PAPER

The Repacking Efficiency for Bandwidth Packing Problem

Jianxin CHEN[†*a)], **Yuhang YANG**[†], *Nonmembers*, *and* **Lei ZHOU**[†], *Student Member*

SUMMARY Repacking is an efficient scheme for bandwidth packing problem (BPP) in centralized networks (CNs), where a central unit allocates bandwidth to the rounding terminals. In this paper, we study its performance by proposing a new formulation of the BPP in the CN, and introducing repacking scheme into next fit algorithm in terms of the online constraint. For the realistic applications, the effect of call demand distribution is also exploited by means of simulation. The results show that the repacking efficiency is significant (e.g. the minimal improvement about 13% over uniform distribution), especially in the scenarios where the small call demands dominate the network.

key words: *bandwidth packing problem, repacking efficiency, centralized networks, average performance*

1. Introduction

Bandwidth packing problem (BPP) occurs in communication networks, where the total profit is maximized by assigning appropriate route to each call. In popular tri-model network architecture: backbone, distributed and access network, the performance bottleneck lies in the access network owning to the high speed switching fabric. The same problem also occurs in a centralized network (CN), where a central unit (CU) connects a set of terminals to the access network or backbone network.

The centralized architecture is popular in communication networks. For example router and access server in wireline networks, base station, access point in wireless networks and anchor node or cluster in ad hoc and wireless sensor networks are all belonging to CUs. Then in these CNs the BPP is that how to utilize the limited bandwidth efficiently in the CU.

There have a lot of results on the BPP through the entire network, but little has been done for the CN. Although Next Fit (NF) is a feasible online algorithm for BPP in CN, its performance is not satisfactory. Readily for synchronous bandwidth allocation in the CU, bandwidth is allocated periodically, and there the duration of each period is named packing window (PW). Herein some enhancement could be added for the existence of PW. Among them repacking scheme is a representative. Although it performs efficiently, still we do not how much the performance could be improved. That is our point of departure here.

In this work, we formulate the BPP in the CN that seeks to maximize the bandwidth utilization while keeping the online property. Several repacking algorithms based on NF are proposed for it. Repacking efficiency is evaluated by means of simulations. For realistic applications, call demand distributions are different. Thus that how the distribution of call demand affects the repacking efficiency has also been discussed. Our computation results show that in CN BPP, the repacking scheme favors the small call demands.

The remainder is organized as follows. Section 2 introduces the related works. Section 3 formulates the BPP in the CU, and Sect. 4 introduces several repacking algorithms with one call lookahead. In Sect. 5 average performance is discussed, and in the last section a brief conclusion is drawn.

2. Related Work

Previous BPP considered the maximal revenue over the entire network by assigning appropriate paths to each call, given the call demands, possible paths and related costs. Initial work on the BPP focused on the heuristic methods. Cox et al.[1] used a genetic algorithm and permutation based approach to dynamic call path assignment. Anderson et al.[2] strengthened the permutation approach with tabu search. Laguna and Glover [3] developed a non-permutation tabu search algorithm. Amiri and Barkhi [7] used Lagrangian Relaxation to obtain heuristic solutions to the multi-hour BPP. Parker and Ryan [4], [5] present an efficient column generation technique to the proposed linear programming relaxation problem.

These research focused on the route selection, without considering the quality of service (QoS) to users. QoS related metric such as delay has a major effect on the utilization of network resources. Rolland et al.[6] and Amiri et al.[7] proposed a new formulation for the BPP, maximizing revenues generated from the routed calls while minimizing the queueing delay.

For the stochastic characteristic of call arrival in communication networks, Coffman et al.[8], [9] computed call admission capacities for certain linear networks; in particular, they determined the maximum call arrival rate such that the number of delayed calls at any time remains finite in expected value.

These studies on BPP focused on the entire network, neglecting the performance bottleneck in the centralized networks just as we have mentioned. For the BPP in CN,

Manuscript received April 24, 2006.
Manuscript revised February 5, 2007.

[†]The authors are with the Department of Electronics Engineering, Shanghai Jiaotong University, Shanghai, P. R. China 200240.
[*]Presently, with the College of Computer, Nanjing University of Posts&Telecommunications.
a) E-mail: chjx2002@hotmail.com

DOI: 10.1093/ietisy/e90–d.7.1011

Next Fit (NF) is a feasible solution beacuse it works according to the call arrival sequence, i.e. first-come-first-serve. However, its performance is poor, e.g., in the worst-case 2 and in the average-case 4/3 [10]–[12]. Other online algorithms such as Best Fit, Harmonic algorithm perform much better, but the available bandwidth during each PW constrains them implementing in realistic applications. Fortunately, the randomness characteristic of arrival calls makes the repacking be a good technique for the performance improvement.

About the repacking scheme, to our best knowledge, little has been studied. In 1990 Gambosi et.al obtained the worst performance ratio 1.5 [13]. Later Ivković and Lloyd [14] developed a new repacking algorithm which improved the worst performance ratio to 1.25. In [15] Grove present a similar definition as repacking, i.e. lookahead online algorithm. He achieved the optimal performance 1.691 for bounded-space algorithms. Moreover, he obtained the solution about the optimal repacking condition, i.e. the total size of the lookahead item set is at least 4 when the bin capacity is unit. Then about the repacking scheme, there are no other solutions, especially about the average performance, which is more cared about in BPP.

3. Problem Formulation

In this section we formulate the BPP in the CU. Without loss of generality, we assume the available bandwidth during each PW is fixed in the CU, then the objective in our BPP is to maximize the bandwidth utilization during each PW as well as keep the online property. For convenience we refer the online property to the computation complexity of scheduling decision.

Before formulation, we introduce the following notations:

N the set of nodes around the CU
M the set of all calls to the CU
T_w the duration of PW
B the available bandwidth units during T_w
M_n the set of calls during PW n
d_n^m the demand of call $m \in M_n$
Q_n^f in the nth PW , the first call that current bandwidth can not hold
$d(Q_n^f)$ the demand of the Q_n^f
C_n in the nth PW, the set of allocated calls
$S(C_n)$ the number of calls in C_n
C_n^i the i^{th} call from the top to down sequence in C_n
$d(C_n^i)$ the demand of call C_n^i
R_n the available bandwidth during the nth PW
$T(A)$ the computation time for algorithm A to find the solution

As illustrated in Fig. 1, the BPP in the CU can now be defined as follows:

Given available bandwidth B and a set of call requests (a call table) M during each PW T_w, we seek to maximize the bandwidth utilization in terms of PW T_w constraint.

We assume that the available bandwidth, the duration of packing window and all call demands are known. We also make some assumptions which are typically used in modeling the queueing phenomena in packet networks. Specifically, we assume that the CU has infinite buffers to backlog user calls waiting for the transmission on the links, and call durations follow a uniform distribution.

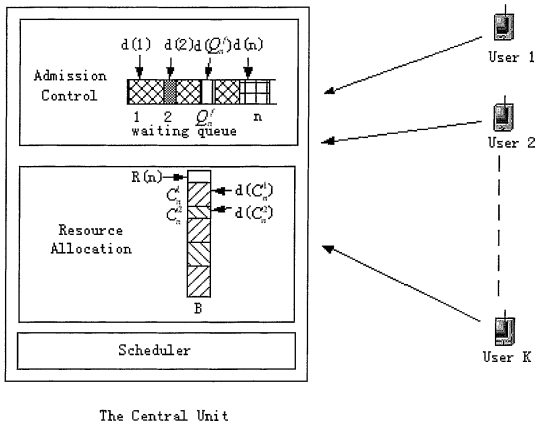

Fig. 1 The BPP in a central network.

Under these assumptions, the BPP could be modeled as a Markov chain [17], in which the available bandwidth was taken as the state of the Markov during each PW. The calls are items with sizes equal to the call demands.

We define the decision variable as

$$\phi_n^m = \begin{cases} 1, & if\ call\ m\ is\ packed\ during\ PW\ n, \\ 0, & otherwise. \end{cases}$$

Now, the wasted bandwidth under algorithm A during the nth PW becomes $B - \sum_{m \in M_n} d_n^m \phi_n^m$. In that, the last term in the objective function represents total used bandwidth units of allocated calls. Then the average bandwidth utilization under algorithm A could be written as

$$\overline{\Re}_A^{\infty} = \lim_{n \to \infty} \frac{1}{nB} \sum_n \sum_{m \in M_n} d_n^m \phi_n^m \tag{1}$$

The problem can now be properly formulated as follows:

Problem P:

$$max \left\{ \overline{\Re}_A^{\infty} = \lim_{n \to \infty} \frac{1}{nB} \sum_n \sum_{m \in M_n} d_n^m \phi_n^m \right\} \tag{2}$$

s.t.

$$T(A) \le T_W, \phi_n^m \in \{0, 1\}, d_n^m \in \{1, \cdots, B\}. \tag{3}$$

The format of the processing time for finding the solution during each PW is constrained. Thus the processing time T(A) implicitly measures the algorithm complexity (The larger T(A) means more computation complexity of algorithm).

Usually in order to evaluate the algorithm performance, the average performance ratio $\overline{R}_A^{\infty}$ is defined as follows

$$\overline{R}_A^{\infty} = \lim_{n\to\infty} E\left[\frac{A(M)}{OPT(M)}\right] \tag{4}$$

where $A(M)$ denotes the bandwidth used by the algorithm A for packing set M, and $OPT(M)$ denotes the minimum bandwidth required to pack call set M. If perfect packing could be performed, then $OPT(M)$ approaches to $d(M)$ ($d(M)$ is the sum of all calls in the set M), hence the following equation holds

$$\overline{\Re}_A^{\infty} = 1/\overline{R}_A^{\infty} \tag{5}$$

Either over discrete or continuous uniform distribution, the performance ratio of NF algorithm is 4/3. In our previous work [16] a more appropriate performance of NF algorithm was obtained for the packet networks.

Since the constraint of online computation complexity depends on the specific network scenarios, NF algorithm is a general solution for the sake of simplicity. Moreover, the stochastic characteristic of arrival calls makes the repacking a good enhancement mechanism, which deals with the allocated calls when the current available bandwidth can not hold the next call. Although the strength of such scheme is evident, still there has no answer about its efficiency. Here we study it based on NF algorithm, which happens when the current bandwidth can not hold the next call during the PW. It is similar to lookahead with one item problem [15]. Then our BPP with repacking (BPPR) is formulated as follows:

Problem BPPR:

$$min\left\{B - \sum_{m\in\{C_n, Q_n^f\}} d_n^m \phi_n^m\right\} \tag{6}$$

s.t.

$$T(SRS) \le T_W, \tag{7}$$

$$\phi_n^m \in \{0, 1\}, \tag{8}$$

$$d_n^m \in \{1, \cdots, B\}, \tag{9}$$

$$d(Q_n^f) \in \{1, \cdots, B\}, \tag{10}$$

$$C_n = \{C_n^i\}, i \in \{0, 1, \cdots, S(C_n)\}, \tag{11}$$

$$d(C_n^i) \in \{1, \cdots, B\}. \tag{12}$$

The set $\{C_n, Q_n^f\}$ of calls during the current PW is defined as the repacking set. Then the BPPR is to try to utilize bandwidth as much as possible for each packing during the PW, and repacking happens to the repacking set. In next section for this BPPR we present several algorithms.

4. Repacking Algorithms

In our BPPR, at the beginning of the nth PW , all calls are packed according to NF scheme till the current available bandwidth can not hold the next call, i.e. $R_n < d(Q_n^f)$. Now the problem is how to deal with the repack set. The following algorithms to BPPR are introduced according to the computation complexity of implementation.

Algorithm 1: Next Fit with the Top Call Repacking (NFTCR)

NFTCR is the most simplest repacking algorithm, which finds the better one between the top call C_n^1 and the call Q_n^f. For example, if $0 < R_n < d(Q_n^f)$, and there exists $(R_n + d(C_n^1)) \ge d(Q_n^f) > d(C_n^1)$, pack call Q_n^f instead of the C_n^1. Call C_n^1 would be packed first in the *(n+1)*th PW .

Note that NFTCR works simply for the reason that only the top call C_n^1 in C_n might be repacked, which do not need to retrieve all other calls in C_n. Compared with NF, during each window NFTCR adds only one step operation, i.e. $\Theta(1)$ computation complexity.

Algorithm 2: Next Fit with the First Call Repacking (NFFCR)

In NFTCR, only the top call C_n^1 might be repacked. If C_n^1 can not satisfy the repacking condition, no efficiency could be yielded. Since at the instant when the repacking happens, there have packed more than one calls, hence we can retrieve the calls from top to down, till find the first appropriate call as follows.

If $0 < R_n < d(Q_n^f)$ and $d(C_n^1) > d(Q_n^f)$, find the first call C_n^i s.t. $((R_n + d(C_n^i)) \ge d(Q_n^f) > d(C_n^i))$. Pack call Q_n^f instead of the C_n^i. Call C_n^i would be packed first in the next PW.

It is evident that if $S(C_n) = 1$, NFFCR equals NFTCR. Compared with NFTCR, no more than $S(C_n)$ steps operation is added to retrieve the first appropriate call in NFFCR, i.e. $O(S(C_n))$.

Algorithm 3: Next Fit with the Best Call Repacking (NFBCR)

In NFFIR algorithm, only the first appropriate call might be repacked from top to down sequence, but it may not be the best one, i.e. if there are more than one call could meet the repacking condition, only the first call would be repacked. If we modify the above rule, let the best call be repacked, then we could achieve the minimum wasted space in the condition where at most one call could be repacked. It is formulated as follows

If $0 < R_n < d(Q_n^f)$ and $d(C_n^1) > d(Q_n^f)$, find the best repacked call C_n^t, s.t. $t = argmin\{d(C_n^i) + R_n - Q_n^f | d(C_n^i) + R_n - Q_n^f \ge 0; i = 1, \cdots, S(C_n)\}$. Pack call Q_n^f instead of the C_n^t. Call C_n^t would be packed first in the next PW.

If the set of C_n is decreasing according to the call demand, NFBCR would be equal to NFFCR. Otherwise, it is evident that NFBCR performs much better than NFFCR.

Algorithm 4: Next Fit with the Optimal Repacking (NFOR)

In above algorithms, in order to find the minimal repacking efficiency, only one item is permitted to be repacked, which reduces the computation complexity. In such case we can find the optimal repacking efficiency by solving the BWPR problem. Note that BWPR is a knapsack problem, which could be solved by dynamic programming. The concrete procedure is

step (1): T=$S(C_n)$+1, $C_n^t = Q_n^f$;
for (w = 0 to B) $opt[0, w] = 0$;

for (k = 0 to T) $opt[k, 0] = 0$;
step (2): for (k = 1 to t)
for (w = 1 to B)
if ($w - d(C_n^k) < 0$)
$opt[k, w] = opt[k-1, w]$;
else
$opt[k, w] = max\{d(C_n^k) + opt[k-1, w - d(C_n^k)], opt[k-1, w]\}$;
step (3): return $opt[T, B]$.

In above procedure, the set of $opt[i, j]$ is used to store the optimal calls among the first i calls with the available bandwidth j units. Hence the optimal solution could be found in the set of $opt[T, B]$, and the other remaining calls in the repacking set have to be packed in the next PW. During each PW, to find the optimal solution, the computation complexity is $O((S(C_n) + 1)B)$, which increases with the available bandwidth. In some specific networks such as ad hoc or wireless sensor network, it is not suggested to use for the energy limit, which constrains the online computation complexity.

5. Computational Results

The repacking schemes presented in above section were coded in C++. A number of computational experiments were performed using an Intel Mobile Pentium 1.5 GHz, 512 M memory computer running under Windows XP operation system. To evaluate the effectiveness of those procedures, we assumed all calls are backlogged, the demand of which is over discrete uniform sequences $\{1, j\}$ consisting of 10^7 calls, where j is available bandwidth units during one PW and among $\{5, 10, 15, \cdots, 50\}$ (the larger value of j could be normalized). Although we have not defined the duration of PW, our simulation is enough to illustrate the repacking efficiency. On the other hand, for the duration of PW depends on the specific application environment, here it is no use to define the specific value.

5.1 Repacking Efficiency over Uniform Distribution

To show the repacking efficiency, we also list the average performance of NF algorithm in Fig. 2. It is evident that all repacking algorithms work much better than NF. Over discrete uniform distribution the minimal repacking efficiency is yielded about 13% by NFTCR, and the maximal efficiency is obtained about 17% by NFOR in the case where only one call is looked ahead. According to the above procedure, algorithm NFBCR should work better than NFFCR algorithm, and both of them should work much better than NFTCR. But in fact over uniform distribution the efficiency improvement of NFBCR from NFTCR is not significant, which is about 0.2%. The reason is mainly due to the fact that the available bandwidth units during each PW is not large enough (here we assume 50 bandwidth units). If the available bandwidth units during each PW increase greatly, the difference of efficiency improvement between NFBCR and NFTCR would be more evident. The similar reason exists for the efficiency improvements of NFBCR and NFFCR algorithms to NFTCIR when call demand is over uniform distribution. Compared to the optimal repacking algorithm, NFBCR and NFFCR algorithms work less 3%, but their implementation complexities are much less than that of NFOR algorithm, which could be seen in Table 1.

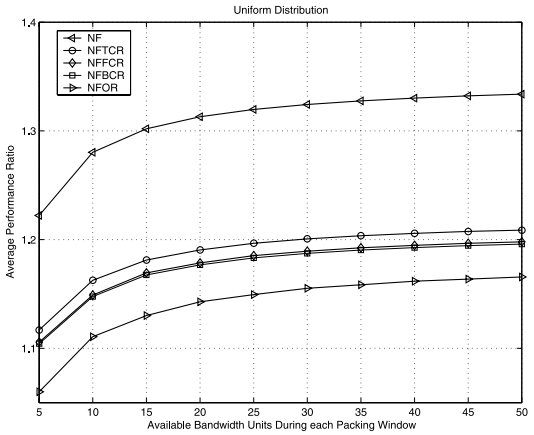

Fig. 2 Average performance over uniform distribution.

Table 1 Computation complexity for uniform distribution (in seconds).

B	NF	NFTCR	NFFCR	NFBCR	NFOR
5	6.960	7.461	7.771	7.731	27.279
10	7.200	8.192	8.222	8.101	36.682
15	7.822	8.582	9.424	8.682	42.311
20	7.921	8.405	8.926	8.843	46.326
25	8.012	8.903	8.973	8.862	49.902
30	8.202	8.933	9.012	8.883	53.327
35	8.134	8.482	8.993	8.893	56.581
40	8.445	9.124	9.043	8.873	59.806
45	8.212	8.883	9.063	9.073	62.921
50	9.213	8.842	9.263	8.863	66.005

For all above repacking algorithms, their computation complexities list according to the increasing sequence: NFTCR, NFFCR, NFBCR and NFOR. But in realistic implementation, it is not so. In Table 1, we can find the computation complexities of NF, NFTCR, NFFCR and NFBCR are nearly similar, and NFOR is rather more complex than others. During each PW, the computation complexities of NFTCR, NFFCR, and NFBCR are separatively $\Theta(1)$, $O(S(C_n))$, and $\Theta(S(C_n))$, while NFOR's is $\Theta(B \cdot S(C_n))$ ($y = \Theta(x)$ denotes y equal to x, and y=O(x) denotes y less than x). In Table 1 some results appear abnormal. For example, the computation complexity of NFFCR should be less than that of NFBCR, but in many cases they are not so. Such results occur owning to the specific algorithm implementation code. Note that in our simulation the code of NFFCR to search the first appropriate call is

overload=$d(Q_n^f) - R_n$;
$j = 1$;
while $(i > 0) \wedge (j > 0)$\{
if $(d(C_n^i) \geq overload) \wedge (d(C_n^i) < d(Q_n^f))j = 0$;
$i = i - 1$;\}.

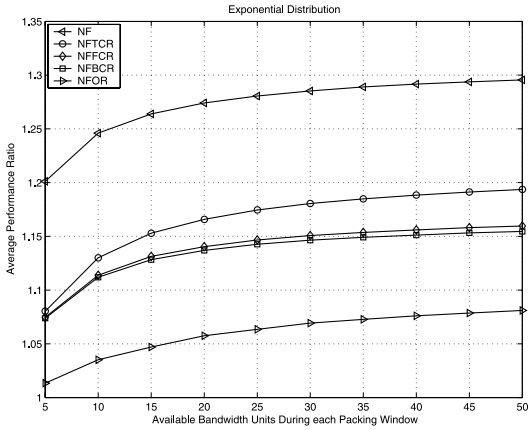

Fig. 3 Average performance over exponential distribution.

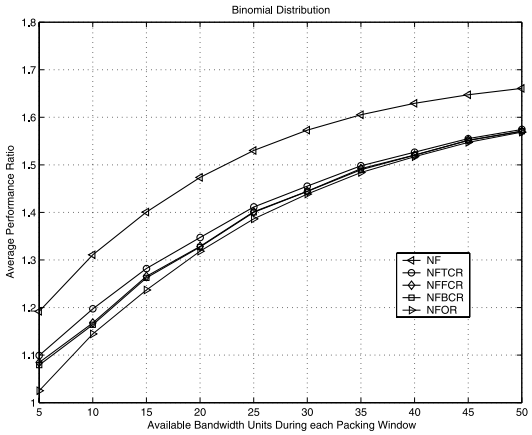

Fig. 4 Average performance over binary distribution.

The decision variable j is used to denote the finding of the first appropriate call when it equals 0 . The searching operation for the most appropriate call in NFBCR is programmed

$BC = d(Q_n^f)$;
while $(i > 0)$\{
if $(d(C_n^i) \geq overload) \wedge (d(C_n^i) < BC)$ $BC = d(C_n^i)$;
$i = i - 1$; \}.

From them, the reason for the abnormal phenomenon is evident because during each searching operation NFFCR algorithm performs a more decision operation than NFBCR. Hence when the number of available bandwidth units B is not very large, NFFCR even works worse than NFBCR in terms of computation complexity.

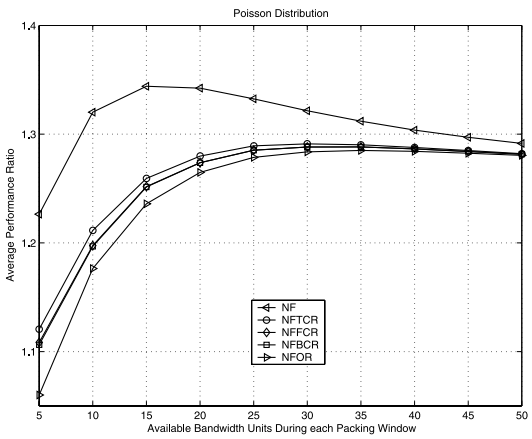

Fig. 5 Average performance over Poisson distribution.

5.2 Effect of the Call Demand Distribution

In above discussion, the call demand is assumed over uniform distribution. In realistic communication networks, it is difficult to define. Although in the Internet, the "typical" distribution of call demand is tri-modal, with the modes at or around 40 bytes, 560 bytes, and 1500 bytes [18], in specific communication networks such as ad hoc or wireless sensor network, no definite mode about call demand could be found. Hence in order to evaluate the repacking efficiency, we compare the algorithms over different distributions, such as exponential, poisson and binomial distribution. Their means of call demand are all equal to $(1+B)/2$, and performances under different algorithms are depicted in Fig. 3–Fig. 5.

It is evident that the call demand distribution affects repacking efficiency significantly. Among these distributions, all repacking algorithms work best when the call demand is over exponential distribution, and all repacking algorithms work worst over binomial distribution. Furthermore the repacking scheme is most efficient when the call demand is over exponential distribution but it works worst efficiently while over poisson distribution. The reason is that repacking scheme favors the small call demands much more than the large call demands. For example, over exponential distribution the optimal repacking efficiency is about 21%, while over uniform distribution, it is about 15%, and over poisson distribution, it is only little, less than 3% when the number of bandwidth units is 50 duration one PW. For the same principle, the difference of repacking efficiencies between NFFCR or NFBCR and NFTCR is about 4% over exponential distribution, which is much larger than that over uniform distribution.

While over binomial distribution, as most of call demands lay around the mean $(B+1)/2$, both numbers of the larger call demand and the smaller call demand are relatively small, then the repacking efficiency is not so distinct. Under NFOR algorithm, the performance improvement is about 10%, which is a little better than that of NFTCR, NFFCR, or NFBCR algorithm. Moreover the efficiency difference between NFTCR and NFFCR (or NFBCR) is also not obvious.

Over poisson distribution, an interesting phenomena appears, i.e. the repacking efficiency improves irregularly. For all other distributions, the repacking efficiency under each repacking algorithm improves and converges as the number of available bandwidth units increases. While over poisson distribution, the trend is anomalous. But for dif-

ferent bandwidth units during each PW, the average performance under the optimal repacking algorithm increases slowly, also little improvement could be obtained compared to NFTCR, NFFCR, or NFBCR algorithm. This result occurs because the number of large call demands decreases exponentially according to the poisson distribution definition. Hence repacking implementation seldom happens. Therefore a little repacking efficiency could be yielded over poisson distribution.

According to the above discussion, we may deduce that the call demand distribution plays an important role on the repacking efficiency. The repacking efficiency improves as the number of smaller of call demands increases, on the contrary, the repacking scheme works poorly. Hence the repacking scheme favors the communication networks where the small call demands are in the majority.

6. Conclusions

In this paper, we studied the repacking efficiency for the bandwidth packing problem in the centralized networks by introducing a BPPR. Then four NF-based repacking algorithms have been proposed to solve it. Their performances are exploited when the call demand is over uniform distribution. In addition, we discuss the effect of call distribution on the repacking efficiency for the realistic applications. Our study showed that the repacking scheme performs well and favors the scenarios where the small call demands are in the majority. Implicitly when call demands are over uniform distribution the average performance improvement reaches about 15%; while over exponential distribution it arrives at 21%.

References

[1] L. Cox, L. Davis, and Y. Qui, "Dynamic anticipatory routing in cuircuit-switched telecommunications networks," in Handbook of Genetic Algorithms, ed. L. Davis, pp.113–127, Van Norstrand/Reinhold, New York, 1991.
[2] C.A. Anderson, K. Fraughnaugh, M. Parker, and J. Ryan, "Path assignment for call routing: An application of tabu search," Annals of Operations Research, vol.41, pp.301–312, 1993.
[3] M. Laguna and F. Glover, "Bandwidth packing: A tabu search approach," Management Science, vol.39, pp.492–500, 1993.
[4] K. Park, S. Kang, and S. Park, "An integer programming approach to the bandwidth packing problem," Manage. Sci., vol.42, pp.1277–1291, 1996.
[5] C. Villa and K. Hoffman, "A column-generation and branch-and-cut approach to the bandwidth-packing problem," J. Res. Natl. Inst. Stand. Technol., vol.111, pp.161–185, 2006.
[6] E. Rolland, A. Amiri, and R. Barkhi, "Queueing delay guarantees in bandwidth packing," Computers and Operations Research, vol.26, pp.921–935, 1999.
[7] A. Amiri, "The multi-hour bandwidth packing problem with response time guarantees," Information Technology and Management, vol.4, pp.113–127, 2003.
[8] E.G. Coffman, Jr., A. Feldmann, N. Kahale, and B. Poonen, "Computing call admission capacities in linear networks," Prob. Eng. Inf. Sci., vol.13, no.4, pp.387–406, 1999.
[9] E.G. Coffman, Jr., and A.L. Stolyar, "Bandwidth packing," Algorithmica, vol.29, pp.70–88, 2001.
[10] E.G. Coffman, Jr., S. Halfin, A. Jean-Marie, and P. Robert, "Stochastic analysis of a slotted FIFO communication channel," IEEE Trans. Inf. Theory, vol.39, no.5, pp.1555–1566, 1993.
[11] E.G. Coffman, Jr., M.R. Garey, and D.S. Johnson, "Approximation algorithms for bin packing: A survey," in Approximation Algorithms for NP-Hard Problems, ed. D. Hochbaum, pp.46–93, PSW Publishing, Boston, 1996.
[12] M. Nir and R. Raphael, "Packet scheduling with fragmentation," IEEE Infocom, pp.427–436, 2002.
[13] G. Gambosi, A. Postiglione, and M. Talamo, "New algorithms for on-line bin packing," in Proc. First Italian Conference, Algorithms and Complexity, ed. R. Petreschi, G. Aussiello, D.P. Bovet, pp.44–59, World Scientific, Singapore, 1990.
[14] Z. Ivkovic and E.L. Lloyd. "Fully dynamic algorithms for bin packing: Being (Mostly) myopic helps," SIAM Journal on Computing, vol.28, no.2, pp.574–611, 1998.
[15] E.F. Grove. "Online bin packing with lookahead," ACM SODA, pp.430–436, 1995.
[16] J. Chen, Y. Yang, H. Zhu, and P. Zeng, "A bounded item bin packing problem over discrete distribution," TAMC 2006, pp.108–117, 2006.
[17] J. Chen and L. Gong, "Two bandwidth packing algorithms in a centralized wireless network and their average-case analysis," ICICS 2005, pp.628–632, Bangkok, Thailand, 2005.
[18] http://ipmon.sprint.com/packstat/viewresult.php? NULL: pktsz: sj-25.0-020419

Jianxin Chen received the BS degree in Automation Control engineering from Liaoning Petroleum & Chemical University, Liaoning, P.R. China, in 1996, received the MS degree in Computer Engineering in 2002. He is currently working towards the Ph.D. degree in the Institute of Modern Communication, Department of Electronics Engineering, Shanghai Jiao Tong University (SJTU), P.R. China. His research interests include protocol implementation and algorithm design.

Yuhang Yang graduated from the Electronic Engineering Department of Chengdu Institute of Meteorology in 1982. From 1984 to 1987, Yang studied telecommunications and computer networking and received an MSEE from Aston University, Great Britain. Now he is professor of the Department of Electronic Engineering, Shanghai Jiao Tong University. Mr. Yang was awarded respectively as one of the top ten honorees in the National Electronics Achievement Award in 1994, and top honors in the Technology Improvement Award by the Electronics Ministry of Chinese government in 1995. Yang was honored as the most outstanding person "Cross the Century" by the National Education Committee in 1997. In recent years, his research interest lies mainly in the field of Broadband Wireless, Grid Networking, Information Security and Online Video Distribution. As the leading person of the projects, he is in charge of the global planning of all the projects and makes sure that the research and development for applications stands in the front of the subject.

Lei Zhou received the BS degree in wireless communication engineering from Nanjing University of Posts and Telecommunications, Nanjing, P.R. China, in 1999, received the MS degree in signal and information processing in 2002. He is currently working towards the Ph.D. degree in the Institute of Image Communication and Information Processing, Department of Electronics Engineering, Shanghai Jiao Tong University (SJTU), P.R. China. His research interests include image process and digital TV.