OpenSIPS clustering and balancing of media servers

Bogdan-Andrei Iancu
CEO Voice System
Founder OpenSIPS Project

Once upon a time there was a Switch and a PBX.....

.....there was OpenSIPS and Yate.

And they found out they complete each other ...

..... and decided to work out an easy integration.

They decide to make complex VoIP simpler to deploy.

OpenSIPS and Yate are the first Open Source projects to join efforts for building IP based clusters.

The partnership started in mid 2008 aiming to develop a frame for easy integration between OpenSIP as a switch and Yate as a PBX.

OpenSIPS developed a new functionality:

Load Balancing = call routing based on the realtime load of peer elements :

- To control a cluster of heterogeneous PBX/media servers.
- To be driven (via real-time feedback) by the cluster elements.

What makes Load Balancing so special?

- It is able to monitor the load (as ongoing calls) for each peer.
- Instead of blind routing (like dispatcher module), it selects the target based on required resources and available load.

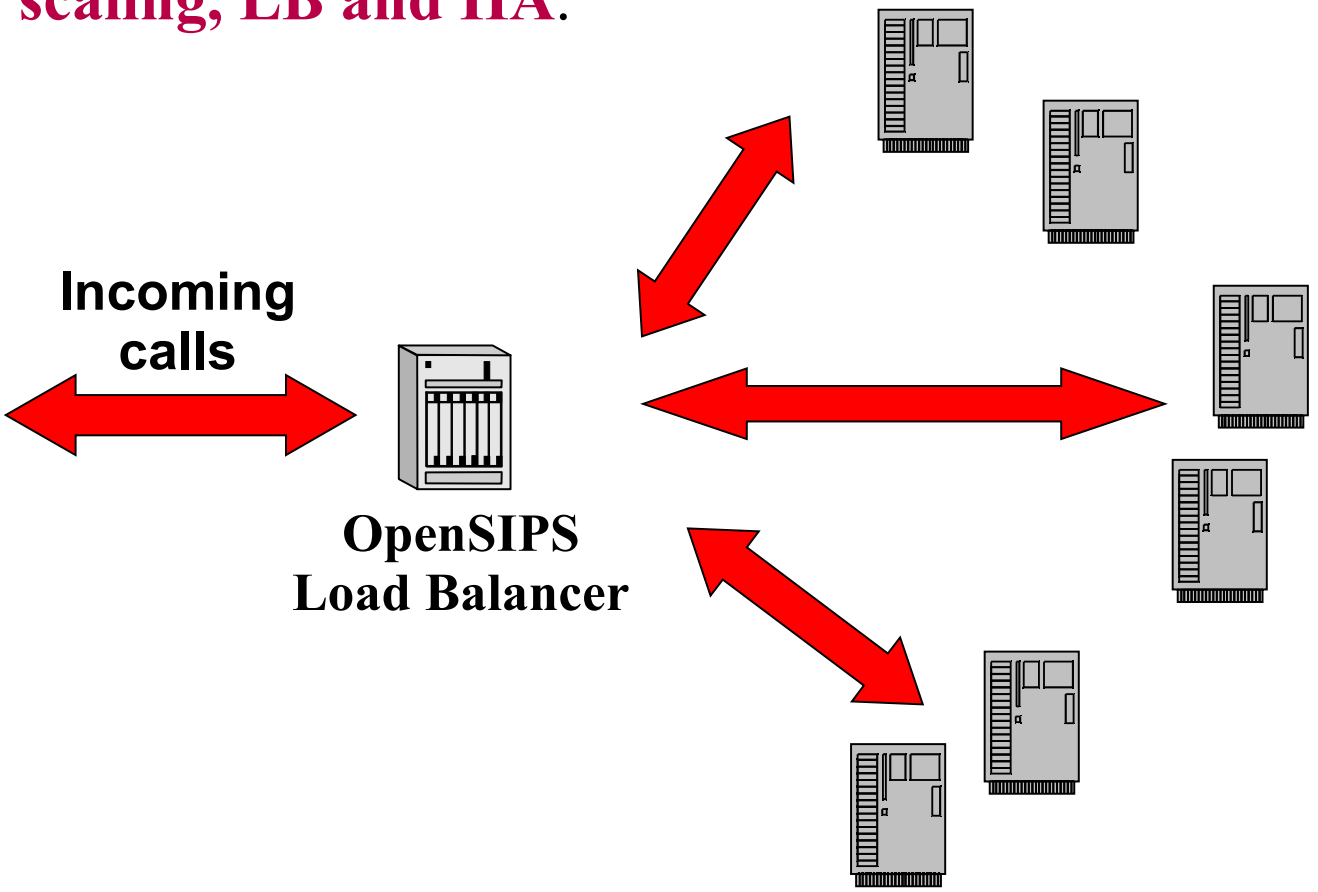
What makes Load Balancing so special?

- It is an interesting feature for a proxy as proxies are typically only transaction stateful (no dialog state).
- It is able to provide failover to the peers from the cluster (even if the peers do not have any support) – detection and re-routing.
- Can combine routing with LB functionality.

Clustering scenarios

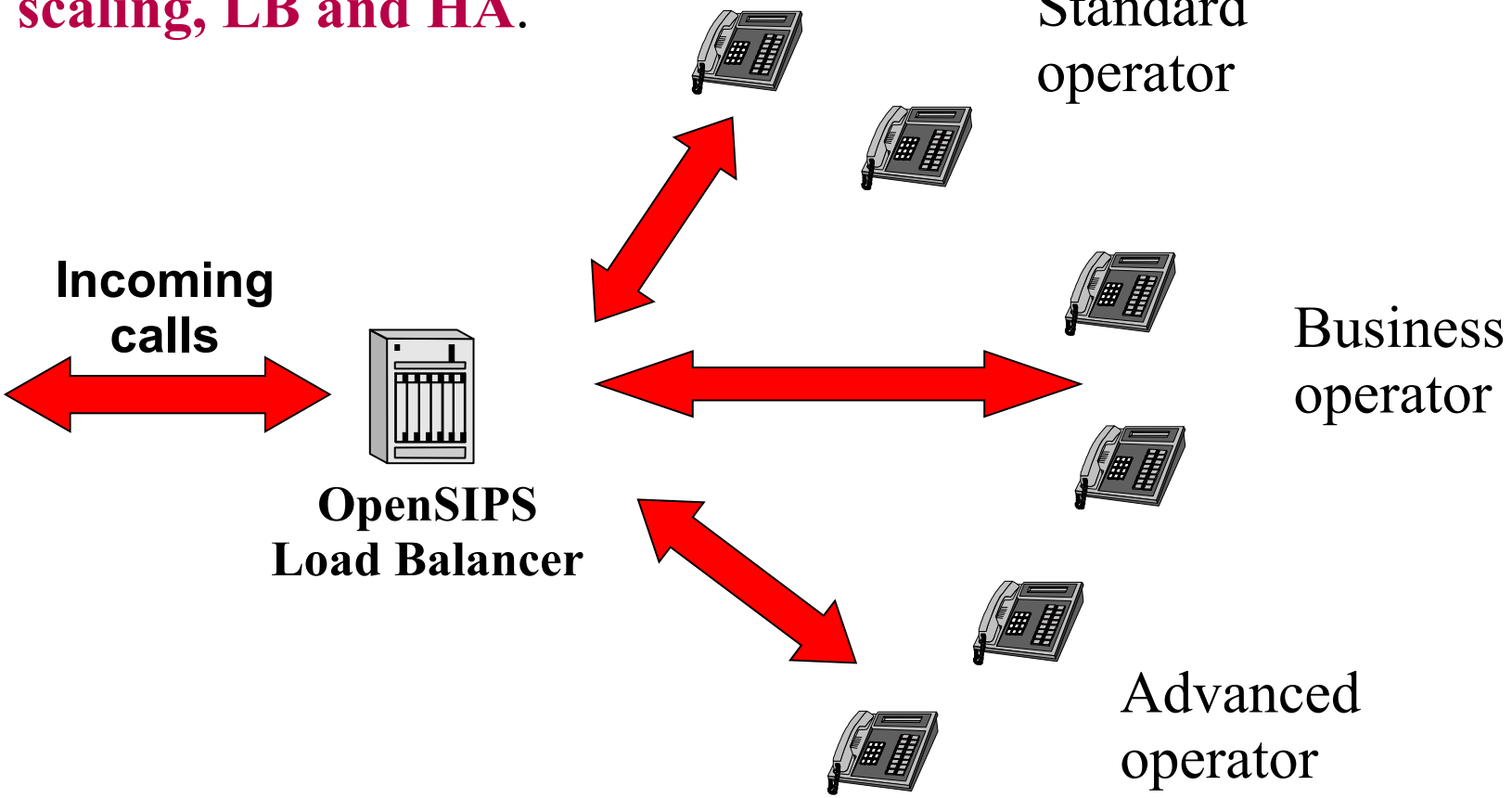
Inbound- service LB

Fronting small/medium size PBX-based services for **scaling, LB and HA**.



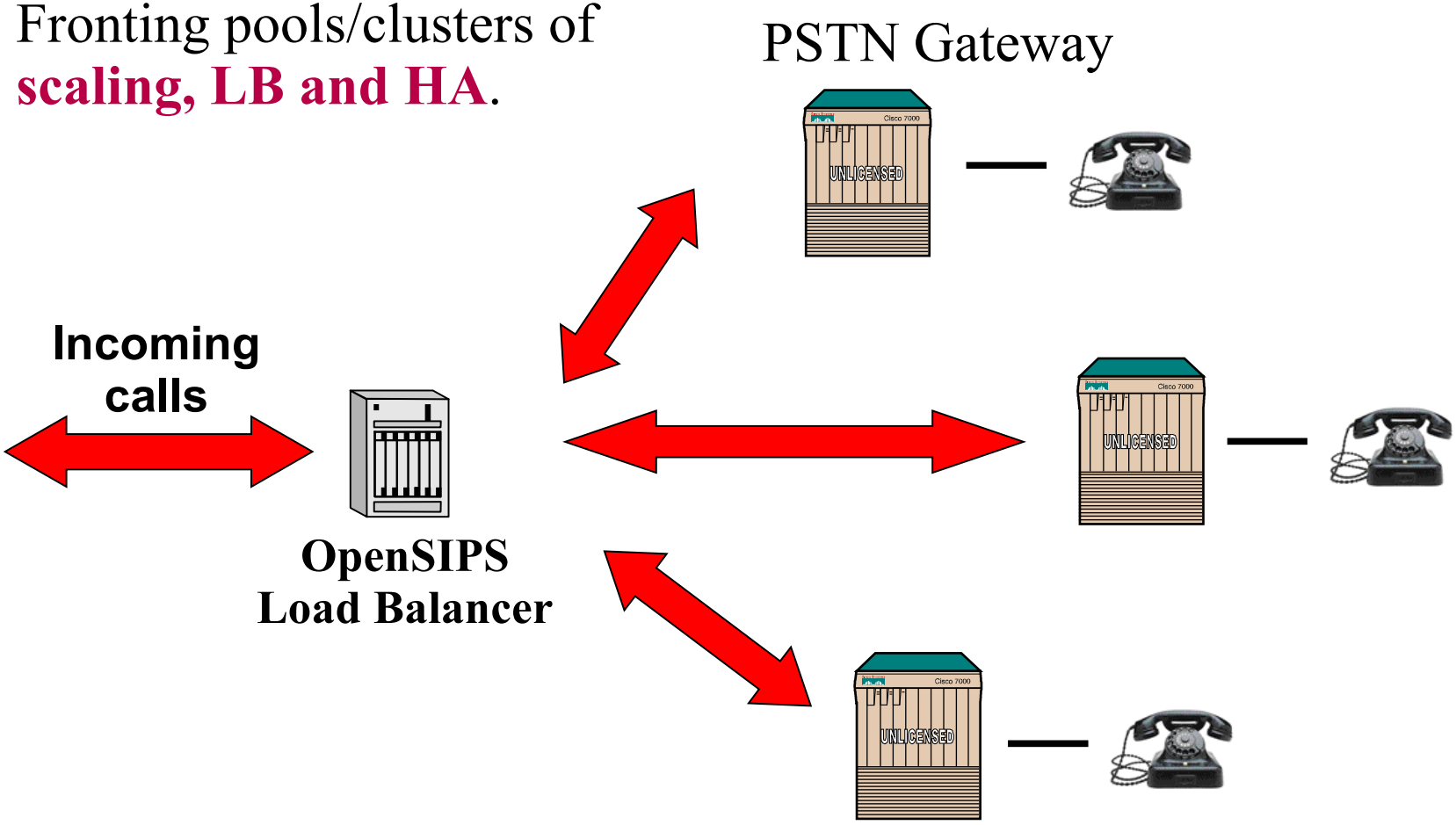
Inbound- Call Center

Controller for a Call Center
scaling, LB and HA.



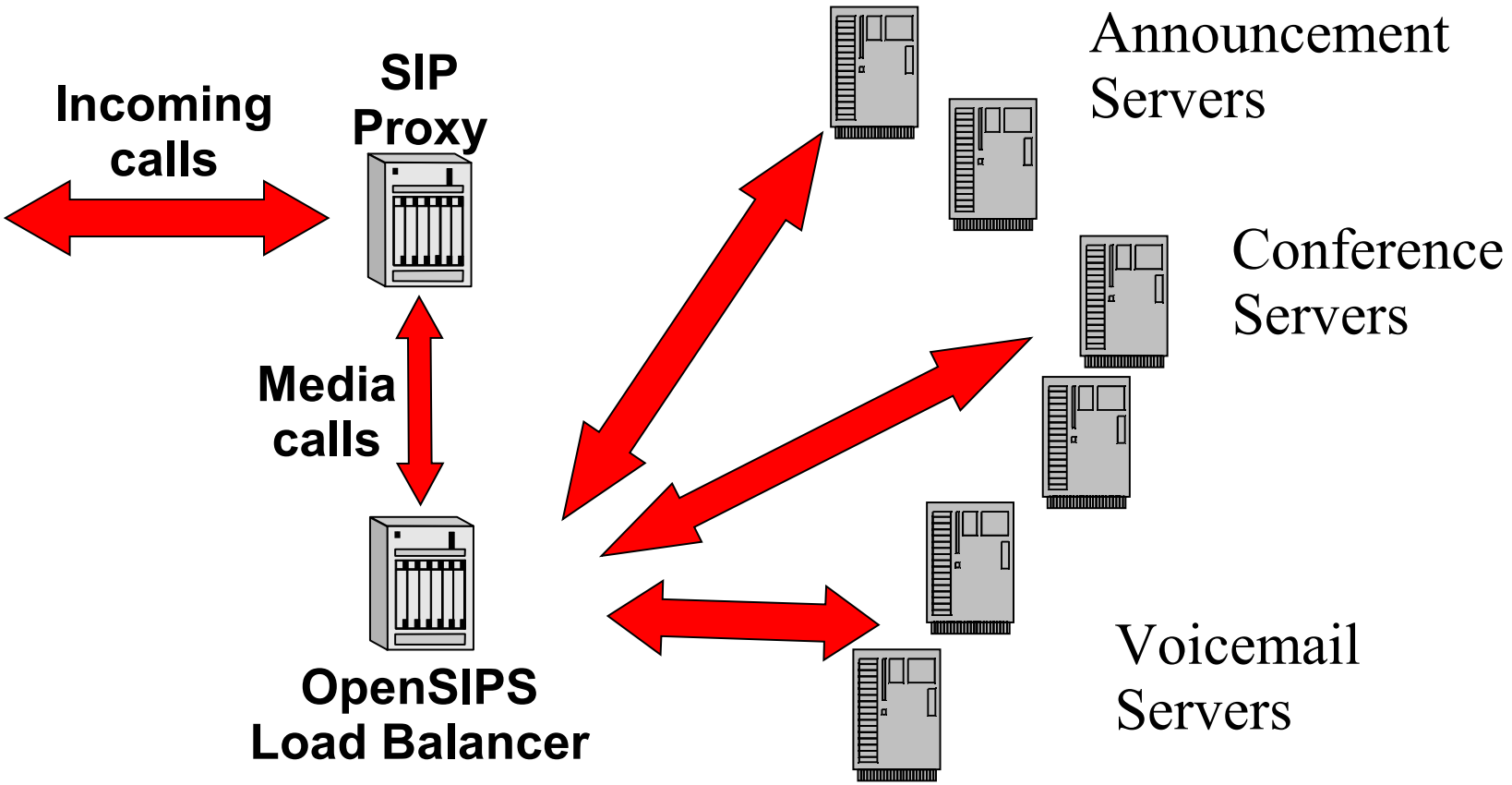
Inbound- PSTN termination

Fronting pools/clusters of **scaling, LB and HA**.



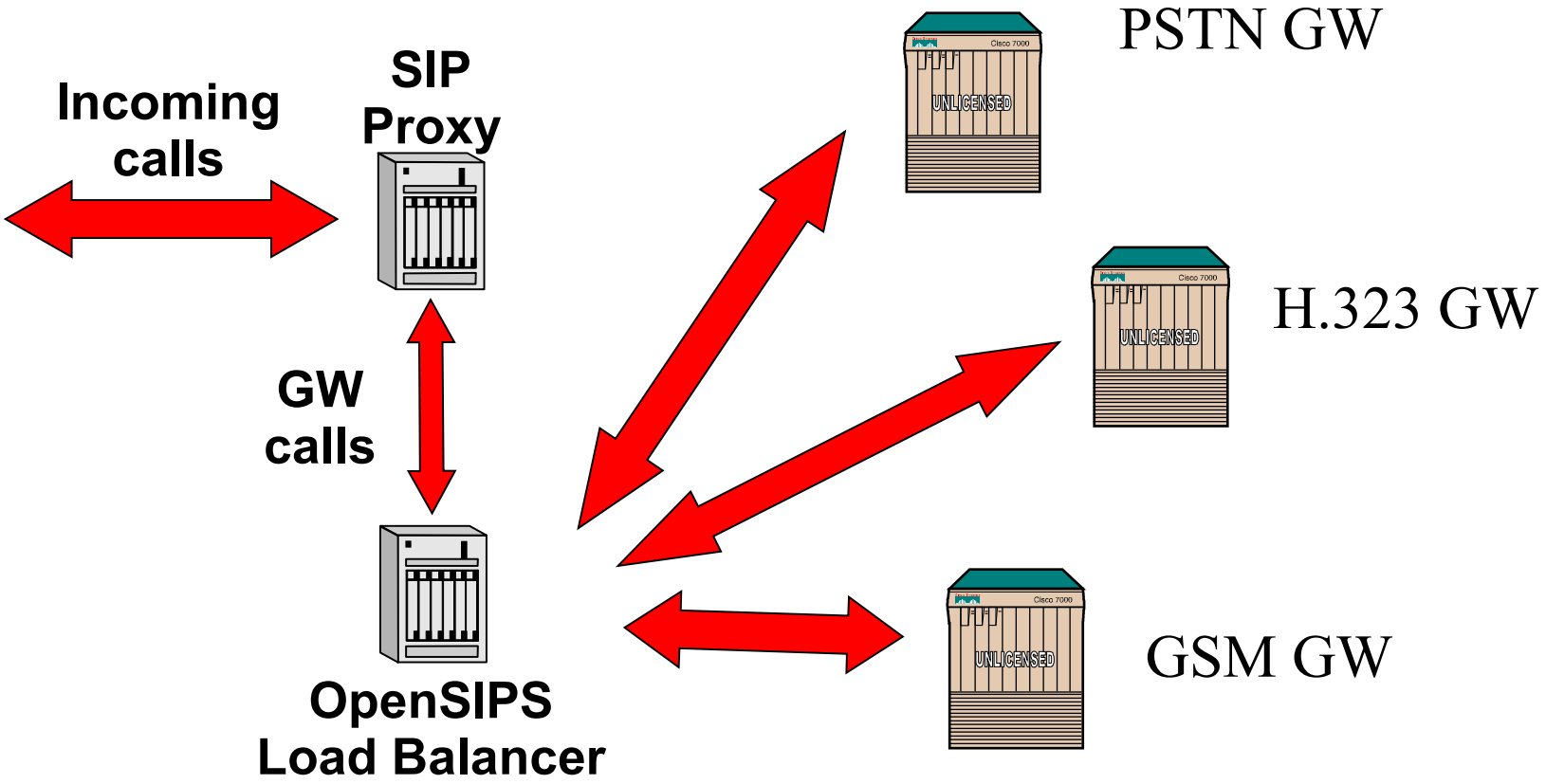
Outbound- Media Services

Fronting pools/clusters of heterogeneous media servers **scaling, LB and HA**.

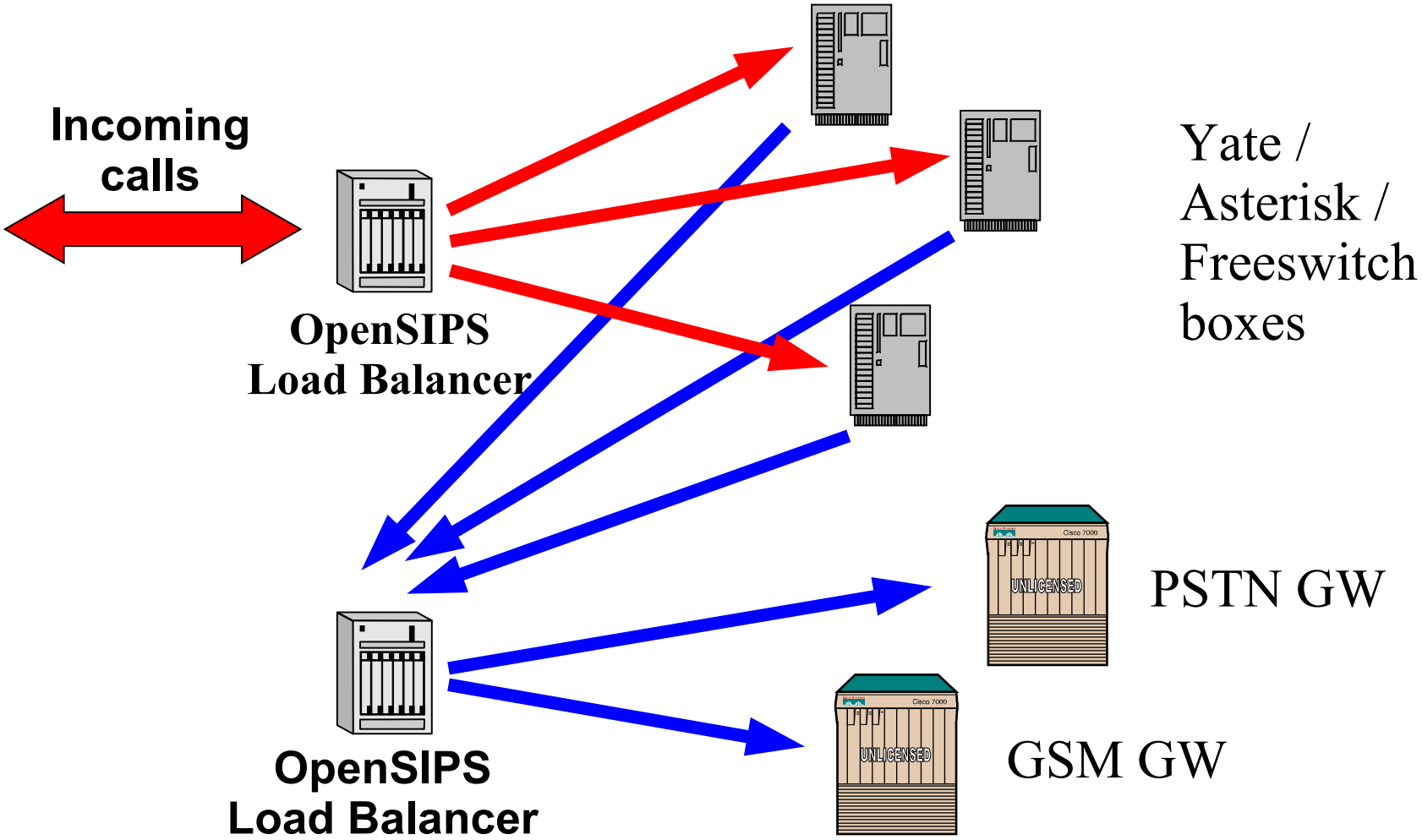


Outbound- GW routing

Routing to pools/clusters of heterogeneous Gateways **scaling, LB and HA**.



Mixed scenario



Implementation details

Implementation Overview

Requires the dialog support (to count the ongoing calls for each peer) → dialog module.

Destinations/peer are identified by their SIP address.

The destinations are not homogeneous:

- may have different capacities (supported calls)
- may offer/provide different resources (services, functionalities)

Resources = capabilities of a destination/peer

Example:

- a set of servers for media related services
- each server may offer a combination of :
  - Transcoding
  - Voicemail
  - Conference
  - Announcement
  - PSTN GW

Mixed peer/destination : each of it may offer a different set of services/resources

LB Group = a set of peer/destinations used for a specific load-balancing scenario.

Example:

- a mixed scenario with inbound and outbound LB
- Group “0” may contain all the PBXs (inbound part)
- Group “1” may contain all the GWs (outbound part)

Peer definition

Definition of each peer/destination contains:

- the set of offered resources
- for each resource, the capacity / maximum load
- group it belongs to
- the address (as SIP URI)

The capacity of a peer is defined as the number of concurrent calls the peer call handler (per resource).

Peer definition - example

Set of 4 peers:

(1) 30 channels for transcoding, 32 for PSTN GW

(2) 100 voicemail channels and 10 for transcoding

(3) 50 voicemail channels and 300 for conferencing

(4) 10 voicemail, 10 conference, 10 transcoding and 32 PSTN GW

ID	Group	SIP URL	Resource
1	1	sip:192.168.2.10	tran=30;pstn=32
2	1	sip:192.168.2.12	vm=100;tran=10
3	1	Sip:192.168.2.15	vm=50;conf=300
4	1	Sip:192.168.2.20	vm=10;conf=10;tran=10;pstn=32

Invoking LB

To invoke the Load Balancing logic, you need:

- the LB group
- the required resources

These are detected in the OpenSIPS routing script, based on whatever information is appropriated.

Example:

- looking at RURI, if PSTN, VM or conference
- looking at codecs from SDP, if transcoding is needed

Load Balancing logic

- Get the set of peers that belong to required group
- Select from the set only the peers able to provide the required resources
- For the selected peers, LB will evaluate the current load for each required resource
- The winning peer is the one with the biggest value for the minimum available load per resource.

LB logic – example I

Set of 4 peers:

(1) 30 channels for transcoding, 32 for PSTN GW

(2) 100 voicemail channels and 10 for transcoding

(3) 50 voicemail channels and 300 for conferencing

(4) 10 voicemail, 10 conference, 10 transcoding and 32 PSTN GW

when calling load_balance("1","transc;pstn") →

Step 1:

only boxes (1) and (4) will be selected at as they offer both transcoding and pstn (required resource)

LB logic – example II

Step 2: Evaluating the load

(1) transcoding - 10 channels used; PSTN - 18 used
(4) transcoding - 9 channels used; PSTN - 16 used

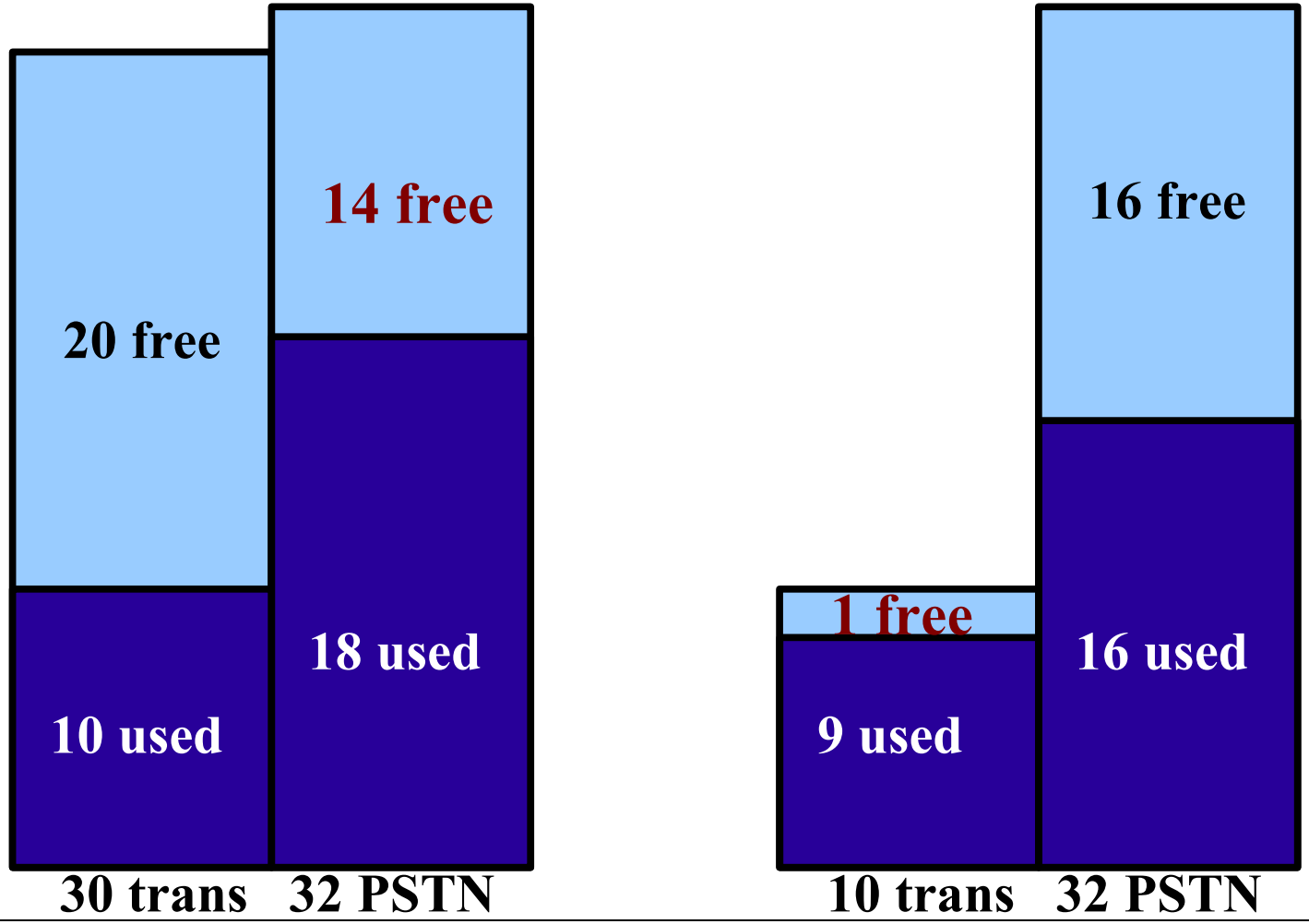
evaluating available load (capacity-load) :
(1) transcoding - 20 channels free; PSTN - 14 free
(4) transcoding - 1 channels free; PSTN - 16 free

Step 3: Evaluating the minimum available load

(1) 14 - for PSTN
(4) 1 – for transcoding

Step 4: Select (1) as it has the biggest minimum available load

LB logic – example III



LB management

LB module provides a command (via Management Interface) for reloading the whole LB data at runtime without loosing affecting the LB process.

This allows addition/removal of peers, changes in resources.

OpenSIPS Control Panel (opensips-cp) will soon release a new tool for LB provisioning.

Runtime resize

LB module provides a command (via Management Interface) to allow resizing the capacity of resource of a peer without requiring a reload of the whole data.

This “resizing” allows LB engine to receive feedback from the peer about their status.

Can be used for :

- Automatic adjusting of the LB engine
- Admin provisioning of LB engine

Example 1:

a GW resource may increased or decreased by an admin following the addition or removal of cards from the GW box.

Example 2:

a monitoring tool running on a peer may trigger the resize (or even disable) of a resource following the detection of a failure on the peer box.

Roadmap

The Snowball effect

Next to be done

More of failure detection:

- Improvement of the failover mechanism
- Peer probing for auto re-enabling of failed peers

Peer management:

- Allow disable/enable of peers via Management Interface
- OpenSIPS-CP tool for web provisioning

Thank you for your attention
You can find out more at www.opensips.org

Questions are welcome