Flow Definition and Identification, QoS Service definition and parameters
TSN

Norman Finn
Huawei Technologies Co, Ldt

TSN stream identification

- The "Stream identification function" is defined in IEEE Std™ 802.1CB, *Frame Replication and Elimination for Reliability*. It inspects a frame and generates a *locally-significant* integer stream_handle.
- Stream identification is defined for, and is applicable to, any "end system" or "relay system". It is not limited to 802 stations or bridges.
- IEEE 802.1CB specifies how an end or relay system can use the stream_handle to perform frame replication and elimination.
- IEEE Std™ 802.1Q-2018 specifies how an IEEE 802 bridge or end station can use the stream_handle for QoS functions, including TSN.
- **There is no provision in any IEEE 802.1 standard for using the stream_handle for the purpose of selecting on which port(s) to output a packet.**

What fields determine the stream_handle?

- **Input port number**
- **Source MAC address**
- **Destination MAC address**
- **VLAN ID**
- **L2 priority**
- **IP source address**
- **IP destination address**
- **Next protocol (UDP/TCP)**
- **Source port number**
- **Destination port number**
- **DSCP**

IEEE Std 802.1CB *FRER* stream_handle uses

- The stream_handle can send a packet through an instance of a state machine that adds a layer 2 sequence number tag.
- The stream_handle can cause each packet of a stream to be replicated, each copy having a possibly different stream_handle.
- The stream_handle can send a packet through an instance of a packet elimination state machine, which operates on the sequence number.
- The stream_handle can direct the packet through a rewrite function that can:
  - Delete a layer 2 sequence number tag.
  - Rewrite the destination MAC address and add/remove/change the VLAN ID.

IEEE Std 802.1Q bridge stream_handle uses

- IEEE 802.1Q defines a set of ”stream gates” for an 802.1Q bridge:
- Based on the stream_handle, a frame can be:
  - Directed to a red/yellow/green flow meter.
  - Directed to a time-scheduled on/off gate.
  - Directed to a packet length filter.
  - Directed to a timed total-byte-count filter.
  - Assigned an “IPV” that determines to what class of service (queue) the packet is assigned.

New standards in progress

IEEE P802.1CBdb FRER Extended Stream Identification Functions

- Extends the fields used for the Stream identification function to arbitrary mask-and-match.

IEEE P802.1DC Quality of Service Provision by Network Systems

- Provides normative references to IEEE 802.1Q and 802.1CB so that any Standards Development Organization can specify how their systems provide the QoS of an IEEE 802.1Q bridge.

TSN Output queue selection methods

Standards **complete in IEEE 802.1Q-2018**:

- Strict priority.
- Weighted priority.
- Per-priority feedback from next hop.
- Credit Based Shaper: Similar to the IntServ leaky bucket. Can be applied to (up to) 8 output queues per port.
- Time-scheduled gates: Each of 8 output queues per port is turned on/off independently using a repeating schedule that can be tied to a network-synchronized clock.
  - Corallary: Cyclic Queuing and Forwarding (CQF), a 2- or 3-buffer rotation at constant rate giving constant delay per hop.

TSN Output queue selection methods

Standard nearing completion (in 2019, IEEE P802.1Qcr):

- Two-level queuing.
- Any number of per-flow shaping state machines at the first level.
- Existing 8 output queues per port at the lower level.

TSN Output queue selection methods

Thus, IEEE 802.1 covers **two different kinds of TSN Streams**:

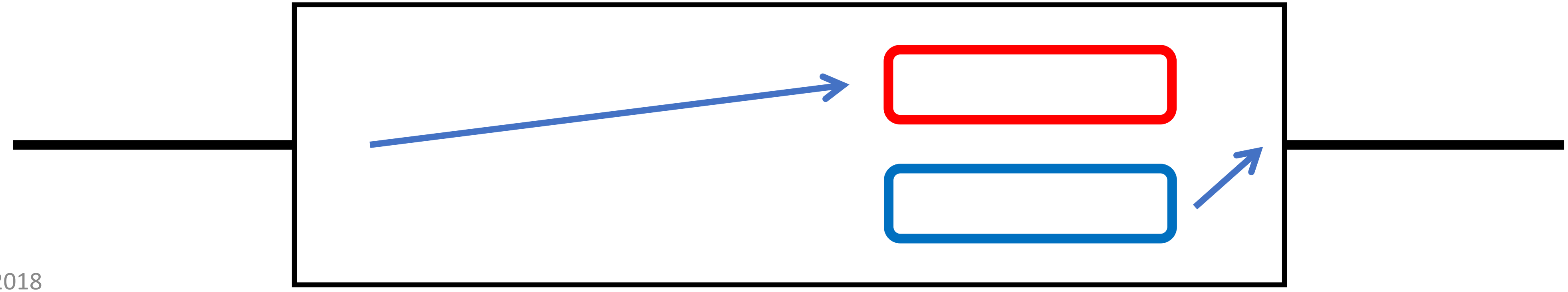
- "**Continuous**" streams: Independent, no inter-stream coordination, proof against worst-case interference. (Typical of audio/video or service provider.)
  - Characterized by bandwidth, maximum packet size.
  - Latency and buffer requirements can be computed rapidly and dynamically.
- "**Scheduled**" streams: Resources such as buffers can be reused at different times by different Streams. (Typical of industrial automation.)
  - Characterized by a repeating pattern of timed transmissions.
  - Computing the network schedule is difficult and requires heuristics to solve.

Cyclic Queuing and Forwarding

- IEEE Std 802.1Qci Per-Stream Filtering and Policing plus IEEE Std 802.1Qbv time-scheduled queue gates can be combined to produce:
- IEEE Std 802.1Qch Cyclic Queuing and Forwarding

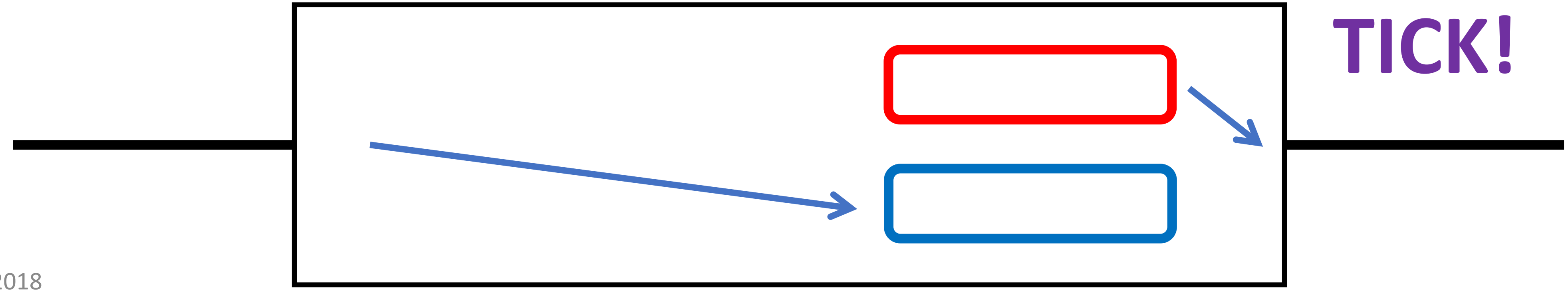
Cyclic Queuing and Forwarding

- Two-buffer version: Two buffers per port. Input and output buffers swap at the same moment, once every cycle, period T_C. Small guard band to allow for transit and forwarding time. All relay nodes are synchronized and swap buffers at the same moment. Cycle time T_C > transit time + forwarding time + clock inaccuracy + max data transmit time.



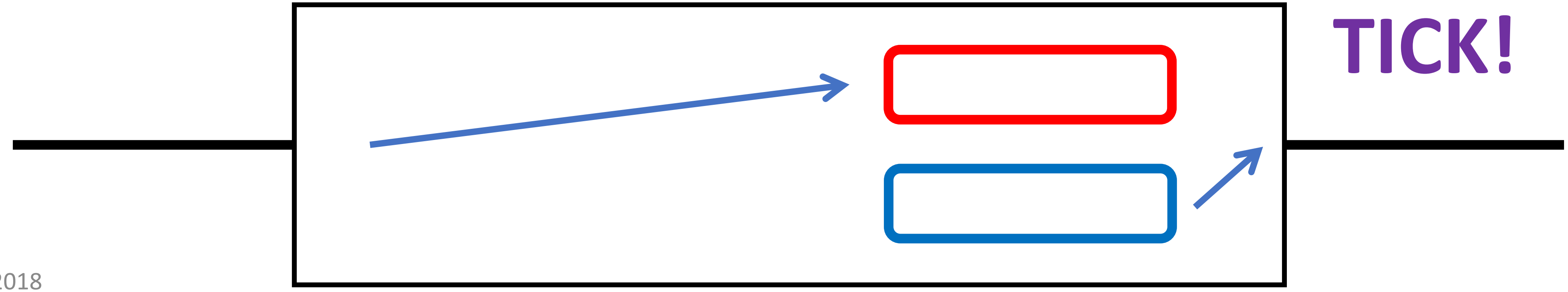
Cyclic Queuing and Forwarding

- Two-buffer version: Two buffers per port. Input and output buffers swap at the same moment, once every cycle, period T_C. Small guard band to allow for transit and forwarding time. All relay nodes are synchronized and swap buffers at the same moment. Cycle time T_C > transit time + forwarding time + clock inaccuracy + max data transmit time.

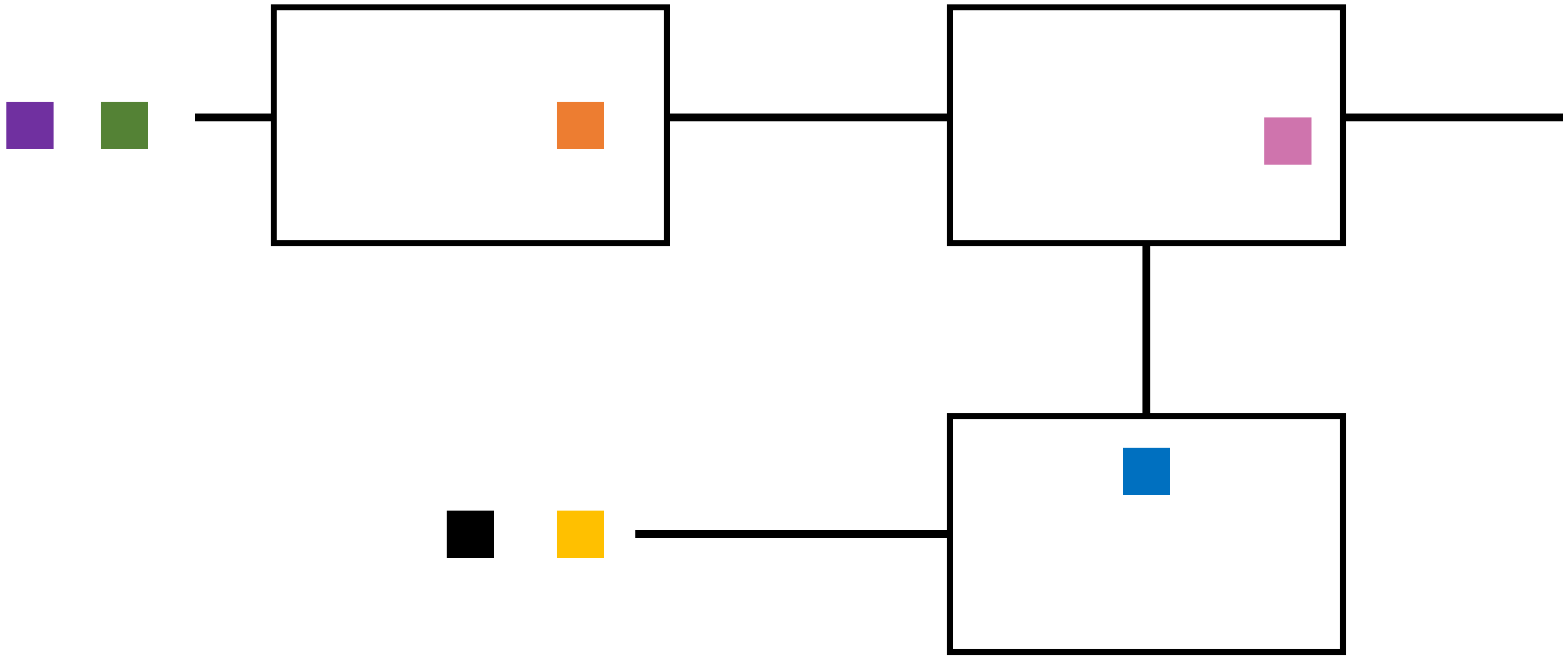


Cyclic Queuing and Forwarding

- Two-buffer version: Two buffers per port. Input and output buffers swap at the same moment, once every cycle, period T_C. Small guard band to allow for transit and forwarding time. All relay nodes are synchronized and swap buffers at the same moment. Cycle time T_C > transit time + forwarding time + clock inaccuracy + max data transmit time.

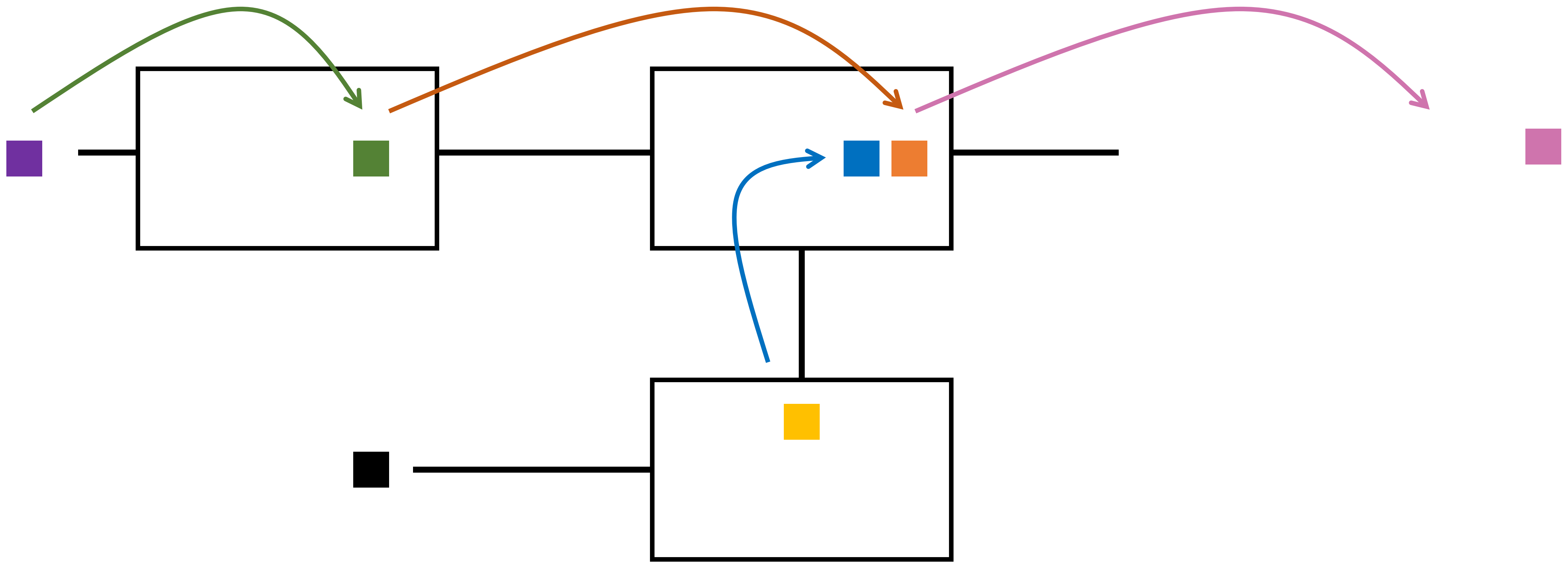


Cyclic Queuing and Forwarding



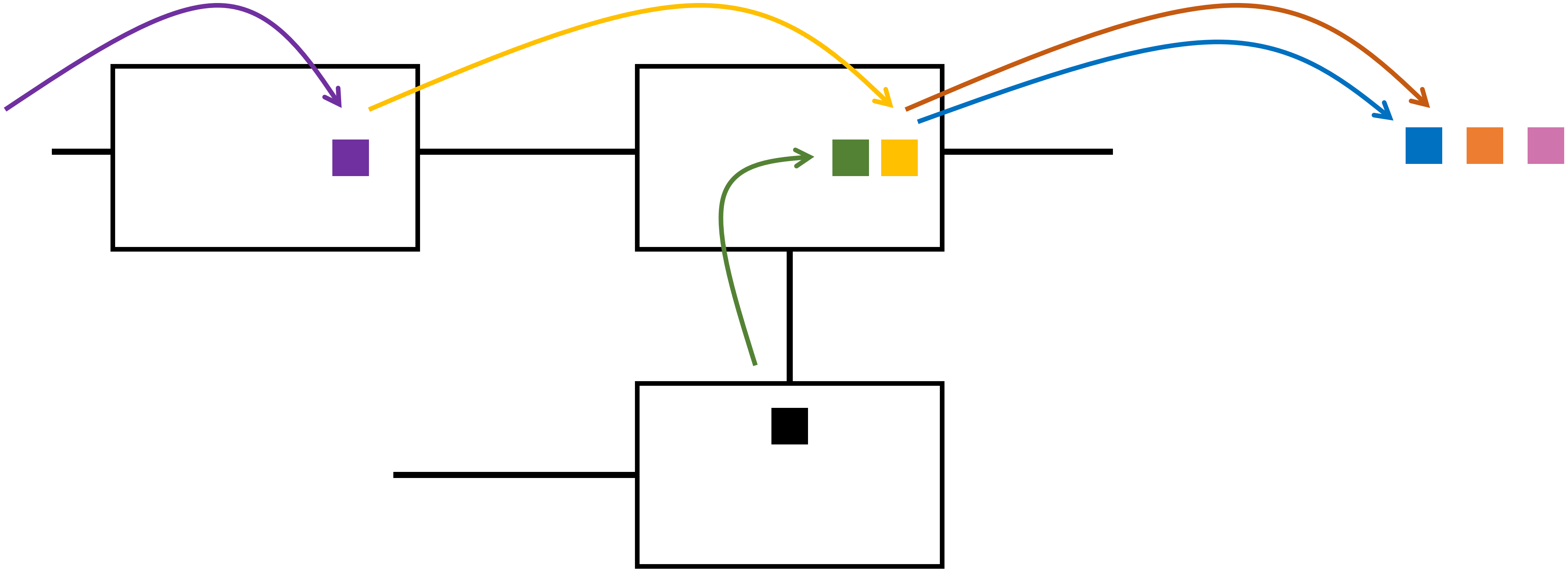
Every packet in the network moves one hop each cycle.

Cyclic Queuing and Forwarding



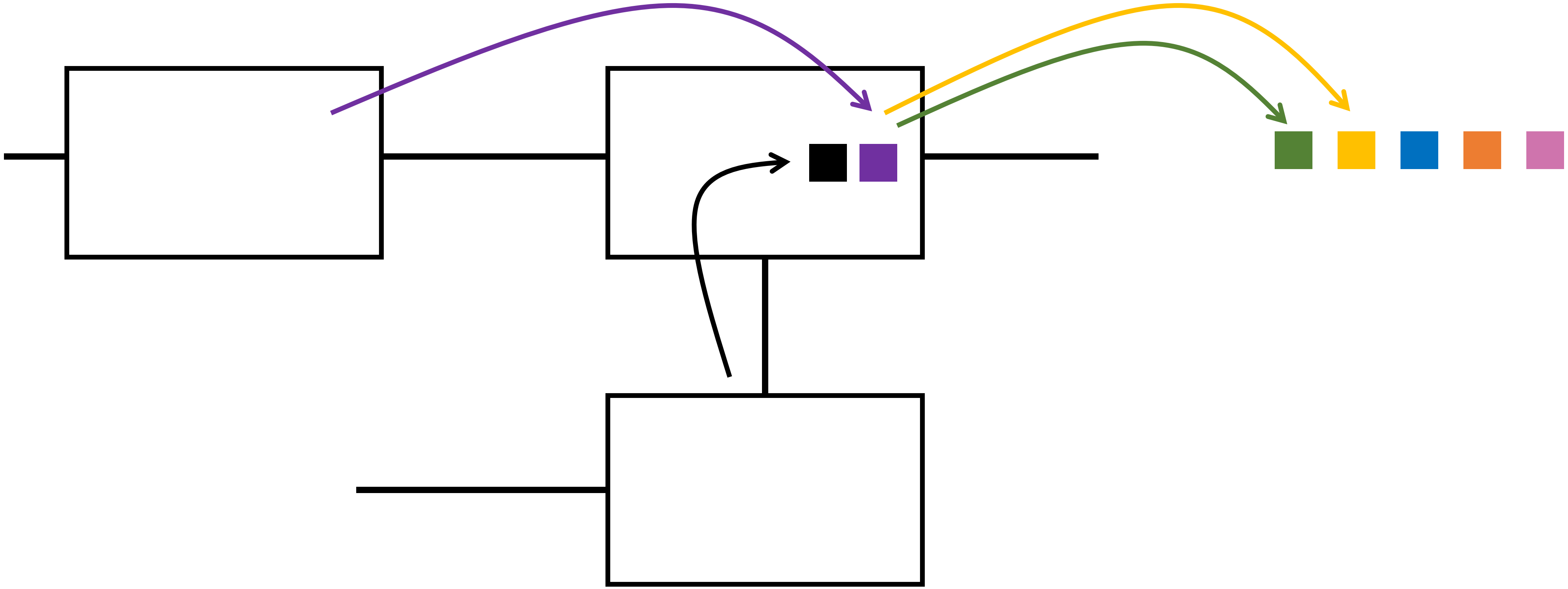
Every packet in the network moves one hop each cycle.

Cyclic Queuing and Forwarding



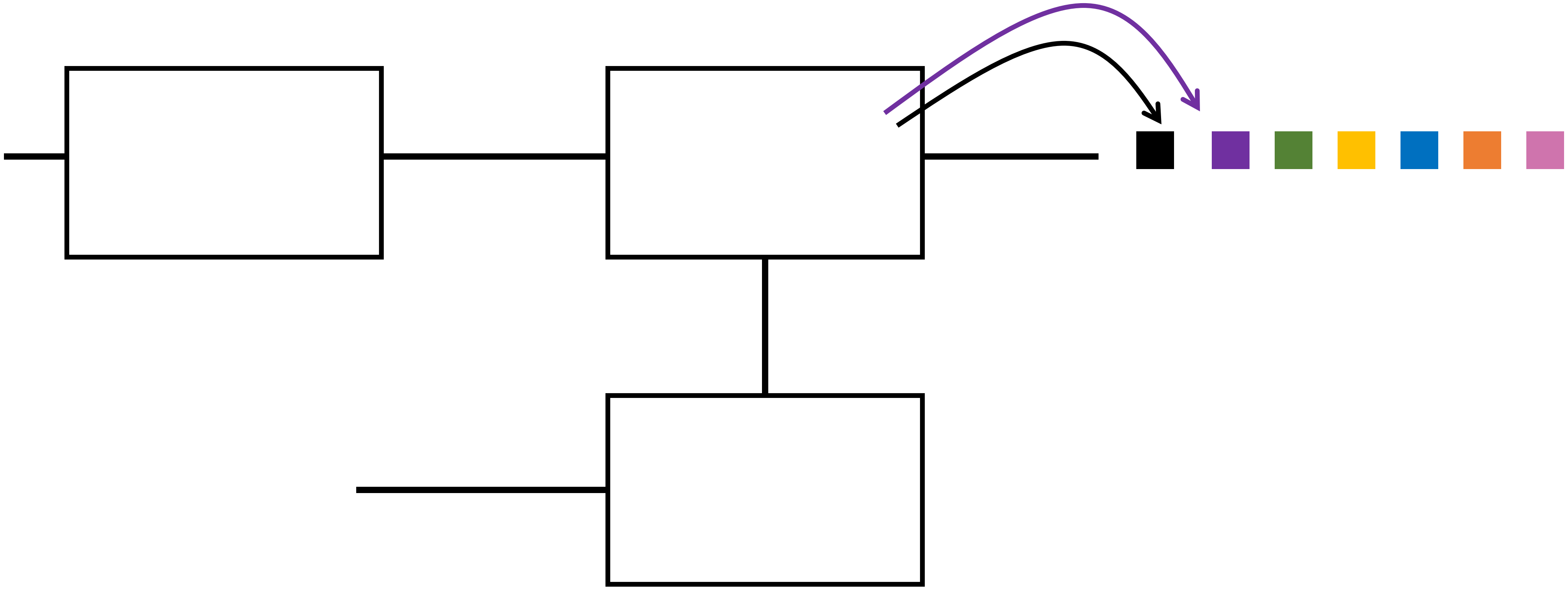
Every packet in the network moves one hop each cycle.

Cyclic Queuing and Forwarding



Every packet in the network moves one hop each cycle.

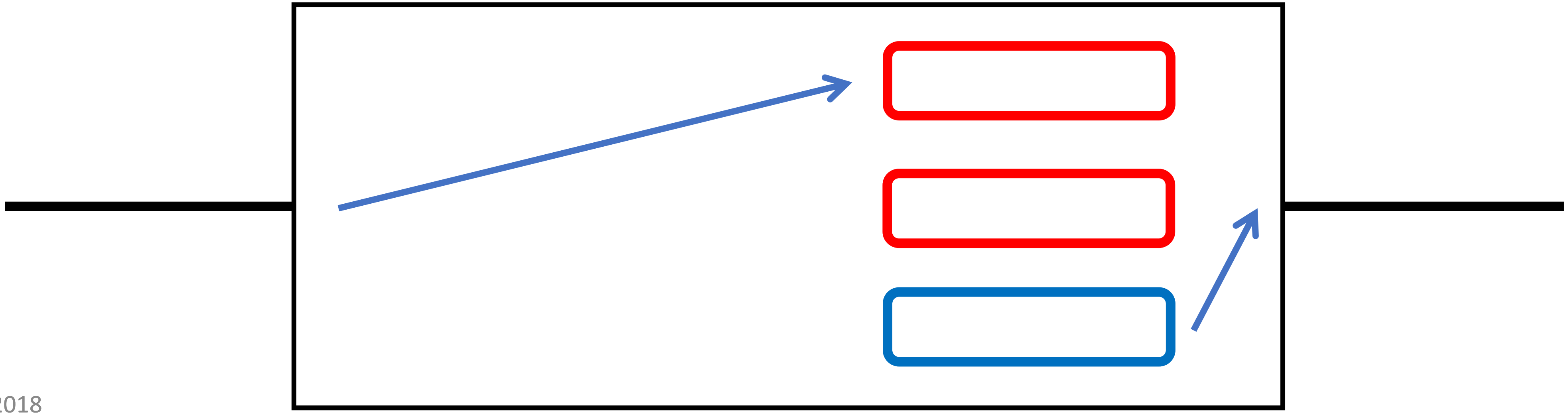
Cyclic Queuing and Forwarding



Every packet in the network moves one hop each cycle.

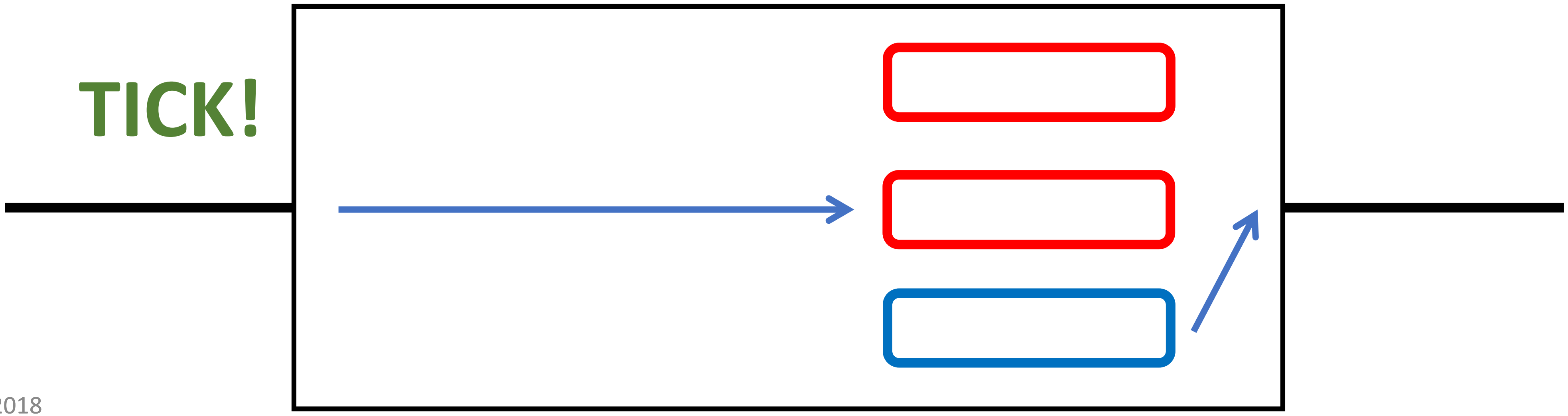
Cyclic Queuing and Forwarding

- Three-buffer version: Three buffers per port. Same as two-buffer version, but input buffer swap is out-of-phase with output buffer swap to allow for arbitrary link delay.



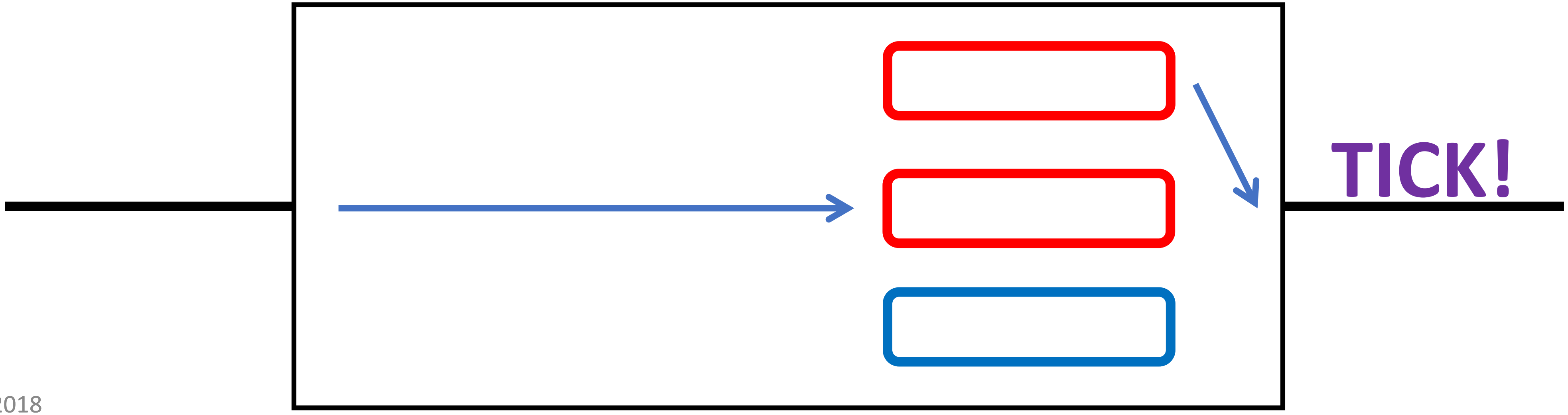
Cyclic Queuing and Forwarding

- Three-buffer version: Three buffers per port. Same as two-buffer version, but input buffer swap is out-of-phase with output buffer swap to allow for arbitrary link delay.



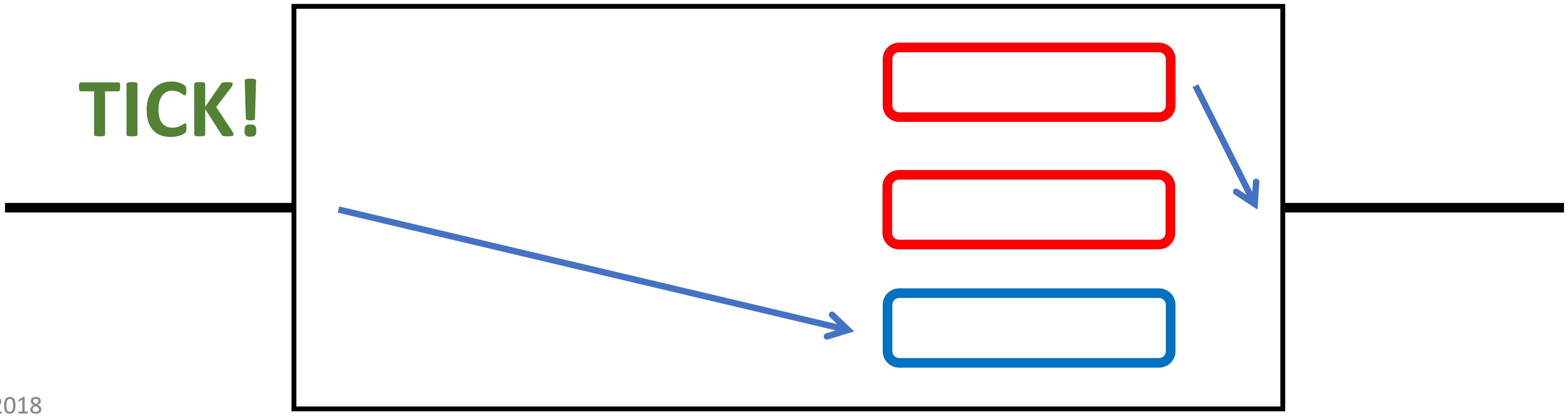
Cyclic Queuing and Forwarding

- Three-buffer version: Three buffers per port. Same as two-buffer version, but input buffer swap is out-of-phase with output buffer swap to allow for arbitrary link delay.



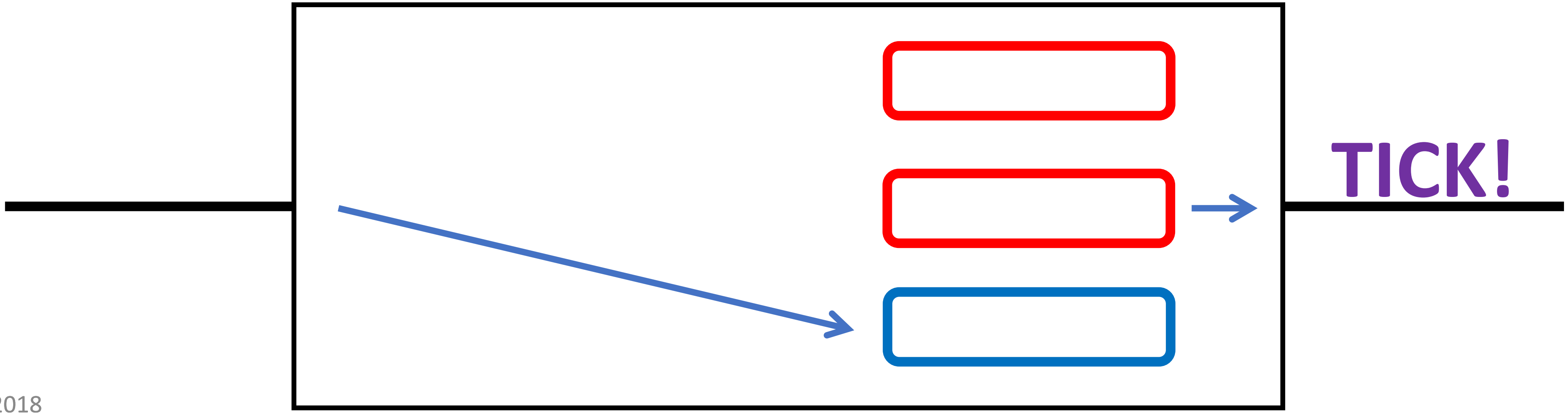
Cyclic Queuing and Forwarding

- Three-buffer version: Three buffers per port. Same as two-buffer version, but input buffer swap is out-of-phase with output buffer swap to allow for arbitrary link delay.



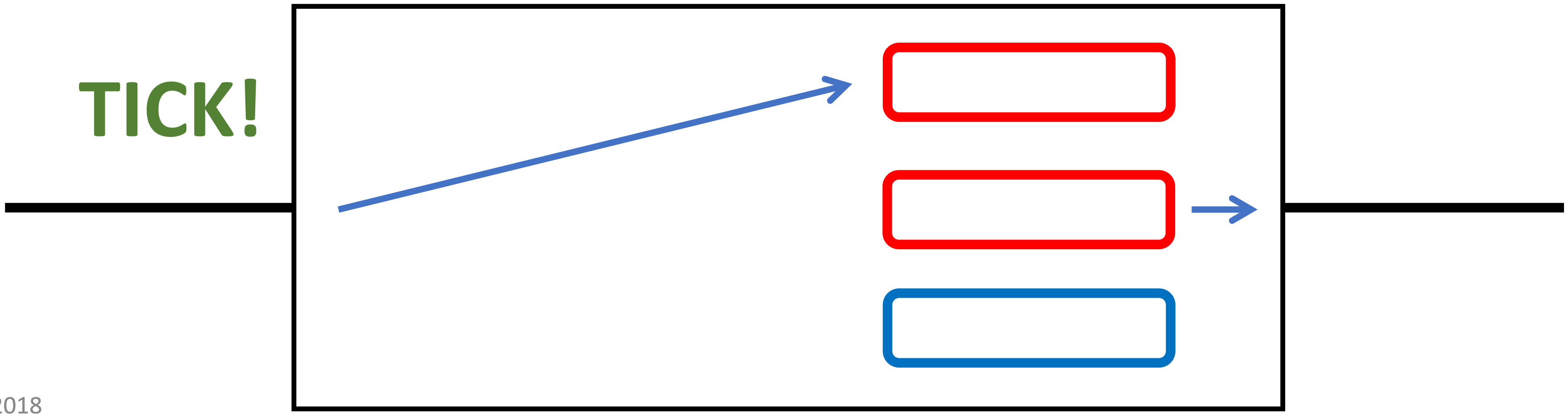
Cyclic Queuing and Forwarding

- Three-buffer version: Three buffers per port. Same as two-buffer version, but input buffer swap is out-of-phase with output buffer swap to allow for arbitrary link delay.



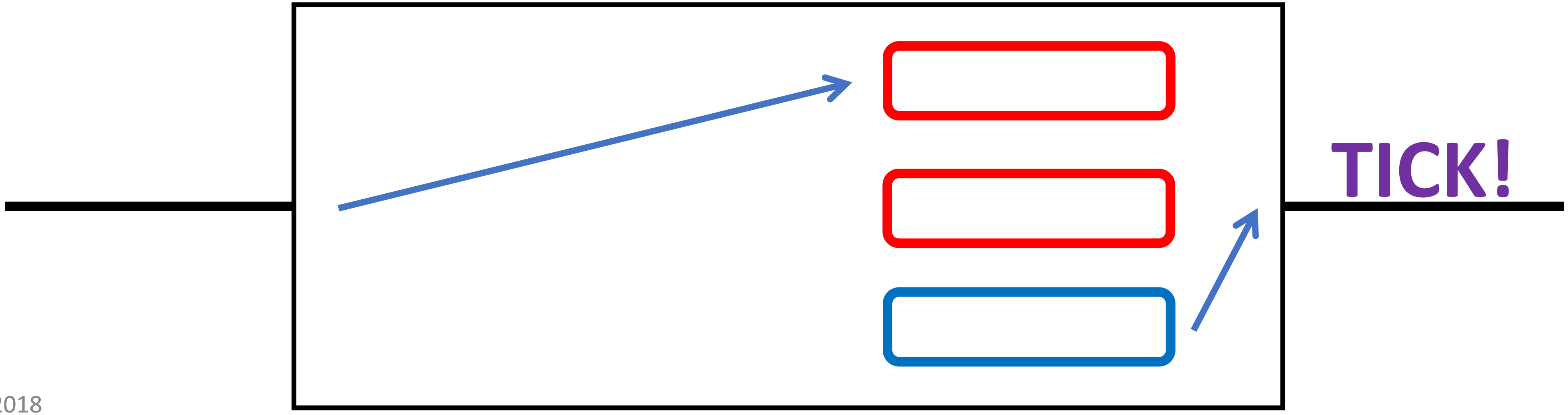
Cyclic Queuing and Forwarding

- Three-buffer version: Three buffers per port. Same as two-buffer version, but input buffer swap is out-of-phase with output buffer swap to allow for arbitrary link delay.



Cyclic Queuing and Forwarding

- Three-buffer version: Three buffers per port. Same as two-buffer version, but input buffer swap is out-of-phase with output buffer swap to allow for arbitrary link delay.



Cyclic Queuing and Forwarding

- Computing the delay is much simpler than the calculus used in bucket/credit schemes: every packet spends two or three cycles T_C at each hop, plus an integral number of cycles T_C (maybe 0) in transit and forwarding delay per hop.
- Resource allocation is trivial: total bandwidth cannot exceed that which can be transmitted in one cycle.
- Multiple buffer sets with different cycle times can run on a single port to supply different classes of service.