15-441 Computer Networking

Lecture 27 – Research Directions

Outline

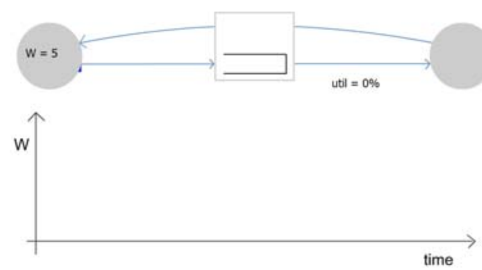
- Transport
- Wireless
- Probing

Areas

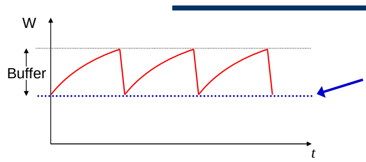
- Router interactions
  - FQ, RED → Blue, CHOKe, CSFQ, XCP…
  - **Small buffer routers**
- New congestion control designs
  - Delay based (Vegas)
  - Long-term TCP fair (TFRC)
  - Others: bionomial, BIC/CUBIC
- Other issues
  - Large bandwidth-delay product networks
  - Delay Tolerant Networks (**DTN**)
  - Congestion control outside TCP (congestion controlled UDP, general congestion management)

Single TCP Flow

Router with large enough buffers for full link utilization



Summary Buffered Link

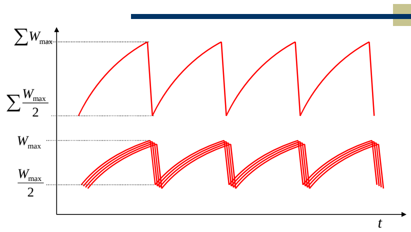


- With sufficient buffering we achieve full link utilization
  - The window is always above the critical threshold
  - Buffer absorbs changes in window size
    - Buffer Size = Height of TCP Sawtooth
    - Minimum buffer size needed is 2T*C
  - This is the origin of the rule-of-thumb

Example

- 10Gb/s linecard
  - Requires 300Mbytes of buffering.
  - Read and write 40 byte packet every 32ns.
- Memory technologies
  - DRAM: require 4 devices, but too slow.
  - SRAM: require 80 devices, 1kW, $2000.
- Problem gets harder at 40Gb/s
  - Hence RLDRAM, FCRAM, etc.
- Rule-of-thumb makes sense for one flow
  - Typical backbone link has > 20,000 flows
  - Does the rule-of-thumb still hold?

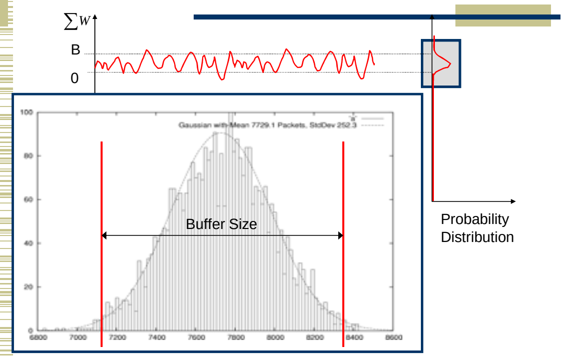
If flows are synchronized



- Aggregate window has same dynamics
- Therefore buffer occupancy has same dynamics
- Rule-of-thumb still holds.

7

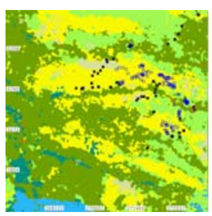
If flows are not synchronized



8

Delay Tolerant Networks: Motivation

- Rural area (buses, mail trucks, infostations)
- Mobile routers w/disconnection (e.g., ZebraNet)
- Sensor networks (e.g., Data mules)
- Deep space
- Underwater
- …



Internet vs. DTN

Unstated Internet assumptions

- Exist some end-to-end paths
- End-to-end RTT is low
  - At most a few seconds, and typically less than 500 ms
- Use retransmission for reliability
- Packet switching is the right abstraction

DTN characteristics

- Contact connectivity is intermittent and hard to predict
  - May not exist e2e paths
- Large delay (can be hours or even days!)
- High link error and low capacity
  - Resource budget can limit transmissions
- Different network architectures (e.g., TCP/IP won't work)

DTN Research Issues

- Naming, addressing, location management
- Routing on dynamic graphs
- Scheduling
- Security
- Applications
- Practical deployment issues
- …

Routing on Dynamic Graphs

- DTN routing runs over a time-varying topology
  - Links come and go, sometimes predictably
- Inputs
  - Time varying topologies (S, D, c(t), d(t))
  - Traffic demands, vertex buffer limits, mobility patterns
- Goal: determine route and schedule to optimize some metric
  - E.g., delay, throughput, resource consumption
- Consider two scenarios:
  - Nodes move randomly
  - Node movement is predictable

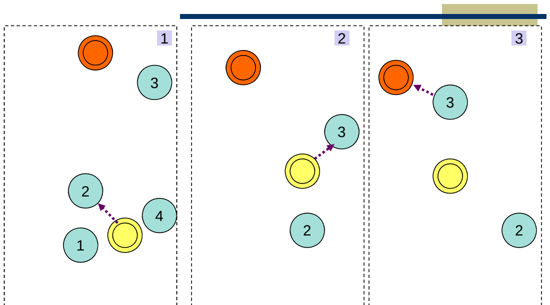
Two Extremes: Flooding versus Direct Contact

- Flooding: each node forwards any non duplicated message to any other node it encounters
  - Pros: low delay
  - Cons: high transmission overhead
- Direct contact: the source holds the data until it comes in contact with the destination
  - Pros: minimal resources
  - Cons: long delay
- Can we do better?
  - Some degree of replication (obvious)
  - Replication + erasure codes

Replication

- Source sends r identical copies over the first r contacts
- Relay nodes directly send to the destination
- Improve by using history information:
  - Each node keeps track of the probability a given node will be able to deliver its message
  - It replicates to r highest ranked relays based on delivery probability

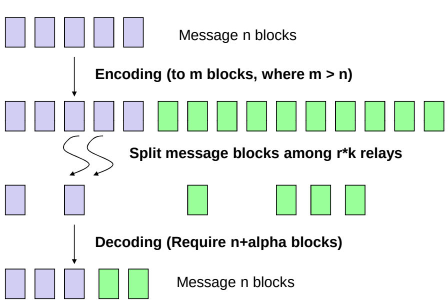
Illustration



Erasure Codes

- Rather than seeking particularly “good” contacts, we “split” messages and distribute to more contacts to increase chance of delivery
  - Same number of bytes flowing in the network, now in the form of coded blocks
  - Partial data arrival can be used to reconstruct the original message
    - Given a replication factor of ***r***, (in theory) any ***1/r*** code blocks received can be used to reconstruct original data
- Potentially leverage more contacts opportunity ➔ reduce worse-case latency
  - Reduces “risk” due to outlier bad contacts

Erasure Codes



Summary: Forwarding Algorithms

Algorithm	Who	When	To whom
Flood	All nodes	New contact	All new
Direct	Source only	Destination	Destination
Simple Replication(r)	Source only	New contact	*r* first contacts
History (r)	All nodes	New contact	*r* highest ranked
Erasure Coding (ec-r)	Source only	New contact	*kr (k>=1)* first contacts (k is related to coding algorithm)

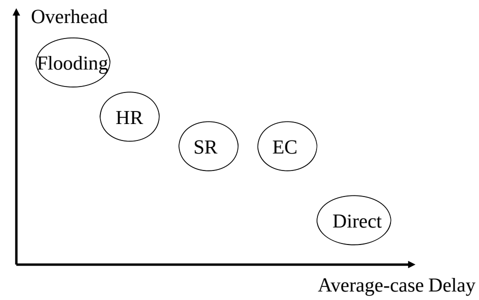
Evaluation Methodology

- Use a real-world mobility trace collected from the initial ZebraNet test deployment in Kenya, Africa, July, 2004



- Only one node returned 32-hour uninterrupted movement data
  - Weather and waterproofing issues
- Semi-synthetic group model
  - Statistics of turning angles and walking distance

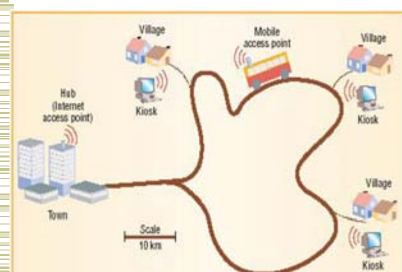
Summary



Big Picture

- Goal: bring Internet connectivity to rural areas
- Approach: rural kiosks
  - 150,000+ operational in India
  - Ministry of Info. Tech. plans to set up 100,000 more in next two years
- Kiosks connectivity
  - Dial-up
    - slow (28 kbps)
    - flaky (due to harsh environment)
  - Very Small Aperture Terminal
    - expensive (satellite link)
    - spare parts are hard to get
  - Long range WiFi
    - still experimental
    - expensive up front cost (for 18m tower)

Mechanical Backhaul*



Picture from Daknet project

A bus carrying a 802.11 access point (Daknet project)

*Term suggested by A.A. Penzias

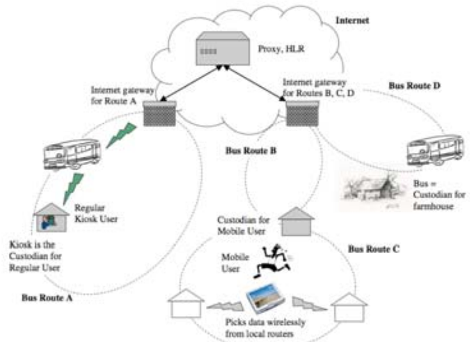
Challenges

- Both ends of a ‘connection’ are not simultaneously present
  - Can’t use standard TCP/IP, DNS, SSL
- Mostly disconnected, rarely connected
  - Opposite of usual assumptions
    - for example, made by Mobile IP, HIP, I3, PCMP etc.
- Low cost, high reliability, and secure
  - Need to share resources without compromising integrity

Design Principles

- Lower costs by sharing
- Store and forward self-describing data
- Separate locationing and addressing
- Use all links, opportunistically, if necessary
- Separate data and control plane
- Proxies for legacy support
- Replication for reliability

Architecture overview

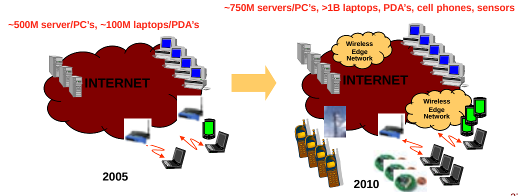


Outline

- Transport
- Wireless
- Probing

Common Case in the Future Internet

- Historic shift from PC's to mobile computing and embedded devices…
  - >2B cell phones vs. 500M Internet-connected PC's in 2005
  - >400M cell phones with Internet capability, rising rapidly
  - Sensor deployment just starting, but some estimates ~5-10B units by 2015



Trends: Density & Management

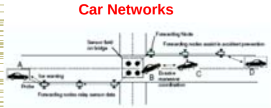
- Densities of unlicensed devices already high
  - Limits performance due to scarce spectrum
  - Need more spectrum or more efficient use
- Channel allocation suggests poor management
  - Makes problems such as interference worse
  - Security management probably worse
  - Won't get better → need automation

	#APs	Max degree
Portland	8683	54
San Diego	7934	76
San Fran	3037	85
Boston	2551	39

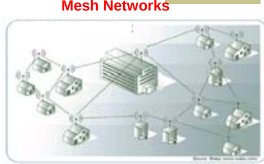
Channel	%
6	51
11	21
1	14
10	4

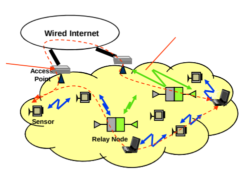
Trends: Growing Application Diversity

Collision Avoidance: Car Networks



Mesh Networks





Ad-Hoc/Sensor Networks

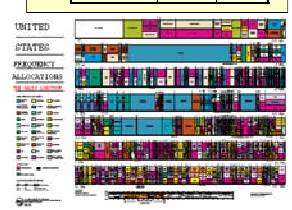
Wireless Home Multimedia

Trends: Spectrum Scarcity

- Densities of unlicensed devices already high

	#APs	Max @ 1 spot
Portland	8683	54
San Diego	7934	76
San Fran	3037	85
Boston	2551	39

- Spectrum is scarce → will get worse
  - Improve spectrum utilization (currently 10%)



New Directions

- Spectrum is scarce → will get worse
  - Improve spectrum utilization (currently 10%)
  - How can we manage spectrum better?
    - Licensed versus unlicensed, incentives, …
    - Dynamic re-use of spectrum?
- Increasingly diverse applications → MAC diversity?
  - Is 802.11 MAC holding us back?
  - How are wireless networks really used and what is a reasonable MAC design?
- Ubiquitous wireless networks → who will configure manage and optimize the network?
  - Networks need to become self managing
  - Needs to work across diverse technologies

31

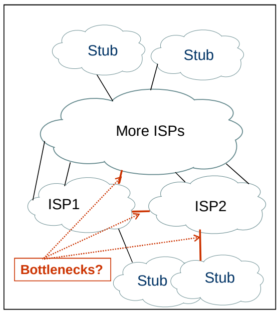
Outline

- Transport
- Wireless
- Probing

32

Internet Bottlenecks

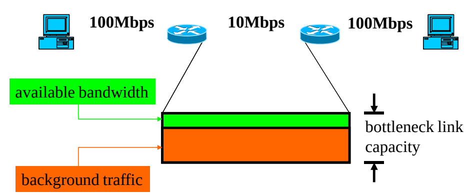
- Where are bottlenecks in the Internet?
  - Ignoring access links
- What is the capacity of these bottlenecks?
- Why do we want to know?
  - Picking a server
  - Multi-homing picking a network
  - Network diagnostic
  - Overlay network optimization



33

Available Bandwidth

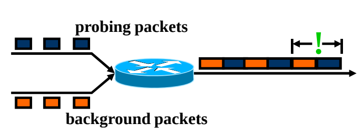
- The residual bandwidth on the bottleneck link



Measurement Techniques

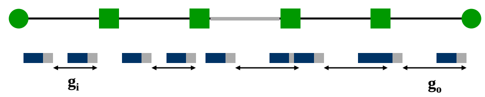
- Passive measurement: record the statistics of ongoing network transmission.
  - SNMP, NetFlow
  - SPAND
- Active measurement: inject probing packets into the network.
  - Traceroute, ping
  - Bprobe, nettimer, pathchar, cprobe, pathload
    http://www.caida.org/tools/taxonomy/performance.xml

Packet Pair Probing: Intuition



- Gap between the two packets of a packet pair can change as it travels over a link
  - Interleaving with competing traffic
  - Effect of low-capacity paths
- Packet trains similarly mingle with the background traffic

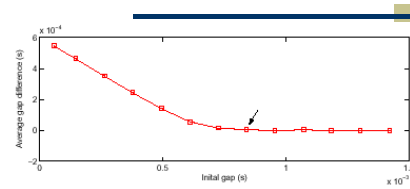
Packet Pair Probing for Available Bandwidth



- Effects on the bottleneck link dominate the packet gap changes along the path.
  - Other links do not affect the gap much
- Comparing the gap at the source and destination may yield information about the bottleneck and thus the end-to-end available bandwidth.

But it is not that simple ..

A Simple Experiment



- Send sequence of trains with increasing gap and record average output gap
- Turning point: packet train exactly fills the unused bandwidth on the bottleneck link
  - Faster: flooding the link
  - Slower: leaving unused capacity
- Analytical model backs up this intuition

Estimating the Available Bandwidth

1. Find the turning point by probing with trains with different input gaps until input = output
2. Estimate the available bandwidth:

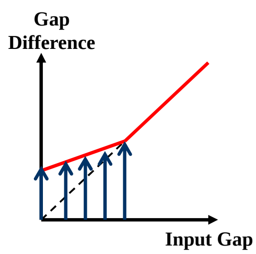
   By using the packet train rate (PTR)

   $B_a = \Sigma p_b \ / \Sigma g_o$

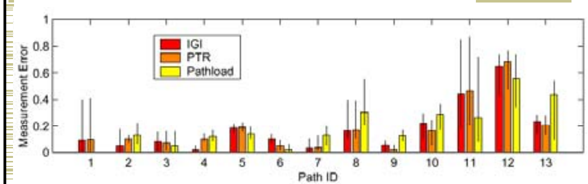
   \- or -

   By estimating the competing bandwidth observed by different pairs in the train (IGI)

   $B_a = B_o - (\Sigma g_o^+ \ / \Sigma g_o) * B_o$



Accuracy



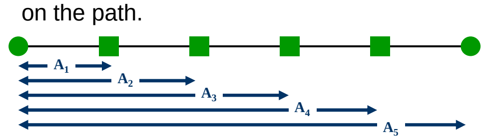
- Experiment on 13 Internet paths, with different network properties
- Refer to our paper for detail properties of the Internet paths

Measurement Time

Path ID	*IGI/PTR* (s) (5%, **median**, 95%)	Pathload (s) (5%, **median**, 95%)	Ratio($\frac{Pathload}{IGI/PTR}$) median
1	(1.60, **2.05**, 6.27)	(14.98, **30.56**, 31.03)	13.22
2	(0.58, **0.73**, 1.56)	(13.67, **15.37**, 31.81)	20.86
3	(0.11, **0.11**, 0.18)	(7.55, **13.17**, 14.91)	99.78
4	(0.49, **0.52**, 0.52)	(11.78, **12.26**, 12.76)	23.48
5	(0.78, **0.80**, 0.83)	(15.58, **15.86**, 16.55)	19.75
6	(0.62, **0.80**, 1.20)	(49.07, **56.18**, 62.24)	70.08
7	(0.51, **0.51**, 0.67)	(14.01, **22.40**, 28.51)	45.94
8	(1.01, **1.02**, 1.27)	(27.57, **31.51**, 47.62)	27.80
9	(0.24, **0.30**, 0.30)	(15.35, **16.14**, 27.66)	65.81
10	(1.27, **1.27**, 1.50)	(20.95, **21.04**, 21.77)	16.50
11	(1.03, **1.10**, 2.03)	(19.97, **25.78**, 38.52)	23.45
12	(2.17, **2.32**, 3.60)	(19.24, **21.54**, 42.00)	9.20
13	(1.10, **1.11**, 1.13)	(12.24, **12.76**, 47.22)	11.24
Geometric Mean			26.39

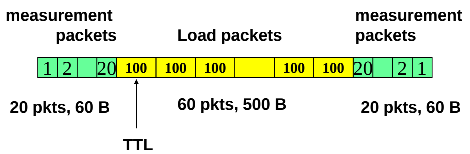
Yes, but Where Is the Bottleneck?

- Can we measure the available bandwidth on every link along the path?
- Requires us to measure the gap not only at the source and destination but at every router on the path.

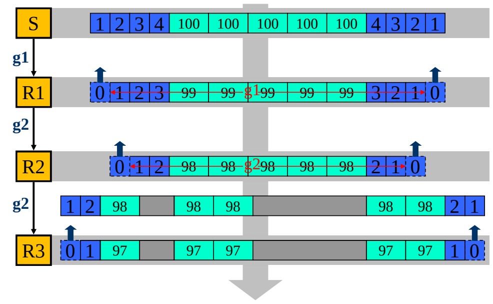


- Idea: combine a PTR train with measurement packets that probe the routers.

Recursive Packet Train (RPT)



Transmission of RPT

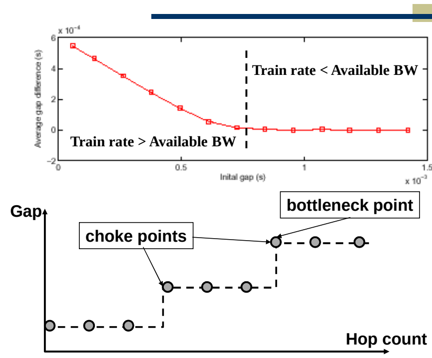


Example: 139.82.0.1

```
[root@azure src-all]# ./pathneck -f 139.82.0.1
1080697381.041055 139.82.0.1 500 60 0

00   0.773   2871   128.2.223.254    0 GIGROUTER.NET.CS.CMU.EDU
01   0.770   2744     128.2.0.12     0 CORE0-VL1000.GW.CMU.NET
02   4.886   1627    128.2.33.233    0 HYPER-VL502.GW.CMU.NET
03   1.135   2748  192.88.115.185    0 bar-cmu-ge-4-0-0-1.psc.net
04   1.008   2872   192.88.115.18    0 beast-bar-g4-0-1.psc.net
05  10.244  15738  192.88.115.124    0 abilene-psc.abilene.ucaid.edu
06  26.106   2995     198.32.8.65    0 atlang-washng.abilene.ucaid.edu
07  44.839   2871  198.32.252.253    0 abilene-oc3.ampath.net
08 149.269   8368  198.32.252.238    0 rnp.ampath.net
09 147.519   9865  200.143.254.94    0 rj-fast4_1.bb3.rnp.br
10 148.261   9740  200.143.254.77    0 200.143.254.77
11 157.876  83448    200.20.94.81    0 rederio-atm-puc.rederio.br
12 324.145  24989   200.20.91.250    0 200.20.91.250
13 298.034  51093    139.82.59.65    0 139.82.59.65
14 317.538  31082     139.82.59.3    0 b40.rdc.puc-rio.br
15 321.010  26731   139.82.183.60    0 mec-fw.mec.puc-rio.br
```

Detecting the Bottleneck



But Not So Fast

- Bursty traffic: can cause “random” changes in the train length.
- Bursty traffic on the reverse path can affect the measurement packets.
  - Adds more noise to the data
- ICNP packet generation time on routers can be long and variable.
  - Adds even more noise
- Transmit multiple train and combine/average results