Convex Optimization and Congestion Control

Fabian Wirth

University of Würzburg

23.07. – 03.08.2012

Outline

• Part I:	Convexity and Convex Functions	Lectures 1, 2, 3
• Part II:	Convex Optimization	Lectures 4 and 5
• Part III:	Numerical Methods	Lectures 6 and 7
• Part IV:	Congestion Control	Lecture 8
• Part V:	Utility Based Congestion Control	Lecture 9
• Part VI:	Miscellaneous Problems in Networks	Lecture 10 (we shall see)

Part III: Numerical Methods

- III.1: Unconstrained Problems
- III.2: Constrained Problems
- III.3: Interior Point Methods

Outline

1 Unconstrained Problems

2 Constrained Problems

3 Interior Point Methods

Outline

1 Unconstrained Problems

2 Constrained Problems

3 Interior Point Methods

Outline

1 Unconstrained Problems

2 Constrained Problems

3 Interior Point Methods

Problem Statement

The Convex Optimization Problem with Inequality Constraint

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_j(x) \leq 0 \qquad\qquad j = 1, \dots, m \\ & Ax = b\,. \end{aligned} \tag{1}$$

Assumptions

(i) f, g_j are convex and twice continuously differentiable,

(ii) $\mathcal{D} \subset \mathbb{R}^n$ is open,

(iii) $A \in \mathbb{R}^{p \times n}$ with $\operatorname{rank} A = p < n$

(iv) Slater condition is satisfied, i.e. there exists a strictly feasible point.

A Reformulation

Replace Inequality Constraints by Indicator Function

Define I_- as the indicator function of the nonpositive reals by

$$I_-(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ \infty & \text{if } x > 0\,. \end{cases}$$

The problem can be reformulated as

$$\begin{aligned} \text{minimize} \quad & f(x) + \sum_{j=1}^{m} I_-(g_j(x)) \\ \text{subject to} \quad & Ax = b\,, \end{aligned}$$

This new formulation has no inequality constraints, but an unpleasant discontinuity at the boundary of the feasibility set.

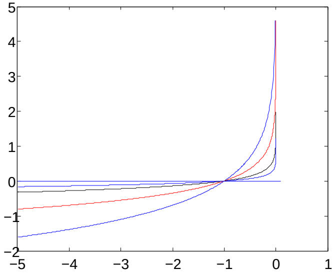
Approximating the Indicator Function

The Approximation

Define I_- as the indicator function of the nonpositive reals by

$$I_-(x) = \begin{cases} 0 & \text{if } x \le 0 \\ \infty & \text{if } x > 0\,. \end{cases}$$

$$\hat{I}_t(x) = -\frac{1}{t}\log(-x)\,, \quad x \in (-\infty, 0)\,.$$



A Family of Approximating Optimization Problems

$$\begin{aligned} \text{minimize} \quad & f(x) + \sum_{j=1}^{m} \hat{I}_t(g_j(x)) \\ \text{subject to} \quad & Ax = b\,, \end{aligned}$$

or equivalently

$$\begin{aligned} \text{minimize} \quad & f(x) + \sum_{j=1}^{m} -\frac{1}{t}\log(-g_j(x)) \\ \text{subject to} \quad & Ax = b\,. \end{aligned}$$

The Logarithmic Barrier Function

$$\phi(x) := -\sum_{j=1}^{m} \log(-g_j(x))$$

Central Path

Central Path

We consider the equivalent problem obtained by multiplying the objective function by t. We thus consider

$$\begin{aligned} \text{minimize} \quad & tf(x) + \phi(x) \\ \text{subject to} \quad & Ax = b\,. \end{aligned}$$

Definition

Consider the family of optimization problems for $t > 0$. Assume that for each $t > 0$ there is a unique optimal solution $x^*(t)$. The *central path* is defined as the set of optimal points for the problem with parameter t, i.e.

$$\{x^*(t) \mid t > 0\}\,.$$

The points $x^*(t)$ are called *central points*.

Central Path and KKT

KKT Conditions for the Central Path

the central points are characterized by the necessary and sufficient conditions that they are strictly feasible, i.e.

$$Ax^*(t) = b\,, \quad g_j(x^*(t)) < 0\,, \quad j = 1, \dots, m$$

and there exists a $\nu^* \in \mathbb{R}^p$ such that

$$\begin{aligned} 0 &= t\nabla f(x^*(t)) + \nabla\phi(x^*(t)) + A^\top \nu^* \\ &= t\nabla f(x^*(t)) + \sum_{j=1}^{m} \frac{1}{-g_j(x^*(t))} \nabla g_j(x^*(t)) + A^\top \nu^*\,. \end{aligned}$$

A Bound Using Duality

Lagrange Multipliers

Using strict feasibility of x^* we can define

$$\lambda_j^*(t) = -\frac{1}{t g_j(x^*(t))} > 0\,,$$

Hence $(\lambda^*(t), \nu^*/t)$ is feasible for the dual problem.

Optimal Dual Points

We now divide this equation from the KKT conditions by t

$$0 = t\nabla f(x^*(t)) + \sum_{j=1}^{m} \frac{1}{-g_j(x^*(t))} \nabla g_j(x^*(t)) + A^\top \nu^* .$$

and obtain

$$\nabla f(x^*(t)) + \sum_{j=1}^{m} \lambda_j^* \nabla g_j(x^*(t)) + \frac{1}{t} A^\top \nu^* = 0 .$$

Optimality

$x^*(t)$ is a minimizer for the Lagrangian function $L(x, \lambda^*, \nu^*/t)$.

Bounds

As $x^*(t)$ is a minimizer for the Lagrangian function $L(x, \lambda^*, \nu^*/t)$ we have

$$\begin{aligned} g_L(\lambda^*, \nu^*/t) &= f(x^*(t)) + \sum_{j=1}^{m} \lambda_j^* g_j(x^*(t)) + \frac{1}{t}(\nu^*)^\top (Ax^*(t) - b) \\ &= f(x^*(t)) - \frac{m}{t}, \end{aligned}$$

Bounds

As $x^*(t)$ is a minimizer for the Lagrangian function $L(x, \lambda^*, \nu^*/t)$ we have

$$\begin{aligned} g_L(\lambda^*, \nu^*/t) &= f(x^*(t)) + \sum_{j=1}^{m} \lambda_j^* g_j(x^*(t)) + \frac{1}{t}(\nu^*)^\top (Ax^*(t) - b) \\ &= f(x^*(t)) - \frac{m}{t}, \end{aligned}$$

Bounding the optimal value

The duality gap between $x^*(t)$ and $g_L(\lambda^*, \nu^*/t)$ is m/t, so

$$f(x^*(t)) - p^* \leq \frac{m}{t}.$$

This shows

$$f(x^*(t)) \to p^* \quad \text{as} \quad t \to \infty.$$

KKT Interpretation

KKT

$x^*(t)$ is characterized by the conditions that there exist (λ^*, ν^*) such that

$$\begin{array}{rcll}
g_j(x^*(t)) & \leq & 0, & j = 1, \dots, m \\
Ax^*(t) - b & = & 0 & \\
\lambda_j^* & \geq & 0 & j = 1, \dots, m \\
-\lambda_j^* g_j(x^*) & = & \dfrac{1}{t} & j = 1, \dots, m \\
\nabla f(x^*(t)) + \displaystyle\sum_{j=1}^{m} \lambda_j^* \nabla g_j(x^*(t)) + A^\top \nu^* & = & 0\,. &
\end{array}$$

KKT Interpretation

KKT

$x^*(t)$ is characterized by the conditions that there exist (λ^*, ν^*) such that

$$\begin{aligned} g_j(x^*(t)) &\leq 0, && j = 1, \dots, m \\ Ax^*(t) - b &= 0 && \\ \lambda_j^* &\geq 0 && j = 1, \dots, m \\ -\lambda_j^* g_j(x^*) &= \frac{1}{t} && j = 1, \dots, m \\ \nabla f(x^*(t)) + \sum_{j=1}^{m} \lambda_j^* \nabla g_j(x^*(t)) + A^\top \nu^* &= 0\,. && \end{aligned}$$

The central points are approximately solutions of the KKT conditions.
This approximation improves as $t \to \infty$

The Barrier Method

Algorithm

Input Strictly feasible point $x \in \operatorname{dom} f$, $t^0 > 0$, $\mu > 1$ and error bound $\varepsilon > 0$.

Repeat

1. Centering step:
   Compute $x^*(t)$ by minimizing $tf(x) + \phi(x)$ subject to $Ax = b$
   Use Newton's method.
2. Update: $x := x^*(t)$
3. Stopping criterion: **Quit** if $m/t \le \varepsilon$.
4. Increase t: $t := \mu t$.

Barrier Method

Choosing μ: A tradeoff

- μ is close to 1: t^k, t^{k+1} are close to one another. $x^*(t^k)$ may be expected to be a good initial guess for the next problem. The inner Newton algorithm may be expected to converge quickly. But: many outer approximations as t increases rather slowly.
- μ large: Few outer iterations. But: inner iterations may need a large number of steps.

Barrier Method

Choosing μ: A tradeoff

- μ is close to 1: t^k, t^{k+1} are close to one another. $x^*(t^k)$ may be expected to be a good initial guess for the next problem. The inner Newton algorithm may be expected to converge quickly. But: many outer approximations as t increases rather slowly.
- μ large: Few outer iterations. But: inner iterations may need a large number of steps.

Choosing t^0

- t^0 large: very likely the resulting optimization problem is numerically unpleasant and the initial Newton algorithm may take considerable time.
- t^0 small: many outer steps again reducing performance.

$$\inf_{\nu} \left\| t\nabla f(x^0) + \nabla\phi(x^0) + A^\top \nu \right\|_2$$

Outline

• Part I:	Convexity and Convex Functions	Lectures 1, 2, 3
• Part II:	Convex Optimization	Lectures 4 and 5
• Part III:	Numerical Methods	Lectures 6 and 7
• Part IV:	Congestion Control	Lecture 8
• Part V:	Utility Based Congestion Control	Lecture 9
• Part VI:	Miscellaneous Problems in Networks	Lecture 10 (we shall see)

Part IV: Congestion Control

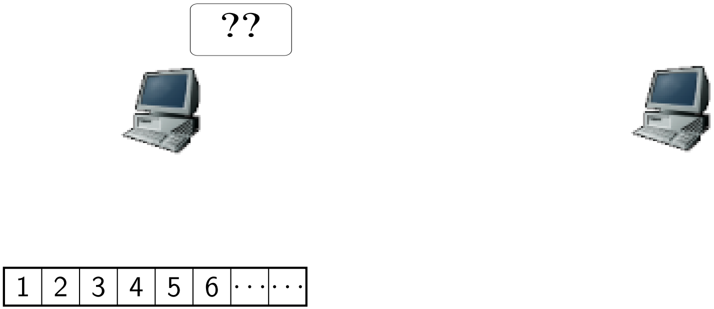
- IV.1: Basics of TCP
- IV.2: Dynamics of Deterministic AIMD
- IV.3: Utility Based Congestion Control

Outline

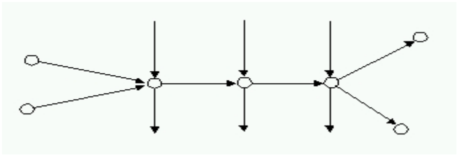
4 Modeling of TCP flows
- The AIMD algorithm
- A linear model I
- A linear model II
- Model Validation

5 Analysis of TCP

Data transfer

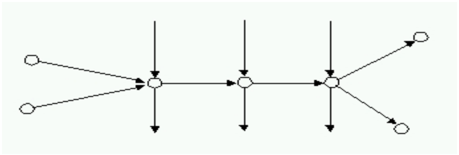


The network consists of a number of sources and sinks that communicate



via network links ('wires') and routers ('queues').

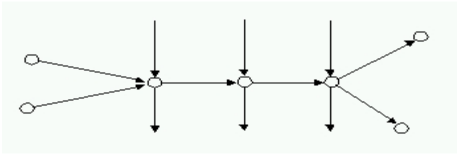
The network consists of a number of sources and sinks that communicate



via network links ('wires') and routers ('queues').

Packets are sent from sources to sinks and 'in flight' packets are on the wire or in the network queues.

The network consists of a number of sources and sinks that communicate



via network links ('wires') and routers ('queues').
Packets are sent from sources to sinks and 'in flight' packets are on the wire or in the network queues.
If packets arrive at a router at a greater rate than the service rate, the queue starts to fill and eventually packets are lost.

Questions:

How much information should be provided to the individual source, so that it can send data in an intelligent way ?

Questions:

How much information should be provided to the individual source, so that it can send data in an intelligent way ?

Desired network properties:

Network fairness and efficient use of network resources.

Questions:

How much information should be provided to the individual source, so that it can send data in an intelligent way ?

Desired network properties:

Network fairness and efficient use of network resources.

• (One) definition of network fairness is that n sources competing for a bandwidth B obtain a share of B/n each.

Questions:

How much information should be provided to the individual source, so that it can send data in an intelligent way ?

Desired network properties:

Network fairness and efficient use of network resources.

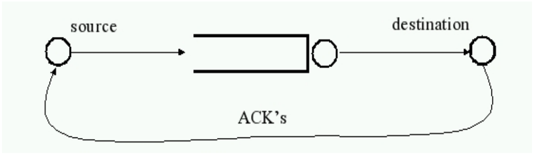
• (One) definition of network fairness is that n sources competing for a bandwidth B obtain a share of B/n each.
• Network resources such as available bandwidth should be fully utilised by the network sources as they becomes available without causing congestion collapse.

Reliable Delivery

TCP is an acknowledgement based protocol that ensures reliable packet delivery and uses “minimal” information.

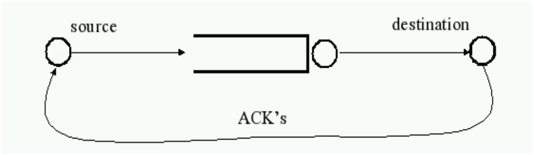
Reliable Delivery

TCP is an acknowledgement based protocol that ensures reliable packet delivery and uses “minimal” information.



Reliable Delivery

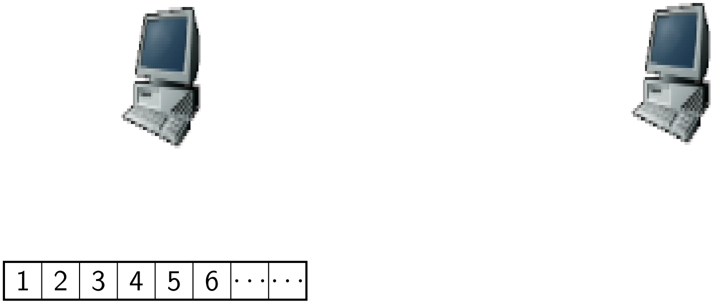
TCP is an acknowledgement based protocol that ensures reliable packet delivery and uses “minimal” information.



Missing acknowledgements are used to infer lost packets and to initiate a resend - many versions of TCP.
Senders receive information about lost packets only after a transport delay.
This delay, the RTT, is time-varying due to the nature of the queues.

The AIMD algorithm

AIMD = additive increase, multiplicative decrease.

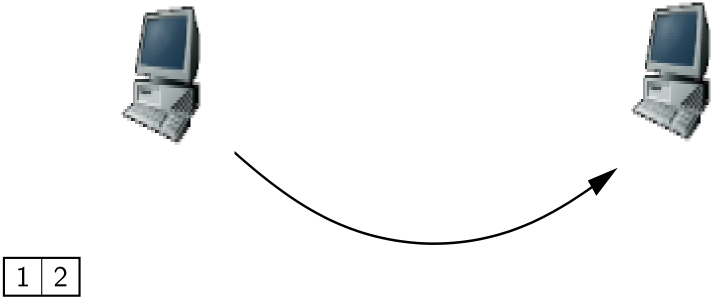


The AIMD algorithm

AIMD = additive increase, multiplicative decrease.

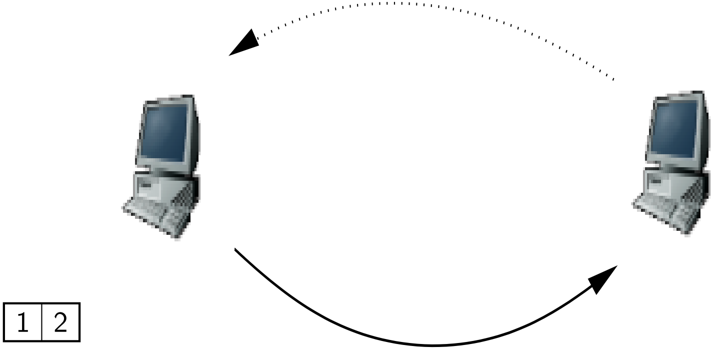
The AIMD algorithm

AIMD = additive increase, multiplicative decrease.



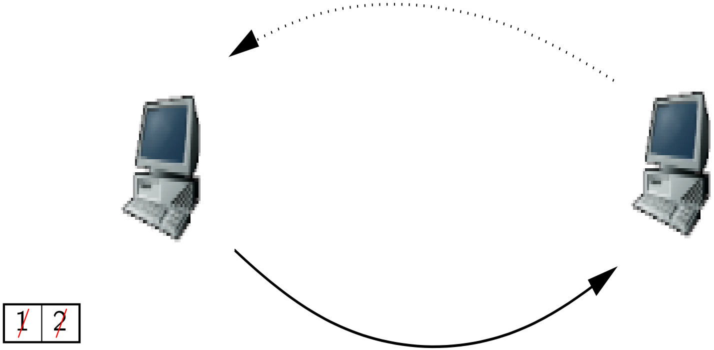
The AIMD algorithm

AIMD = additive increase, multiplicative decrease.



The AIMD algorithm

AIMD = additive increase, multiplicative decrease.

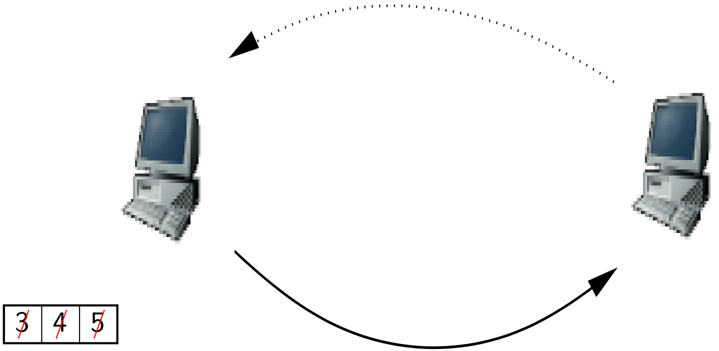


The AIMD algorithm

AIMD = additive increase, multiplicative decrease.

The AIMD algorithm

AIMD = additive increase, multiplicative decrease.

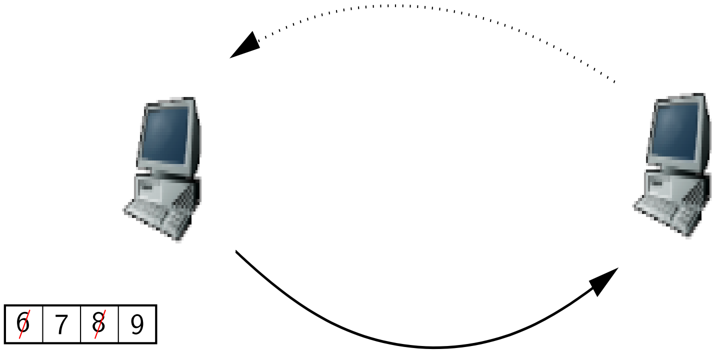


The AIMD algorithm

AIMD = additive increase, multiplicative decrease.

The AIMD algorithm

AIMD = additive increase, multiplicative decrease.



The AIMD algorithm

AIMD = additive increase, multiplicative decrease.

The AIMD algorithm

Current TCP implementations are characterized by two constants:

The AIMD algorithm

Current TCP implementations are characterized by two constants:

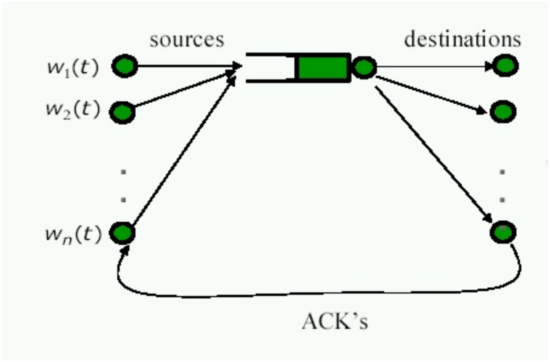
- the additive increase constant $\alpha \geq 1$.
  (In your computer $\alpha = 1$.)

The AIMD algorithm

Current TCP implementations are characterized by two constants:

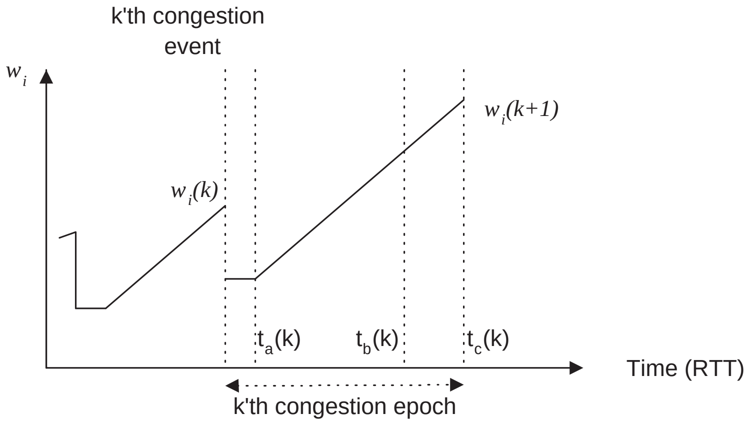
- the additive increase constant $\alpha \geq 1$.
  (In your computer $\alpha = 1$.)
- the multiplicative decrease factor $0 < \beta < 1$.
  (In your computer $\beta = 1/2$.)

Many sources, one bottleneck

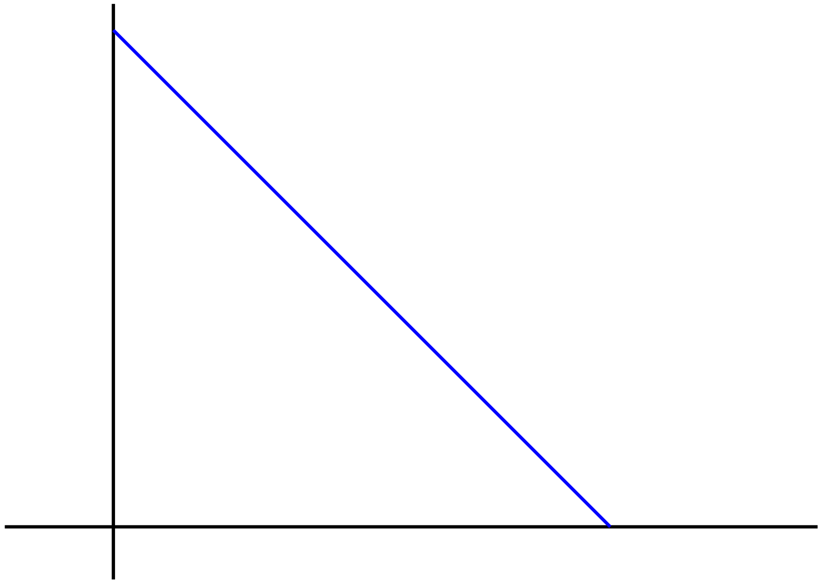


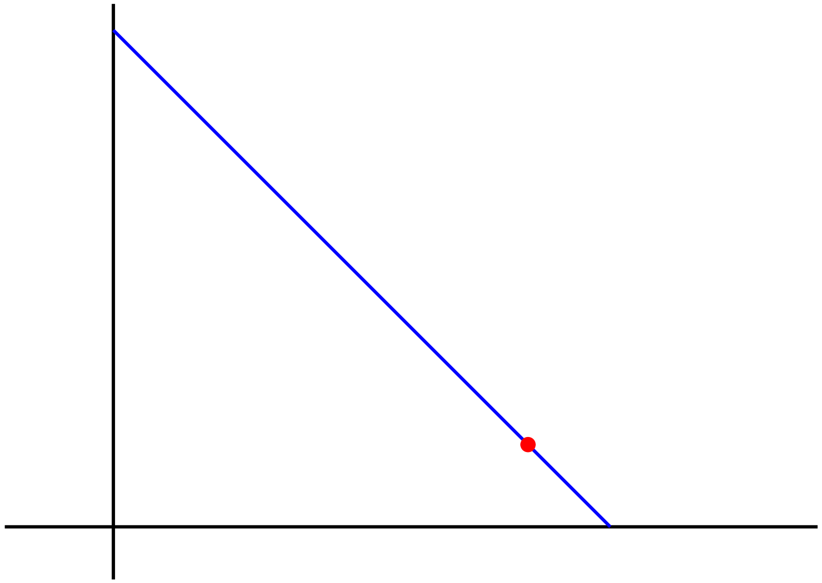
$w_i(k) \triangleq$ the number of packages sent by the i-th source at the k-th congestion event.

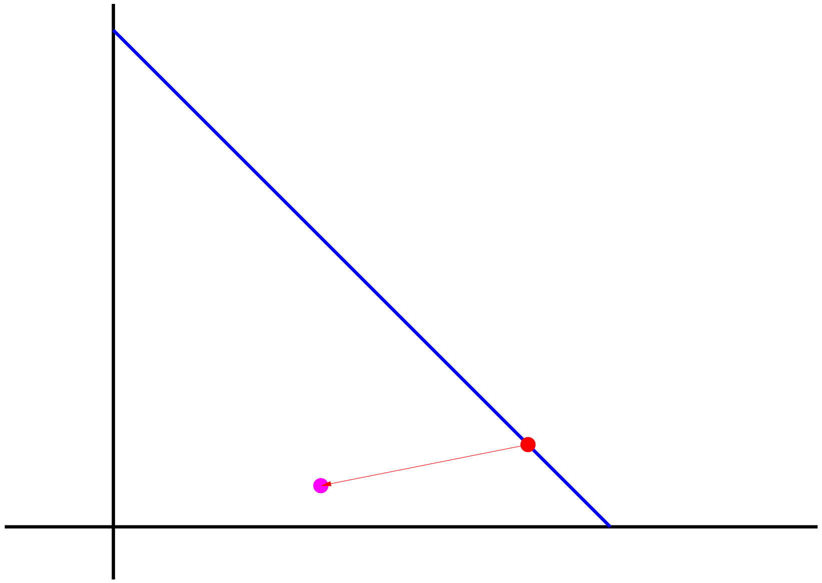
Time evolution for one source

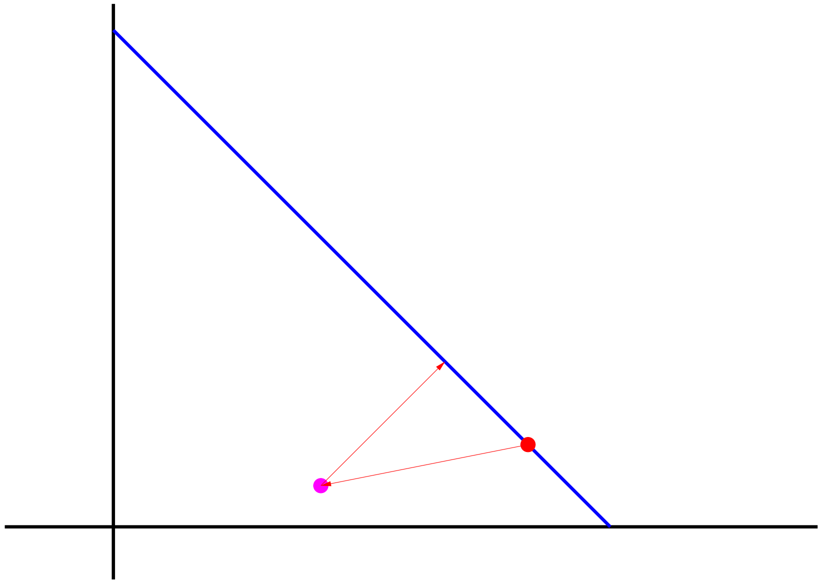


$w_i(k) \triangleq$ the number of packages sent by the i-th source at the k-th congestion event.

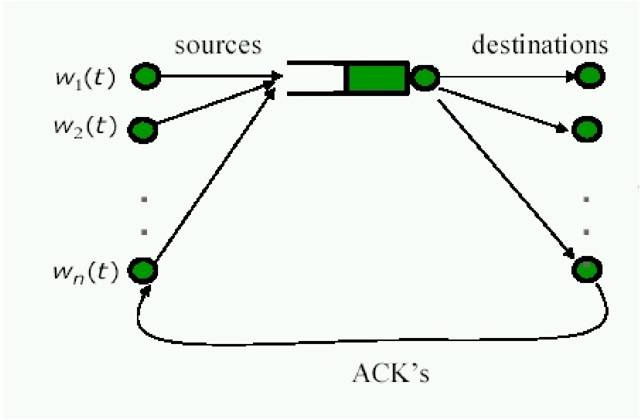








Many sources, one bottleneck

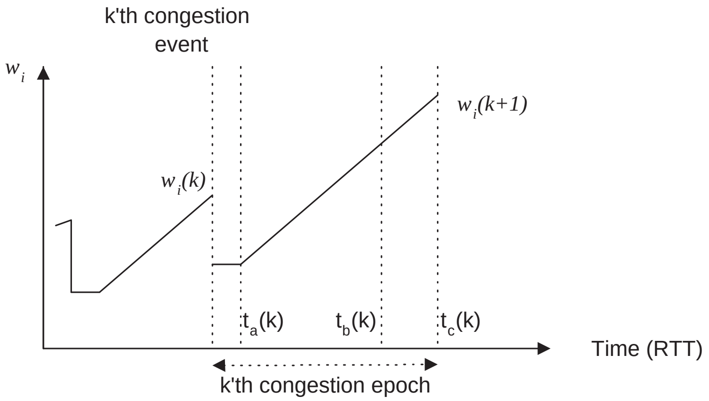


$w_i(k) \mathrel{\hat{=}}$ the number of packages sent by the i-th source at the k-th congestion event.

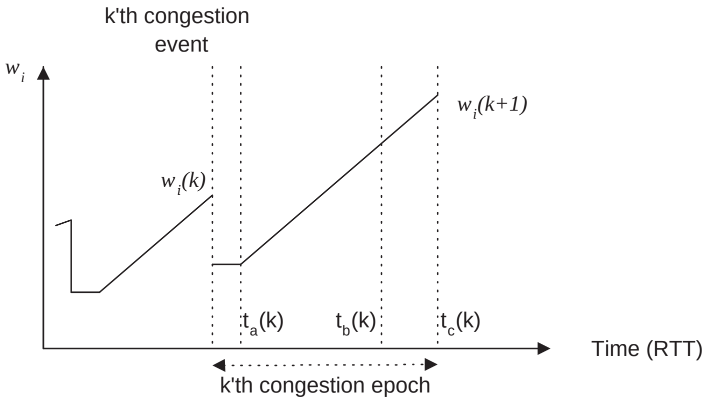
Assumptions I

- Congestion occurs in a single bottleneck.
- Congestion is noticed one RTT after it happens.
- Buffer size of bottleneck is small, i.e. RTT can be approximated by a constant.
- ACKs are not lost.
- RTT is the same for all sources.
- The network is synchronized.

Many sources, one bottleneck

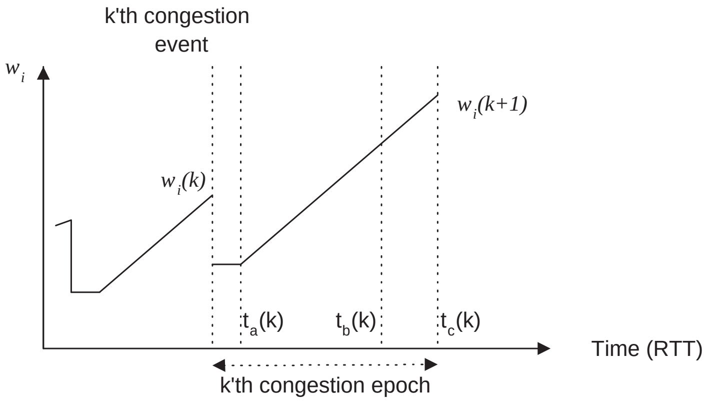


Many sources, one bottleneck



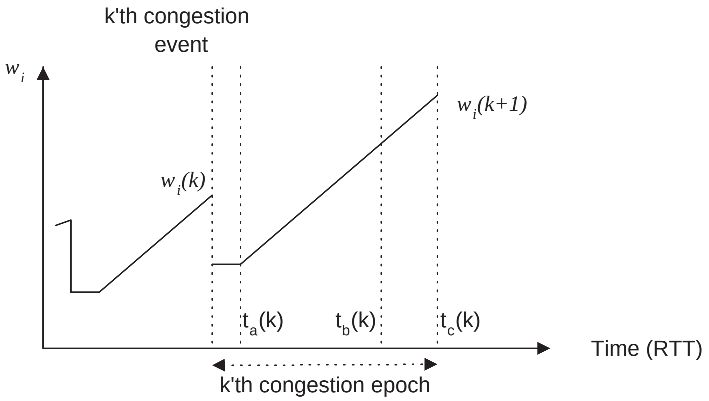
$$w_i(k+1) = \beta_i w_i(k) + \alpha_i(t_c(k) - t_a(k))$$

Many sources, one bottleneck



$$w_i(k+1) = \beta_i w_i(k) + \alpha_i(t_c(k) - t_a(k))$$
$$\sum \alpha_i(t_c(k) - t_a(k)) = P - \sum \beta_i w_i(k) + \sum \alpha_i$$

Many sources, one bottleneck



$$w_i(k+1) = \beta_i w_i(k) + \alpha_i(t_c(k) - t_a(k))$$
$$\sum \alpha_i(t_c(k) - t_a(k)) = P - \sum \beta_i w_i(k) + \sum \alpha_i$$
$$\sum w_i(k) = P + \sum \alpha_i$$