Model-Based Estimation of Streaming Performance

draft-ko-ippm-streaming-performance

Ken Ko ken.ko@adtran.com

IPPM – Orlando – IETF86 – March 10-15, 2013

Video streaming

- Over half of all consumer Internet traffic since 2011[1]
- Near-RT traffic
  - Marginal throughput can cause excessive wait time before video starts
  - Throughput variation can cause interruption in video playout ("video freeze")
- In US, FCC has measured video streaming performance for last two years[2]
  - Send a stream at a constant bit rate
  - Write received data to buffer
  - Read data from buffer at constant rate
  - Record metrics related to buffer fill

[1] Cisco Visual Networking Index, May 30, 2012

[2] Methodology - Measuring Broadband America Report February 2013

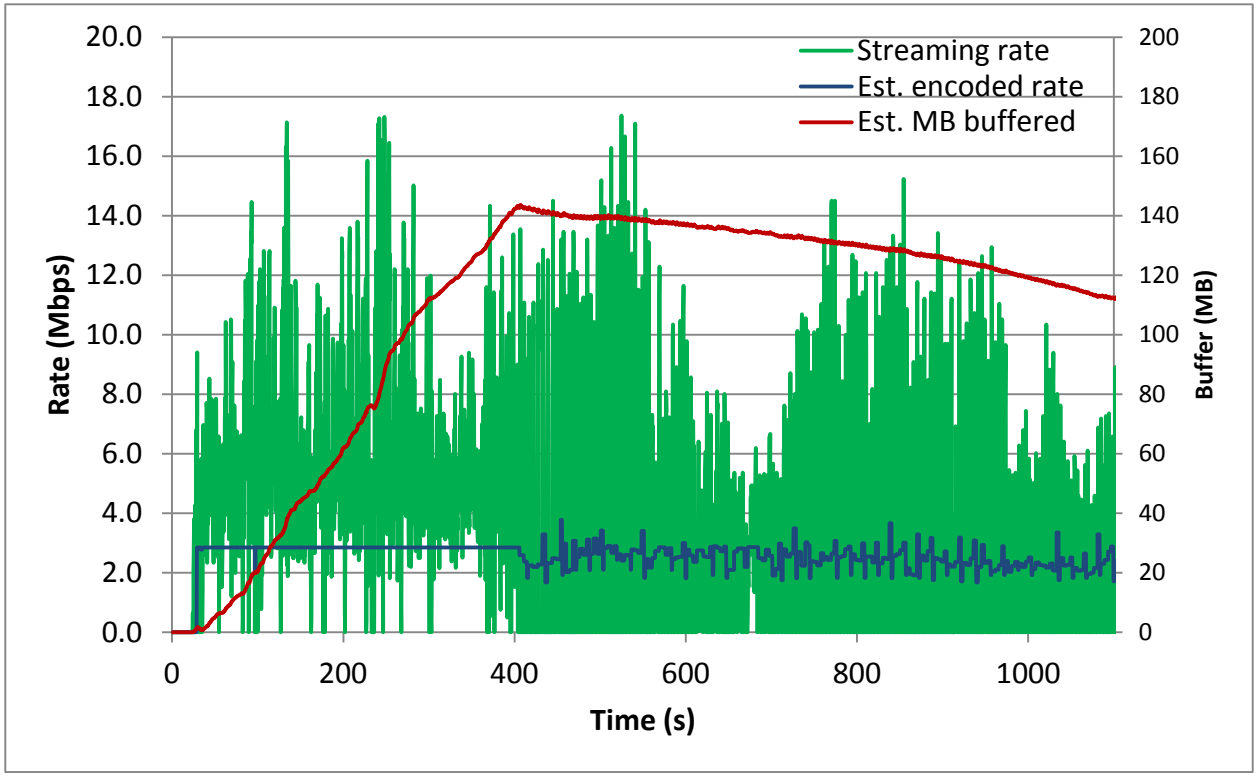
Advantages of Model-Based Methodology

- Current state
  - Dedicated test generates traffic on network
  - Limited to one streaming rate
  - Result is almost binary (does it work at that rate or not?)
- Proposed methodology
  - Use (or re-use) TCP throughput results
    - Throughput test is probably already part of performance test suite
    - No additional network traffic required
  - One TCP sample metric can be run against model many times
    - Different encoded rates
    - Different buffer depths
  - Can determine the maximum supported streaming rate to a close tolerance

OTT Video Characteristics

- Multiple protocols under "OTT video streaming" umbrella
- Common characteristics
  - Almost always over TCP
  - Average encoded rate, whether CBR or VBR
    - Ignore adaptive rate streaming for the moment (see backup)
  - Usually an initial rate higher than average to fill buffer
  - Usually transmitted in bursts separated by idle time
    - Average transmitted rate $\approx$ average encoded rate after initial burst

Example OTT Video Stream



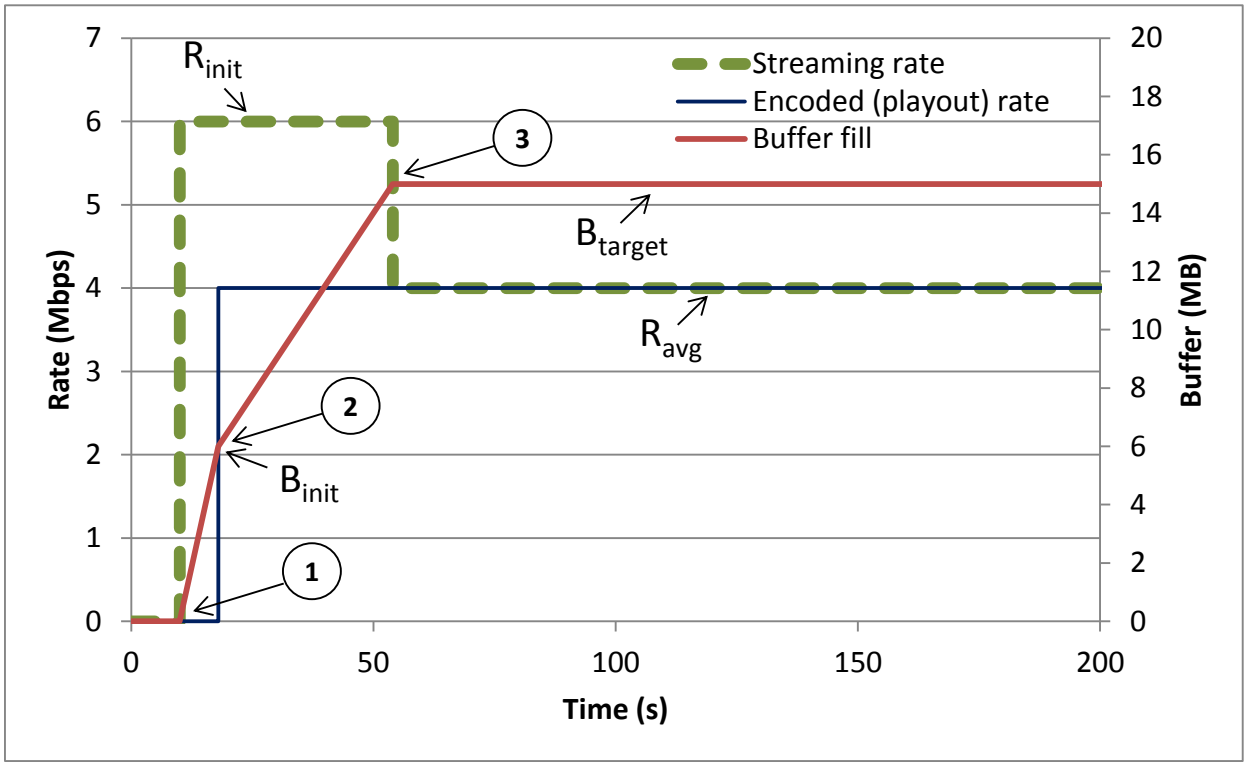
- Initial streaming rate (< 400 seconds) ≈ 5.8 Mbps
- Average encoded rate ≈ 2.9 Mbps

Streaming model definition

Streaming Model Parameters and States

- Parameters
  - **Rinit**: initial streaming rate
    - Maximum rate at which buffer initially fills
  - **Ravg**: average encoded rate
    - Rate at which the dejitter buffer is emptied
    - Also, maximum rate at which buffer fills in MAINTAIN state
  - **Binit**: initial buffer fill depth
    - Depth at which buffer starts to be emptied
  - **Btarget**: Target buffer fill depth
    - Depth at which intended streaming rate = encoded rate
- States
  - **FILL_NOPLAY**: Write to buffer at min(Rinit, Tput), do not read
  - **FILL_PLAY**: Write to buffer at min(Rinit, Tput), read at Ravg
  - **MAINTAIN**: Write to buffer at min(Ravg, Tput), and read at Ravg

Model Behavior (ideal)



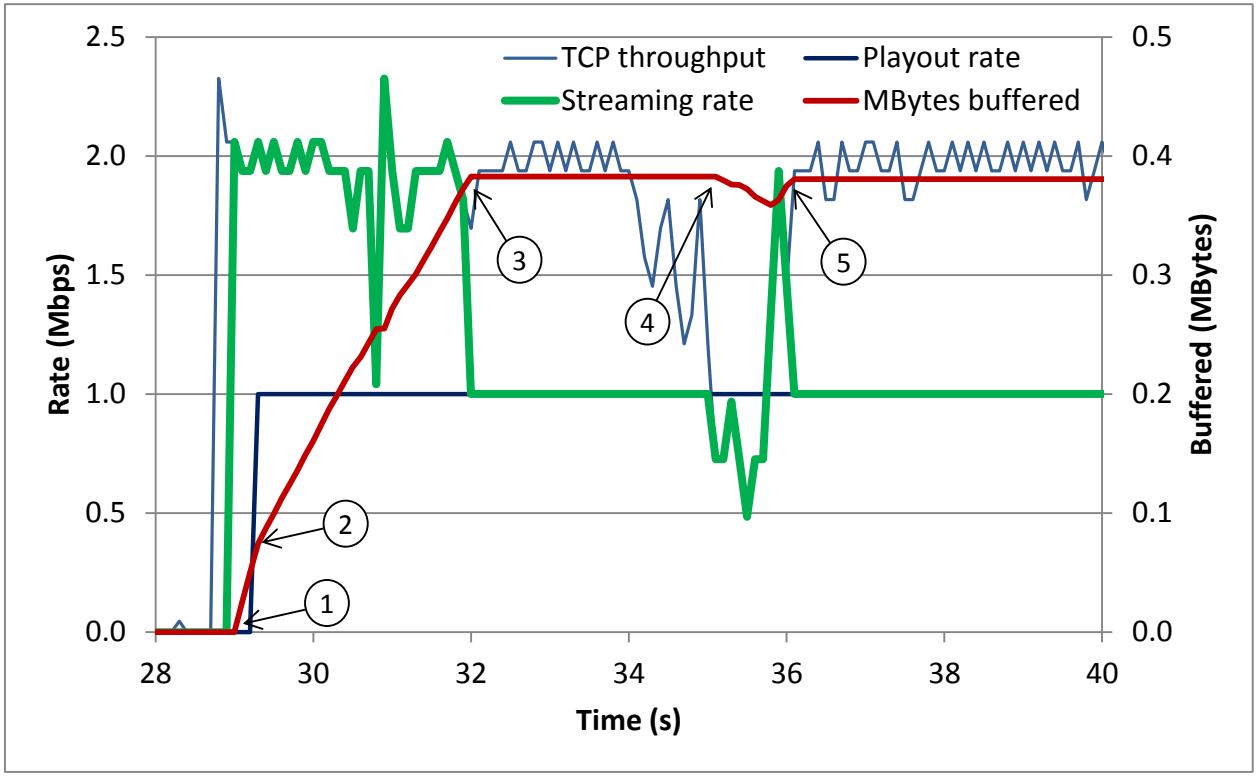
1. Streaming starts at Rinit: buffer filling
2. Buffer fill reaches Binit: playout starts
3. Buffer fill reaches Btarget: streaming rate reduced to Ravg

Model-based methodology with examples

Model-Based Methodology

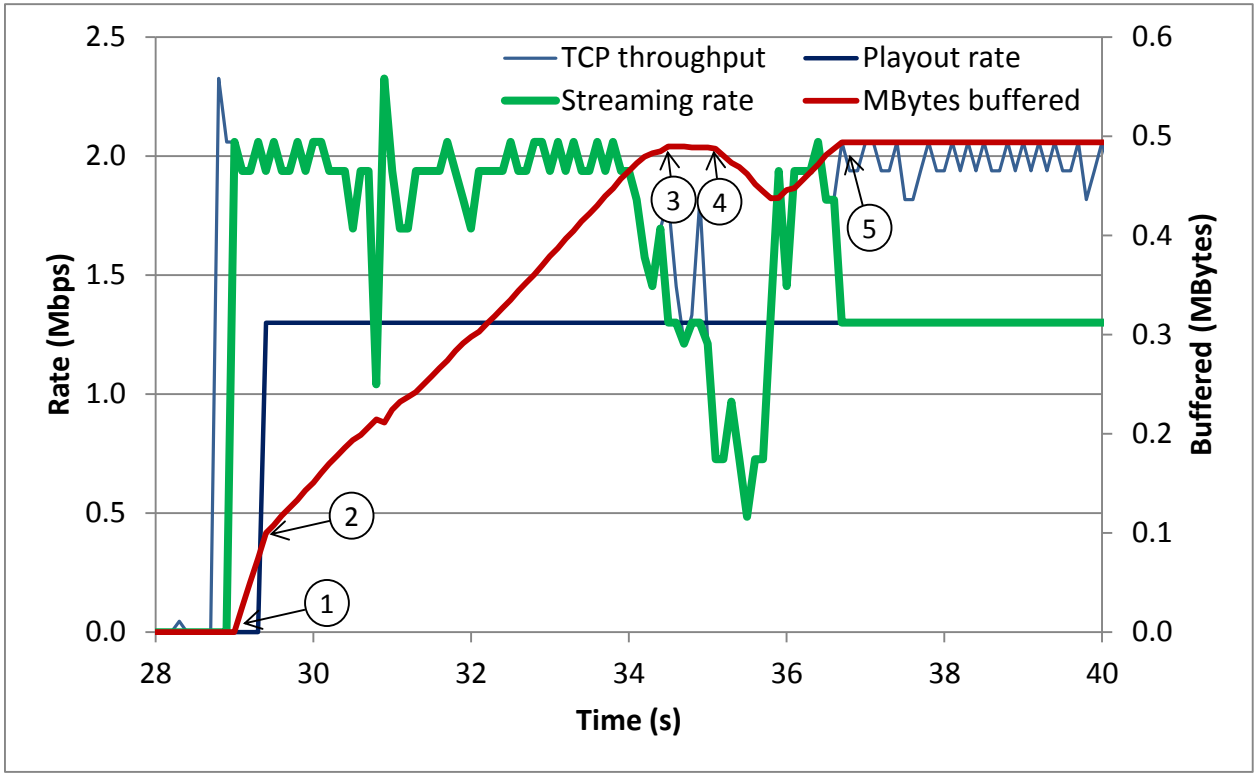
1. Perform TCP throughput test
   - Collect short-term TCP throughput values at regular intervals
   - Sample metric = series of short-term throughput values
2. Define model parameter values
3. Apply streaming model to sample metric
   - Generate derived sample metric showing buffer fill over time
   - Generate statistics based on derived sample metric
4. If desired, go to step 2 and define different parameter values

Example: Ravg = 1.0 Mbps



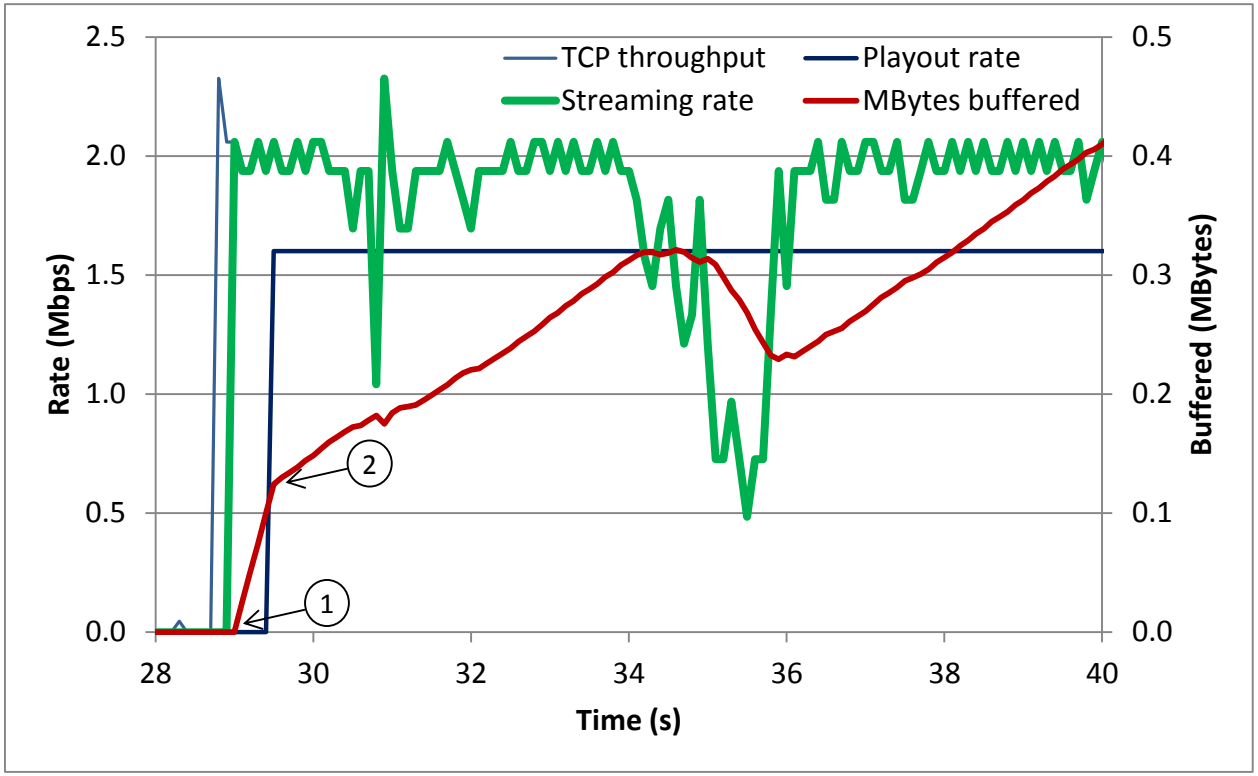
1. Streaming starts at Rinit: buffer filling
2. Buffer fill reaches Binit: playout starts
3. Buffer fill reaches Btarget: streaming rate →Ravg
4. Reduced throughput, buffer < Btarget: streaming rate →Rinit
5. Buffer back at Btarget: streaming rate →Ravg

Example: Ravg = 1.3 Mbps



1. Streaming starts at Rinit: buffer filling
2. Buffer fill reaches Binit: playout starts
3. Buffer fill reaches Btarget: streaming rate →Ravg
4. Reduced throughput, buffer < Btarget: streaming rate →Rinit
5. Buffer back at Btarget: streaming rate →Ravg

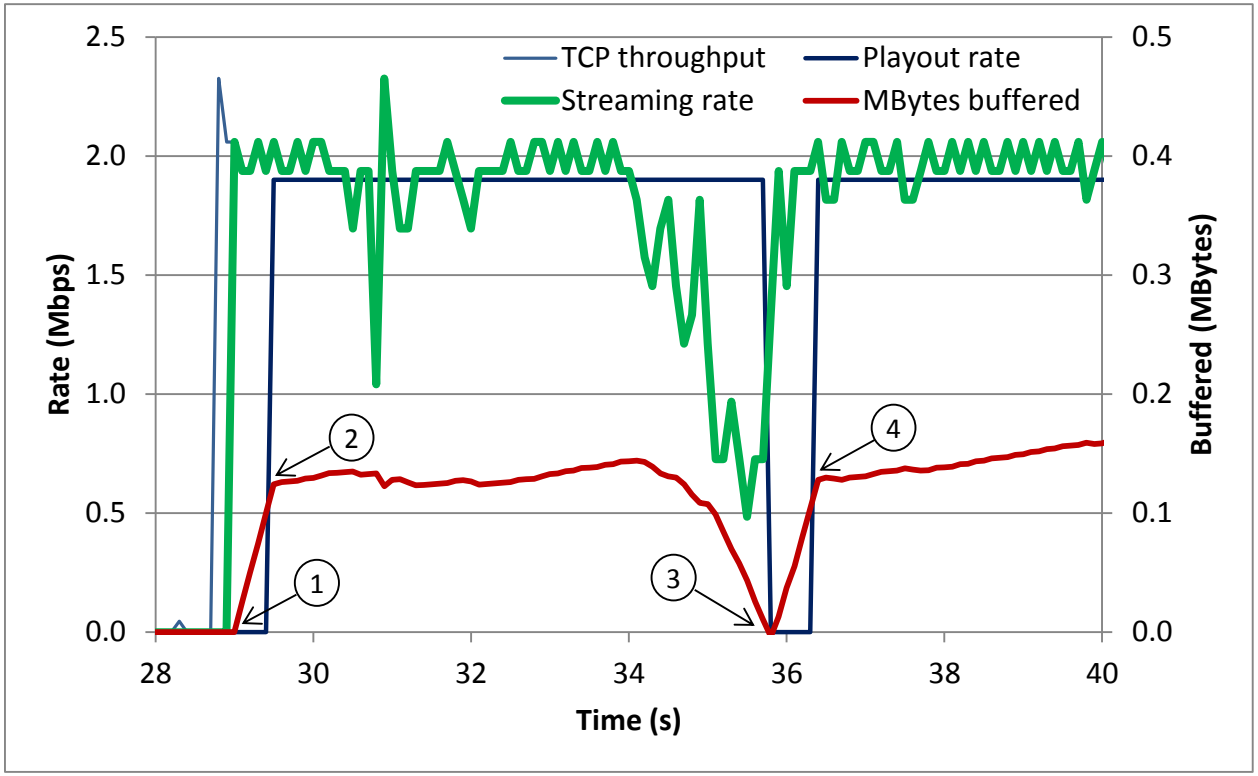
Example: Ravg = 1.6 Mbps



1. Streaming starts at Rinit: buffer filling
2. Buffer fill reaches Binit: playout starts

Buffer fill never reaches Btarget

Example: Ravg = 1.9 Mbps



1. Streaming starts at Rinit: buffer filling
2. Buffer fill reaches Binit: playout starts

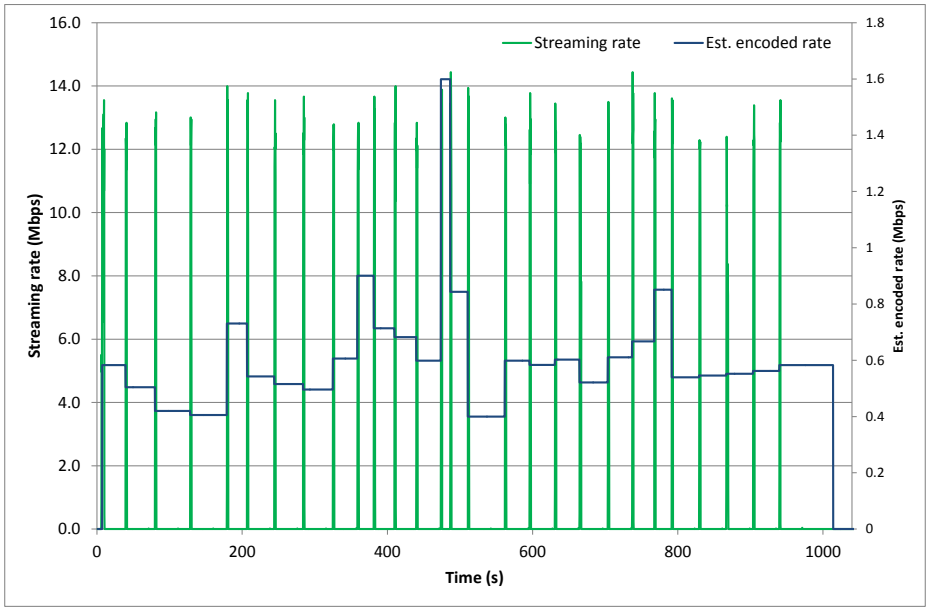
   Buffer fill never reaches Btarget
3. Buffer under-run: playout stops (video freeze)
4. Buffer fill reaches Binit: playout restarts

Backup Slides

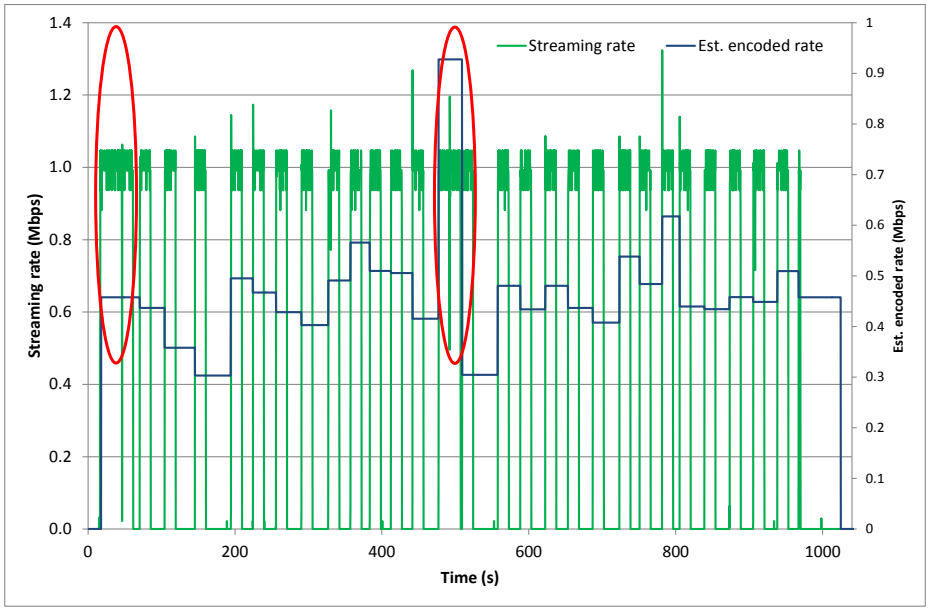
Model Pseudocode

```
For k = 1 to k_max      // Run the simulation for // each time interval T(k)
   Switch(Model_state)

      // Buffer filling, no playout
      Case FILL_NOPLAY:
         B(k) = B(k-1) + min(Finit, R(k))
         If B(k) >= Btarget then
            Model_state = MAINTAIN
         Else If B(k) >= Binit then
            Model_state = FILL_PLAY
         End if

      // Buffer filling, media playing out
      Case FILL_PLAY:
         B(k) = B(k-1) + min(Finit, R(k)) - P
         If B(k) >= Btarget then
            Model_state = MAINTAIN
         Else If B(k) <= 0 then
            B(k) = 0
            Model_state = FILL_NOPLAY
         End if

      // Buffer at target, media playing out
      Case MAINTAIN:
         B(k) = B(k-1) + min(Fmaint, R(k)) - P
         If B(k) <= 0 then
            B(k) = 0

         Model_state = FILL_NOPLAY
         Else If B(k) < Btarget then

         Model_state = FILL_PLAY
         End if

      End switch

Next k
```

Bursty Delivery vs. Average Rate



(a)



(b)

- Same content
  - (a) received over >15 Mbps channel
  - (b) received over channel shaped to 1 Mbps
- Blue traces show estimated encoded rate
- Rate-limited graph shows spreading of bursts
  - Red circles highlight where received bursts have merged

The closer the encoded rate is to the channel rate, the closer the received traffic resembles the streaming model

What About VBR and Adaptive Rate Streaming?

- VBR encoding
  - Encoded (and transmitted) bit rate varies around an average value
  - Commonly used but not modeled here
    - Ravg models CBR
  - Can VBR be characterized in a way that enhances the model?
- Adaptive rate streaming
  - Streams available at multiple encoded rates
  - Receiver can request different rate based on channel performance
  - Better to characterize performance by testing against model at different rates