

UKAIRO: Internet-Scale Bandwidth Detouring

Thom Haddow* Sing Wang Ho+ Jonathan Ledlie† Cristian Lumezanu‡ Moez Draief+ Peter Pietzuch*

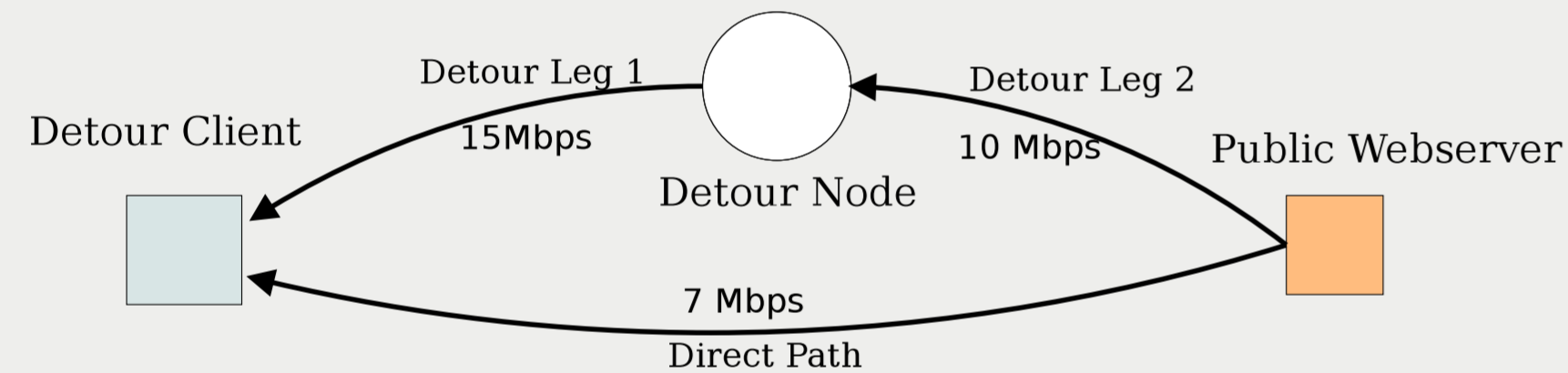
*Imperial College London, Dept. of Computing; +Imperial College London, Dept. of Electrical Engineering; †Nokia Research; ‡Georgia Tech

Motivation

- ▶ Bandwidth is an increasingly critical and user visible Internet metric
 - ▶ Trend towards cloud-based storage architectures
 - ▶ Rapidly increasing media sizes
 - ▶ Last-mile no longer bandwidth bottleneck
- ▶ CDNs are an excellent approach for certain cases, but...
 - ▶ Not all content is popular or statically cacheable.
 - ▶ CDNs generally cost *providers* to provision
- ▶ Can improve bandwidth on arbitrary Internet paths through **detour routing**

Detour routing

Detour routing: Redirecting Internet traffic via tertiary hosts to improve upon the metrics of the default Internet path

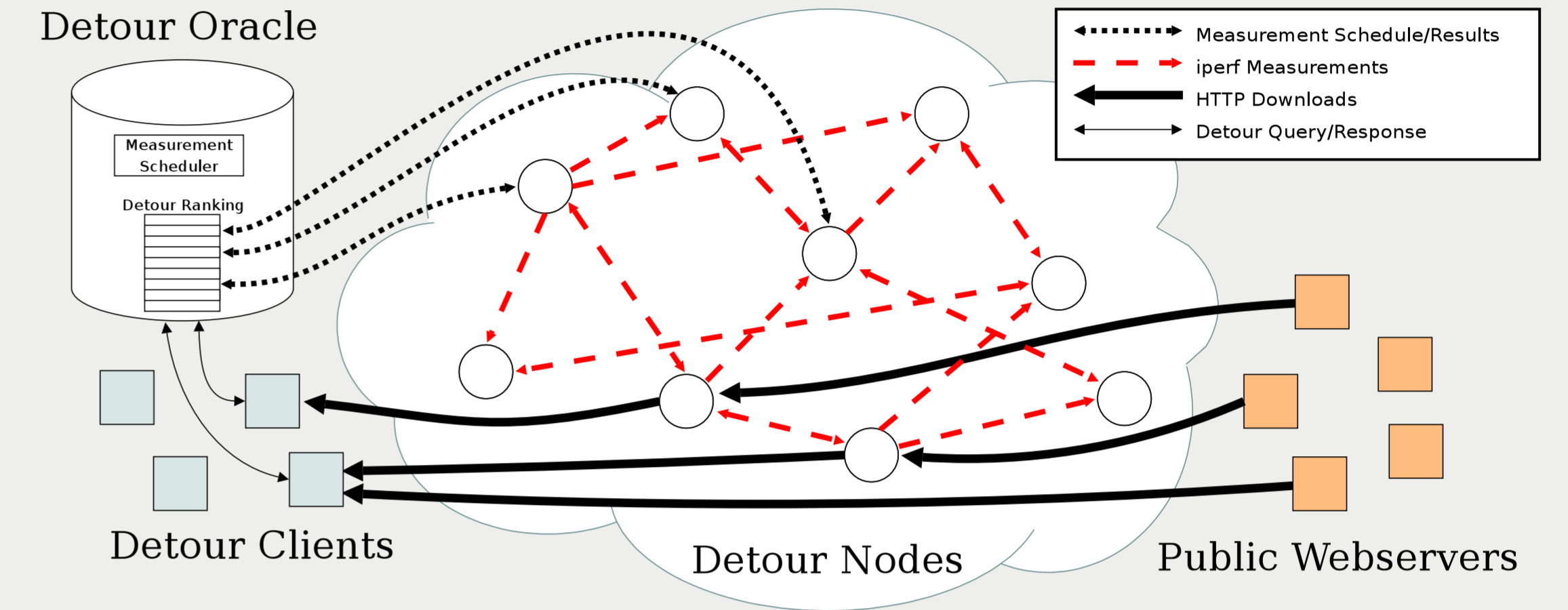


- ▶ Does not require adapting infrastructure/protocols
- ▶ Works for many metrics: latency, jitter, loss, availability...
- ▶ For bandwidth, large-scale measurements show 75% of paths could be improved by **at least 1Mbps and 20%**
- ▶ **Main challenges:** Scalably identifying detour nodes, and selecting an effective detour node for a given path

Architecture overview

Ukairo: A service which discovers and exploits alternative detour paths to improve client performance to arbitrary Internet destinations

- ▶ **Detour Oracle**
 - ▶ Schedules periodic **TCP throughput** measurements between **detour nodes**
 - ▶ Ranks detour nodes according to detouring potential
- ▶ **Detour clients**
 - ▶ Implemented as HTTP client library or proxy
 - ▶ Gets multiple detour node suggestions from oracle
 - ▶ Uses **dynamic path selection** to select best **detour node** at time of use
- ▶ **Routing plane**
 - ▶ SOCKS proxies are deployed on all **detour nodes**, enabling arbitrary TCP connections to be relayed via them



Making detouring scalable

Problem: Whole-network all-to-all bandwidth measurement unscalable

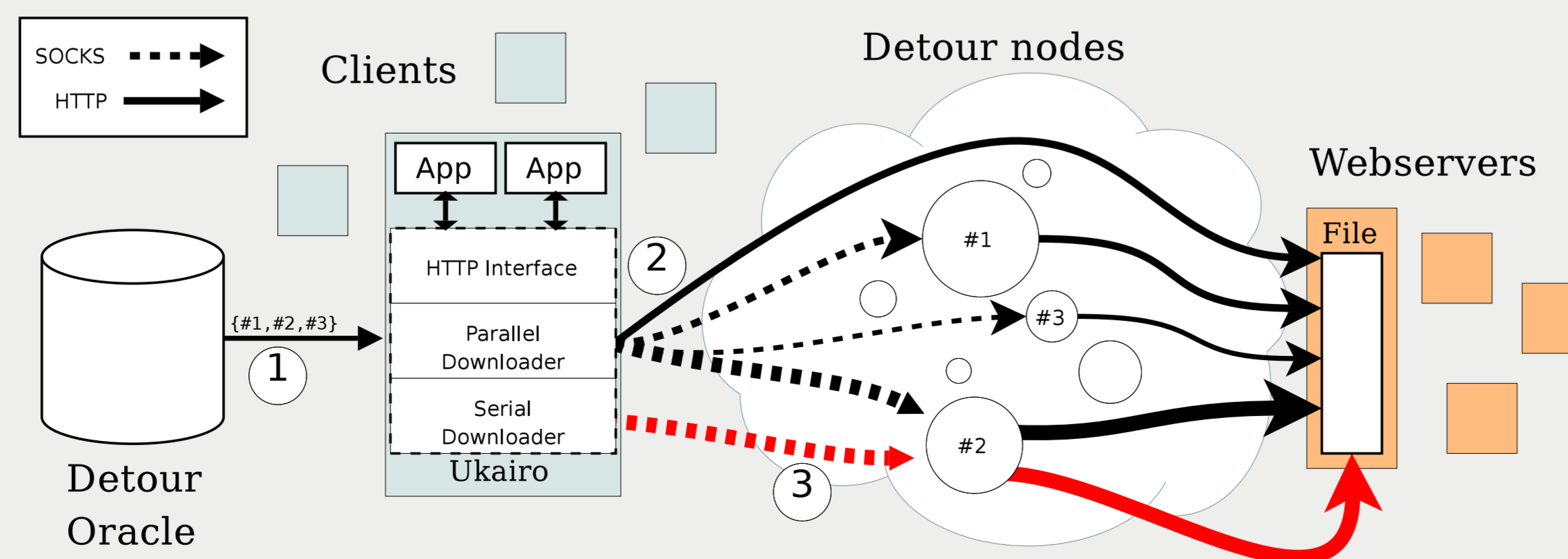
- ▶ Measure bandwidth between a **limited set of nodes**
- ▶ Relative detour effectiveness of nodes can be **ranked** without performing all-to-all measurements
- ▶ Large scale PlanetLab measurement suggests **75% of potential global detour improvement** can be extracted using only the detours found through measuring bandwidth between **50 randomly chosen nodes**
- ▶ From a random selection of 50 Internet nodes, around half (~20) will act as effective detours

Problem: Clients must select appropriate detour node for each destination at time of use

- ▶ Dynamic path selection possible for small sets of potential paths
- ▶ Measurement analysis suggests each client only needs 5 specific detour nodes (the **detour set**) to achieve near optimal bandwidth improvement across all Internet paths
- ▶ Detour set selection methods based on **best-ranked** and **load-balanced** randomisation prove effective
- ▶ Identifying client-specific detour set directly is an open problem
 - ▶ Adaptive strategies can tune detour set over successive transfers

Client detouring

- ▶ **Path switching**
 - ▶ Easily implemented for HTTP by exploiting GET idempotence and Range header to enable partial downloads over different paths
- ▶ **Dynamic path selection**
 1. Get **detour set** from **oracle**
 2. Perform **limited** download over direct and detour paths in **parallel**
 - ▶ Stop after 5 seconds (long enough to reach TCP steady-state on most paths without causing disruption)
 3. Complete download over best path



Detouring results

- ▶ **Detour set selection**
 - ▶ **load-balanced** approach comparable to **best-ranked**
 - ▶ Scales to 20 simultaneous clients
 - ▶ Most failures (path misprediction and overheads) result in than **< 25%** performance loss
 - ▶ Significant bottleneck in using PL destinations
- ▶ **Public webservers**
 - ▶ Detouring can improve bandwidth on over 80% of paths
 - ▶ **70% median improvement**

