1 Simulations

1.1 4x4 Crossbar

`Bestimmen und begründen Sie wie die Werte zustande kommen`

- `avg Input Queue Length`
  - Generated Packets
    - size: $s = 512\,\mathrm{b}$
    - send interval: $t = \mathrm{uniform}(1\,\mu\mathrm{s}, 10\,\mu\mathrm{s}) = 5.5\,\mu\mathrm{s}$
  - Connection to XBar
    - datarate: $1\,\mathrm{Gbps}$

$$r_{\text{generated}} = \frac{512\,\mathrm{b}}{5.5\,\mu\mathrm{s}} \rightarrow l_{\text{q,avg}} = 0 \tag{1}$$

- `avg End-to-End Latency`
  - (no delays inside buffers, because the generated datarate, even for all 4 apps, is lower than a single datarate channels maximumthroughput)
  - minimum
    - `App --> C --> Inport --> C --> Outport --> C --> App`
    - delay for packet: $t_{delay} = 512ns$ (per DatarateChannel C)

    $\Rightarrow\ t_{e2e,min} = 1.536\,\mu\mathrm{s}$
  - maximum
    - all apps send to same destinatino and arbiter has to do round robin for all packets 4 times the dely for the Datarate Cahnnel inside XBar

    $\Rightarrow\ t_{e2e,max} = 3.027\,\mu\mathrm{s}$
  - on avg, the mimal case is 7 times more likely.

    The minimal case can exist in 24 possible constellations (connecting each app to a different app). While the maximum case can only exist in 4 possible constellations (all apps sending to one of 4 apps).

    $$\Rightarrow t_{e2e,avg} \approx \left(24 \cdot t_{e2e,min} + 4 \cdot t_{e2e,max}\right)/28 = 1.7\,\mu\mathrm{s}$$

    (This is a coarse estimation, otherwise the cases inbetween need to be consudered)
- `avg Arbiter Request Queue Length`
  - minimum
    - queue is empty, because all arbitration requests can be fulfilled instantly

    $\Rightarrow\ l_{arbq,min} = 0$
  - maximum

    - * que is filled with 3 waiting requests, because all apps are wanting to send their packets to the same output port

    ⇒ $l_{arbq,max} = 3$

  - on avg (analogously to t_{e2e})

$$\Rightarrow l_{arbq,avg} \approx (24 \cdot 0 + 4 \cdot 3)\,/28 = 0.42$$

- `avg Arbiter Request Queue Time` ((36 * 512) + (8 * 1024) + (4 * 1536))/92
- `avg Output Buffer Queue Length`
- `avg Throughput`