CSCD 433/533
Advanced Networks
Spring 2016

Lecture 21
Congestion Control and Queuing Strategies
Topics

• Congestion Control and Resource Allocation
  • Flows
  • Types of Mechanisms
  • Evaluation Criteria
    • Effective
      • Throughput, Delay and Power
    • Fair
  • Queuing Strategies
Introduction

- The goal of congestion control is ....
- Regulate traffic flow to avoid saturating or overloading intermediate nodes in the network
Introduction

- So far, have looked at congestion control in terms of TCP in CSCD 330
- Congestion window
- Happens on the end systems
- TCP's philosophy is to let it happen, control the congestion by sending less
- Taught that routers don't have too much to do with congestion
- They do manage queues and can let end systems know that congestion is happening
Congestion: Effects

- Congestion is undesirable because it can cause:
  - Increased delay, due to queuing
  - Packet loss, due to buffer overflow
  - Reduced throughput, due to packet loss and retransmission

- Analogy: “rush hour” traffic
Buffering: A Solution?

- Buffering in switches/routers can help alleviate short term or transient congestion problems, but...

- Under sustained overload, buffers will still fill up, and packets will be lost
  - Only *defers* congestion problem

- More buffers means more queuing delay
  - Beyond a certain point, more buffering makes the congestion problem *worse*
    - Answer
    - Because of increased delay and retransmission
Congestion Control, Resource Allocation and Provisioning

- Are ... active areas of research
- Problem that crosses all network layers

- **Resource Allocation - Definition**
  - Process where network elements try to meet competing demands that applications have for network resources
  - Not always possible to meet all demands for network ... when too many packets queued then,

- **Congestion control - Definition**
  - Describes how network responds to or controls overload conditions
Congestion Control, Resource Allocation and Provisioning

- **Provisioning** is
  - Long term solution to congestion
  - Involves
    - Spare routers,
    - Purchase extra bandwidth and
    - Allocate resources
  - Helps with temporary congestion conditions
Layers Involved

- Congestion Control and Resource Allocation
  - Complimentary concepts
  - Networks can take an active role
    - Allocate resources,
    - If done well, congestion avoided
  - Ask … what layers of network is resource allocation and provisioning done?
  - Where is congestion control implemented?
Introduction

- Resource allocation, Provisioning
  - Network layer
- Congestion control
  - Network layer, transport layer, link layer
Flows in Networks

- Up to now, defined network as being either
  - Connectionless – datagram or
    Connection oriented – circuit or virtual circuit
- In reality … too rigid a definition
  - Datagram model says that each datagram travels independently of other datagrams
  - But, connections send stream of datagrams from a pair of hosts through a set of routers
  - Known as a **flow**, we have looked at these before
- **Flow** — Sequence of packets sent between source and destination pair following the same route through the network in a given amount of time
Flows in Networks

• Flows
  • Can be defined between source/dest hosts or between source/dest host/port pairs
  • A flow between a source/dest port pair is same as a channel
  • Flows are something routers can see and manipulate
  • Router maintains some state for flows
Flow – Routers Can See Them
Resource Allocation (RA)

• Many ways to classify RA mechanisms
• Examine a few in the next few slides

• Router Centric vs. Host Centric
  • **Router Centric**
    • Each router responsible for deciding when packets are forwarded
    • Selects when packets are to be dropped
  • **Host Centric**
    • End hosts observe network conditions and adjust behavior according
Resource Allocation (RA)

• **Host vs. Router**
  - Reality …. lots of overlap between two schemes
  - Both used to help with allocation of resources
  - Routers do some allocation and hosts adjust behavior in response to traffic conditions, How do hosts do this?

• **Reservation Based vs. Feedback Based**
  - Reservation based systems, host asks network for certain capacity at time flow is established
  - Each router allocates enough resources to satisfy request
  - If request can’t be satisfied, because of over committed resources, router rejects flow
Resource Allocation

- Reservation vs. Feedback Approach
  - Feedback Approach
    - End hosts send data without first reserving any capacity and then adjust sending rate according to feedback they receive
- Windows Based vs. Rate Based
  - Windows Based
    - TCP is an example of Windows based, where receiver advertises a window to sender
    - How much buffer space receiver has limits amount of data sender can transmit
    - Supports flow control
Resource Allocation (RA)

• **Windows Based vs. Rate Based**
  
  • **Windows Based**
    • Similar to TCP mechanism, routers use Window Advertisement to reserve buffer space
  
  • **Rate Based**
    • Use number bits/Sec receiver or network is able to absorb
    • Several multimedia protocols use rate based
    • For example if receiver says it can handle 1 Mbps, sender sends frames within that limit
Evaluation Criteria

• How do you evaluate Resource Allocation?
  • Two ways
    • Is it Effective? Does it work?
    • Is it Fair? Distributes resources among competitors
  • Effective
    • Throughput and Delay
    • Want as much throughput with as little delay
      • To increase throughput, increase packets
    • Yet is a relationship between throughput and delay
Evaluation Criteria

- **Effective continued …**
  - Higher number of packets in the network
  - Increases the length of queues at routers
  - Longer queues means more delay

- **Network Designers Specify the Relationship**
  - Called the **Power of the Network**
    \[
    \text{Power} = \frac{\text{Throughput}}{\text{Delay}}
    \]
  - Power might not be the best metric for judging resource allocation effectiveness
  - Based on network model that assumes infinite queues
Evaluation Criteria

- Power = Throughput / Delay
  - Objective is to maximize ratio
  - Function of how much load you place on network
  - The load in turn is set by resource allocation mechanism
Evaluation Criteria

• Ideally, resource allocation would operate at peek of curve

• Left of peak
  • Too conservative
  • Not allowing enough packets in

• Right of peak
  • Too generous, increases delay due to queues
Fair Resource Allocation

• Fairness
  • Another criterion to judge resource allocation
  • When several flows share a particular link
  • Want each flow to get an equal share of bandwidth
  • So, how do you quantify fairness of a congestion control mechanism?
Fairness Index

\[ f(x_1, x_2, \ldots, x_n) = \frac{\left( \sum x_j \right)^2}{n \sum x_j^2} \]

- If all \( n \) flows, get 1 unit of data per second
- The ratio becomes \( \frac{n^2}{n \times n} = 1 \), 1 is the maximum value if all flows have equal throughput
- Suppose, one flow receives a throughput of \( 1 + a \)

- Fairness index becomes: \( \frac{n^2 + 2na + a^2}{n^2 + 2na + na^2} \)
  With a larger denominator, the index < 1
- One flow is either getting more or less than rest, if \( a \) is a negative value.
Queueing Discipline

- Each router must implement some queuing discipline
  - Governs how packets are buffered while waiting to be transmitted
- Queuing Algorithm
  - Allocates bandwidth and buffer space
  - Affects latency of a packet – how long packet waits to get transmitted
- Two common Queuing Algorithms
  - **FIFO** – First In First Out
  - **FQ** – Fair Queueing
FIFO

- **FIFO**
  - If packet arrives first, first to get transmitted
  - If buffer is full, last one gets discarded
  - If last packet arrives and queue is full, packet gets dropped – *tail drop*
  - **FIFO and tail drop** – simplest queuing implementation
    - Other drop policies possible – more later
FIFO with Tail Drop

First in, first to transmit
As long as there is buffer space

Last arriving packets get dropped
Is FIFO Good for Congestion?

• Because **FIFO** is default queuing in Internet
  • Does nothing for congestion control
  • Easy to implement
  • Causes packets to be lost in bursts
  • Can lose many packets from a single flow…

• **What other queuing schemes are there?**

• **Priority Queue**
  • Idea, mark each packet with priority
  • Mark could be placed in Type of Service field in IP packet
Priority Queue

- Router implements separate FIFO queue per priority class
- Always transmit from highest priority queue first
- Could there be any problems with this?
Priority Queue

• With each priority queue
  • FIFO is used
• Doesn’t make QOS guarantees
• Just allows high priority packets preference

• Problem
  • High priority queue can starve out all other queues
  • For this to work, need limits on how much high priority traffic to allow
Queuing Disciplines

• **Fair Queuing (FQ)**
  • Yet another strategy for queues
  • Tries to maintain a fair allocation of resources without under utilizing network resources

• **What does it do?**
  • Maintains separate queue for each flow
  • Router services queues round robin
  • When flow sends too much traffic, queue fills
    • Packets from that flow are dropped
Fair Queuing

- Source can’t increase share of network capacity at expense of other flows
- Does not provide feedback for sources
  - No way to tell router’s state
- Segregate traffic into separate queues
- Keeps one sender from hogging all the bandwidth
- Should be used with end-to-end algorithm that does congestion control
Fair Queuing

- Each queue gets a turn as long as packets are queued
Notes on Fair Queuing

- **Link never left idle if packets in queue**
  - Queuing scheme called, *Work preserving*
  - If I am only one sending, can use entire bandwidth
  - But, when other flows happen … they get a share of bandwidth and my capacity will decrease
  - If link fully loaded and n flows sending data
    - Can’t use more than 1/nth of link bandwidth
- **If I try to send more**
  - My packets get increasingly large timestamps
  - Sit in queue longer awaiting transmission
  - Queue will eventually overflow
More Queue Management

- Previous schemes based on **creative buffer management**
  - Manage buffers and do not notify senders of congestion
- Next schemes anticipate congestion and take steps to prevent it from happening
  - More aggressive in approach and more intelligent in selection of packets to drop
Random Early Detection (RED)

- Queuing discipline with proactive packet discard
  - Picks packets and drops them, uses algorithm
  - Anticipate congestion and take early avoidance action
  - Does not penalize bursty traffic
  - Control average queue length (buffer size) within bounds… therefore, control average queuing delay
Discard probability is calculated for each packet arrival at the output queue based on:

- Current weighted average queue size, and
- Number of packets sent since previous packet discard
Generalized RED Algorithm

Calculate average queue size, $\text{avg}$

\[
\text{if } \text{avg} < \text{TH}_{\text{min}} \\
\text{Queue the packet}
\]

\[
\text{else if } \text{TH}_{\text{min}} \leq \text{avg} < \text{TH}_{\text{max}}
\]

Calculate probability $P_a$

With probability $P_a$

discard the packet

\[
\text{else with probability } 1 - P_a
\]

Queue the packet

\[
\text{else if } \text{avg} \geq \text{TH}_{\text{max}}
\]

Discard the packet
RED Algorithm

- Maintains running average of queue length
- If \( \text{avgq} < \min_{th} \) do nothing
  - Low congestion, send packets through
- If \( \text{avgq} > \max_{th} \), drop packet
  - Protection from misbehaving sources
  - Does not penalize all packets in a flow
  - Better distribution of dropped packets
Reading: Chapter 5.3

End
Topics

• Congestion Control and Resource Allocation
  • Flows
  • Types of Mechanisms
  • Evaluation Criteria
    • Effective
      • Throughput, Delay and Power
    • Fair
  • Queuing Strategies
Introduction

- So far, have looked at congestion control in terms of TCP in CSCD 330
- Congestion window
- Happens on the end systems
- TCP's philosophy is to let it happen, control the congestion by sending less
- Taught that routers don't have too much to do with congestion
- They do manage queues and can let end systems know that congestion is happening
Congestion Control, Resource Allocation and Provisioning

- Are ... active areas of research
- Problem that crosses all network layers

**Resource Allocation - Definition**
- Process where network elements try to meet competing demands that applications have for network resources
- Not always possible to meet all demands for network ... when too many packets queued then,

**Congestion control - Definition**
- Describes how network responds to or controls overload conditions
Congestion Control, Resource Allocation and Provisioning

• Provisioning is
  • Long term solution to congestion
  • Involves
    • Spare routers,
    • Purchase extra bandwidth and
    • Allocate resources
  • Helps with temporary congestion conditions
Layers Involved

• Congestion Control and Resource Allocation
  • Complimentary concepts
  • Networks can take an active role
    • Allocate resources,
    • If done well, congestion avoided
  • Ask … what layers of network is resource
    allocation and provisioning done?
  • Where is congestion control implemented?
Introduction

• Resource allocation, Provisioning
  • Network layer
• Congestion control
  • Network layer, transport layer, link layer
Flows in Networks

• Up to now, defined network as being either
  • Connectionless – datagram or
    Connection oriented – circuit or virtual circuit
• In reality … too rigid a definition
  • Datagram model says that each datagram travels independently of other datagrams
  • But, connections send stream of datagrams from a pair of hosts through a set of routers
  • Known as a flow, we have looked at these before
• Flow – Sequence of packets sent between source and destination pair following the same route through the network in a given amount of time
Flows in Networks

- **Flows**
  - Can be defined between source/dest hosts or between source/dest host/port pairs
  - A flow between a source/dest port pair is same as a channel
  - Flows are something routers can see and manipulate
  - Router maintains some state for flows
Flow – Routers Can See Them
Resource Allocation (RA)

• Many ways to classify RA mechanisms
• Examine a few in the next few slides

• Router Centric vs. Host Centric
  • **Router Centric**
    • Each router responsible for deciding when packets are forwarded
    • Selects when packets are to be dropped
  • **Host Centric**
    • End hosts observe network conditions and adjust behavior according
Resource Allocation (RA)

• **Host vs. Router**
  - Reality …. lots of overlap between two schemes
  - Both used to help with allocation of resources
  - Routers do some allocation and hosts adjust behavior in response to traffic conditions, *How do hosts do this?*

• **Reservation Based vs. Feedback Based**
  - Reservation based systems, host asks network for certain capacity at time flow is established
  - Each router allocates enough resources to satisfy request
  - If request can’t be satisfied, because of over committed resources, router rejects flow
Resource Allocation

- Reservation vs. Feedback Approach
  - Feedback Approach
    - End hosts send data without first reserving any capacity and then adjust sending rate according to feedback they receive
- Windows Based vs. Rate Based
  - Windows Based
    - TCP is an example of Windows based, where receiver advertises a window to sender
    - How much buffer space receiver has limits amount of data sender can transmit
    - Supports flow control
Resource Allocation (RA)

• Windows Based vs. Rate Based
  
  • Windows Based
    • Similar to TCP mechanism, routers use Window Advertisement to reserve buffer space
  
  • Rate Based
    • Use number bits/Sec receiver or network is able to absorb
    • Several multimedia protocols use rate based
    • For example if receiver says it can handle 1 Mbps, sender sends frames within that limit
Evaluation Criteria

• How do you evaluate Resource Allocation?
  • Two ways
    • Is it Effective? Does it work?
    • Is it Fair? Distributes resources among competitors
  • Effective
    • Throughput and Delay
    • Want as much throughput with as little delay
      • To increase throughput, increase packets
    • Yet is a relationship between throughput and delay
Evaluation Criteria

• **Effective continued …**
  • Higher number of packets in the network
  • Increases the length of queues at routers
  • Longer queues means more delay

• **Network Designers Specify the Relationship**
  • Called the **Power of the Network**
    \[
    \text{Power} = \frac{\text{Throughput}}{\text{Delay}}
    \]
  • Power might not be the best metric for judging resource allocation effectiveness
  • Based on network model that assumes infinite queues
Evaluation Criteria

• Power = Throughput / Delay
  • Objective is to maximize ratio
  • Function of how much load you place on network
  • The load in turn is set by resource allocation mechanism
Evaluation Criteria

• Ideally, resource allocation would operate at peek of curve
• Left of peak
  • Too conservative
  • Not allowing enough packets in
• Right of peak
  • Too generous, increases delay due to queues
Fair Resource Allocation

• Fairness
  • Another criterion to judge resource allocation
  • When several flows share a particular link
  • Want each flow to get an equal share of bandwidth
  • So, how do you quantify fairness of a congestion control mechanism?
Fairness Index

\[ f(x_1, x_2, \ldots, x_n) = \left( \frac{\sum x_j}{n \sum x_j^2} \right)^2 \]

- If all \( n \) flows, get 1 unit of data per second
- The ratio becomes \( \frac{n^2}{n \times n} = 1 \), 1 is the maximum value if all flows have equal throughput
- Suppose, one flow receives a throughput of \( 1 + a \)

- Fairness index becomes: \( \frac{n^2 + 2na + a^2}{n^2 + 2na + na^2} \)
  With a larger denominator, the index < 1
- One flow is either getting more or less than rest, if \( a \) is a negative value.
Queuing Discipline

- Each router must implement some queuing discipline
  - Governs how packets are buffered while waiting to be transmitted
- Queuing Algorithm
  - Allocates bandwidth and buffer space
  - Affects latency of a packet – how long packet waits to get transmitted
- Two common Queuing Algorithms
  - **FIFO** – First In First Out
  - **FQ** – Fair Queueing
FIFO

- FIFO
  - If packet arrives first, first to get transmitted
  - If buffer is full, last one gets discarded
  - If last packet arrives and queue is full, packet gets dropped – **tail drop**
- FIFO and tail drop – simplest queuing implementation
  - Other drop policies possible – more later
FIFO with Tail Drop

First in, first to transmit
As long as there is buffer space

Last arriving packets get dropped
Is FIFO Good for Congestion?

• Because **FIFO** is default queuing in Internet
  • Does nothing for congestion control
  • Easy to implement
  • Causes packets to be lost in bursts
  • Can lose many packets from a single flow…

• What other queuing schemes are there?
• **Priority Queue**
  • Idea, mark each packet with priority
  • Mark could be placed in Type of Service field in IP packet
Priority Queue

- Router implements separate FIFO queue per priority class
- Always transmit from highest priority queue first
- Could there be any problems with this?

Figure – Multilevel Feedback Queue Scheduling
Priority Queue

- With each priority queue
  - FIFO is used
- Doesn’t make QOS guarantees
- Just allows high priority packets preference

**Problem**
- High priority queue can starve out all other queues
- For this to work, need limits on how much high priority traffic to allow
Queuing Disciplines

• **Fair Queuing (FQ)**
  • Yet another strategy for queues
  • Tries to maintain a fair allocation of resources without under-utilizing network resources

• **What does it do?**
  • Maintains separate queue for each flow
  • Router services queues round robin
  • When flow sends too much traffic, queue fills
    • Packets from that flow are dropped
Fair Queuing

• Source can’t increase share of network capacity at expense of other flows
• Does not provide feedback for sources
  • No way to tell router’s state
• Segregate traffic into separate queues
• Keeps one sender from hogging all the bandwidth
• Should be used with end-to-end algorithm that does congestion control
Fair Queuing

• Each queue gets a turn as long as packets are queued
Notes on Fair Queuing

- **Link never left idle if packets in queue**
  - Queuing scheme called, *Work preserving*
  - If I am only one sending, can use entire bandwidth
  - But, when other flows happen ... they get a share of bandwidth and my capacity will decrease
  - If link fully loaded and n flows sending data
    - Can’t use more than 1/nth of link bandwidth

- **If I try to send more**
  - My packets get increasingly large timestamps
  - Sit in queue longer awaiting transmission
  - Queue will eventually overflow
More Queue Management

• Previous schemes based on **creative buffer management**
  • Manage buffers and do not notify senders of congestion
• Next schemes anticipate congestion and take steps to prevent it from happening
  • More aggressive in approach and more intelligent in selection of packets to drop
Random Early Detection (RED)

- Queuing discipline with proactive packet discard
  - Picks packets and drops them, uses algorithm
  - Anticipate congestion and take early avoidance action
  - Does not penalize bursty traffic
  - Control average queue length (buffer size) within bounds… therefore, control average queuing delay
RED Buffer Management

Discard probability is calculated for each packet arrival at the output queue based on:

- Current weighted average queue size, and
- Number of packets sent since previous packet discard
Generalized RED Algorithm

Calculate average queue size, $\text{avg}$
\begin{enumerate}
\item \textbf{if} $\text{avg} < TH_{min}$
  \begin{itemize}
  \item Queue the packet
  \end{itemize}
\item \textbf{else if} $TH_{min} \leq \text{avg} < TH_{max}$
  \begin{itemize}
  \item Calculate probability $P_a$
  \item With probability $P_a$
  \item discard the packet
  \item \textbf{else with probability} $1 - P_a$
  \item Queue the packet
  \end{itemize}
\item \textbf{else if} $\text{avg} \geq TH_{max}$
  \begin{itemize}
  \item Discard the packet
  \end{itemize}
\end{enumerate}
RED Algorithm

- Maintains running average of queue length
- If avgq < min\textsubscript{th} do nothing
  - Low congestion, send packets through
- If avgq > max\textsubscript{th}, drop packet
  - Protection from misbehaving sources
  - Does not penalize all packets in a flow
  - Better distribution of dropped packets
Reading: Chapter 5.3

End