The BGP-Inspect Project

Manish Karir, Jon Oberheide (Merit)
Dionysus Blazakis, John Baras (UMd)

The Problem

- Large amounts of data are now, or soon will be available:
  - Route Views, RIPE Archives, PREDICT, etc
- The problem is no longer access to raw data but how to extract useful information from the raw data
- Need tools that can:
  - Scale to large input datasets
  - Provide useful data summarizations
  - Are easy to use
  - Provide useful information
- BGP-Inspect
  - Goal is to attempt to make it easier to use raw data from archives such as Route Views, by pre-processing, reformatting and indexing the data

Outline

- BGP-Inspect and BGPdb
  - Overview and new features
- BGP-Inspect in Action!
  - Case studies
- Conclusions, Future Work and Discussion

BGP-Inspect: Why and What

- Analyzing MRT Data:
  - Large volumes of data ~RV-66G compressed
  - Extracting useful information requires writing custom parsers even for basic information
  - Lots and lots of redundancy
- Approach:
  - Preprocess Route Views data
  - Remove redundancy as much as possible
  - Use data compression to the extent possible
  - Build efficient indices to help queries
  - Pre-compute and store commonly used statistics at data load time not at query time
  - Build easy to use interface

BGPdb

- BGPdb is the core of the BGP-Inspect system
- BGPdb represents the pre-processed database, which is queried by the BGP-Inspect interface
- Provides some useful techniques that maybe applied to processing other large datasets not just BGP datasets

BGPdb – Techniques and Algorithms

- Removing redundancy from BGP datasets
  - ASPATH, COMMUNITY, UPDATE Msgs are repeated over and over, only time changes
- Compressed-Chunked Files
  - Compromise between size and usability
- B+ Tree indices
  - Indexing based on time, this enables fast time-range queries
- Caching while processing input datasets
  - Messages are repetitive, so keep cache of previous processing for speedup

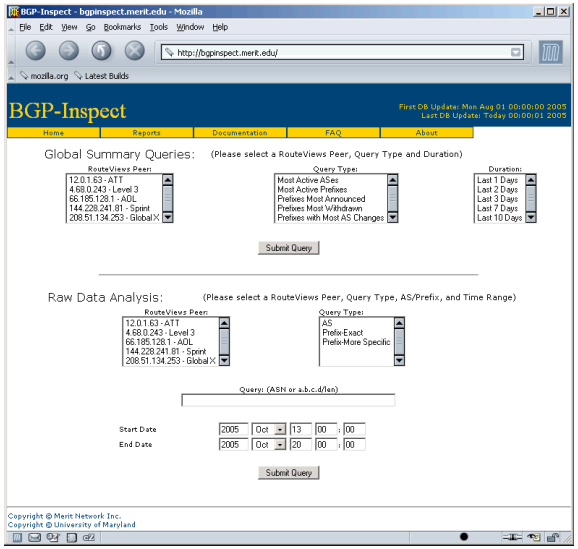
BGP-Inspect: Current State

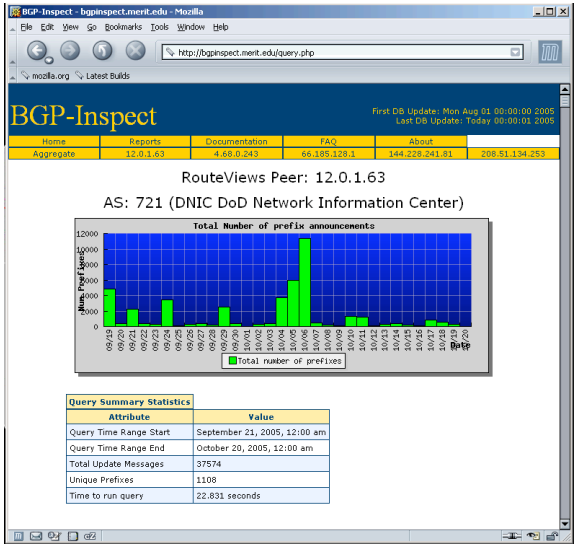
BGP-Inspect – Beta v0.5
http://bgpinspect.merit.edu

Dataset: August 1 - Present
Current BGPdb size: 102GB
Currently indexing data for 5 peers (AT&T, Level 3, AOL, Sprint, Global X)

- Example queries (per peer, 1,7,10 days):
  - Most active AS's
  - Most active prefixes
  - Prefixes with most OriginAS changes

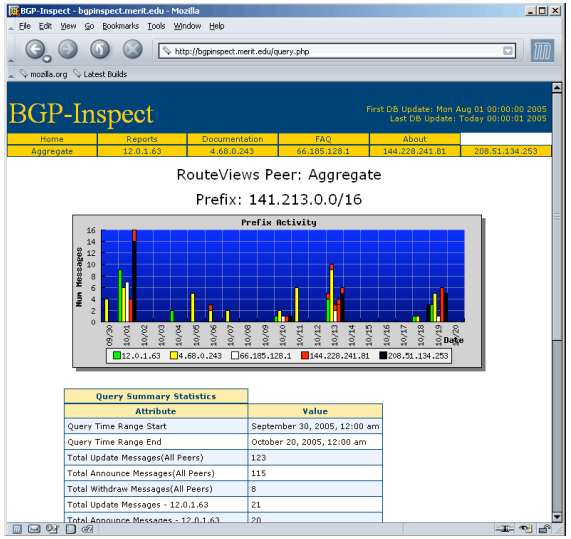
- Raw Data Analysis(per peer)
  - Prefix/AS, Time Range
  - Uniques prefixes by AS
  - OriginAS changes for a prefix
  - Time to run query
  - More specific prefixes announced

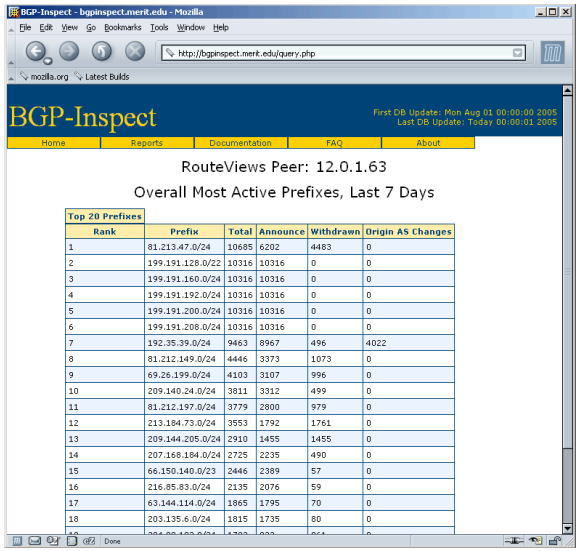




BGP-Inpsect: Current State (2)

- Equipment
  - Dell 2650 - Web and DB server
  - Dell 2850, dual Xeon with NFS mounted 500GB SATA
- Traffic?
  - ~30 unique IPs per day





Newly Added Features

- Daily auto-load script
  - Cron job
  - Fetch the previous days Route Views data from the public server
  - Insert everything into BGPdb
    - Both for raw queries…
    - …and for the summary statistics.
  - “Productionized” - everything hardened for public use

Newly Added Features (2)

- Multi-peer queries
  - As requested from last session, a response from multiple peers can be requested from a single query.
  - Tabbed Interface
    - One aggregated tab giving summary info and a multi-bar graph…
    - …and a tab with detailed results for each peer.

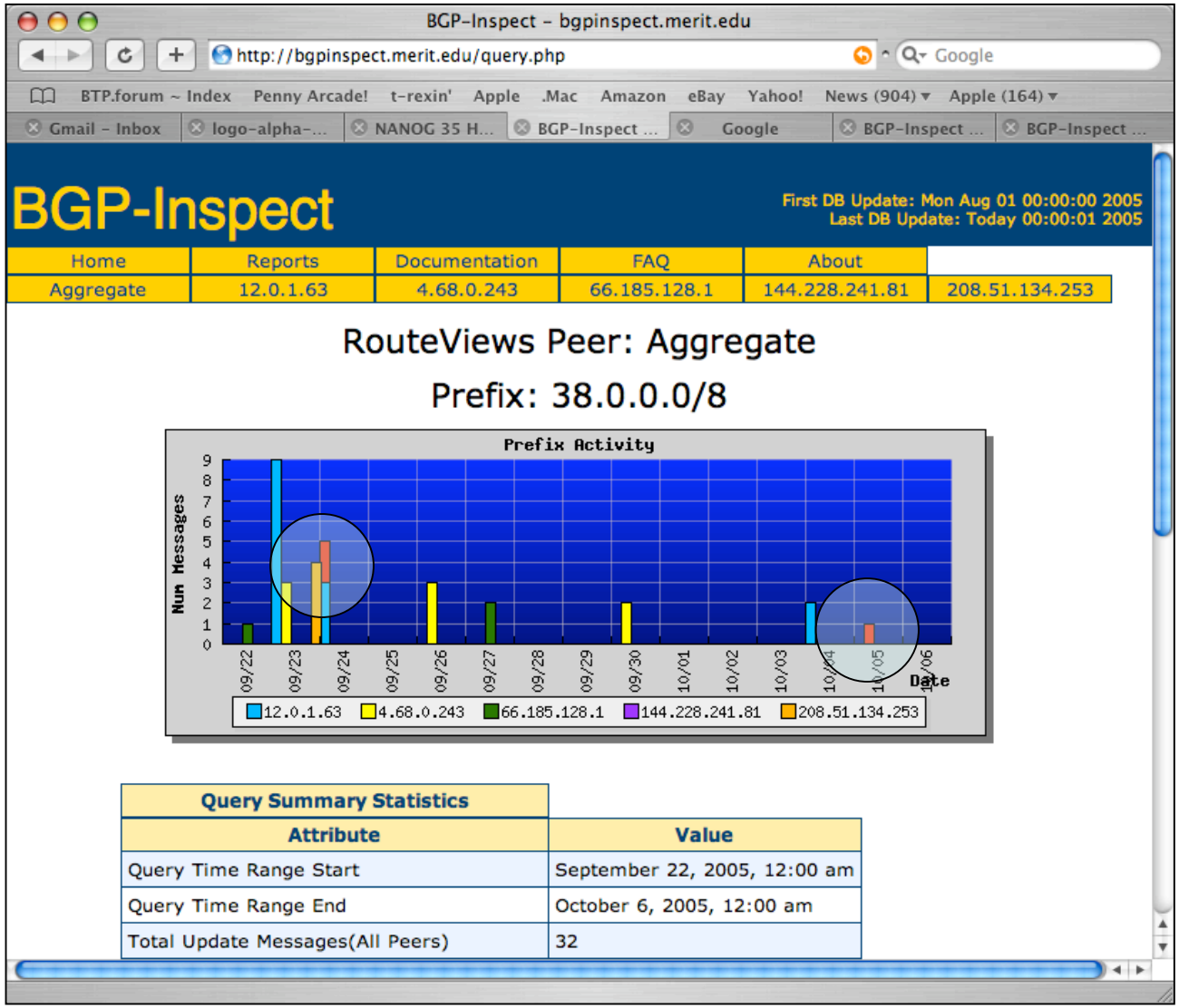
Newly Added Features (3)

- Internal fixes and clean-ups
  - Redesigned web infrastructure
    - Cleaner, more structured PHP as opposed to a C CGI implementation
    - UI enhancements (more to come?)
  - New build system to facilitate a source release in the future

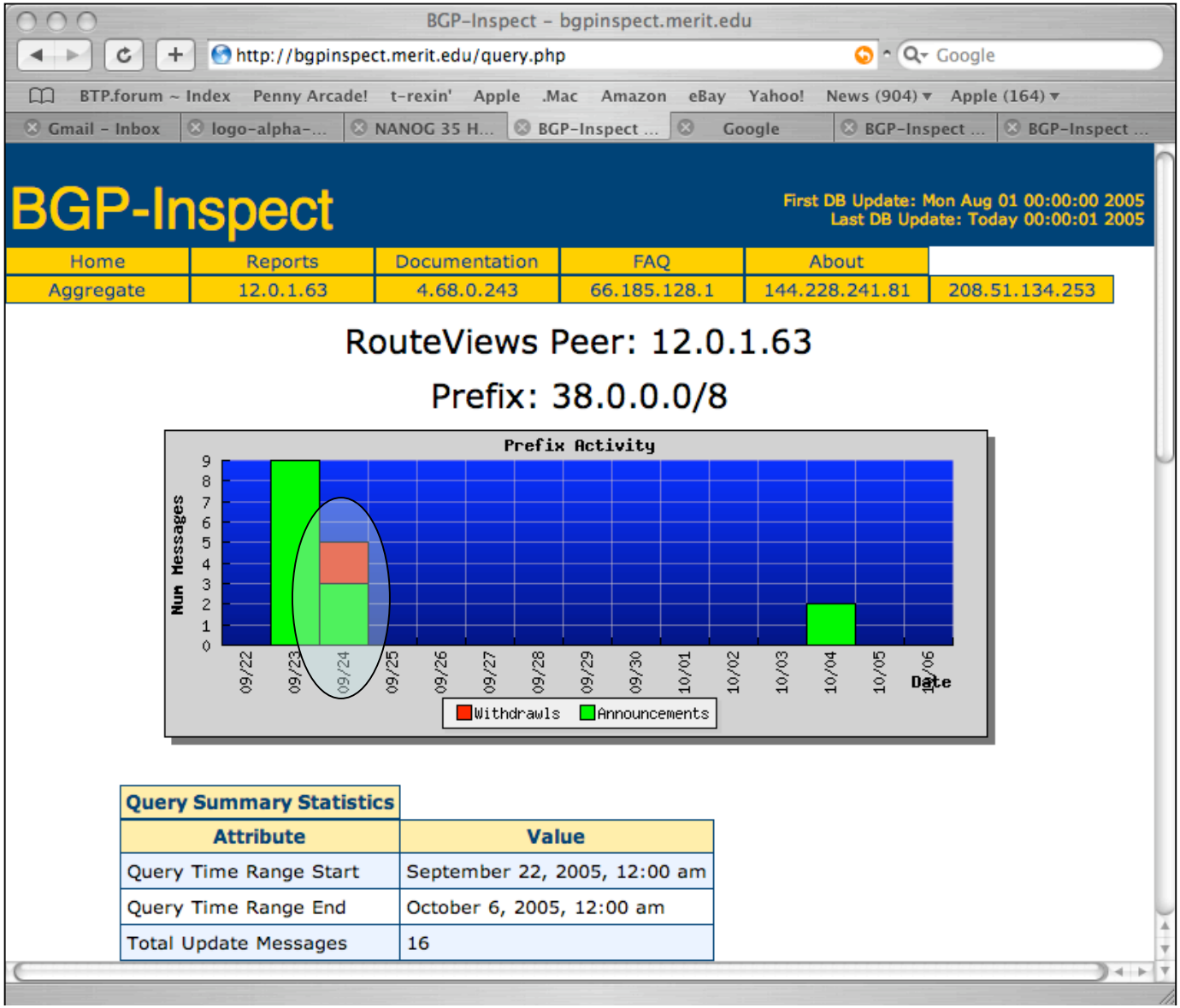
Case Study: Cogent de-peering

- **On Oct 5th 2005, Level 3 and Cogent stopped peering directly with each other.** In order to examine the impact of this event we ran a raw data analysis query for 38.0.0.0/8 (Cogent) for the various Route View peers (that we currently index). This determines the reach-ability impact of this event on 38/8 from the 5 peers in our data. Our initial analysis ranges from September 22nd through October 6th.

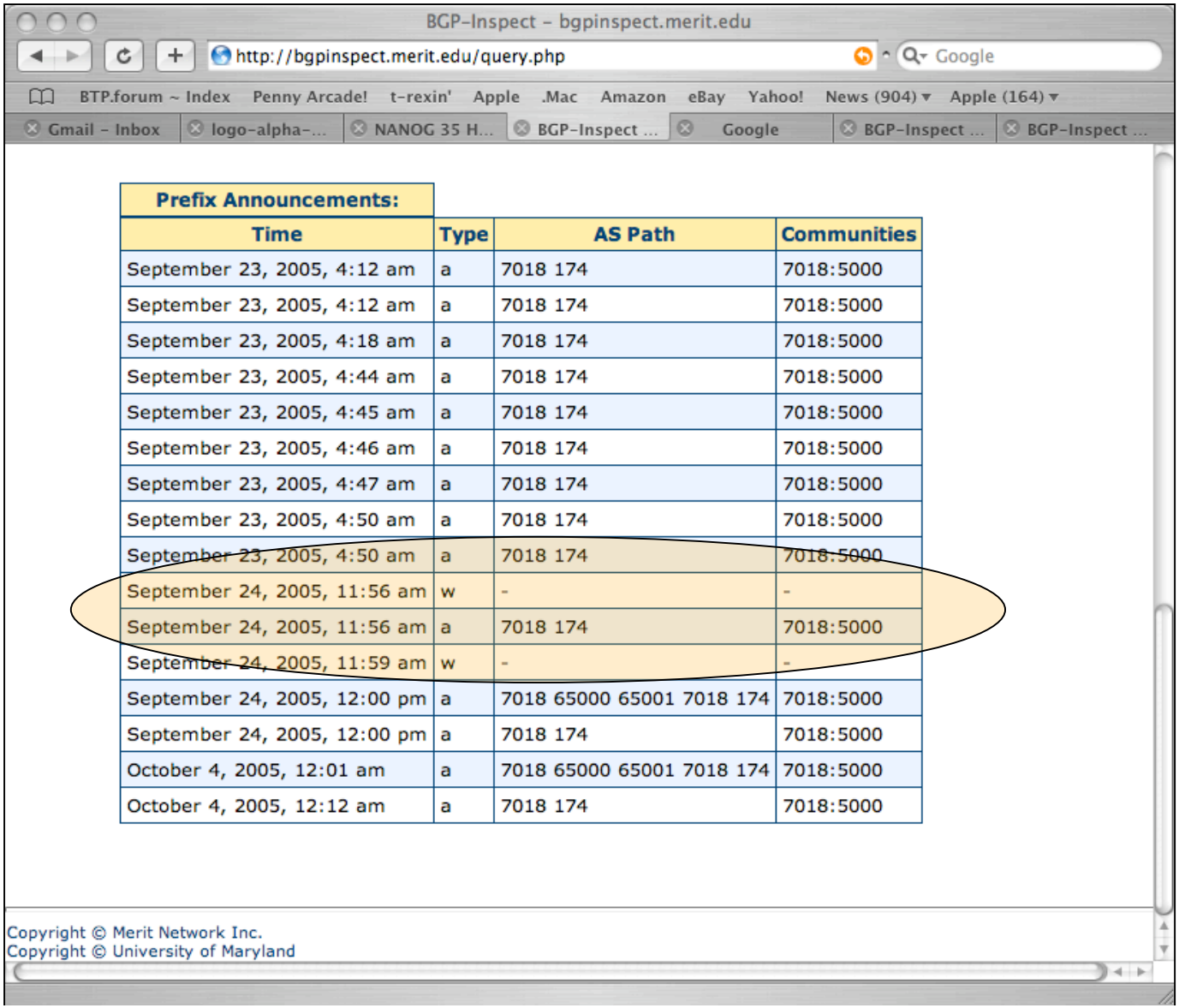
Case Study (cont.) - Aggregate



Case Study (cont.) - AT&T



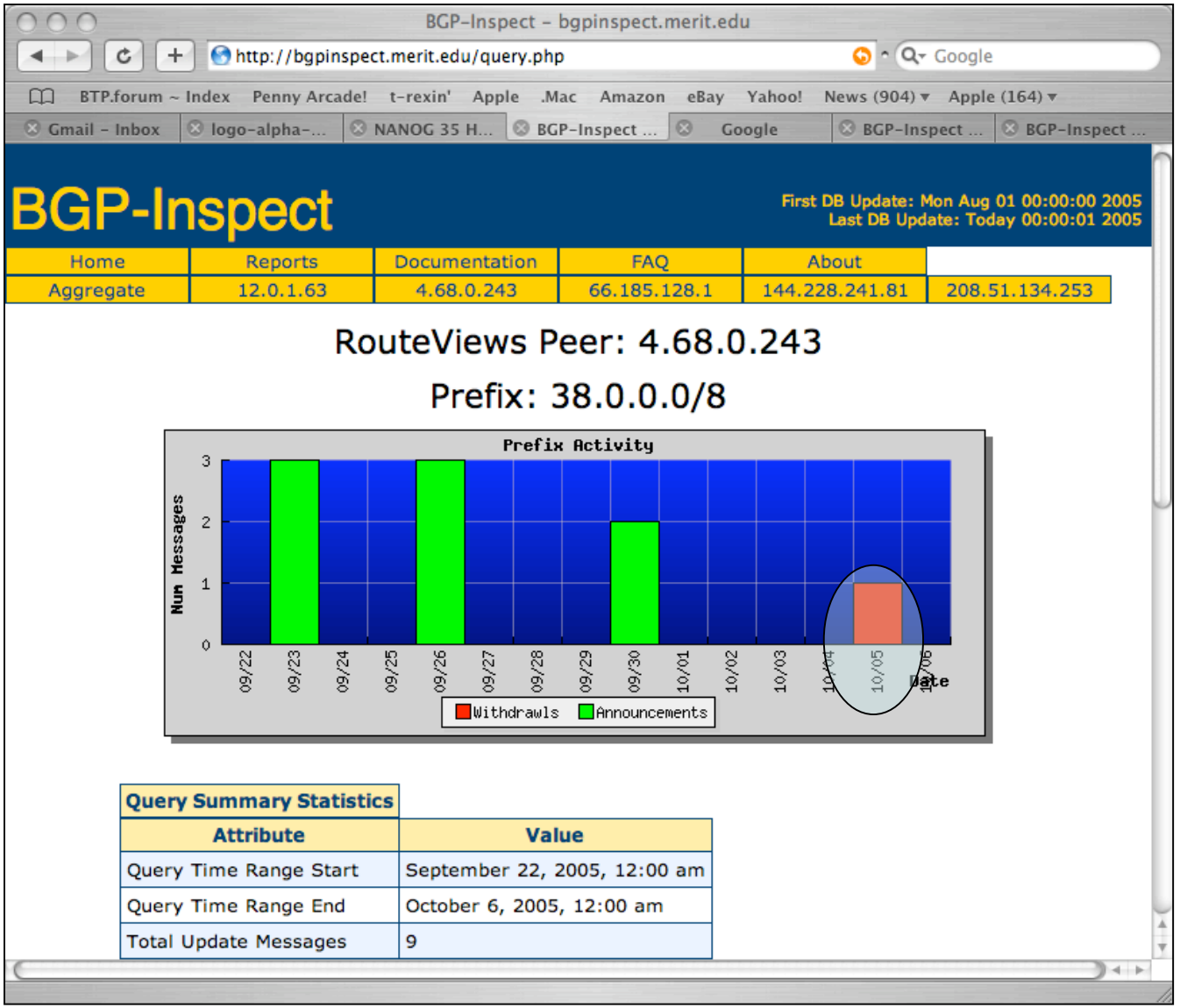
Case Study (cont.) - AT&T



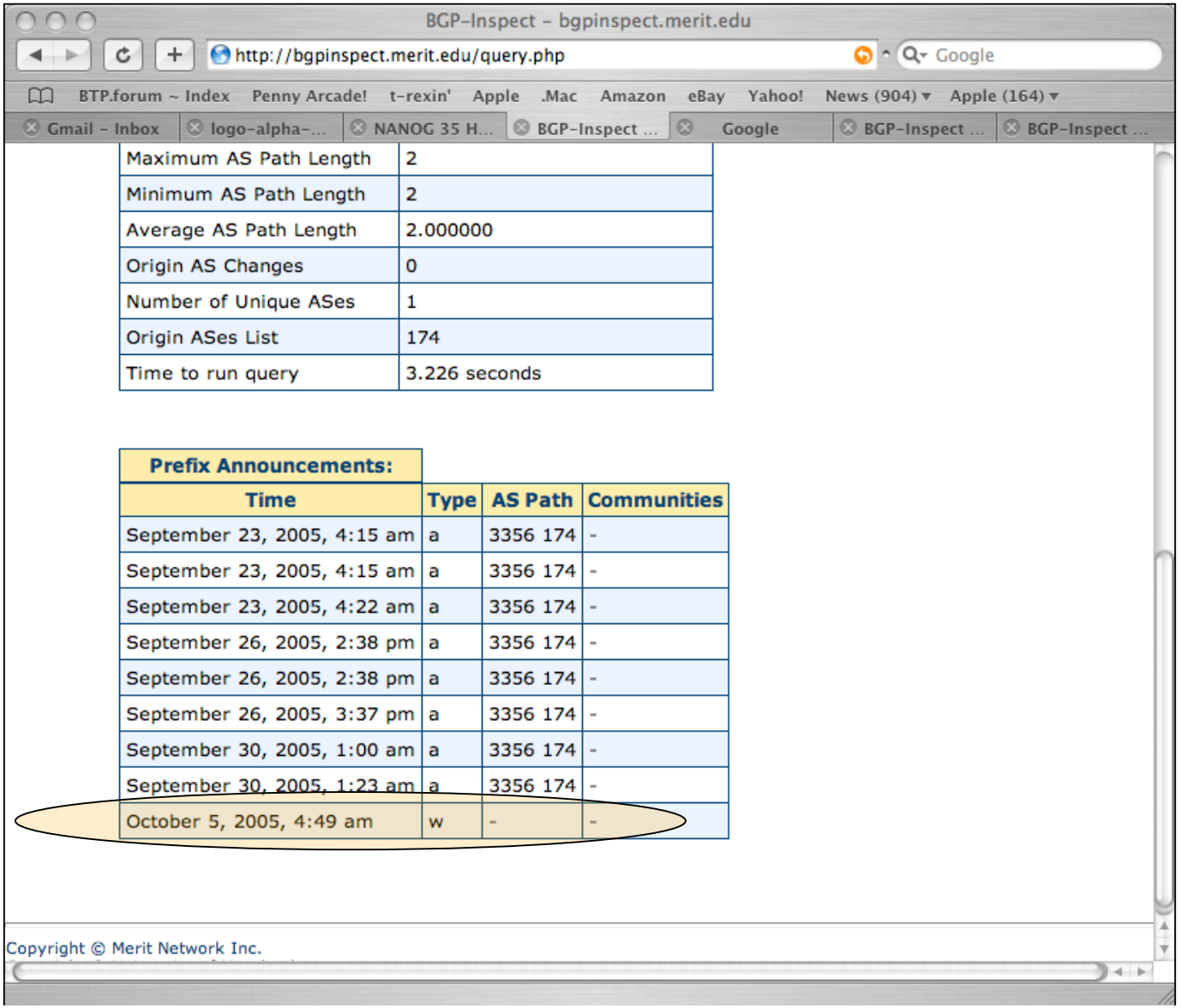
Prefix Announcements:

Time	Type	AS Path	Communities
September 23, 2005, 4:12 am	a	7018 174	7018:5000
September 23, 2005, 4:12 am	a	7018 174	7018:5000
September 23, 2005, 4:18 am	a	7018 174	7018:5000
September 23, 2005, 4:44 am	a	7018 174	7018:5000
September 23, 2005, 4:45 am	a	7018 174	7018:5000
September 23, 2005, 4:46 am	a	7018 174	7018:5000
September 23, 2005, 4:47 am	a	7018 174	7018:5000
September 23, 2005, 4:50 am	a	7018 174	7018:5000
September 23, 2005, 4:50 am	a	7018 174	7018:5000
September 24, 2005, 11:56 am	w	-	-
September 24, 2005, 11:56 am	a	7018 174	7018:5000
September 24, 2005, 11:59 am	w	-	-
September 24, 2005, 12:00 pm	a	7018 65000 65001 7018 174	7018:5000
September 24, 2005, 12:00 pm	a	7018 174	7018:5000
October 4, 2005, 12:01 am	a	7018 65000 65001 7018 174	7018:5000
October 4, 2005, 12:12 am	a	7018 174	7018:5000

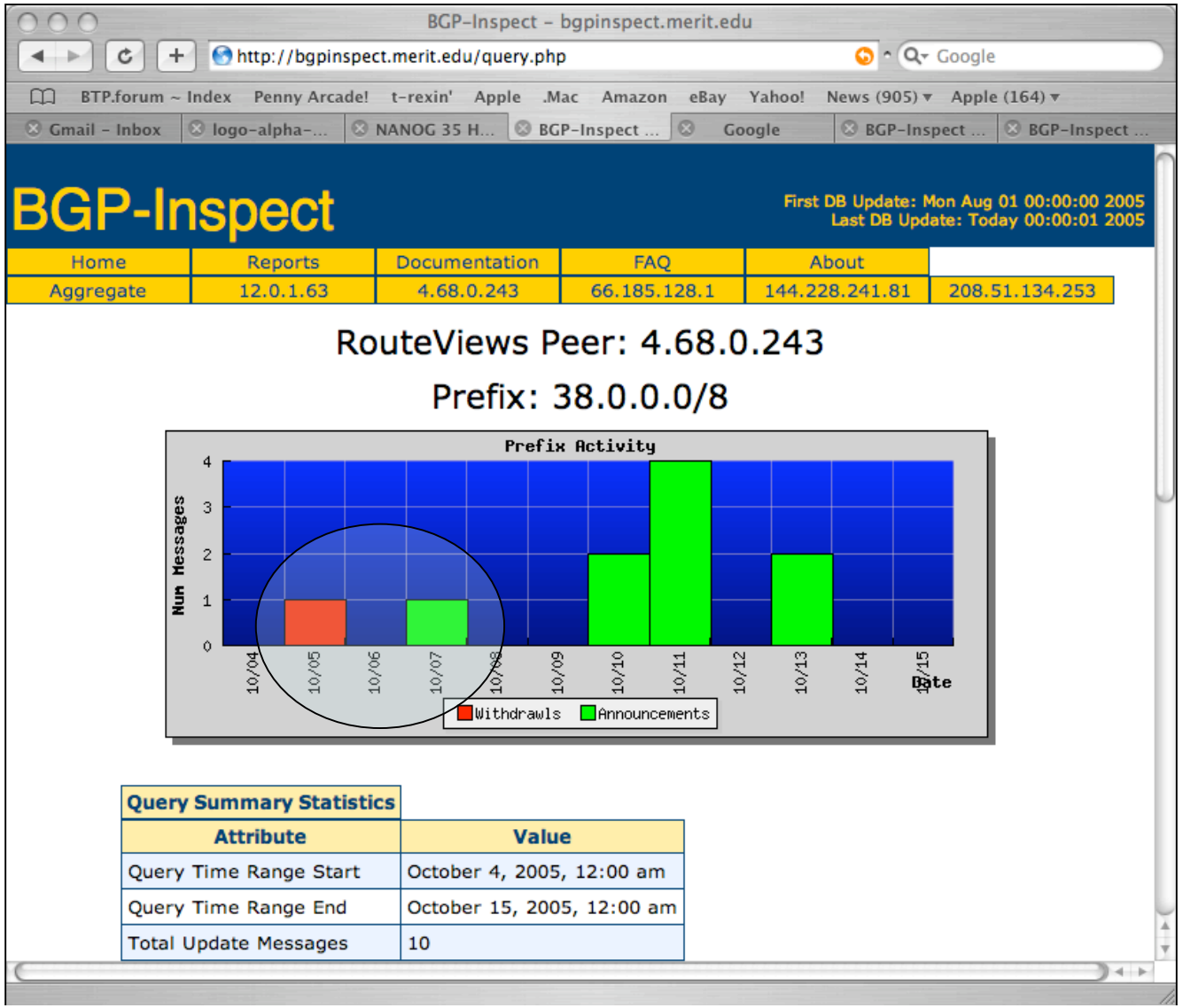
Case Study (cont.) - Level 3



Case Study (cont.) - Level 3



Case Study (cont.) - Level 3



Conclusions and Future Work

- Efficient tools for BGP analysis are needed and we've created some.
- BGP-Inspect is available at http://bgpinspect.merit.edu and your feedback is very much appreciated.

- Future…
  - More interesting things with the multiple peer response UI (different ways of highlighting the differences between peers)
  - pyBGPdb - a python interface to the BGPdb database providing fast raw queries
  - Automated anomaly detection using these tools