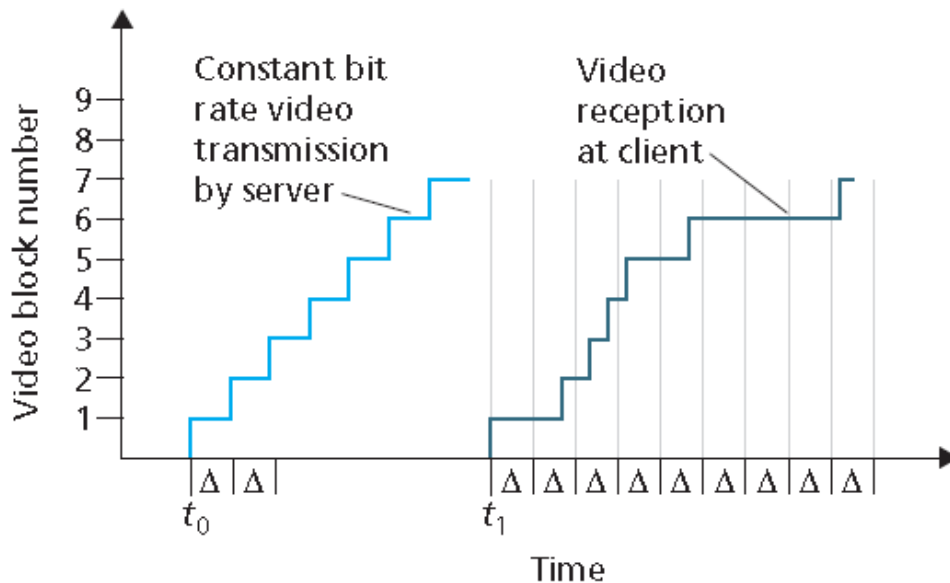


TRAVAUX DIRIGES N° 3

Exercice 1:

Supposons qu'une vidéo est codée à un débit fixe au niveau du serveur, et chaque bloc vidéo contient donc des trames vidéo qui doivent être lu sur la même durée fixe, Δ . Le serveur transmet le premier bloc vidéo à t_0 , le deuxième bloc à $t_0 + \Delta$, le troisième bloc à $t_0 + 2\Delta$, et ainsi de suite. Une fois que le client commence la lecture, chaque bloc doit être lu Δ secondes après le bloc précédent.

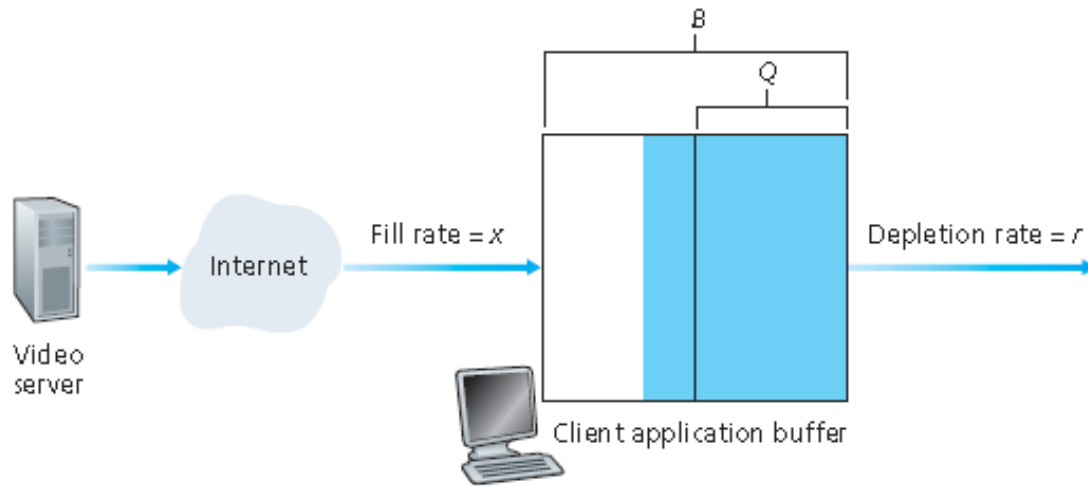
- a) Supposons que le client commence la lecture dès que le premier bloc arrive à t_1 . Dans la figure ci-dessous, combien de blocs de vidéo (y compris le premier bloc) seront arrivés au client à temps pour leur lecture? Expliquer ?
- b) Supposons que le client commence la lecture à $t_1 + \Delta$. Combien de blocs de vidéo (y compris le premier bloc) seront arrivés au client à temps pour leur lecture? Expliquer ?
- c) Dans le même scénario en (b), quel est le plus grand nombre de blocs stockés dans le buffer du client, en attente de la lecture? Expliquer ?
- d) Quel est le plus petit délai de lecture pour assurer que chaque bloc vidéo est arrivé à temps pour sa lecture?



Exercice 2

Considérons la Figure ci-dessous. B indique la taille du buffer de l'application client et Q indique le nombre de bits qui doivent être mis en tampon avant que l'application cliente commence la lecture. De plus, r indique le débit de consommation vidéo. Supposons que le débit du serveur est constant et égale x bps (chaque fois que le tampon client n'est pas plein).

- a) Supposons que $x < r$. Rappelez l'effet sur la lecture. Déterminez la durée de chaque période de lecture continue et de coupure en fonction de Q , r et x .
- b) Supposons maintenant que $x > r$. A quelle instant $t = t_f$ le buffer de l'application cliente devient plein?



Problem 1

- a) Client begins playout as soon as first block arrives at t_1 and video blocks are to be played out over the fixed amount of time, d . So it follows that second video block should be arrived before time $t_1 + d$ to be played at right time, third block at $t_1 + 2d$ and so on. We can see from figure that only blocks numbered 1,4,5,6 arrive at receiver before their playout times.
- b) Client begins playout at time $t_1 + d$ and video blocks are to be played out over the fixed amount of time, d . So it follows that second video block should be arrived before time $t_1 + 2d$ to be played at right time, third block at $t_1 + 3d$ and so on. We can see from figure that video blocks numbered from 1 to 6 except 7 arrive at receiver before their playout times.
- c) Maximum two video blocks are ever stored in the client buffer. Video blocks numbered 3 and 4 arrive before $t_1 + 3d$ and after $t_1 + 2d$, hence these two blocks are stored in the client buffer. Video block numbered 5 arrives before time $t_1 + 4d$ and after $t_1 + 3d$, which is stored in the client buffer along with already stored video block numbered 4.
- d) The smallest playout delay at the client should be $t_1 + 3d$ to ensure that every block has arrived in time.

Problem 2

- e) During a playout period, the buffer starts with Q bits and decreases at rate $r - x$. Thus, after $Q/(r - x)$ seconds after starting playback the buffer becomes empty. Thus, the continuous playout period is $Q/(r - x)$ seconds. Once the buffer becomes empty, it fills at rate x for Q/x seconds, at which time it has Q bits and playback begins. Therefore, the freezing period is Q/x seconds.
- f) Time until buffer has Q bits is Q/x seconds. Time to add additional $B - Q$ bits is $(B - Q)/(x - r)$ seconds. Thus the time until the application buffer becomes full is $\frac{Q}{x} + \frac{B-Q}{x-r}$ seconds.