Series of nodes

The bounded variance network calculus

The bounded-variance network calculus

- The **bounded-variance network calculus** (giacomazzi-Saddemi 2008) is founded on two main assumptions that lead to an approximated calculation of delay, rather than true upper bounds
- The first approximation is that, in order to calculate the distribution of delay in a network node, the Choe’s and Shroff’s Maximum Variance Asymptotic (MVA) upper bound is applied
- Consequently, traffic is assumed to be Gaussian
- In fact, the MVA provides upper bounds for the probability that the delay in a network node exceeds a given threshold, given Gaussian input traffic flows
- Under specific conditions, the methods of the referred work can be applied also with non-Gaussian traffic flows as with a large number of flows the distribution of the compound traffic converges to a Gaussian under the broad hypotheses of the Central Limit Theorem
- With non-Gaussian traffic, the MVA provides approximations of delay, rather than upper bounds, with a tightness that decreases as the deviation of the distribution of traffic from a Gaussian grows

The bounded-variance network calculus

- The bounded-variance network calculus applies iteratively the MVA to the nodes composing the end-to-end path of a traffic flow
- At a given node, the inputs of the MVA are the two-moment descriptions (average value and variance) of the traffic flows feeding the node
- Given the statistical characterization of traffic flows at the input of a scheduler, the bounded-variance network calculus evaluates the distribution of traffic streams at the output of the scheduler
- In this way, it is possible to apply the MVA to the subsequent scheduler and to proceed iteratively
- As a consequence, the bounded-variance network calculus can be constructed as a sequence of simple and clear steps

The bounded-variance network calculus

- Given a sequence of nodes, numbered from 1 to *H*, crossed by a tagged traffic flow, the statistical description of the output traffic of node *h* is calculated, given its input traffic
- Since the input of node *h*+1 is the output of the *h*th node, it is possible to obtain a statistical description of the traffic offered to node *h*+1 as a function of the input traffic of node *h*
- By proceeding iteratively through the series of nodes constituting the end-to-end path of the tagged flow, the probability density of the delay at the *h*th node is calculated by exploiting the MVA
- The final step takes into account that, in a series of nodes, the end-to-end delay is equal to: $d_{e2e}=d_1+d_2+...+d_H$

The bounded-variance network calculus

The second assumption of the bounded-variance network calculus is that the random variables are assumed to be statistically independent, therefore, the density of the end-to-end delay is calculated as:
$f_{de2e}(t)=d_{d1}(t)+f_{d2}(t)+...+f_{dH}(t)$

- from which delay is straightforwardly computed
- While this assumption introduces an additional approximation in the framework, it simplifies radically calculations, as it is not necessary to calculate an end-to-end service curve, that is, the most complex task of classical network calculus is avoided
- In our framework, the delay distribution at each node is calculated with the service curve of the isolated node and the distortion of the covariance structure of traffic crossing nodes is accounted for with a novel method for the calculation of the variance of the cumulative traffic at the node’s output, given the characterization of input traffic flows

The bounded-variance network calculus

- While this assumption introduces an additional approximation in the framework, it simplifies radically calculations, as it is not necessary to calculate an end-to-end service curve, that is, the most complex task of classical network calculus is avoided
- In our framework, the delay distribution at each node is calculated with the service curve of the isolated node and the distortion of the covariance structure of traffic crossing nodes is accounted for with a novel method for the calculation of the variance of the cumulative traffic at the node’s output, given the characterization of input traffic flows
- The approximations provided by the bounded-variance network calculus, tested in sample scenarios widely used to evaluate the performance of stochastic network calculus methods, are surprisingly good and they improve dramatically the tightness of the current best bounds of the stochastic network calculus

The bounded-variance network calculus

- The bounded variance network calculus, is a framework for the extension of the two-moment analysis to end-to-end paths where traffic flows cross multiple schedulers
- The hardest problem to be faced by multi-node network analysis is the characterization of the output traffic of a scheduler, a very difficult but unavoidable task, as the traffic output by a scheduler is the input traffic of the next scheduler
- The bounded variance network calculus addresses this problem by providing an approximation of the variance of the scheduler’s output traffic, in order to be able to iterate the MVA equations
- This approximation is based on a novel inequality on the variance of the minimum of two bivariate Gaussian random variables
- This inequality (Giacomazzi-Bellini 2008) states that, given two bivariate Gaussian random variables *x* and *y*, the variance of *z*=min(*x*,*y*) is bounded as

$$\operatorname{var}(z) \le \max\left(\operatorname{var}(x), \operatorname{var}(y)\right)$$

The bounded-variance network calculus

- This inequality provides an upper bound of the variance of the output traffic of a node as a function of the node's input traffic
- In fact, let us consider a tagged flow crossing a sequence of *H* schedulers, where each scheduler provides differentiated services to multiple service classes, numbered from 1 to Q_h and, in the *h*th scheduler, the flow is served in service class q_h
- By denoting the input and output cumulative traffic for the tagged traffic flow at the *h*th scheduler of the end-to-end path as

$$X_{q_h}^{\text{in}}(t), X_{q_h}^{\text{out}}(t)$$

- respectively, and by referring to the service envelope of the tagged flow at the *h*th scheduler as

$$S_{q_h}(t)$$

- it can be easily shown that the cumulative output traffic of the tagged flow at the *h*th scheduler is given by:

$$X_{q_h}^{\text{out}}(t) = \min\left(X_{q_h}^{\text{in}}(t), S_{q_h}(t)\right)$$

The bounded-variance network calculus

- This relation is derived as follows. The scheduler cannot output a traffic greater than the offered traffic, it is easily recognized that

$$X_{q_h}^{\text{out}}\left(t\right) \le X_{q_h}^{\text{in}}\left(t\right)$$

- Moreover, for a strict priority scheduler:

$$S_{q_h}\left(t\right) = \max\left(0, Ct - \sum_{j=1}^{q_h-1} X_j^{\text{in}}\left(t\right)\right)$$

- that is, the available service

$$X_{\text{av,q}_h}^{\text{out}}\left(t\right)$$

- for the reference traffic flow is statistically lower-bounded by the total service capacity, minus the envelopes of the input traffic flows of service classes with better service priority

The bounded-variance network calculus

- Let us define the *available capacity*

$$\Xi_{\mathrm{av},\mathrm{q}_h}^{\mathrm{out}}\left(t\right)$$

- as the output line capacity available for the reference traffic flow
- The available capacity differs from the available service, as the latter represents an output traffic, while the former is the potential volume of traffic that the scheduler can output for the reference traffic flow
- It is easily understood that, for the strict priority scheduler:

$$\Xi_{\mathrm{av},\mathrm{q}_h}^{\mathrm{out}}\left(t\right)=\max\left(0,Ct-\sum_{j=1}^{q_h-1}X_j^{\mathrm{in}}\left(t\right)\right)$$

The bounded-variance network calculus

- In fact, in a static priority scheduler, the amount of service capacity available for the reference traffic flow does not depend on the backlog of traffic stored in the corresponding queue, as it depends solely on the volume of traffic offered by the service classes with a smaller priority index
- Therefore, the available capacity represents the available capacity for the reference traffic flow, independently on the volume of traffic that it offers to the scheduler, that is, the input traffic of class *i* may be equal to, greater than or lower than the minimally backlogged input traffic and, in all these cases, the capacity is given by

$$\Xi_{\mathrm{av,q}_h}^{\mathrm{out}}(t) = \max\left(0, Ct - \sum_{j=1}^{q_h-1} X_j^{\mathrm{in}}(t)\right)$$

- The scheduler cannot output, for the reference traffic flow, a traffic greater than the available capacity, therefore,

$$X_{\mathrm{q}_h}^{\mathrm{out}}(t) \leq \Xi_{\mathrm{av,q}_h}^{\mathrm{out}}(t)$$

The bounded-variance network calculus

- Thus,

$$X_{q_h}^{\text{out}}(t) \le S_{q_h}(t)$$

- as the scheduler cannot serve more traffic than the available service capacity
- In conclusion,

$$B_{q_h}^{\text{out}}(t) = \min\left(B_{q_h}^{\text{in}}(t), S_{q_h}(t)\right)$$

- is a statistical traffic envelope for the traffic output by the scheduler for service class *i*
- Therefore,

$$\operatorname{var}\left(X_{q_h}^{\text{out}}(t)\right) \le \operatorname{var}\left(B_{q_h}^{\text{out}}(t)\right) \le \max\left(\operatorname{var}\left(B_{q_h}^{\text{in}}(t)\right), \operatorname{var}\left(S_{q_h}(t)\right)\right)$$

The bounded-variance network calculus

- Given a sequence of *H* nodes crossed by a tagged traffic flow, we proceed iteratively starting from node 1 and, since

$$X_{q_{h+1}}^{\text{in}}(t) \approx B_{q_h}^{\text{out}}(t)$$

- we can calculate a statistical characterization of the input traffic of each node through
- Therefore, we can calculate with the MVA the probability density of the delay d_h at the *h*th node

Finally, since in a series of *H* nodes the end-to-end delay is equal to $d_{e2e}=d_1+d_2+...+d_H$

we calculate the density of the end-to-end delay as $f_{de2e}(t)=d_{d1}(t)+f_{d2}(t)+...+f_{dH}(t)$

The bounded-variance network calculus

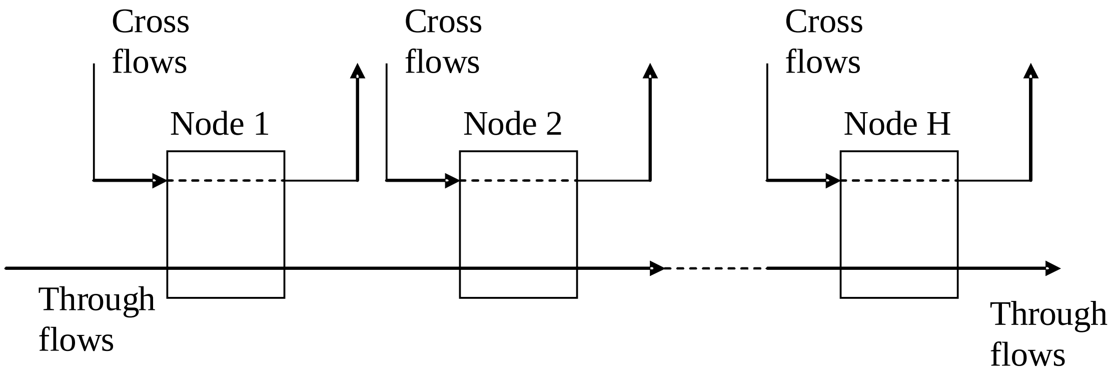
- Finally, the probability of violating an end-to-end delay threshold is

$$\Pr\left(d_{e2e} > d\right) = \int_{d}^{\infty} f_{d_{e2e}}\left(\tau\right) d\tau.$$

- Similar considerations can be applied to other types of schedulers

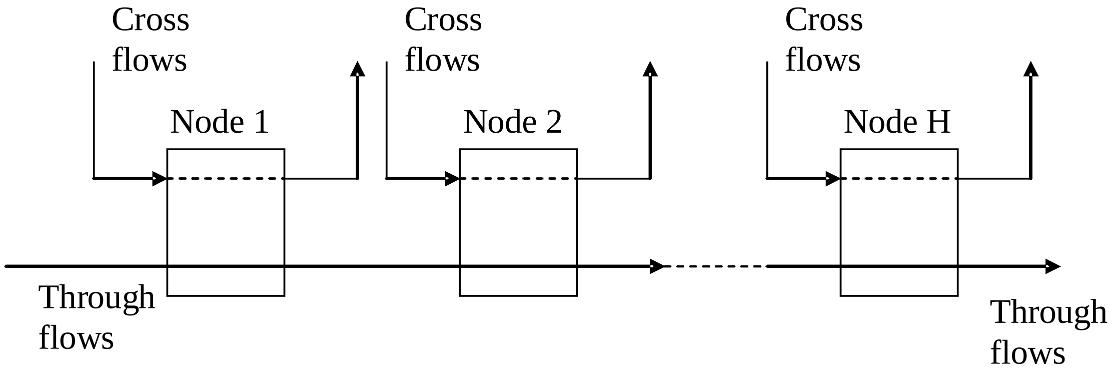
An example of application

- We consider a sequence of *H* strict-priority nodes
- We refer to the flows which traverse the network as *through* flows, and to the flows which transit the network as *cross* flows
- We have a series of *H* nodes, and each node receives and forwards to the next node in the chain the through traffic flows
- Each cross flow traverses one node and then it exits the network
- We suppose that each node acts as a Static Priority scheduler with two traffic classes
- The cross and through traffic flows are served with the high and low priority, respectively
- Therefore, cross flows belong to service class 1 and through flows are served in class 2



An example of application

- We specialize our calculations for a commonly adopted scenario where both through and cross sources are modeled as the Markov-Modulated On-Off sources
- The total number of through flows is equal to N_2 and the number of cross flows that traverse each node is equal to N_1
- The parameters of the Markov model of cross and through flows are (λ_1, μ_1, P_1) and (λ_2, μ_2, P_2), respectively. The output capacity of each node is equal to C and it is shared among N_1+N_2 flows

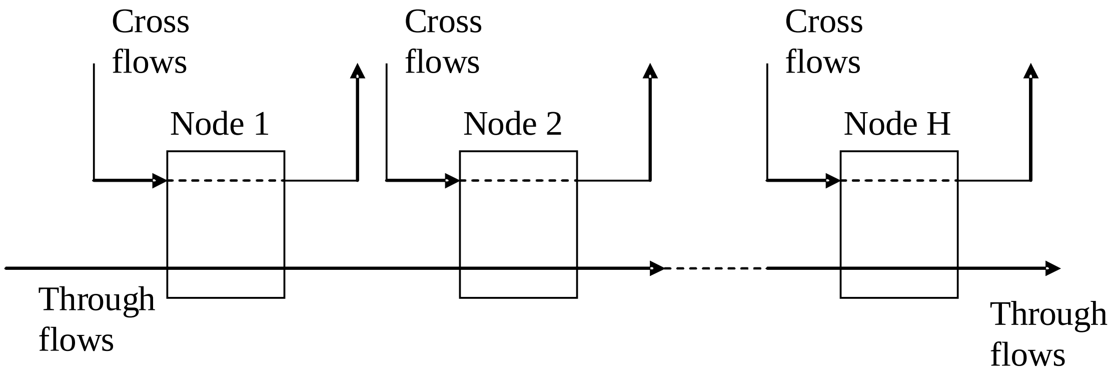


An example of application

- We have the objective of calculating a bound for the end-to-end delay of through flows and we need the service envelope of through flows at the generic node *h*
- We will obtain the result by applying the bounded variance network calculus
- The traffic envelope of the aggregate input cross flows at the *h*-th scheduler is referred to as and it evaluates to

$$E\left(B_{1,h}^{in}(t)\right) = N_1 r_1 t$$

$$\mathrm{var}\left(B_{1,h}^{in}(t)\right) \leq N_1 r_1 b_1 t$$



An example of application

- The service envelope of the through flows at the *h*th scheduler, referred to as $S_{2,h}(t)$, is

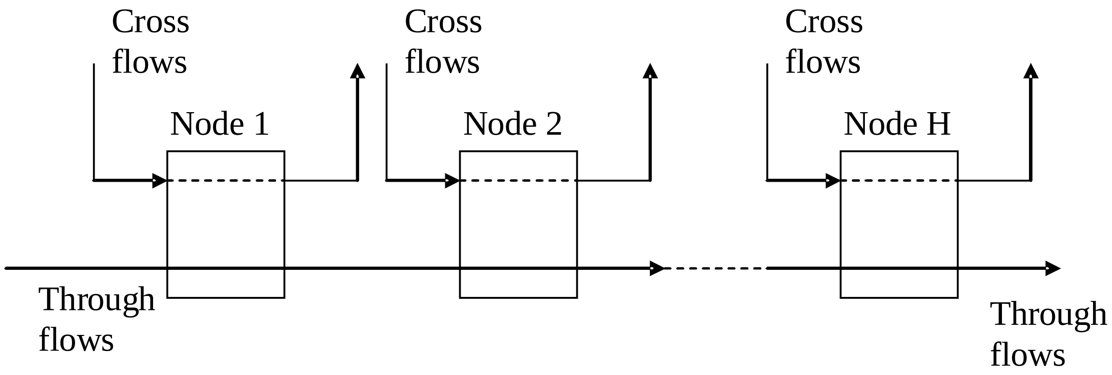
$$E\left(S_{2,h}(t)\right) \approx Ct - N_1 r_1 t$$

$$\operatorname{var}\left(S_{2,h}(t)\right) \approx N_1 r_1 b_1 t$$

- The fresh through flows at the first node have the traffic envelope

$$E\left(B_{2,1}^{in}(t)\right) = N_2 r_2 t$$

$$\operatorname{var}\left(B_{2,1}^{in}(t)\right) \le N_2 r_2 b_2 t$$



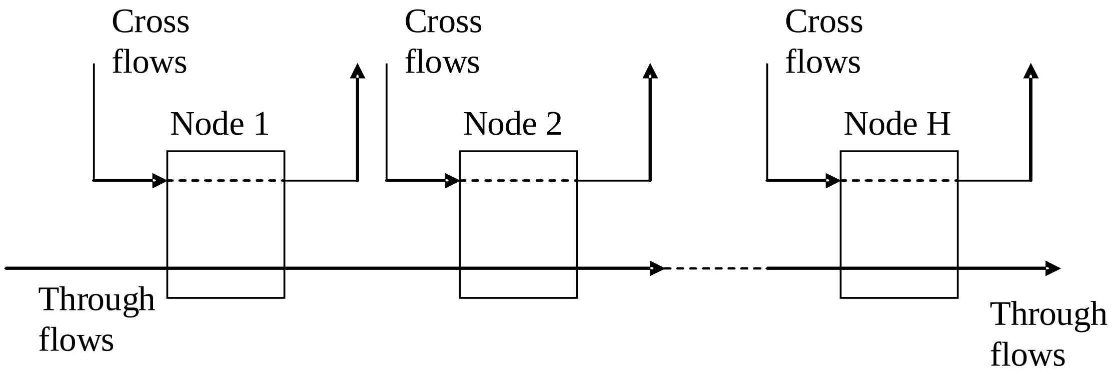
An example of application

- we proceed with the following conservative approximation

$$E\left(B^{in}_{2,1}(t)\right)=N_2 r_2 t$$

$$\operatorname{var}\left(B^{in}_{2,1}(t)\right)\le \max\left(N_2 r_2 b_2 t, N_1 r_1 b_1 t\right)$$

- that will allow us to derive very simple closed-form expressions of the end-to-end delay



An example of application

- We obtain the variance of the traffic envelope of through flows at the input of node 2:

$$\operatorname{var}\left(B_{2,2}^{in}(t)\right)=\operatorname{var}\left(B_{2,1}^{out}(t)\right)\le\max\left(\max\left(N_2r_2b_2t,N_1r_1b_1t\right),N_1r_1b_1t\right)=\max\left(N_2r_2b_2t,N_1r_1b_1t\right).$$

- By iterating this procedure for the downstream nodes, we easily observe that we obtain the same bound for the variance of the through traffic at the input of all the Static Priority schedulers:

$$\forall h\in[1,H]:\operatorname{var}\left(B_{2,h}^{in}(t)\right)\le\max\left(N_2r_2b_2t,N_1r_1b_1t\right).$$

An example of application

- Therefore, we can calculate the delay bound violation probability at the *h*th scheduler:

$$\forall h \in [1, H] : \Pr\left(D_2 > d_2\right) \leq \exp\left(-2\frac{C - A_2}{B_2^2}\left(\left(C - A_1\right)B_2 - \left(C - A_2\right)B_1\right)d_2\right)$$

- where

$$A_1 = N_1 r_1 \qquad A_2 = N_1 r_1 + N_2 r_2$$

$$B_1 = N_1 r_1 b_1 \quad B_2 = N_1 r_1 b_1 + \max\left(N_2 r_2 b_2, N_1 r_1 b_1\right)$$

An example of application

- To simplify the notation we write

$$P\left(D_2 > d_2 \right) \le \exp\left(- \kappa_2\, d_2 \right)$$

- The probability density of delay for the through (class 2) flows at node *h* is equal to

$$f_{d_{i,h}}\left(t \right) = \kappa_2 \exp\left(- \kappa_2\, t \right)$$

- and we find the density of the end-to-end delay as:

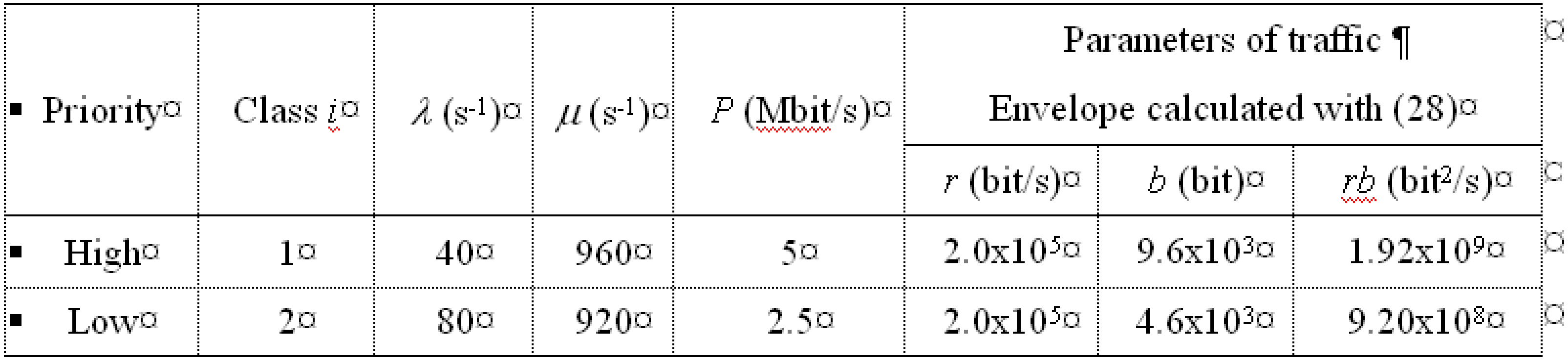
$$f_{d_{2,e2e}}\left(t \right) = \frac{\kappa_2^H t^{H-1}}{\left(H - 1 \right)!} \exp\left(- \kappa_2\, t \right) .$$

- Finally,

$$\Pr\left(d_{i,e2e} > t \right) = \int_t^{\infty} f_{d_{i,e2e}}\left(\tau \right) d\tau = e^{-\kappa_i t} \sum_{j=0}^{H-1} \frac{\left(\kappa_i t \right)^j}{j!}.$$

An example of application

- We use the following numerical parameters:

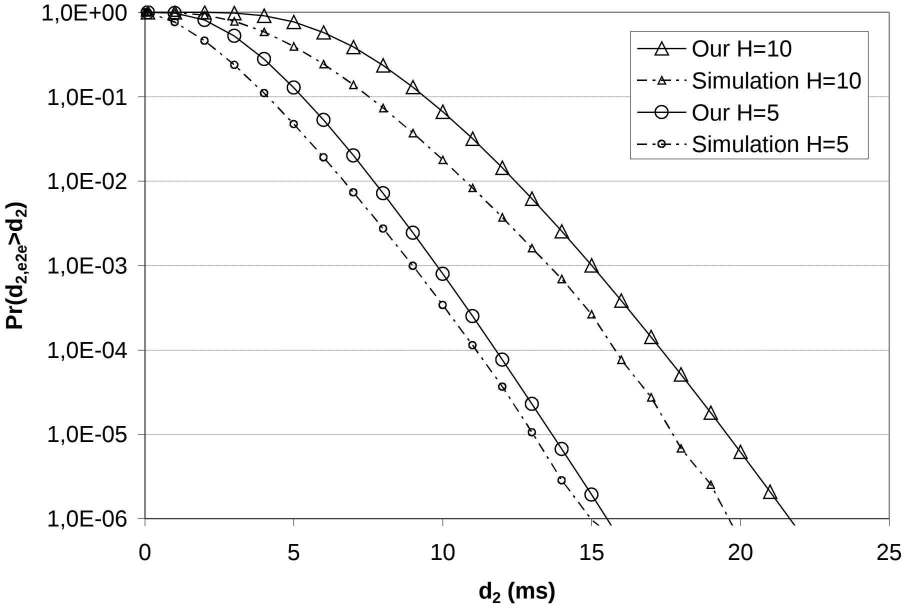


Priority	Class i	λ (s^{-1})	μ (s^{-1})	P (Mbit/s)	Parameters of traffic ¶ Envelope calculated with (28)		
					r (bit/s)	b (bit)	rb (bit^2/s)
High	1	40	960	5	$2.0x10^5$	$9.6x10^3$	$1.92x10^9$
Low	2	80	920	2.5	$2.0x10^5$	$4.6x10^3$	$9.20x10^8$

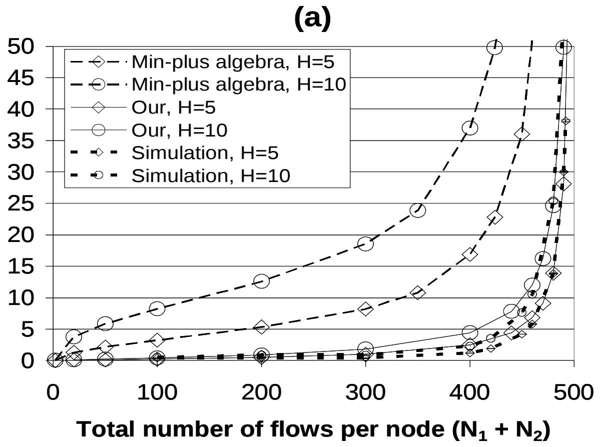
- And the capacity of the output links of all schedulers is equal to 100 Mbit/s

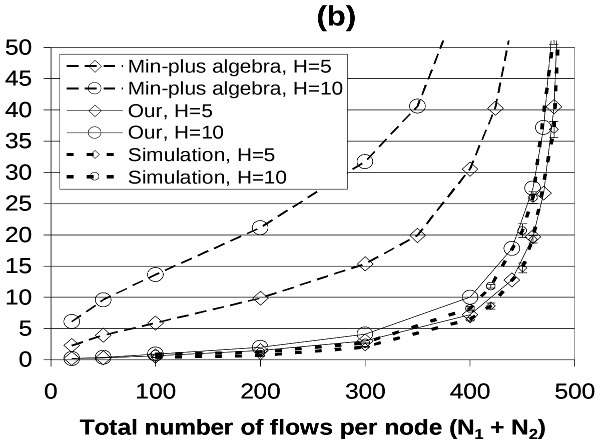
An example of application

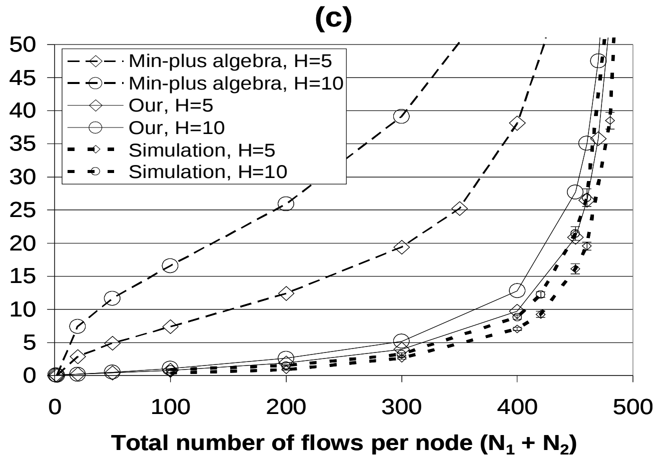
- The figure shows the quality of the analytical approximation for different values of the end-to-end delay threshold and different number of hops
- The approximation is rather good



An example of application



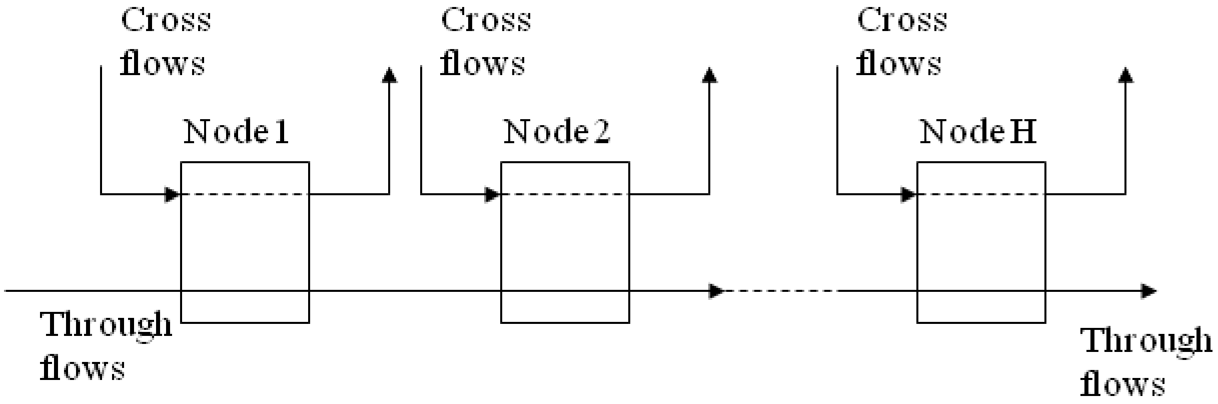




- End-to-end delay bound exceeded with probability 10-1 (**a**), 10-6 (**b**), and 10-9 (**c**) as a function of the total number of flows per node (*N*1+*N*2) with *H*=5 and *H*=10. For each value of *H* the Figure shows results obtained with (a) min-plus algebra, (b) our closed-form analysis, and (c) simulation.

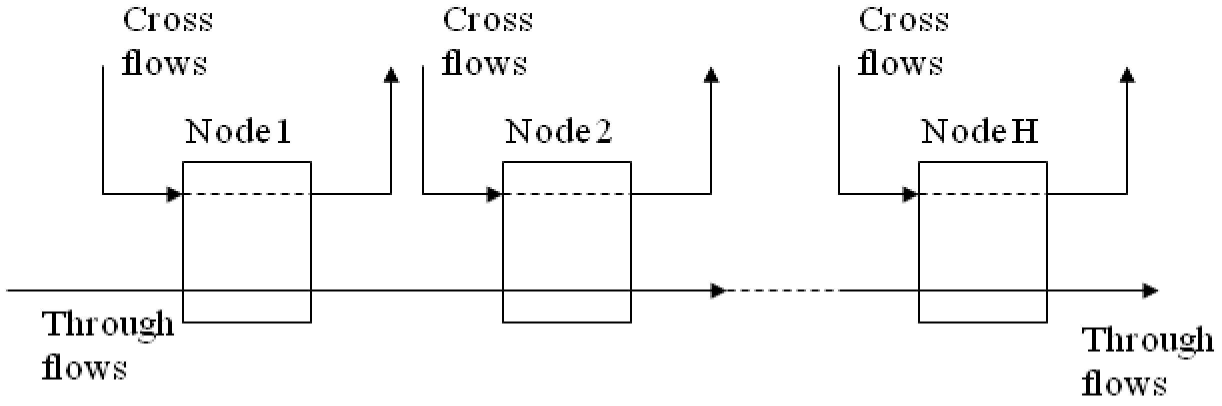
An example (1)

- For example, let us consider a scenario where an aggregate flow traverses the network (through flows), and at each node another aggregated flow crosses the network
- We have a series of *H* nodes, and each node receives and forwards to the next node in the chain the through traffic flows
- Each cross flow traverses one node and then it exits the network



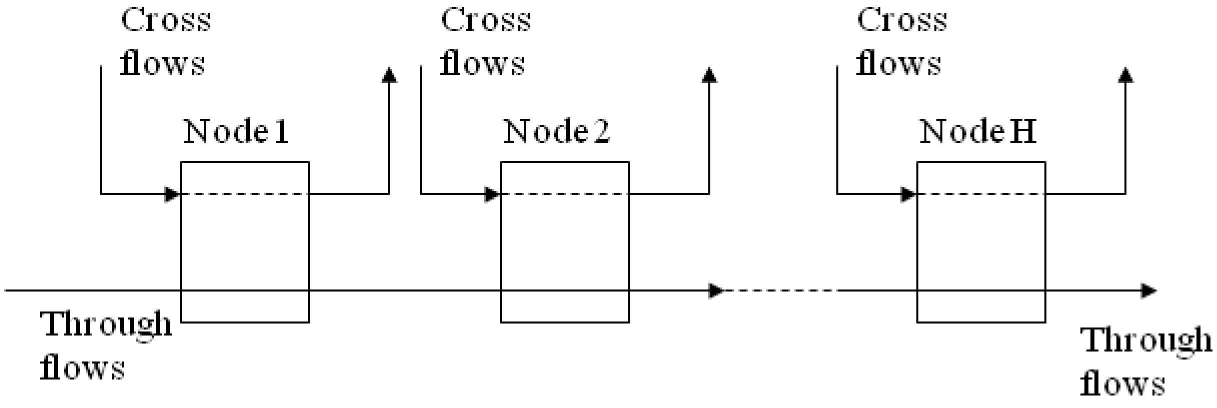
An example (2)

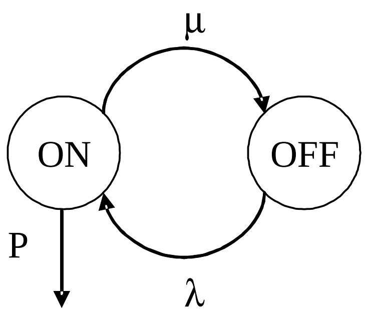
- The total number of through flows is equal *N* and also the number of cross flows that traverse one node is equal to *N*
- Therefore, each node treats 2*N* flows, *N* through and *N* cross flows The output capacity of each node is equal to *C* and it is shared among 2*N* flows
- Each node acts as a simple FIFO (First In First Out) scheduler



An example (3)

- For the sake of simplicity, we consider the case of traffic micro-flows modeled as two-state continuous Markov chains; an aggregate is composed by *N* micro-flows
- The transition rate from the OFF state to the ON state is λ [s^{-1}], while the rate of the inverse transitions is equal to λ [s^{-1}]
- In the ON state, a micro flow transmits at a constant rate of *P* [bit/s]
- No traffic is generated in the OFF state
- These sources could model variable bit-rate G.726 VoIP codecs or UDP data sources

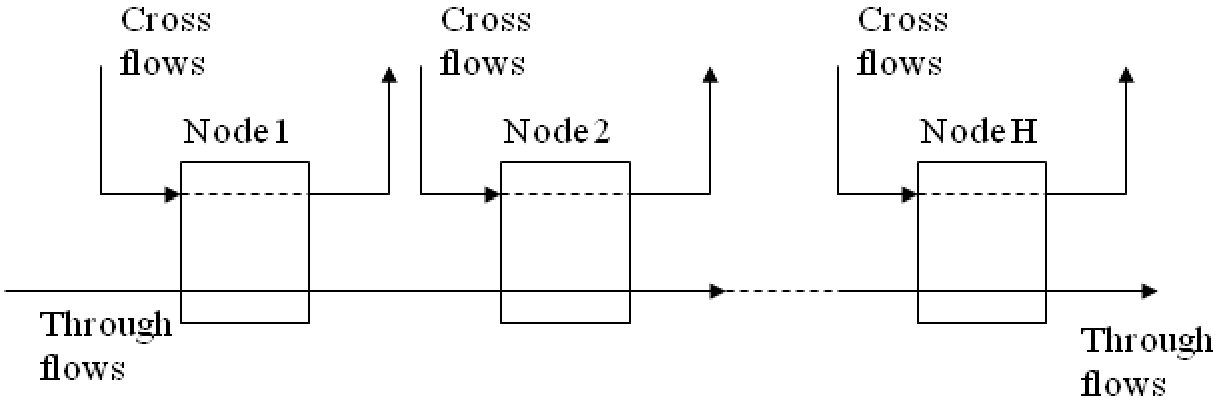




An example (4)

- With the bounded variance network calculus, we calculate the probability density of delay of through traffic at a generic node as

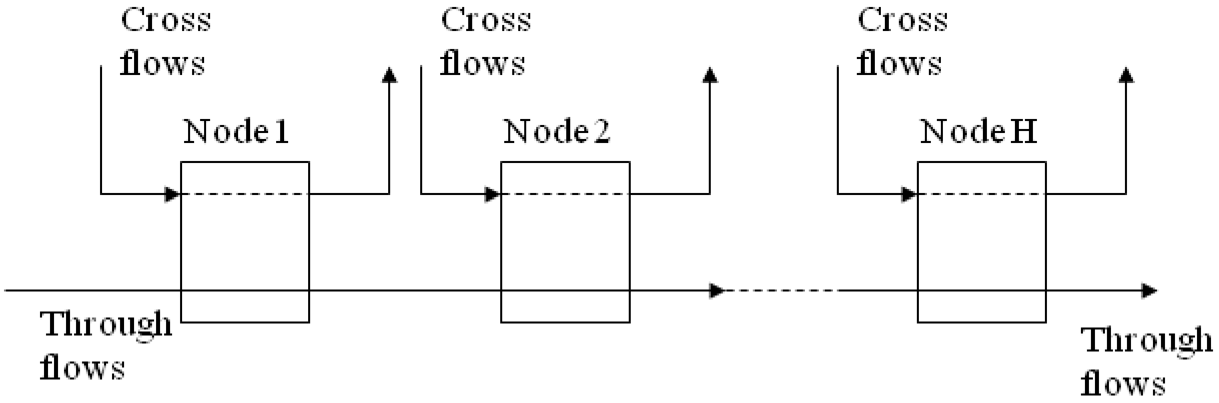
$$P\left(d_h > t\right) \le \exp\left(- \frac{2C\left(C - 2N \frac{\lambda}{\left(\lambda+\mu\right)} P\right)}{4N \frac{\lambda\mu}{\left(\lambda+\mu\right)^3} P^2} t \right) = e^{-\kappa t}$$



An example (5)

- The probability density of end-to-end delay is equal to

$$f_{d_{e2e}}\left(t\right)=\frac{\kappa^{H}t^{H-1}}{\left(H-1\right)!}\exp\left(-\kappa t\right)$$

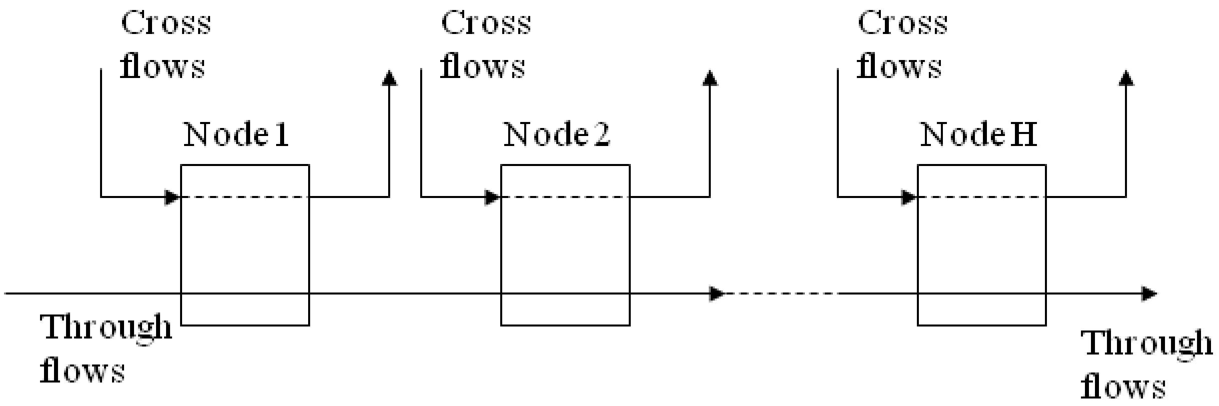


An example (6)

- The probability of exceeding the end-to-end delay bound *d* is equal to

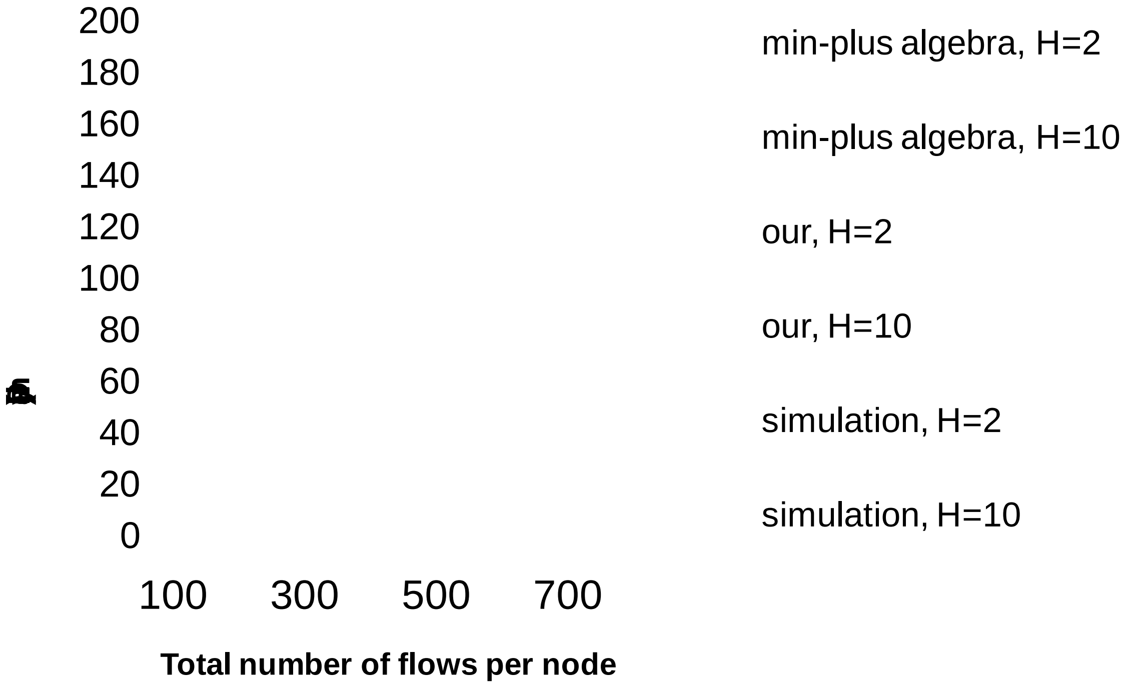
$$\Pr\left(d_{e2e} > t\right) = e^{-\kappa t}\sum_{j=0}^{H-1}\frac{\left(\kappa t\right)^{j}}{j!}$$

- Note that the solution is analytical

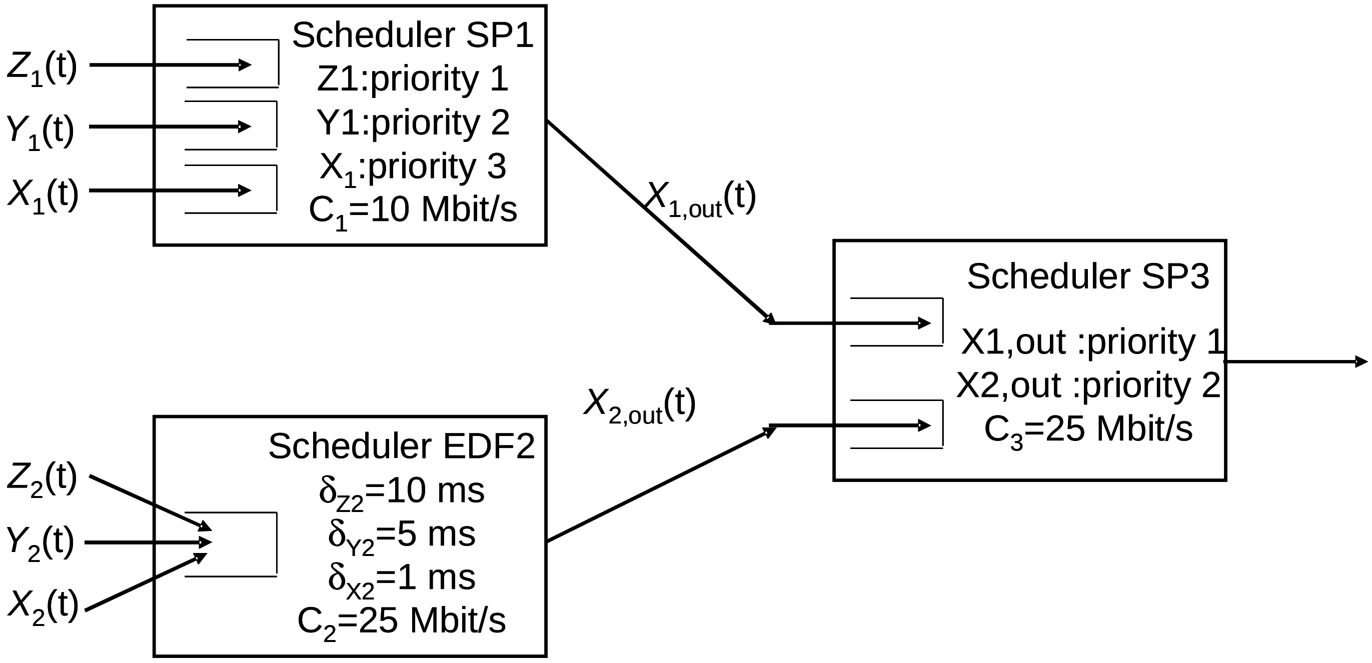


An example (7)

- The graph compares, for the selected scenario, the delay bounds of the min-plus algebra, the approximated delay of the bounded variance network calculus and the real performance of the network measured by computer simulation
- The difference is striking: For example, with *H* =10 and 300 flows, the min-plus algebra provides a bound of around 160 ms, while the delay bound of the bounded-variance network calculus is equal to 3.58 ms, the real delay is of 3.4 ms



A multinode case



Scheduler SP1

Z1:priority 1

Y1:priority 2

X_1:priority 3

C_1=10 Mbit/s

Scheduler EDF2

δ_{Z2}=10 ms

δ_{Y2}=5 ms

δ_{X2}=1 ms

C_2=25 Mbit/s

Scheduler SP3

X1,out :priority 1

X2,out :priority 2

C_3=25 Mbit/s

TABELLA PARAMETRI

r_X1 (Mbit/s)	3
b_X1 (kbit)	100
r_X2 (Mbit/s)	4
b_X2 (kbit)	12,5
r_Z1 (Mbit/s)	2
b_Z1 (kbit)	150
r_Z2 (Mbit/s)	1
b_Z2 (kbit)	50
m_Y1 (Mbit/s)	3,2
a_Y1 (Mbit s)	5
H_Y1	0,99
m_Y2 (Mbit/s)	2,5
a_Y2 (Mbit s)	5
H_Y2	0,85
C1 (Mbit/s)	10
C2 (Mbit/s)	25
C3 (Mbit/s)	25

Statement of the problem

- In the figure, X1(t) has a linear variance envelope with parameters r_X1=3Mbit/s and b_X1=100 kbit
- Y1(t) is fGt with parameters m_Y1 = 3.2 Mbit/s, a_Y1 = 5 Mbit . s, H_Y1=0.99
- Z1(t) has a linear variance envelope with parameters r_Z1=2Mbit/s, b_Z1=150 kbit
- Scheduler 1 is SP; flows X1(t), Y1(t) and Z1(t) are served with priority 3, 2 and 1, respectively
- The capacity of the scheduler output line is 10 Mbit/s
- X1,out(t) is the output of X1(t)
- X2(t) has a linear variance envelope with parameters r_X2= 4 Mbit/s, b_X2= 12.5 kbit
- Y2(t) is fGt with parameters m_Y2 = 2.5 Mbit/s, a_Y2 = 5 Mbit . s, H_Y2 = 0.85
- Z2(t) has a linear variance envelope with parameters r_Z2= 1 Mbit/s, b_Z2= 50 kbit
- Scheduler 2 is EDF, X2(t) has service deadline δX2=1 ms, Y2(t) has service deadline deadline δY2=5 ms and Z2(t) has service deadline δZ2=10 ms
- The capacity of the scheduler output line is 25 Mbit/s
- ìX2,out(t) is the output of X2(t)
- Scheduler 3 is SP where X1,out(t) is served with priority 1 and X2,out(t) with priority 2
- The capacity of the scheduler output line is 25 Mbit/s
- **Calculate the probability that the delay of flow X2,out(t) exceeds 90 ms in scheduler 3**

Solution (scheduler 1)

- Let us consider scheduler 1; the input flows have average value and variance given by

$$\begin{cases} E\left(Z_1(t)\right) = r_{Z_1}t = 2\times10^6 t \text{ bit/s} \\ \operatorname{var}\left(Z_1(t)\right) = r_{Z_1}b_{Z_1}t = 2\times10^6\times0.15\times10^6 t \ = 0.3\times10^6 t \text{ bit}^2\text{/s} \end{cases}$$

$$\begin{cases} E\left(Y_1(t)\right) = m_{Y_1}t = 3.2\times10^6 t \text{ bit/s} \\ \operatorname{var}\left(Y_1(t)\right) = m_{Y_1}a_{Y_1}t^{2H_{Y1}} = 3.2\times10^6\times5\times10^6 t^{1.98} = 16\times10^{12} t^{1.98} \text{ bit}^2\text{/s} \end{cases}$$

$$\begin{cases} E\left(X_1(t)\right) = r_{X_1}t = 3\times10^6 t \text{ bit/s} \\ \operatorname{var}\left(X_1(t)\right) = r_{X_1}b_{X_1}t = 0.3\times10^{12} t \text{ bit}^2\text{/s} \end{cases}$$

Solution (scheduler 1)

- The service envelope for X1(t) is

$$S_{X_1}(t) = Ct - Y_1(t) - Z_1(t)$$

$$E\left(S_{X_1}(t)\right) = Ct - E\left(Y_1(t)\right) - E\left(Z_1(t)\right) = 10\times10^6 t - 3.2\times10^6 t - 2\times10^6 t = 4.8\times10^6 t \text{ bit/s}$$

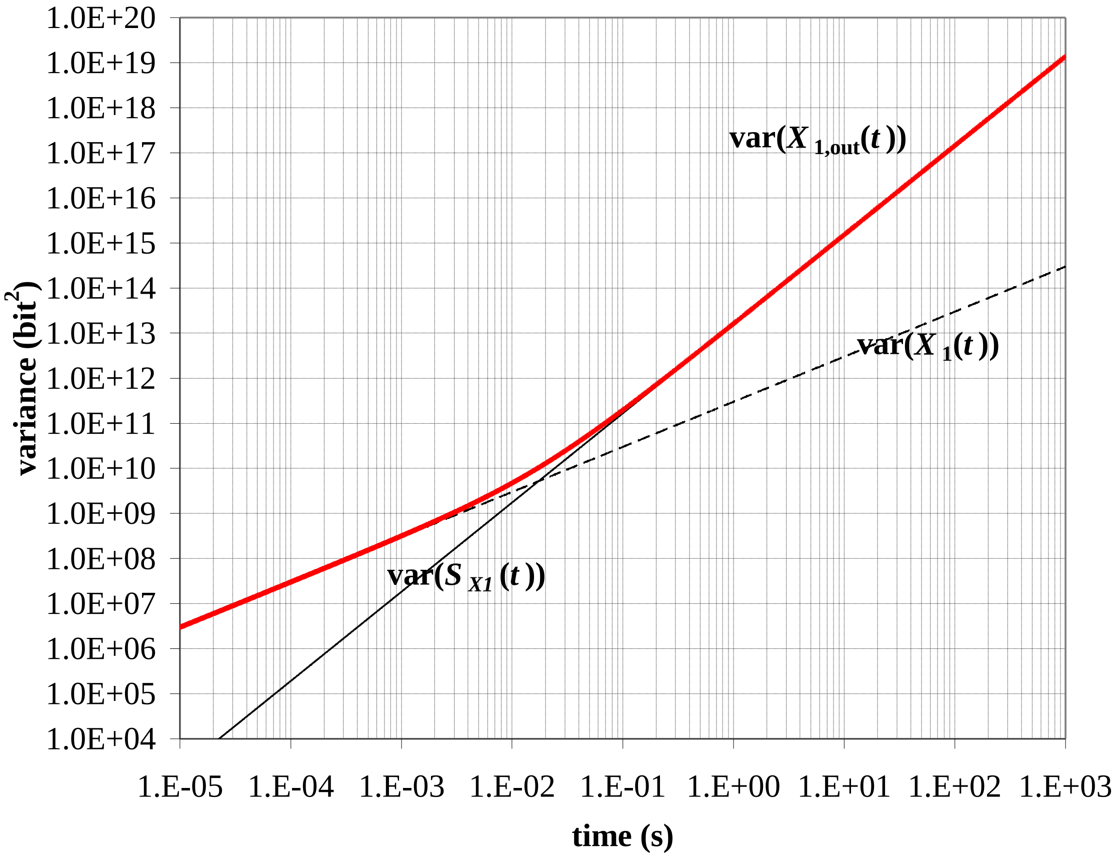
$$\mathrm{var}\left(S_{X_1}(t)\right) = \mathrm{var}\left(Y_1(t)\right) + \mathrm{var}\left(Z_1(t)\right) = 0.3\times10^6 t + 16\times10^{12} t^{1.98} \mathrm{bit}^2/\mathrm{s}$$

- The average value and variance of X1out(t) are

$$E\left(X_{1,\mathrm{out}}(t)\right) = E\left(X_1(t)\right)$$

$$\mathrm{var}\left(X_{1,\mathrm{out}}(t)\right) = \max\left(\mathrm{var}\left(X_1(t)\right), \mathrm{var}\left(S_{X_1}(t)\right)\right)$$

Solution (scheduler 1)



- The figure shows the variance of X1out(t)

Solution (scheduler 2)

- In scheduler 2, the inputs are

$$\begin{cases} E\left(Z_2\left(t\right)\right) = r_{Z_2}t = 1\times10^6 t \text{ bit/s} \\ \operatorname{var}\left(Z_2\left(t\right)\right) = r_{Z_2}b_{Z_2}t = 1\times10^6\times0.05\times10^6 t \ = 0.05\times10^6 t \text{ bit}^2\text{/s} \end{cases}$$

$$\begin{cases} E\left(Y_2\left(t\right)\right) = m_{Y_2}t = 2.5\times10^6 t \text{ bit/s} \\ \operatorname{var}\left(Y_2\left(t\right)\right) = m_{Y_2}a_{Y_2}t^{2H_{Y2}} = 2.5\times10^6\times5\times10^6 t^{1.7} = 12.5\times10^{12} t^{1.7} \text{ bit}^2\text{/s} \end{cases}$$

$$\begin{cases} E\left(X_2\left(t\right)\right) = r_{X_2}t = 4\times10^6 t \text{ bit/s} \\ \operatorname{var}\left(X_2\left(t\right)\right) = r_{X_2}b_{X_2}t = 4\times10^6\times0.0125\times10^6 t = 0.05\times10^{12} t \text{ bit}^2\text{/s} \end{cases}$$

Solution (scheduler 2)

- The service envelope for X2(t) is

$$E\left(S_{X_2}(t)\right) = Ct - m_{Y_2}\left(t - \max\left(\delta_{Y_2} - \delta_{X_2}\right)\right) - r_{Z_2}\left(t - \max\left(\delta_{Z_2} - \delta_{X_2}\right)\right)$$

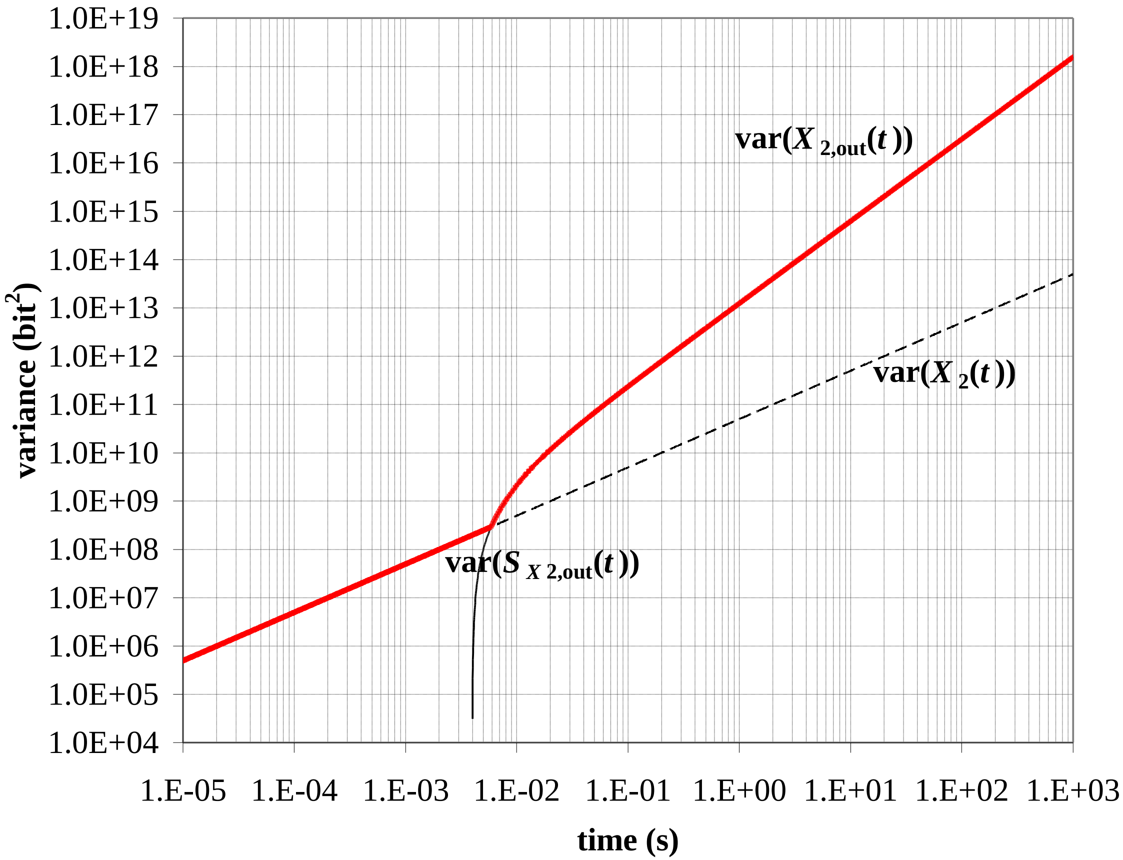
$$\operatorname{var}\left(S_{X_2}(t)\right) = m_{Y_2} a_{Y_2}\left(t - \max\left(\delta_{Y_2} - \delta_{X_2}\right)\right)^{2H_{Y2}} + r_{Z_2} b_{Z_2}\left(t - \max\left(\delta_{Z_2} - \delta_{X_2}\right)\right)$$

- The average value and variance of X1out(t) are

$$E\left(X_{2,\mathrm{out}}(t)\right) = E\left(X_2(t)\right)$$

$$\operatorname{var}\left(X_{2,\mathrm{out}}(t)\right) = \max\left(\operatorname{var}\left(X_2(t)\right), \operatorname{var}\left(S_{X_2}(t)\right)\right)$$

Solution (scheduler 2)



- The figure shows the variance of X2out(t)

Solution (scheduler 3)

- In scheduler 3, X2out(t) has service envelope with the following features

$$S_{X_{2,out}}(t) = Ct - X_{1,out}(t)$$

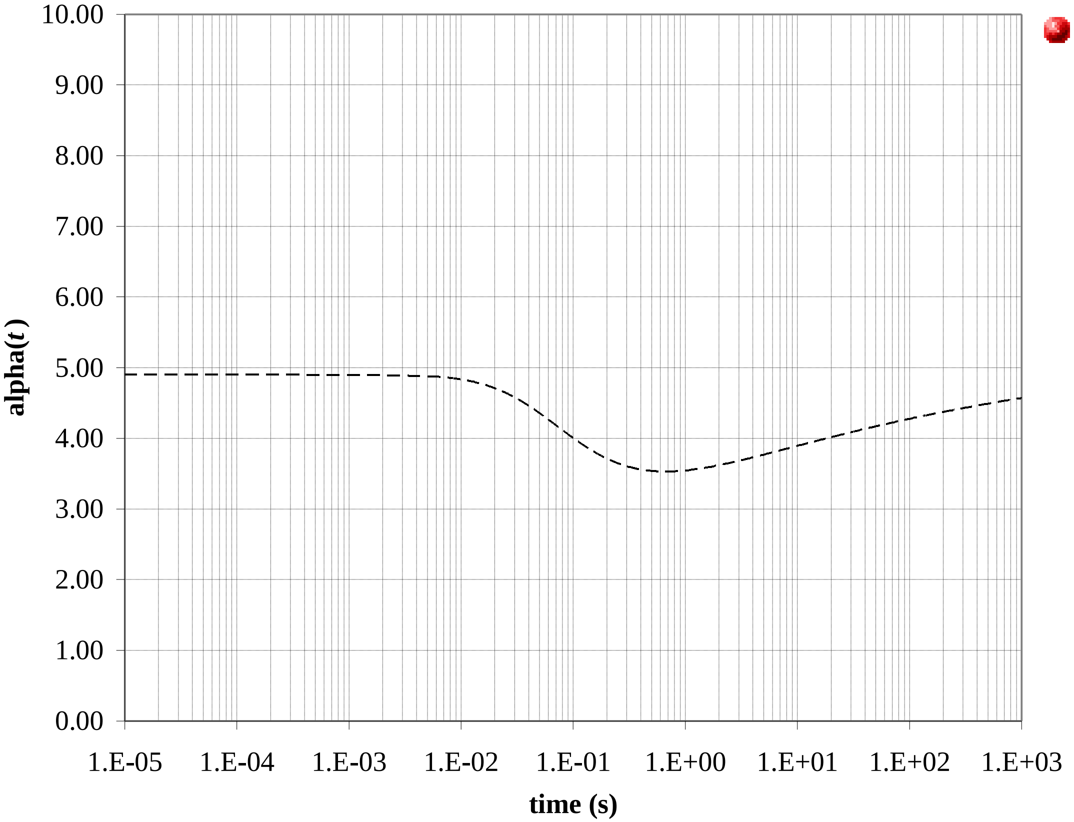
$$E\left(S_{X_{2,out}}(t)\right) = Ct - E\left(X_{1,out}(t)\right) = Ct - E\left(X_1(t)\right)$$

$$\mathrm{var}\left(S_{X_{2,out}}(t)\right) = \mathrm{var}\left(X_{1,out}(t)\right)$$

- The $\alpha(t)$ function is then given by

$$\alpha_{X\,2,out(t)} = -\frac{E\left(X_{2,out}(t)\right) - E\left(S_{X_{2,out}}(t+d)\right)}{\sqrt{\mathrm{var}\left(X_{2,out}(t)\right) + \mathrm{var}\left(S_{X_{2,out}}(t+d)\right)}}$$

Solution (scheduler 3)

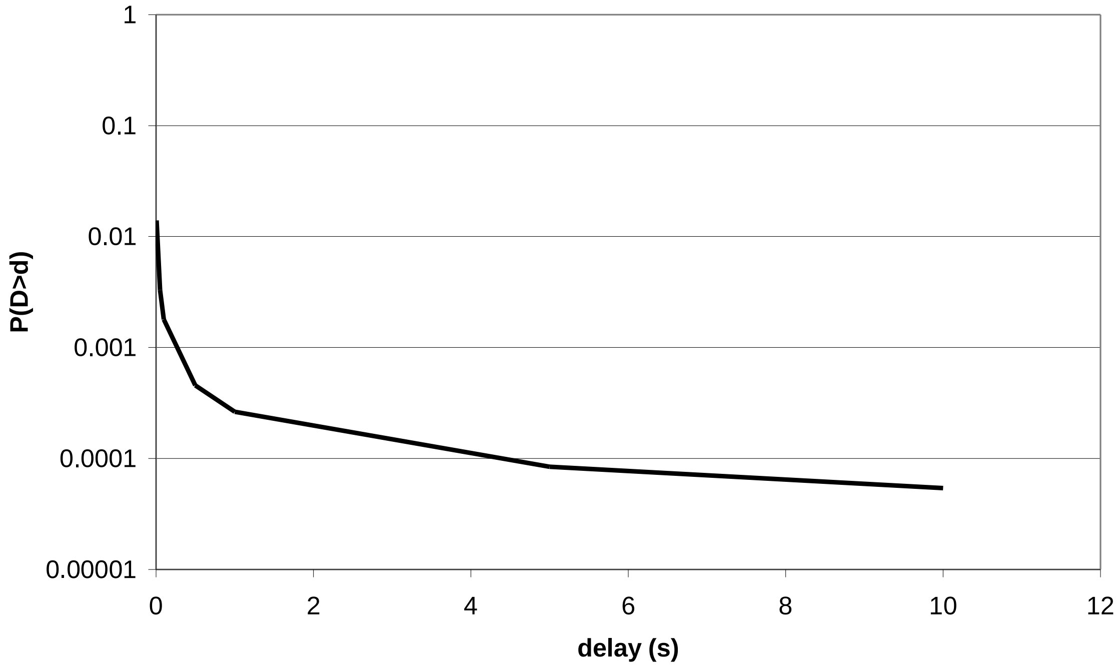


- The minimum value of alpha is 3.53 thus the probability of exceeding the delay threshold is equal to 0.0019

- The figure plots the alpha function

Further considerations

- The procedure can be repeated with different delay thresholds, as shown in the figure
- Remarkably, the delay distribution of X2out in scheduler 3 is fat-tailed
- This is due to the interference form LRD flows

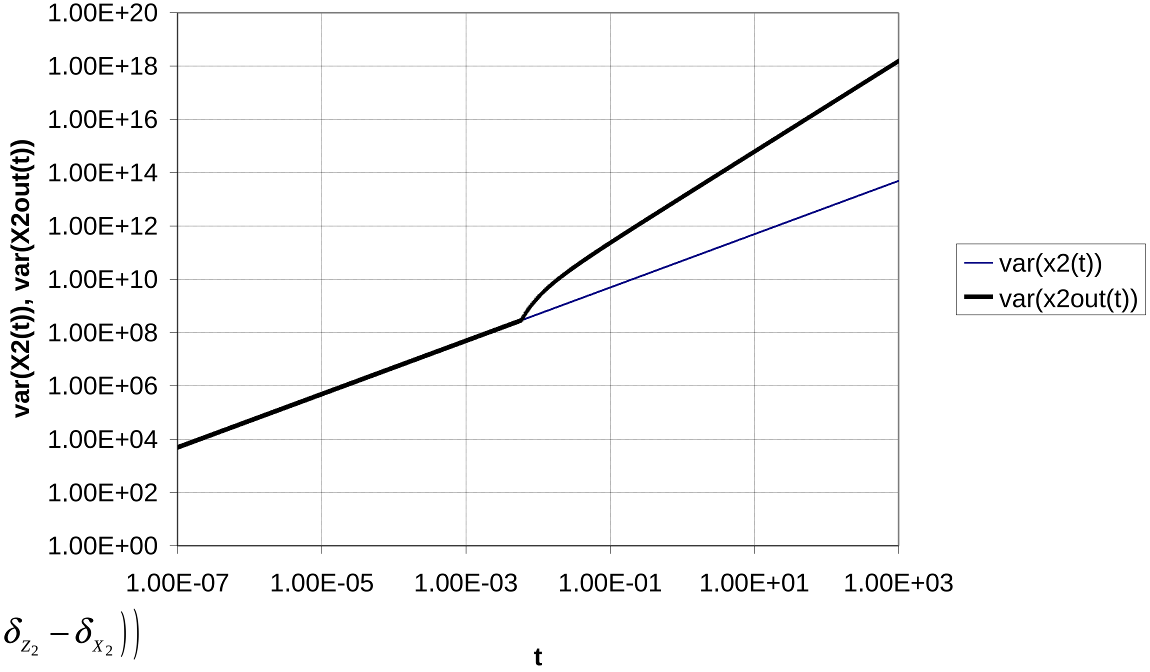


On the variance of X2out(t)

- It is interesting to note that the variance of X2out(t) differs significantly from that of the fresh flow X2(t)
- We recall that

$$\begin{cases} E\left(X_2(t)\right) = 4\times 10^6 t \text{ bit} \\ \operatorname{var}\left(X_2(t)\right) = 0.05\times 10^{12} t \text{ bit}^2 \end{cases}$$

- Then, for $S_{X2}(t)$ we have



$$\begin{cases} E\left(S_{X_2}(t)\right) = Ct - m_{Y_2}\left(t - \max\left(0, \delta_{Y_2} - \delta_{X_2}\right)\right) - r_{Z_2}\left(t - \max\left(0, \delta_{Z_2} - \delta_{X_2}\right)\right) \\ \operatorname{var}\left(S_{X_2}(t)\right) = m_{Y_2} a_{Y_2}\left(t - \max\left(0, \delta_{Y_2} - \delta_{X_2}\right)\right)^{2H_{Y2}} + r_{Z_2} b_{Z_2}\left(t - \max\left(0, \delta_{Z_2} - \delta_{X_2}\right)\right) \end{cases}$$

$$\begin{cases} E\left(S_{X_2}(t)\right) = 25\times 10^6 t - 25\times 10^6\left(t - 4\times 10^{-3}\right) - 1\times 10^6\left(t - 9\times 10^{-3}\right) \\ \operatorname{var}\left(S_{X_2}(t)\right) = 125\times 10^{12} \times \left(t - 4\times 10^{-3}\right)^{1.7} + 50\times 10^9\left(t - 9\times 10^{-3}\right) \end{cases}$$

On the variance of X2out(t)

- Thus,

$$\mathrm{var}\left(X_{2,\mathrm{out}}(t)\right)=\max\left(\mathrm{var}\left(X_2(t)\right),\mathrm{var}\left(S_{X_2}(t)\right)\right)=$$

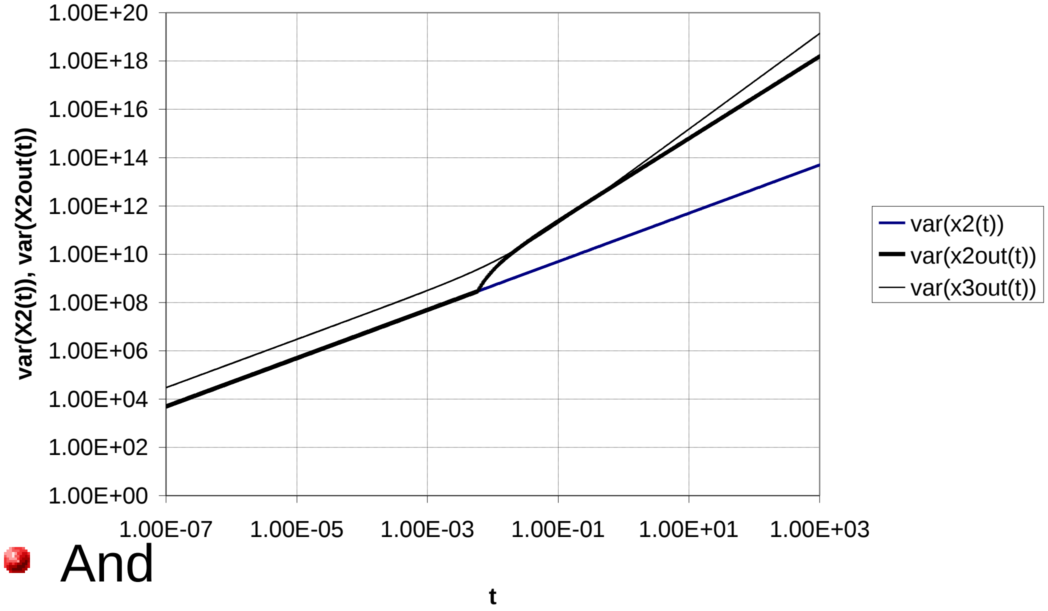
$$=\max\left(0.05\times10^{12}t,125\times10^{12}\times\left(t-4\times10^{-3}\right)^{1.7}+50\times10^{9}\left(t-9\times10^{-3}\right)\right)$$

- Then, $X_{2out}(t)$ is offered to the third Strict Priority scheduler, and we want to calculate the variance of the output flow of this scheduler (we call it $X_{3out}(t)$)
- var(X_{3out}(t))=max(var(X_{2out}(t)), S(X_{2out}(t))):

$$\mathrm{var}\left(S_{X_{2,out}}(t)\right)=\mathrm{var}(X_{1,out}(t))$$

- where

$$\mathrm{var}\left(X_{1,out}(t)\right)=\max\left(\mathrm{var}\left(X_1(t)\right),\mathrm{var}\left(S_{1,out}(t)\right)\right)$$



- And

$$\mathrm{var}\left(S_{X1,out}(t)\right)=\mathrm{var}\left(Z_1(t)\right)+\mathrm{var}\left(Y_1(t)\right)=$$

$$=2\times10^{6}\times150\times10^{3}t+3.2\times10^{6}\times5\times10^{6}\times t^{1.98}=$$

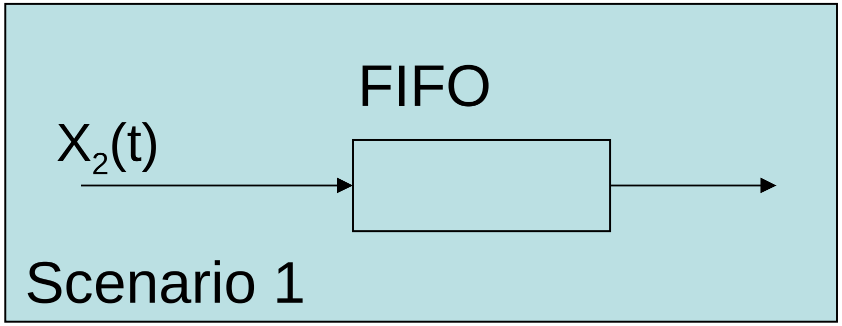
$$=300\times10^{9}t+16\times10^{12}\times t^{1.98}$$

- Thus

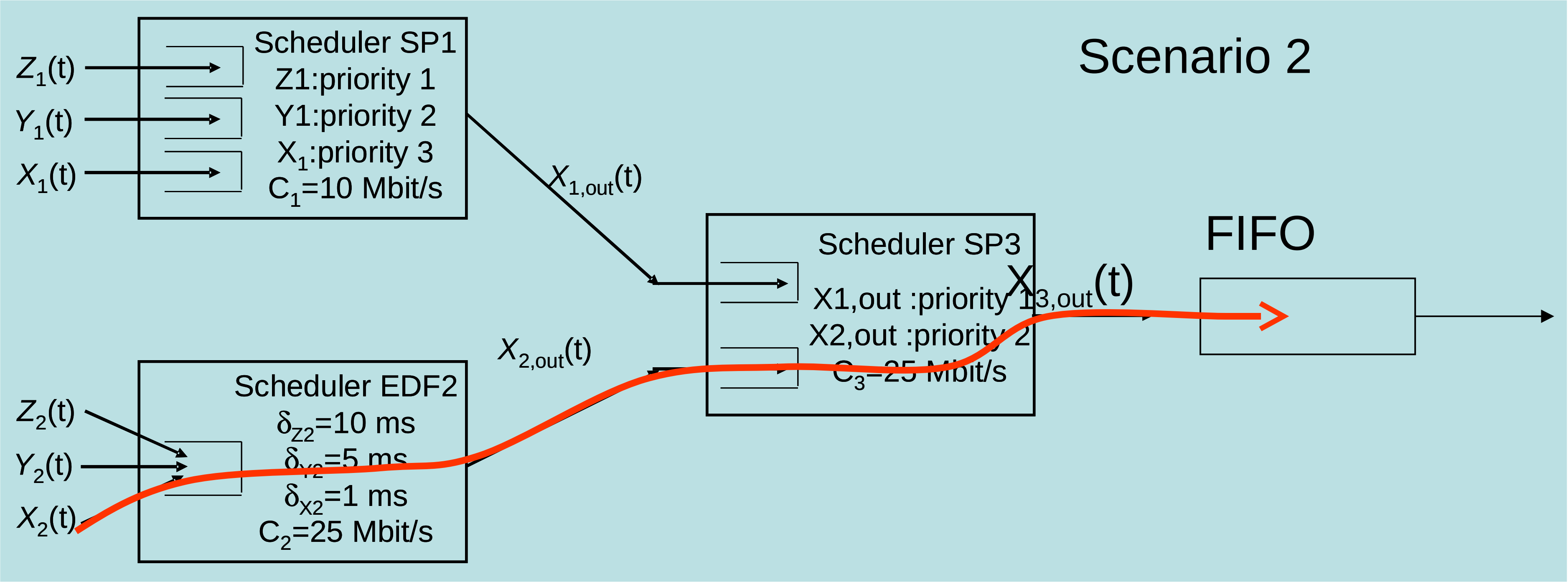
$$\mathrm{var}\left(X_{3,out}(t)\right)=$$

$$\max\left(300\times10^{9}t,\max\left(300\times10^{9}t,300\times10^{9}t+16\times10^{12}\times t^{1.98}\right)\right)$$

Analysis of the behavior of X3out(t)



- Now, we consider two scenarios
  - Scenario 1: X2(t) is directly offered to a FIFO scheduler
  - Scenario 2: X2(t) is offered to the same FIFO scheduler after having crossed the network (i.e., we offer X3out(t))



Analysis of the behavior of X3out(t)

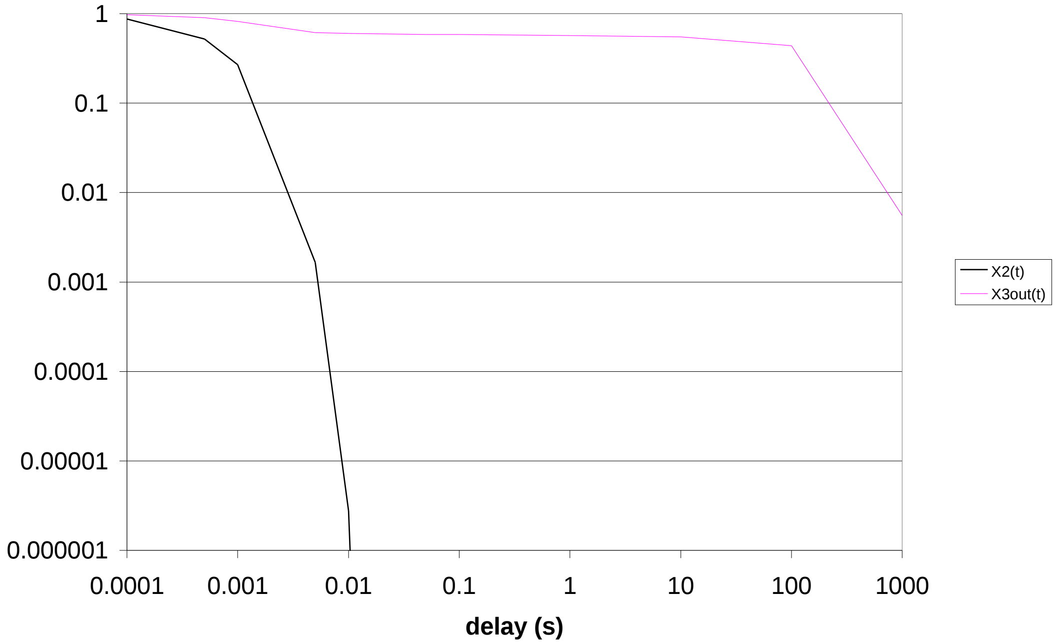
- Now, the calculation is simple, as it is sufficient to calculate the alpha function in both cases:

$$\alpha_{X_2}\left(t\right)=-\frac{m_{X_2}t-C\left(t+d\right)}{\sqrt{\operatorname{var}\left(X_2\left(t\right)\right)}};\alpha_{X_{3,out}}\left(t\right)=-\frac{m_{X_2}t-C\left(t+d\right)}{\sqrt{\operatorname{var}\left(X_{3,out}\left(t\right)\right)}};$$

- By repeating the calculation for different values of d, we obtain the delay curve in both cases

Analysis of the behavior of X3out(t)

- The difference between the queueing behavior of X2(t) and X3out(t) is impressive
- X2(t) acquires a heavy LRD when it crosses the network
- Actually, it interferes with a flow with *H*=0.99, an extremely high value of hurst parameter
- X2(t) seems to acquire a LRD behavior
- This is a known fact: LRD tends to spread in the Internet
- However, the methods described in this courses are the first allowing us to calculate analytically this kind of behavior


delay (s)
X2(t)
X3out(t)