Back Suction: Service Guarantees for Latency-Sensitive On-Chip Networks

Jonas Diemer, Rolf Ernst
diemer@ida.ing.tu-bs.de
NOCS 2010, 05 May 2010, Session N6
Motivation and Introduction

Back Suction Architecture

Operational Example

Experimental Evaluation and Conclusion
Motivation

- General-purpose many-core
  - Consumer devices (phones, PCs)
- Competing application requirements
  - General-purpose (office, games, ...)
  - Real-time streaming (augmented reality, SDR, ...)
  - Quality-of-Service support required for simultaneous execution
  - Run-time flexibility
- Baseline: packet-switched wormhole NoC
  - 8x8 mesh with distributed shared cache
Best Effort (BE) Traffic

- From general purpose applications
  - Mostly cache traffic (data, protocol)
- **Latency-sensitive**: Application performance degrades with higher latency → Important traffic
- Behavior unknown
Guaranteed Throughput (GT) Traffic

- From real-time streaming apps
- Streams of traffic from producer to consumer
- Regular access patterns → Required minimum data rates are known
- **Latency-tolerant**: Performance does not degrade with higher latency (up to a certain latency bound)

![Latency and Utility Graph]

- **Utility**
- **Latency**
- **Hard RT**
- **Soft RT**
Existing NoCs with QoS: Guarantees First!

Most existing NoCs treat best-effort traffic as “second-class citizen”

- Static allocation of time slots
  - E.g. AEthereal [Goossens], SuperGT [Marescaux]
  - Best-effort traffic only to fill unused slots → high BE latency

- Dynamic scheduling of VCs + priorities
  - E.g. MANGO [Bjerregaard], QNoC [Bolotin], [AlFaruque], Globally-Synchronized Frames [Lee]
  - Best-effort traffic on lowest priority → high BE latency
Goal: Guarantees and Low BE Latency

Real-time Traffic
- Utility
- Throughput Guarantees
- Latency
- Hard RT
- Soft RT

Best-Effort Traffic
- Utility
- Limited Prioritization
- Latency
Idea of Back Suction: Selective prioritization

- Prioritize BE traffic by default for optimal latency

- Prioritize GT traffic only due to insufficient progress
  - Signal low buffer occupancy towards upstream router
  - Asserted by sink at limited rate, propagates towards source as buffers deplete (“Back Suction”)

- Result: GT traffic (mostly) in the background, using reserved VC buffers

- Reverse of Back Pressure flow control
  - Prioritize instead of throttle
  - On low buffer occupancy instead of high buffer occupancy
  - Techniques are complementary
Back Suction Architecture

- One set of VC reserved per GT stream at run-time (see paper for details)
- Sink asserts back suction to last router limited rate
- Threshold Module at every VC
  - Generate Back Suction signal on low occupancy towards upstream
  - Avoid GT idle-progress during low buffer occupancy
Operational Example

- Simplified Routers (1 cycle delay)
- Upper VC: Used by BE traffic
- Lower VC: Reserved for GT stream

![Diagram of Router 1 and Router 2 with Flit, Flit transfer, Asserted Signal, Deasserted Signal, BE, GT, Thr., Arbiter, Rate Limit, Back Suction]
Operational Example (initial condition)

- Assumptions:
  - Rate Limit asserted
  - All GT buffers sufficiently filled → No back suction asserted
Operational Example (t=0)

- R1 receives BE flit (prioritized)
- R2 sends GT flit (idle progress)
- R1 cannot send (no flits above threshold)
Operational Example (t=1)

- R1 receives + sends BE flit (prioritized)
- R2 cannot send (no flits above threshold)
Operational Example (t=2)

- All routers send BE flits (prioritized)
Operational Example (t=3)

- Rate limit deasserted → sink asserts **back suction**
- R2 sends GT flit (**back suction**)
- R1 sends BE flit (prioritized)
- R1 receives GT flit (idle progress)
Operational Example (t=4)

- R2 GT buffer occupancy dropped → **propagate back suction**
- R1 + R2 send GT flits (back suction)
- R1 receives BE flit (prioritized)
Operational Example (t=5)

- Rate limit asserted $\rightarrow$ sink deasserts back suction
- R2 sends BE flit (prioritized)
- R1 sends GT flit (back suction)
- R1 receives BE flit (prioritized)
Operational Example (t=6)

- R2 has enough GT flits → back suction deasserted
- R1 asserts back suction (propagated)
- R1 receives GT flit (back suction)
- R1 + R2 send BE flits (prioritized)
Operational Example (t=7)

- R1 has enough GT flits → back suction deasserted (after 1 cycle!)
- R1 + R2 send and receive BE flits (prioritized)
Analysis of Real-Time Guarantees

- Suction requests at sink modeled by worst-case event arrival
- Round-robin scheduling analysis at every router (similar to Network Calculus)
  - Worst-case suction backlog
  - Back Suction threshold, VC buffer size, feasibility
  - Suction event model for upstream router

- Analysis performed online as an admission control

- See paper for details
Experimental Evaluation

- Simulation Setup
  - SystemC cycle accurate model of 8x8 mesh
  - Traffic modeled by traffic generators

- Comparing 3 techniques
  - GT prioritized (common approach)
    - BE may only use idle slots
  - Distributed Traffic Shaping (DTS, our previous work)
    - BE prioritized,
    - Traffic shapers at every router port limit BE rate
  - Back Suction (this paper)
Experimental Evaluation – Setup

- GT stream: 
  (0,2) → (5,4)
  - Requested throughput: 2 B/cycle (50% link BW)

- Rest sends BE traffic
  - Varying load

- Shapers (ejection, DTS)
  - Period $T = 8$
  - Tokens $c = 4$

- Measure BE latency at (1,2), overlaps with stream
Achieved GT Throughput

- No guarantees for DTS with low buffer size

Requested GT throughput
Achieved BE Latency (Tornado Traffic)

- Latency improvement: 32%
- Back Suction similar to DTS
  - With lower buffer requirements!
Conclusion

- Back Suction Flow Control: Buffer occupancy controls prioritization
  - Prioritize BE for low latency (under low load)
    - Improve latencies for BE traffic by up to 32%
  - Prioritized GT for guarantees (under insufficient progress)
    - Throughput guarantees by formal real-time analysis
  - Simple implementation
    - Similar performance to previous DTS scheme with lower buffer cost
- Enable predictable communication in a latency-sensitive general-purpose architecture
Thank You for Your Attention!

Jonas Diemer, diemer@ida.ing.tu-bs.de