No wasted cycles if contention is low (transmit right away)
  - × Lots of collisions at high loads

4 February 2021

Antonio Franques – University of Illinois at Urbana-Champaign
• Wireless on-chip scenario
  • Physically constrained – need for lightweight MAC protocol
  • Unlike off-chip scenarios, the environment is static and known beforehand
    • All nodes are synchronized
    • Collisions can always be detected
  • Protocols must scale to many cores and adapt to changes in traffic

<table>
<thead>
<tr>
<th></th>
<th>Low load</th>
<th>High load</th>
<th>Variability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random access</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Token passing</td>
<td>×</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>??</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Contribution: *Fuzzy-Token*

- We propose *Fuzzy-Token*, a hybrid protocol based on two basic approaches: token passing, and random access.
- We evaluate the performance of *Fuzzy-Token* with a synthetic traffic suite and real application traces.
- We compare the obtained performance with that of a token-passing and a random-access protocol for wireless NoCs, called BRS (*Mestres et al, 2016*).
2 operation modes
  - *Focused* $\Rightarrow$ only the token holder can transmit (collision-free guarantee)
  - *Fuzzy* $\Rightarrow$ all nodes inside *Fuzzy Area* (except token holder) that have pending packets can transmit with probability $p_i$

Mode can change at each step
  - If in *focused* and token holder doesn’t transmit $\Rightarrow$ switch to *fuzzy*
  - If in *fuzzy* and collision $\Rightarrow$ switch to *focused*
**Fuzzy-Token: Main Idea (2/2)**

- **Fuzzy Area** size controls amount of contention at each step
  - Increase size (after idle step) when the load is low to give rapid access to the few nodes that want to transmit
  - Quickly decrease size (after collision) when load increases to minimize further collisions
Fuzzy-Token: Example (1/2)

- **Initial state**
  - Fuzzy Mode (all nodes inside *Fuzzy Area* except token holder may transmit)
  - Token holder: node 0, *Fuzzy Area* size: 5

- **Fuzzy Area** size updated using additive increase multiplicative decrease
  - Increase area size by 1 after each idle step
  - Decrease area size by half (round up) after a collision

![Diagram showing node numbering, Fuzzy mode, Fuzzy mode with collision, Focused mode, and Focused mode with OK]
• Initial state
  • Fuzzy Mode (all nodes inside Fuzzy Area except token holder may transmit)
  • Token holder: node 0, Fuzzy Area size: 5

• Fuzzy Area size updated using additive increase multiplicative decrease
  • Increase area size by 1 after each idle step
  • Decrease area size by half (round up) after a collision
• Initial state
  • Fuzzy Mode (all nodes inside Fuzzy Area except token holder may transmit)
  • Token holder: node 0, Fuzzy Area size: 5

• Fuzzy Area size updated using additive increase multiplicative decrease
  • Increase area size by 1 after each idle step
  • Decrease area size by half (round up) after a collision
• Synthetic traffic latency
  ✓ As good as BRS at low loads (fully open Fuzzy Area, transmit immediately)
  ✓ Almost as good as Token at high loads (very small Fuzzy Area, mostly Token Holder is the only one that can transmit)
  ✓ Dynamic and fast adaptation from one behavior to another as load changes
• Real applications
  • We obtain latency statistics from Multi2Sim on a 64-core chip
  • Benchmarks from PARSEC and CRONO suites
✓ Fuzzy-Token provides latency among the lowest of 3 protocols
✓ 4.4x lower latency than BRS, and 2.6x lower than Token
Fuzzy-Token: Also in the Paper...

- Design decisions
- Related work
- Further analysis on...
  - Tail latency
  - Hotspot traffic
  - Bursty traffic
  - Throughput
  - Energy consumption
Fuzzy-Token: Conclusions

• Hybrid approach combines pros of both token-passing and random-access protocols
  ✓ Low latency at low loads (random-access mode)
  ✓ Low latency and collision-free at high loads (token-passing mode)

• Run both random-access and token-passing methods simultaneously
  ✓ Token is always passed to ensure fairness among nodes
  ✓ Protocol reacts immediately after traffic changes (mode change + Fuzzy Area update)

• All transceivers see same consistent view of wireless channel
  ✓ All nodes are synchronized and proceed in lockstep (no need for explicit messages)

• Evaluation with a synthetic traffic model and real application traces shows Fuzzy-Token achieves lowest latency than baseline protocols in many different scenarios
  ✓ Low/High loads
  ✓ Hotspot/Bursty traffic

Thank you!