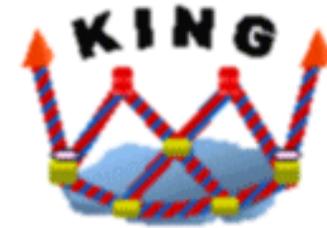




University of Würzburg
Department of Distributed Systems
Prof. Dr.-Ing. P. Tran-Gia



Jens Milbrandt, Michael Menth, Stefan Kopf

Adapative Bandwidth Allocation (ABA): Efficient Traffic Engineering with Admission-Controlled Capacity Tunnels

Outline

- ▷ Overview
 - Static bandwidth allocation (**SBA**)
 - Adaptive bandwidth allocation (**ABA**)
 - Complete capacity reassignment (**CCR**)
 - Selective capacity reassignment (**SCR**)
- ▷ Bandwidth savings potential of ABA
 - With opportunistic traffic model
 - With various dynamic traffic models
 - Comparison of capacity requirements
 - Bandwidth savings of ABA vs. SBA
 - Link specific analysis
- ▷ Conclusion



Problemstellung

- ▷ (Virtual) capacity tunnels used for traffic engineering
 - Border-to-border budgets for network admission control (KING)
 - General tunnels, e.g. label switched paths (LSPs) in (G)MPLS
- ▷ **Problem:** Adequate tunnel sizes required for **changing traffic aggregates**
- ▷ Static bandwidth allocation (**SBA**):
Tunnel capacity dimensioned statically for busy hours
- ▷ Adaptive bandwidth allocation (**ABA**):
Tunnel capacity adapted dynamically to current demand

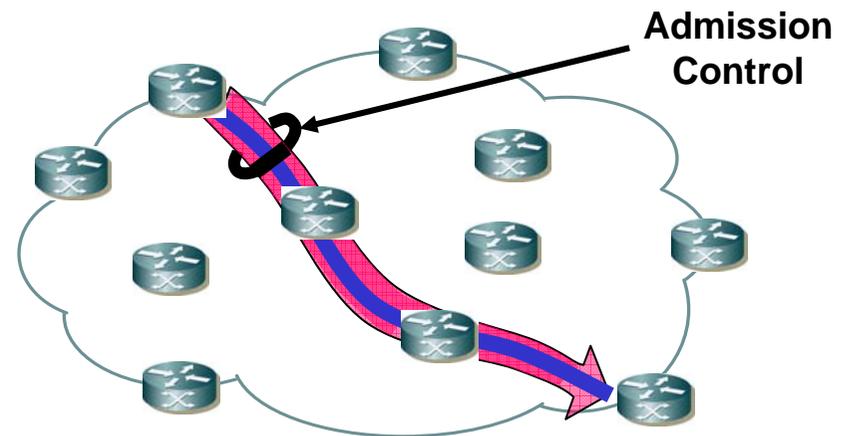


Fig: Border-2-border budget based network admission control

Mechanisms for Adaptive Bandwidth Allocation

Complete capacity reassignment (CCR)

- ▷ Measurements for all b2b traffic aggregates

Simple to implement but high processing/signalling overhead

control plane
(tolerance interval for tunnel blocking probability)

Selective capacity reassignment (SCR)

- ▷ Measurements for all b2b traffic aggregates

Less provisioning overhead but more complex implementation

control plane
(tolerance interval)

KING Test Network for Opportunistic Traffic Model

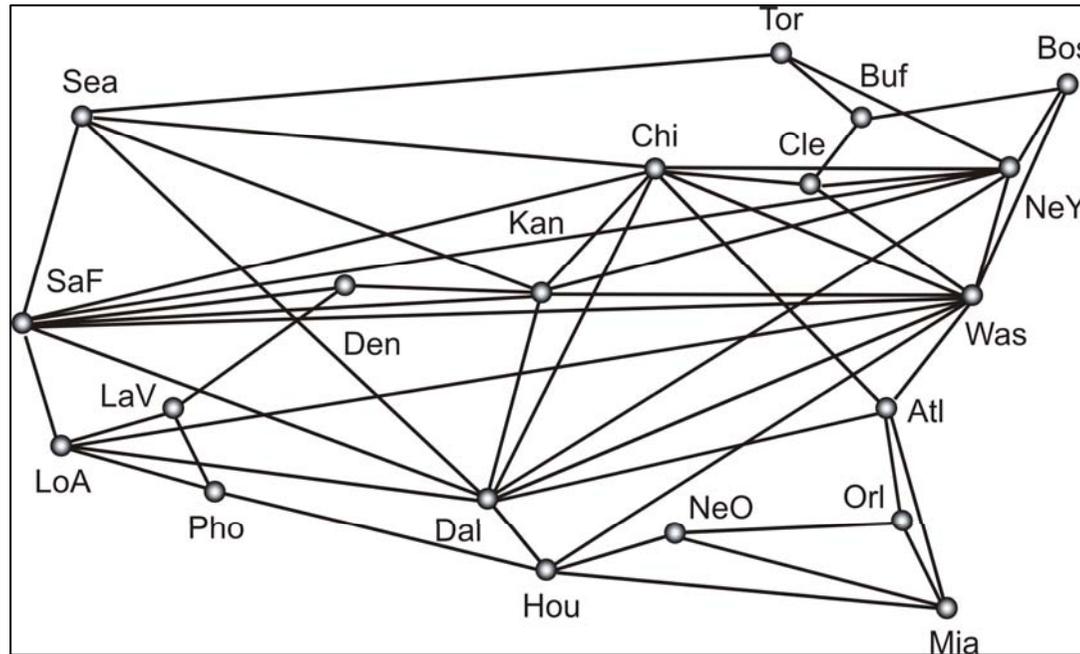


Fig: Network topology of the KING test network

Name(v)	$\pi(v)$ [10^3]	Name(v)	$\pi(v)$ [10^3]
Atlanta	4112	Los Angeles	9519
Boston	3407	Miami	2253
Buffalo	1170	New Orleans	1338
Chicago	8273	New York	9314
Cleveland	2250	Orlando	1645
Dallas	3519	Phoenix	3252
Denver	2109	San Francisco	1731
Houston	4177	Seattle	2414
Kansas	1776	Toronto	4680
Las Vegas	1536	Washington	4923

Tab: City population in the KING network



Opportunistic Traffic Model

- ▷ Traffic matrix construction proportional to city population
- ▷ Traffic matrix scaled by average b2b offered load a_{b2b}
- ▷ Overall offered load a_{tot} identical for SBA / ABA comparison
- ▷ Opportunistic Traffic:
 - Complementary oscillating b2b aggregate rates on each link
 - Constant cumulated aggregate rate on each link
 - Maximum bandwidth savings of 50% possible



Capacity Dimensioning for SBA / ABA

▷ *Capacity Dimensioning for SBA*

- Time-independent traffic matrix A_{max} containing peak aggregate rates
- Time-independent required link capacities c_l
- Calculation of overall required network capacity

$$C^{SBA} = \sum_l c_l$$

▷ *Capacity Dimensioning for ABA*

- Reoptimization of capacity tunnels every 5 minutes
- Multiple, time-dependent traffic matrices $A(t)$
- Multiple, time-dependent required link capacities $c_l(t)$
- Calculation of overall required network capacity

$$C^{ABA} = \sum_l \max_t(c_l(t))$$



Bandwidth Savings with Opportunistic Traffic Model

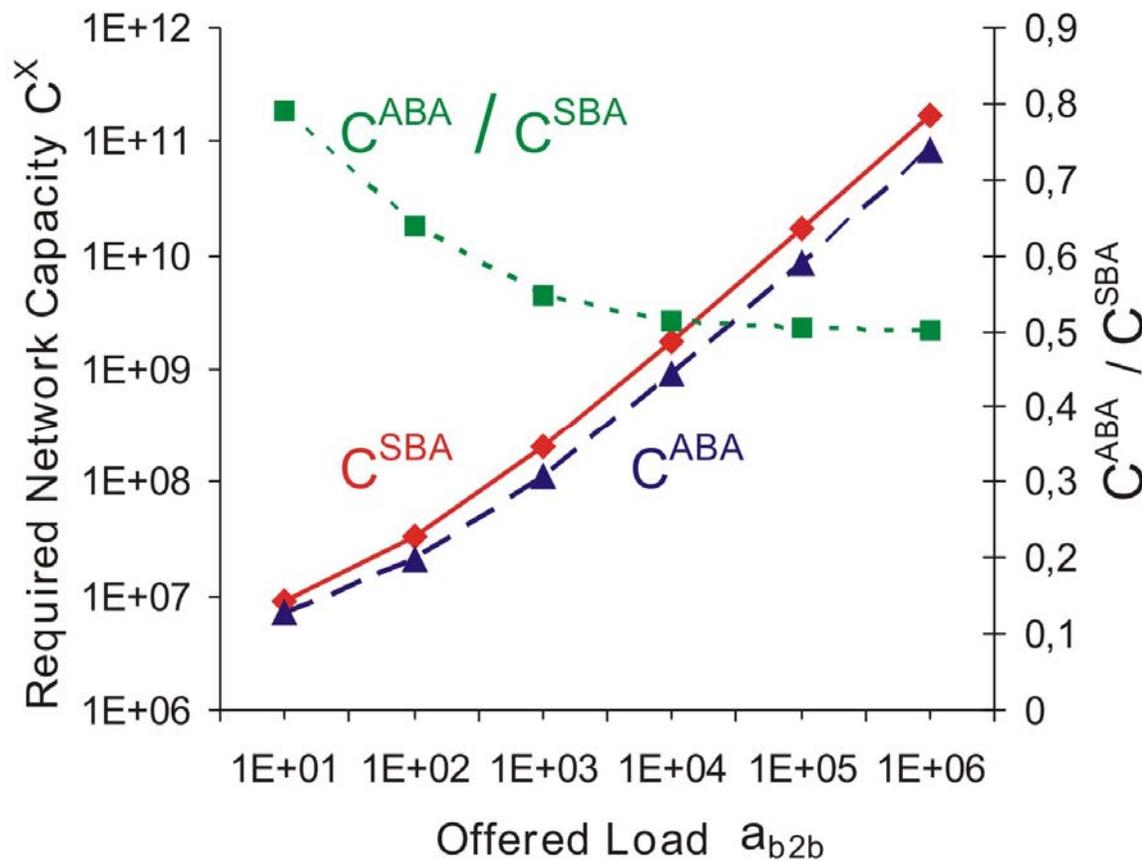


Fig: Bandwidth requirements for SBA / ABA

- ▷ Bandwidth savings depend on offered load
- ▷ Small capacities on average less utilized
- ▷ Large capacities on average better utilized
- ▷ Bandwidth savings increases with tunnel capacities
→ **Economy of scale**

Test Network for Dynamic Traffic Model

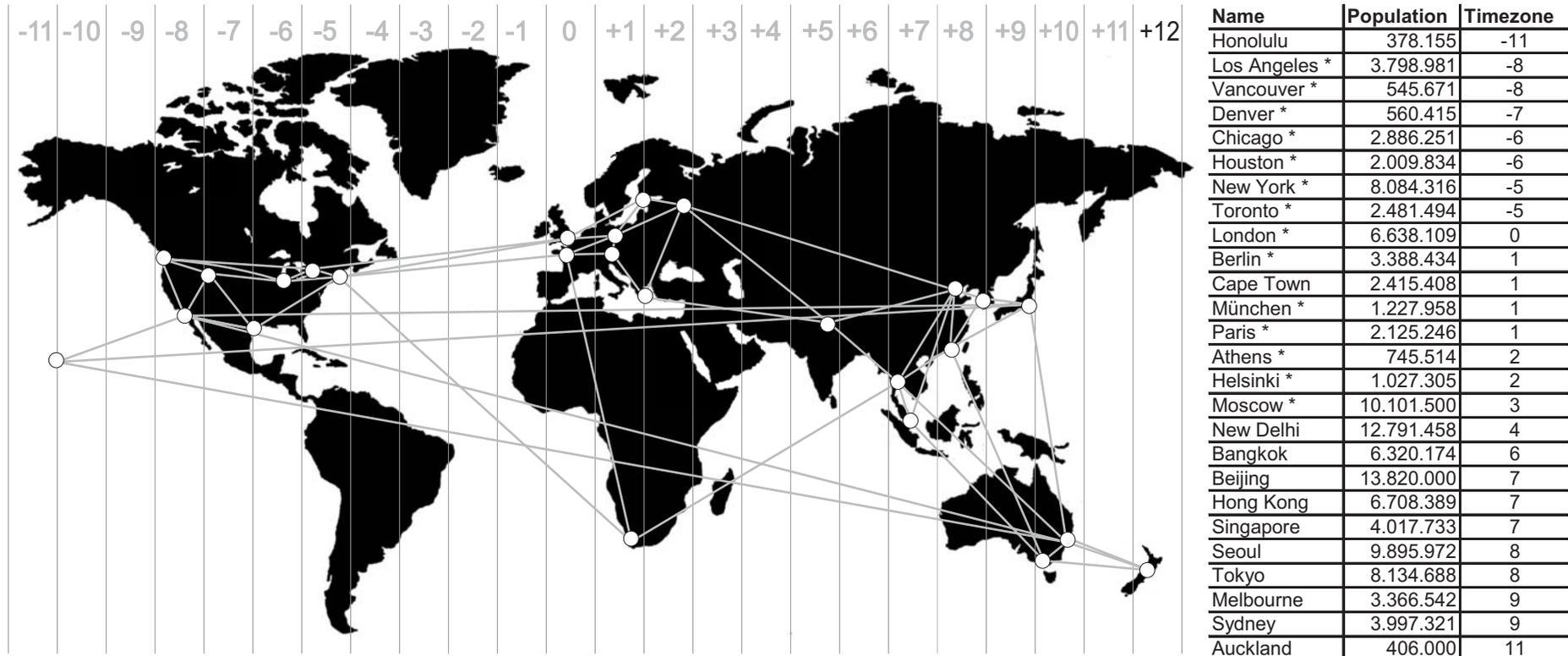


Fig: Network topology, population, and timezones of a world-wide test network



Node Activity Model

- ▷ B2b aggregate oscillations based on **ACTIVE** population
 - Not strictly opportunistic
 - Produced / consumed traffic increases with growing active population
 - Traffic oscillation according to **24h node activity model**

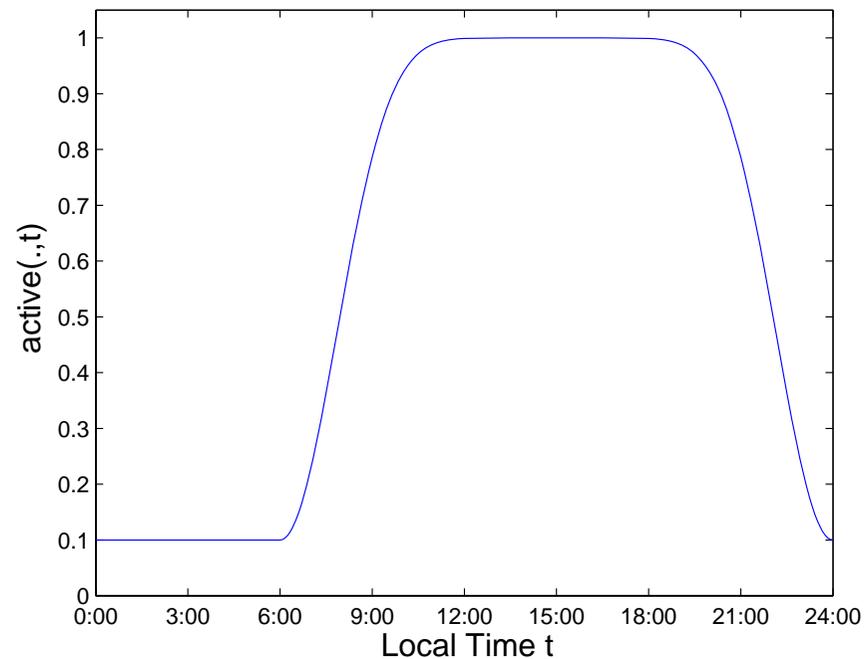


Fig: Node activity over 24 hours



Dynamic Traffic Models

- ▷ Different traffic models according to correlation with node activity
 - Linearity to Provider Activity (**LPA**)
 - Aggregate rates scale linearly to corresponding provider activity
 - Interpretation: client – server apps. (client push, e.g. backup)
 - Linearity to Consumer Activity (**LCA**)
 - Aggregate rates scale linearly to corresponding consumer activity
 - Interpretation: client – server apps. (client pull, e.g. web)
 - Linearity to Provide and Consumer Activity (**LPCA**)
 - Aggregate rates scale linearly to corresponding provider and consumer activities
 - Interpretation: peer-to-peer apps.
- ▷ Similarity of LPA and LCA model



Bandwidth Savings with Dynamic Traffic Models

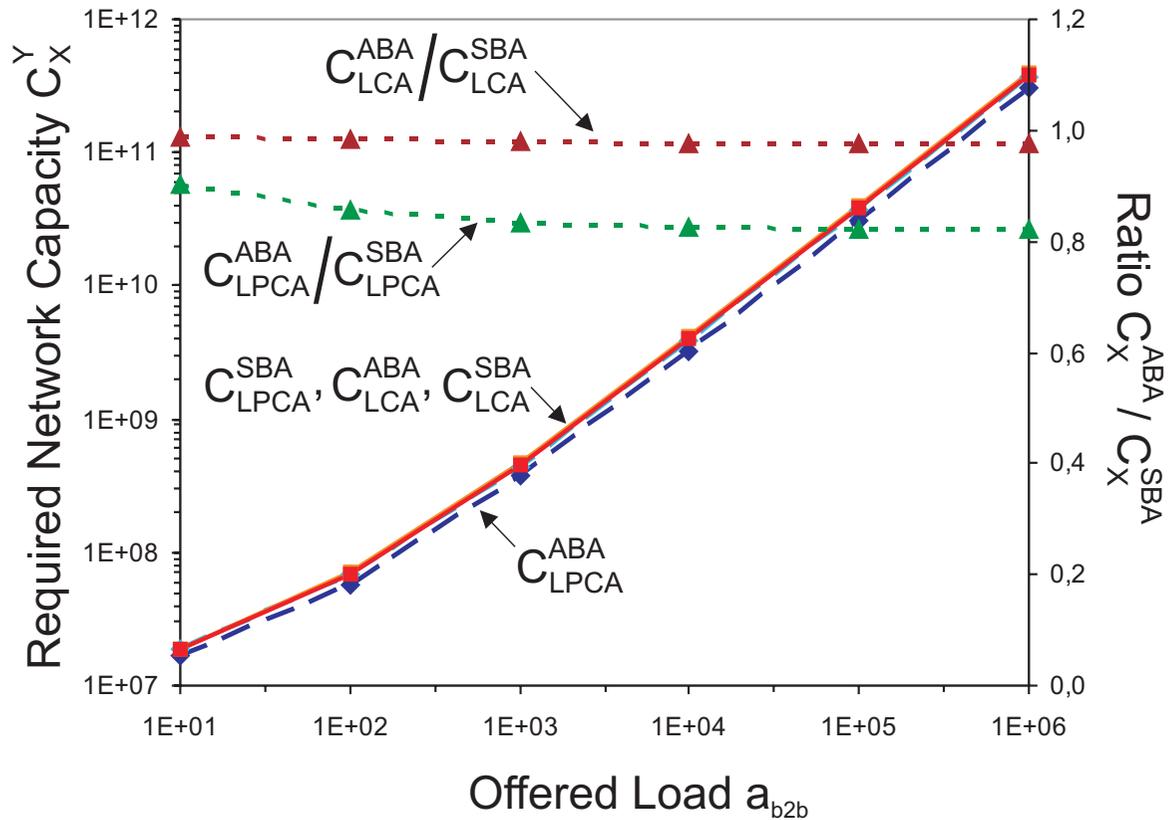


Fig: Bandwidth savings with dynamic traffic models

- ▷ Aggregation level with less impact on bandwidth savings
- ▷ Almost no (~2%) bandwidth savings for LCA model
- ▷ ~18% bandwidth savings for LPCA model

Link Analysis Seoul→Tokio (22 Aggregates)

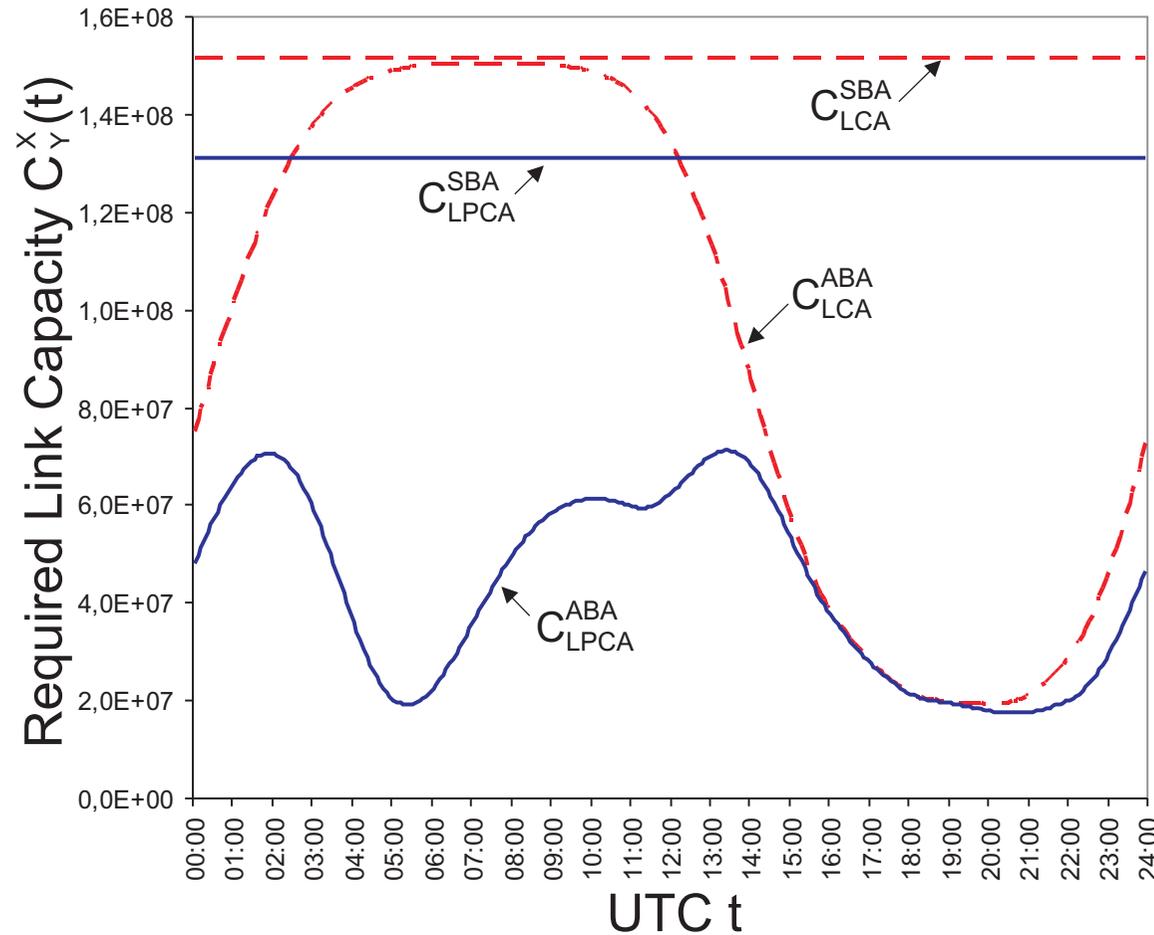
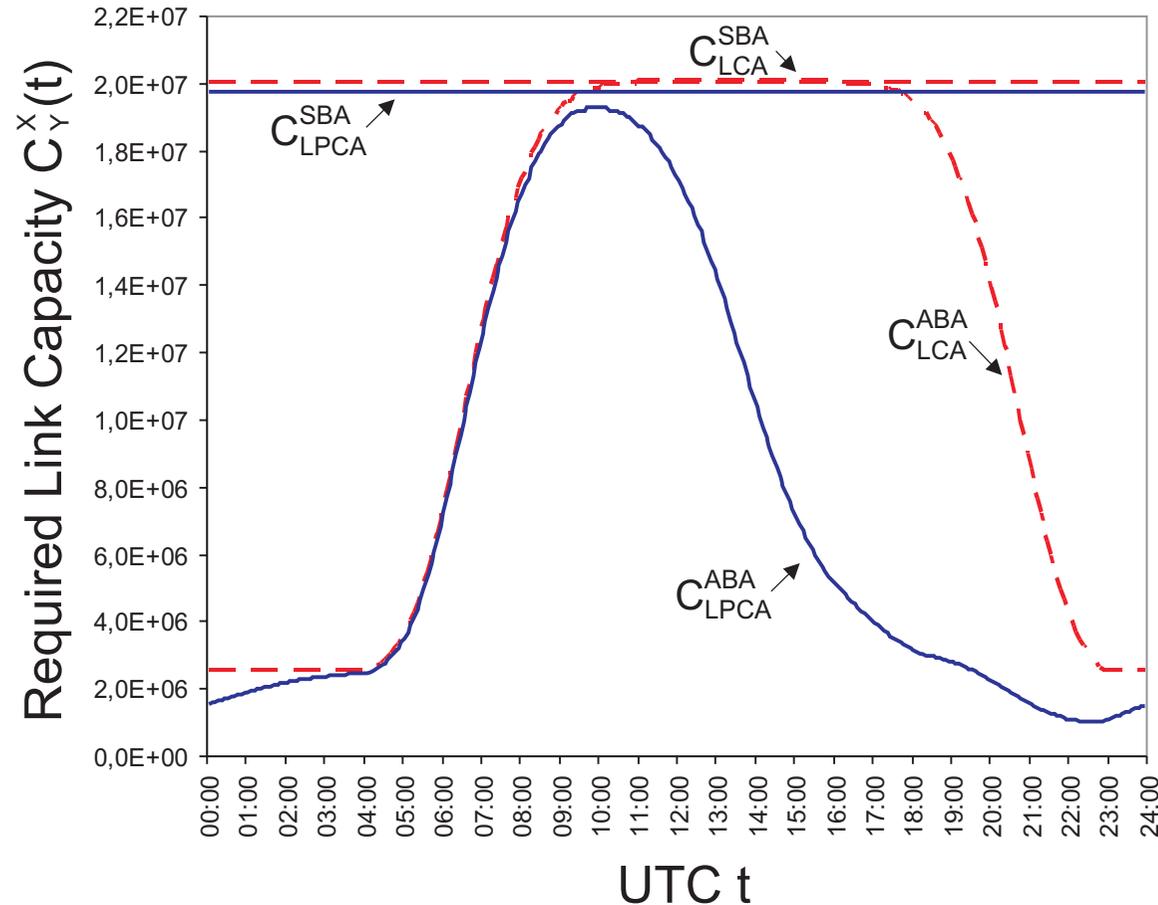


Fig: Time-dependent link capacity requirements

- ▷ Constant capacity requirements for SBA
- ▷ Strongly oscillating capacity requirement for ABA
- ▷ LCA: almost no bandwidth savings with ABA
- ▷ LPCA: ~50% bandwidth savings with ABA
- ▷ Busy hours of aggregates occur at different times

Link Analysis Bangkok→Beijing (16 Aggregates)



- ▷ Busy hour of LPCA shorter than of LCA
- ▷ Busy hours of aggregates occur at the same time
- ▷ Almost identical capacity requirements for ABA and SBA
- ▷ No bandwidth savings possible on this link

Fig: Time-dependent link capacity requirements

Conclusions

- ▷ Virtual tunnels used for traffic engineering
 - b2b budget network admission control (KING)
 - General tunnels, e.g. LSPs in (G)MPLS

- ▷ Bandwidth allocation strategies
 - Static and adaptive bandwidth allocation (SBA, ABA)
 - ABA mechanisms: CCR and SCR

- ▷ Investigation of the bandwidth savings potential
 - Savings of ABA vs. SBA depend on traffic model
 - Opportunistic traffic model: savings increase with offered load
 - Dynamic traffic model: 2 % for LCA and 18% for LPCA
 - Savings on links depend on traffic composition
 - Savings can be increased by suitable time-aware routing/load balancing

