An Empirical Study of Delay Jitter Management Policies

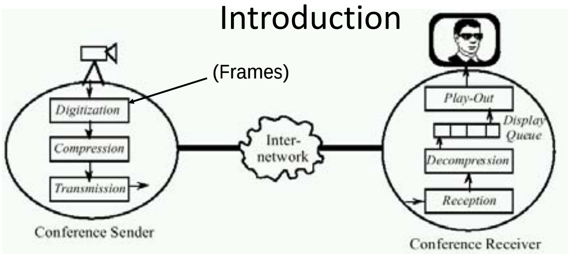
D. Stone and K. Jeffay
Computer Science Department
University of North Carolina, Chapel Hill

ACM Multimedia Systems
Volume 2, Number 6
January 1995

Introduction

- Want to support interactive audio
- "Last mile" is LAN (including bridges, hubs) to desktop
  - Study that
  - (MLC: 1995 LANs looked a lot like today's WANs)
- Transition times vary, causing gaps in playout
  - Can ameliorate with *display queue* (buffer)

Introduction



- *Display latency* – time from acquisition at sender to display at receiver (gap occurs if > previous frame)
- *End-to-end delay* – time from acquisition to decompression
  - Varies in time (transmit + (de)compress), *delay jitter*
- *Queuing delay* – time from buffer to display (change size)

Gaps versus Delay

- Can prevent gaps by having constant delay
  - Network reserves buffers
    - Ala telephone networks
    - But *not* today's Internet
- Plus
  - will still have (unreserved) LAN as "last mile"
  - OS and (de)compression can still cause jitter
- Thus, tradeoff between gaps and delay must be explicitly managed by conferencing system
  - Change size of display queue
  - The larger the queue, the larger the delay and the fewer the gaps and vice versa

This Paper

- Evaluates 3 policies for managing display queue
  - *I-policy* and *E-policy* from [NK92]
    - (*I* is for late data ignored, *E* is for late data expand time)
  - *Queue Monitoring* from this paper
- Empirical study
  - Audioconference (VoIP) on a LAN
  - Capture traces
- Simulator to compute delay and gaps

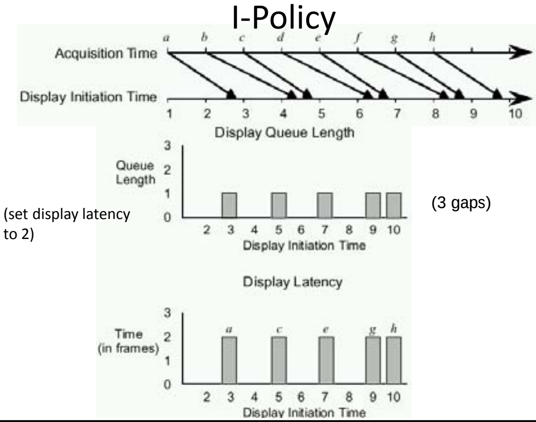
Outline

- Introduction **(done)**
- The I- and E-policies **(next)**
- The Queue Monitoring policy
- Evaluation
- The Study
- Summary

The Effect of Delay Jitter

- If display latency worse than largest end-to-end latency, then no gaps
  - (When is this not what we want?)
- Playout with low latency and some gaps *preferable* to high-latency and no gaps
- What if a frame arrives after its playout time?
- Two choices:
  - *I-policy* – single fixed latency (the queue parameter), so discard
  - *E-policy* – late frames always displayed, so expand playout time

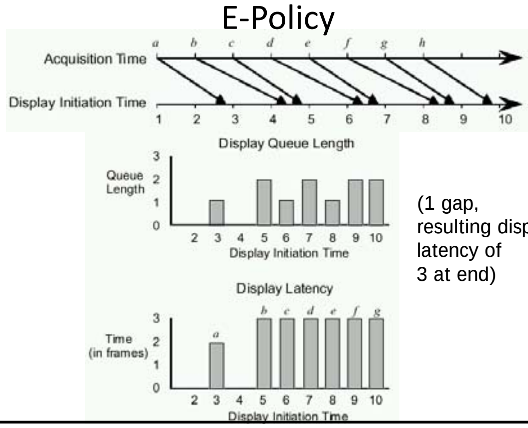
I-Policy



(set display latency to 2)

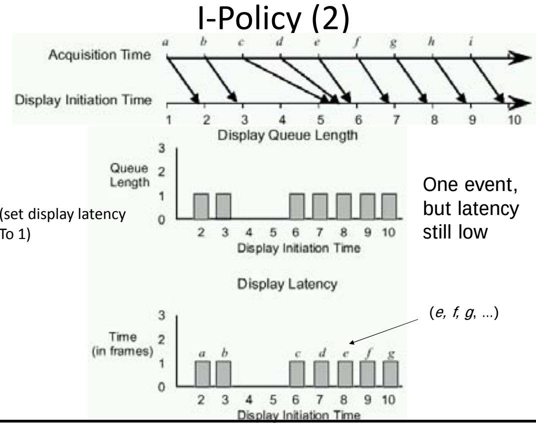
(3 gaps)

E-Policy



(1 gap, resulting display latency of 3 at end)

I-Policy (2)

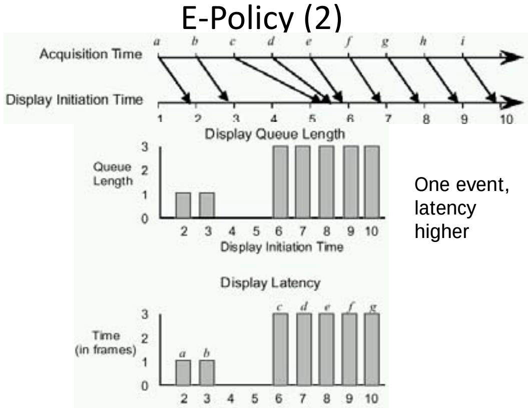


(set display latency To 1)

One event, but latency still low

(*e*, *f*, *g*, …)

E-Policy (2)



One event, latency higher

Policy Summary

- Display latency chosen implicitly with *E-policy*
- Choose it explicitly with *I-policy*
- What is the right display latency amount?
  - Depends on application
    - Example: surgeon interacting during operation vs. viewing televised lecture
  - Depends on network and machines
    - Can vary across a long run
- So, need a policy that allows display latency to be chosen dynamically

Outline

- Introduction (done)
- The I- and E-policies (done)
- The Queue Monitoring policy (next)
- Evaluation
- The Study
- Summary

Adjusting Display Latency

- VoIP with silence detection can be modeled as series of *talkspurts*
  - Sound and then silence
- Adjust display latency between talkspurts
- [NK92] said observe last *m* fragments, discard *k* largest delays and choose display latency as greatest delay
  - Recommend $m > 40$ and $k = 0.07 \times m$
- (Other approaches proposed, since)

Monitor the Display Queue

- Measuring end-to-end latency is difficult because needs synchronized clocks
- Instead, observe length of display queue over time
  - If end-to-end delay constant, queue size will remain the same
  - If end-to-end delay increases, queue shrinks
  - If end-to-end delay decreases, queue expands
- If queue length > 2 for some time, can reduce queue (hopefully) without causing a gap
  - "some time" is parameter, *n*, in frame times
  - Implement with counters for each of *m* frames in queue
  - If any of the *m* times > *n*, discard frame and reset
    - (However, keep queue at least 2)
  - Use QM-120 as default
    - Adjust every 120 frames (about 2 seconds)

Outline

- Introduction (done)
- The I- and E-policies (done)
- The Queue Monitoring policy (done)
- Evaluation (next)
- The Study
- Summary

Comparing Policies

- If A has lower latency *and* gaps than B, then A is better
- If A lower latency, but also A more gaps then which is better?
  - Depends upon
    - relative amounts
    - resolution
    - application requirements
  - Few standards

Comparing Policies

- Assume:
  - Differences in latency of 15 ms or more significant
  - Difference in gap rate of 1 per minute significant
- A is better than B if either gap or latency better *and* other is same or better
- Equal if same in both dimensions
- Incomparable if each is better in one dimension
- Note, for *I-policy*, synchronized clocks difficult
  - Instead, delay first packet for amount of time (try 2 and 3 frames in this paper)

Outline

- Introduction **(done)**
- The I- and E-policies **(done)**
- The Queue Monitoring policy **(done)**
- Evaluation **(done)**
- The Study **(next)**
- Summary

The Study

- Run videoconference
  - Use audio only
- Record end-to-end delay
- Input into simulator to evaluate different policies
  - Effectively, a trace-driven simulation
  - Ensures network conditions “the same” when comparing policies

Videoconference

- Built at UNC
- Runs on IBM PS/2
- Uses UDP
- IBM-Intel ActionMedia 750
  - 30 fps, 256x240, 8-bit color (6-8 k frames)
    - But video is disabled
  - Audio 60 fps, 128 kb/second into 16.5ms frames (266 byte packets)

Network

- 10 Mb Ethernets and 16 Mb token rings
- 400 Unix workstations and Macs
- NFS and AFS (file systems)
- Send machine → token-ring → gateway → department Ethernet → bridge → department Ethernet → gateway → token-ring → Display machine

Data

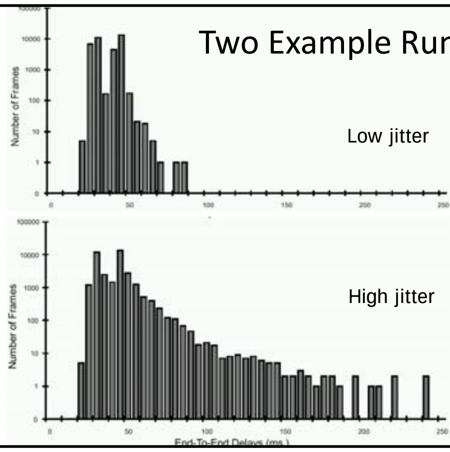
- Gather data for 10 minute interval
- 28 Runs total
  - 24 runs between 6am and 5pm
  - 4 runs between midnight and 1am
- Record:
  - Acquisition times
  - Display times
  - Adjust times for clock difference and drift
- Input traces into simulator
  - Outputs average display latency
  - Outputs average gap rate

Basic Data

Run	Time of Day	Avg. Delay ms.	Max. Delay ms.	Lost Frames	Duplicate Frames
1	06:03	38	76	1	0
2	06:25	38	88	3	0
3	06:36	37	171	5	0
4	06:47	37	105	1	0
5	08:03	38	115	1	0
6	08:14	37	73	2	0
7	08:25	38	184	7	0
8	08:36	39	157	1	0
9	10:02	41	186	23	0
10	10:16	40	124	4	0
11	10:31	41	213	7	0
12	10:49	40	140	6	0
13	11:57	39	110	5	0
14	12:08	41	138	5	0

(Comments?)

Two Example Runs



Results

Run	I-Policy 2 (I-2) Latency ms.	Gaps /min.	I-Policy 3 (I-3) Latency ms.	Gaps /min.	E-Policy Latency ms.	Gaps /min.	QM (QM-120) Latency ms.	Gaps /min.	QM vs. I2	QM vs. I3	QM vs. E
1	80	0.1	97	0.1	75	0.2	66	0.3	0	+	0
2	75	0.5	91	0.3	72	0.5	66	0.6	0	+	0
3	69	3.6	86	2.8	140	0.9	68	1.4	+	+	+
4	65	0.7	82	0.4	104	0.6	65	0.6	0	+	+
5	71	0.6	88	0.4	93	0.5	68	0.5	0	+	+
6	70	0.3	86	0.2	76	0.4	70	0.5	0	+	0
7	73	2.9	90	1.6	106	1.2	72	1.9	+	+	+
8	62	5.1	79	2.4	106	0.9	75	1.3	+	+	+
9	81	23.0	98	12.6	118	2.8	87	7.6	+	+	
10	70	14.6	87	3.6	113	0.8	78	3.9	+	0	
11	66	25.2	83	6.9	133	1.4	83	4.8		+	
12	71	9.6	87	3.4	114	0.9	76	2.7	+	0	
13	67	9.6	84	2.8	96	0.8	72	2.1	+	0	
14	72	15.1	88	3.9	101	1.1	80	3.9	+	0	

QM-120 better than I-2 for all but 11
(I-2 has gap per 2 seconds vs per 11 seconds)

Results

15	76	4.4	92	1.7	117	0.9	79	2.9	+	-	
16	68	18.6	85	8.0	114	1.8	80	6.6	+	+	
17	77	22.0	93	12.1	146	1.8	88	7.5	+	+	
18	76	13.0	93	4.1	131	0.7	85	4.8	+	0	
19	66	5.0	82	1.3	87	0.9	72	1.8	+	0	+
20	73	11.3	90	4.1	98	1.6	77	3.7	+	0	
21	70	12.8	87	6.1	159	1.5	76	3.6	+	+	
22	79	1.4	95	0.5	100	0.6	77	1.0	0	+	+
23	75	0.4	91	0.2	84	0.4	74	0.6	0	+	0
24	77	39.6	93	15.0	104	1.8	87	5.2	+	+	
25	65	0.8	81	0.4	66	0.7	65	0.7	0	+	0
26	64	5.4	81	0.9	122	0.6	69	1.5	+	0	+
27	70	7.8	87	3.8	107	1.3	73	3.5	+	0	
28	76	0.3	93	0.2	75	0.3	73	0.4	0	+	0

QM-120 better than I-3 for all but 15
Latency of QM-120 better than that of I-3

Better than E for low jitter runs

Summary Results

QM Better	18	18	8
QM Equivalent	9	9	6
QM Worse	0	1	0
Incomparable	1	0	14

- If want low latency, not large gap rate
→ QM out performs all I-policies, E-policies

Threshold as a Parameter

- Vary thresholds for adjusting queue latency
- 30 frame times (.5s)
- 60 frame times (1s)
- 120 frame times (2s)
- 600 frame times (10s)
- 3600 frame times (1 min)

Results

Run	QM (30) Latency ms.	Gaps /min.	QM (60) Latency ms.	Gaps /min.	QM (120) Latency ms.	Gaps /min.	QM (600) Latency ms.	Gaps /min.	QM (3600) Latency ms.	Gaps /min.	120 vs. 30	120 vs. 60	120 vs. 600	120 vs. 3600
1	64	0.3	65	0.3	66	0.3	73	0.3	75	0.2	0	0	0	0
2	65	0.7	65	0.7	66	0.6	66	0.6	67	0.6	0	0	0	0
3	67	1.7	67	1.4	68	1.4	74	1.4	103	1.1	0	0	0	+
4	65	0.6	65	0.6	65	0.6	69	0.6	83	0.6	0	0	0	+
5	67	0.5	68	0.5	68	0.5	69	0.5	81	0.5	0	0	0	0
6	70	0.5	70	0.5	70	0.5	70	0.5	76	0.4	0	0	0	0
7	70	2.3	71	1.9	72	1.9	77	1.7	95	1.4	0	0	0	+
8	68	2.0	70	1.5	75	1.3	83	1.0	97	1.0	0	0	0	+
9	77	13.1	83	9.0	87	7.6	102	4.9	117	3.0	+	+	-	
10	72	6.6	75	5.0	78	3.9	89	1.6	98	1.0	+	+	-	
11	72	8.3	76	6.3	83	4.8	98	3.4	124	1.7	+	+		
12	72	5.3	74	3.3	76	2.7	86	1.9	103	1.2	+	0	0	
13	69	3.5	70	2.7	72	2.1	82	1.4	91	1.0	+	0	0	
14	74	6.7	76	6.0	80	3.9	92	1.8	99	1.2	+	+	-	

Comments?

Summary

QM-120 Better	13	8	0	5
QM-120 Equivalent	15	20	17	9
QM-120 Worse	0	0	7	0
Incomparable	0	0	4	14

- QM-600 is best relative to QM-120
- QM-120 better than all others
- (MLC: what about in between? Should be optimal for each setting)
- Also:
  - QM-3600 similar to E-policy
  - QM-30 and QM-60 similar to I-2

Decay Thresholds

- Want to converge slowly to lowest latency
- Define *base threshold* for queue length of 3
- Define *decay factor* for other queue lengths
- Base of *3600*, decay of *2* would have:
  - Wait 3600 frame times when queue is 3
  - 1800 for 4
  - 900 for 5
  - …

Results

Run	QM (120) Latency ms.	QM (120) Gaps /min.	QM (120,2) Latency ms.	QM (120,2) Gaps /min.	QM (600,2) Latency ms.	QM (600,2) Gaps /min.	QM (3600,2) Latency ms.	QM (3600,2) Gaps /min.	120 vs. 120,2	120 vs. 600,2	120 vs. 3600,2
1	66	0.3	66	0.3	73	0.3	75	0.2	0	0	0
2	66	0.6	66	0.6	66	0.6	67	0.6	0	0	0
3	68	1.4	67	1.6	68	1.4	78	1.4	0	0	0
4	65	0.6	65	0.6	68	0.6	82	0.6	0	0	+
5	68	0.5	68	0.5	68	0.5	72	0.5	0	0	0
6	70	0.5	70	0.5	70	0.5	76	0.4	0	0	0
7	72	1.9	71	1.9	72	1.8	82	1.5	0	0	0
8	75	1.3	74	1.5	79	1.0	89	1.0	0	0	0
9	87	7.6	85	8.5	97	5.7	113	3.2	0	-	
10	78	3.9	78	4.2	88	1.7	97	1.0	0	-	
11	83	4.8	81	5.2	91	3.6	110	2.1	0	-	
12	76	2.7	75	2.8	82	2.0	94	1.3	0	0	
13	72	2.1	72	2.1	81	1.4	91	1.0	0	0	
14	80	3.9	79	4.0	90	2.0	99	1.2	0	-	

Summary Results

QM-120 Better	1	0	2
QM-120 Equivalent	27	17	12
QM-120 Worse	0	11	1
Incomparable	0	0	13

- QM-(120,2) didn't help
- QM-(600,2) better than QM-120
  - Also better than QM-600 by decreasing latency and gap rate almost the same
- QM-(3600,2) better than QM-120
  - Also better than QM-3600
- So, decay is useful for large base thresholds, but may hurt for small base thresholds

Summary

- Will always be delay
  - From network or OS or …
- Need to adjust queue latency
  - QM-(600,2) is the best, QM-120 almost as good
- Queue monitoring can be effective
  - 35-40 ms delay, variation up to 200ms, even 80 ms when quiet
- Run 3 Best vs. E-policy
  - E: 140ms, .9 gaps/min
  - QM-(600,2): 68ms, 1.4 gaps/min
- Run 24 Best vs. I-policy
  - I: 93 ms, 15 gaps/min
  - QM-(600,2): 90 ms, 4 gaps/min
- QM is flexible, can be tuned to app or user

Future Work?

Future Work

- Compare against I-policy where threshold changes each talkspurt
- Compare using different metrics, say that combine latency and gaps or looks at distribution
  - PQ studies to measure tradeoffs
- Larger networks
- Combine with repair
- Other decay strategies for QM