TECHNISCHE UNIVERSITÄT
CAROLO-WILHELMINA
ZU BRAUNSCHWEIG

ASP-DAC 2010
20 Jan 2010
Session 6C

Efficient Throughput-Guarantees for Latency-Sensitive Networks-On-Chip

Jonas Diemer, Rolf Ernst
TU Braunschweig, Germany
diemer@ida.ing.tu-bs.de

Michael Kauschke
Intel, Germany

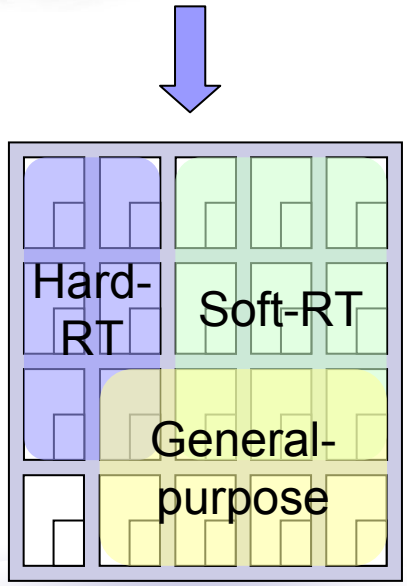
Outline

- Motivation and Introduction
- QoS-Aware Link Arbitration Scheme
- Real-time Analysis
- Experimental Results
- Conclusion

Motivation

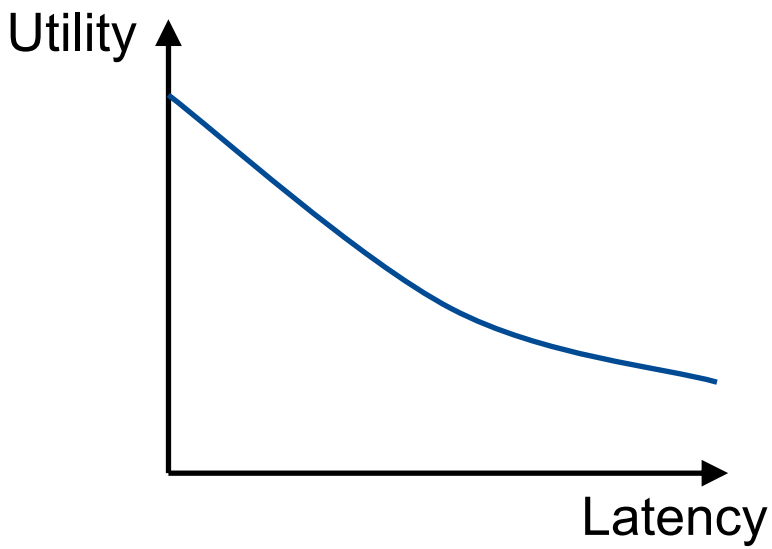
- General-purpose many-core
  - Distributed caches
- Competing application requirements
  - General-purpose (office, …)
  - Real-time / Streaming
    - Soft-RT, e.g. augmented reality
    - Hard-RT, e.g. software-defined radio
- **Quality-of-Service** support required for simultaneous execution

Source: Nokia



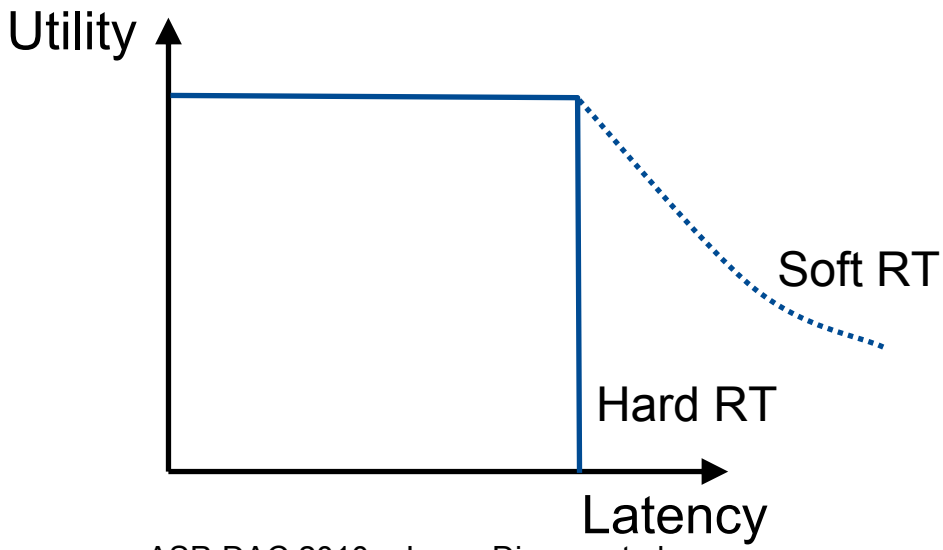
Best Effort (BE) Traffic

- From general purpose applications
  - Mostly cache traffic
- Behavior unknown and bursty
- Latency-sensitive: Application performance degrades with higher latency



Guaranteed Throughput (GT) Traffic

- From real-time streaming apps
  - Large transfers
- Requirements known
- Deadline-oriented: Requires guarantees
- Latency-tolerant: Performance does not improve with lower latency (up to a certain latency bound)

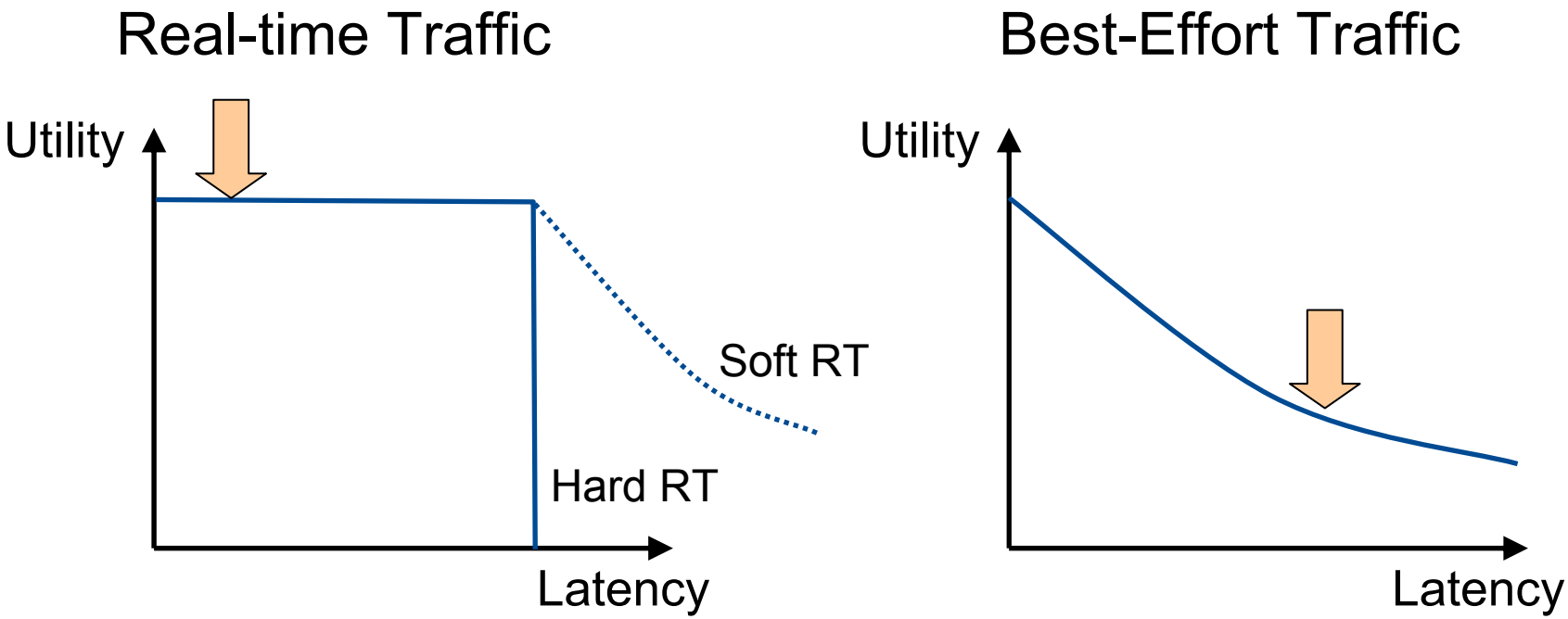


Related Work: NoCs with QoS

- Static allocation of time slots
  - E.g. AEthereal [Goossens], SuperGT [Marescaux]
  - Formal guarantees (for throughput, latency, …)
  - Contention free routing
  - Best-effort traffic only to fill unused slots → high latency
- Dynamic scheduling of VCs + priorities
  - E.g. MANGO [Bjerregaard], QNoC [Bolotin], BAA [AlFaruque], Globally-Synchronized Frames [Lee]
  - Guarantees depend on actual traffic model
  - Best-effort traffic on lowest priority → high latency

Common practice: Favor traffic with guarantees, ignore timing of best-effort traffic

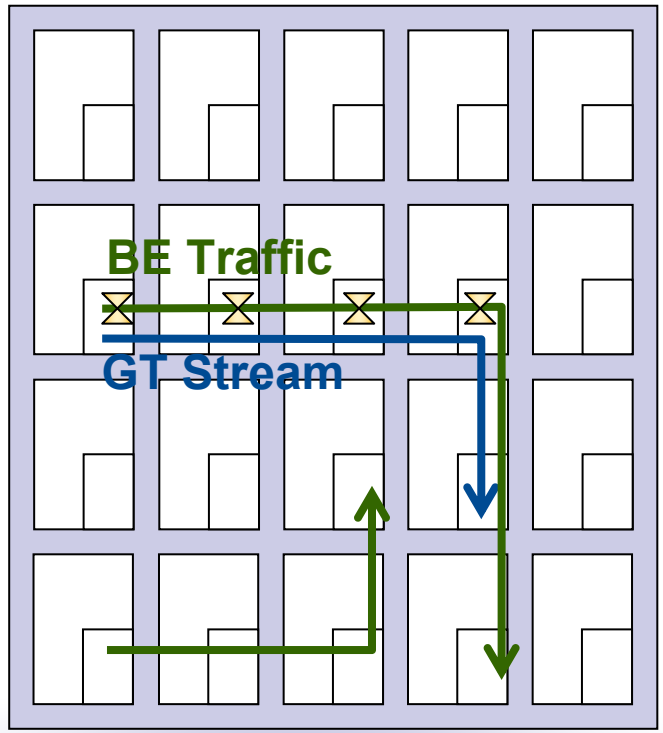
Goal: Guarantees **and** Low Latency



- Prioritize BE traffic
- Limit BE rate to leave enough free bandwidth
- Initial version [Diemer09]
  - Virtual-cut through switching

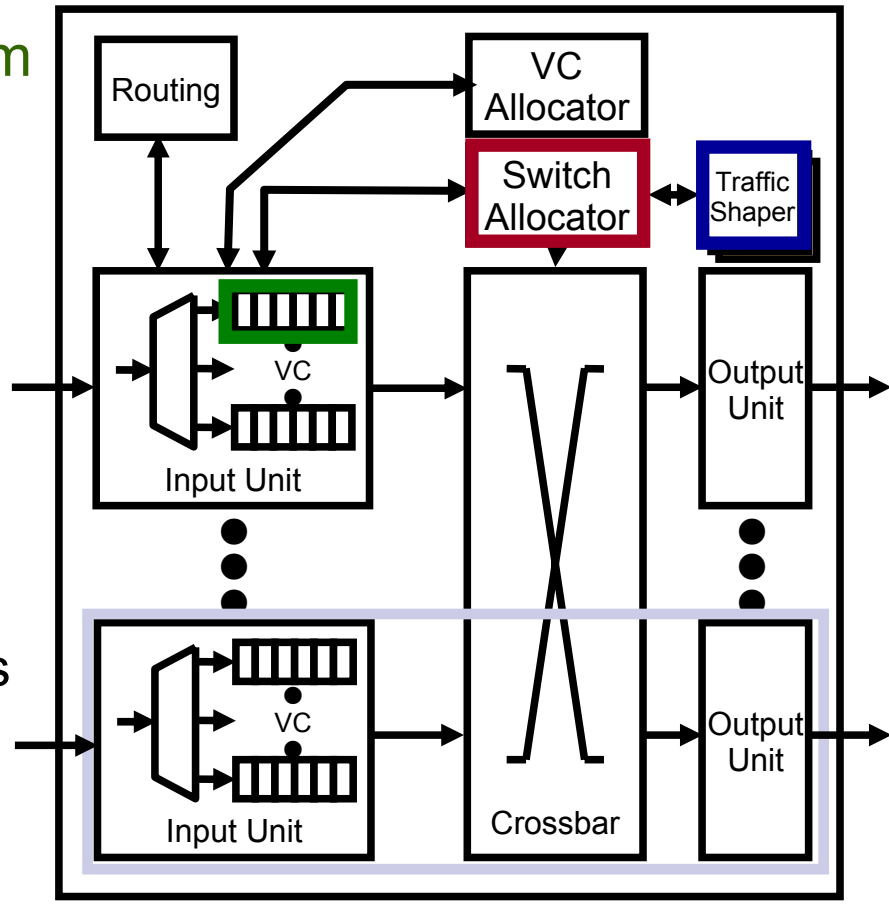
Approach: Distributed Traffic Shaping

- Packet-switched mesh
  - Static XY-routing
  - Wormhole switching
    - Packets composed of flits (head, body, tail)
- Routers
  - Input buffering
  - Virtual channels (VC)
  - 4 pipeline stages
- Modify switch arbitration in routers



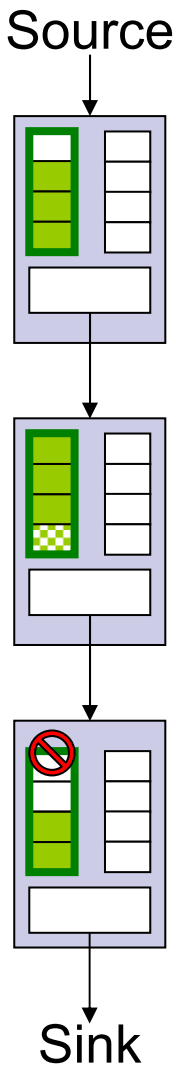
Modifications in Router

1. Reserve VC for each GT stream
   - Because VC arbitration can not be bounded
   - Dynamic allocation at run-time
2. Prioritize BE traffic
   - For optimal latency
   - Buffer GT flits on contention
   - GT progresses during idle slots
3. Limit average BE rate
   - To guarantee throughput
   - Traffic shapers for each output port



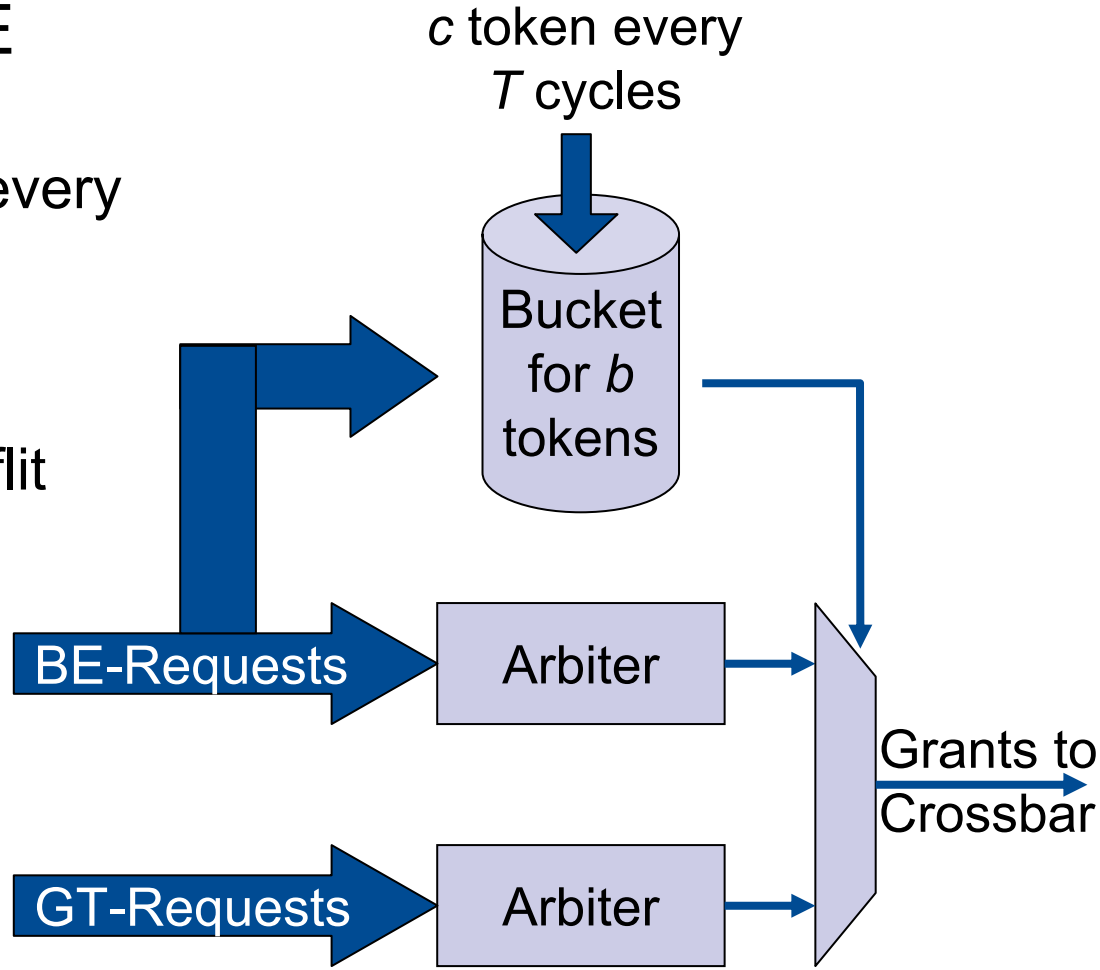
VC Reservation: Packet Concatenation

- **Reserve** a set of VC for every **GT stream**
- Problem: VC can only hold flits from 1 packet
  - Only one status register (e.g. for destination)
  - “Bubbles” between GT packets during which VC can not accept new flits
  - But: all packets in GT stream can share status register (same destination)
- Solution: merge multiple GT packets
  - Simple – Just omit tail/head flits between packets
    - GT stream appears as a single huge worm
  - “Delimiter” flit type to distinguish packets at sink
- Added benefits
  - No extra logic for VC reservation
  - Skip routing and arbitration



Arbitration: Prioritize and Limit BE rate

- Token Bucket for BE requests
  - *c* tokens are added every *T* cycles
  - *b* tokens maximum
  - 1 token removed for every prioritized BE flit
- Selective priority arbiter
  - Separate arbiter for each traffic class
  - Controlled by bucket
    - Prioritize BE if buckets in token

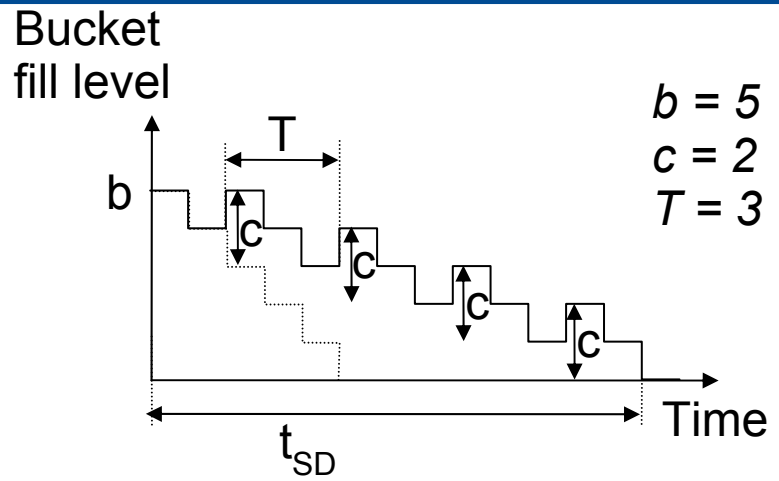


Real-Time Analysis

- Average BE token rate: *c/T*
  - Resulting guaranteed average GT rate: $r_{GT} = 1 - c/T$
- Up to *b* tokens can accumulate for a BE burst
  - During this time, GT traffic does not receive its average rate
  - Blocked GT traffic must be buffered locally during this time so it can be sent later
  - Otherwise, there may not be enough GT flits later when BE is idle and accumulates new tokens
- Hence: VC must be large enough to buffer GT flits arriving at a rate of r_{GT} during a BE burst

Real-Time Analysis (2)

- Maximum duration of a BE burst
  - Longest time (in cycles) for a **full** bucket to deplete when BE is constantly trying to send
  - 1 token is removed every cycle
  - *c* new tokens arrive every *T* cycles
  - The first addition of new tokens happens after c tokens have been removed (otherwise, tokens would be dropped due to bucket limit)



$$t_{SD} = b + \left\lceil \frac{t_{SD} - c}{T} \right\rceil \cdot c$$

- Integer fix-point problem, solved by iterating
  - Start with e.g. $t_{SD} = b$

Real-Time Analysis (3)

- Maximum amount of GT flits arriving during BE burst
  - Assuming a GT rate of $r_{GT} = 1 - c/T$

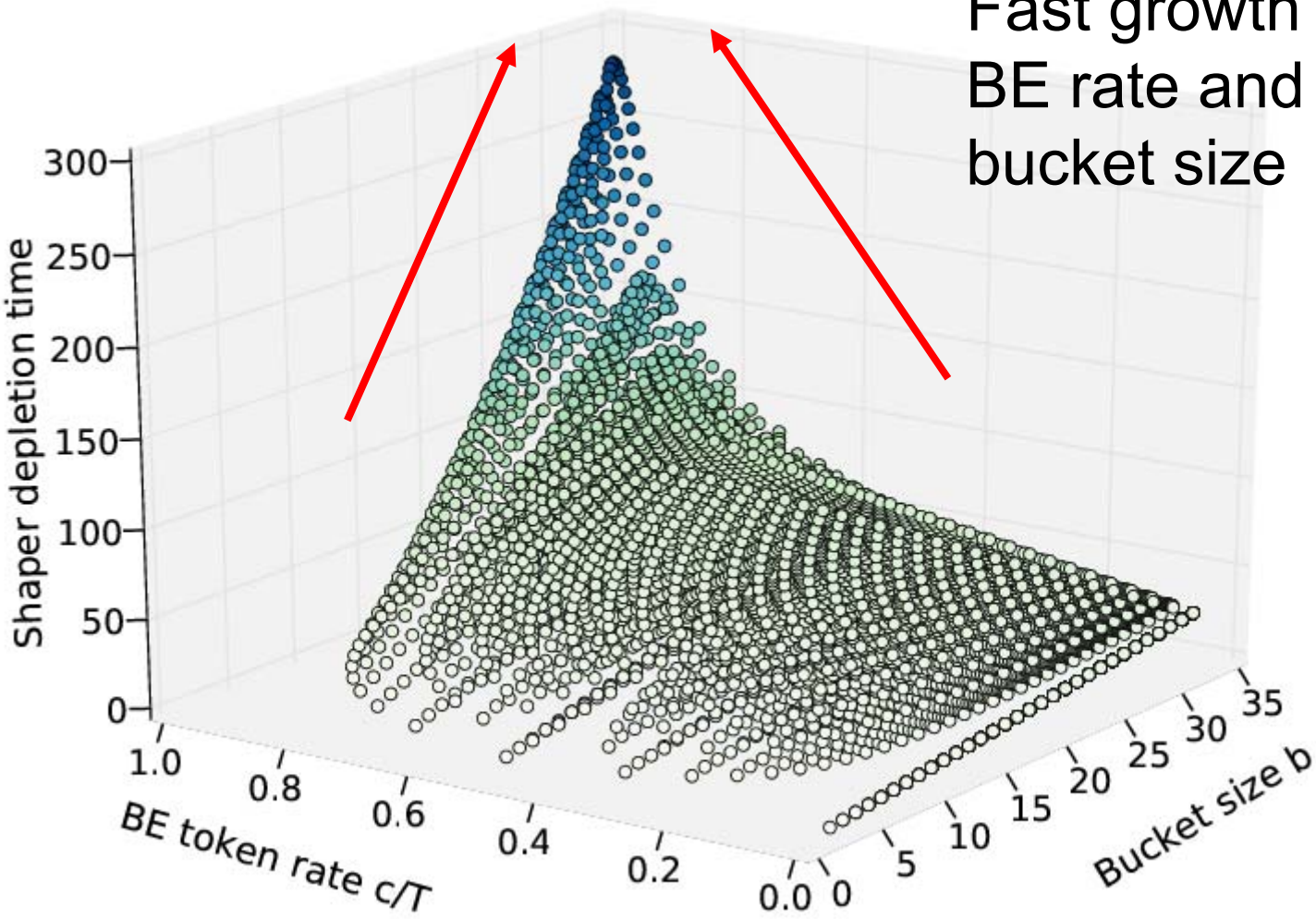
$$s_{GT} = \lceil r_{GT} \cdot t_{SD} \rceil$$

- GT VC buffer must be at least s_{GT} flits large

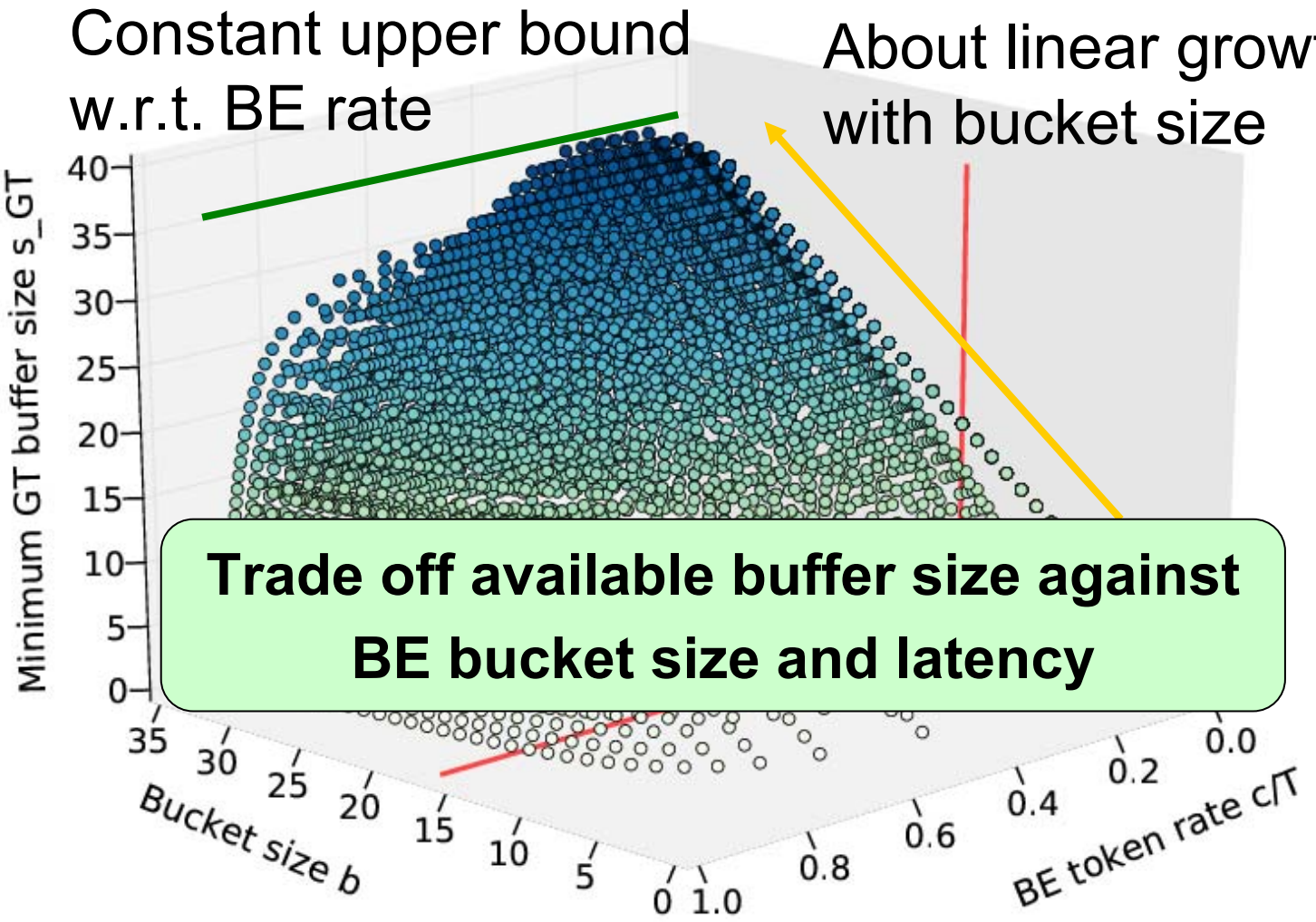
- Overlapping GT connections may interfere with each other
  - Handled by injection shaping of GT streams (see Paper)

Shaper Depletion Time – Design Space

- *b = [2, 32]; T = [2, b]; c = [0, T-1]*



Minimum Required Buffer Size



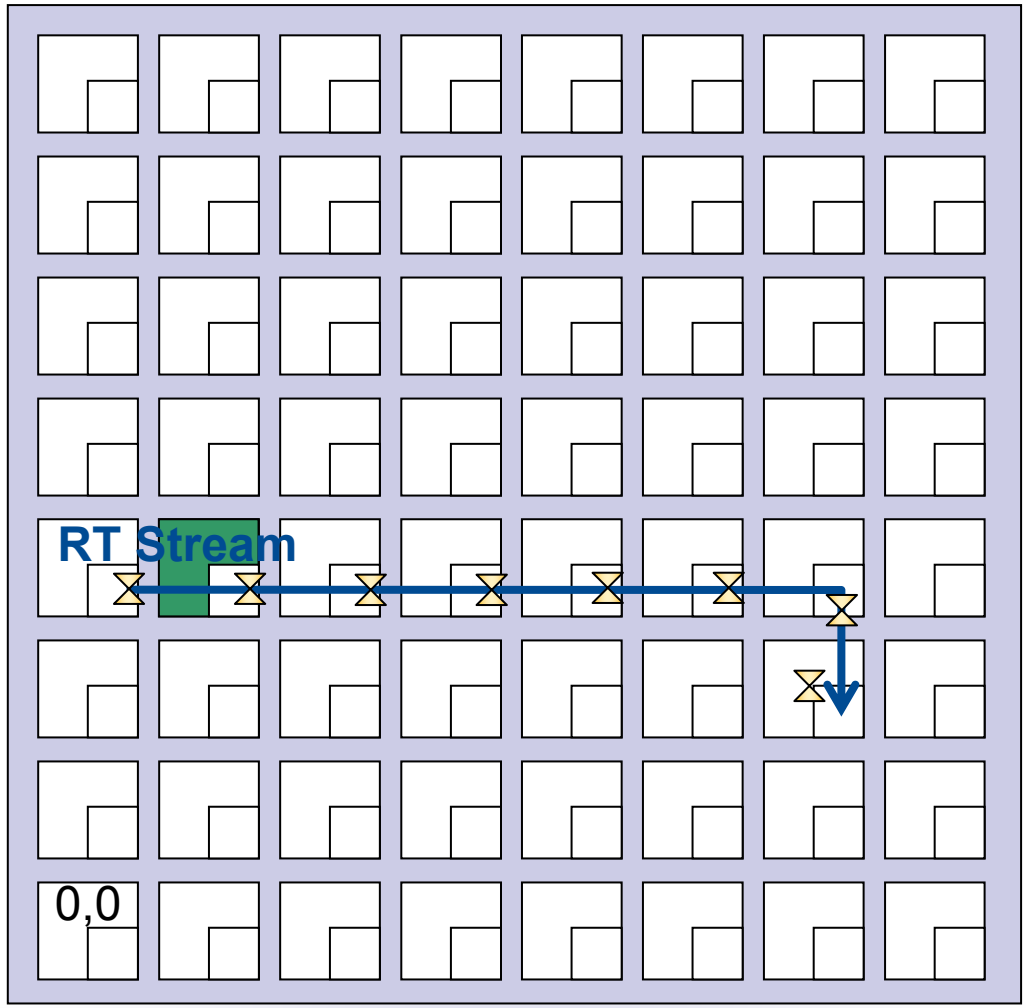
Experimental Evaluation

- Simulation Setup
  - SystemC cycle accurate model of 8x8 mesh
  - Traffic modeled by traffic generators
- Single real-time stream: (0,3) → (6,2)
  - Requested throughput: 2 bytes/cycle (50% of link BW)
- Remaining nodes send random BE traffic
  - Different patterns (Uniform, Tornado), varying load

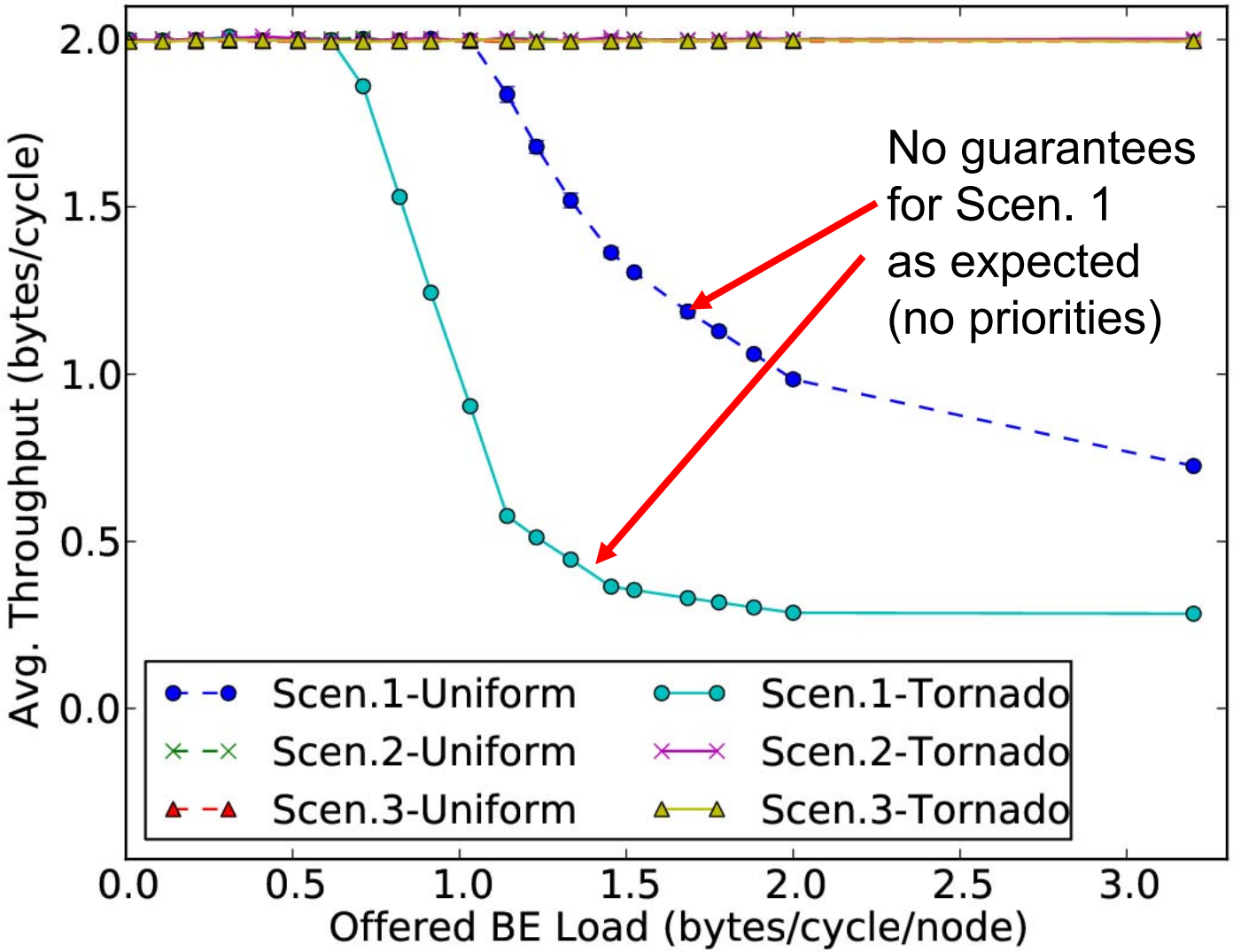
- 3 Scenarios
  - **Scenario 1**: BE+GT on same priority (**no QoS**)
  - **Scenario 2**: GT prioritized (naïve approach)
  - **Scenario 3**: BE prioritized + Traffic Shaping (our approach)

Scenario 3 – Shaper Setup

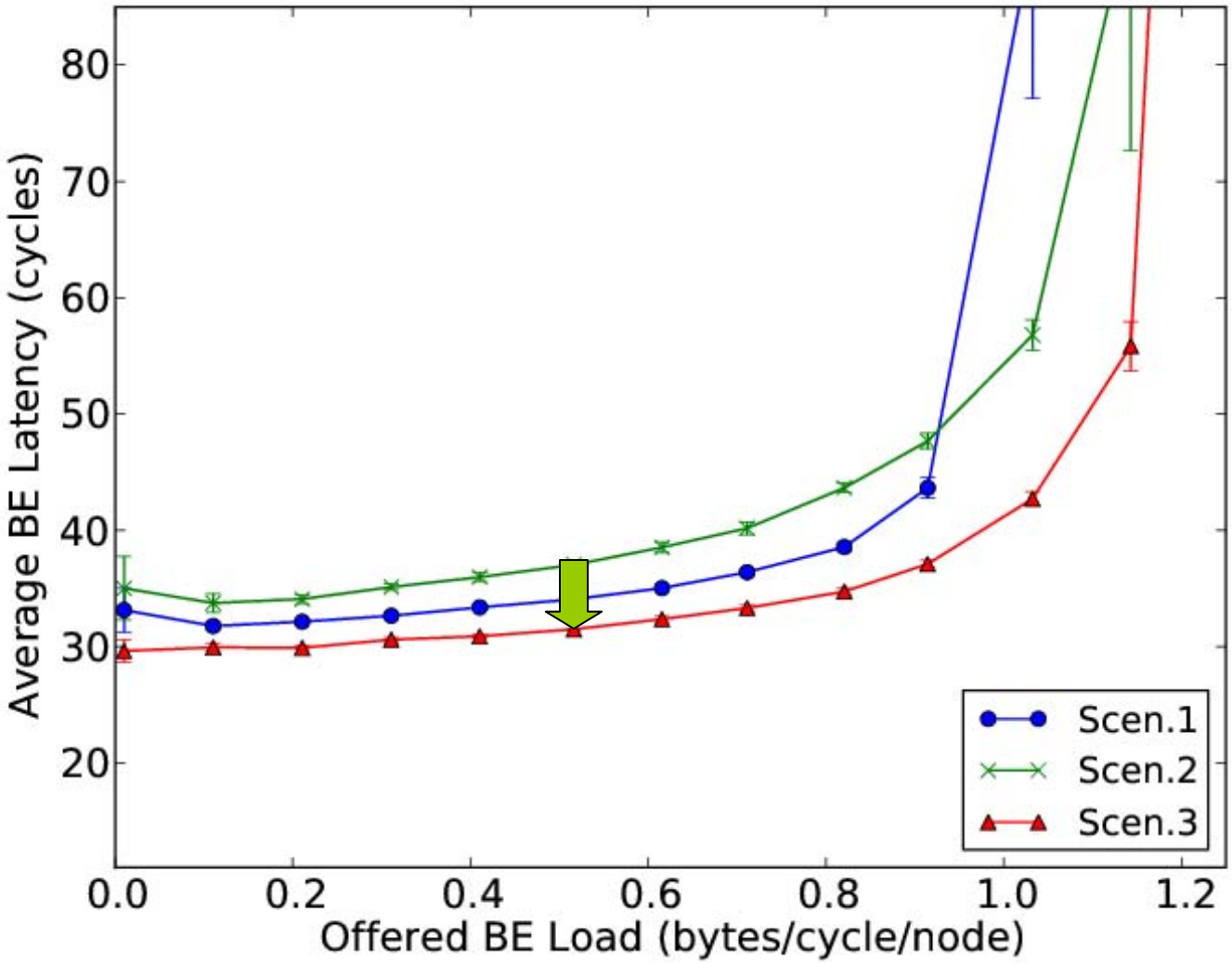
- GT stream: (0,3)→(6,2)
- Bucket size *b* = *8*
- Period *T* = *8*
- Token per period *c* = *4*
  - along GT route
  - *c* = *8* (no BE limit) otherwise
- Measure BE latency at (1,3)



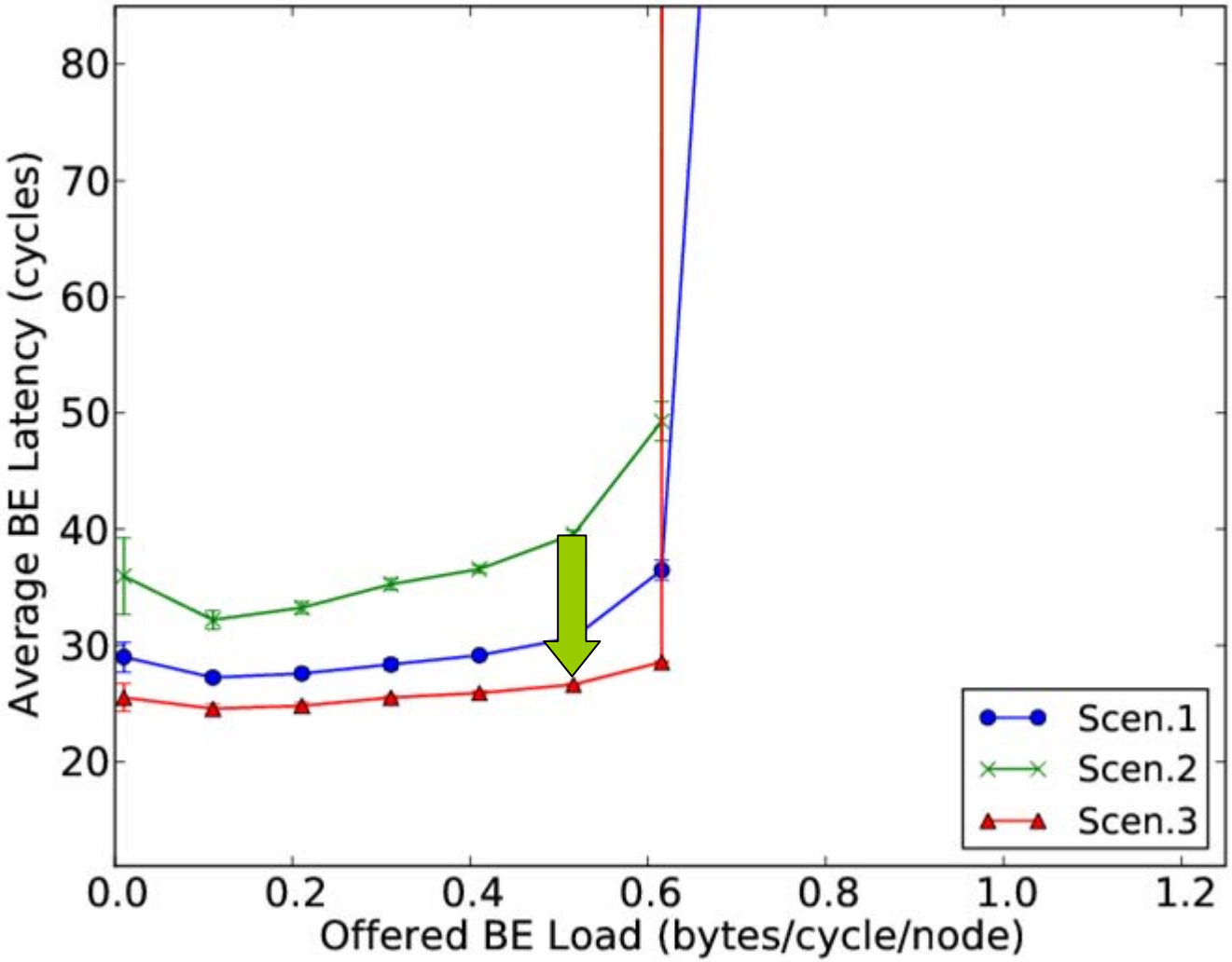
Results: Achieved Throughput of GT stream



Results: BE-Latency Uniform Random

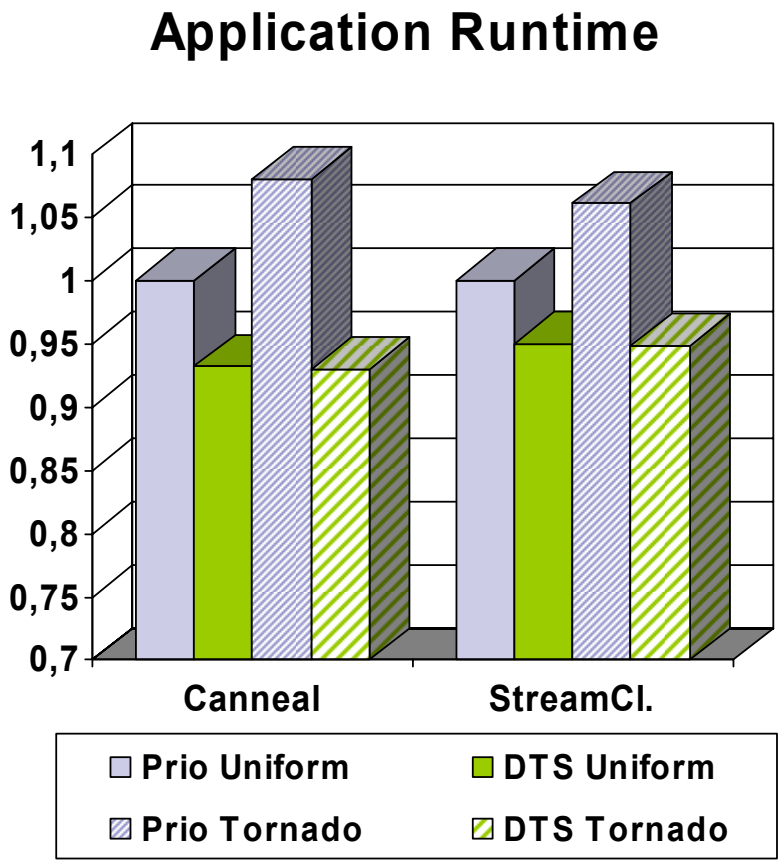


Results: BE-Latency Tornado



Effect on BE Applications

- Application traces with NoC feedback
- CPU at (1,3)
  - Dual-issue
  - 8kB L1 cache, 1 cycle
  - PARSEC benchmarks
    - Streamcluster, Canneal
    - Latency sensitive
- Perfect L2 cache at (5,3)
  - Always hits in 1 cycle
  - Connected to CPU via NoC using BE traffic



Conclusion

- NoC for QoS on a general-purpose processor
  - Prioritization and distributed traffic shaping of BE-traffic
  - Throughput guarantees for specific streams
    - Formal analysis of required buffer sizes
  - Good latencies for best-effort traffic
    - Latency improvements of up to 47%
    - BE application speedup of up to 14%
- Future Work
  - Improve buffer requirements
  - Additional traffic classes and resources

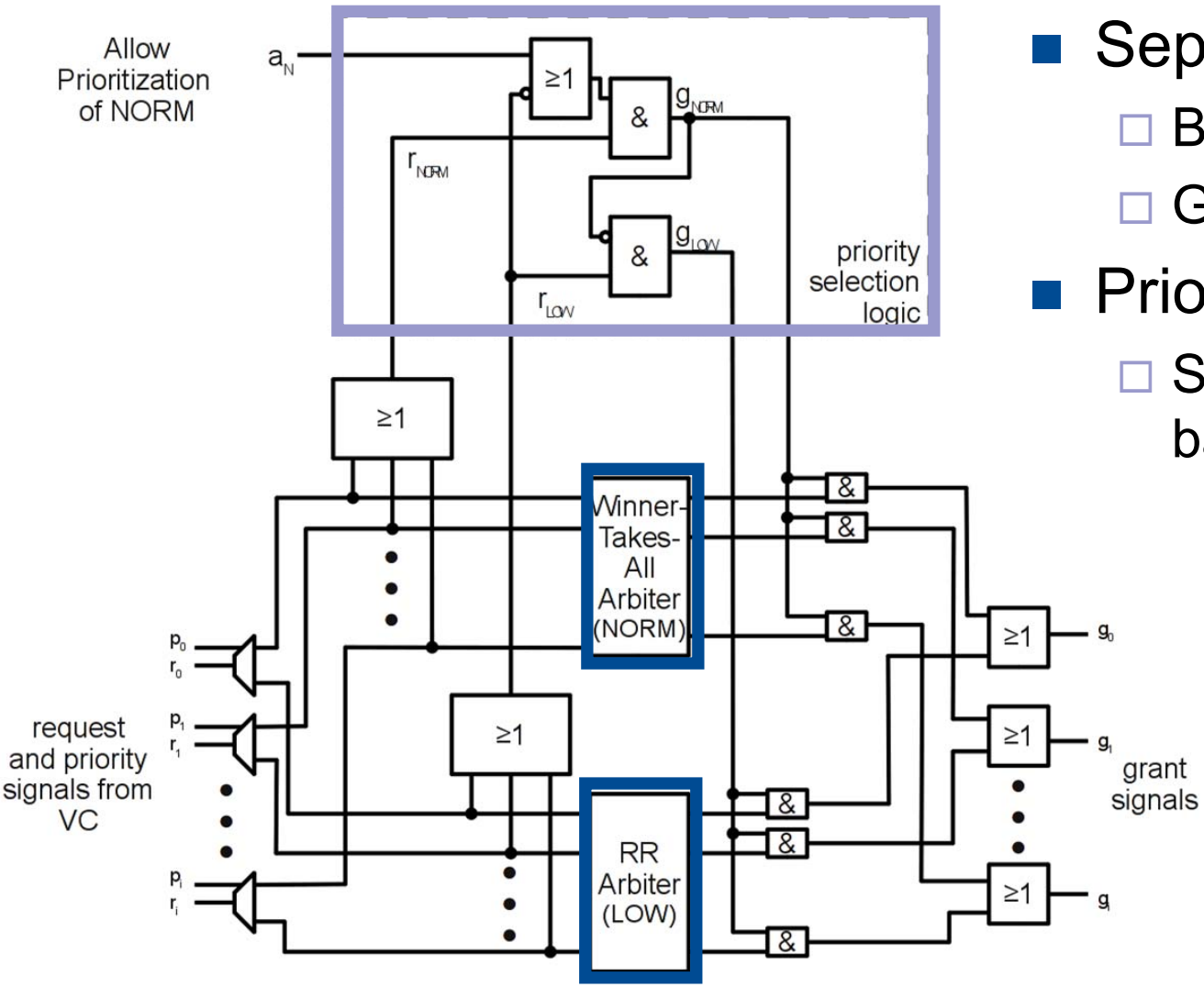
Thank You
for Your Attention!

Questions?

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

Backup

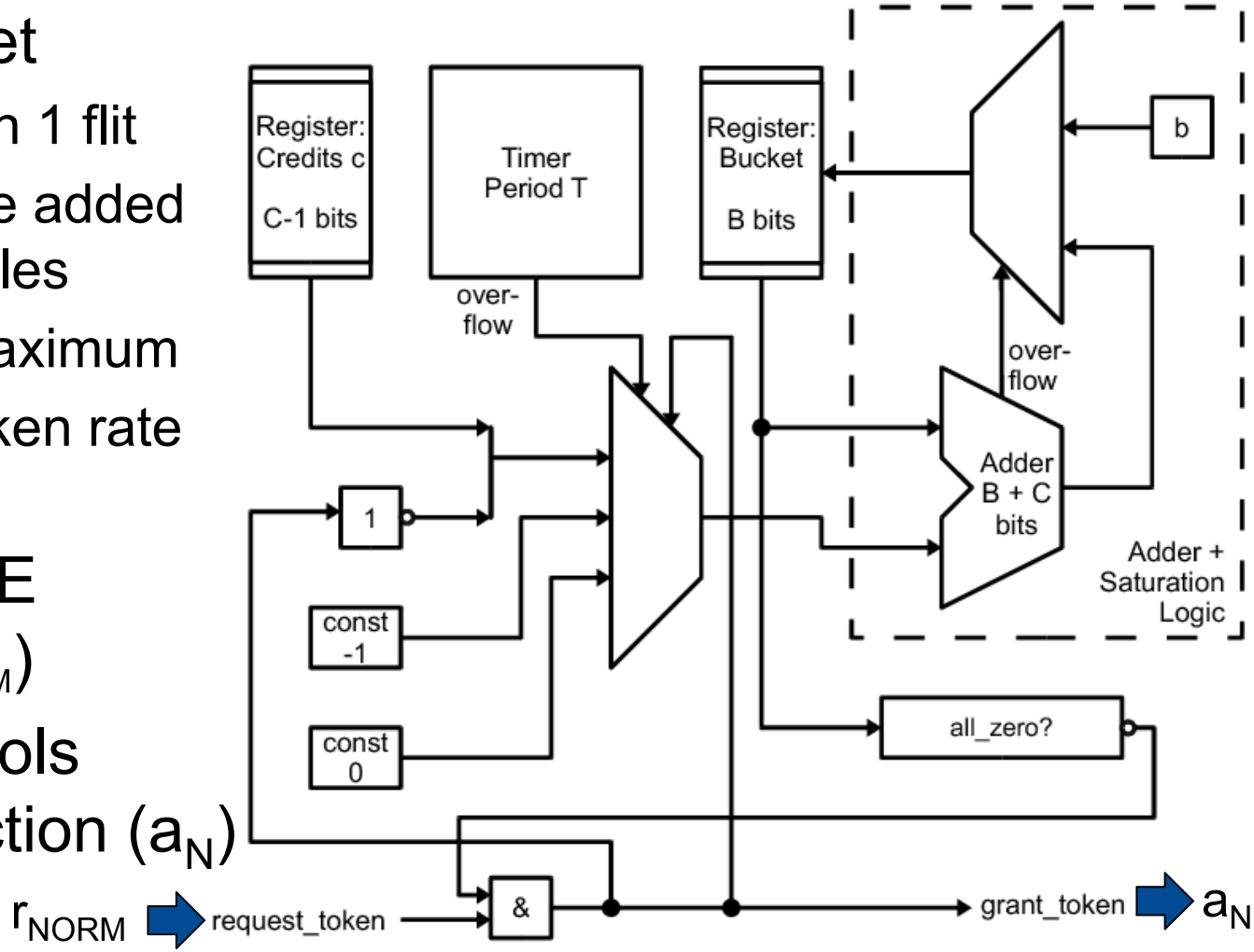
Prioritize BE: Selective-Priority Arbiter



- Separate arbiters
  - BE: Winner-takes-all
  - GT: Round-robin
- Priority selection logic
  - Select BE or GT based on
    - Signal a_N
    - Presence of BE/GT

Limit BE rate: Token Bucket Shaper

- Token Bucket
  - Token worth 1 flit
  - *c* tokens are added every *T* cycles
  - *b* tokens maximum
  - Average token rate c/T
- Input: Any BE request (r_{NORM})
- Output controls priority selection (a_N)



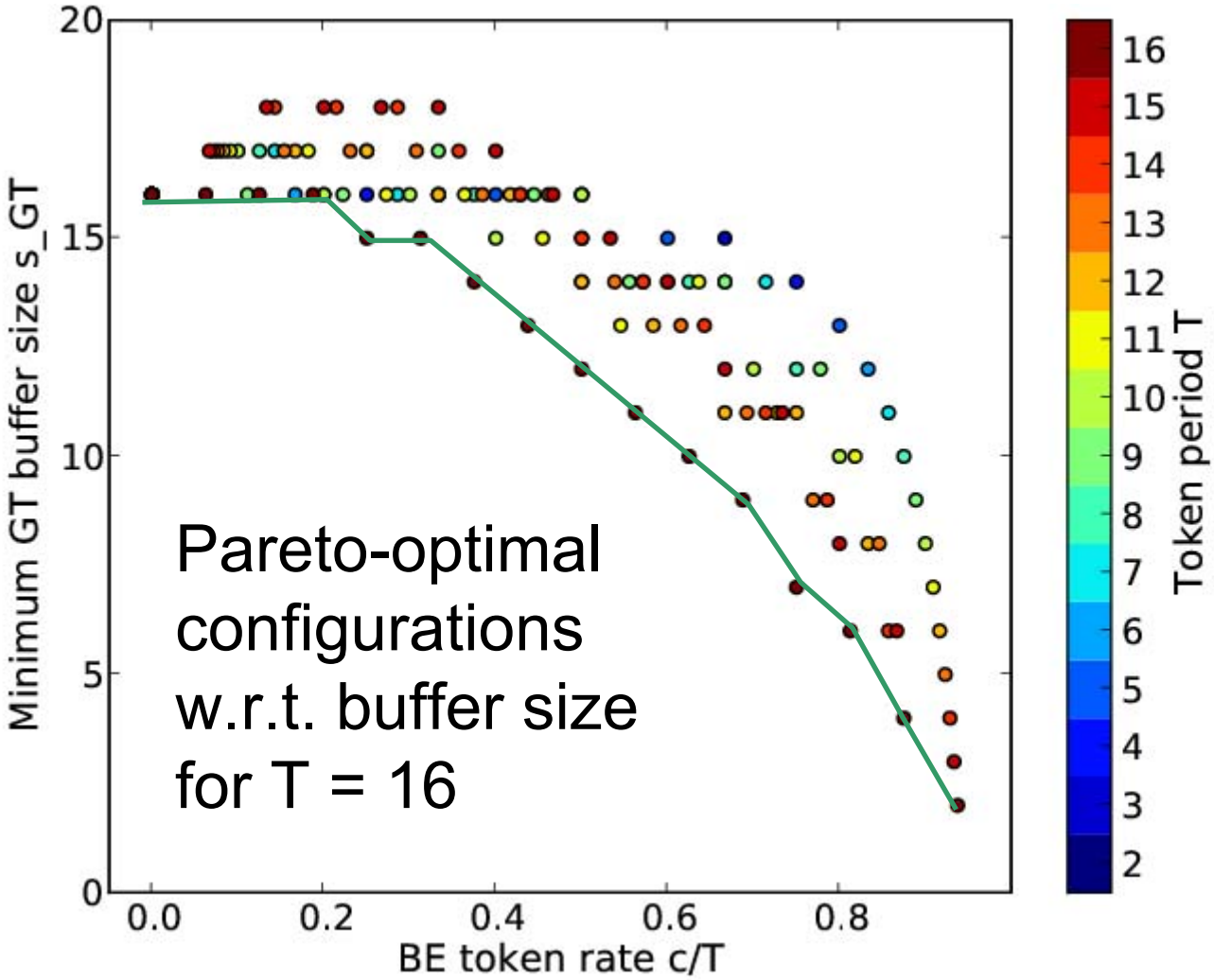
Distributed Traffic Shaping – Costs

- Hardware requirements
  - Reservation of one VC per GT stream in every router (dynamic)
  - Selective priority arbiter (one per output port on every router)
    - 2 arbiters, selection logic, mux, demux
  - Traffic shaper (one per output port on every router)
    - 2 registers, counter, adder, logic

Overlapping GT Connections

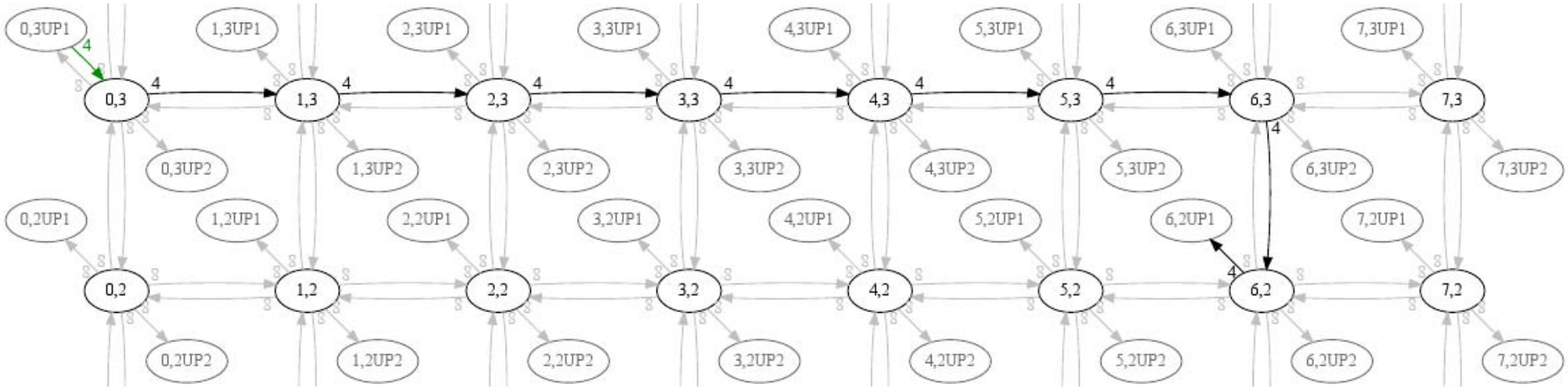
- Multiple GT streams may share the same output port
  - Reserve separate set of VCs for every stream
  - Set up BE shapers so that the combined bandwidth is guaranteed
- GT streams can interfere with each other
  - GT streams are not isolated from each other
  - Guarantee that overlapping GT streams do not use more than their requested bandwidth
    - Injection shapers limiting the GT rate
    - Not required for non-overlapping streams

Buffer Requirements for b=16



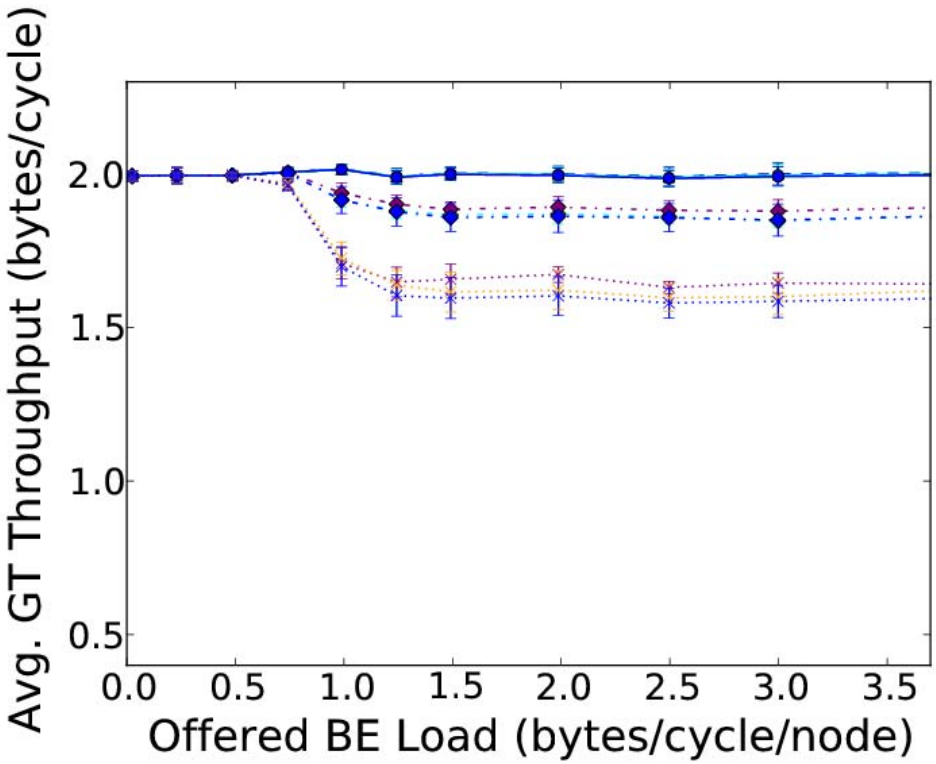
Scenario 3 Shaper Setup

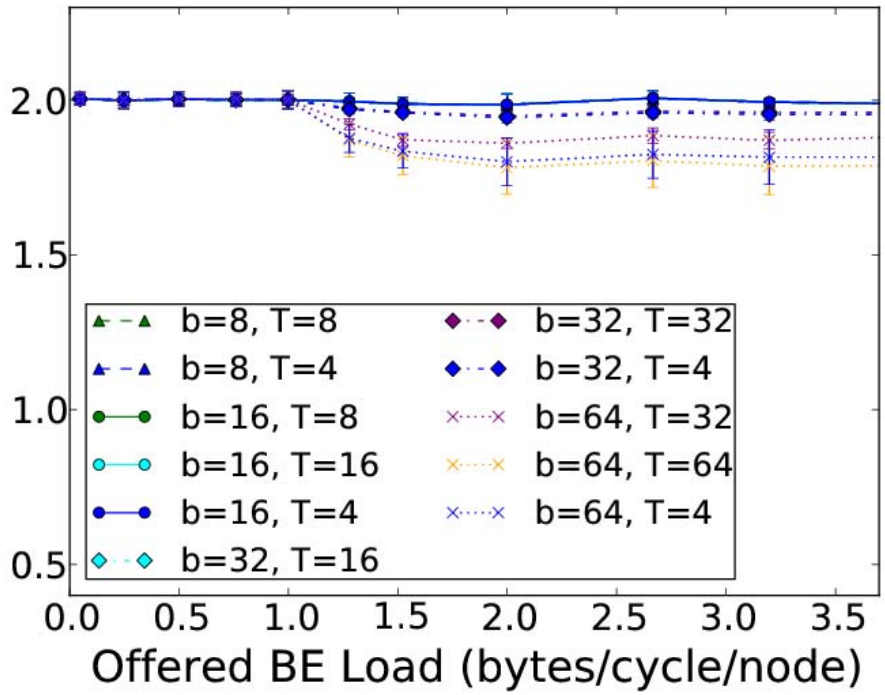
- Bucket size *b* = *8*
- Period *T* = *8*
- Credits per period *c* = *4* along GT route, *c* = *8* (no BE limit) otherwise



Effect of Shaper Settings (1)

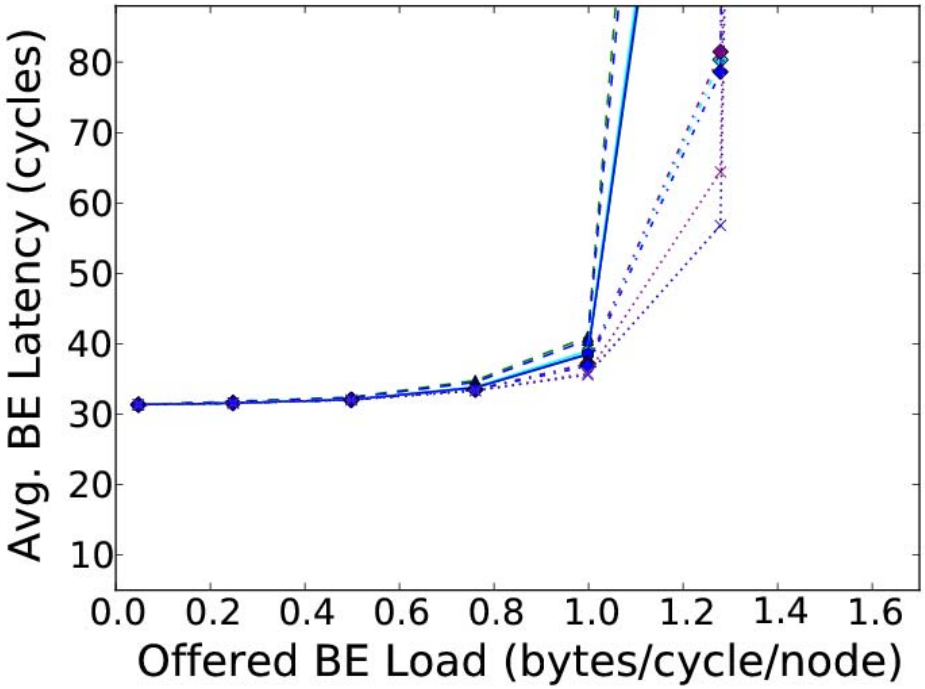
- GT throughput not met for large bucket sizes
  - As predicted in Analysis (insufficient buffer size)

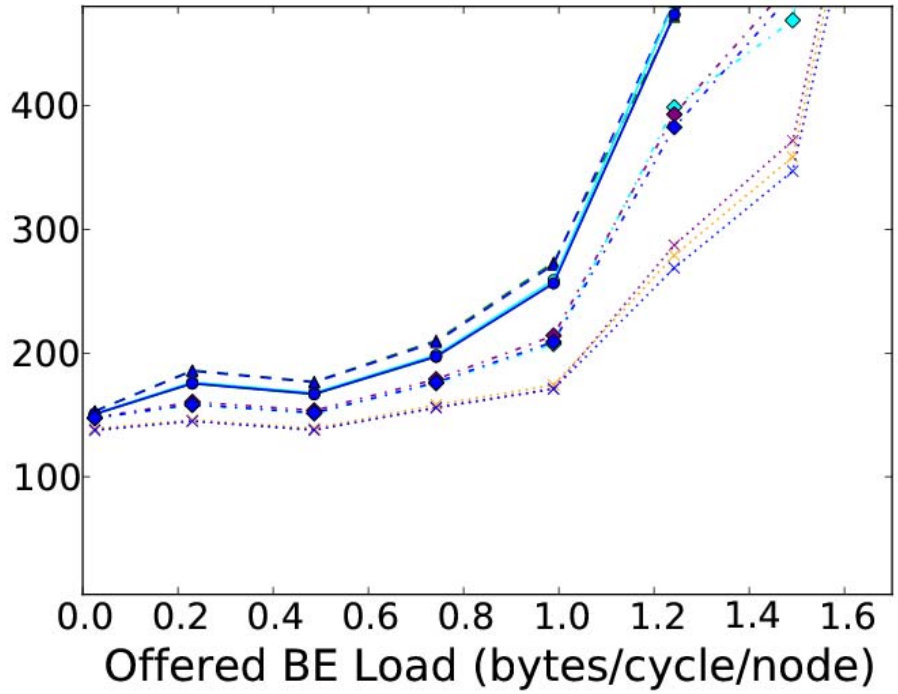




Effect of Shaper Settings (2)

- Increasing the bucket size reduces latency
  - Mostly for large packet sizes
- Reducing period *T* slightly improves latency
  - Effect minor for this traffic pattern (non-bursty)





Overlapping GT Streams

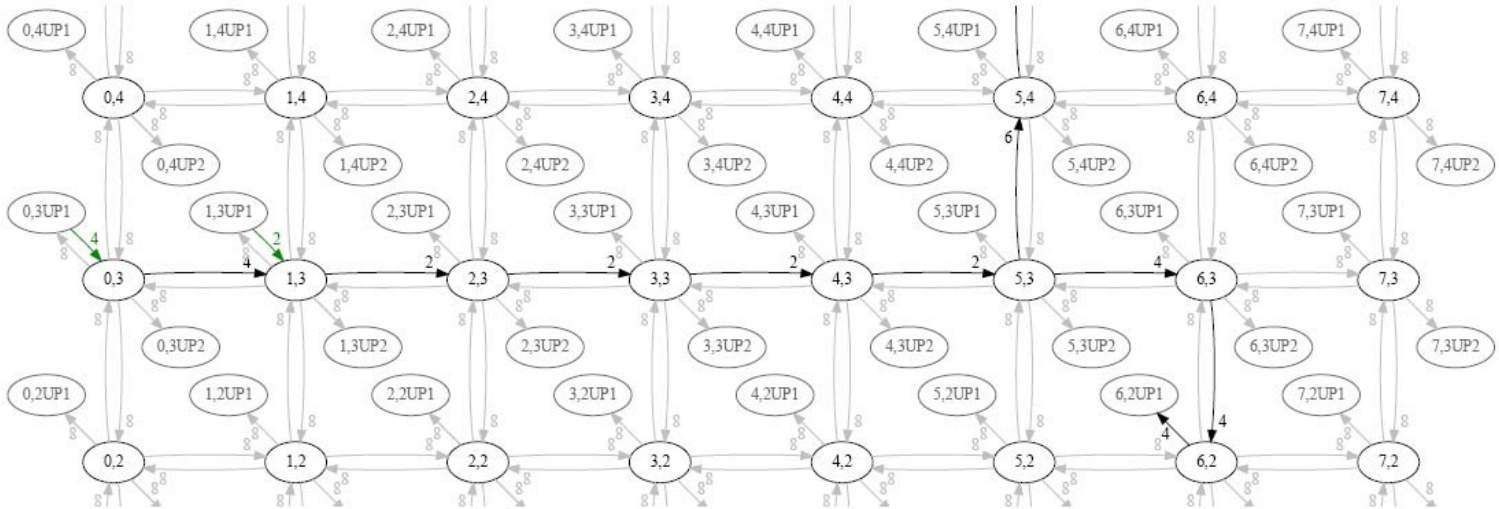
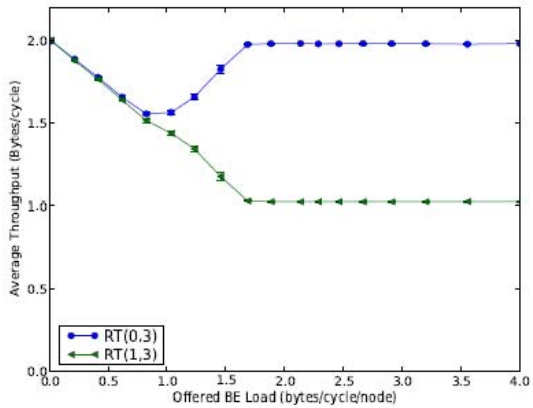
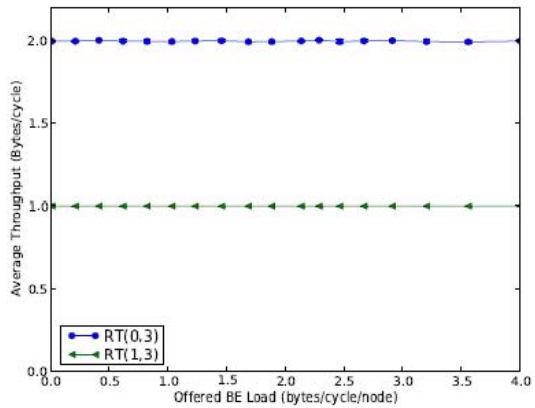


Figure 13: Shaper Configuration for two GT streams: (0,3)→(6,2)@2B/cycle and (1,3)→(5,7)@1B/cycle



(a) Injection shapers disabled



(b) Injection shapers enabled

Concept: Distributed Traffic Shaping

- Reserve VC for GT
  - Because VC arbitration can not be bounded
- Prioritized BE traffic for optimal latency
  - Buffer GT flits (instead of BE) on contention
- Limit average BE rate to fulfill guarantees
  - Using distributed traffic shapers