On Effectively Exploiting Multiple Wireless Interfaces in Mobile Hosts

Cheng-Lin Tsao and Raghupathy Sivakumar
Georgia Institute of Technology

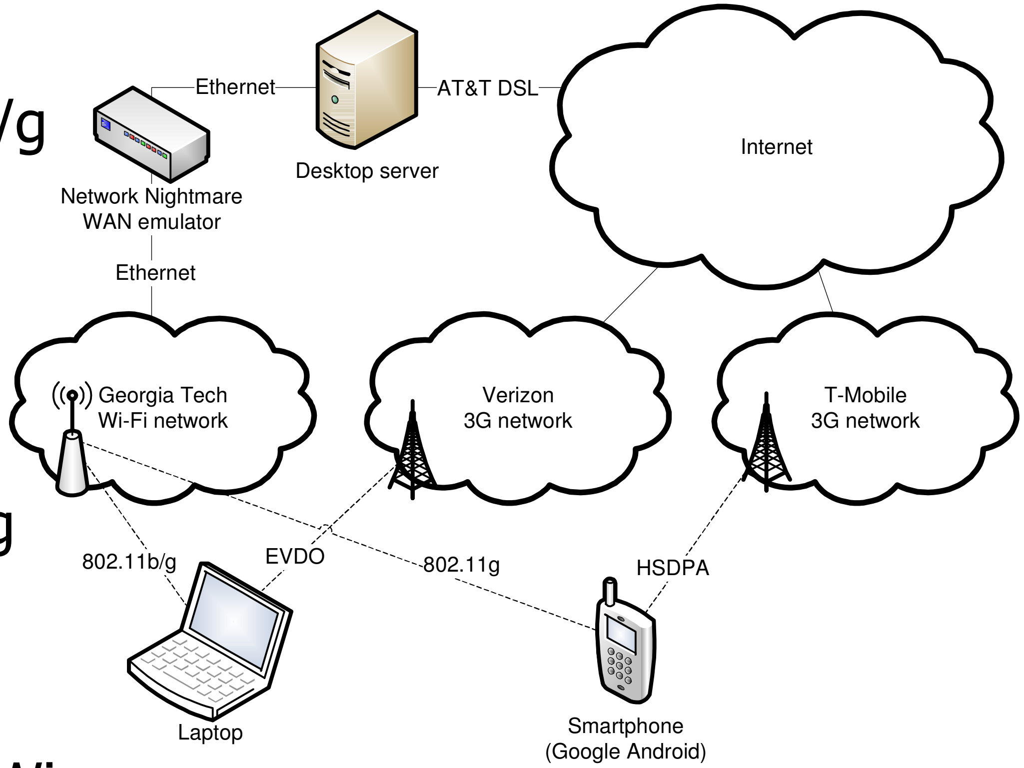
ACM CoNEXT '09
Dec 4, 2009
Rome, Italy

Introduction

- Multi-homed wireless devices
  - Laptops equipped with Wi-Fi, WWAN, Bluetooth, WiMAX
  - Smartphones equipped with Wi-Fi and 2.5G/3G
    - BlackBerry, iPhone, Google Android
  - Only one interface used at a time
    - Priority-based switching
- What is the best approach to leverage the multiple interfaces available at a mobile device in terms of user performance?

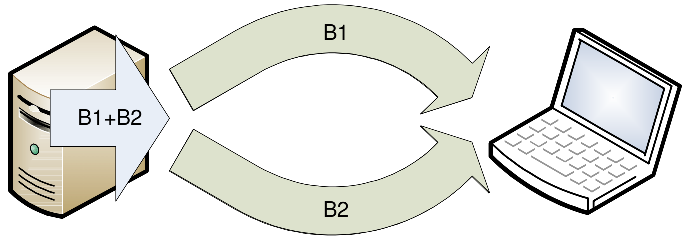
Experimental Testbed

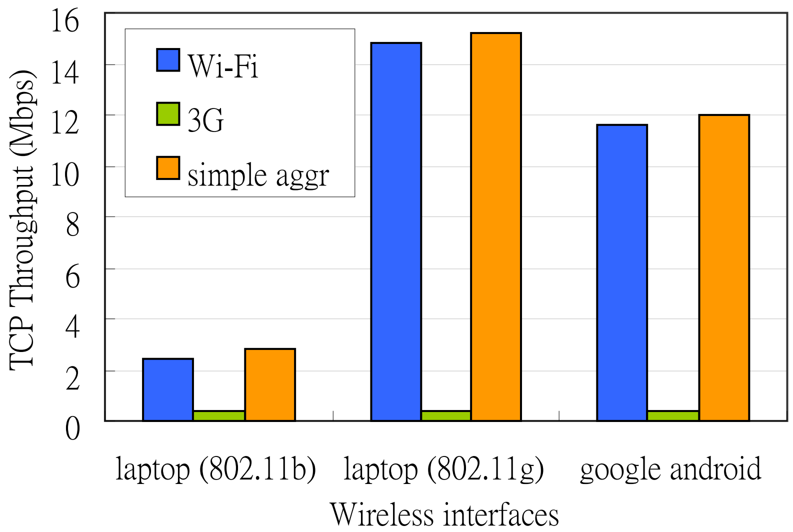
- Real-life heterogeneous wireless testbed
  - Laptop
    - Atheros 802.11a/b/g PCMCIA
    - Verizon USB727 EVDO
  - Google Android G1
    - Embedded 802.11g
    - T-Mobile HSDPA
  - Desktop server
    - WAN emulator on Wi-Fi to mimic Internet



Motivation

- Simple aggregation
  - Ex. pTCP [ICNP`02]
  - Sum of available BW
  - Marginal benefits
- Achieving performance better than the sum of the parts?
- Leveraging heterogeneity
  - Wi-Fi: high bandwidth
  - 3G: high availability, allocated resources





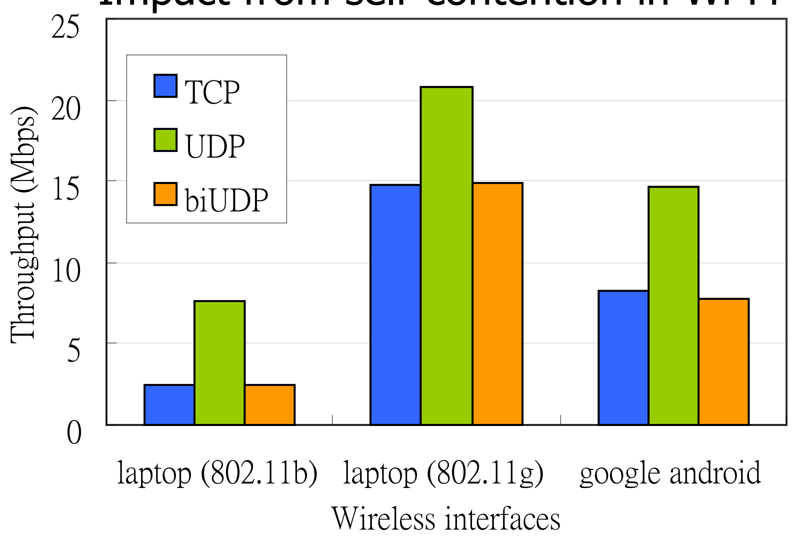
Super-Aggregation

- Concept: transferring data over multiple cooperating interfaces w/ high-layer knowledge
- Design
  - Three generic principles
  - TCP-specific throughput enhancement in Wi-Fi + 3G
  - Extension to other protocols/wireless interfaces
- Realization
  - Client software changes that work w/ legacy servers
  - Layer-3.5 TCP acceleration in Wi-Fi + 3G networks
  - Prototyped and evaluated on laptop and smartphone

Principle 1: Selective Offloading

- Concept
  - Selectively offload certain portions of the transferred data to the low-bandwidth interface
- Relation to TCP: self-contention
  - Between uplink ACK and downlink DATA
  - PHY/MAC overheads
  - Degrading throughput by 30%~70%
  - Verified with bidir. UDP
    - 1464B data and 32B ack

Impact from self-contention in Wi-Fi



Solution 1: Offloading-ACK

- Idea
  - Diverting uplink ACKs from Wi-Fi to 3G
- Challenges
  - Insufficient bandwidth on 3G
  - Long RTT on 3G degrades overall TCP throughput
- Solution
  - Fractional offloading
  - Opportunistic operation

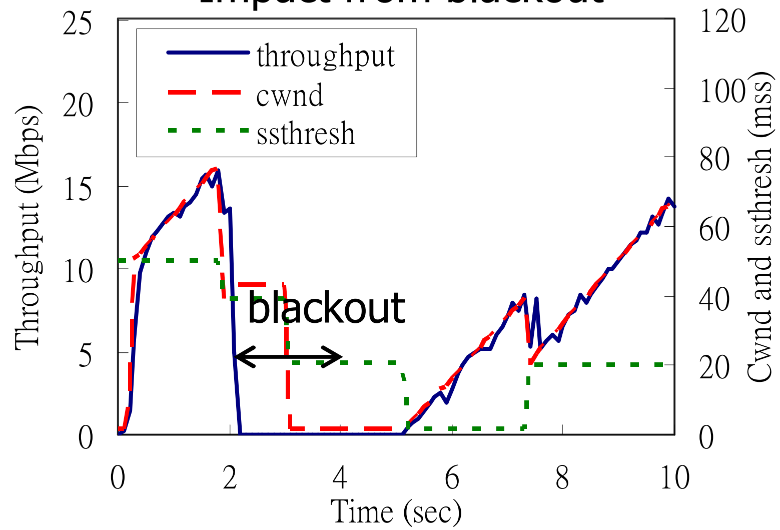
Offloading-ACK Details

- Fractional offloading
  - Offloading sustainable fractions on 3G
  - Discarding ACKs carrying no additional information
- Opportunistic operation
  - Offloading when RTT inflation has little impacts, like cwnd more than a threshold
  - Heuristic: ssthresh value of TCP
    - Default value: 20 mss

Principle 2: Proxying

- Concept
  - Use the low-bandwidth interface for critical control when the high-bandwidth one is temporarily down
- Relation to TCP: blackouts
  - Blackout: fading or handoff
    - Vehicle net: up to 75 sec[1]
  - Impacts
    - RTO timeout
    - Unnecessary idle
    - Slow slow-start

Impact from blackout



[1]V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden, "A measurement study of vehicular internet access using in situ wi-fi networks," Mobicom `06.

Solution 2: Proxying-blackout-freeze

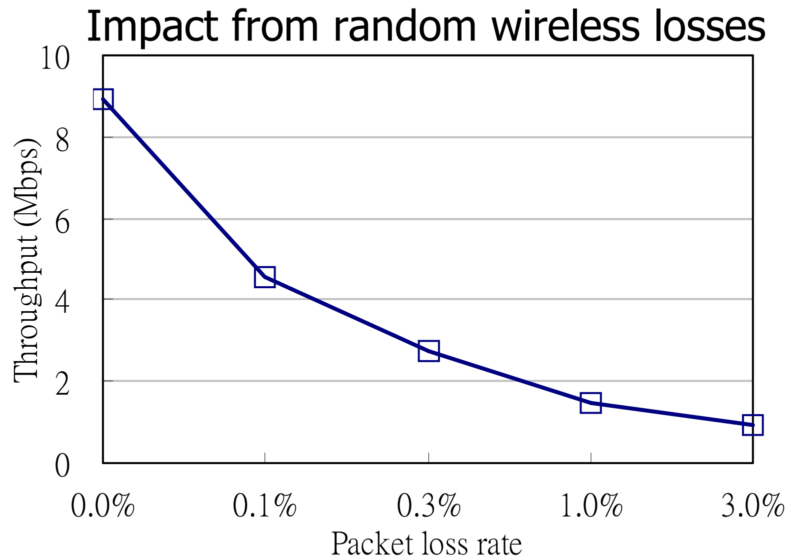
- Idea
  - Use the 3G link to notify the TCP sender about blackouts on Wi-Fi
- Challenges
  - Real-time blackout detection with low overhead
  - Freezing a TCP connection during blackout
- Solution
  - Hybrid blackout detection
  - Freezing TCP with flow control

Proxying-blackout-freeze Details

- Freezing TCP with flow control
  - Sending a zero-window advertisement on 3G to make TCP enter persist mode
  - Resuming the TCP connection w/ the original flow window via 3G
- Hybrid blackout detection
  - Passive monitoring of received packets/beacons
  - Active probing when no activity for a certain period
    - ICMP probing if no packet for more than 200 ms

Principle 3: Mirroring

- Concept
  - Intelligently mirror the certain portion of the transferred data on the low-bandwidth interface
- Relation to TCP: random losses
  - Caused by interference, fading, or long distance
  - Interpreted by TCP as congestion
  - 0.1% packet loss rate reduces TCP throughput by 49%



Solution 3: Mirroring-loss-fetching

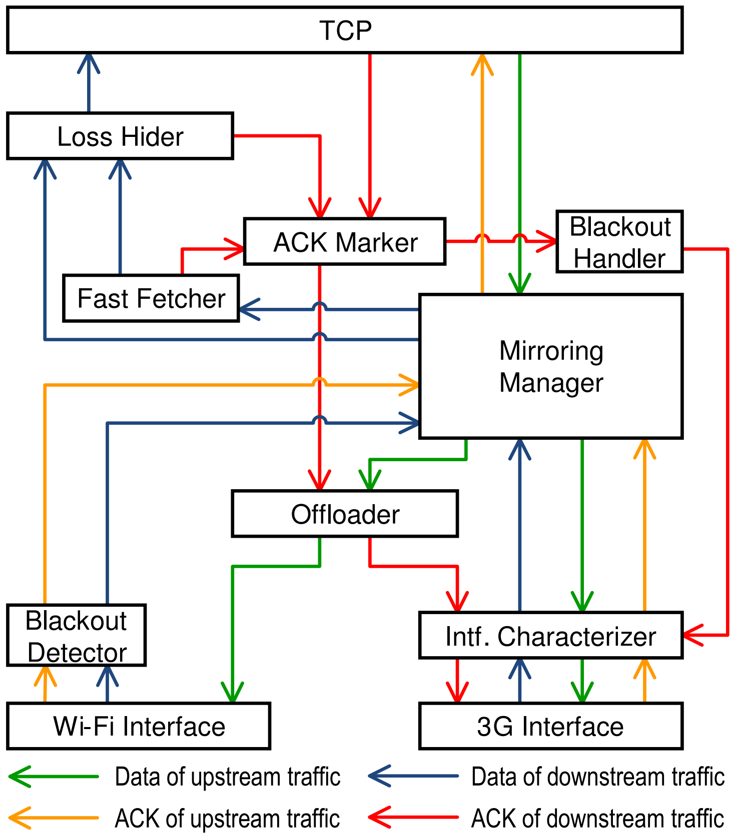
- Idea
  - Hide random losses in the original connection and fetch the lost packets in the mirror connection
- Challenges
  - Decoupling TCP congestion control and reliability
  - Mirroring the TCP connection on 3G
  - Efficiently fetching lost packets on 3G
- Solution
  - Loss distinction
  - Connection mirroring
  - Fast fetching

Mirroring-loss-fetching Details

- Loss distinction
  - Receiving corrupted frames indicates random losses
- TCP connection mirroring
  - Replay messages exchanged in the original connection
  - Offset TCP sequence numbers
  - Verify identical data received in the mirror connection
- Selected and fast fetching
  - Proactively acknowledge unneeded packets
  - Place a guard time before fetching the desired packets
    - 256 ms in prototype

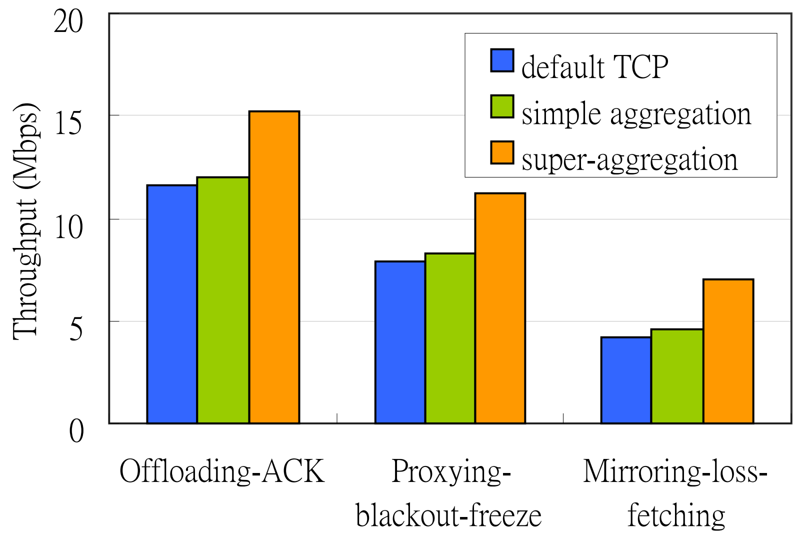
Super-Aggregation Architecture

- Software Architecture
  - Client-only changes
  - Layer-3.5 middleware
  - Transparency to TCP & link layers
- Integrated operations
  - Offloading-ACK
  - Proxying-blackout-freeze
  - Mirroring-loss-fetching



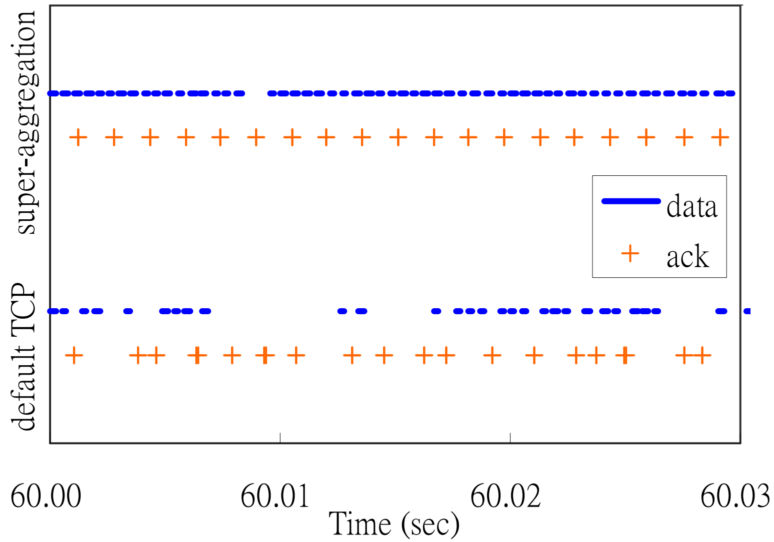
Performance Evaluation

- Performance metric
  - Overall throughput of bulk data transfer
  - Improvement over simple aggregation
- Improvement on Android
  - Offloading: 26%
  - Proxying: 35%
    - Blackouts (2 sec every 20 sec)
  - Mirroring: 52%
    - Random packet loss 0.3%



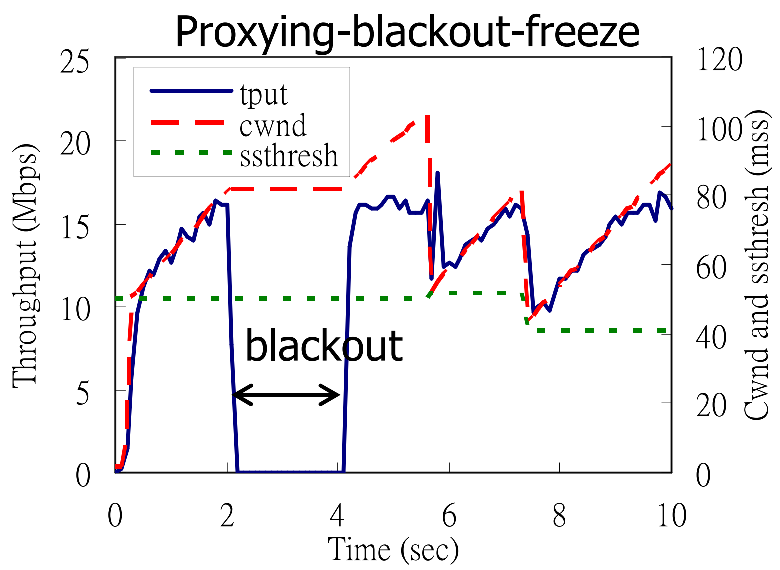
Offloading-ACK Analysis

- Avoiding TCP self-contention in Wi-Fi
  - Packets captured with tcpdump
  - Self-contention observed in default TCP
  - Offloading-ACK utilizes the Wi-Fi downlink



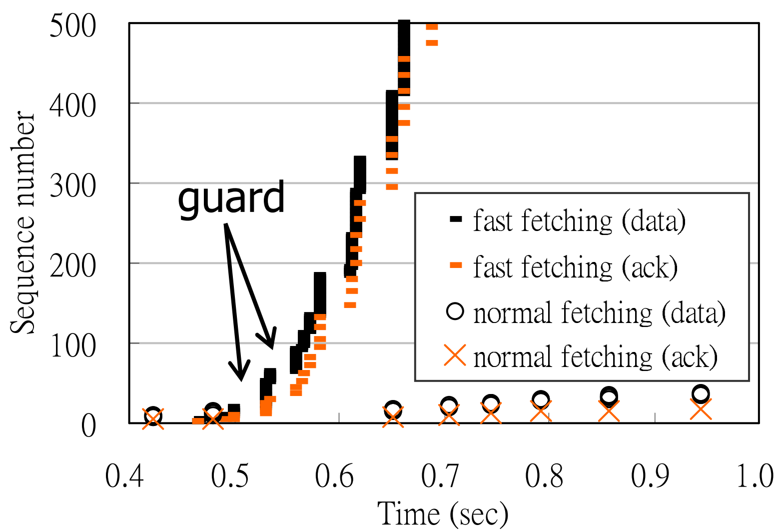
Proxying-blackout-freeze Analysis

- Minimizing the impact from blackouts
  - Same blackout period introduced
  - Avoiding slow start
  - Quick resumption after link recovery
  - Maintaining cwnd and ssthresh



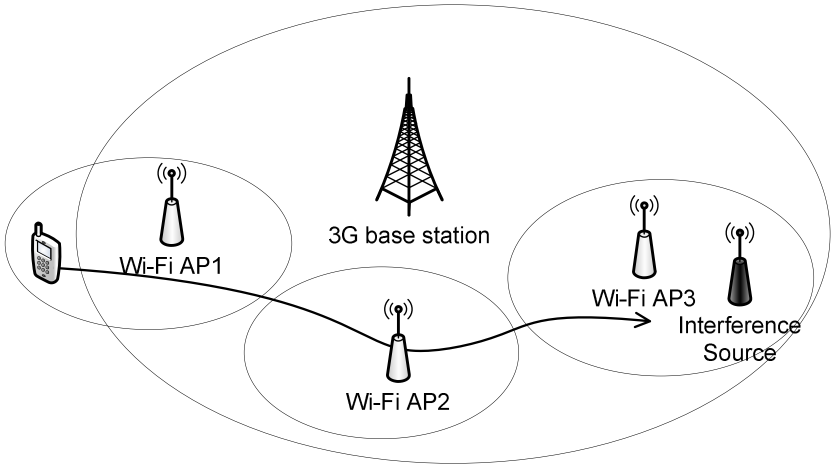
Mirroring-loss-fetching Analysis

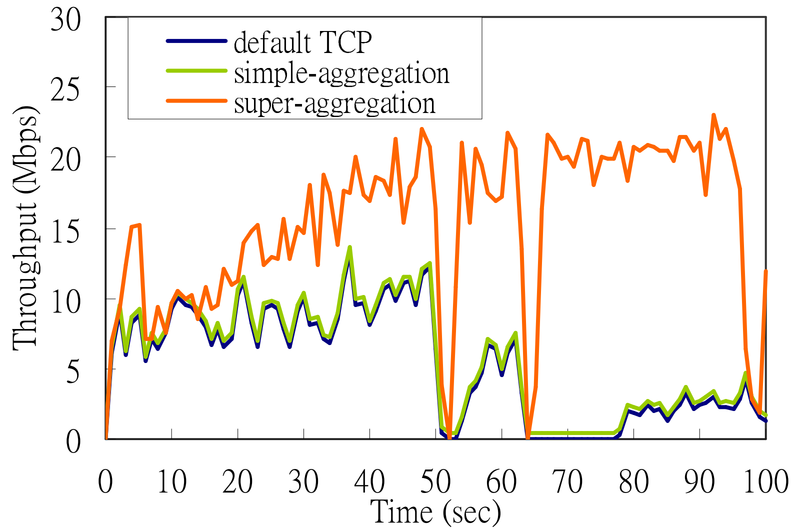
- Recovering lost packets efficiently
  - Recovering 393 packets out of 100k packets (0.3% loss on Wi-Fi)
  - Fast fetching recovers lost segments 36x faster
  - Guard time makes sure packet delivery on 3G



Integrated Operation Performance

- Integrated operations
  - Evaluated with a scenario with blackouts and random losses
    - Enter 3G when t=5
    - 2-sec blackouts when t=50 & t=62.5
    - 1% packet loss at AP3
  - Improving throughput by 169% in the scenario





Related Works

- Simple aggregation
  - pTCP [ICNP`02], WAMP [Globecom`99], RMTP [ICNP`01], $MC^2$ [ToMC`07], MAR [MobiSys`04]
  - Wireless specific: R^2CP [MobiCom`05] and PRISM [ToMC`07]
  - Requiring two-point deployment
- TCP enhancement over a single wireless network
  - Random losses: Snoop [MobiCom`95], WTCP [MobiCom`99]
  - Blackout: Freeze-TCP [Infocom`00]
- Multi-interface mechanisms for energy efficiency
  - CoolSpots [MobiSys`06], Cell2Notify [MobiSys`07], Context-for-Wireless [MobiCom`06]

Concluding Remarks

- Study super-aggregation of heterogeneous wireless interfaces
- Propose super-aggregation principles
  - Offloading-ACK
  - Proxying-blackout-freeze
  - Mirroring-loss-fetching
  - Generalization to rate-adaptive video streaming and more other wireless technologies
- Design and prototype the integrated architecture
- Evaluate on laptop/smartphone in testbed

Thank you!