SIP Overload Control
IETF Design Team Summary

Eric Noel
Carolyn Johnson
AT&T Labs

Introduction

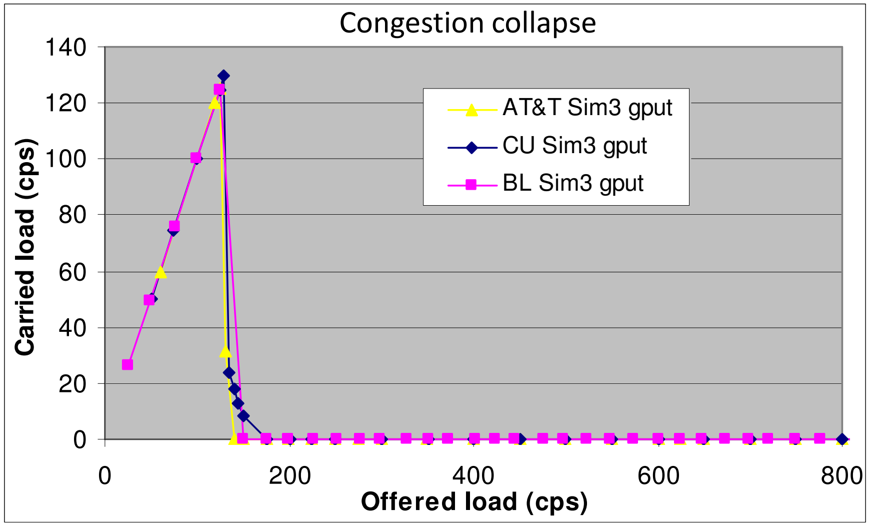
- The IETF has chartered a design team to perform research on SIP overload controls and lay the groundwork for possible standardization:
  - The design team was chaired by Alcatel-Lucent,
  - It included members from AT&T Labs, Bell Labs/Alcatel-Lucent, Columbia University, Sonus Networks, Nortel, British Telecom,
- The team has developed four independent simulation tools and has conducted extensive simulations:
  - Confirm the problem of the current SIP specification,
  - Evaluate potential solutions in steady-state and transient simulations,
- Close collaboration between AT&T labs (Eric Noel, Carolyn Johnson), Alcatel-Lucent (Volker Hilt, Fangzhe Chang), Columbia University (Charles Shen) and Sonus Networks (Ahmed Abdelal).

SIP Overload Control Design Team

- Team Members:
  - Eric Noel, Carolyn Johnson (AT&T Labs)
  - Volker Hilt, Fangzhe Chang (Bell Labs/Alcatel-Lucent)
  - Charles Shen, Henning Schulzrinne (Columbia University)
  - Ahmed Abdelal, Tom Phelan (Sonus Networks)
  - Mary Barnes (Nortel)
  - Jonathan Rosenberg (Cisco)
  - Nick Stewart (British Telecom)
- Four independent simulation tools:
  - AT&T Labs, Bell Labs/Alcatel-Lucent, Columbia University, Sonus Networks
- Bi-weekly conference calls.

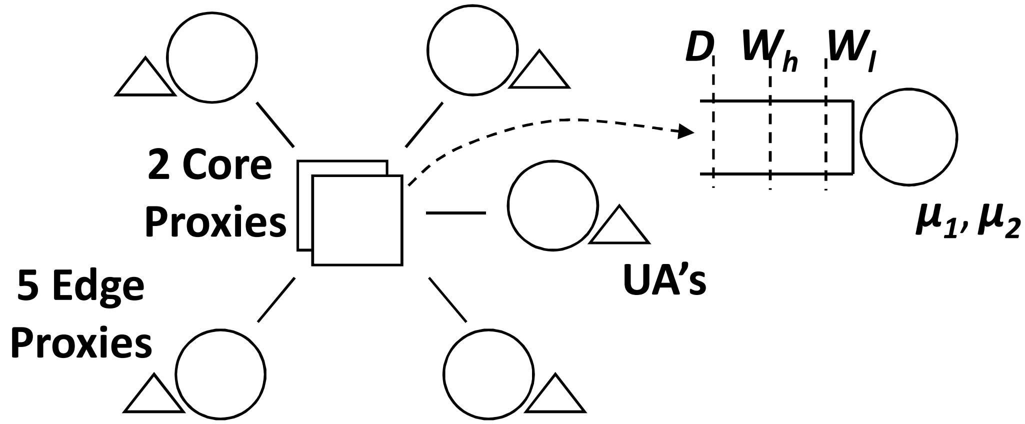
Motivation

- Users expect VoIP/SIP networks to perform as well as the existing PSTN. So efficient handling of overload conditions is crucial to cope with, for instance:
  - Call floods: emergencies, media stimulated calling, …
  - Decrease in processing capacity: component failures, …
  - Avalanche restart: recovery from failures (i.e. power outage), provisioning errors, …



- Researchers have demonstrated VoIP/SIP networks can experience congestion collapse,
- SIP retransmission algorithm and limited overload control capability are the main causes,
- Therefore a SIP overload control mechanism that provides high throughput during times of overload is needed.

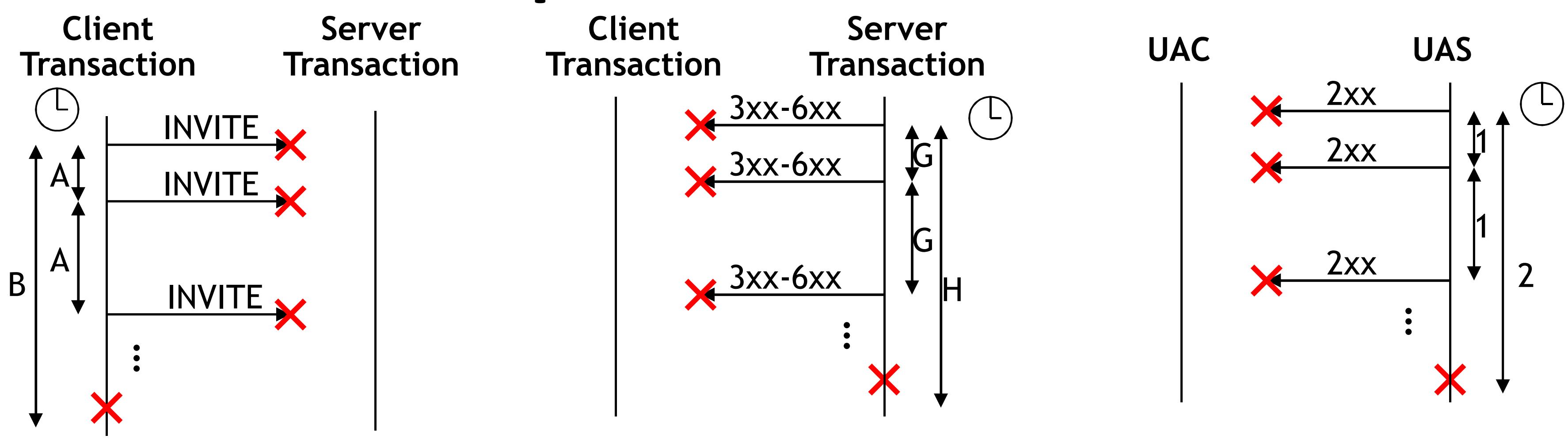
Benchmark Model



- All proxies modeled as a queuing system:
  - Message rate of 500/sec for accepted messages, 3,000/sec for rejected INVITEs,
  - Queue size: 500 messages
  - Internal overload control considered
- Media path congestion is not considered.

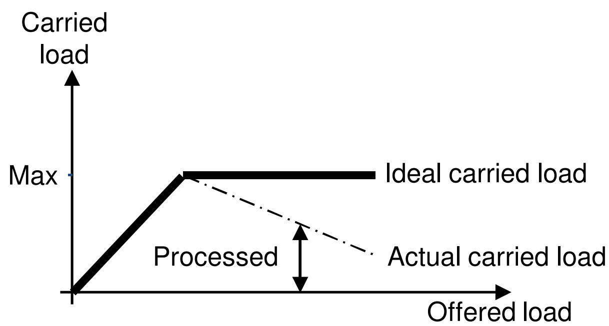
- We focused on server-to-server controls,
  - Overload controls to throttle UA's should be very different than the ones to throttle servers,
- Our benchmark network consisted of UA's establishing calls with one another across a network of 5 edge proxies and 2 core proxies,
  - We used the standard 7 SIP messages call model,
- All proxies were assumed to be stateful and signaling messages within the same call traverse the same set of proxies,
- To evaluate our server-to-server controls and eliminate other sources of interaction, UA's and Edge Proxies were assumed to have infinite capacity.

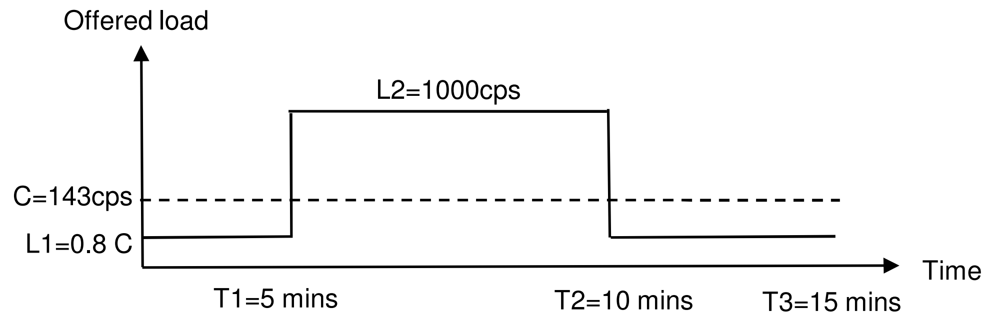
SIP Protocol Implementation



- SIP is an IETF standardized signaling protocol (application-layer) for creating, modifying, and terminating sessions with one or more entities.
  - SIP uses timers for application-level retransmission and message timeouts,
  - Most retransmissions timers are used for UDP only,
  - Short timer initialized to T1(500msec) is doubled every retransmission while long timer is proportional to T1 (64xT1=32sec),
- SIP provides the 503 response (Service Unavailable) for overload control. After receiving a 503 response, a proxy will:
  - Stop sending requests to this server for a number of seconds defined in the Retry-After header (if honored by recipient),
  - Retry at an alternative proxy if one is available or reject the request back to the UAC,
- Each simulation model included detailed implementation of SIP INVITE & non-INVITE client & server transaction state machines.

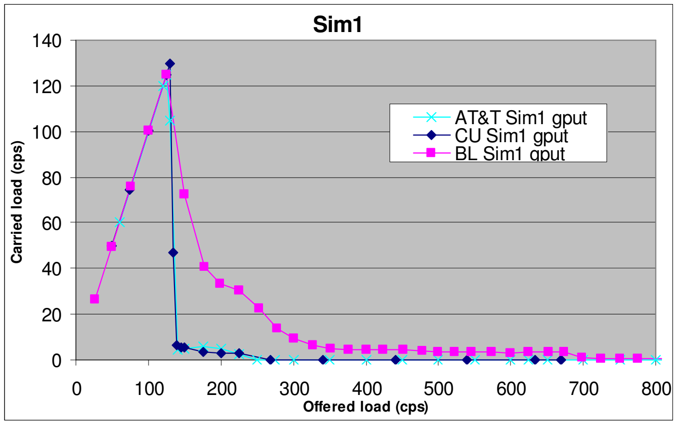
Controls Test Scenarios and Evaluation Metrics

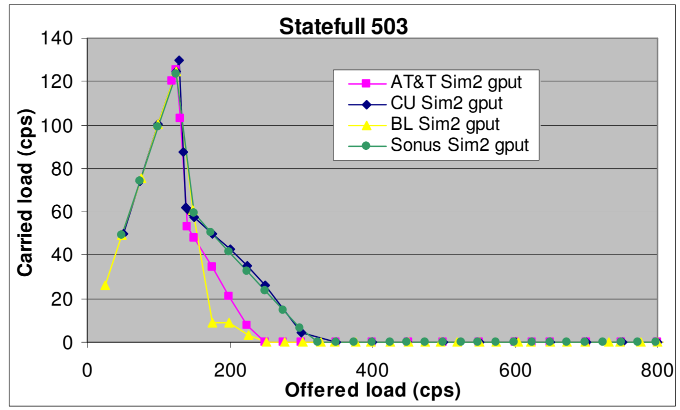


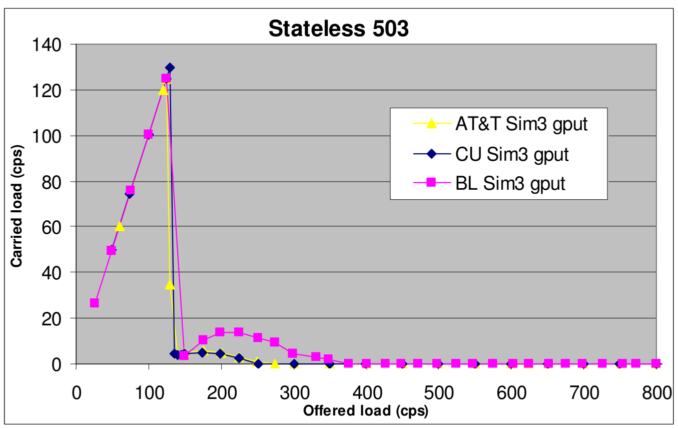


- Overload control algorithms were evaluated under steady-state conditions,
  - For a given offered load, simulations ran until steady-state was achieved,
  - Sensitivity analysis on transmission impairments (message loss and delay), number of proxies and traffic matrix.
- Transient analysis was also considered, initially offered load consisted of a step function,
- Metrics considered were:
  - Goodput or carried load,
  - Core proxy queue and CPU utilization,
  - Convergence speed,
  - Post dial delay.

Simulation models calibration (no controls)

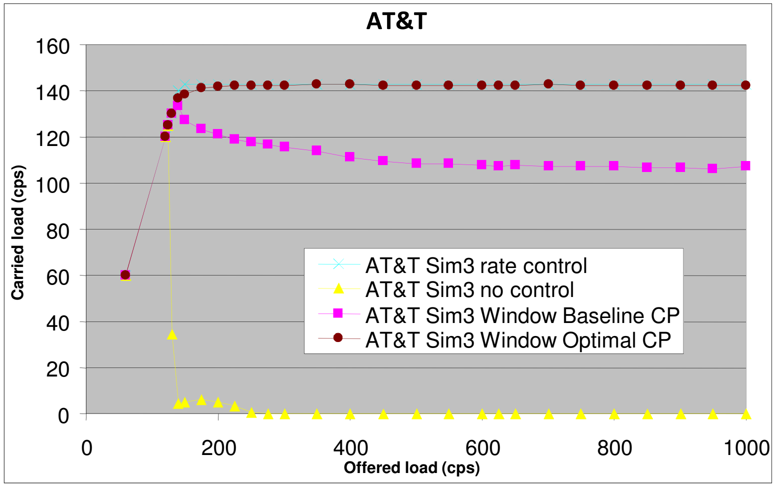


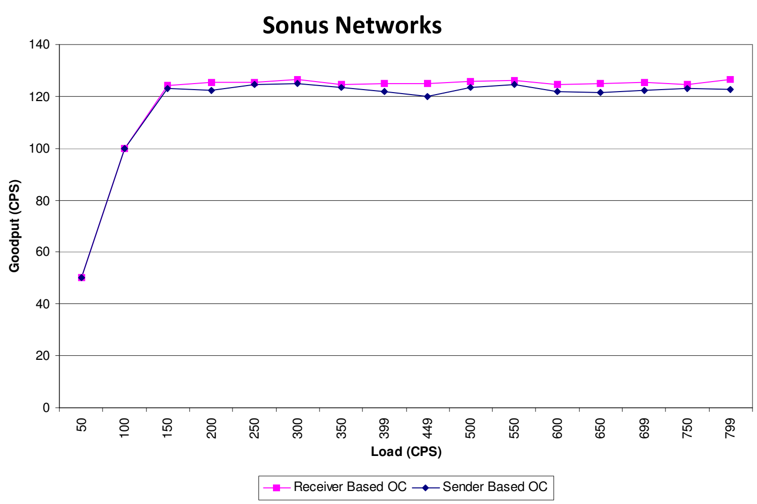




- Within team, two modes of operation were suggested: rejecting new INVITE's statelessly (minimal processing penalty) or statefully,
- Stateless mode of operation uncovered state transition inconsistency in RFC3261 that was resolved in parallel elsewhere (draft-sparks-sip-invfix-00.txt),
- Simulation calibration between different teams turned out to be challenging as each interpreted RFC3261 somewhat differently,
- It took ~6months for models to show a reasonable level of agreement.

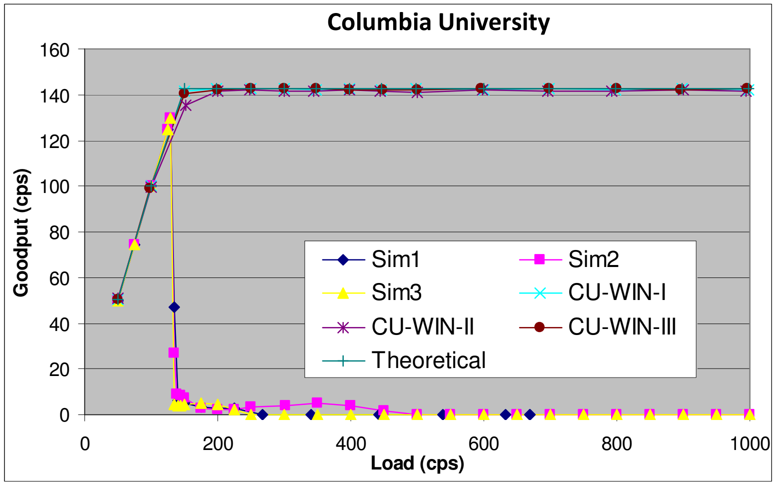
Results for steady-state conditions

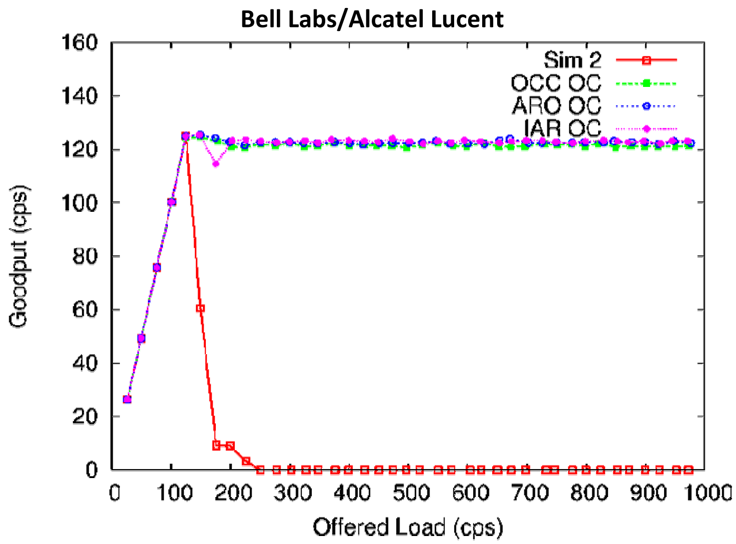




- AT&T rate & window controls:
  - Core proxies internal overload control based on queue fill,
  - Rate and window tuned to activate prior internal control limits,
  - In rate control, core proxies estimate control rate based on queue delay deviation from target and distribute allowed rates via SIP responses to edge proxies,
  - Edge proxies throttle based using an percent blocking algorithm,
  - In window control, edge proxies maintain a window of unacknowledged INIVITEs based of core proxies responses (503's shrink window 1xx & 2xx open window)
- Sonus Networks receiver & sender based controls:
  - Core proxies internal overload control adjusts allowed call rate based on CPU utilization deviation from a target value.
  - Excess calls are rejected via 503-Service Unavailable.
  - Core proxies distribute individual allowed rates to upstream proxies via SIP messages.
  - Upstream proxies adjust their throttles based on the received allowed rates.
  - In sender based control, upstream proxies dynamically adjust rates based on the overload notifications from downstream proxies (503-Service Unavailable) such that the overload notification rate converges close to a target overload notification rate.

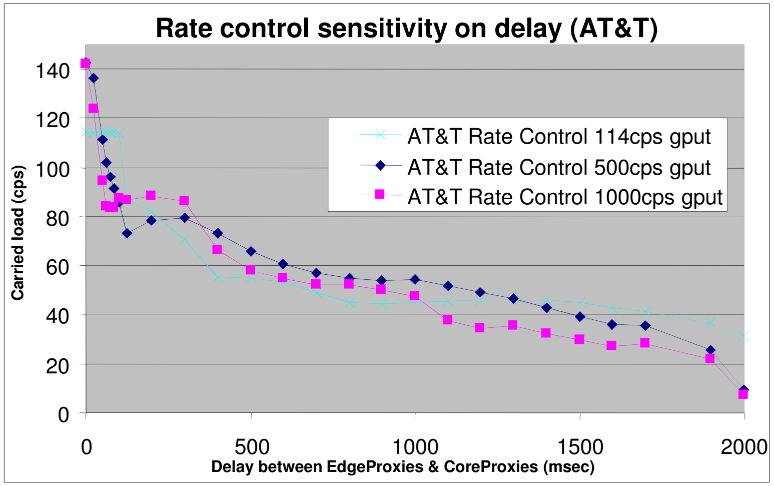
Results for steady-state conditions

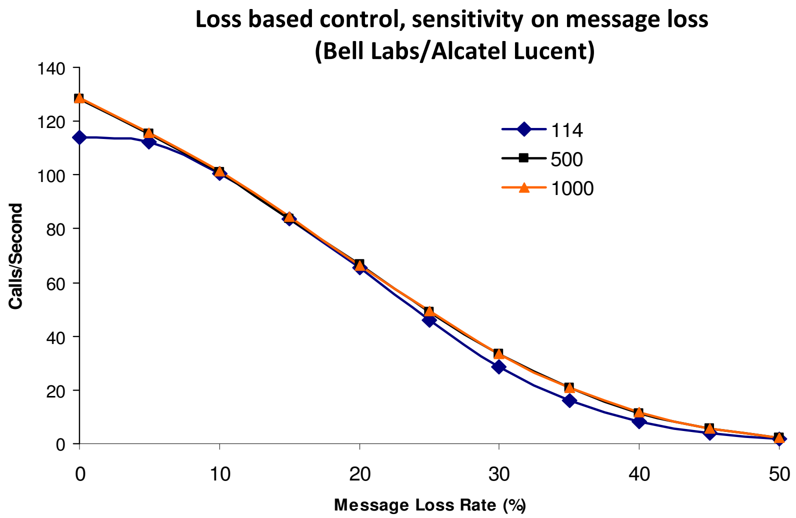


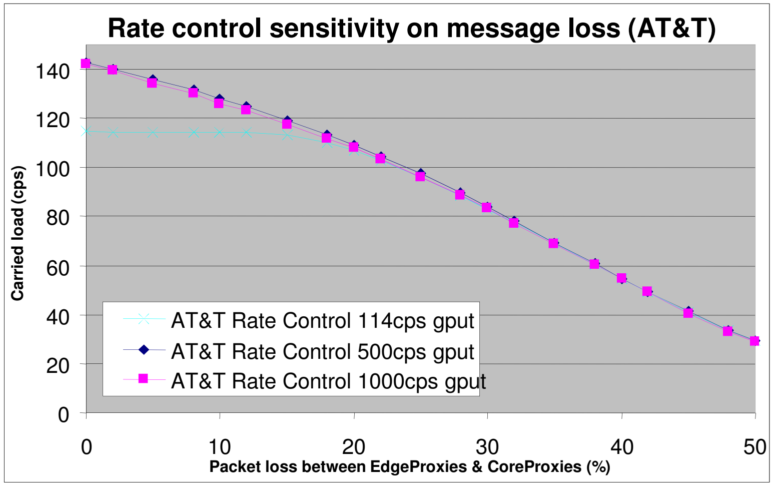


- Columbia University window-based overload control:
  - SIP session as control unit, dynamically estimated from processed SIP messages,
  - Receiver (Core Proxy) dynamically computes available window and feedback piggybacked in responses/requests,
  - Three different window adaptation algorithms:
    - CU-WIN-I: keep current estimated sessions below total allowed sessions given target delay
    - CU-WIN-II: open up the window after a new session is processed
    - CU-WIN-III: discrete version of CU-WIN-I, divided into control intervals
- Bell Labs/Alcatel Lucent loss-based overload control simulation :
  - Feedback-loop between receiver (core proxy) and sender (edge proxy).
  - Receiver driven control algorithm (estimates current processor utilization, compares to target processor utilization, multiplicative increase and decrease if loss-rate to reach target utilization).
  - Sender adjusts the load it sends to receiver based on the feedback received using percent-blocking.
  - Overload control algorithms: Occupancy algorithm (OCC), ARO algorithm (ARO), improved ARO (IAR).

Results for sensitivity on transmission impairments

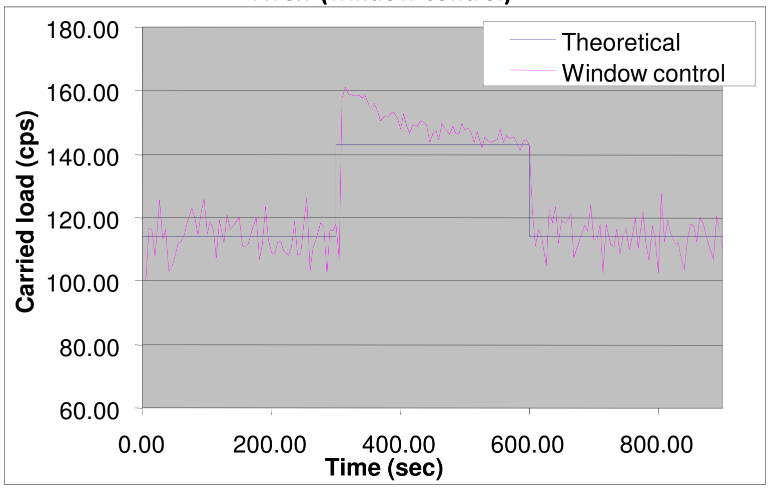




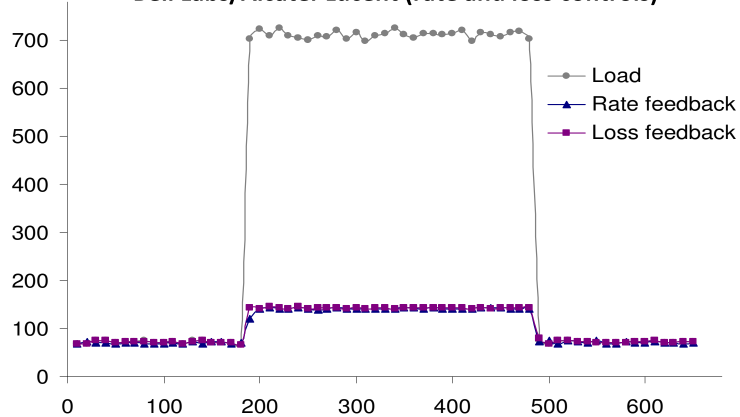


Results for transient conditions

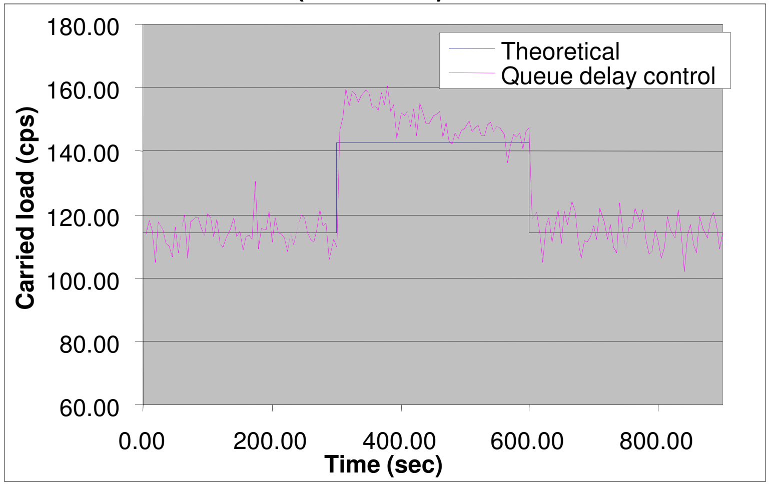
AT&T (window control)



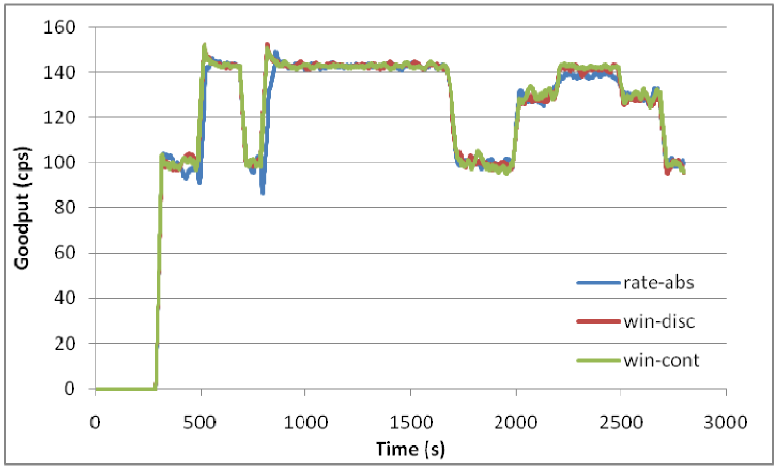
Bell Labs/Alcatel-Lucent (rate and loss controls)



AT&T (rate control)



Columbia University (rate and window control)



Conclusion

- With increasing size of VoIP/SIP deployment, efficient handling of overload conditions is crucial for VoIP/SIP service providers,
- IETF design team simulation results showed overload control algorithms produce stable VoIP/SIP traffic behavior and maintain high throughput under various overloads,
- Additional Info:
  - IETF draft RFC draft-hilt-soc-overload-design (V. Hilt coordinator),
  - 07'ITC20 and 09'ITC21 (E. Noel, C. Johnson),
  - ICNP'08 (V. Hilt, I. Widjaja),
  - IPTComm'08 (C. Shen, H. Schulzrinne, E. Nahum),
  - IEEE CCNC'11 (A. Abdelal, W. Matragi).