ICC: An Incentive-Compatible Inter-Cloud Communication Traffic Management Mechanism

Manos Dramitinos and George D. Stamoulis
Network Economics and Services (N.E.S.) Group,
Department of Informatics, Athens University of Economics and Business,
76 Patission Str., Athens, GR 10434, Greece
Tel: +30 210 8203693, Fax: +30 210 8203686
Email: {mdramit, gstamoul}@aueb.gr

***Abstract*—In this paper we introduce the Inter-Cloud Communication Traffic Management mechanism (ICC). ICC performs rate control over the ISP transit link(s) aiming to attain a target reduction of transit cost. ICC reduces ISPs transit charge by shaping a portion of the inter-domain traffic (e.g. delay-tolerant inter-cloud traffic), which has been marked as time-shiftable by the traffic source, i.e. by the business customer of the ISP such as a cloud/data center. In particular, ICC reduces the transmission rate of marked traffic at peak 5-min billable intervals and increases it at off-peak, acting at even shorter timescales according to a novel rate-adaptation algorithm. We evaluate numerically ICC by employing real traffic traces. The results reveal that ICC can significantly reduce the ISP transit charge and thus can be a promising and practically applicable solution for ISPs while also being beneficial for their customers whom the ISP should incentivize by sharing its savings.**

I. Introduction - Motivation

Internet services such as video, gaming, are used by millions of geographically dispersed people. Clouds, Content Delivery Networks (CDNs), and online social networks offer complex services where resources, data and service points are virtualized and instantiated at/migrated to multiple locations so as to efficiently serve the users. Inter-cloud communication is thriving [1] also due to the emerging federations of clouds and CDNs such as CDNI [2], OnApp [3], XIFI [4], Zentera [5]. This "back-office" Web traffic does not pertain to actual users' flows and as argued in [6] is a large portion of the Internet inter-domain traffic, being up to 30% and occasionally 40-50% of the total traffic of popular Internet transit links. Similar findings over Telefonica's infrastructure are reported in [7], where a simple store-and-forward algorithm for bulk data transfers among datacenters of different time zones attains substantial cost savings. Hence, *a large portion of transit traffic is not time-critical and substantial transit cost savings can be attained for ISPs by smart traffic management*.

In this paper we propose ICC, an incentive-compatible inter-cloud communication optimization mechanism, based on the fact that both the ISP and its customer can highly benefit if the ISP shapes delay-tolerant traffic - referred to as "time-shiftable" (or manageable) - because he can thus reduce the transit link 95^{th} percentile and thus the associated transit cost.

II. Our Approach: The ICC Mechanism

The Inter-Cloud Communication mechanism (ICC), as also indicated by its name, has been designed to manage Inter-Cloud Communication/Web back-office traffic in an ISP-friendly way. The rationale of ICC is as follows: Clouds/data centers know their customers' traffic: A portion of the traffic is related to interactive or the so-called "zero-tolerance" cloud services with strict delay requirements and for which no quality degradation is acceptable. At the same time, bulk data transfers for backups, replication of data among CDN caches etc., are delay-tolerant. This traffic can be shaped essentially without any negative impact for the cloud or its users but with potentially significant positive impact on reducing the home ISP's traffic peaks and the resulting transit charge. Thus, the home ISP can incentivize its customers (clouds/data centers/CDNs) to mark their time-shiftable traffic, so that it can be shaped over the transit link. ISP's customers determine *which* portion of the traffic will be managed by ICC. The algorithm developed in this paper for this purpose is named STraS (Shiftable Traffic Scheduling). STraS determines *when* it is beneficial for the ISP to send this traffic and at *which rate*. The STraS specification, its properties and evaluation are the main contributions of this paper. Note that ICC is designed to operate on the transit link aggregates without inspecting individual flows, which ensures its scalability and applicability in real networks.

Since for some inter-cloud data transfers, e.g. data replication, there are multiple candidate destinations, ICC comprises a cloud layer that selects the *optimal destination* for such transfers, by considering the sender's constraints and preferences, and the network and cloud current load and performance metrics. Invoking the ICC cloud layer functionality is optional. ICC is in accordance with the design-for-tussle principle [9]: The clouds and data centers, rather than the ISP, make all the cloud-related decisions (i.e. selection of the destination and marking of traffic as shiftable). Though this paper focuses on the ICC traffic management and the STaS algorithm, the ICC cloud layer is also briefly presented for completeness reasons. A representative instantiation of the ICC mechanism, depicting its constituent components, is provided as Figure 1.

Shiftable Traffic Scheduling (STraS) implements the ICC smart rate adaptation algorithm at the ISP transit link(s) for the portion of the aggregate inter-domain traffic that is marked as time-shiftable. STraS, combined with the identification of the time-shiftable traffic at the traffic source (via marking such as DiffServ codepoint, VLAN id) constitute the minimum - yet still highly powerful and effective - ICC configuration and deployment. All the other ICC components are optional.

Smart information Service (SmaS) computes the mini-

978-3-901882-77-7 © 2015 IFIP

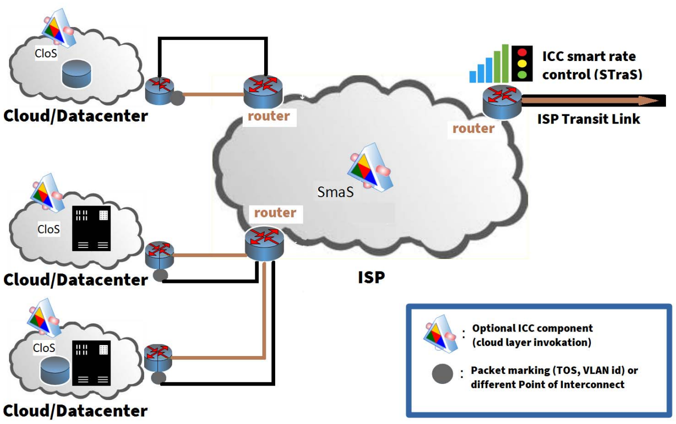

Fig. 1. ICC network and cloud layer components.

mum network path cost P_d for the paths connecting the ISPs domain with the IP address of each candidate datacenter traffic sink d. We specify P_d to be the BGP hop count, since this information is readily available and non-volatile over time, as opposed to Quality of Service (QoS) metrics.

Cloud Scheduler (CloS) provides to SmaS a list of destinations and receives the network path cost P_d for each d. CloS selects the optimal destination $d^\star$ so as to minimize the inter-cloud communication flow total cost $c_d(P_d, L_d)$ over $< d > \in D$: $d^\star = argmin_{d \in D}\{c_d(P_d, L_d)\}$. c_d is a function of i) the network "cost" P_d provided by SmaS, and ii) the cloud cost L_d of destination d; L_d can be a monetary cost or a proxy of energy cost and load. In line with ALTO [8], we assume P_d and L_d to be abstract costs, depicting the difficulty to handle the flow at d. Hence: $d^\star = argmin_{d \in D}\{\alpha \cdot P_d + \beta \cdot L_d\}$, where α, β are the relative weights of the network and cloud costs.

III. ICC Traffic Management

A. The ICC STraS Algorithm

Shiftable Traffic Scheduling (STraS), depicted as Figure 2, is the ICC traffic management algorithm. It is employed by an ISP to reduce the 95^{th} percentile transit charge by shaping the rate of a portion of this traffic (e.g. delay-tolerant cloud traffic) that has been marked accordingly by the ISP's business customer (e.g. cloud/datacenter), shifting its transmission at off-peak intervals; real-time traffic is not further delayed. The STraS module receives as input the target 95^{th} percentile to be attained, the number of times/epochs y that STraS will be invoked in the current 5-min interval now, $1 \leq now \leq end$; end denotes the number of 5-min intervals within the billing period (typically month); $epoch\_index$ denotes the index of the epoch within the 5-min interval now, i.e. $1 \leq epoch\_index \leq y$ and $cagr$ is the cumulative aggregate growth rate of traffic.

To apply ICC STraS, the ISP initially selects the target 95^{th} percentile C_{target} based on traffic history and STraS decides on the volume of time-shiftable traffic sent over the transit link in each 5-min interval of the billing period as follows: each 5-min interval is sliced in y *epochs*; e.g. for $y = 10$, each interval is sliced in 10 30-sec epochs. At each epoch start, the algorithm decides how much time-shiftable traffic is sent (lines 17-34 of STraS) using certain thresholds $epoch\_tholds[]$, one

```
1:  procedure STraS(C_target, y, cagr, now, epoch_index)
2:      if now > (0.95 · end) then    ▷ Interval now is in the last 5% 5-min intervals of the billing period
3:          n ← &num_intervals_C_target_violated()
4:          if now ≤ (end − n) then return −1   ▷ -1 denotes no rate control needed for time-shiftable traffic
5:          end if
6:      end if
7:
8:      ns_rate_historical[1, ..., y] ← queryDB(now, epoch_index);   ▷ Get historical rates of non-shiftable traffic from traffic statistics
9:      ns_rate_previous ← queryDB(now, epoch_index);   ▷ Get the non-shiftable traffic rates of the previous epoch
10:
11:     for j ← 1, y − 1 do
12:         epoch_tholds[j] ← 0.9
13:         limit[j] ← epoch_tholds[j] · C_target
14:     end for
15:     epoch_tholds[y] ← 1        ▷ Different threshold for y
16:
17:     for j ← 1, y − 1 do
18:         ns_expected ← ((1+cagr)·ns_rate_historical+ns_rate_previous)/2
19:         if ns_expected < limit[j] then return limit[j] − ns_expected ▷ Max costless time-shiftable traffic to send
20:         else
              return 0
21:         end if
22:     end for
23:
24:     if epoch_index = y then    ▷ Different logic for the last epoch (y) of each 5-min interval
25:         real_volume ← 0
26:         for j ← 1, y − 1 do
27:             real_volume ← real_volume + &query_total_traffic_sent(now, j)
28:         end for
29:         limit[y] ← (epoch_tholds[y] · C_target · 300 − real_volume) · (y/300)
30:         ns_expected ← ((1+cagr)·ns_rate_historical+ns_rate_previous)/2
31:         if ns_expected < limit[j] then return limit[j] − ns_expected ▷ Max costless time-shiftable traffic to send
32:         else
              return 0
33:         end if
34:     end if
35: end procedure
```

Fig. 2. The ICC STraS algorithm.

per epoch index (lines 11-15 of STraS). The thresholds help to ensure that the time-shiftable traffic plus the real-time traffic (unknown a priori but estimated) will not exceed the C_{target} rate over the corresponding 5-min interval. The values of these thresholds depend on the confidence of the ISP regarding the accuracy of his traffic statistics and predictions. In an attempt to fix potential errors in this estimate, the algorithm a) uses both historical traffic statistics and the actual rate metered in the previous epoch (lines 18 and 30 of STraS) and b) the sending rate of the time-shiftable traffic at the last epoch (y) of the 5-min interval takes into account what was actually sent in the previous ($y-1$) epochs (lines 25-29 of STraS).

STraS also uses the fact that C_{target} can be violated at no cost in as many as 5% of the 5-min intervals of the billing period, due to the 95^{th} percentile rule: In the pseudocode of Figure 2 this 5% pertains to the last 5% intervals of the billing period minus the number of intervals where the algorithm has exceeded the desired target C_{target} (lines 2-6 of STraS). This basic ICC definition is henceforth referred to as "ICC Fixed At max rate at the end" (ICC_FA). An alternative variation named "ICC Fixed At Peak periods with max rate (ICC_FAP) select the intervals with top 5% values throughput, assuming that these are known to the ISP. ICC_FAP has typically inferior performance to that of ICC_FA in terms of the 95^{th} percentile attained, since ICC_FAP is is prone to the errors that are due to false predictions on the peak intervals. In case both ICC_FA and ICC_FAP can always accurately predict expected traffic, then they both attain the same 95^{th} percentile, i.e. C_{target}; see Section V. Even then, ICC_FAP increases the load of the transit link at peak intervals as opposed to to ICC_FA, which transmits with max rate only when there exists queued manageable traffic towards the end of the billing period, and typically removes time-shiftable traffic from peak intervals, thus further benefiting QoS-sensitive traffic. Therefore, we have opted for the specification of Figure 2, i.e. ICC_FA.

B. Fundamental Properties

1) Granularity of control: Splitting the 5-min interval to y epochs allows fast adaptation, finer rate control than related works, and good performance under unforeseen traffic patterns.

2) Simplicity-Scalability: ICC is designed to operate on traffic aggregates, without providing per-flow management or guarantees, which would be unrealistic for the transit link.

3) Implementation: Implementation is on-going in [10] using the Juniper MX240 router *premium policer* and *aggregate policer* functionality for policing and dynamically modify the rate of the router's dedicated queue for time-shiftable traffic.

4) Optimization Potential: The attained transit cost savings depend inevitably on the portion of the traffic that is manageable by ICC, the traffic pattern, mix and variability over time.

5) Net Neutrality: Theoretically, ICC could manage many types of non time-critical (background) traffic. However, we restrict attention to inter-cloud traffic and specify that marking the time-shiftable traffic should be done by the traffic sources, i.e. the ISP's business customers (clouds, datacenters, CDNs) themselves. incentivized via pricing.

6) Incentive-Compatible Pricing: ICC pricing incentivizes the ISP business customers to allow ICC to manage the time-shiftable portion of their traffic. In particular, we specify that the ISP will return a cut p, where $0 \leq p \leq 1$, of the total transit cost savings $ISPsavings$ attained due to ICC, to each one of his customers $i \in I$ that accepted ICC traffic management for a portion of their traffic of volume vol_i as a discount to their billing charge by the ISP. Hence, each customer's discount will be proportional to his fair share on the total volume of traffic managed by ICC, as indicated in equation (1).

$$discount_i = p \cdot ISPsavings \cdot \frac{vol_i}{\sum_{j \in I} vol_j} \quad (1)$$

This formula has been chosen since it is simple, increasing in the volume of data offered to ICC, and does not require traffic metering or monitoring. This rationale also affects how we envision that $ISPsavings$ could be computed: Computing accurately the value of the resulting transit link 95^{th} percentile under both the presence and absence of ICC, denoted as $perc_{ICC}$ and $perc_{BE}$, would be infeasible in practice since ICC inevitably alters the traffic profile and the resulting charge in a real ISP network. Therefore, and in order to keep the computational complexity low, we opt for the approach that $perc_{BE}$ is computed on the basis of traffic statistics by multiplying the $perc_{BE}$ of a billing period of a previous year when ICC was not operating times the cumulative aggregate growth rate.

IV. Related Work

Related works for reducing the ISP's 95^{th} percentile are NetStitcher [7] and Dynamic Traffic Management [11].

NetStitcher focuses on inter-datacenter communication. It uses unutilized resources across datacenters and backbone links, taking advantage of time-zone differences for bulk transfers, e.g., data replication among datacenters. In particular, NetStitcher uses a store-and-forward algorithm to *schedule* data transfers. Scheduling is driven by predictions on the availability of leftover bandwidth at links and storage constraints at storage relay nodes. NetStitcher operates per flow, as opposed to ICC which operates on inter-domain traffic aggregates; NetStitcher requires many dispersed relay nodes in order to be deployed and it performs well only for data transfers where the source and destination are significantly far apart, across several time zones. Thus, its scope and applicability are much more narrow than that of ICC.

Dynamic Traffic Management (DTM) reduces the 95^{th} percentile costs of a *multi-homed ISP*[1] by splitting optimally the traffic among his transit links. Thus, if peak periods occur simultaneously at all the transit links of multi-homed ISPs, this approach has its limitations in the gains it can achieve. The DTM rationale is complementary to ICC, since both are about shifting some traffic: DTM shifts traffic in space (links) and ICC in time. Therefore, it is possible to combine DTM and ICC for multi-homed ISPs in order to achieve joint optimization by applying DTM to optimally split the traffic among the transit links and then invoke ICC for attaining even lower 95^{th} percentile per transit link. In fact, an integrated implementation of DTM and ICC was carried out in [10].

Compared to related work, ICC has certain unique features: ICC, as opposed to NetStitcher, operates in small time scales (secs), resulting in fast rate adaptation and thus in significant savings even in cases where ISP's prior expectations regarding traffic are inaccurate. ICC is deployed at the ISP transit link without requiring backbone relay nodes or any additional components at the customer premises, as opposed to NetStitcher and DTM. ICC provides incentives to the ISP's

[1] A multi-homed ISP purchases global Internet connectivity (transit) from at least two higher-tier (typically Tier-1) ISPs.

business customers to endorse ICC and operates on the transit link traffic aggregate rather than relying on rate control of each inter-cloud source-destination pair flow, as is the case for both NetStitcher and DTM. Finally, ICC is applicable to both multi- and single-homed ISPs.

V. Assessment

A. Methodology

We conducted a plethora of experiments to assess ICC. In these experiments, we employed actual network traces from two European ISPs and the Greek research network (GRnet) and under different information sets. The duration of these traces varies, ranging from hours to several weeks. We have implemented in Java an evaluation framework capable of processing traces and comparing the performance of ICC with that of Best Effort without ICC. Since we have not done packet-level simulations, our approach cannot capture the extra benefit for the real-time (non-manageable) traffic in terms of performance due to the shifting of load from peak to off-peak epochs. Nevertheless, this additional gain of ICC can be deciphered from the resulting traffic patterns.

We have performed a step-by-step evaluation, focusing on the performance of the STraS algorithm. Initially, we assume *full information*, i.e. the ISP can accurately predict the expected real-time traffic per epoch. Thus, $ns\_expected$ (line 30 of StraS) is always equal to the ex post revealed $real\_volume$. This allows us to initially assess ICC, in particular both ICC_FA and ICC_FAP introduced in Section II, without the inevitable traffic prediction errors. Thus, we can better understand the main properties of ICC and the key performance issues. Next, we introduce the traffic learning and estimation aspects of STraS (lines 27-31 of STraS) and assess the ICC performance - this ICC variation is henceforth referred to as ICC_STATS - and the impact of key parameters such as C_{target}, y, $epoc_t holds$, also performing a sensitivity analysis over them for multiple traces.

B. Results: ICC with full information

We consider the case of a transit link of a European ISP, containing the throughput for a total duration of one week. The value for the 95^{th} percentile is 2894456 bps. Then we apply ICC_FA and ICC_FAP for $y = 10$ and $C_{target} = 2577000$bps, i.e. an 11% reduction of the 95^{th} percentile is sought. This is not necessarily the optimal target value by any definition, just an attempt to see if we can indeed achieve a such considerable reduction; the manageable portion of the transit link traffic varies over time and is approximately 18% of the total traffic volume throughout the trace duration. ICC operates over $y = 10$ epochs, i.e. traffic shaping is performed per 30sec with different thresholds over the rate for the first 9 of 10 total epochs per 5-min interval. Hence the shape of the traffic when shaping is performed is as depicted in Figure 3. In Figure 4, we compare the traffic pattern of ICC_FA and Best Effort without ICC for a "zoomed" subset of samples (two days) so as to better illustrate the effect of the STraS traffic shaping on the traffic pattern: It is evident that values higher than C_{target} are reduced so that each group of 10 30-sec epochs that constitute a 5-min billing interval has an average rate of at most C_{target}; also the first 9 values are different than the tenth, due to the setting of $epoch\_tholds[]$. Note that there are 20160 values at the plots since for presentation reasons we plot the average rate per epoch, i.e. there are 20160 samples of 30-sec duration per week.

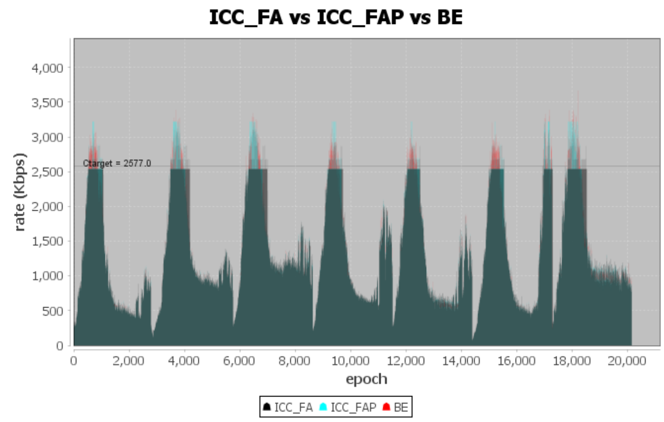

Fig. 3. The traffic patterns attained with and without ICC.

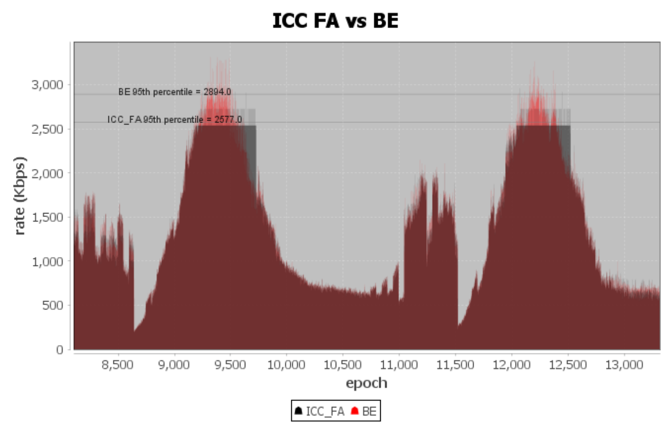

Fig. 4. The effect of ICC traffic shaping at peak intervals.

We also compare ICC_FAP and ICC_FA in Figure 5 where it is shown that ICC_FAP injects more traffic during peak intervals, as opposed to ICC_FA which would opportunistically send with maximum rate only at the end of the billing period and only if queued manageable traffic exists in the transit link buffer waiting to be sent. The latter condition does not apply for this trace, due to the relatively low value of the average throughput compared to the C_{target} value. This is also why ICC_FAP results in much less extra delays for the manageable traffic than ICC_FA although the reverse applies to the non-manageable traffic. Note that to estimate the delay of the non-manageable traffic, we increase a counter initially set to 0 every time one bit is delayed per one epoch of $300/y$ secs due to the STraS rate control. This means that 1 bit delayed for 10 epochs will increase the counter by 10. The respective values of the total delay counters are $7.68 \cdot 10^{11}$ for ICC_FA and $1.60 \cdot 10^{11}$ for ICC_FAP, thus ICC_FA causes roughly 5 times total extra delay for manageable traffic compared to ICC_FAP.

The evaluation indicates that the C_{target} goal is always met by ICC assuming that there is accurate prediction of the expected real-time traffic. ICC can shape the traffic over the transit link according to the values of C_{target} and the traffic

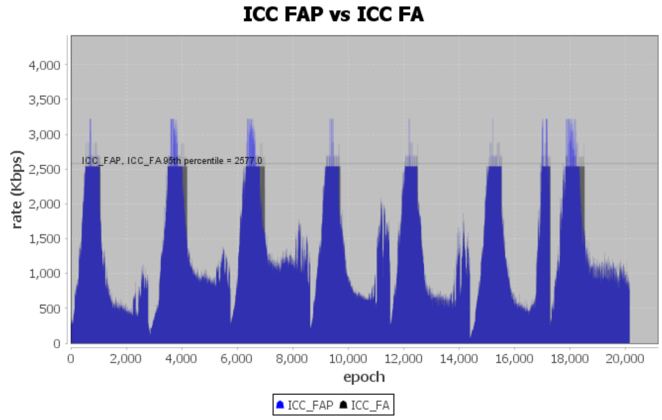

Fig. 5. The traffic pattern attained under ICC_FAP and ICC_FA.

threshold values of the y epochs. Thus, the ICC traffic patterns are predictable, while both ICC and Best Effort Internet exhibit time-of-day traffic patterns periodicity. Moreover, the more aggressive the setting of C_{target} is (i.e. the lower its value selected given the utilization of the link) the higher the respective delays and delay penalties for manageable traffic are, since more data are delayed so as not to violate C_{target}. Indeed, a low value of C_{target} - compared to the value of the Best Effort 95^{th} percentile - indicates that the effective rate under ICC will be considerably lower. This can be attained by increasing the number of intervals and the respective number of packets that will be queued and thus delayed in all these intervals where the link is utilized at values beyond C_{target}. On the contrary, if the C_{target} value is higher than the average link throughput, it is likely that ICC_FA will not have to transmit with the maximum rate at the end of the billing period if the link is not heavily utilized at these epochs. This also shows that even higher transit cost savings can be attained for this experiment, since the variability of the traffic is quite high and ICC_FA does not even send at the maximum rate in the last intervals of the billing period (here week).

C. Results: ICC with statistics-based traffic expectation

We now proceed to assess the ICC mechanism when using statistics to predict the non-manageable traffic per epoch, i.e. the ICC_STATS variation. ICC_STATS pertains to the actual case where an ISP would deploy ICC at its transit link(s).

This experiment is run over a trace file of a European provider. The trace is of one-week duration and the portion of the time-shiftable traffic is roughly 20%. Since ICC_STATS relies on historical data in order to predict the expected traffic per epoch, we use the first day of the trace as the training set. Hence, the first day comprises the historical data that will be used by ICC_STATS in order to make predictions for the other 6 days of the trace to follow. Clearly, the inevitable differences with the traffic patterns of the days to follow will cause ICC_STATS to under-estimate or over-estimate the expected traffic to handle, thus deviating from the optimal traffic management of ICC_FA, which is assumed to have perfect information. Both variations of ICC, namely ICC_FA and ICC_STATS are configured to run $y = 10$ times, i.e. every 30 seconds within each 5-min interval. For the experiments of this subsection and in order to facilitate the reader we set all 10 $epoch\_tholds[]$ values to the same value per run, e.g. all values are set to 1. Thus, we can simplify the cross-comparison of different runs and better understand the impact of this parameter on the ICC_STATS performance. Setting different values for each of the ten $epoch\_tholds[]$ values would result in a very large set of combinations for the sensitivity analysis to be performed over $epoch\_tholds[]$, thus greatly increasing the complexity of our analysis. Clearly though the most important value is the value of $epoch\_tholds[y]$, i.e. the last value (line 15 of STraS); indeed, this value determines how closely to C_{target} in the rate of the entire 5-min interval can be (line 29 of STraS). A high value for this last-epoch threshold indicates the ISP's confidence on the quality of the traffic statistics and on clear periodicity properties of the transit link traffic patterns. This means that if $epoch\_tholds[y] = 1$ takes its highest possible value, namely 1, any predictions by ICC_STATS, underestimating the non-manageable traffic pattern will result in violation of C_{target}, and hence reduce the savings of transit cost attained. On the contrary, more conservative choices of lower values of $epoch\_tholds[y]$, e.g. 0.90 , can remedy this effect, but on the other hand ICC can thus introduce higher delays to the manageable trafficICC by constraing more the rate than it is actually needed to in order to attain C_{target}. Determining the "optimal" $epoch\_tholds[]$ values cannot be computed analytically a priori. In practice, the ISP would do a trial-and-error process across billing periods in order to fine-tune this parameter and enhance ICC performance.

TABLE I. ICC_STATS WORST-CASE PERFORMANCE ASSESSMENT

	Attained 95^{th} Percentile $(epoch\_tholds[1,...,y] = 1)$				
BE - no ICC	ICC_FA	ICC_FAP	ICC_STATS	C_{target}	ICC_STATS deviation(%)
4674870	3285000	3285000	3607089	3285000	9.80
4674870	3559000	3559000	3847349	3559000	8.10
4674870	3833000	3833000	4077187	3833000	6.37
4674870	4107000	4107000	4260687	4107000	3.74
4674870	4381000	4381000	4417069	4381000	0.82
4674870	4654000	4654000	4653527	4654000	-0.01
					Average: 4.80%

The comparison results for the worst case for ICC_STATS, i.e. when the $epoch\_tholds[]$ values are all set to 1 for $C_{target} = 4381000$ are provided as Table I. The ICC_STATS performance is rather satisfactory: The maximum deviation is 10% for the lowest value of C_{target} sought, the deviation improves as C_{target} increases, and the average deviation computed over all C_{target} values is 4.80%. It is important though that ICC_STATS always manages to attain better results compared to Best Effort without ICC and the 95^{th} percentile attained is close to the C_{target} sought (yet higher than it) even for the worst case. As expected, since they rely on full information, ICC_FA and ICC_FAP always attain the value of C_{target} sought. Hence, we refrain from including them in the results to follow for brevity reasons.

Moreover, we have investigated the sensitivity of ICC_STATS to the statistical properties of the manageable traffic. When using traces, the statistical properties of the manageable and non-manageable traffic are in principle unknown. A way to remedy this would be to generate for comparison reasons artificial traces of manageable traffic, which however are related to those found in the actual traces available. We do this as follows: for each experiment, after reading the throughput values of the manageable traffic from the respective trace file, we generate a trace with the same total traffic volume sent in the billing period but with the individual rate values of manageable traffic per epoch being replaced: In particular, the trace samples are replaced in the experiment with values uniformly randomly generated from the $[0.8, 1.2]$ interval times the average throughput of manageable traffic. We then re-run ICC_STATS with this “artificial” trace. Multiple experiments indicate that the statistical properties of this portion of the traffic have some yet somehow limited impact on the results attained in terms of average values and do not alter the qualitative findings of the experimental assessment. This is due to the epoch-by-epoch aggregation and queueing of the time-shiftable packets that is done by ICC during peak intervals when rate control is activated. This gradually mitigates the impact of traffic variability of the manageable traffic, with the only decisive factor being whether there exists substantial volume of time-shiftable packets queued or not. For the experiment of Table I, the average ICC_STATS deviation is reduced to 3.61% (from 4.80%) with the respective values per run being $6.83\%, 5.4\%, 4.12\%, 2.95\%, 1.91\%, 0.46\%$.

Table II depicts a sensitivity analysis of ICC_STATS for various values of C_{target} and $epoch\_tholds[]$, which is taken as fixed for all epochs. Results indicate that different values of $epoch\_tholds[]$ for a given C_{target} value result in different performance of the mechanism in terms of the 95^{th} percentile attained. This is reasonable, since the prediction errors on the future traffic to be handled have different impact depending on $epoch\_tholds[]$. Note that the maximum deviation is always less than 10% and typically much less. For several runs, there is negative deviation of the cost reduction attained by ICC_STATS from the C_{target} goal. This indicates that a larger discount was actually attained by ICC_STATS than the one originally sought. This applies particularly for larger values of C_{target} and smaller values of $epoch\_tholds[]$.

The assessment is also affected by the quality of the traces used. The 1-week long traces used, due to the inherently different traffic patterns per day of week result in more erroreneous predictions by ICC_STATS than the case where multiple-week traces would be available with the first week being the training set. However, the mechanism's key benefits are evident: ICC_STATS **always** results in much lower 95^{th} percentile than Best Effort without ICC with a typically low deviation from the C_{target} sought, also benefiting real-time traffic due to lower bursts. The ISP can employ ICC starting with a conservative setting of the $epoch\_tholds[]$ values, e.g. in the area of $[0.85-0.90]$ and gradually fine-tune these values via a trial-and-error among the consecutive billing periods.

TABLE II. ICC_STATS SENSITIVITY ANALYSIS

Attained 95^{th} Percentile			
$epoch\_tholds[] = 0.85$			
BE - no ICC	C_{target}	ICC_STATS	ICC_STATS deviation(%)
4674870	3285000	3268679	-0.50
4674870	3559000	3437469	-3.41
4674870	3833000	3627181	-5.36
4674870	4107000	3834504	-6.63
4674870	4381000	4036428	-7.86
4674870	4654000	4212442	-9.48
			Average: -5.54%
$epoch\_tholds[] = 0.90$			
BE - no ICC	C_{target}	ICC_STATS	ICC_STATS deviation(%)
4674870	3285000	3381226	2.92
4674870	3559000	3582225	0.65
4674870	3833000	3797732	-0.92
4674870	4107000	4011633	-2.32
4674870	4381000	4201133	-4.11
4674870	4654000	4356671	-6.39
			Average: -1.69%
$epoch\_tholds[] = 0.91$			
BE - no ICC	C_{target}	ICC_STATS	ICC_STATS deviation(%)
4674870	3285000	3410791	3.82
4674870	3559000	3609757	1.42
4674870	3833000	3831876	-0.03
4674870	4107000	4047543	-1.45
4674870	4381000	4221327	-3.64
4674870	4654000	4384063	-5.80
			Average: -0.94%
$epoch\_tholds[] = 0.92$			
BE - no ICC	C_{target}	ICC_STATS	ICC_STATS deviation(%)
4674870	3285000	3434814	4.56
4674870	3559000	3639854	2.27
4674870	3833000	3866020	0.86
4674870	4107000	4080928	-0.63
4674870	4381000	4253913	-2.90
4674870	4654000	4404175	-5.37
			Average: -0.20%
$epoch\_tholds[] = 0.93$			
BE - no ICC	C_{target}	ICC_STATS	ICC_STATS deviation(%)
4674870	3285000	3454156	5.15
4674870	3559000	3671885	3.176
4674870	3833000	3900517	1.76
4674870	4107000	4107957	0.02
4674870	4381000	4282814	-2.24
4674870	4654000	4421640	-4.99
			Average: 0.48%
$epoch\_tholds[] = 0.95$			
BE - no ICC	C_{target}	ICC_STATS	ICC_STATS deviation(%)
4674870	3285000	3510412	6.86
4674870	3559000	3735947	4.97
4674870	3833000	3966478	3.48
4674870	4107000	4174374	1.64
4674870	4381000	4338118	-0.98
4674870	4654000	4464214	-4.078
			Average: 1.98%

VI. CONCLUSIONS

We have introduced, specified and assessed the Inter-Cloud Communication Traffic Management mechanism (ICC), a novel traffic management mechanism that controls the ISP transit link(s) throughput so as to attain a target transit cost. ICC is designed to manage inter-cloud and back-office Web traffic. ICC operates in very small time scales (10 secs), and is applicable to both multi-homed and single-homed ISPs. The assessment of the mechanism using actual ISP traces has demonstrated the key properties, benefits and potential value of ICC for practical cases. We plan to employ more and longer traces and appropriate traffic models for additional evaluations of ICC and fine-tuning of its intelligence.. Taking also into account the cloud layer of the ICC mechanism, and combining both the network load and the energy cost of the IT infrastructure of the datacenters, will enable further optimizations on the management of inter-domain traffic for the specific case of cloud services. Finally, an actual implementation of the ICC mechanism was recently carried out in [10], using which will allow us to better capture the mechanism's benefits for the network even at packet-level and flow-level granularity.

ACKNOWLEDGMENT

This work has been performed partly in the framework of the EU ICT Project SmartenIT (FP7-2012-ICT-317846) [10]. The authors would like to thank all SmartenIT partners for useful discussions on the subject of the paper.

REFERENCES

[1] Lightwave, Datacenter Interconnectivity: http://www.lightwaveonline.com/data-center/data-center-interconnectivity.html

[2] Content Delivery Networks Interconnection (cdni): https://datatracker.ietf.org/wg/cdni/documents/

[3] OnApp Federation: http://onapp.com/federation/

[4] XIFI Infrastructure Federation: https://www.fi-xifi.eu/federation.html

[5] Zentera Cloud Federation: http://http://zentera.net/architecture/

[6] Enric Pujol and Philipp Richter and Balakrishnan Chandrasekaran and Georgios Smaragdakis and Anja Feldmann and Bruce Maggs and K. C. Ng. "Back-Office Web Traffic on The Internet", Proceedings of ACM IMC 2014, Vancouver, Canada, November 2014.

[7] Nikolaos Laoutaris, Nikolaos, Michael Sirivianos, Xiaoyuan Yang, and Pablo Rodriguez. "Inter-datacenter bulk transfers with netstitcher." ACM SIGCOMM. 2011.

[8] Application-Layer Traffic Optimization: https://datatracker.ietf.org/wg/alto/documents/

[9] David D. Clark, John Wroclawski, Karen R. Sollins, and Robert Braden. "Tussle in cyberspace: dening tomorrows internet", IEEE/ACM Trans. Netw., vol. 13, pp. 462-475, 2005.

[10] EU ICT Project SmartenIT (FP7-2012-ICT-317846) http://http://www.smartenit.eu/

[11] Zbigniew Dulinski, Rafal Stankiewicz, "Dynamic traffic management mechanism for active optimization of ISP costs", CNSM, Switzerland, October 2013. http://doi.ieeecomputersociety.org/10.1109/CNSM.2013.6727864