

## HTTP-NG

با نزدیک شدن به اتمام این کتاب، HTTP دهمین تولد خود را جشن می‌گیرد و این یک دهه کامل برای این پروتکل اینترنتی بوده است. امروزه HTTP اکثریت مطلق ترافیک دیجیتال را در سراسر جهان جابجا می‌کند.

اما همانطور که HTTP در سال‌های نوجوانی خود رشد می‌کند، با چند چالش روبرو می‌شود. از برخی جهات، سرعت پذیرش HTTP از طراحی آن جلوتر رفته است. امروزه، مردم از HTTP به عنوان پایه‌ای برای بسیاری از برنامه‌های کاربردی مختلف، بر روی بسیاری از فناوری‌های مختلف شبکه استفاده می‌کنند.

این فصل به تشریح برخی از روندها و چالش‌های آینده HTTP و پیشنهادی برای معماری نسل بعدی به نام HTTP-NG می‌پردازد. در حالی که گروه کاری برای HTTP-NG منحل شده است و پذیرش سریع آن در حال حاضر بعید به نظر می‌رسد، با این وجود برخی از جهت‌گیری‌های بالقوه آینده HTTP را مشخص می‌کند.

### HTTP's Growing Pains

HTTP در ابتدا به عنوان یک تکنیک ساده برای دسترسی به محتوای چندرسانه‌ای مرتبط از سرورهای اطلاعات توزیع شده تصور شد. اما، در دهه گذشته، HTTP و مشتقات آن نقش بسیار گسترده‌تری را ایفا کرده‌اند.

HTTP/1.1 اکنون Tagging و Fingerprinting را برای ردیابی نسخه‌های سند، روش‌هایی برای پشتیبانی از بارگذاری اسناد و تعامل با Gateway های برنامه‌ای، پشتیبانی از محتوای چند زبانه، امنیت و احراز هویت، حافظه پنهان (Cache) برای کاهش ترافیک، Pipelining برای کاهش تأخیر، اتصالات مداوم ( Persistent Connections) برای کاهش زمان راه اندازی و بهبود پهنای باند و دسترسی به محدوده برای پیاده سازی بروز رسانی‌های جزئی. برنامه‌های افزودنی و مشتقات HTTP حتی فراتر رفته‌اند و از انتشار اسناد، ارائه برنامه‌ها، پیام‌های دلخواه، پخش ویدئو و پایه‌های دسترسی چندرسانه‌ای بی‌سیم پشتیبانی می‌کنند. HTTP در حال تبدیل شدن به نوعی "سیستم عامل" برای برنامه‌های کاربردی رسانه‌های توزیع شده است.

طراحی HTTP/1.1، در حالی که به خوبی در نظر گرفته شده است، شروع به نشان دادن برخی فشارها کرده است زیرا HTTP بیشتر و بیشتر به عنوان یک بستر یکپارچه برای عملیات‌های پیچیده از راه دور استفاده می‌شود. حداقل چهار ناحیه وجود دارد که HTTP برخی از دردهای رشد را نشان می‌دهد:

### Complexity

HTTP بسیار پیچیده است و ویژگی‌های آن به یکدیگر وابسته هستند. اجرای صحیح نرم‌افزار HTTP به دلیل نیازهای پیچیده و در هم تنیده و اختلاط مدیریت اتصال، مدیریت پیام و منطق عملکردی، قطعاً دردناک و مستعد خطا است.



## Extensibility

گسترش تدریجی HTTP دشوار است. بسیاری از برنامه‌های HTTP قدیمی وجود دارند که ناسازگاری‌هایی را برای برنامه‌های افزودنی پروتکل ایجاد می‌کنند، زیرا فاقد فناوری برای برنامه‌های افزودنی عملکرد مستقل هستند.

## Performance

HTTP ناکارآمدی عملکرد دارد. بسیاری از این ناکارآمدی‌ها با پذیرش گسترده فناوری‌های دسترسی بی‌سیم با تأخیر بالا و بازده کم جدی‌تر خواهند شد.

## Transport dependence

HTTP حول یک پشته شبکه TCP/IP طراحی شده است. در حالی که هیچ محدودیتی برای زیر پشته‌های جایگزین وجود ندارد، کار کمی در این زمینه انجام شده است. HTTP برای اینکه به عنوان یک پلتفرم پیام‌رسانی گسترده‌تر در برنامه‌های کاربردی تعبیه شده و بی‌سیم مفید باشد، نیاز به پشتیبانی بهتر از زیرپشته‌های جایگزین دارد.

## HTTP-NG Activity

در تابستان ۱۹۹۷، کنسرسیوم جهانی وب پروژه ویژه‌ای را برای بررسی و پیشنهاد نسخه جدید اصلی HTTP راه‌اندازی کرد که مشکلات مربوط به پیچیدگی، توسعه پذیری، عملکرد و وابستگی حمل و نقل را برطرف می‌کرد. این HTTP جدید HTTP: Next Generation (HTTP-NG) نام داشت.

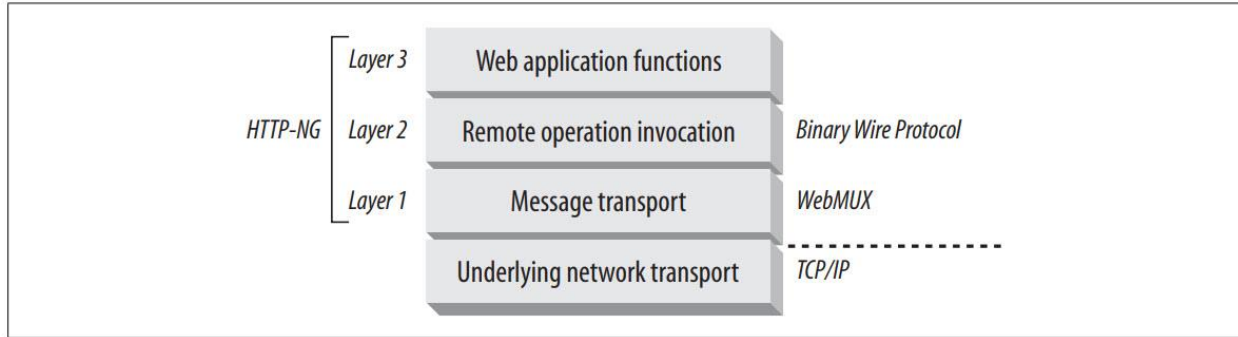
مجموعه‌ای از پیشنهادات HTTP-NG در جلسه IETF در دسامبر ۱۹۹۸ ارائه شد. این فناوری به طور گسترده پیاده‌سازی نشده است (و ممکن است هرگز اجرا نشود)، اما HTTP-NG نشان‌دهنده جدی‌ترین تلاش برای گسترش نسل HTTP است. بیایید HTTPNG را با جزئیات بیشتری بررسی کنیم.

## Modularize and Enhance

موضوع HTTP-NG را می‌توان در این کلمات توصیف کرد: مدولار کردن (Modularize) و تقویت کردن (Enhance).

گروه کاری HTTP-NG به جای اینکه مدیریت اتصال، مدیریت پیام، منطق پردازش سرور و روش‌های پروتکل را در هم آمیخته باشد، مدولار کردن پروتکل را در سه لایه پیشنهاد کرد که در شکل زیر نشان داده شده است:





لایه ۱، لایه انتقال پیام، بر ارائه پیام‌های غیر شفاف بین نقاط پایانی، مستقل از عملکرد پیام‌ها تمرکز دارد. لایه انتقال پیام از زیرپشته‌های مختلف (مثلاً پشته‌ها برای محیط‌های بی‌سیم) پشتیبانی می‌کند و بر مشکلات تحویل کارآمد پیام و مدیریت آن تمرکز می‌کند. تیم پروژه HTTP-NG پروتکلی به نام WebMUX برای این لایه پیشنهاد کرد.

لایه ۲، لایه فراخوانی راه دور، عملکرد Request/Response را تعریف می‌کند که در آن کلاینت‌ها می‌توانند عملیات منابع سرور را فراخوانی کنند. این لایه مستقل از انتقال پیام و معنای دقیق عملیات است. این فقط یک راه استاندارد برای فراخوانی هر عملیات سرور ارائه می‌دهد. این لایه تلاش می‌کند تا چارچوبی توسعه‌پذیر و شی‌گرا را بیشتر شبیه به CORBA، DCOM و Java RMI ارائه کند. تیم پروژه HTTP-NG، Binary Wire Protocol را برای این لایه پیشنهاد کرد.

لایه ۳، لایه برنامه وب، بیشتر منطق مدیریت محتوا را فراهم می‌کند. تمام متدهای HTTP/1.1 (GET، POST، PUT، و غیره) و همچنین پارامترهای هدر HTTP/1.1، در اینجا تعریف می‌شوند. این لایه همچنین از سایر خدمات ساخته شده بر روی فراخوانی راه دور مانند WebDAV پشتیبانی می‌کند.

هنگامی که اجزای HTTP مدولار می‌شوند، می‌توان آن‌ها را برای ارائه عملکرد بهتر و عملکرد غنی‌تر ارتقا داد.

### Distributed Objects

بسیاری از اهداف فلسفه و عملکرد HTTP-NG به شدت از سیستم‌های ساختار یافته، شی‌گرا و اشیاء توزیع شده مانند CORBA و DCOM وام گرفته شده‌اند. سیستم‌های اشیاء توزیع شده می‌توانند به توسعه پذیری و عملکرد ویژگی‌ها کمک کنند.

جامعه‌ای از محققان از سال ۱۹۹۶ بر سر همگرایی بین HTTP و سیستم‌های پیچیده‌تر اشیاء توزیع شده بحث کرده‌اند. برای کسب اطلاعات بیشتر در مورد مزایای الگوی اشیاء توزیع شده برای وب، مقاله اولیه



Xerox PARC با عنوان "مهاجرت وب به سوی اشیاء توزیع شده" را مطالعه نمایید  
(<ftp://ftp.parc.xerox.com/pub/ilu/misc/webilu.html>).

فلسفه جاه طلبانه یکسان سازی وب و اشیاء توزیع شده مقاومت در برابر پذیرش HTTP-NG در برخی جوامع ایجاد کرد. برخی از سیستم‌های اشیاء توزیع شده گذشته از اجرای سنگین وزن و پیچیدگی رسمی رنج می‌بردند. تیم پروژه HTTP-NG تلاش کرد تا برخی از این نگرانی‌ها را در الزامات برطرف کند.

### Layer 1: Messaging

بیابید نگاهی دقیق‌تر به سه لایه HTTP-NG بیندازیم که از پایین‌ترین لایه شروع می‌شود. لایه انتقال پیام به تحویل کارآمد پیام‌ها، مستقل از معنا و هدف پیام‌ها مربوط می‌شود. لایه انتقال پیام بدون توجه به پشته شبکه واقعی، یک API برای پیام‌رسانی را فراهم می‌کند.

این لایه بر بهبود عملکرد پیام‌رسانی تمرکز دارد، از جمله:

- **Pipelining** و دسته بندی (Batching) پیام‌ها برای کاهش تأخیر رفت و برگشت
- استفاده مجدد از اتصالات برای کاهش تأخیر و بهبود پهنای باند تحویلی
- چندین جریان پیام به صورت موازی، روی یک اتصال، برای بهینه سازی اتصالات مشترک و در عین حال جلوگیری از Starvation جریان پیام
- تقسیم بندی پیام کارآمد برای آسان‌تر کردن تعیین مرزهای پیام

تیم HTTP-NG بیشتر انرژی خود را برای توسعه پروتکل WebMUX برای انتقال پیام لایه یک سرمایه گذاری کرد. WebMUX یک پروتکل پیام با کارایی بالا است که پیام‌ها را در یک اتصال TCP مالتی پلکس تقسیم می‌کند. ما WebMUX را در ادامه این فصل با کمی جزئیات بیشتر مورد بحث قرار خواهیم داد.

### Layer 2: Remote Invocation

لایه میانی معماری HTTP-NG از فراخوانی روش راه دور پشتیبانی می‌کند. این لایه یک چارچوب Request/Response عمومی را فراهم می‌کند که در آن کلاینت‌ها عملیات منابع سرور را فراخوانی می‌کنند. این لایه به پیاده سازی و مفاهیم عملیات خاص (کش، امنیت، منطق روش و غیره) مربوط نمی‌شود. این فقط به اینترفیس مربوط می‌شود تا به کلاینت‌ها اجازه دهد تا از راه دور عملیات سرور را فراخوانی کنند.

بسیاری از استانداردهای فراخوانی روش از راه دور در حال حاضر در دسترس هستند (CORBA، DCOM، و جاوا RMI) و این لایه برای پشتیبانی از همه ویژگی‌های فوق العاده این سیستم‌ها در نظر گرفته نشده است. با این حال، یک هدف صریح برای گسترش غنای پشتیبانی HTTP RMI از آنچه





توسط HTTP/1.1 ارائه شده وجود دارد. به طور خاص، هدفی برای ارائه پشتیبانی فراخوانی از راه دور عمومی تر، به شیوه‌ای قابل توسعه و شی گرا وجود دارد.

تیم HTTP-NG، Binary Wire Protocol را برای این لایه پیشنهاد کرد. این پروتکل از یک فناوری با کارایی بالا و توسعه‌پذیر برای فراخوانی عملیات توصیف شده روی سرور و انتقال نتایج پشتیبانی می‌کند. در ادامه این فصل، در مورد Binary Wire Protocol با جزئیات بیشتری صحبت می‌کنیم.

### Layer 3: Web Application

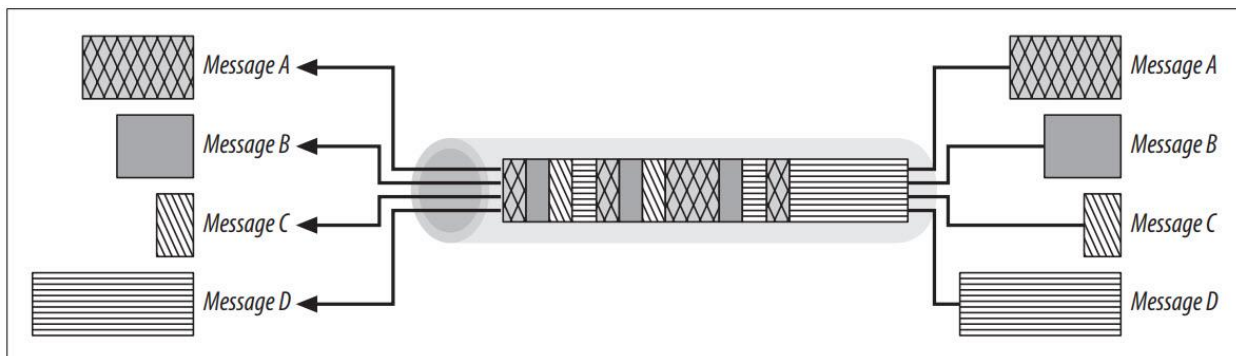
لایه Web Application جایی است که معناشناسی و منطق برنامه کاربردی انجام می‌شود. گروه کاری HTTP-NG از وسوسه گسترش ویژگی‌های برنامه HTTP خودداری کرد و در عوض بر زیرساخت‌های رسمی تمرکز نمود. این لایه سیستمی را برای ارائه خدمات ویژه برنامه توصیف می‌کند. این خدمات یکپارچه نیستند. API های مختلف ممکن است برای برنامه‌های مختلف موجود باشد. به عنوان مثال، برنامه وب برای HTTP/1.1 یک برنامه کاربردی متفاوت از WebDAV را تشکیل می‌دهد، اگرچه ممکن است برخی از بخش‌های مشترک را به اشتراک بگذارند. معماری HTTP-NG به چندین برنامه اجازه می‌دهد تا در این سطح همزیستی داشته باشند، امکانات زیربنایی را به اشتراک بگذارند و مکانیزمی برای افزودن برنامه‌های کاربردی جدید فراهم می‌کند.

فلسفه لایه Web Application این است که عملکردی معادل برای HTTP/1.1 و رابط‌های افزونه (Extension Interfaces) ارائه دهد، در حالی که آن‌ها را در چارچوبی از اشیاء توزیع شده قابل توسعه باز می‌سازد. می‌توانید در <http://www.w3.org/Protocols/HTTP-NG/1998/08/draft-larner-nginterfaces-00.txt> درباره رابط‌های لایه Web Application اطلاعات بیشتری کسب کنید.

### WebMUX

گروه کاری HTTP-NG بیشتر انرژی خود را در توسعه استاندارد WebMUX برای انتقال پیام سرمایه گذاری کرده است. WebMUX یک سیستم پیام پیچیده و با کارایی بالا است که در آن پیام‌ها می‌توانند به صورت موازی از طریق یک اتصال TCP مالتی پلکس منتقل شوند. جریان‌های پیام فردی که با نرخ‌های مختلف تولید و مصرف می‌شوند، می‌توانند به طور مؤثری روی یک یا تعداد کمی از اتصالات TCP بسته‌بندی و چندگانه شوند.





در اینجا برخی از اهداف مهم پروتکل WebMUX آورده شده است:

- طراحی ساده
- کارایی بالا
- Multiplexing – چندین جریان داده (از پروتکل‌های سطح بالاتر دلخواه) می‌توانند به صورت پویا و کارآمد در یک اتصال واحد، بدون توقف داده‌ها در انتظار تولیدکنندگان کند قرار گیرند.
- کنترل جریان مبتنی بر اعتبار – داده‌ها با نرخ‌های متفاوتی تولید و مصرف می‌شوند و فرستنده‌ها و گیرندگان مقادیر متفاوتی از حافظه و منابع CPU در دسترس دارند. WebMUX از یک طرح کنترل جریان "مبتنی بر اعتبار" استفاده می‌کند، که در آن گیرندگان علاقه خود را به دریافت داده‌ها برای جلوگیری از بن بست کمبود منابع اعلام می‌کنند.
- Alignment Preserving – تراز داده‌ها در جریان چندگانه حفظ می‌شود تا داده‌های باینری را بتوان به طور موثر ارسال و پردازش کرد.
- Rich Functionality – این رابط به اندازه کافی غنی است که از یک API سوکت پشتیبانی کند.

همچنین شما می‌توانید اطلاعات بیشتری در مورد پروتکل WebMUX را در <http://www.w3.org/Protocols/MUX/WD-mux-980722.html> مطالعه نمایید.

## Binary Wire Protocol

تیم HTTP-NG پروتکل Binary Wire را برای بهبود نحوه پشتیبانی پروتکل HTTP نسل بعدی از عملیات راه دور پیشنهاد کرد.

HTTP-NG، object type ها را تعریف می‌کند و به هر نوع شی فهرستی از متها اختصاص می‌دهد. به هر نوع شی یک URI اختصاص داده می‌شود، بنابراین می‌توان توضیحات و متدهای آن را Advertise



کرد. به این ترتیب، HTTP-NG یک مدل اجرایی توسعه پذیرتر و شی گراتر از مدل ارائه شده با HTTP/1.1 را پیشنهاد می کند، که در آن همه متدها به صورت ایستا در سرورها تعریف شده بودند.

پروتکل Binary Wire درخواست های فراخوانی عملیات را از کلاینت به سرور و پاسخ های نتیجه عملیات را از سرور به کلاینت از طریق یک اتصال stateful حمل می کند. اتصال stateful کارایی بیشتری را فراهم می کند.

پیام های درخواست شامل عملیات، شی مورد نظر و مقادیر داده اختیاری است. پیام های پاسخ وضعیت خاتمه عملیات، شماره سریال درخواست تطبیق (که به ترتیب دلخواه درخواست ها و پاسخ های موازی اجازه می دهد) و مقادیر بازگشتی اختیاری را بازمی گردانند. علاوه بر پیام های درخواست و پاسخ، این پروتکل چندین پیام کنترل داخلی را تعریف می کند که برای بهبود کارایی و استحکام اتصال استفاده می شود.

همچنین شما می توانید در <http://www.w3.org/Protocols/HTTP-NG/1998/08/draft-janssen-httpng-wire-00.txt> درباره پروتکل Binary Wire اطلاعات بیشتری کسب نمایید.

### Current Status

در پایان سال ۱۹۹۸، تیم HTTP-NG به این نتیجه رسید که برای ارائه پیشنهادات HTTP-NG به IETF برای استانداردسازی خیلی زود است. این نگرانی وجود داشت که صنعت و جامعه هنوز به طور کامل با HTTP/1.1 سازگار نشده اند و اینکه معماری مجدد HTTP-NG قابل توجه در الگوی اشیاء توزیع شده بدون یک برنامه انتقال روشن بسیار مخرب بود.

در همین راستا دو پیشنهاد ارائه شد:

به جای تلاش برای ارتقای کل معماری HTTP-NG در یک مرحله، تمرکز بر فناوری حمل و نقل WebMUX پیشنهاد شد. اما در زمان نگارش این مقاله، علاقه کافی برای ایجاد یک گروه کاری WebMUX وجود نداشته است.

تلاشی برای بررسی اینکه آیا انواع پروتکل های رسمی را می توان به اندازه کافی برای استفاده در وب، شاید با استفاده از XML، انعطاف پذیر ساخت یا خیر، آغاز شد. این امر به ویژه برای یک سیستم اشیاء توزیع شده که قابل گسترش است، مهم است. این کار هنوز در حال انجام است.

در زمان نگارش این مقاله، هیچ تلاش مهمی برای هدایت HTTP-NG در حال انجام نیست. اما، با استفاده روزافزون از HTTP، استفاده رو به رشد آن به عنوان پلتفرمی برای کاربردهای متنوع، و پذیرش روزافزون





فناوری اینترنت بی‌سیم و مصرف‌کننده، برخی از تکنیک‌های پیشنهاد شده در HTTP-NG ممکن است در سال‌های نوجوانی HTTP قابل توجه باشد.

### For More Information

<http://www.w3.org/Protocols/HTTP-NG>

<http://www.w3.org/Protocols/MUX/WD-mux-980722.html>

<http://www.w3.org/Protocols/HTTP-NG/1998/08/draft-janssen-httpng-wire-00.txt>

<http://www.w3.org/Protocols/HTTP-NG/1998/08/draft-larner-nginterfaces-00.txt>

<ftp://ftp.parc.xerox.com/pub/ilu/misc/webilu.html>

