Challenges to network heterogeneously brought by wireless

Katsushi Kobayashi
ikob@acm.org

Outline

- Exposing Network Properties Underlying Paths
  - i-Path: Network Transparency
  - End-point Adaptation
- Network architecture to support various Latency Requirements
  - Architecture
  - Scheduler
- Conclusion

- NICT/Japan Funded Project 2008-2011
  “i-Path: Network Transparency Project”

- NSF-NICT Joint Project 2010 - 2012
  “Accommodating Network Evolution and Heterogeneity by Improving Network Transparency” NSF #1032226

- Japanese personnel:
  Prof. Shigeki Goto, Waseda Univ.
  Katsushi Kobayashi, AIST
  Ichiro Murase, Waseda Univ., MRI
  Akihiro Shimoda, Waseda Univ.
  Dai Mochinaga, MRI

- USA personnel:
  Prof. Srinivasan Seshan, CMU
  Athula Balachandran, CMU
  Dongsu Han, CMU

Transparency inside Internet

- No visibility mechanism, i.e., end-to-end principle
  - Transport stack ”estimates” inside network.
    - Bandwidth difference from 54Kbps to 10Gbps.
    - Frequently changing network condition in mobile.
    - Corruption loss in wireless is not negligible.
  - P2P peer selection, CDN server selection.
    - Not to mitigate traffic demand for backbone.
  - Unable to inspect provisioned path
    - Just acknowledgement from control plane

Network paradigms: Who should be smart ?

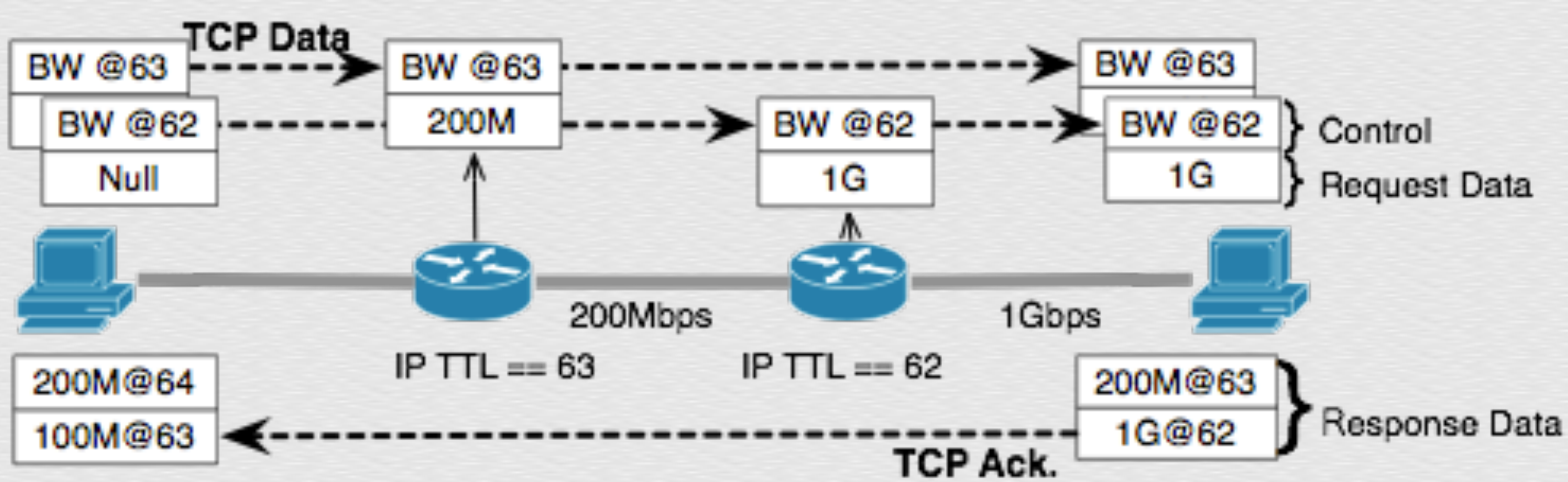
- Internet: Dumb minimal network with smart terminal, i.e., e2e.
- Telephone: Smart network with dumb terminal
- QoS provisioning: smarter network with Internet
  - end expects network to do something
- i-Path is another network paradigms, network exposes inside.
  - end asks what network can do
  - ➡ Router has rich information
    - 300K+ prefixes and its attributes, link BW, link utilization, corruption, congestion, MAC states, calendar clock, location, temperature, i/f description, CPU load, operator's contact, configuration, feature set....

i-Path: component

- Router
- End system
- Disclosure policy

i-Path: Router

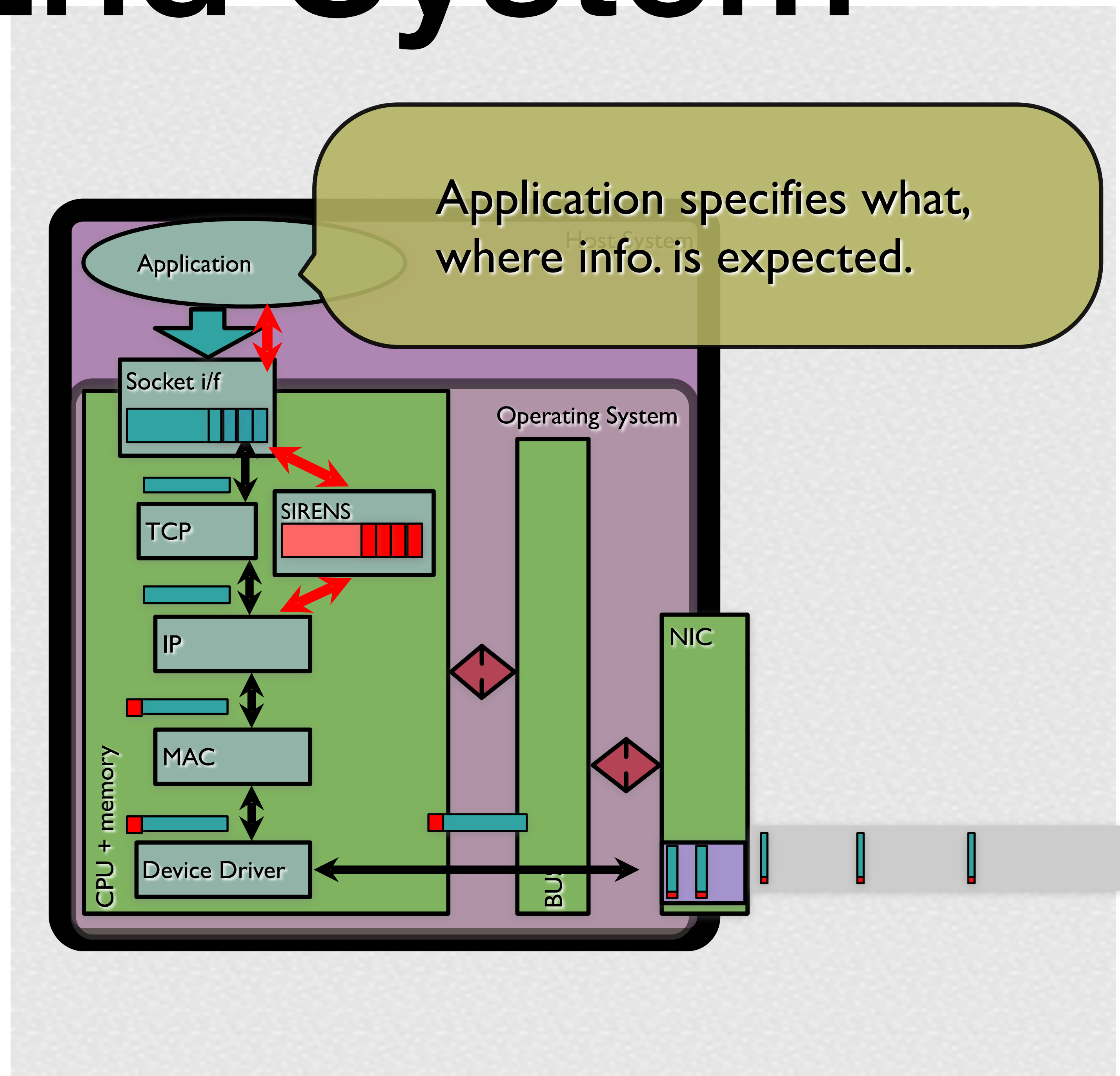
- Expose network information
- In-band cross-layer designed for transport
  - Jack up approach with shim layer middle of IP and transport
    - ETEN: focuses on satellite
    - PTP: Header growth with prepending data at each router hop
    - Congestion control with more network support
      - XCP, TCP-QS, RCP, ...



SIRENS: one packet collects one router's data

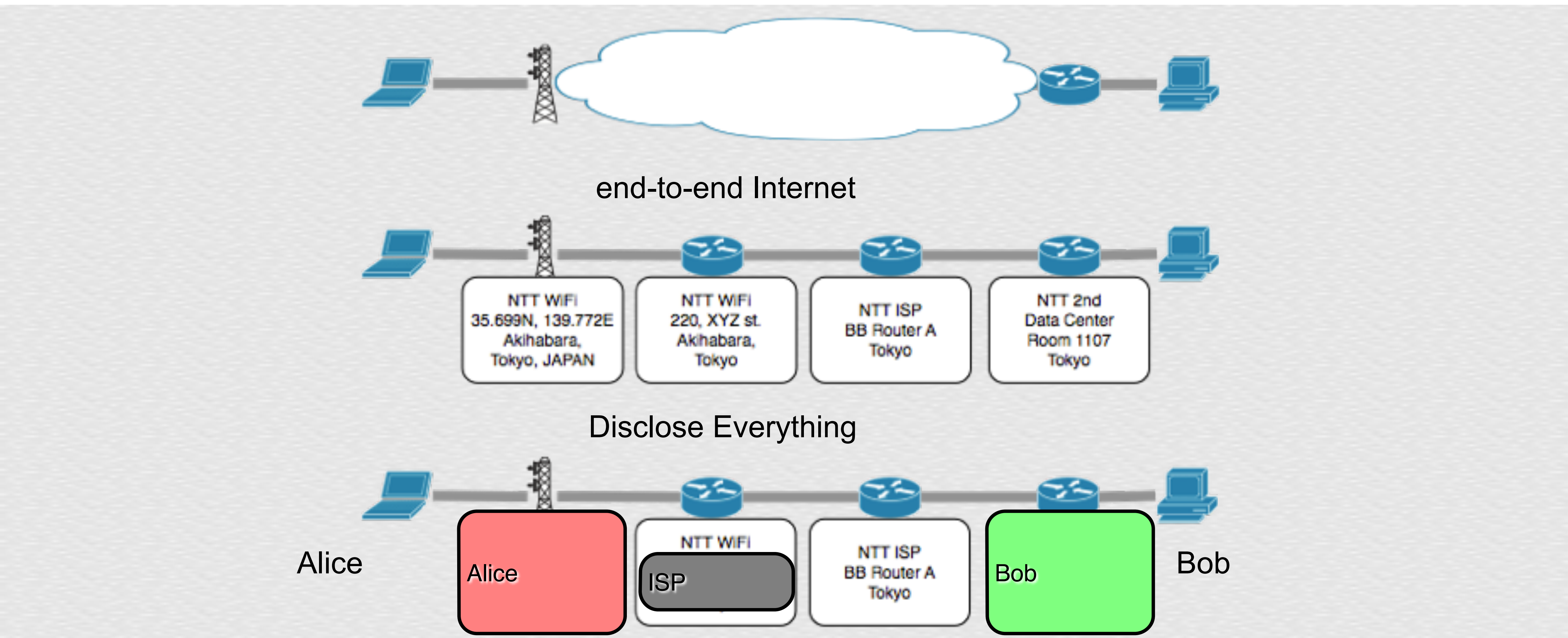
i-Path: End System

- Application accesses info with socket API:
  - `setsockopt()`, `getsockopt()`
- what: link BW, geolocation, BW utilization, length of queue, corrupted loss
- where: TTL range



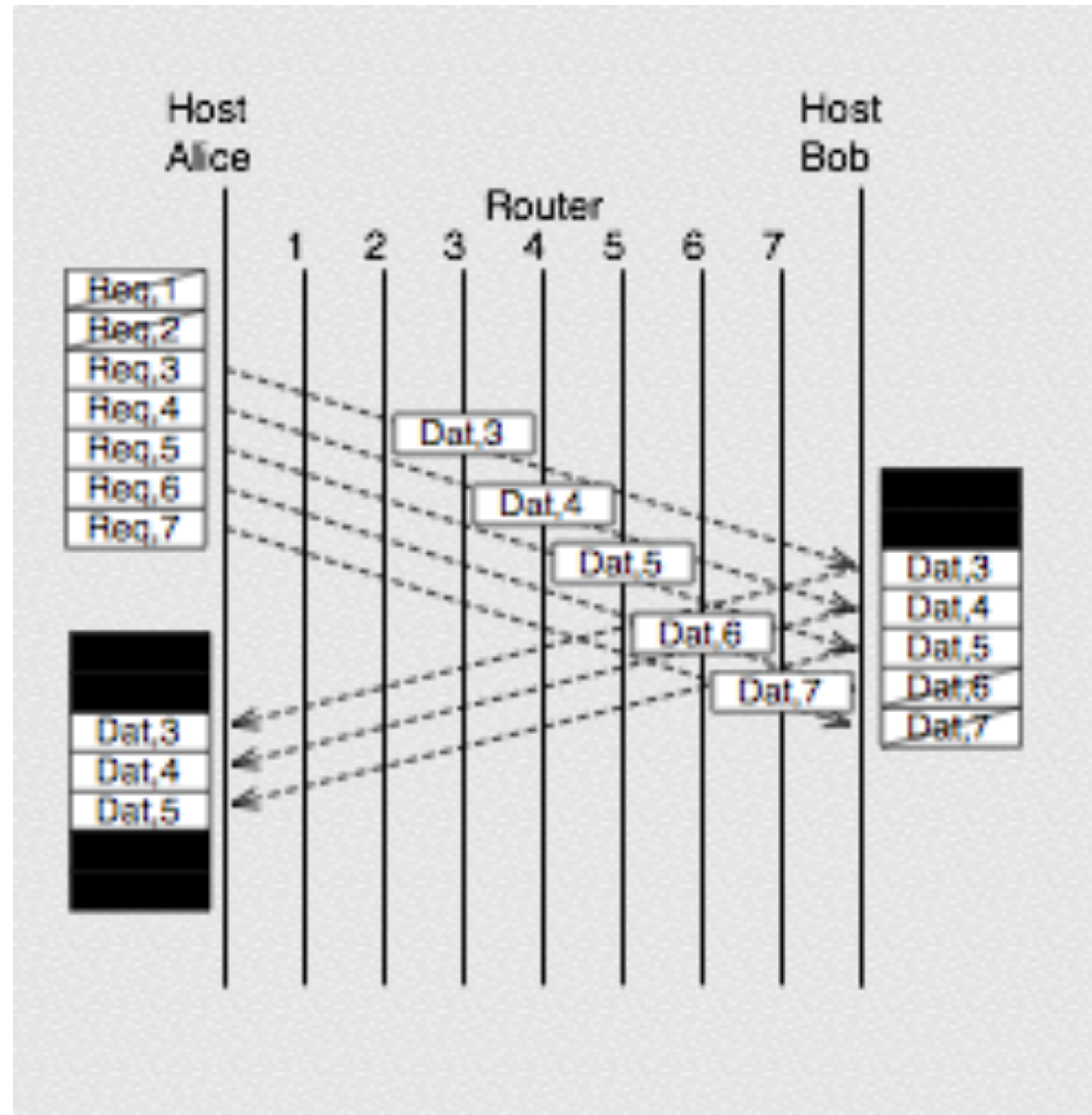
i-Path: Disclosure Policy

- Follow disclosure policies among ends, and ISPs., e.g.
  - User does not want to show privacy sensitive info. as geo. location
  - ISP does not want to show infrastructure info.



Selective Disclosure

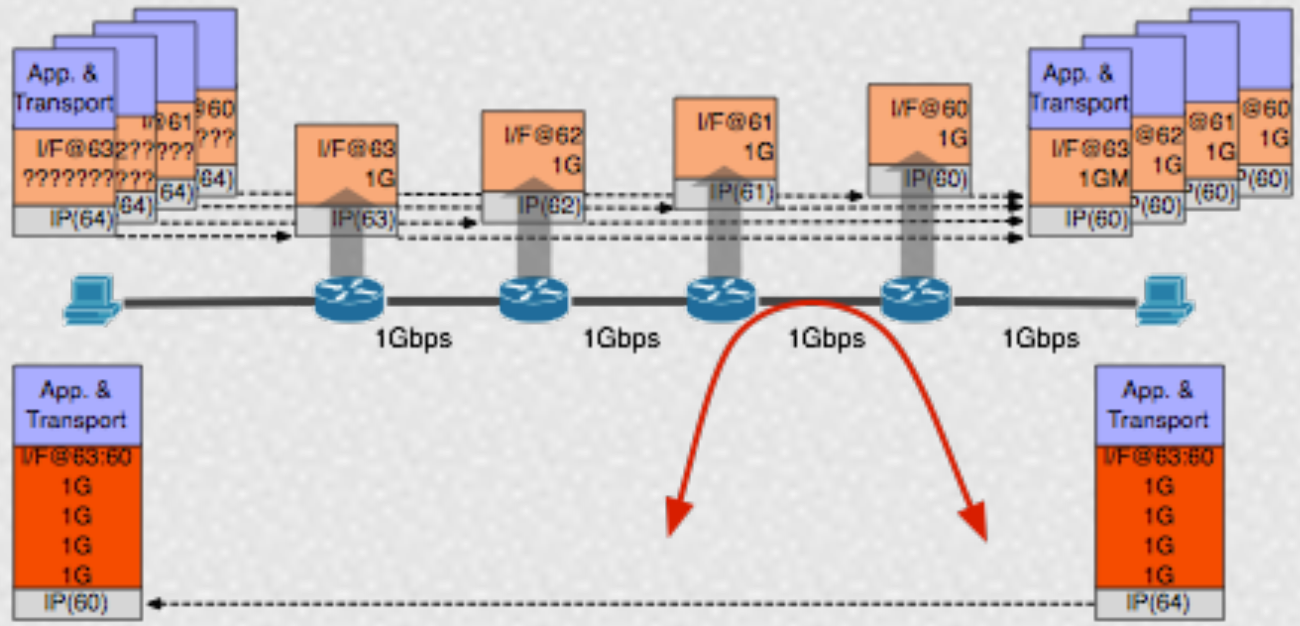
- Policy:
  - Alice & Bob allow to disclose beyond 3rd hop routers.
- Implementation:
  - Alice does not send req. for neighbor & next neighbor routers, i.e.,1st & 2nd hop.
  - Bob does not send back res. as Alice, i.e., 6th & 7th hop.
- Result:
  - Alice obtains 3-5 hops' data.
  - Bob obtains 3-7 hops' data

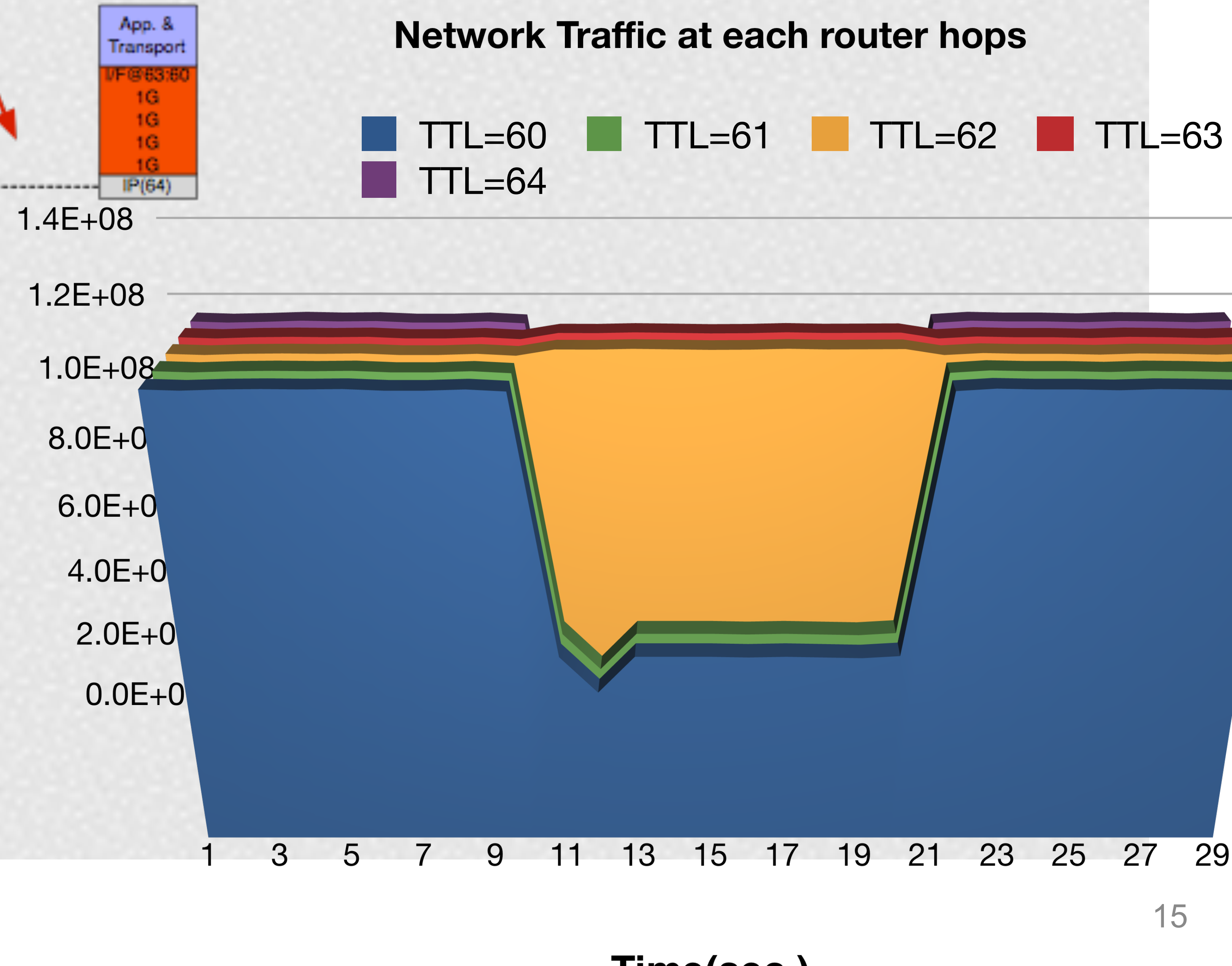


What i-Path brings for ?

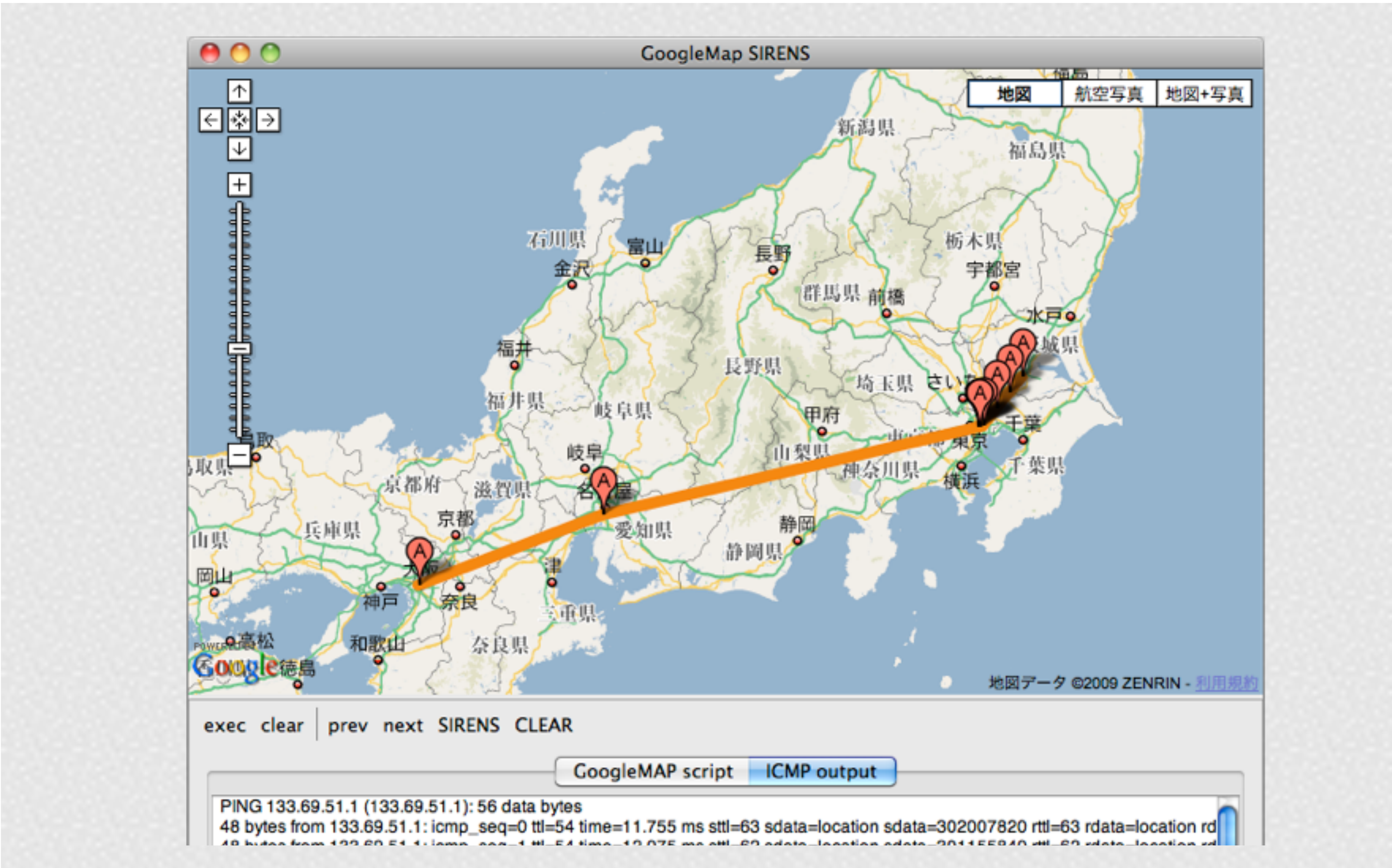
- Enhancing transport performance
- Optimal peer/server selection in P2P/CDN
- Offer optimal rate encoding in VoD service
- Better service with geographical location
- Region control in contents distribution using node location
- Input your ideas !

i-Path:visualize link capacities





i-Path: Geo-trace



i-Path: Current status

- Implementation:
  - i-Path router and end-system
    - FreeBSD/MacOS X/Linux(incl. Android)
    - Windows : postponed
    - http://i-path.goto.info.waseda.ac.jp/trac/i-Path/
  - Socket API C, C++, Python, Java (JNI)
- Deployment:
  - 6 routers in JGN2+, 12 in Lab.

Outline

- Exposing Network Properties Underlying Paths
  - i-Path:Network Transparency
  - End-point Adaptation
- Network architecture to support various Latency Requirements
  - Architecture
  - Scheduler
- Conclusion

TCP Variants

- TCP proposals for …
  - TCP for wireless / mobile
  - TCP for data centers
  - TCP for high delay-BW product paths
  - TCP for small latency
- Solutions either…
  - Assume that are working only in target environment
  - Are limited in their techniques because of generality/backward compatibility
- i-Path can provide information of underlying paths, TCP stack is able to optimize behavior not with estimations.

Streaming Video

- Rate adaptation
  - Need to choose encoding rate
  - Adapt quickly to changes in fair-share of network bandwidth
  - End-point based probing tends to be error-prone and high overhead
- Reliability
  - Not enough time to perform ARQ-based loss recovery
  - FEC-based schemes often incur high overhead
  - i-Path can help identify router support
- Scaling
  - Multicast and overlays are complex

Outline

- Exposing Network Properties Underlying Paths
  - i-Path
  - End-point Adaptation
- Network architecture to support various Latency Requirements
  - Architecture
  - Scheduler
- Conclusion

Latency requirements from applications

Dedicated / Provisioned

- HFT (High Frequency Trading)
  Competing < μs
- Data Center
  Tight job deadlines for interactive services
- VoIP
  150ms one-way delay (ITU-T G-114)

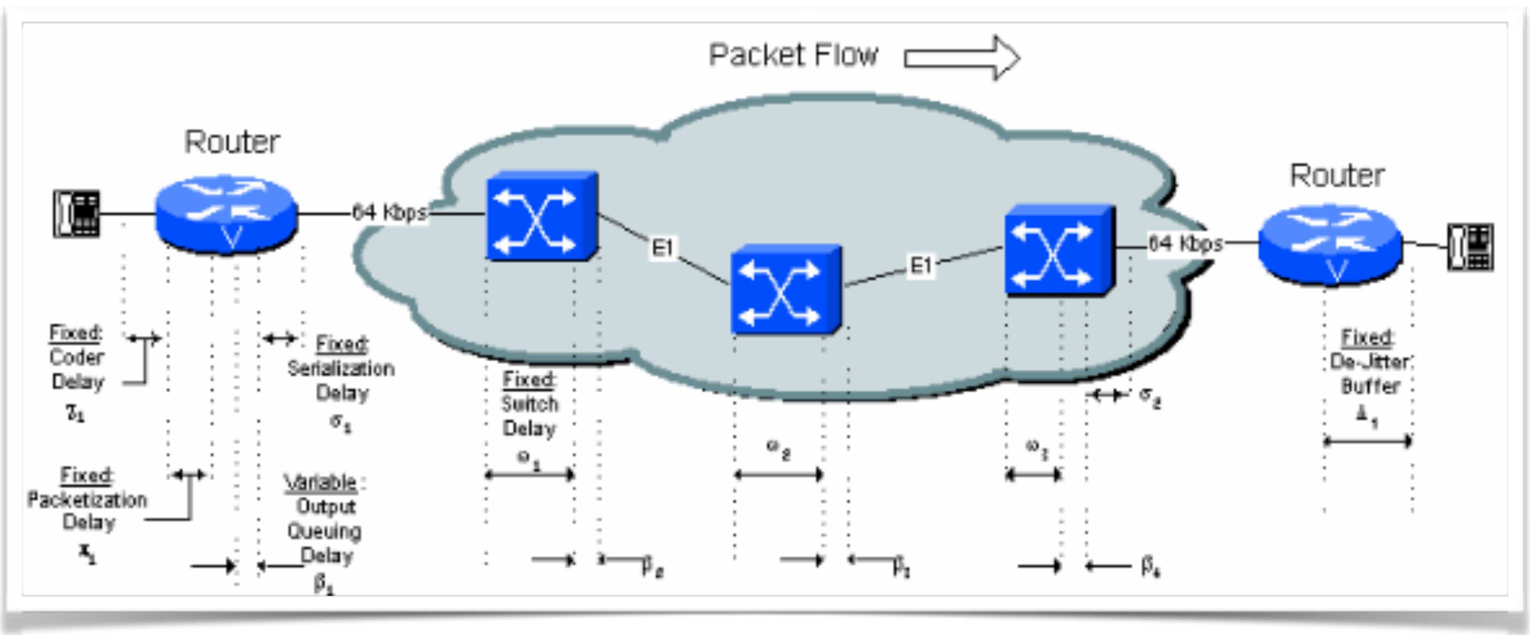
Best Effort

- e-Commerce
  < 4s rule for keeping customer attention
- On-line games
  > 50ms latency makes significant effects in FPS
- Interactive Web services

Latency in detail

Cause of latency in case of VoIP* :

1. Processing (Coder)
2. Packetization
3. Serialization
4. Queueing / Buffering
5. Network Switching (Propagation + Switching)
6. De-jitter



CISCO, Understanding Delay in Packet Voice Networks,
http://www.cisco.com/c/en/us/support/docs/voice/voice-quality/5125-delay-details.html

Simple best-effort

- Major and primary traffic on Internet

  Pros: Minimal technical and economic demands

  Cons: No-QoS, congestion control required, DDoS risk

- Can achieve rough "*flow-Rate fairness*" by TCP congestion control
  - Packet loss as a congestion signal
  - End systems reactions to packet losses

“*Flow-rate fairness*” and latency support simultaneously

- Should work :
  - with neither per-flow state (IntServ),
    more than one queue (DiffServ)
  - without admission control for inter-ISP deployment.
  - coexisting with existing best-effort
  - incremental deployability
  - with minimum modifications

LAWIN : Architecture and Protocol

- Applications specify per-packet latency-limit according their requirements
  - e.g. IP option, flow label
- Routers schedule packets with taking advantages of per-packet indications
  - Latency-aware scheduler to replace FCFS/DropTail

Outline

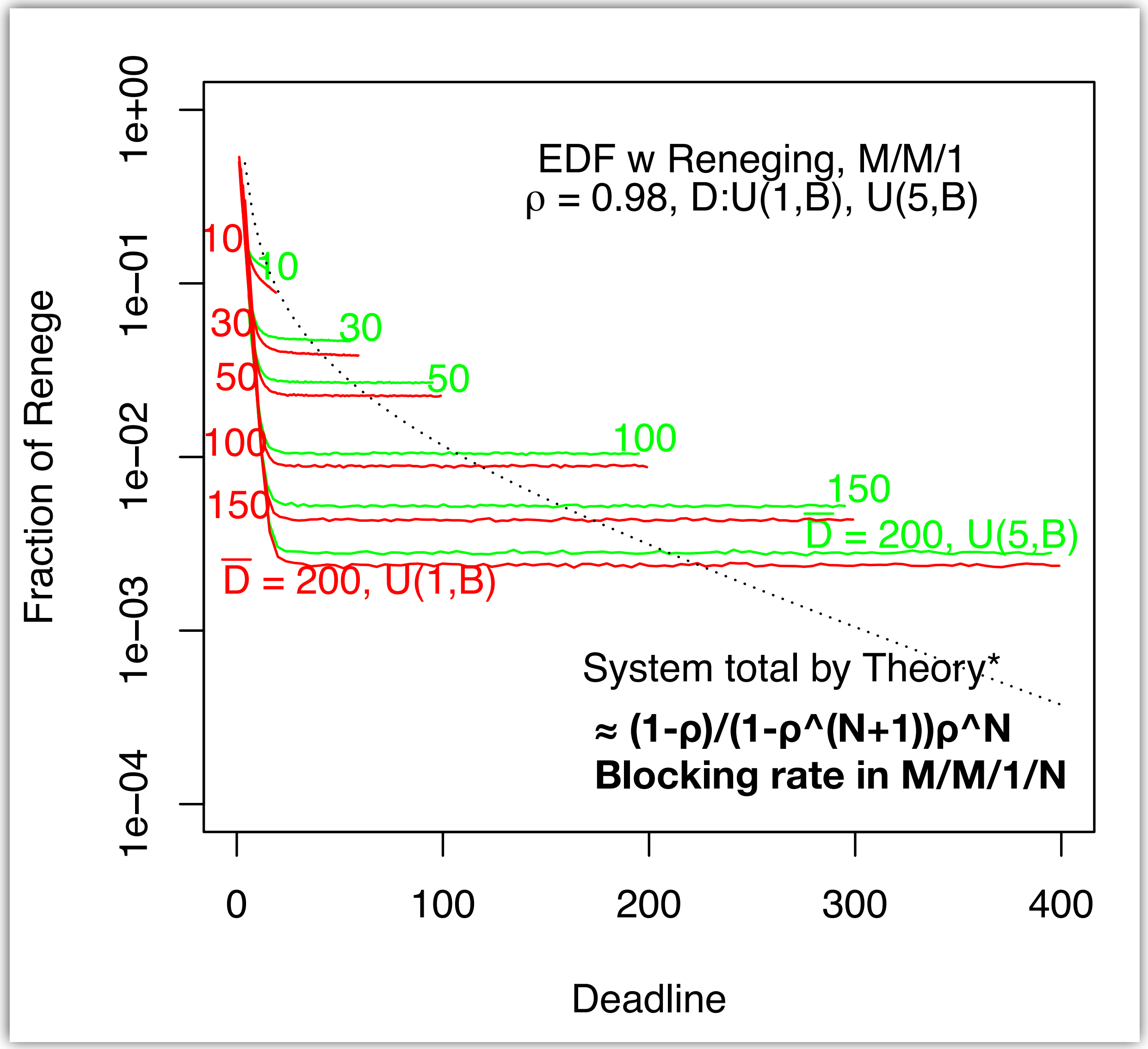
- Exposing Network Properties Underlying Paths
  - i-Path
  - End-point Adaptation
- Network architecture to support various Latency Requirements
  - Architecture
  - Scheduler
- Conclusion

Earliest Deadline First (EDF) with reneging

- EDFR is a latency aware scheduler better than simple EDF
    - EDF is blocked by elapsed data
    - EDFR drops packets if elapsed their deadlines
- Similar property to FCFS in terms of packet loss rate
    - EDFR imposes fair loss-rate to all flows regardless of their deadline requirements as FCFS
    - Loss-rates rely on average deadline, and is same as corresponding capacity's FCFS

Loss property of EDFR scheduler

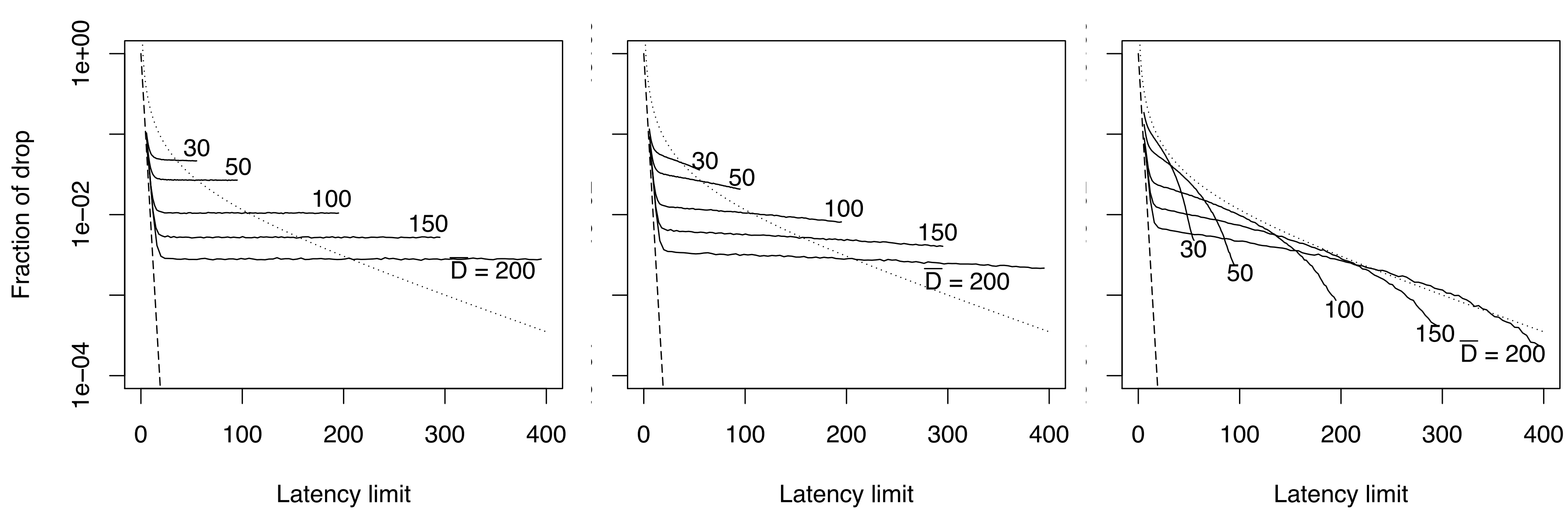
- Flat Bottoms : similar to FCFS
  - Fair loss-rate to all flows regardless of their deadline requirements as FCFS
  - Loss-rates rely on average deadline, and is same as corresponding capacity's FCFS
- Steep Cliffs : can be avoidable
  - Corresponds block rates when a new arrival deadline is too small



*L. Kruk,J. Lehoczky,K. Ramanan,and S. Shreve. "Heavy traffic analysis for EDF queues with reneging", The Annals of Applied Probability, 21(2):484–545, 2011

EDF with Reneging Later arrivals (EDFRL) - Preliminary -

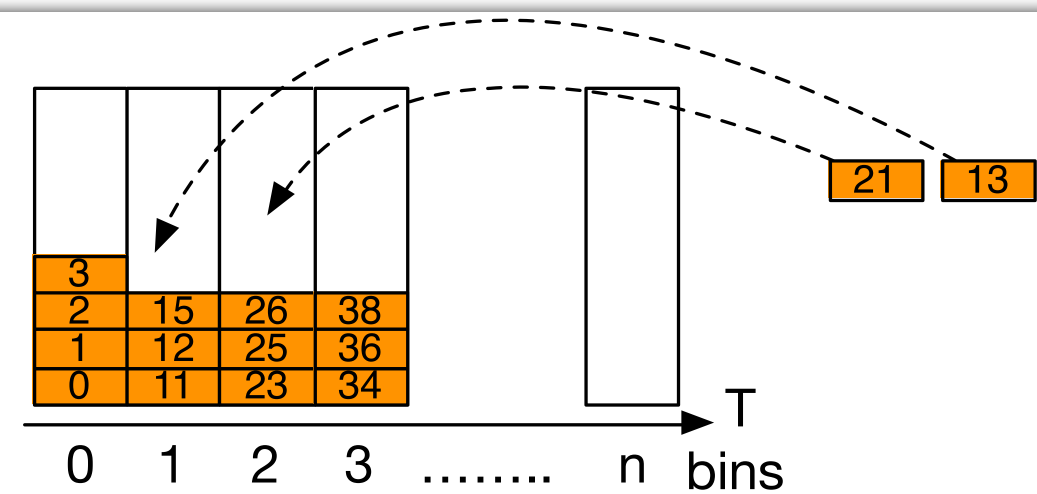
- EDFRL provides loss-rate bias with deadline
  - On best-effort networks :
    To impose higher loss to shorter DL; incentive for applications to choose longer
  - Can be realized just to replace priority queue sub-scheduler in EDFR by FCFS
  - On managed networks :
    To provide higher priority to shorter deadlines replacing FCFS with LCFS



(A) EDFR (B) EDFRL(slot width = 1) (C) EDFRL(slot width = 10)

Calendar Queue*

- O(1) scheduler, if dequeue from only top of bin
  - Event Simulator (NS2)
  - Packet pacing / Traffic shaper



Calendar Queue

Outline

- Exposing Network Properties Underlying Paths
  - i-Path
  - End-point Adaptation
- Network architecture to support various Latency Requirements
  - Architecture
  - Scheduler
- Conclusion

Conclusion

- Exposing Network Properties Underlying Paths
- Network architecture to support various Latency Requirements