

## TCP Congestion Control over OBS

### OBS, TCP, Congestion Control

#### Key Points!

- Optical Burst Switching (OBS) Networks
  - Shift control complexity from optical to electrical layer
  - Lower the switching granularity
  - Dynamic bandwidth efficient
  - All-optical bufferless
- **However,**
  - Highly synchronized
  - Suffers from random burst contention losses

## Impact of Burst Dropping on TCP

- When a burst containing segments from different TCP sources is dropped, multiple TCP sources will throttle back their transmission
  - When a burst containing multiple segments from a single TCP flow is dropped, it can fatally affect that TCP flow transmission (concentrated loss)
  - Waste of routing, assembly, and signalling efforts performed at the IP-access network
  - Multiple simultaneous burst drops in a congested network can cause a network-wide loss in throughput (*global synchronization problem*)
    - May address a profound malicious impact on the long-lasting high-bandwidth TCP flows
- ➡ Need to develop an effective approach to solve the false congestion-detection problem

Random burst contention loss causes TCP sources to throttle back their transmission unnecessary!

## Impact of Burst Delay on TCP

- **Assembly delay penalty**
  - Typical assembly time is between few hundreds of *ns* to few hundreds of *ms* (*i.e.*, sometimes the burst assembly delay is few times larger than the link propagation delay!!)
  - Assembly delay takes place at both network edges and affects the *data segments* and the *Ack* packets.
- ➡ Reduces the link utilization since it increases the RTT and the RTO
- ➡ Introduces buffering constraints at the edge nodes
- **Retransmission penalty**
  - TCP is busy retransmitting lost packets in a burst that contain many packets
- ➡ Few new packets can be sent due to prolong retransmission period
- **Delay the first loss gain (DFL)**
  - Enlarges the transmission unit from packet to burst!
- ➡ The larger the burst size (# of assembled packets from the same source) the larger the throughput and the larger the DFL

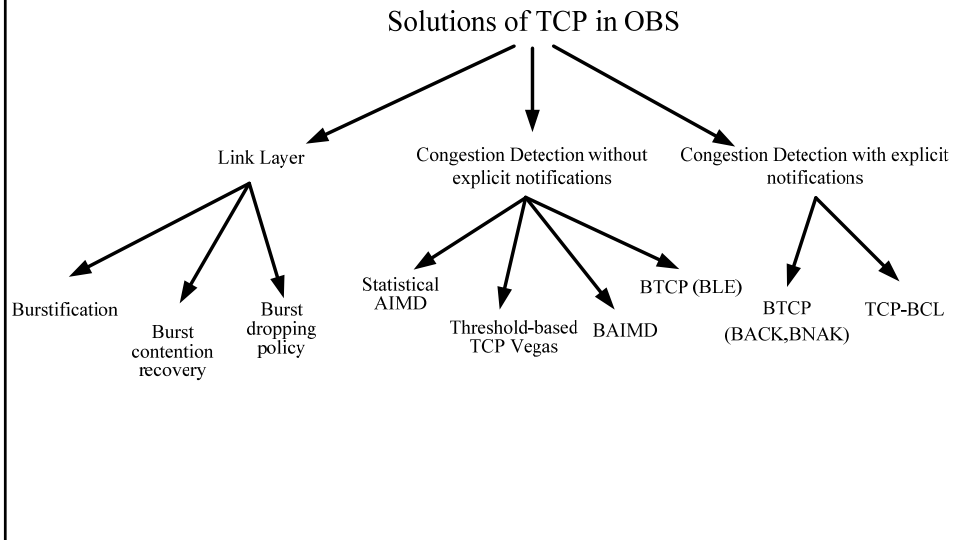
## OBS Taxonomy

- Barebone OBS
  - Burst contention results into an immediate burst loss!
- OBS with burst contention resolution (BCR)
  - Burst retransmission
  - Burst deflection
  - Burst segmentation
  - Burst Buffering (FDL)
    - ➡ Reduce burst loss probability
    - ➡ Introduce additional delay

## Methodologies

- Link-Layer solution
  - Mechanisms undergoing in the OBS domain
  - May not be sufficient; not adaptive to the TCP dynamics
- Congestion detection with explicit notification
  - Can effectively solve the false congestion problem
  - Cause signalling and nodal processing overhead
- Congestion detection without explicit notification
  - TCP senders estimate/evaluate the OBS congestion status
  - Less computation overhead

## Taxonomy of TCP over OBS



## Link-Layer Solutions

- Burstification Process
  - Adaptive burst assembly (AAP)
- Burst Contention Recovery
  - Retransmission
  - Deflection
  - Optical Buffering
  - Burst Segmentation
- TCP with burst Acknowledgment
- TCP Decoupling
- Retransmission-Count based Dropping Policy

## Statistical AIMD (SAIMD)

- Adapts the congestion detection without explicit notification
- Uses RTT to sense the network congestion
- Adopts the framework of GAIMD =>  $(\alpha, \beta)$  instead of (1,0.5)
- Derive a histogram curve by the statistics of the long-term measured RTT
  - The long-term RTT statistics represent the overall effect on the TCP sender due to network topology, routing strategy, and long-term traffic distribution
  - In the modeling, we assume the spectrum to follow a Normal distribution

## Statistical AIMD (SAIMD) (cont.)

- For every TCP segment loss
  1. Derive *avg\_rtt\_N* as the short-term average RTT, and position it in the spectrum of RTT obtained from the long-term statistics
  2. Obtain the confidence that the current network is in congestion state
  3. Use confidence to determine a beta value for *cwnd* adjustment corresponding to the segment drop
- If the short-term RTT is similar to or even less than the long-term RTT, the segment loss event is more likely caused by random contention => *cwnd* is slightly cut

## Extreme Cases

- Case 1: SAIMD starts in a congested network
  - The short and long term RTT statistics are very close. Hence  $\beta=1$
  - The *cwnd* will not be *reduced properly*
  - ➔ SAIMD eventually TO as a response to a packet loss and enters the slow start
- Case 2: SAIMD operates with no RTT variation
  - Rare event, but possible in barebone OBS.
  - The short and long term RTT statistics are also very close. Hence  $\beta=1$ .
  - ➔ SAIMD uses TO as a reaction to network congestion

## TCP Congestion Detection with Explicit Notifications

- Burst TCP with burst Ack/Nack
  - Partitions TCP rounds at the edge or core nodes
  - Acks are received at the TCP sender before the actual segment delivery
  - Nacks are triggered from the network core
  - ➔ Complicates the functionality of the network nodes and introduces extra computation overhead.
  - ➔ Violates the TCP semantics!
- TCP with burst contention loss (TCP-BCL)
  - Combines GAIMD ( $\alpha, \beta$ ) with burst-contention notification from the edge nodes
  - Uses burst-loss statistics to estimate the reason of the burst loss
  - ➔ Reduces the core-network computation overhead!

## TCP Congestion Detection without Explicit Notifications

- Burst TCP with Burst Length Estimation (BLE)
  - Solves the problem of TCP false TOs
  - Maintains a burst *cwnd* (*burst\_wd*)
  - Benefits from the TCP TO behavior and the number of segments sent
  - ➡ The accuracy of estimation is a challenge!
- Burst AIMD (BAIMD)
  - Estimates the network load to control the *cwnd* size
- Statistical AIMD (SAIMD)
  - Uses RTT statistics
  - Defines network congestion based on confidence
  - ➡ Fails to work on networks with no RTT variation!
- Threshold-based TCP Vegas
  - Reduces Vegas congestion reaction to non-congestion RTT increases
  - ➡ Optimizing threshold is a challenge!

## Delay-based TCPs (Vegas)

- TCP Vegas
  - Detects network congestion at earlier stages
  - Reduces the sending rate linearly
  - ➡ 20-50% fewer retransmissions
  - ➡ Improves throughput 37% to 71% compared to TCP Reno
- OBS
  - Burst contention occurs even at low traffic loads
    - Bufferless
    - One-way signaling
  - Contention resolution schemes
    - Burst deflection
    - Burst retransmission
    - ➡ Reduce burst loss probability
    - ➡ Introduce additional delay

## TCP Vegas over OBS

- Vegas can NOT detect congestion in a barebone OBS network
  - RTT in barebone OBS network varies little!!!

*What happens for Vegas over OBS network with burst retransmission or burst deflection?*

- Case 1: low traffic loads
  - Burst retransmission and burst deflection introduce higher burst delay
  - Vegas detects sudden RTT increase due to retransmission or deflection
  - Results in unnecessary reduction of the *cwnd* size!
- Case 2: high traffic loads
  - Higher congestion in OBS results in even higher burst delay
  - Vegas detects more frequent RTT increase

## Contributions<sup>1</sup>

Schemes	Solution Category		OBS Type		Devices Involved			Problems Addressed		
	Notification	Without Notification	Barebone	BCR	TCP Sender	OBS edge	OBS core	Random Burst Loss	False TO	Packet Reordering
Burst AIMD		√	√	√	√			√	√	
TCP-BCL	√		√	√	√	√		√	√	√
TCP-ENG	√		√	√	√	√	√	√	√	√
Statistical AIMD		√		√	√			√	√	
Threshold-based Vegas		√		√	√			√	√	

1. B. Shihada & P-H. Ho, "Transmission Control Protocol (TCP) in Optical Burst Switched Networks: Issues, Approaches, and Challenges", *IEEE Communications Surveys and Tutorials*, Second Quarter, Vol. 10, No. 2, 2008.

## Open Research Problems

- Burst delivery fairness
  - Merit-based channel allocation
  - Batch scheduling algorithms
  - Random early discard (RED)
- Slow convergence
  - To increase the TCP *cwnd* from half to full utilization of 10Gbps with 1.5 Kbyte packets, we need 1 hour with 100ms RTT and  $p < 10^{-9}$
- Effect of burst delay and dropping on
  - Fast TCP
  - High-Speed TCP
  - Binary Increase Control (BIC)
  - XCP
- TCP performance evaluation
  - Packet-oriented
  - Fluid Models
    - Very large number of TCP flows
    - Poisson arrival of loss events
    - Strong correlation between losses in one RTT
  - Synchronization models